

國立臺灣大學工程科學及海洋工程學系

碩士論文

Department of Engineering Science and Ocean Engineering

College of Engineering

National Taiwan University

Master Thesis

3D 列印之路徑規劃演算法

於三軸氣壓式並聯機構機械臂之研究

The Development of Path Planning Algorithm

for 3D Printing

in a Three-Axial Pneumatic Parallel Manipulator

溫志培

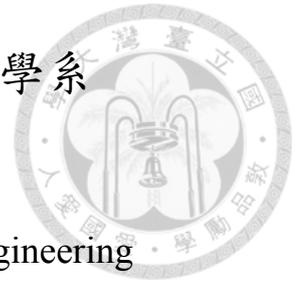
Chih-Pei Wen

指導教授：江茂雄 博士

Advisor: Dr.-Ing. Mao-Hsiung Chiang

中華民國 105 年 1 月

January, 2016



誌謝



本論文的完成承蒙指導教授江茂雄老師的授業解惑，感謝老師在專業領域上的教導，同時培養具獨立思考及解決問題的能力，讓我收穫良多。老師亦時常分享人生經驗及待人處事的道理，讓我以正面態度面對接下來人生的挑戰。在此，學生致上最衷心的感謝與祝福。

感謝口試委員郭振華教授、江茂欽教授、黃金川教授，於百忙之中不吝撥冗，給予本論文提供寶貴的意見與建議，部分見解是學生未曾思考過的問題，非常感謝各位教授，讓本論文更加的完善。

感謝實驗室同窗雲璠、雲傑、喬陽，在這段時間的相互幫助、鼓勵、研究討論，有幸能與你們成為同窗，希冀畢業後仍保持聯絡，延續這難得的緣分。特別感謝學長維新，教導機台的架構與使用方法；特別感謝同窗喬陽，協助模擬軟體的操作與影片錄製；特別感謝學弟俊辰，協助錄製機台的實驗影片。感謝實驗室的學長姐與學弟妹們，在這段時間的幫忙，有幸能認識大家。

感謝求學過程中遇到的同儕，無論僅是課程上分組的組員、舉辦活動的成員，亦或是已成為摯友，彼此互相砥礪、成長、蛻變，讓我認識到不一樣的自己。感謝求學過程中遇到的長輩，無私分享生活、求學、工作的寶貴經驗，讓我在面臨類似情況時，能更有智慧地去面對。

最後，感謝我摯愛的家人們，父親溫春得先生、母親潘美玲女士、弟弟溫博修。感謝從小到大的栽培與照顧，在生活與課業上給予最大的支持與鼓勵，讓我順利完成學業，無後顧之憂，有幸能與你們成為家人是我這輩子最幸福的事，在此獻上我最高的謝意。

謹以此論文，獻給我最敬愛的家人、師長及所有關心我的人。

溫志培

僅誌於台北公館

民國一百零五年一月

中文摘要



本研究旨在發展 3D 列印的軌跡規劃演算法並應用於三軸氣壓式並聯機構機械臂，置重點於 3D 列印的軌跡規劃於三軸氣壓式並聯機構機械臂，結合實驗室已發展之三軸氣壓式並聯機構機械臂之運動學分析與控制器設計，以模擬及實際實驗驗證。

在 3D 列印的軌跡規劃方面，將欲列印的物體，採用圖論(Graph Theory)的向量形式建立。透過深度優先搜尋(Depth-First Search, DFS)定義一個平面的所有分歧路徑，並由基因演算法(Genetic Algorithm, GA)計算如何以最低代價連接所有分歧路徑。最後將每個平面的路徑串接，即可得軌跡規劃。

在三軸氣壓式並聯機構機械臂的運動學分析方面，採用幾何向量的理論與空間中向量迴圈的封閉性質，透過逆向與順向運動學的定義分別推導出致動器與運動平台的關係。

在三軸氣壓式並聯機構機械臂的控制器設計方面，單軸氣壓伺服系統採用雙迴圈回授控制策略，其中包含內圈的壓力控制與外圈的位置控制。根據上述方法，並額外採用逆向動力學控制策略，以實現三軸氣壓式並聯機構機械臂的控制與解決三軸的非線性耦合。

在本論文最後，透過數值模擬，檢測三軸氣壓式並聯機構機械臂之推導模型與 3D 列印之軌跡規劃的正確性。為證明實用性，藉由實驗室已建立之三軸氣壓式並聯機構機械臂實驗系統的實驗，輸入與數值模擬相同的軌跡，驗證控制器的效能與 3D 列印整合三軸氣壓式並聯機構機械臂的可行性。

關鍵詞：3D 列印、軌跡規劃、深度優先搜尋(DFS)、基因演算法(GA)、氣壓伺服系統、並聯式機構機械臂、運動學分析、軌跡追蹤控制

ABSTRACT



This study aims to develop 3D-printing path planning algorithms and applies to a three-axial pneumatic parallel manipulator. The emphasis is on the research of 3D-printing path planning algorithms, integrating the three-axial pneumatic parallel manipulator which has developed on its kinematic analysis and controller design in lab before, and verifying the performance through the whole system simulations and experiments.

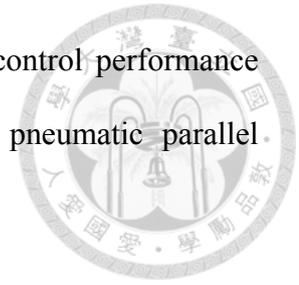
In path planning algorithms for 3D printing, the desired-printing object was established from graph theory as vector form. From the view of a layer, all sub-paths are defined through the depth-first search, and the genetic algorithm is used to find the minimum costs linking sub-paths. After cascading all layers, the overall path is accomplished.

In analysis of kinematics, the geometric method is introduced to solve the relation of manipulator between actuated joints and moving platform through vector-loop closure equations, including inverse and forward kinematics.

In controller design, control strategy of single-axial pneumatic servo system is applied with dual-loop feedback control scheme, i.e. inner pressure control and outer position control. Based on that, controller of three-axial pneumatic parallel manipulator is established with extra inverse dynamics control strategy to decouple the nonlinear terms.

Finally, numerical simulations are carried out to verify the correctness of the derived models and the path-planning trajectories. To show the practicality, real-time experiments are implemented in the test rig of three-axial pneumatic parallel mechanism

robot with the same trajectories in simulations for testifying the control performance and the possibility of 3D printing integrating with three-axial pneumatic parallel manipulator.



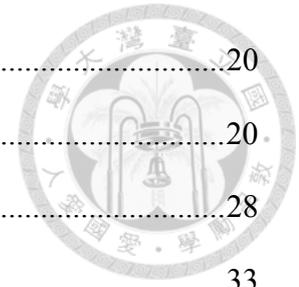
Keywords: 3D printing, path planning, depth-first search (DFS), genetic algorithm (GA), pneumatic servo system, parallel manipulator, kinematic analysis, path tracking control

CONTENTS

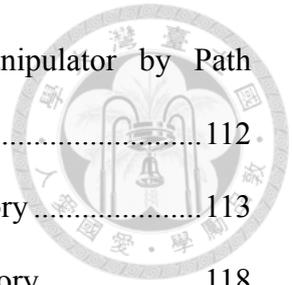


口試委員會審定書	#
誌謝	i
中文摘要	ii
ABSTRACT	iii
CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	xix
Chapter 1 Introduction	1
1.1 Background	1
1.2 Literature Review	3
1.2.1 Parallel Manipulator	3
1.2.2 Pneumatic Servo System	4
1.2.3 3D Printing Technology	5
1.3 Motivation	6
1.4 Outline of Thesis	7
Chapter 2 System Overview	8
2.1 Mechanism Description	8
2.2 Test Rig Layout	11
2.2.1 Pneumatic Servo Positioning System	11
2.2.2 Overall Manipulator System	12
Chapter 3 Path Planning Algorithms for 3D Printing	15
3.1 Path Planning Model	15

3.2	Single-Layer Path Planning Strategy	20
3.2.1	Path Generation	20
3.2.2	Path Optimization	28
3.3	Layer-to-Layer Path Planning Strategy	33
Chapter 4	Analysis of Kinematics	36
4.1	Geometry of the Manipulator	36
4.2	Analysis of Inverse Kinematics	40
4.3	Analysis of Forward Kinematics	41
Chapter 5	Controller Design	43
5.1	Control Strategy of the Single-Axial Pneumatic Servo System	43
5.2	Control Strategy of the Three-Axial Pneumatic Parallel Manipulator	47
Chapter 6	Simulations and Experiments	50
6.1	Verifications of Kinematic Model	51
6.1.1	Circle-shape Line Trajectory	53
6.1.2	Sphere-shape Line Trajectory	57
6.1.3	Solid Cuboid Trajectory	61
6.1.4	Solid Polyhedron Trajectory	72
6.1.5	Solid Complex 3D Trajectory	83
6.2	Simulations of Three-Axial Pneumatic Parallel Manipulator by ADAMS and SIMULINK	86
6.2.1	Simulation of Circle-shape Line Trajectory	87
6.2.2	Simulation of Sphere-shape Line Trajectory	92
6.2.3	Simulation of Solid Cuboid Trajectory	97
6.2.4	Simulation of Solid Polyhedron Trajectory	102
6.2.5	Simulation of Solid Complex 3D Trajectory	107



6.3	Experiments of Three-Axial Pneumatic Parallel Manipulator by Path Tracking Control of End-Effector.....	112
6.3.1	Experimental Results of Circle-shape Line Trajectory.....	113
6.3.2	Experimental Results of Sphere-shape Line Trajectory.....	118
6.3.3	Experimental Results of Solid Cuboid Trajectory	123
6.3.4	Experimental Results of Solid Polyhedron Trajectory.....	128
6.3.5	Experimental Results of Solid Complex 3D Trajectory.....	133
Chapter 7	Conclusions.....	138
	REFERENCES.....	140



LIST OF FIGURES



Fig. 2.1	Three-axial pyramidal pneumatic parallel manipulator.....	9
Fig. 2.2	Joint-link configuration of the parallel manipulator.....	9
Fig. 2.3	Photograph of the three-axial pyramidal pneumatic parallel manipulator ...	10
Fig. 2.4	Test rig layout of the pneumatic servo positioning system.....	11
Fig. 2.5	Test rig layout of the overall manipulator system	13
Fig. 3.1	Slicing a desired printing object equally	16
Fig. 3.2	The definition of layers.....	16
Fig. 3.3	Equal-sized rectangular mesh on a layer: (a) front view (b) top view.....	17
Fig. 3.4	The position of a point (red dot) substitutes component mesh.....	17
Fig. 3.5	A directed graph with 5 vertices and 5 edges	18
Fig. 3.6	Double-directions directed graph on a layer before path planning.....	19
Fig. 3.7	Single-direction directed graph between layers before path planning.....	19
Fig. 3.8	The priorities of traversing directions (1 is the highest priority).....	21
Fig. 3.9	The linkages between each two points based on traversing directions: (a) 6 points (b) double-directions linking strategy.....	21
Fig. 3.10	A general example of modified-DFS.....	24
Fig. 3.11	The exception in modified-DFS: (a) 11 points (b) some un-traversed points after modified-DFS.....	26
Fig. 3.12	Solution of the exception in modified-DFS: (a) selecting a new start point nearest to the start point of Sub-path 1 (b) using modified-DFS again and checking the counter when completed.....	26
Fig. 3.13	A complete example of modified-DFS including the exception: (a) 45 points	

	(b) 4 sub-paths after modified-DFS.....	27
Fig. 3.14	Notations of sub-paths for Genetic Algorithm	28
Fig. 3.15	An example of two initial populations	29
Fig. 3.16	An example of ideal shortest total lengths.....	29
Fig. 3.17	An example of crossover	30
Fig. 3.18	An example of mutation at Sub-path 2.....	30
Fig. 3.19	An example of the exception in step D (i).....	32
Fig. 3.20	An example of the exception in step D (ii).....	32
Fig. 3.21	Solution of the exception at Sub-path 1 in step E.....	32
Fig. 3.22	Path planning on a layer: (a) 9 points (b) planned path trajectory.....	33
Fig. 3.23	Layer-to-layer path planning on a 3-layers object.....	34
Fig. 3.24	The exception in layer-to-layer path planning: (a) after single-layer path planning on Layer i (b) a non-existent start point (purple dot) on Layer $i+1$	35
Fig. 3.25	Solution of the exception in layer-to-layer path planning: (a) creating the non-existent start point on Layer $i+1$ (b) linking to nearest point and changing the start point on Layer $i+1$	35
Fig. 4.1	Schematic diagram of the three-axial pyramidal parallel manipulator.....	37
Fig. 4.2	Geometry of one typical limb.....	37
Fig. 4.3	Vector representation for one typical limb	38
Fig. 5.1	Dual-loop control scheme for position control of the pneumatic cylinder system	45
Fig. 5.2	PI control with back-calculation anti-windup method.....	46
Fig. 5.3	Schematic diagram of position control of single-axial pneumatic cylinder system	46

Fig. 5.4	Inner-loop control design of inverse dynamics control for the three-axial pyramidal pneumatic parallel manipulator	48
Fig. 5.5	Schematic diagram of inverse dynamics control for the three-axial pyramidal pneumatic parallel manipulator	48
Fig. 5.6	Schematic diagram of the overall control system.....	49
Fig. 6.1	Procedure of model verification for inverse and forward kinematics	51
Fig. 6.2	Designed circle-shape line trajectory for end-effector	53
Fig. 6.3	Calculated joint space trajectories via inverse kinematics for the circle-shape line trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator ..	54
Fig. 6.4	Calculated task space trajectories via forward kinematics for the circle-shape line trajectory: (a) X-axis (b) Y-axis (C) Z-axis	54
Fig. 6.5	Comparison between the planned and the calculated end-effector for the circle-shape line trajectory: (a) front view (b) top view	55
Fig. 6.6	Numerical error between the planned and the calculated end-effector for the circle-shape line trajectory: (a) X-axis (b) Y-axis and (C) Z-axis	56
Fig. 6.7	Designed sphere-shape line trajectory for end-effector.....	57
Fig. 6.8	Calculated joint space trajectories via inverse kinematics for the sphere-shape line trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator	58
Fig. 6.9	Calculated task space trajectories via forward kinematics for the sphere-shape line trajectory: (a) X-axis (b) Y-axis (C) Z-axis.....	58
Fig. 6.10	Comparison between the planned and the calculated end-effector for the sphere-shape line trajectory: (a) front view (b) top view	59
Fig. 6.11	Numerical error between the planned and the calculated end-effector for the sphere-shape line trajectory: (a) X-axis (b) Y-axis and (C) Z-axis.....	60

Fig. 6.12	Desired printing object of solid cuboid	61
Fig. 6.13	Desired printing object model of solid cuboid: (a) all points (b) after path planning	63
Fig. 6.14	Path planning for solid cuboid trajectory on the 1st layer ($z = -300$ mm): (a) after modified-DFS (b) after modified-GA	64
Fig. 6.15	Path planning for solid cuboid trajectory on the 2nd layer ($z = -270$ mm): (a) after modified-DFS (b) after modified-GA	65
Fig. 6.16	Path planning for solid cuboid trajectory on the 3rd layer ($z = -240$ mm): (a) after modified-DFS (b) after modified-GA	66
Fig. 6.17	Calculated joint space trajectories via inverse kinematics for the solid cuboid trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator	67
Fig. 6.18	Calculated task space trajectories via forward kinematics for the solid cuboid trajectory: (a) X-axis (b) Y-axis (C) Z-axis.....	67
Fig. 6.19	Comparison between the planned and the calculated end-effector for the solid cuboid trajectory.....	68
Fig. 6.20	Numerical error between the planned and the calculated end-effector for the solid cuboid trajectory: (a) X-axis (b) Y-axis and (C) Z-axis.....	68
Fig. 6.21	Comparison between the planned and the calculated end-effector for the solid cuboid trajectory of the 1st layer ($z = -300$ mm): (a) front view (b) top view	69
Fig. 6.22	Comparison between the planned and the calculated end-effector for the solid cuboid trajectory of the 2nd layer ($z = -270$ mm): (a) front view (b) top view	70
Fig. 6.23	Comparison between the planned and the calculated end-effector for the solid cuboid trajectory of the 3rd layer ($z = -240$ mm): (a) front view (b) top	

	view	71
Fig. 6.24	Desired printing object of solid polyhedron	72
Fig. 6.25	Desired printing object model of solid polyhedron: (a) all points (b) after path planning	74
Fig. 6.26	Path planning for solid polyhedron trajectory on the 1st layer ($z = -300$ mm): (a) after modified-DFS (b) after modified-GA	75
Fig. 6.27	Path planning for solid polyhedron trajectory on the 2nd layer ($z = -270$ mm): (a) after modified-DFS (b) after modified-GA	76
Fig. 6.28	Path planning for solid polyhedron trajectory on the 3rd layer ($z = -240$ mm): (a) after modified-DFS (b) after modified-GA	77
Fig. 6.29	Calculated joint space trajectories via inverse kinematics for the solid polyhedron trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator	78
Fig. 6.30	Calculated task space trajectories via forward kinematics for the solid polyhedron trajectory: (a) X-axis (b) Y-axis (C) Z-axis	78
Fig. 6.31	Comparison between the planned and the calculated end-effector for the solid polyhedron trajectory.	79
Fig. 6.32	Numerical error between the planned and the calculated end-effector for the solid polyhedron trajectory: (a) X-axis (b) Y-axis and (C) Z-axis.....	79
Fig. 6.33	Comparison between the planned and the calculated end-effector for the solid polyhedron trajectory of the 1st layer ($z = -300$ mm): (a) front view (b) top view	80
Fig. 6.34	Comparison between the planned and the calculated end-effector for the solid polyhedron trajectory of the 2nd layer ($z = -270$ mm): (a) front view (b) top view.....	81

Fig. 6.35	Comparison between the planned and the calculated end-effector for the solid polyhedron trajectory of the 3rd layer ($z = -240$ mm): (a) front view (b) top view	82
Fig. 6.36	Desired printing object of solid complex 3D: (a) front view (b) side view (c) top view	83
Fig. 6.37	Calculated joint space trajectories via inverse kinematics for the solid complex 3D trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator	84
Fig. 6.38	Calculated task space trajectories via forward kinematics for the solid complex 3D trajectory: (a) X-axis (b) Y-axis (C) Z-axis.....	84
Fig. 6.39	Comparison between the planned and the calculated end-effector for the solid complex 3D trajectory.....	85
Fig. 6.40	Numerical error between the planned and the calculated end-effector for the solid complex 3D trajectory: (a) X-axis (b) Y-axis and (C) Z-axis	85
Fig. 6.41	Schematic diagram of the three-axial closed-loop path tracking control	86
Fig. 6.42	SIMULINK model of the three-axial closed-loop system.....	86
Fig. 6.43	Designed circle-shape line trajectory for end-effector	87
Fig. 6.44	Simulation results of end-effector path tracking control for A-axis cylinder via a circle-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	88
Fig. 6.45	Simulation results of end-effector path tracking control for B-axis cylinder via a circle-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	89
Fig. 6.46	Simulation results of end-effector path tracking control for C-axis cylinder via a circle-shape line trajectory: (a) position tracking response (b) position	

	tracking error (c) control signal	90
Fig. 6.47	Estimated end-effector position tracking response for a circle-shape line trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	91
Fig. 6.48	Designed sphere-shape line trajectory for end-effector.....	92
Fig. 6.49	Simulation results of end-effector path tracking control for A-axis cylinder via a sphere-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	93
Fig. 6.50	Simulation results of end-effector path tracking control for B-axis cylinder via a sphere-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	94
Fig. 6.51	Simulation results of end-effector path tracking control for C-axis cylinder via a sphere-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	95
Fig. 6.52	Estimated end-effector position tracking response for a sphere-shape line trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	96
Fig. 6.53	Desired printing object of solid cuboid	97
Fig. 6.54	Simulation results of end-effector path tracking control for A-axis cylinder via a solid cuboid trajectory: (a) position tracking response (b) position tracking error (c) control signal	98
Fig. 6.55	Simulation results of end-effector path tracking control for B-axis cylinder via a solid cuboid trajectory: (a) position tracking response (b) position tracking error (c) control signal	99
Fig. 6.56	Simulation results of end-effector path tracking control for C-axis cylinder	

	via a solid cuboid trajectory: (a) position tracking response (b) position tracking error (c) control signal	100
Fig. 6.57	Estimated end-effector position tracking response for a solid cuboid trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	101
Fig. 6.58	Desired printing object of solid polyhedron	102
Fig. 6.59	Simulation results of end-effector path tracking control for A-axis cylinder via a solid polyhedron trajectory: (a) position tracking response (b) position tracking error (c) control signal	103
Fig. 6.60	Simulation results of end-effector path tracking control for B-axis cylinder via a solid polyhedron trajectory: (a) position tracking response (b) position tracking error (c) control signal	104
Fig. 6.61	Simulation results of end-effector path tracking control for C-axis cylinder via a solid polyhedron trajectory: (a) position tracking response (b) position tracking error (c) control signal	105
Fig. 6.62	Estimated end-effector position tracking response for a solid polyhedron trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	106
Fig. 6.63	Desired printing object of solid complex 3D.....	107
Fig. 6.64	Simulation results of end-effector path tracking control for A-axis cylinder via a solid complex 3D trajectory: (a) position tracking response (b) position tracking error (c) control signal	108
Fig. 6.65	Simulation results of end-effector path tracking control for B-axis cylinder via a solid complex 3D trajectory: (a) position tracking response (b) position tracking error (c) control signal	109

Fig. 6.66	Simulation results of end-effector path tracking control for C-axis cylinder via a solid complex 3D trajectory: (a) position tracking response (b) position tracking error (c) control signal	110
Fig. 6.67	Estimated end-effector position tracking response for a solid complex 3D trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	111
Fig. 6.68	Schematic diagram of the overall control system.....	112
Fig. 6.69	Designed circle-shape line trajectory for end-effector	113
Fig. 6.70	Experimental results of end-effector path tracking control for A-axis cylinder via a circle-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	114
Fig. 6.71	Experimental results of end-effector path tracking control for B-axis cylinder via a circle-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	115
Fig. 6.72	Experimental results of end-effector path tracking control for C-axis cylinder via a circle-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	116
Fig. 6.73	Estimated end-effector position tracking response for a circle-shape line trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	117
Fig. 6.74	Designed sphere-shape line trajectory for end-effector.....	118
Fig. 6.75	Experimental results of end-effector path tracking control for A-axis cylinder via a sphere-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	119
Fig. 6.76	Experimental results of end-effector path tracking control for B-axis	

	cylinder via a sphere-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	120
Fig. 6.77	Experimental results of end-effector path tracking control for C-axis cylinder via a sphere-shape line trajectory: (a) position tracking response (b) position tracking error (c) control signal	121
Fig. 6.78	Estimated end-effector position tracking response for a sphere-shape line trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	122
Fig. 6.79	Desired printing object of solid cuboid	123
Fig. 6.80	Experimental results of end-effector path tracking control for A-axis cylinder via a solid cuboid trajectory: (a) position tracking response (b) position tracking error (c) control signal	124
Fig. 6.81	Experimental results of end-effector path tracking control for B-axis cylinder via a solid cuboid trajectory: (a) position tracking response (b) position tracking error (c) control signal	125
Fig. 6.82	Experimental results of end-effector path tracking control for C-axis cylinder via a solid cuboid trajectory: (a) position tracking response (b) position tracking error (c) control signal	126
Fig. 6.83	Estimated end-effector position tracking response for a solid cuboid trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	127
Fig. 6.84	Desired printing object of solid polyhedron	128
Fig. 6.85	Experimental results of end-effector path tracking control for A-axis cylinder via a solid polyhedron trajectory: (a) position tracking response (b) position tracking error (c) control signal	129

Fig. 6.86	Experimental results of end-effector path tracking control for B-axis cylinder via a solid polyhedron trajectory: (a) position tracking response (b) position tracking error (c) control signal	130
Fig. 6.87	Experimental results of end-effector path tracking control for C-axis cylinder via a solid polyhedron trajectory: (a) position tracking response (b) position tracking error (c) control signal	131
Fig. 6.88	Estimated end-effector position tracking response for a solid polyhedron trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	132
Fig. 6.89	Desired printing object of solid complex 3D.....	133
Fig. 6.90	Experimental results of end-effector path tracking control for A-axis cylinder via a solid complex 3D trajectory: (a) position tracking response (b) position tracking error (c) control signal	134
Fig. 6.91	Experimental results of end-effector path tracking control for B-axis cylinder via a solid complex 3D trajectory: (a) position tracking response (b) position tracking error (c) control signal	135
Fig. 6.92	Experimental results of end-effector path tracking control for C-axis cylinder via a solid complex 3D trajectory: (a) position tracking response (b) position tracking error (c) control signal	136
Fig. 6.93	Estimated end-effector position tracking response for a solid complex 3D trajectory: (a) calculated end-effector position (b) calculated end-effector position error.....	137

LIST OF TABLES



Table 2.1	Specifications of system hardware	14
Table 4.1	Geometry parameters of the manipulator	39
Table 5.1	Model parameters of the pneumatic servo system.....	44
Table 6.1	Manipulator parameters for kinematic simulations	52

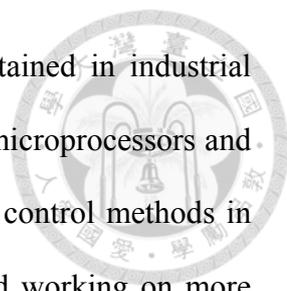
Chapter 1 Introduction



1.1 Background

Robotic manipulators are mighty machines that can achieve various desired movements. Generally, robotic manipulators are divided into two types with respect to their kinematic structures, such as the serial type and the parallel type. The serial manipulator is designed as a series of links which are sequentially connected by actuated joints from a base to an end-effector. The arm-like structure design shows high flexibility on larger-scope operation. However, the open-chain mechanism results in lower positioning accuracy affected by the error superposition of each joint and link, and poor stiffness in handling heavy loads. On the other hand, the parallel manipulator contains multiple closed-loops which consist of several independent kinematic chains connecting a moving platform to a fixed base. The closed-loop mechanism brings the advantages of high stiffness, low inertia and high speed capability. Also, the actuators, the drives, usually positioned on or nearby the fixed base, allow the mechanism of links to be lighter and lead to high rigidity-to-weight ratio. Moreover, in positioning accuracy, the position errors in one single kinematic chain can be averaged by the other chains instead of being accumulative. The only drawbacks are their limited workspace, complex kinematic analysis and extreme difficulty in control design. In recent years, the heavy demands for high speed, high precision and good stiffness have made parallel manipulators win a place in industrial automation.

Pneumatic actuators are powerful mechanical devices that use compressed air as their operating fluid to produce driving force and motion for the payloads. Low cost is the primary reason in industrial applications especially for linear motion. Also, the high power-to-weight ratio is another favorable feature particularly in robotic manipulators.



Furthermore, pneumatic actuators are clean, safe and easily maintained in industrial environment. Over the past few years, the accessibility of low cost microprocessors and pneumatic components has made it possible to use more advanced control methods in pneumatic servo systems. Many researchers, therefore, have started working on more complicated motion control tasks. Comparing to electrical motors with identical power, pneumatic actuators are not competitive in few applications which demand accuracy, versatility, and flexibility. This is due to inherent disadvantages of pneumatic actuators including compressibility of air, high nonlinearity, high friction force, air leakage, lower natural frequency and high complexity in control; nevertheless, researches on robots using pneumatic systems are still popular and have potential for practical applications.

3D printing technology has created a lot of discussions in recent years. Kind of like an evolution of Rapid Prototyping, they not only function as manufacturing prototypes but also apply in many fields such as medical science, amusement and architecture. The most attractive thing is people can rapidly implement any innovative ideals from flat screen to exact object, because of short manufacturing time, and make specialized. 3D printing can provide great savings on assembly costs because of all-in-one prints; meanwhile, it can experiment numerous design iterations without tooling expense and testify the practicability of product concepts. Furthermore, it is possible to challenge mass production method in the future. Besides, various choices of colors and materials, which can be obtained as powder, bring finished prints much diversity. Lately, in the efforts of many projects and companies, 3D printers are more affordable and delicate for home desktop use. However, due to layer-by-layer manufacturing, sometimes it has jagged edges between layers according to resolution and, therefore, needs a smoothing procedure. Nowadays, 3D printing has impacted on many industries, and researchers still devote to extend the possibility of applications.



1.2 Literature Review

1.2.1 Parallel Manipulator

The first parallel robot is an amusement device designed by James E. Gwinnett in 1928. After a decade, a parallel robot, an automated spray painting, was invented by Willard L.V. Pollard. In 1954, the first octahedral hexapod was built for tire-testing by Gough. In 1965, a motion platform with six degrees of freedom (DOF) was designed by D. Stewart [1], becoming the famous Stewart platform. Due to the flaws in six-limbed parallel manipulators, such as complex kinematic analysis and motion coupling, many researchers focused on development of less than six degrees of freedom recently. In 1988, a 3-DOF parallel manipulator, called DELTA robot, was invented by research team leader Reymond Clavel [2]. Since then, the tripod mechanism parallel manipulator with three degrees of freedom has been extensively studied. Closed-form solutions for both inverse and forward kinematics have been developed for the DELTA robot by Pierrot *et al.* [3]. The dynamic model of DELTA robot for control implementation was also developed by Codourey [4]. In 1996, Tsai *et al.* [5] introduced a novel 3-DOF translational platform made up of only revolute joints. Many other 3-DOF parallel manipulators with different structures and configurations have been designed for relevant applications lately, for instance, spherical 3-DOF mechanisms, 3-PRS parallel manipulators and orthoglide parallel robots [6], [7], [8], [9]. In 2005, a serial-parallel hybrid robot for construction works with pneumatic actuator was developed by Choi *et al.* [10]. In 2011, Chiang and Lin developed a parallel manipulator driven by three vertical-axial pneumatic actuators [11]. In 2012, Chiang *et al.* developed two different structural 3-PUU parallel manipulators driven by pneumatic rodless cylinders and implemented in path tracking servo control and 3D stereo measuring system [12], [13].

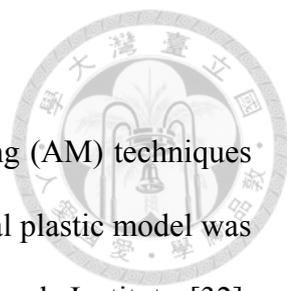
1.2.2 Pneumatic Servo System

The earliest research on pneumatic system was made by J. L. Shearer in 1956. He derived a set of nonlinear differential equations to describe the dynamics of a pneumatic servo system. Since then, many researchers developed complete nonlinear mathematical models for the pneumatic servo system such as the work of Ben-Dov and Salcudean [14], Richer and Hurmuzlu [15], and Wang *et al.* [16]. These early studies established the principles for the understanding and control of the pneumatic servo system.

Recently, pneumatic servo systems have been used on many complex tasks and found suitable for robotic field as presented by Bobrow and McDonell [17], [18] and Moran *et al.* [19]. To conquer the high nonlinearity, low accuracy and low robustness, numerous control strategies have been proposed over the past years. Early works done by Liu and Bobrow [20] used a linearized state space model to develop an optimal regulator for a fixed operating point. The position control of pneumatic servo system using pressure control loop can be found in Noritsugu *et al.* [21] and Lee *et al.* [22]. The adaptive control of pneumatic servo system was mentioned in McDonell and Bobrow [23], Tanaka *et al.* [24], [25], and Li *et al.* [26].

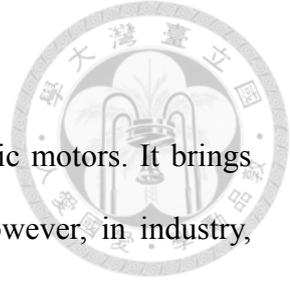
Thanks to great progress in modern nonlinear control theories [27], these tricky problems can also be solved by robust control approaches called sliding mode control (SMC) [28], [29]. But the conventional SMC method is a model-based approach and led to the system model the time-varying and uncertain parameters when deriving a controller. To deal with these issues, Huang *et al.* [30] suggested an adaptive sliding controller, by a functional approximation technique, to handle a nonlinear system containing time-varying and uncertain parameters. Chiang *et al.* [31] proposed a Fourier series based adaptive sliding mode controller with H_∞ tracking performance and applied in position control of rodless pneumatic cylinder systems.

1.2.3 3D Printing Technology



The early 3D-printing studies were from additive manufacturing (AM) techniques in 1980s. In 1981, two AM fabricating method of a three-dimensional plastic model was invented by Hideo Kodama of Nagoya Municipal Industrial Research Institute [32], [33]. In 1984, Charles Hull, who founded 3D Systems, created a process called Stereolithography [34], an AM technique, to establish a prototype system. In 1988, Fused Deposition Modeling (FDM), an AM technique, was invented by Scott Crump who founded Stratasys and sold first FDM-based machine named "3D Modeler" [35], [36]. In 1993, Massachusetts Institute of Technology (MIT) patented a technology called "3 Dimensional Printing techniques" [37] which Z Corporation developed 3D Printers [38] based on, in 1995. In 1996, the term "3D Printer" was first used to refer rapid prototyping machines because of three major products from Stratasys, 3D Systems, and Z Corporation. Since then, several relatively 3D Printers came into the market. In 2005, Z Corporation launched a first high-definition color 3D Printer, Spectrum Z510. In 2006, a well-known project, Reprap Project, which consists of hundreds of collaborators, was aimed to develop a self-replicating low-cost 3D printer [39], [40].

One of 3D-printing manufacturing stages is path planning which has a remarkable impact on overall printing time. The path planning problem includes path generation and path optimization. The perfect case is each vertex on a plane only passes once, called Hamiltonian Circle. The term was from Icosian Game, a mathematical game, invented in 1857 by W. R. Hamilton. However, this scenario merely happened; many researches proposed solving path generation issues, such as ZigZag [41], Contour [42], and Spiral [43], and developed path optimization strategies, such as Combination of Neural Networks (NN) and Genetic Algorithms (GA) [44], [45], and Adaptation of Travelling Salesman Problem (TSP) [46].



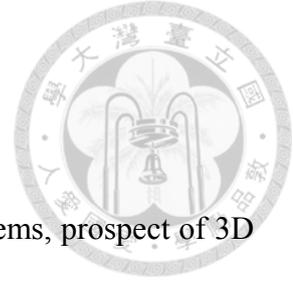
1.3 Motivation

The common commercialized 3D printers are driven by electric motors. It brings users small size and smooth printing because of easy control. However, in industry, pneumatic systems are widely applied owing to low cost and high power-to-weight ratio. Combination of 3D printing concepts and pneumatic system is quite prospective and offers more potential than Rapid Prototyping.

Path planning is crucial to 3D printing and dominates almost overall printing time. Based on the graph theory, a sliced 3D object becomes being composed of vertices in each layer; path generation and optimization help to plan trajectories through all vertices at least once with minimum costs. Besides, the kinematic analysis is useful to build the overall manipulator model. After using the geometric method, the solutions for both the inverse and forward kinematics are obtained by solving the vector-loop equations. Based on the overall manipulator model and intrinsic characteristics of pneumatic actuator system, the proposed control design has a cascade structure with inner and outer feedback control loops. Numerical simulations are used to validate the derived models and the visions of planned trajectories of 3D printing. On the other hand, real-time experiments show the abilities of controller and manipulator; meanwhile, the feasibility of pneumatic-driven 3D printer is verified.

This study integrates the path planning algorithms for 3D printing and the three-axial pneumatic parallel manipulator developed in our lab, AFPCL, instead of the electric motor driven. The goal is to develop algorithms that make the manipulator achieve functions as a 3D printer for verifying the efficiency and accuracy through experiments.

1.4 Outline of Thesis



Chapter 1: Introduction

Introduction of parallel manipulators and pneumatic servo systems, prospect of 3D printing technology, literature review, motivation

Chapter 2: System Overview

Mechanism description, test rig layouts of pneumatic servo positioning system and overall manipulator system

Chapter 3: Path Planning Algorithms for 3D Printing

Introduction to path planning model, path planning strategy for a layer in path generation and optimization, path planning strategy for layer to layer

Chapter 4: Analysis of Kinematics

Illustration of manipulator geometry, introduction of geometric method, derivations of inverse and forward kinematics

Chapter 5: Controller Design

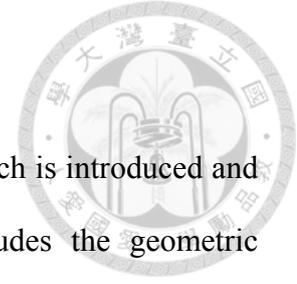
Control strategies of the single-axial pneumatic servo system and three-axial pneumatic parallel manipulator system

Chapter 6: Simulations and Experiments

Verifications of kinematic model, simulations of three-axial pneumatic parallel manipulator by ADAMS and SIMULINK, experiments of three-axial pneumatic parallel manipulator by path tracking control

Chapter 7: Conclusions

Chapter 2 System Overview



In this chapter, the proposed parallel manipulator of this research is introduced and described. The description of the manipulator mechanism includes the geometric structure and the linkage configuration of the parallel manipulator. The layout of the test rig of the manipulator system, including the experimental setup and the operating principle of both the pneumatic servo subsystem and the overall integrated manipulator system, is presented and illustrated in this chapter. The system hardware which contains pneumatic components, sensory devices and a PC-based controller will be listed and described in detail. In addition, a software interface used to execute the control algorithm and monitor the output data in real time will be introduced.

2.1 Mechanism Description

The proposed manipulator is basically composed of three identical limbs, a fixed base, and a moving platform. The structure of the proposed parallel manipulator is shown in Fig. 2.1. A reference frame (x - y - z) is attached to the fixed base at point O . The three identical limbs labeled as A, B and C are connected the moving platform to the stationary base in parallel. Each limb consists of a linear guide-way, a slider which is also the input link, and a pair of parallel-aligned kinematic links. The axes of the linear guide-ways are assembled and connected to the base in the way that the geometric structure of the manipulator is in an inverted pyramidal shape.

Fig. 2.2 shows the joint-link configuration of the manipulator. The three sliders, driven by the pneumatic rodless cylinders, are translated along the linear guide-ways by three one degree of freedom (DOF) prismatic joints. For each limb, a set of parallel kinematic chain connects the slider and the moving platform. The parallel kinematic

chain is assembled by two carbon fiber rods whose ends are linked to the slider and the moving platform by four 3-DOF spherical joints (ball joints).

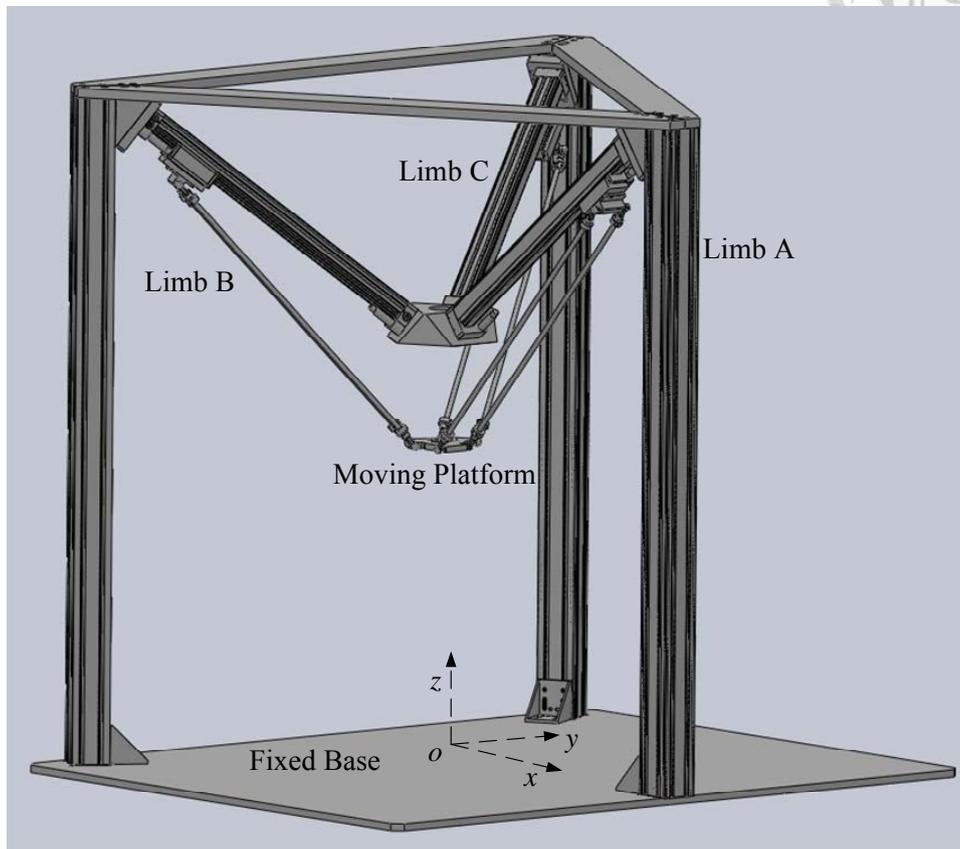
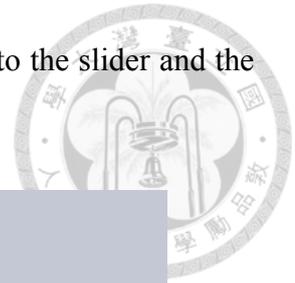


Fig. 2.1 Three-axial pyramidal pneumatic parallel manipulator

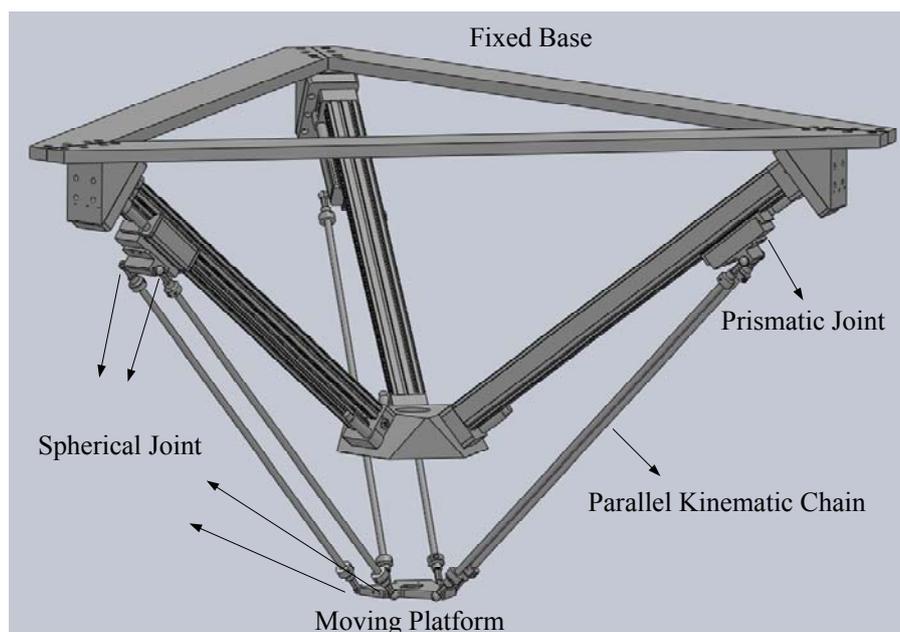


Fig. 2.2 Joint-link configuration of the parallel manipulator

According to the arrangement of the parallel links and the spherical joints in each limb, there exist the so-called "passive degrees of freedom" in the manipulator system. Because these passive DOFs do not provide extra motion to the moving platform and do not increase the mobility of the manipulator, the two spherical joint pairs at the upper and lower ends of the parallel kinematic chain function as two single 2-DOF universal joints and can be seen as a P-U-U (Prismatic-Universal-Universal) configuration. Thus, the overall configuration results in a 3-PUU mechanism in accordance with [14].

Note that the only actuated joints of the manipulator are the three prismatic joints and all spherical joints are passive joints. Besides, the structural characteristics and linkage configuration are similar to famous DELTA parallel robots, and can be classified as a linear-type DELTA robot from [48]. The photograph of the three-axial pyramidal pneumatic parallel manipulator developed in this research is shown in Fig. 2.3.



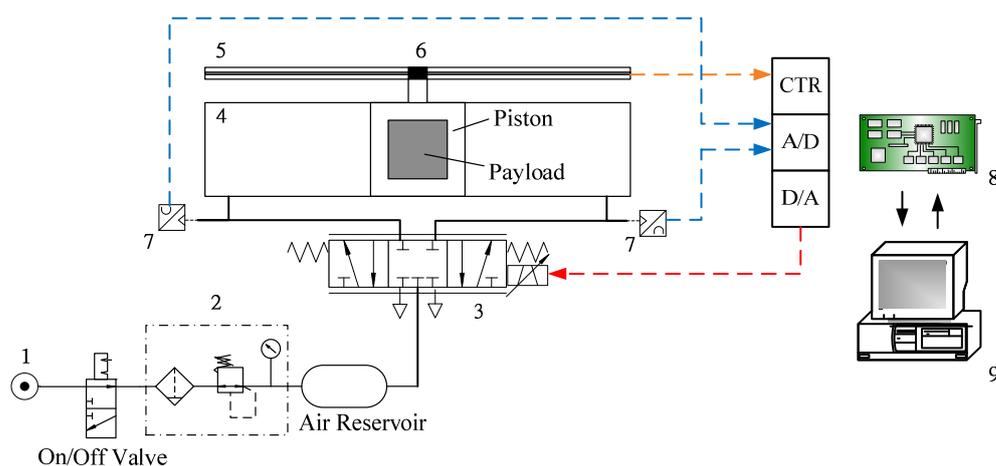
Fig. 2.3 Photograph of the three-axial pyramidal pneumatic parallel manipulator



2.2 Test Rig Layout

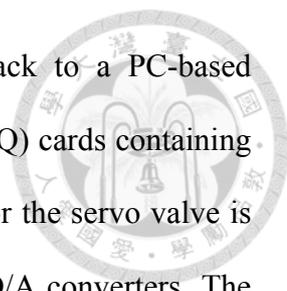
2.2.1 Pneumatic Servo Positioning System

The test rig layout of a pneumatic servo positioning system is shown in Fig. 2.4, which illustrates the single-axial pneumatic actuator system of the manipulator. The pneumatic actuator system comprises an air pressure source, a proportional servo valve and a pneumatic rodless cylinder. The pressure source is provided by an air compressor made by Taiwan Co Sheng, and the supplied air pressure is regulated at 6 bar. The servo valve is a 5/3-way proportional directional control valve made by Festo AG (model MPYE-5-M5) and is used to control the air flowing into the cylinder. A pneumatic rodless cylinder with 25 mm bore and 500 mm stroke (Festo model DGC-25-500) is used as the linear actuator. An optical linear encoder with 1 μm resolution is used as the position sensor and installed on the cylinder to measure the piston's position. Two pressure sensors (Festo model SDE1) are connected to the two ports of the cylinder and used to measure the pressures of the two cylinder chambers.



1. Pressure Source	2. Pressure Regulator	3. Proportional Directional Control Valve
4. Pneumatic Rodless Cylinder	5. Optical Linear Scale	6. Position Sensor
7. Pressure Sensor	8. Interface Card	9. PC-Based Controller

Fig. 2.4 Test rig layout of the pneumatic servo positioning system



In the closed-loop system, the measured signals are fed back to a PC-based controller via the interface cards which are the data acquisition (DAQ) cards containing counters (CTR) and A/D converters. The input command voltage for the servo valve is given from the analogue output ports on the DAQ cards via the D/A converters. The control system is implemented on a Windows-based personal computer. The algorithms for the control system are created and built up in a Simulink model by Matlab software, and the Real Time Windows Target (RTWT) by Mathworks is utilized to automatically generate C codes and executable files from this Simulink model. The generated executable file runs in real time on the personal computer with 1 ms of sampling time (1 kHz sampling frequency) and realizes a real-time control system. This allows easy design and rapid testing of the control algorithms with the actual hardware.

2.2.2 Overall Manipulator System

The layout of the overall three-axial pneumatic parallel manipulator system is shown in Fig. 2.5. The pneumatic servo system of the manipulator contains three proportional servo valves of the same type and three identical pneumatic rodless cylinders. The three rodless cylinders work together as the actuators on the three axes of the manipulator sharing the same pressure source and each axis has a linear encoder which measures the piston position of each cylinder, y_A , y_B and y_C . There are total six pressure sensors which are used to monitor the chamber pressures of the three cylinders, where P_1 and P_2 respectively represent the upper chamber pressure and the lower chamber pressure of each axis cylinder. u_A , u_B and u_C denote the control input signals for the proportional valves of axis A, B and C, respectively.

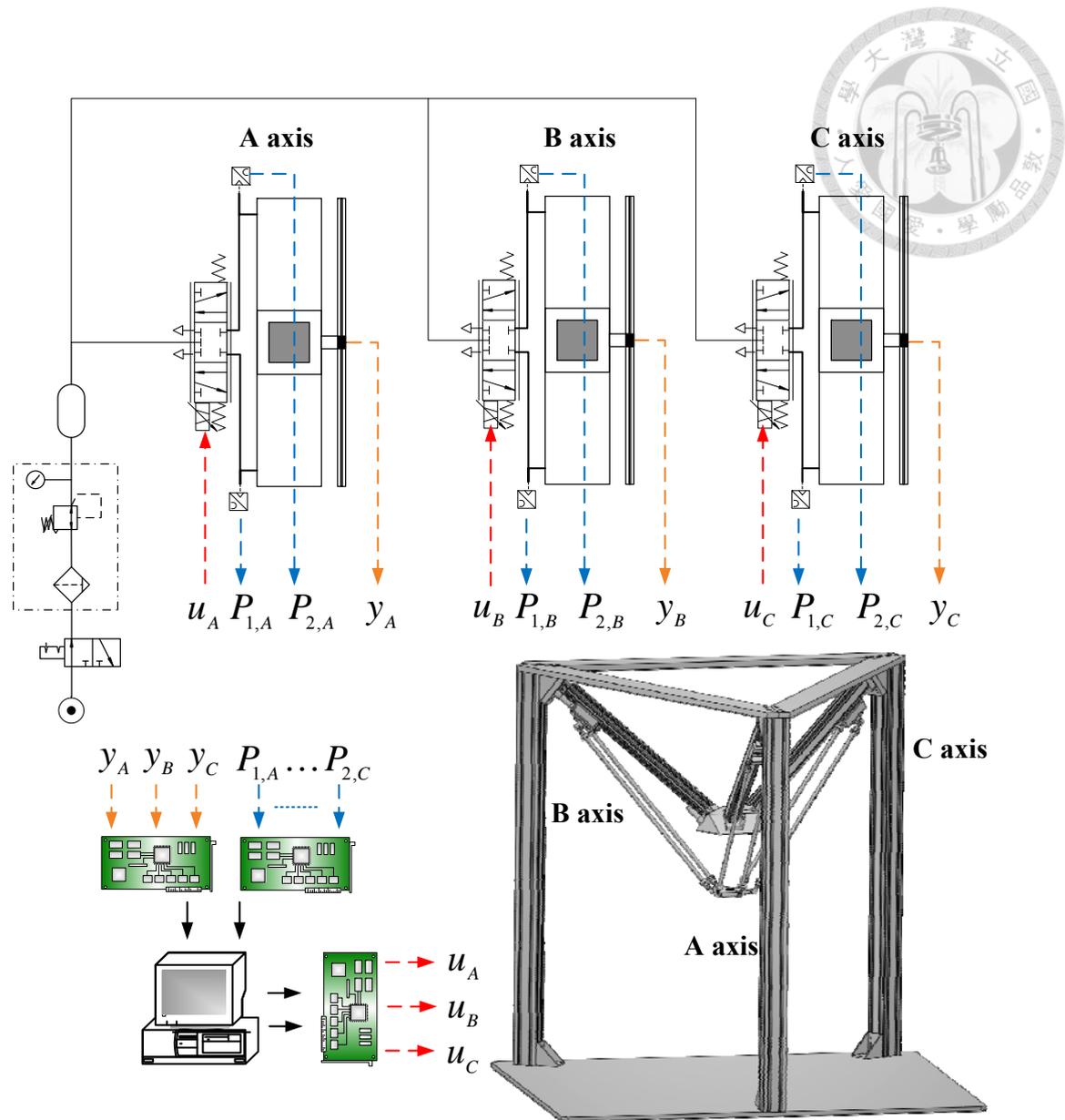


Fig. 2.5 Test rig layout of the overall manipulator system

In the PC-based control unit, three DAQ cards are installed and used to output the control signals and receive the input signal data from the different sensors. The control voltages of three proportional valves are calculated by the real-time control algorithm in the computer and sent to the control valves via the analogue output channels on PCI-1720U DAQ card manufactured by Advantech. The pressure data of cylinder chamber measured by the pressure sensors are recorded by Advantech PCI-1710UL multifunctional card via the analogue input channels. Finally the piston displacements

of cylinders measured by the linear encoders are counted and recorded by the counters on PCI-6601 DAQ card produced by National Instruments. Thus, the motion control of the manipulator end-effector can be achieved by simultaneously controlling the piston positions of the three cylinders with the individual pneumatic servo positioning system. Table 2.1 summarizes the components and the specifications of the system hardware used in the three-axial pneumatic parallel manipulator system.

Table 2.1 Specifications of system hardware

Components	Manufacturer	Type	Specifications
Air Compressor	Taiwan Co Sheng	AU-5	Flow rate: 500 l / min Output pressure: 6 bar
Pneumatic Rodless Cylinder	Festo	DGC-25-500-KF-YSR-A	Piston diameter: 25 mm Stroke: 500 mm
Pneumatic Proportional Directional Control Valve	Festo	MPYE-5-M5-010-B	Valve function: 5/3 way Input voltage: 0 - 10 V
Pressure Sensor	Festo	SDE1-D10	Pressure measuring range: 0 - 10 bar Output voltage: 0 -10 V
Optical Linear Encoder	Jena	LIA20-L301-WA	Resolution: 1 μ m
Data Acquisition Card	Advantech	PCI-1720U	4-ch analog output with 12-bit D/A converter
	Advantech	PCI-1710UL	16-ch analog input with 12-bit A/D converter 16-ch digital input/output
	National Instruments	PCI-6601	4-ch 32-bit counter with 20 MHz maximum source frequency 32-ch digital input/output

Chapter 3 Path Planning Algorithms for 3D Printing

3D-printing process requires the completion of four main tasks, such as Object Orientation, Support Generation, Slicing, and Path Planning. However, path planning has a remarkable impact on the overall manufacturing time and, therefore, the algorithms play a crucial role in solving this problem.

This chapter consists of three parts. The first part is to derive the path planning model. The Graph theory is introduced first, and according to the ways of linkage between each two points, some are built in double-directions directed graph and others are in single-direction directed graph. Both of them are in vector form. The second part is the path planning strategy for a layer. The modified Depth-First Search (DFS) and the modified Genetic Algorithm (GA) are proposed to generate the sub-paths and optimize the linkages of all sub-paths. The third part is the path planning strategy for layer-to-layer, based on the proposed algorithms for a layer to cascade all layers.

3.1 Path Planning Model

Before building the path planning model, the desired printing object has to be sliced equally first, and Fig. 3.1 shows an example. 3D printing is an additive manufacturing, which builds up layer by layer, and the easiest way to establish layers is to slice horizontally. The height of layers depends on the extruder of 3D printers. In order to simplify naming layers in this thesis, the bottom layer, called Layer 1, means the first layer to manufacture, and Layer 2 means the second layer to manufacture, etc. In Fig. 3.2, it is a three-layer object which means the bottom layer is Layer 1 and the top layer is Layer 3. Besides, the number of top layer implies that how many layers the desired printing object is composed of.

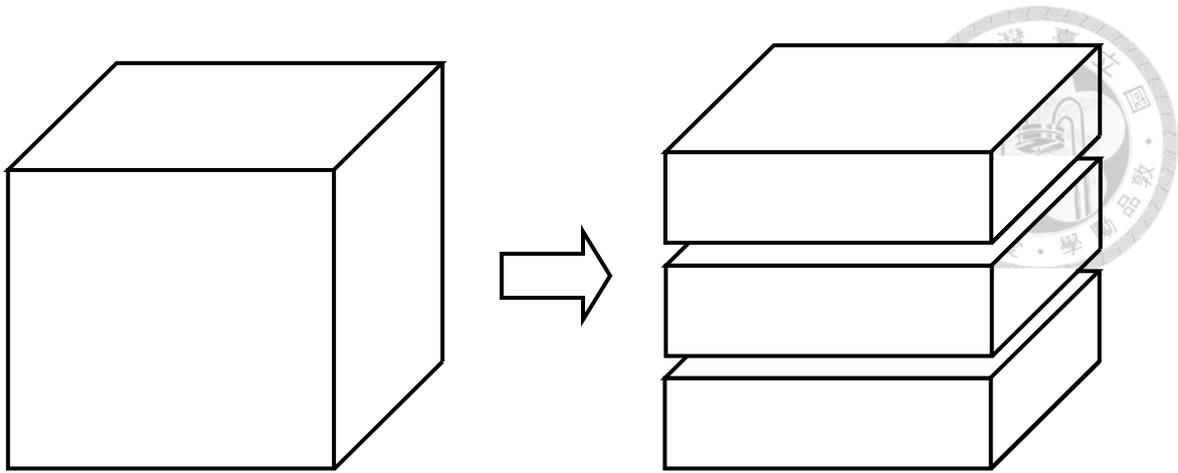


Fig. 3.1 Slicing a desired printing object equally

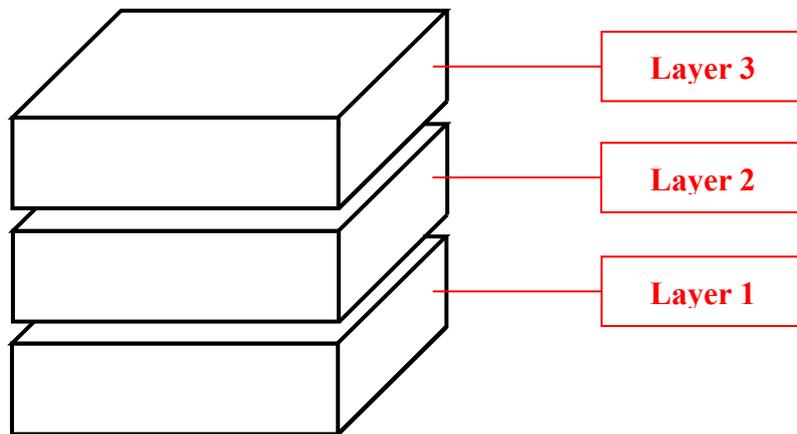


Fig. 3.2 The definition of layers

After slicing the desired printing object, the next step is to make each layer with equal-sized rectangular mesh, according to the resolution and extruder of 3D printers. In Fig. 3.3, a layer meshes into nine small cubes.

The last step is to choose an appropriate position as a point to represent component mesh. The best position of x-y plane, the horizontal direction of slicing, is to pick the center of shape of component mesh; the ideal position of z plane, the perpendicular direction of slicing, is at the bottom surface of component mesh, because of additive manufacturing. The result is depicted in Fig. 3.4. In the end, a 3D object is converted into 2D images, which are composed of simple points.

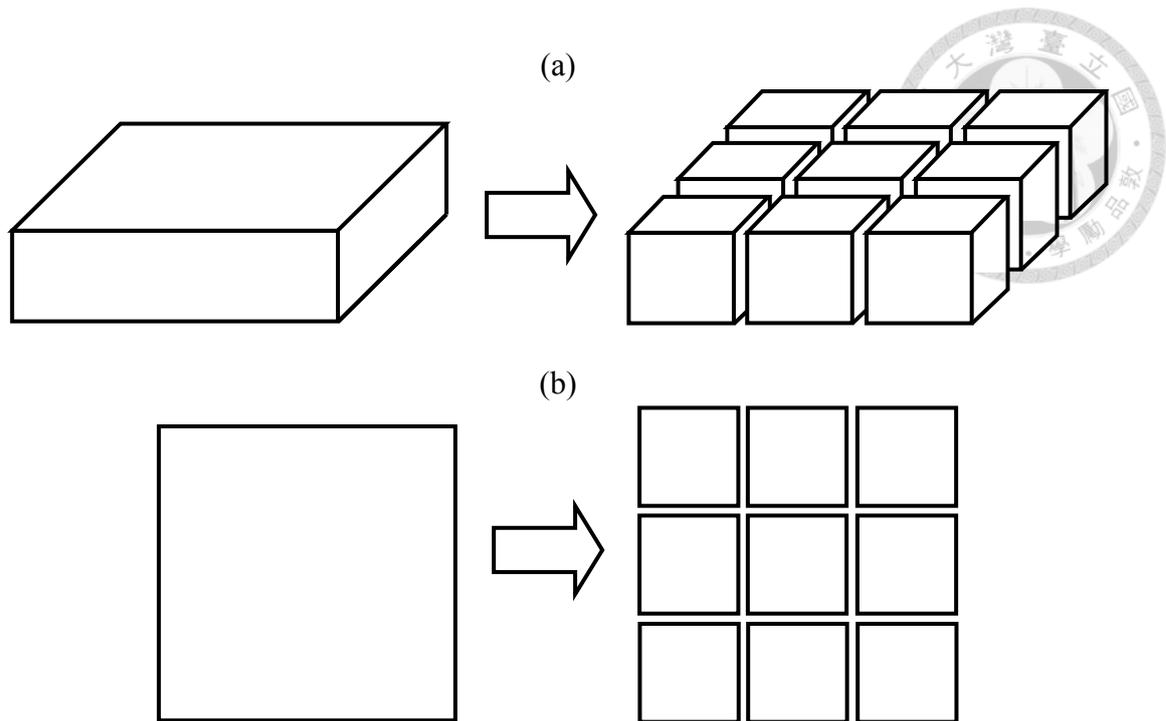


Fig. 3.3 Equal-sized rectangular mesh on a layer: (a) front view (b) top view

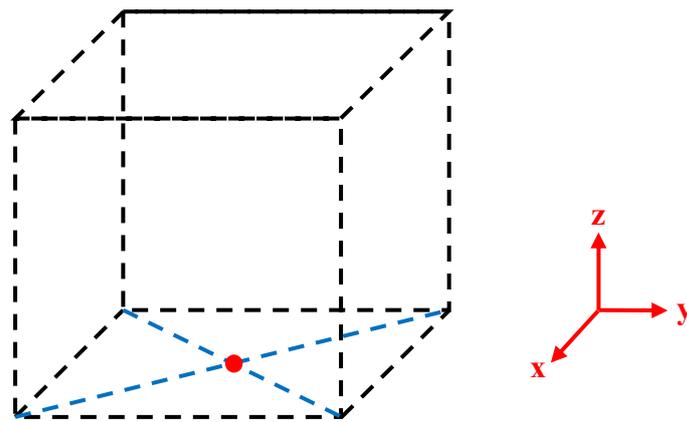
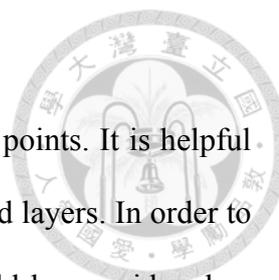


Fig. 3.4 The position of a point (red dot) substitutes component mesh

Graph theory [49] has a wide range of applications in engineering, in biological sciences, and in numerous other areas. A graph can be used to represent almost any physical situation involving discrete objects and a relationship among them. The Königsberg Bridge Problem is perhaps the best-known example in graph theory. It was a long-standing problem until solved by Leonhard Euler in 1736, by means of a graph [50]. Euler wrote the first paper ever in graph theory and thus become the originator of



the theory of graphs as well as of the rest of topology.

In path planning model, the component mesh is represented as points. It is helpful to use graph conveying the relationship between points on a layer and layers. In order to express the path directions of 3D printing, a directed graph G could be considered as going from vertex v_i to vertex v_j or from v_j to v_i , which presented as a start point and an end point in an overall path trajectory. For example, Fig. 3.5 shows a directed graph with five vertices (v_1, v_2, v_3, v_4, v_5) and five edges (e_1, e_2, e_3, e_4, e_5).

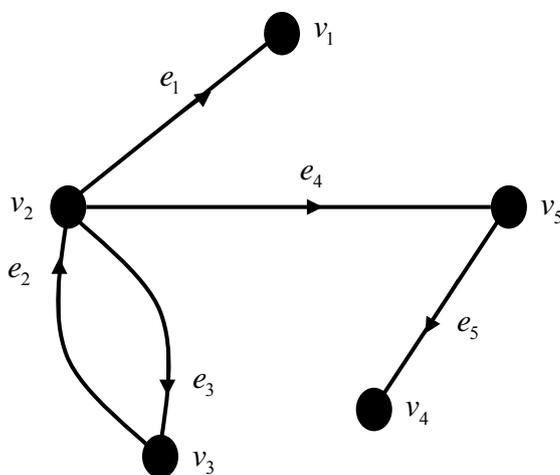


Fig. 3.5 A directed graph with 5 vertices and 5 edges

On a layer, each two of adjacent points should use double directions of directed graph representing two possible straight paths between vertex v_a and vertex v_b which are adjacent points and denoted $\overrightarrow{v_a v_b}$ and $\overrightarrow{v_b v_a}$ as edges, or paths. The reason is to show all possible path trajectories before path planning. In Fig. 3.6, five points $v_1, v_2, v_3, v_4,$ and v_5 on a same layer are used double-directions directed graph to represent the relationship of adjacent points.

Between layers, vertex v_c and vertex v_d , which are on different layers, should use single direction of directed graph representing only one straight path which denoted

$\overrightarrow{v_c v_d}$ as an edge, or a path. Owing to additive manufacturing, the overall path trajectory must start from bottom layer and end at top later; thus, using single direction of directed graph between layers shows the characteristic of 3D printing. In Fig. 3.7, there are three points v_1 , in Layer 1, v_2 , in Layer 2, and v_3 , in Layer 3, which used single-direction directed graph to link layers from bottom to top.

Both two directed graphs, the double-directions and the single-direction, are in vector form to express all possible orientations among points before path planning and also show only an overall expected path trajectory after path planning which are introduced in Section 3.2 and 3.3.

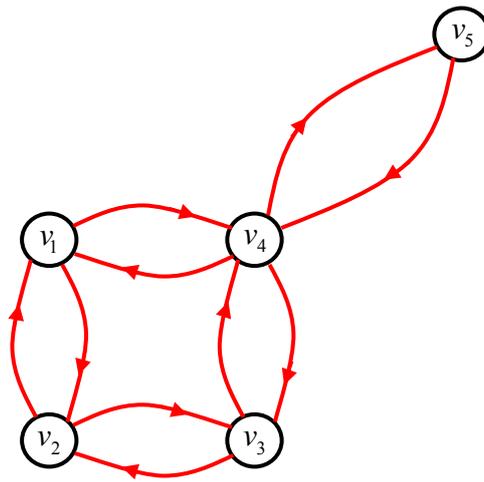


Fig. 3.6 Double-directions directed graph on a layer before path planning

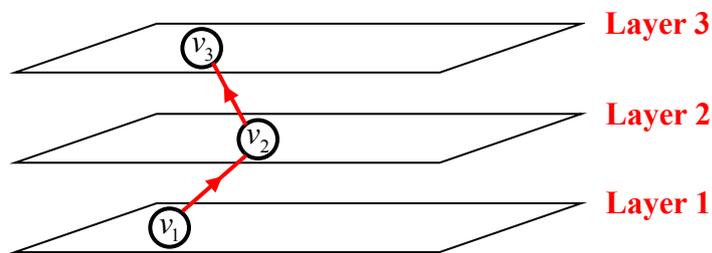


Fig. 3.7 Single-direction directed graph between layers before path planning

3.2 Single-Layer Path Planning Strategy



The algorithms of single-layer path planning are divided into two steps. The first step is to traverse all points on a layer and establish all sub-paths, which is introduced in Section 3.2.1 based on Depth-First Search (DFS), named modified-DFS. The second step is to find a minimum cost, means shortest lengths, of linking all sub-paths on a layer, which is introduced in Section 3.2.2 based on Genetic Algorithm (GA), named modified-GA. After that, a planned path trajectory on a layer is accomplished.

3.2.1 Path Generation

Depth-First Search (DFS) is a powerful technique of systematically traversing the edges of a given graph $G = (V, E)$, consists of a non-empty set of V of vertices and a set E of unordered pairs of vertices of V called edges, such that every edge is traversed exactly once, and each vertex is visited at least once. This technique, also called backtracking, was first formalized and used by Hopcroft and Tarjan in 1974 [51].

But in Section 3.1, each two points on a layer are connected in double-directions directed graph indicating all possible traversing ways; nevertheless, DFS is used on directed or undirected graphs of which edges are certainly exist. Therefore, DFS is not perfectly suitable for this case and needs modifications, named modified-DFS, which means to continue the concept of traversing all vertices on a layer.

As DFS, before using the modified-DFS, the priorities of traversing directions from a vertex to another have to be determined first and are only considered in four possible directions: Right, Left, Front, and Back. Fig. 3.8 shows the priorities of four traversing directions starting from vertex v . Right is the highest priority which means if a vertex is at adjacent right of v , it must traverse this direction first; Front is the lowest

priority which means this direction is the last consideration if other adjacent vertices have traversed before or no other adjacent vertices exist near to v .

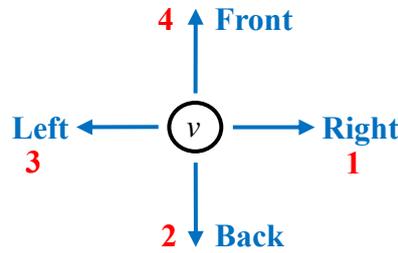


Fig. 3.8 The priorities of traversing directions (1 is the highest priority)

Owing to only four possible traversing directions, the double-directions linkages between each two points on a layer, mentioned in Section 3.1, are therefore also considered in four directions: Right, Left, Front, and Back. In other words, the diagonal double-directions linkages are impossible to happen. For instance, there are six points on a layer in Fig. 3.9(a). The linkages among these six points are only considered in four directions and shown in Fig. 3.9(b). Apparently, the linkage between $v_{2,2}$, denoted the position of (2, 2), and $v_{3,3}$, denoted the position of (3, 3), does not exist.

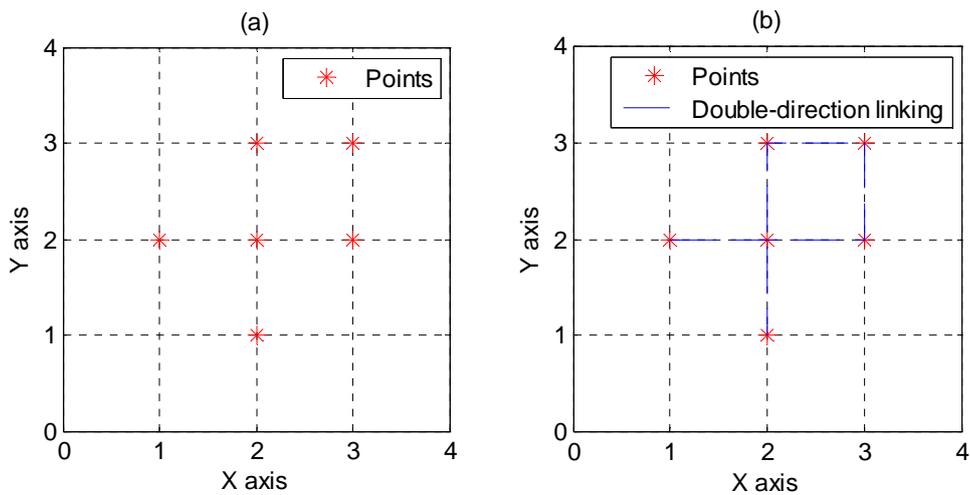
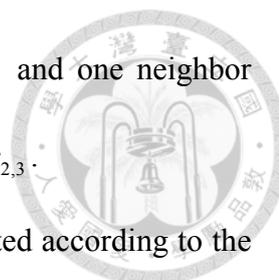


Fig. 3.9 The linkages between each two points based on traversing directions:

(a) 6 points (b) double-directions linking strategy

After the priorities of traversing directions are decided, the modified-DFS works in the following ways with an example shown in Fig. 3.10 for better understanding.

- A.** Select a start point as first sub-path named Sub-path 1. In Fig. 3.10(a), there are nine points on a layer and the start point of Sub-path 1 is $v_{1,4}$, denoted the position of (1, 4).
- B.** Check the neighborhood of start point to see if any of points are available. In Fig. 3.10(a), $v_{2,4}$, denoted the position of (2, 4), is at adjacent right of $v_{1,4}$ and the only neighbor.
- C.** Select a point in the neighborhood of start point except been visited according to the priorities of traversing directions, and mark it as part of Sub-path 1. In Fig. 3.10(b), select $v_{2,4}$, the only neighbor and the highest traversing priority, and link $v_{1,4}$ to $v_{2,4}$ as path trajectory of Sub-path 1.
- D.** Repeat the steps from step **B** to step **C**. If no points are available in the neighborhood of last selected point in Sub-path 1, the first sub-path is finished. In Fig. 3.10(c), the selected point are $v_{2,3}$, denoted the position of (2, 3), $v_{3,3}$, denoted the position of (3, 3), and $v_{4,3}$, denoted the position of (4, 3), in order. No points are available in the neighborhood of $v_{4,3}$ except been visited and, thus, Sub-path 1 is accomplished.
- E.** Go back to the former selected point and check if any of points are available in neighborhood except been visited. If not, repeat step **E** again. If the former selected point is the start point of Sub-path 1, jump to step **I**. In Fig. 3.10(d), the last selected point is $v_{4,3}$ and the former selected point is $v_{3,3}$; sadly, no points are available in the neighborhood of $v_{3,3}$ except been visited. After



repeating step **E** again, the former selected point is $v_{2,3}$ and one neighbor $v_{2,2}$, denoted the position of (2, 2), is at adjacent back of $v_{2,3}$.

- F.** Select an available point based on step **E** except been visited according to the priorities of traversing directions and set it as start point of new sub-path named Sub-path 2. In Fig. 3.10(d), the start point of Sub-path 2 is $v_{2,2}$ which is in neighborhood of $v_{2,3}$, one of selected points in Sub-path 1.
- G.** Check the neighborhood of start point, select a point except been visited according to the priorities of traversing directions and mark it as part of Sub-path 2. Repeat on following selected points till no points are available in neighborhood and, finally, Sub-path 2 is finished. In Fig. 3.10(e), the selected point except start point are $v_{2,1}$, denoted the position of (2, 1), $v_{1,1}$, denoted the position of (1, 1), and $v_{1,2}$, denoted the position of (1, 2), in order. No points are available in the neighborhood of $v_{1,2}$ except been visited and, thus, Sub-path 2 is accomplished.
- H.** Repeat the steps from step **E** to step **G** and name new sub-path as Sub-path 3 if any of points are still available, and so on till the former selected point in step **E** is the start point of Sub-path 1. In Fig. 3.10(f), the last selected point is $v_{1,2}$; after repeating step **E** again and again because of no available points, the former selected point becomes $v_{1,4}$ which is the start point of Sub-path 1.
- I.** If no points are available in the neighborhood of the start point of Sub-path 1 except been visited, modified-DFS is completed. If not, repeat the steps from step **F** to step **H**. In Fig. 3.10(f), no points are available in the neighborhood of $v_{1,4}$ except been visited; therefore, modified-DFS is completed.

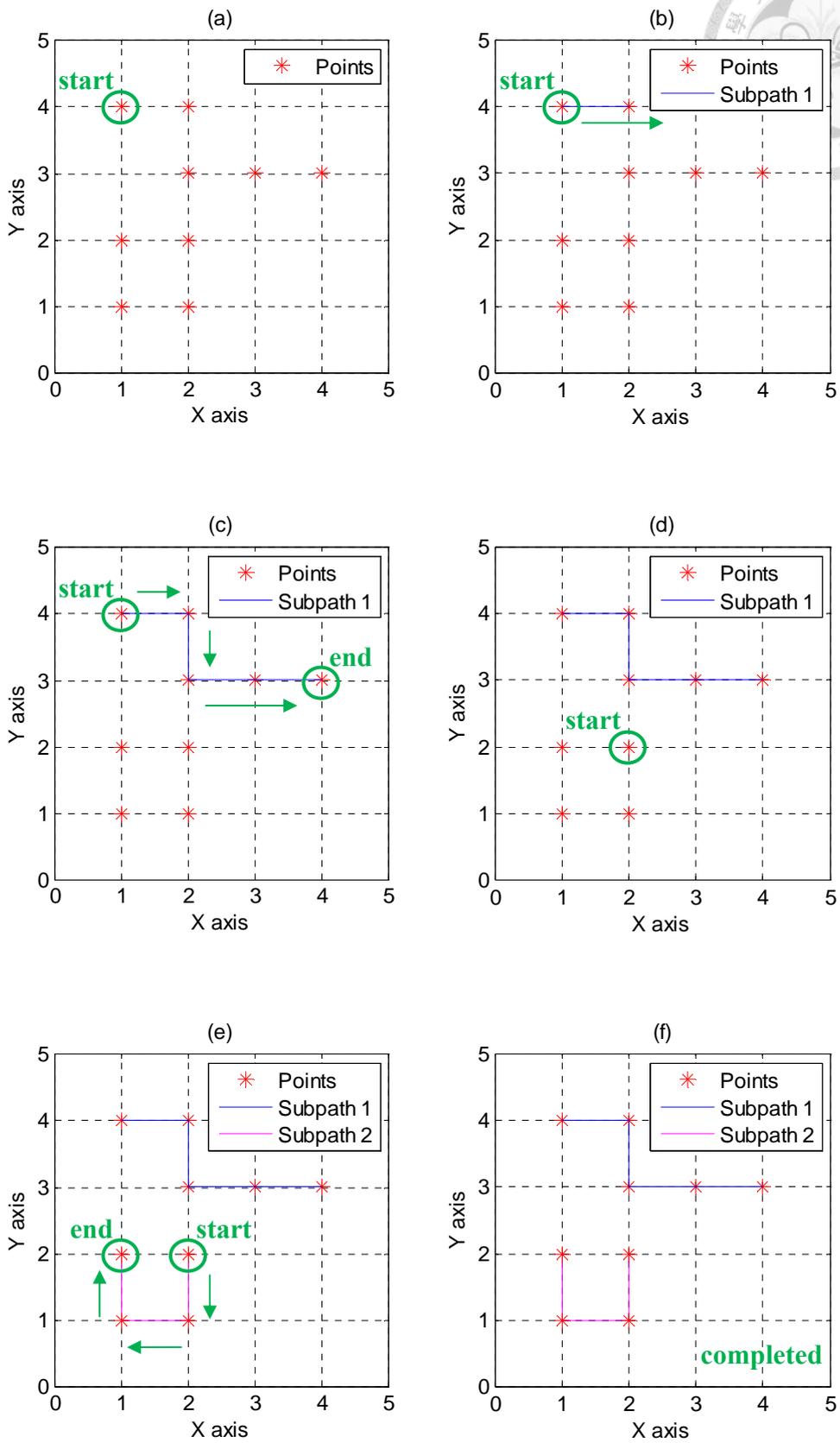
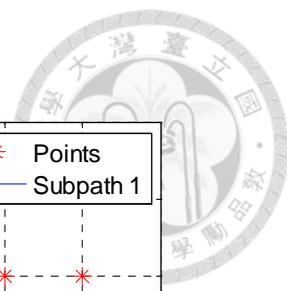


Fig. 3.10 A general example of modified-DFS

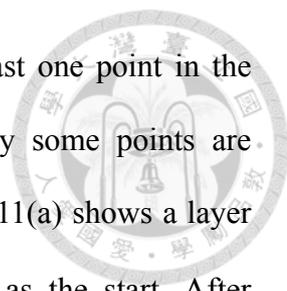


Fig. 3.10 is a general example in which each point has at least one point in the neighborhood of four traversing directions; however, occasionally some points are separated into un-connected groups on a layer. For instance, Fig. 3.11(a) shows a layer with eleven points selected $v_{1,4}$, denoted the position of (1, 4), as the start. After applying modified-DFS, only one sub-path, Sub-path 1, is generated which starts from $v_{1,4}$ and ends at $v_{1,3}$, denoted the position of (1, 3), shown in Fig. 3.11(b). Apparently, Sub-path 1 merely includes eight points and three points are left to wait for traversing.

To solve this situation, the method is adding a counter to calculate how many points have been marked in step **C** and step **G**. Because the total amount of points on a layer is given, the advantage is to make sure each point belongs in one of sub-paths. If the counter does not match up after first modified-DFS, finding a nearest point to the first start point, because modified-DFS ends when backing to the start in step **I**, to turn into a new start point as second modified-DFS, and so on till the counter matches the total amount of points on a layer. By continuing the case in Fig. 3.11, the nearest point to $v_{1,4}$, the start of first modified-DFS, is $v_{4,3}$, denoted the position of (4, 3), which becomes the new start point of second modified-DFS and shown in Fig. 3.12(a). After applying second modified-DFS in Fig. 3.12(b), Sub-path 2 is generated and starts from $v_{4,3}$ and ends at $v_{4,1}$, denoted the position of (4, 1). The counter is eleven which means each point has been traversed and no needs to try third modified-DFS.

Fig. 3.13 is a complete example including the exception. Fig. 3.13(a) shows a layer with forty-five point selected $v_{1,8}$ as the start. In Fig. 3.13(b), Sub-path 1, Sub-path 2 and Sub-path 3 are generated during first modified-DFS; Sub-path 4 is established during second modified-DFS. After these, the counter is forty-five and each point has been traversed, which means the first step of single-layer path planning is finished.

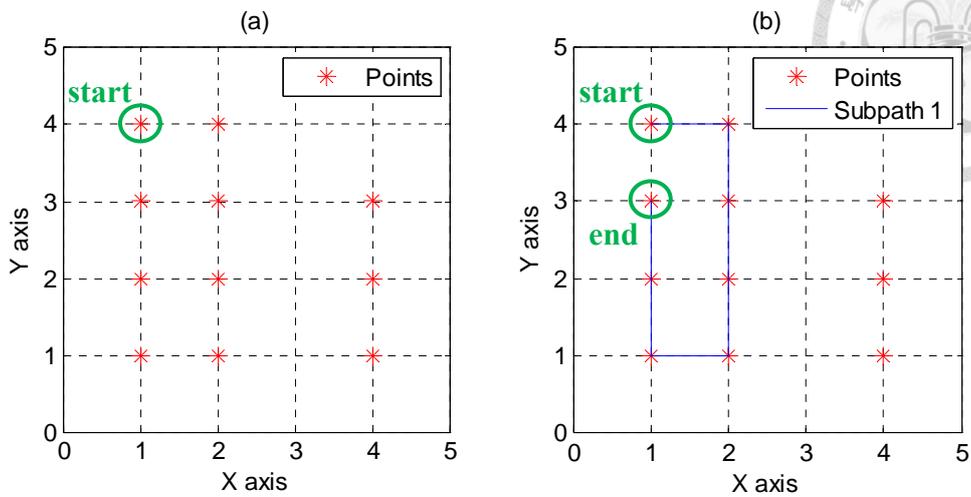


Fig. 3.11 The exception in modified-DFS:

(a) 11 points (b) some un-traversed points after modified-DFS

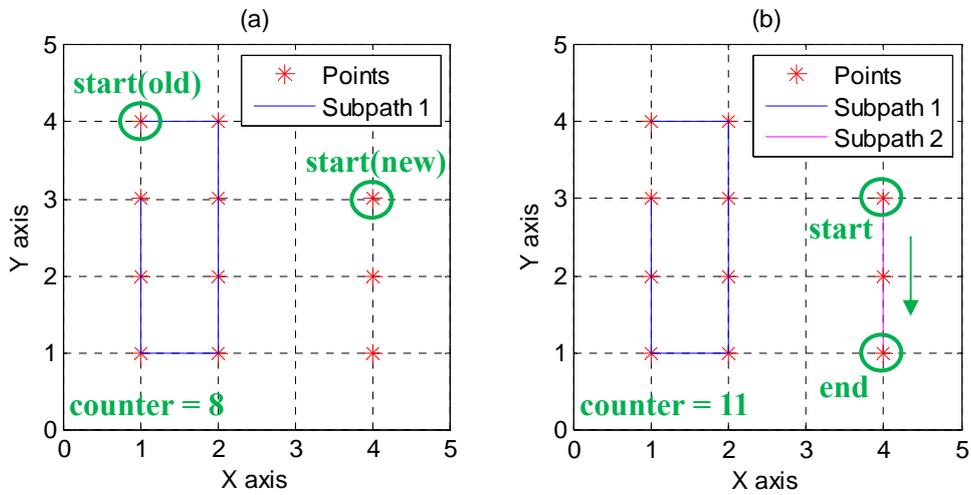


Fig. 3.12 Solution of the exception in modified-DFS:

(a) selecting a new start point nearest to the start point of Sub-path 1

(b) using modified-DFS again and checking the counter when completed

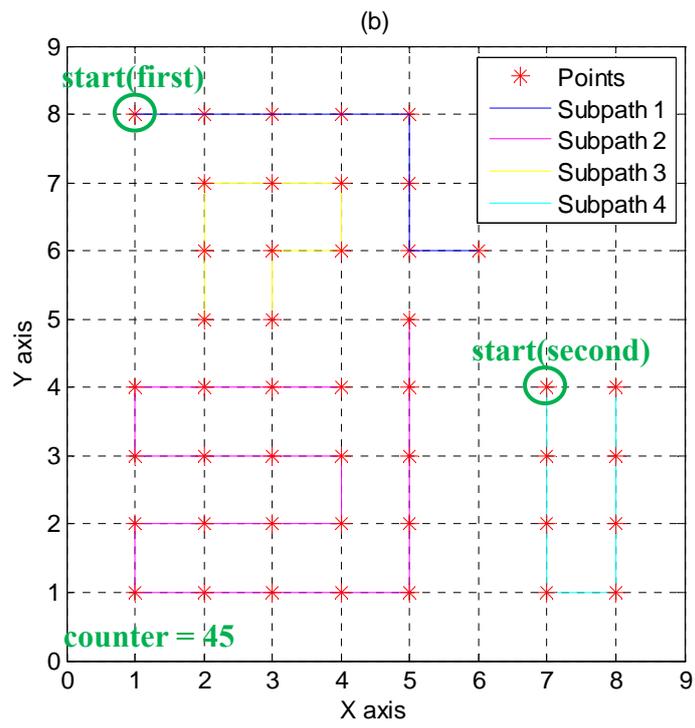
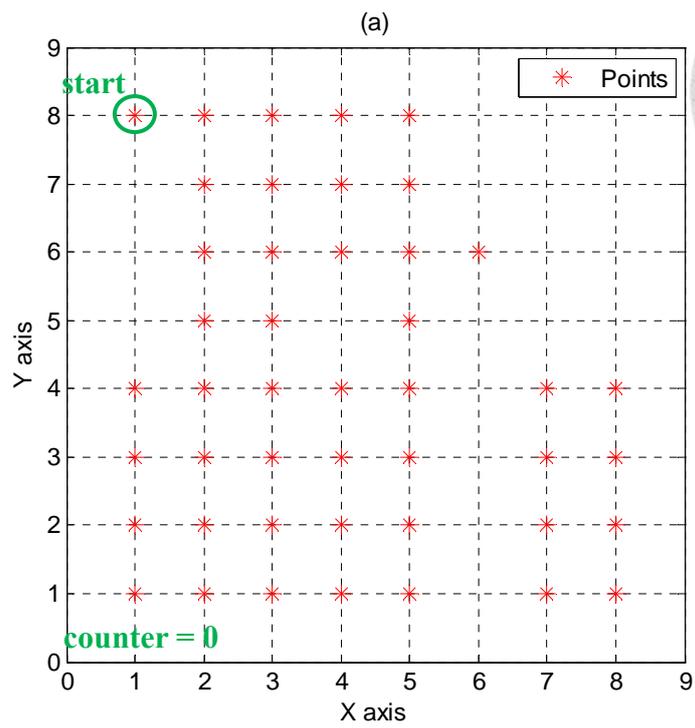
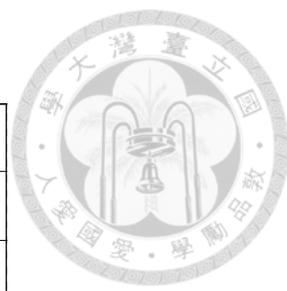
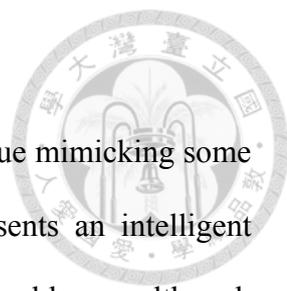


Fig. 3.13 A complete example of modified-DFS including the exception:

(a) 45 points (b) 4 sub-paths after modified-DFS



3.2.2 Path Optimization

Genetic Algorithm (GA) is an adaptive heuristic search technique mimicking some of the processes observed in natural evolution. As such it represents an intelligent exploitation of a random search used to solve optimization problems; although randomized, GA is random by no means, instead exploiting historical information to direct the search into the region of better performance within the search space. The father of the original GA was John Holland who invented it in early 1970's[52].

Before introducing Genetic Algorithm, each sub-path is replaced by a rectangular notation with a numeral, means the number of sub-path, for better understanding the process of GA. By continuing the case in Fig. 3.13, Fig. 3.14 shows four notations to represent each sub-path and all notations are marked by corresponding numbers.

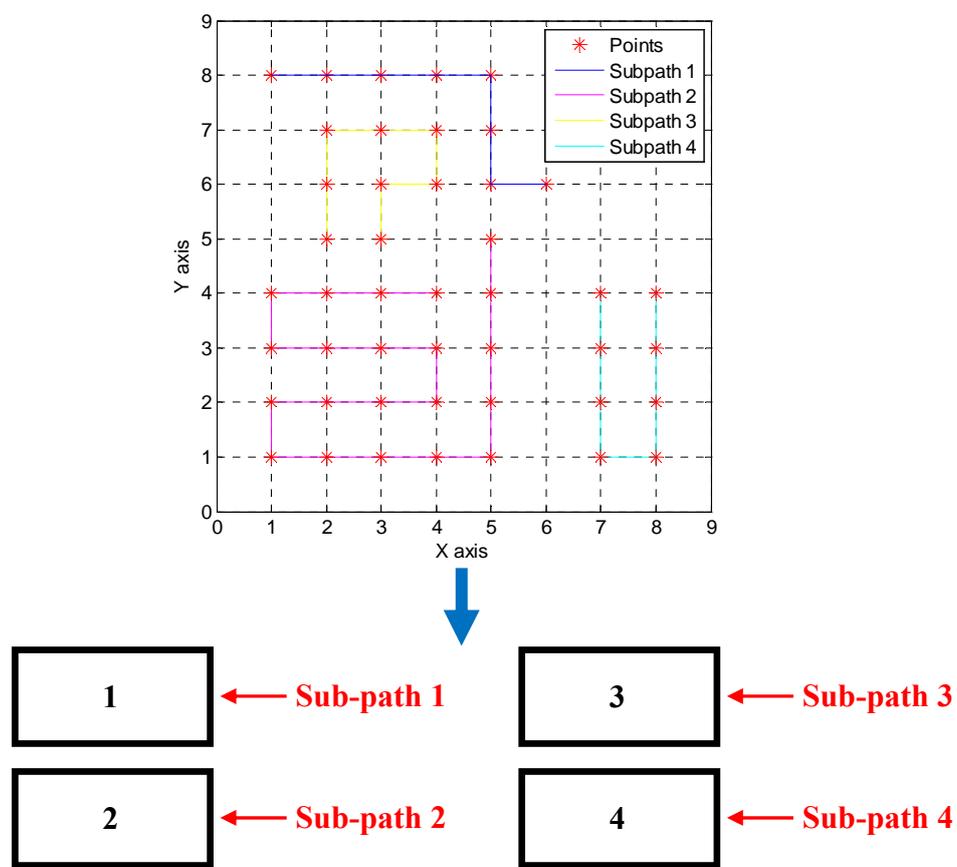


Fig. 3.14 Notations of sub-paths for Genetic Algorithm

The general process of Genetic Algorithm works in the following ways and demonstrates at some steps with illustrations.

- A. Initialization: Create two overall paths of a layer as initial populations. The first comes from Section 3.2.1 and is composed of all sub-paths linking from Sub-path 1 to last sub-path. The second, basically, is same as the first but swapping the second and last linking order. The reason will be explained later.

Fig. 3.15 shows an example of two initial populations established from four sub-paths.

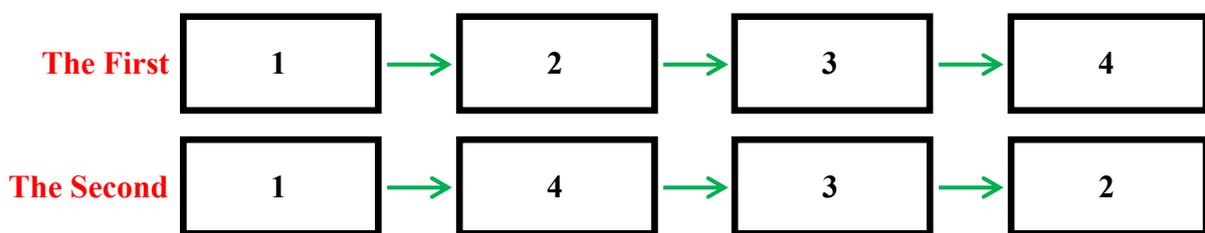


Fig. 3.15 An example of two initial populations

- B. Evaluation: Each individual, an overall path of a layer, of the population is then evaluated for fitness. The fitness value comes from comparing the total lengths of linking all sub-paths of a layer and the ideal shortest total lengths which are that each two of linking-adjacent sub-paths is at length of a side of component mesh. Fig. 3.16 shows an example of ideal shortest total lengths established from four sub-paths.

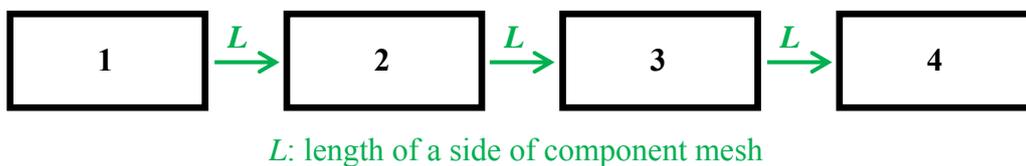


Fig. 3.16 An example of ideal shortest total lengths

- C. Selection: To constantly improve the fitness values, discard the bad designs and keep the best in the populations; thus, the populations always only have

two individuals, the two shortest total lengths of designs so far, as step A.

- D. Crossover:** Create a new individual as next generation, also called offspring, by combining aspects of selected individuals. The goal is that by combining certain traits from two or more individuals, a fitter offspring will inherit the best traits from each of its parents. The method is picking one or more sub-paths of the first as fixed linking-order position of offspring, and then according to linking-order positions of the sub-paths of the second except the sub-paths picked in the first, adding each sub-path to the empty linking-order position of offspring in order. Fig. 3.17 shows an example of crossover established from six sub-paths.

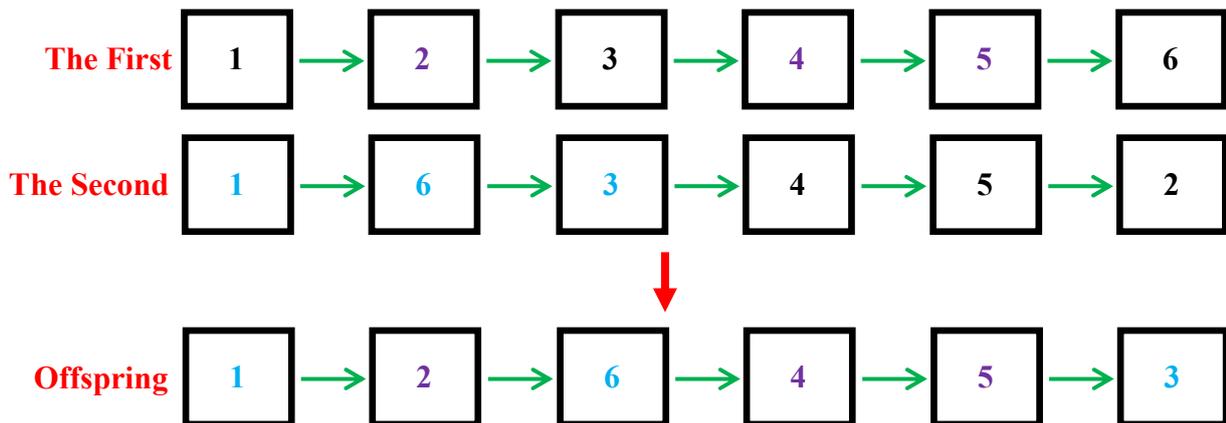


Fig. 3.17 An example of crossover

- E. Mutation:** To add a little bit randomness into the populations' genetics, work by making small changes at random to an individual genome. Each sub-path of an overall path has a low probability to mutate and flips the oriented direction when it happens. Fig. 3.18 shows an example, established from four sub-paths, of mutation occurring at Sub-path 2.

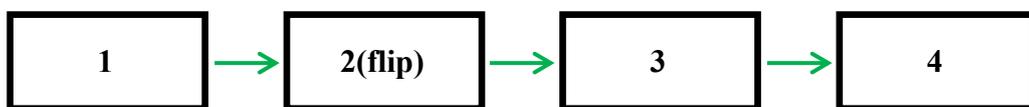
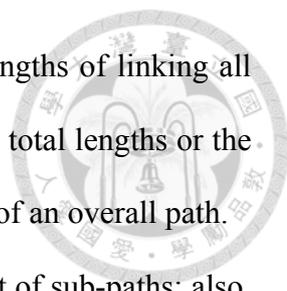


Fig. 3.18 An example of mutation at Sub-path 2



F. Repeat: Keep doing from step **B** to step **E** till the total lengths of linking all sub-paths of a layer is under 1.2 times of the ideal shortest total lengths or the repeated times reaches 2 times of the amount of sub-paths of an overall path.

However, some steps are not suitable for all cases of the amount of sub-paths; also, some linking-order positions of sub-paths bring some problems at special situations. Therefore, the general process of GA needs some modifications and limitations to deal with the exceptions, and modified-GA is established.

To reduce the complexities of layer-to-layer path planning introduced in Section 3.3, the starter of Sub-path 1 should also be the starter of an overall path, which means that Sub-path 1 should be always picked in step **D** to keep the linking order the same. Luckily, Sub-path 1 is always at the first linking-order position and, thus, it does not have to be picked. On the other hand, if the amount of sub-paths is one, modified-GA is useless because there is always one overall path of a layer; if the amount of sub-paths is two, step **A** should be skipped because the two initial populations are the same.

In step **D**, if the picks include the second and last linking-order positions of the first at first round of modified-GA, the offspring will be the same as the first in step **A** and shown in Fig. 3.19; also, if the picks non-include the second and last linking-order positions of the first at first round of modified-GA, the offspring will be the same as the second in step **A** and shown in Fig. 3.20. To solve this situation, at the first round of GA, the picks must only include one sub-path between the second and last linking-order positions of the first.

In step **E**, if Sub-path 1 is flipped, the starter becomes the last point of Sub-path 1. To avoid that, Sub-path 1 has to be split into the starter and the remaining Sub-path 1, named new-Sub-path 1, and then new-Sub-path 1 is flipped. Next, the starter links to the last point of the overall path and the linking orientation from second linking-order

position to last linking-order position has to flip again. Fig. 3.21 shows a solution example established from four sub-paths.

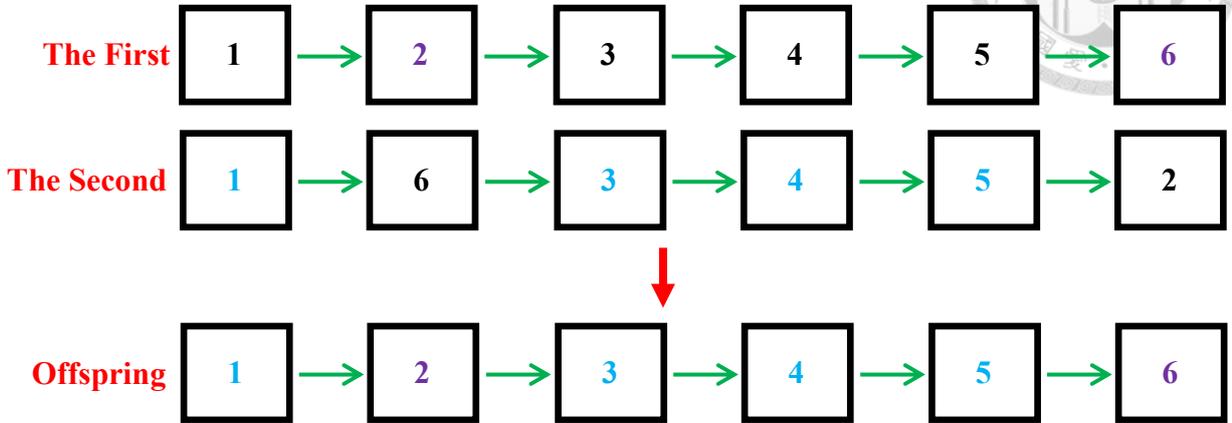


Fig. 3.19 An example of the exception in step D (i)

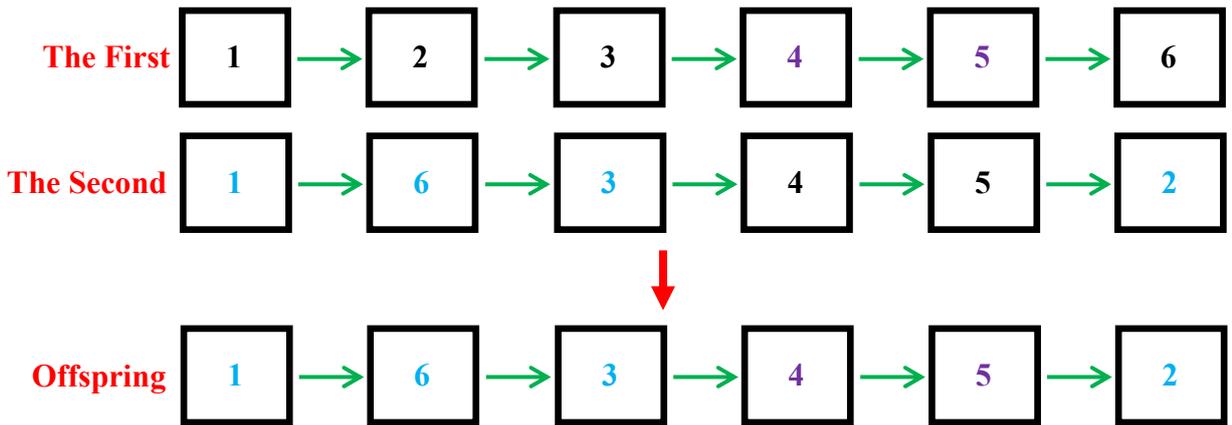


Fig. 3.20 An example of the exception in step D (ii)

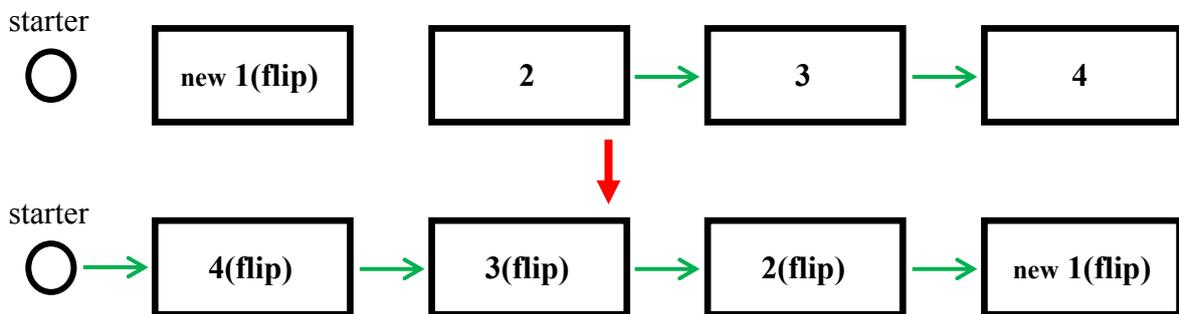
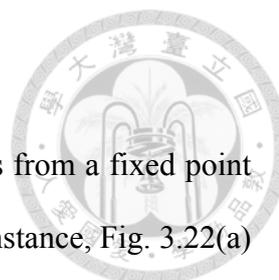


Fig. 3.21 Solution of the exception at Sub-path 1 in step E



3.3 Layer-to-Layer Path Planning Strategy

In Section 3.2, the algorithms of path planning on a layer starts from a fixed point of which the position non-changes when the algorithms finish. For instance, Fig. 3.22(a) shows a layer with nine points which starts at $v_{1,3}$, denoted the position of (1, 3). In Fig. 3.22(b), the path trajectory is accomplished by the modified-DFS and the modified-GA. The start point of path trajectory stays the same at $v_{1,3}$ and the end point is at $v_{2,2}$, denoted the position of (2, 2).

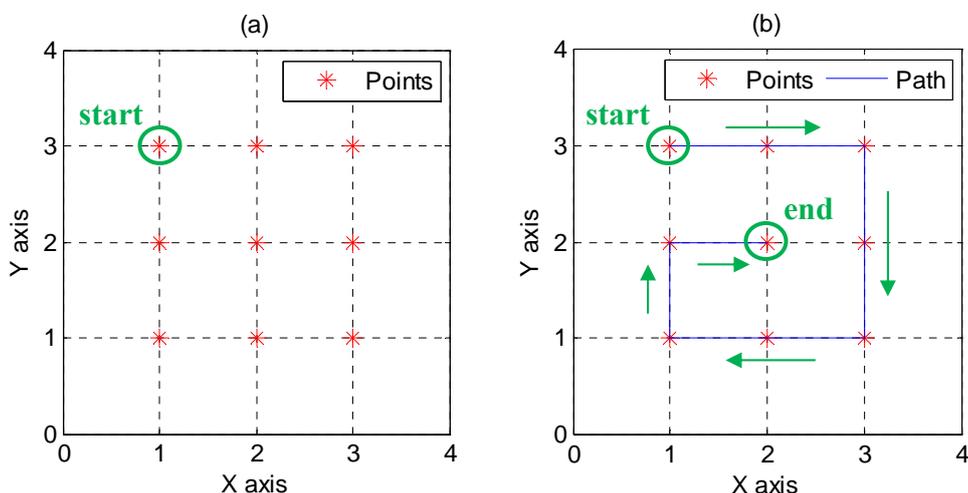


Fig. 3.22 Path planning on a layer: (a) 9 points (b) planned path trajectory

In order to reduce unnecessary movements among points, the strategy of linking two layers is to pick the end point of path trajectory on a layer as a start point, which keeps the same values of x-y plane and changes the value of z plane, to next layer. If the end point is $v_{a,b}$, denoted the position of (a, b) on a layer, the start point on next layer denotes as $v'_{a,b}$, an apostrophe on $v_{a,b}$, which express the same position of (a, b) but in different z-axis. Besides, in algorithm of modified-DFS, each point can be visited only once; thus, it is impossible to appear a notation $v''_{a,b}$, a bi-apostrophe on $v_{a,b}$, as a start



point right after a layer starting at $v'_{a,b}$. In Fig. 3.23, there are two linkages among three layers after layer-to-layer path planning, which are different from linking two arbitrary points between two adjacent layers in Fig. 3.7.

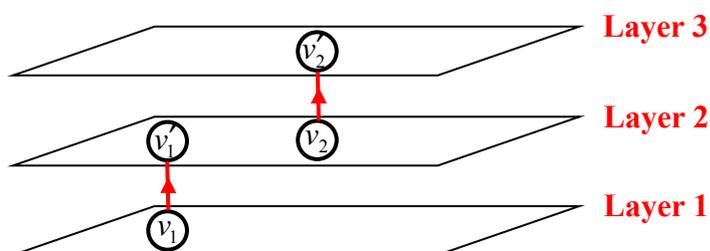


Fig. 3.23 Layer-to-layer path planning on a 3-layers object

However, the start point $v'_{a,b}$ sometimes does not exist on a layer which is the next adjacent layer with an end point $v_{a,b}$ in single-layer path trajectory. For example, Fig. 3.24(a) shows a single-layer planned path trajectory of which the end point is $v_{1,3}$ on Layer i . In Fig. 3.24(b), according to the strategy of layer-to-layer path planning, the start point should be $v'_{1,3}$ but unfortunately non-exists on Layer $i+1$.

To solve this situation, the method is to make the non-existent start point $v'_{a,b}$ become one of points on that layer. The advantage is that the linkage between layers remains the same in former proposed layer-to-layer path planning strategy. Avoiding destroying the original pattern, represented by points, on that layer, finding a nearest point to the start point $v'_{a,b}$ to turn into the new start point $v'_{c,d}$, denoted the position of (c, d); then, linking these two points, $v'_{a,b}$ and $v'_{c,d}$, to become a path, called pre-path, which means it comes before path planning. Continuing the case in Fig. 3.24, the non-existent start point $v'_{1,3}$ becomes one of points on Layer $i+1$ in Fig. 3.25(a). The nearest point to $v'_{1,3}$ is $v'_{3,2}$ which becomes the new start point, shown in Fig. 3.25(b).

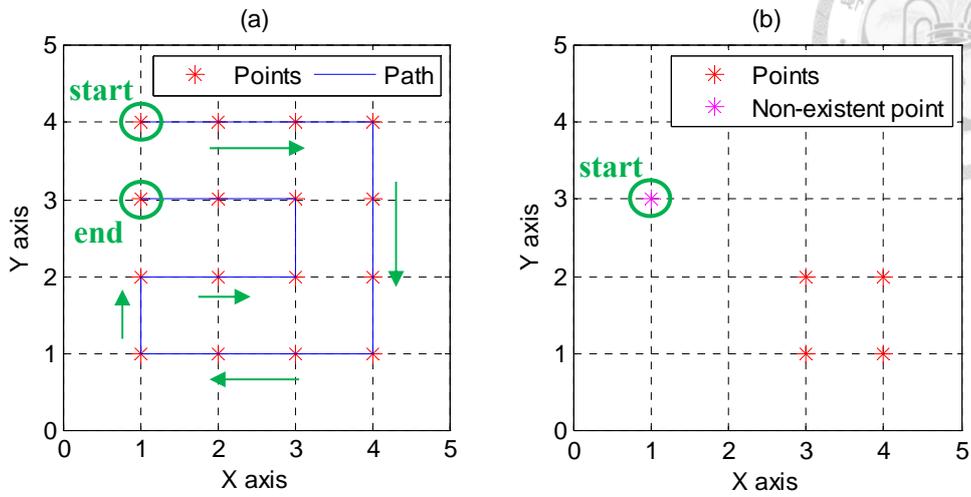


Fig. 3.24 The exception in layer-to-layer path planning:

(a) after single-layer path planning on Layer i

(b) a non-existent start point (purple dot) on Layer $i + 1$

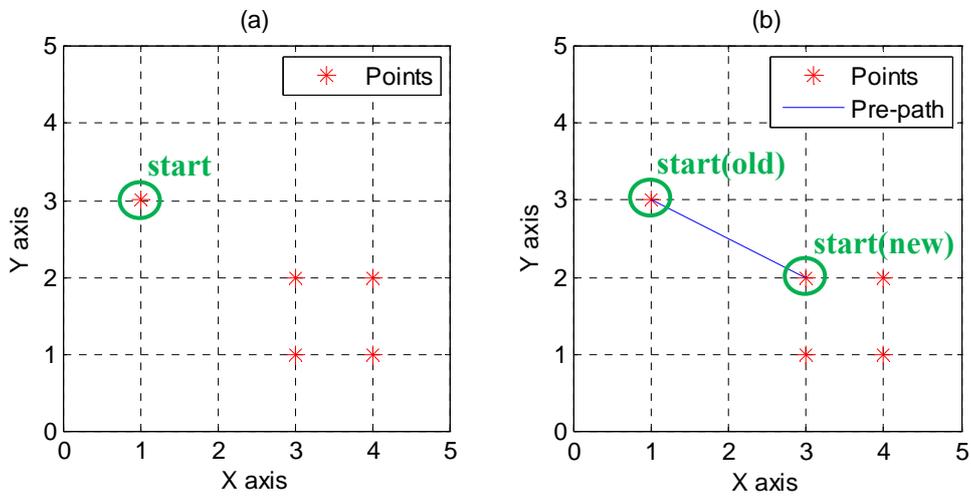


Fig. 3.25 Solution of the exception in layer-to-layer path planning:

(a) creating the non-existent start point on Layer $i + 1$

(b) linking to nearest point and changing the start point on Layer $i + 1$

Chapter 4 Analysis of Kinematics



The kinematic analysis copes with the study of the motion of a manipulator as constrained by the geometry configuration of the links and the joints. In order to plan and control the movement of a manipulator, the relationship of the position, velocity and acceleration between the joints and the links have to be known in advance. That makes kinematic analysis a crucial point in the development of a robotic manipulator system.

Roughly, the manipulator kinematics is divided into two parts, inverse kinematics and forward (or direct) kinematics. The inverse kinematics problem involves finding a set of actuated joint variables that will achieve a known position and orientation of the end-effector of the manipulator. On the contrary, the forward kinematics covers the problem of determining the position and orientation of the end-effector from the given actuated joint coordinates of the manipulator.

In this chapter, the geometry of the manipulator is described first, and then the inverse and forward kinematics will be derived from the geometric method with the vector-loop closure equations [53].

4.1 Geometry of the Manipulator

The schematic diagram of the proposed three-axial pyramidal parallel manipulator is depicted in Fig. 4.1. In order to simplify the analysis, the parallel chain of each limb are assumed to be equivalent to a single link with a pair of spherical joints at its two ends as shown in Fig. 4.1. The geometry and the definition of the joint angles and link lengths for one typical limb are shown in Fig. 4.2.

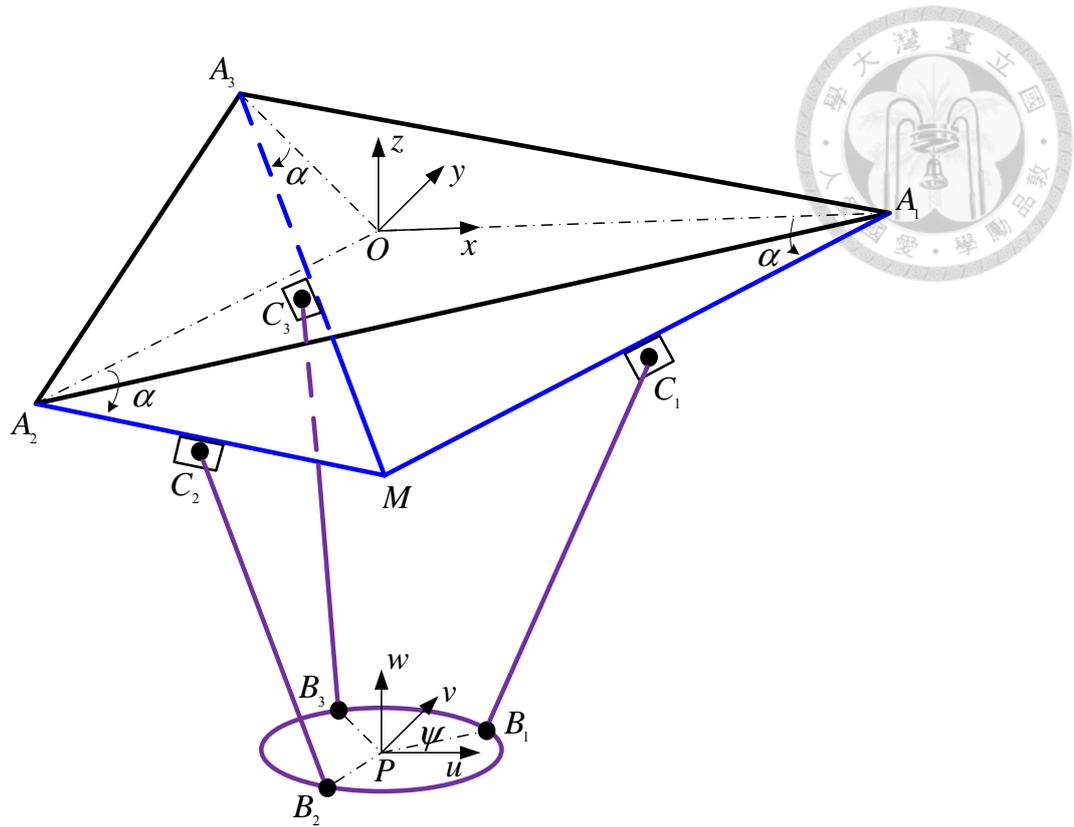


Fig. 4.1 Schematic diagram of the three-axial pyramidal parallel manipulator

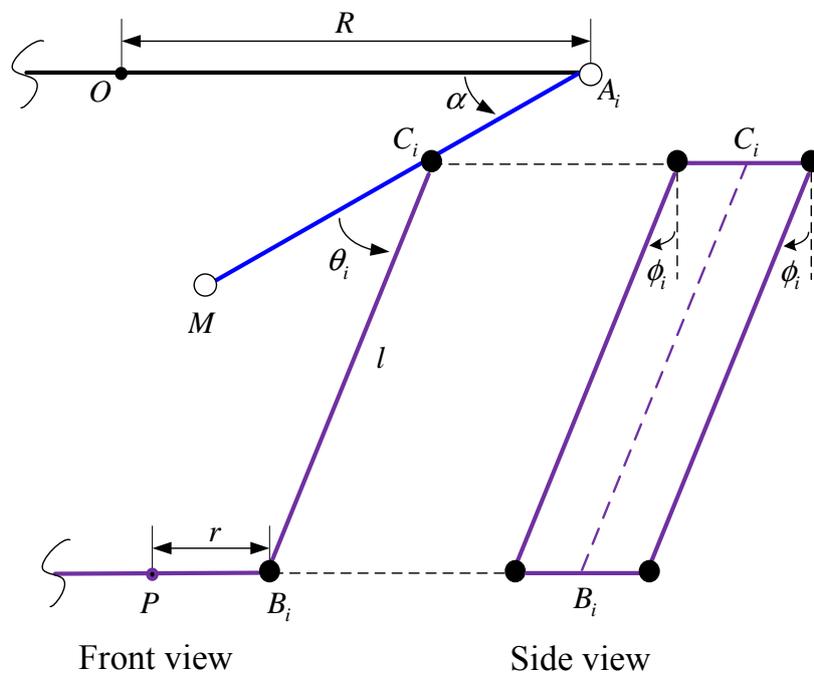


Fig. 4.2 Geometry of one typical limb

As shown in Fig. 4.1, a fixed Cartesian coordinate frame (x - y - z) is assigned at the center point O of the fixed base platform defined by the triangle $\Delta A_1A_2A_3$, and a moving Cartesian coordinate system (u - v - w) is placed at the center point P of the triangle $\Delta B_1B_2B_3$. In Fig. 4.2, the length between O and A_i is denoted by R which is defined as the fixed base radius, r represents the length between P and B_i and which is defined as the moving platform radius, and l is the length of each kinematic chain. The angle α is measured from the fixed base to the axis of actuator (line A_iM) and is defined as the actuator layout angle which is also the orientation angle of the cylinder. θ_i and ϕ_i are the angles which define the orientation of the kinematic chain of each limb, and they change with the position of the moving platform. The vector representation for one typical limb is shown in Fig. 4.3.

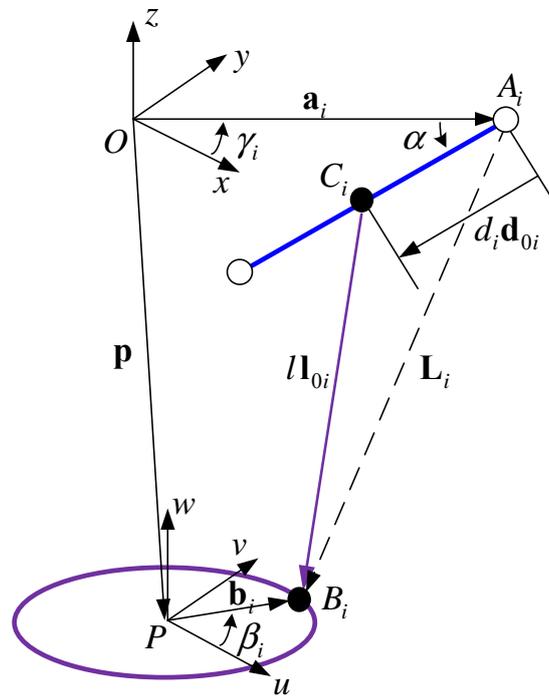


Fig. 4.3 Vector representation for one typical limb

For simplification, let x -axis and u -axis be parallel to each other, and the x -axis directs along the vector $\overline{OA_i}$. Vector $\overline{OA_i}$ is not necessarily parallel to vector $\overline{PB_i}$, and the angle between them is defined as the offset angle ψ . γ_i is the angle measured from the x -axis to $\overline{OA_i}$ in the fixed coordinate frame, and angle β_i is from the u -axis to $\overline{PB_i}$ in the moving frame.

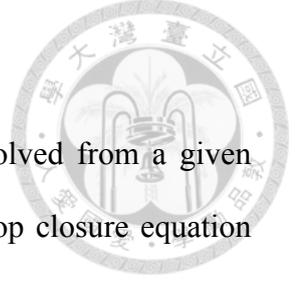
Generally, the position and the orientation of the moving platform with respect to the fixed frame can be described by a position vector

$$\mathbf{p} = \overline{OP} = [p_x \quad p_y \quad p_z]^T. \quad (4.1)$$

The geometry parameters of the manipulator and their design values are listed in the Table 4.1.

Table 4.1 Geometry parameters of the manipulator

Parameter	Description	Value
R	Length between O and A_i ; the fixed base radius	734 mm
r	Length between P and B_i ; the moving platform radius	61 mm
l	Length of the kinematic chain	746 mm
α	Actuator layout angle	38 deg
ψ	Offset angle of moving platform	15.68 deg



4.2 Analysis of Inverse Kinematics

For the inverse kinematics, the actuated joint variables are solved from a given position of the moving platform. Referring to Fig. 4.3, a vector-loop closure equation can be written for each limb:

$$l_i \mathbf{l}_{0i} = \mathbf{L}_i - d_i \mathbf{d}_{0i}, \quad (4.2)$$

with

$$\mathbf{L}_i = \mathbf{p} + \mathbf{b}_i - \mathbf{a}_i, \quad (4.3)$$

where \mathbf{l}_{0i} is the unit vector along $C_i B_i$, d_i represents the linear displacement of i th actuator that is also the cylinder piston position of i th limb, and \mathbf{d}_{0i} is the corresponding unit vector directing along $A_i C_i$.

Squaring the both sides of Eq. (4.2) and rearranging the terms yields

$$d_i^2 - 2d_i \mathbf{d}_{0i}^T \mathbf{L}_i + \mathbf{L}_i^T \mathbf{L}_i - l^2 = 0. \quad (4.4)$$

Solving Eq. (4.4) generates the inverse kinematic solutions:

$$d_i = \mathbf{d}_{0i}^T \mathbf{L}_i \pm \sqrt{(\mathbf{d}_{0i}^T \mathbf{L}_i)^2 - \mathbf{L}_i^T \mathbf{L}_i + l^2}. \quad (4.5)$$

In Eq. (4.5), there exist two solutions for each actuator. However, only the negative square root solution satisfies the current assembly of the mechanism where the three actuators are inclined inward from top to bottom. Thus, the inverse kinematic equation for i th limb is

$$d_i = \mathbf{d}_{0i}^T \mathbf{L}_i - \sqrt{(\mathbf{d}_{0i}^T \mathbf{L}_i)^2 - \mathbf{L}_i^T \mathbf{L}_i + l^2}. \quad (4.6)$$



4.3 Analysis of Forward Kinematics

Given a set of the input actuated joint variables, d_1 , d_2 and d_3 , the position of the moving platform, \mathbf{p} , can be solved by the forward kinematics.

Combining Eq. (4.2) and Eq. (4.3) gives

$$\mathbf{l}_{0i} = \mathbf{p} - \mathbf{e}_i, \quad (4.7)$$

with $\mathbf{e}_i = \mathbf{a}_i + d_i \mathbf{d}_{0i} - \mathbf{b}_i$.

Taking the squares of both sides of Eq. (4.7) yields

$$\mathbf{p}^T \mathbf{p} - 2\mathbf{p}^T \mathbf{e}_i + \mathbf{e}_i^T \mathbf{e}_i = l^2. \quad (4.8)$$

Then, writing Eq. (4.8) three times, once for each $i = 1, 2$ and 3 , gives three equations of \mathbf{p} . Each equation represents a sphere of radius l whose center is located at the endpoint N_i of a vector defined by point O and \mathbf{e}_i . The intersection of these three spheres gives the solutions of forward kinematics.

Subtracting Eq. (4.8) for $i = 1$ from Eq. (4.8) for $i = 2$ and 3 respectively yields

$$\mathbf{p}^T (\mathbf{e}_2 - \mathbf{e}_1) - \delta_2 = 0, \quad (4.9)$$

$$\mathbf{p}^T (\mathbf{e}_3 - \mathbf{e}_1) - \delta_3 = 0, \quad (4.10)$$

where $\delta_2 = (\mathbf{e}_2^T \mathbf{e}_2 - \mathbf{e}_1^T \mathbf{e}_1) / 2$ and $\delta_3 = (\mathbf{e}_3^T \mathbf{e}_3 - \mathbf{e}_1^T \mathbf{e}_1) / 2$.

Taking notice that Eq. (4.9) and Eq. (4.10) represent two linear equations in three unknowns of p_x , p_y and p_z , from which p_x and p_y can be expressed in terms of p_z as:

$$p_x = k_1 + k_2 p_z, \quad (4.11)$$

$$p_y = k_3 + k_4 p_z, \quad (4.12)$$

where

$$k_1 = S_1 / S, \quad k_2 = S_2 / S,$$

$$k_3 = S_3 / S, \quad k_4 = S_4 / S,$$

with

$$\begin{aligned}
S &= (e_{2x} - e_{1x})(e_{3y} - e_{1y}) - (e_{3x} - e_{1x})(e_{2y} - e_{1y}), \\
S_1 &= \delta_2(e_{3y} - e_{1y}) - \delta_3(e_{2y} - e_{1y}), \\
S_2 &= (e_{3z} - e_{1z})(e_{2y} - e_{1y}) - (e_{2z} - e_{1z})(e_{3y} - e_{1y}), \\
S_3 &= \delta_3(e_{2x} - e_{1x}) - \delta_2(e_{3x} - e_{1x}), \\
S_4 &= (e_{2z} - e_{1z})(e_{3x} - e_{1x}) - (e_{3z} - e_{1z})(e_{2x} - e_{1x}).
\end{aligned}$$



Then, substituting (4.11) and (4.12) back into Eq. (4.8) for $i = 1$, yields a quadratic polynomial equation of variable p_z :

$$T_1 p_z^2 + 2T_2 p_z + T_3 = 0, \quad (4.13)$$

where

$$\begin{aligned}
T_1 &= k_2^2 + k_4^2 + 1, \\
T_2 &= k_1 k_2 + k_3 k_4 - e_{1x} k_2 - e_{1y} k_4 - e_{1z}, \\
T_3 &= k_1^2 + k_3^2 - 2e_{1x} k_1 - 2e_{1y} k_3 + e_{1x}^2 + e_{1y}^2 + e_{1z}^2 - l^2.
\end{aligned}$$

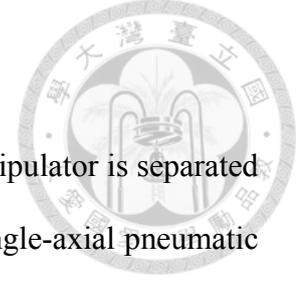
Solving Eq. (4.13) results in

$$p_z = \frac{-T_2 \pm \sqrt{T_2^2 - T_1 T_3}}{T_1}. \quad (4.14)$$

In Eq. (4.14), the two solutions imply two corresponding positions of the moving platform on the z -axis, and which stand for two different configurations of the manipulator. However, only the point below the actuator matches the current assembly of the mechanism. Therefore, the negative square root is the practical solution for the unique feasible configuration of the manipulator. Thus, Eq. (4.11), Eq. (4.12) and the negative version of Eq. (4.14) form the forward kinematic equation of the manipulator:

$$\begin{aligned}
p_x &= k_1 + k_2 p_z, \\
p_y &= k_3 + k_4 p_z, \\
p_z &= \frac{-T_2 - \sqrt{T_2^2 - T_1 T_3}}{T_1}.
\end{aligned} \quad (4.15)$$

Chapter 5 Controller Design



The controller design of the three-axial pneumatic parallel manipulator is separated into two parts. The first part is about the controller design of the single-axial pneumatic cylinder system. The proposed controller consists of an inner pressure control loop and an outer position control loop. The second part presents the control design for the overall three-axial manipulator system. The overall control scheme uses the inverse dynamics control approach to decouple the nonlinear manipulator system and realizes the joint space position control with an outer-loop controller. In the end, the position tracking control of the manipulator end-effector is attained by using the inverse kinematics combined with the joint space tracking control system.

5.1 Control Strategy of the Single-Axial Pneumatic Servo System

In general, the dynamic model of a pneumatic cylinder system can be described by the following equations:

$$\dot{m}_a = f_a(P_a, P_s, P_{atm}, u),$$

$$\dot{m}_b = f_b(P_b, P_s, P_{atm}, u),$$

$$\dot{P}_a = -\frac{kP_a A_p}{V_{0,a} + A_p y} \dot{y} + \frac{kRT}{V_{0,a} + A_p y} \dot{m}_a,$$

$$\dot{P}_b = \frac{kP_b A_p}{V_{0,b} + A_p(L-y)} \dot{y} + \frac{kRT}{V_{0,b} + A_p(L-y)} \dot{m}_b,$$

$$M\ddot{y} + F_f = F_p,$$



where

\dot{m}_a, \dot{m}_b are the mass flow rates into chambers a and b ,

P_a, P_b are the chamber pressures of a and b ,

u is the input control signal,

\dot{P}_a, \dot{P}_b are the overall pressure dynamics in chambers a and b ,

$V_{0,a}, V_{0,b}$ are the inactive residual volumes in chambers a and b ,

y is the piston displacement,

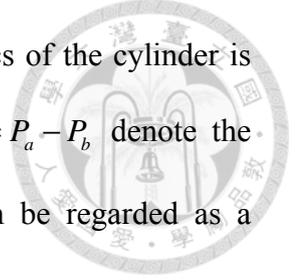
F_f is the friction force, and

F_p is the force acting on the cylinder piston.

The constants for the dynamic model of a pneumatic cylinder system are summarized in the Table 5.1.

Table 5.1 Model parameters of the pneumatic servo system

Parameter	Description	Value
P_s	Absolute supply pressure	7.013 bar
P_{atm}	Ambient atmospheric pressure	1.013 bar
k	Specific heat ratio	1.4
R	Ideal gas constant	287 J/(kg·K)
T	Absolute temperature of air	293 K
A_p	Piston area of rodless cylinder (25 mm bore)	$4.9087 \times 10^{-4} \text{ m}^2$
L	Cylinder stroke length	0.5 m
M	Moving mass	0.732 kg



From the system model, it is obvious that the motion dynamics of the cylinder is dominated by the pressure difference of the chambers. Let $\Delta P = P_a - P_b$ denote the pressure difference of the cylinder, the piston-load dynamics can be regarded as a simple 2nd-order system which is expressed as

$$M\ddot{y} + F_f = A_p \cdot \Delta P. \quad (5.1)$$

Eq. (5.1) implies that the control input into the system is ΔP ; in other words, ΔP controls the dynamics of the cylinder piston. Thus, the position control of the piston can be achieved by designing a control law for the pressure difference ΔP . However, the pressure difference of the cylinder is regulated by the servo valve rather than directly commanded by the position controller. Thus, in order to set the required pressure difference that controls the piston position, it is necessary to include an inner control loop for pressure regulation. The proposed control scheme is shown in Fig. 5.1.

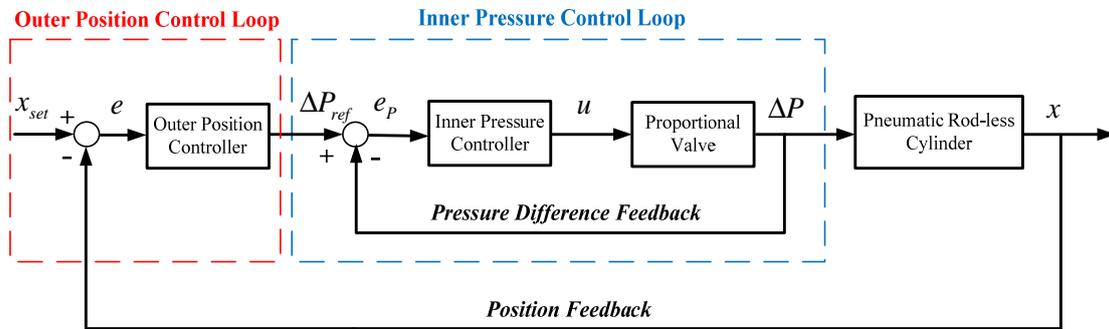


Fig. 5.1 Dual-loop control scheme for position control of the pneumatic cylinder system

For the inner-loop control design, a proportional-integral (PI) controller with anti-windup function, to conquer the phenomenon when the controlled actuator reaches the saturation limits of the actuation that it can supply, is used in the inner loop to control the pressure difference in the cylinder. The selected anti-windup method is the back-calculation approach [54] which is illustrated in Fig. 5.2.

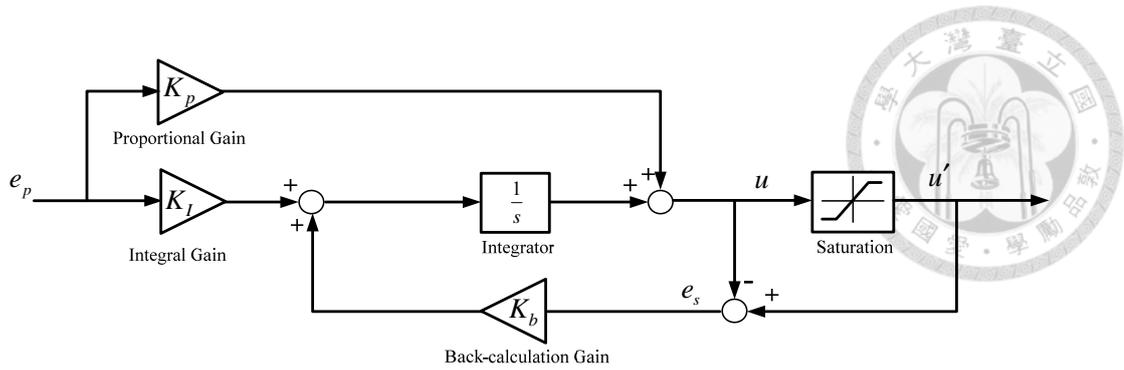


Fig. 5.2 PI control with back-calculation anti-windup method

$$u(t) = K_{p,in} \cdot e_p(t) + \left[K_{i,in} + (K_{b,in} \cdot e_s(t)) \right] \cdot \int_0^t e_p(t) d\tau, \quad (5.2)$$

where $K_{p,in}$, $K_{i,in}$ and $K_{b,in}$ are the proportional, integral and back-calculation gains of the inner controller.

For the outer-loop control design, a proportional-derivative (PD) control is implemented in the outer loop as the position controller.

$$\Delta P_{ref} = K_{p,out} \cdot e(t) + K_{d,out} \cdot \frac{d}{dt} e(t), \quad (5.3)$$

where $K_{p,out}$ and $K_{d,out}$ are the proportional and derivative gains of the outer-loop controller.

Finally, the schematic diagram of the closed-loop control system for the single-axial pneumatic cylinder is shown in Fig. 5.3.

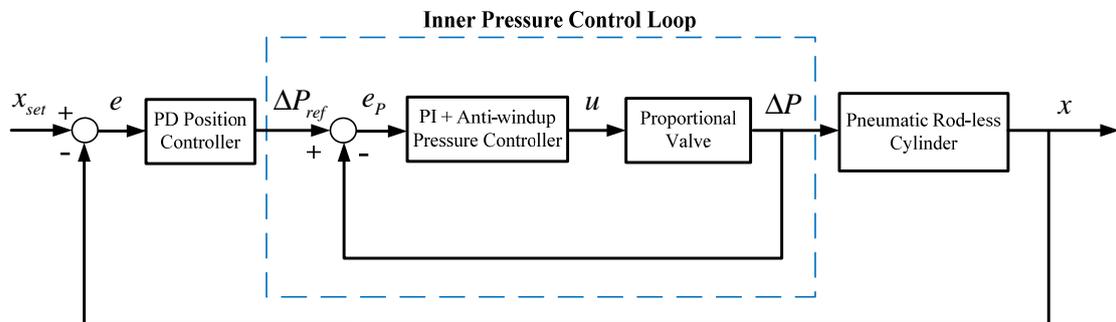


Fig. 5.3 Schematic diagram of position control of single-axial pneumatic cylinder system

5.2 Control Strategy of the Three-Axial Pneumatic Parallel Manipulator



Inverse dynamics control (IDC) or the so-called computed torque control is one of the most common control strategies for robotic manipulators. The idea of inverse dynamics control is to include the dynamics of the manipulator system into the control design and use the manipulator dynamics to decouple and linearize the nonlinear system [55]. The dynamic equations for the manipulator can be expressed in the matrix form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}, \quad (5.4)$$

$$\ddot{\mathbf{q}} = \mathbf{u}_q, \quad (5.5)$$

where

$\mathbf{q} = [y_A \quad y_B \quad y_C]^T$ is the vector of actuated joint displacements,

$\mathbf{M}(\mathbf{q})$ is the inertia matrix which is symmetric and positive definite,

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the matrix of centrifugal and Coriolis forces,

$\mathbf{G}(\mathbf{q})$ is the gravity force vector.

$\boldsymbol{\tau} = [f_A \quad f_B \quad f_C]^T$ is the vector of actuated joint forces, and

\mathbf{u}_q is the control input.

Let $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})$ be the nonlinear coupling term, the inner-loop control scheme of inverse dynamics control is shown in Fig. 5.4. The force-pressure transformation procedure shown in the figure is the calculation of the reference pressure difference from the computed forces for each cylinder actuator, and which can be made by dividing the force by the piston area.

After the decoupling process using inverse dynamics control, the system structure allows a design of the outer-loop control \mathbf{u}_q with a proportional-derivative (PD)

control scheme which can be described as

$$\mathbf{u}_q = \ddot{\mathbf{q}}_d + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e}, \quad (5.6)$$

where \mathbf{q}_d denotes the desired joint trajectory, $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$ is defined as the vector of joint tracking errors, and \mathbf{K}_P , \mathbf{K}_D are proportional and derivative gain matrices.

Finally, the overall inverse dynamics control scheme is shown in Fig. 5.5.

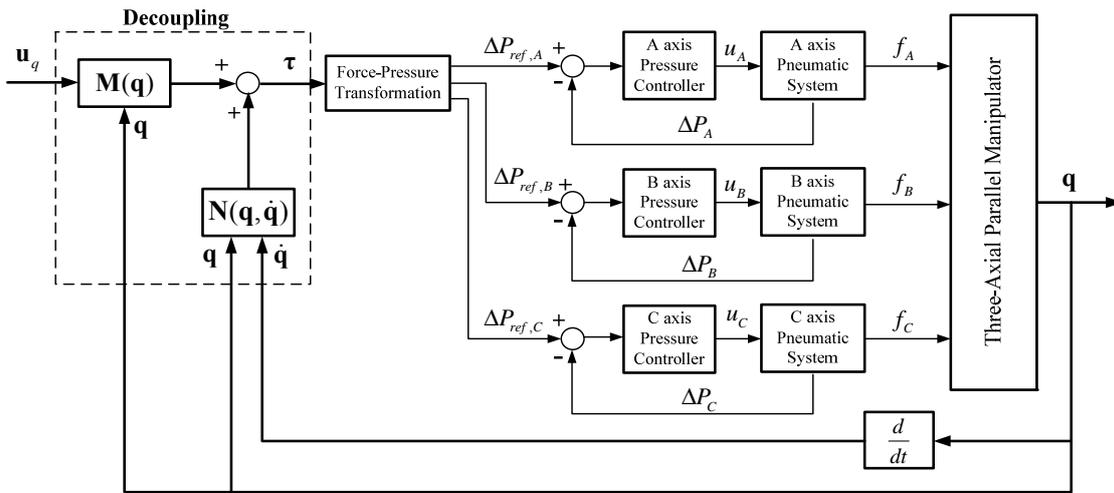


Fig. 5.4 Inner-loop control design of inverse dynamics control for the three-axial pyramidal pneumatic parallel manipulator

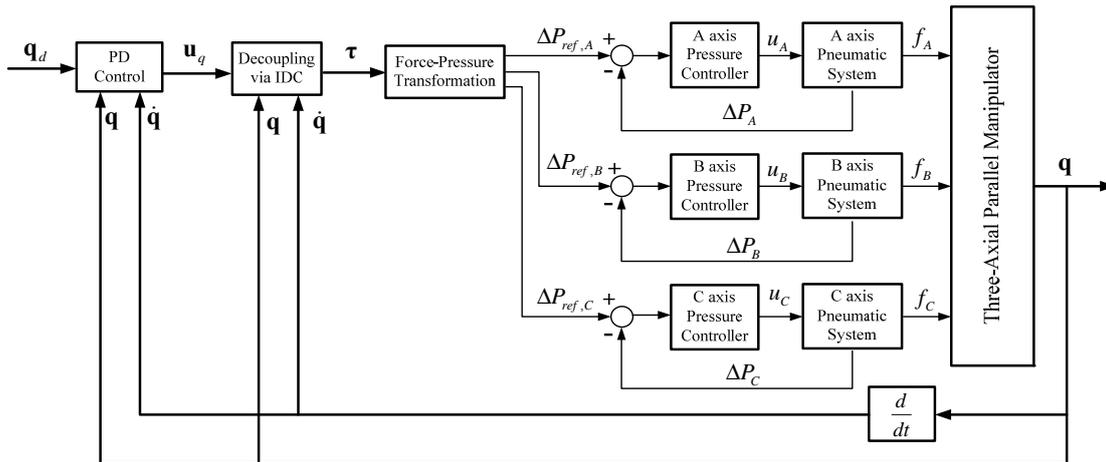


Fig. 5.5 Schematic diagram of inverse dynamics control for the three-axial pyramidal pneumatic parallel manipulator

In order to realize the position tracking control of the manipulator end-effector, the kinematic model is considered and implemented in the control system. First, a desired moving trajectory of end-effector is planned and setup in the control system, then the end-effector trajectory is mapped into the actuated joint path via the inverse kinematics derived in Chapter 4. By using the above control scheme to control the piston position of each cylinder and make them follow the actuated joint path evaluated by the inverse kinematics, the trajectory tracking of the end-effector can be achieved.

The overall control system of the three-axial manipulator is shown in Fig. 5.6, where $\mathbf{x}_d = [x_{set} \ y_{set} \ z_{set}]^T$ is the desired end-effector position, and $\mathbf{x} = [x \ y \ z]^T$ represents the output end-effector position measured from the origin of the fixed frame coordinate system.

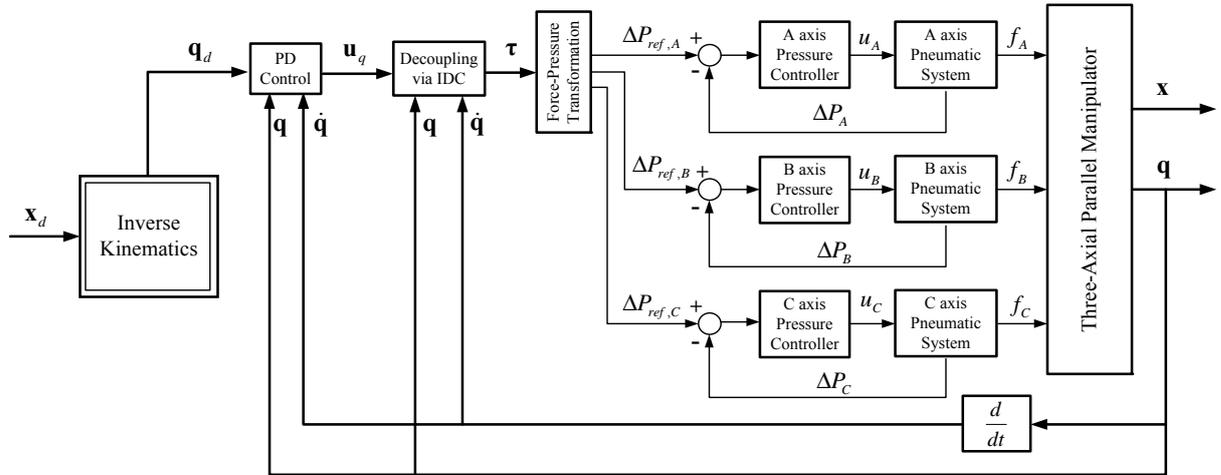
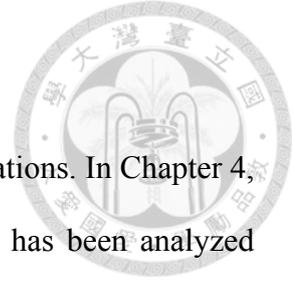


Fig. 5.6 Schematic diagram of the overall control system

Chapter 6 Simulations and Experiments



This chapter is composed of three parts. The first part is verifications. In Chapter 4, the kinematic model of three-axial pneumatic parallel manipulator has been analyzed and derived. Through inputting basic line trajectories, it is helpful to check the correctness and accuracy of the theoretical model. Also, the algorithms of path planning for 3D printing, developed in Chapter 3, are estimated through the kinematic model to predict the movements of actuator on each limb. The second part is simulations. By utilizing the interface of ADAMS and MATLAB/SIMULINK software, co-simulations can be accomplished through inputting the same trajectories in the first part. The third part is experiments. In Chapter 5, the controller design of three-axial pneumatic parallel manipulator has been established, called path tracking control. This control strategy is testified for the effectiveness and performance through all of the same trajectories in the first part. The organization of this chapter is as follows:

6.1 Verifications of Kinematic Model

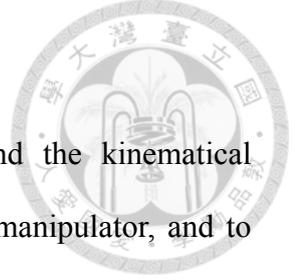
Circle-shape Line Trajectory, Sphere-shape Line Trajectory, Solid Cuboid Trajectory, Solid Polyhedron Trajectory, Solid Complex 3D Trajectory

6.2 Simulations of Three-Axial Pneumatic Parallel Manipulator by ADAMS and SIMULINK

Circle-shape Line Trajectory, Sphere-shape Line Trajectory, Solid Cuboid Trajectory, Solid Polyhedron Trajectory, Solid Complex 3D Trajectory

6.3 Experiments of Three-Axial Pneumatic Parallel Manipulator by Path Tracking Control of End-Effector

Circle-shape Line Trajectory, Sphere-shape Line Trajectory, Solid Cuboid Trajectory, Solid Polyhedron Trajectory, Solid Complex 3D Trajectory



6.1 Verifications of Kinematic Model

The purpose of deriving kinematic models is to understand the kinematical relations between actuator motion and end-effector motion of the manipulator, and to achieve the position control of end-effector in task space. In order to verify the correctness and accuracy of the inverse and forward kinematic models derived in Chapter 4, five trajectories are designed and used in this section, including two basic line trajectories and three trajectories after path planning in Chapter 3.

The verification procedure for validating kinematic models is shown in Fig. 6.1.

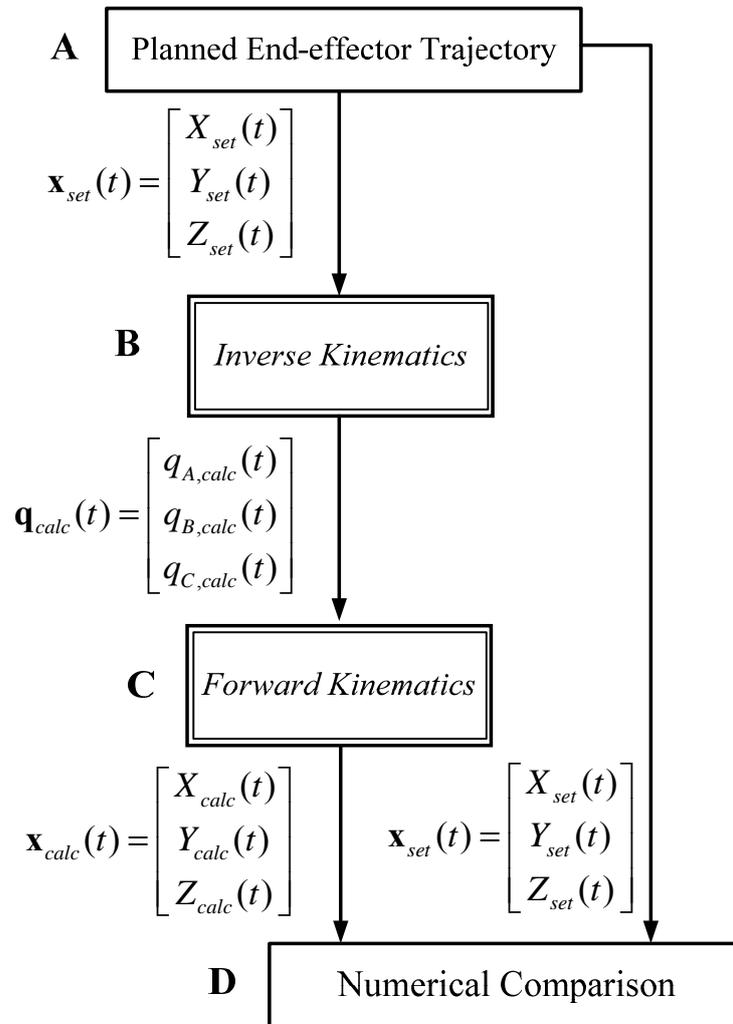


Fig. 6.1 Procedure of model verification for inverse and forward kinematics



The verification procedure in Fig. 6.1 includes 4 steps:

- A. Plan a desired end-effector trajectory ($x_{set}(t)$).
- B. Input the designed end-effector trajectory ($x_{set}(t)$) into the inverse kinematic model and output a set of joint space trajectories ($q_{calc}(t)$) for the actuators.
- C. Input the actuated joint trajectories ($q_{calc}(t)$) into the forward kinematic model and output the computed end-effector trajectory ($x_{calc}(t)$).
- D. Compare the computed end-effector trajectory with the designed trajectory.

$$Error = x_{set}(t) - x_{calc}(t).$$

The verifications are programmed with the MATLAB software. The algorithms for the inverse and forward kinematics are both coded in MATLAB language using the inverse and forward kinematic equations derived in Chapter 4 (see Eq. (4.6) and Eq. (4.15)). The designed geometry parameters of manipulator which are used for the verifications are given in Table 6.1. The following subsections illustrate the verification results of five trajectories.

Table 6.1 Manipulator parameters for kinematic simulations

Parameter	Description	Value
R	Fixed base radius	734 mm
r	Moving platform radius	61 mm
l	Length of kinematic chain	746 mm
α	Actuator layout angle	38 deg
ψ	Offset angle of moving platform	15.68 deg



6.1.1 Circle-shape Line Trajectory

The designed circle-shape line trajectory is shown in Fig. 6.2 and has four segments of the set path.

The corresponding joint trajectories calculated from the inverse kinematics are shown in Fig. 6.3, where Fig. 6.3(a), (b) and (c) represent the output joint trajectories for axes A, B and C, respectively.

Then, the end-effector trajectory based on the calculated actuated joint trajectories is computed from the forward kinematics and compared to the designed end-effector trajectory. The verification results are shown in Fig. 6.4 and Fig. 6.5. The numerical calculation error between the planned and the calculated end-effector trajectories are presented in Fig. 6.6. The calculation errors on each axis are below $\pm 2.0 \times 10^{-5}$ mm.

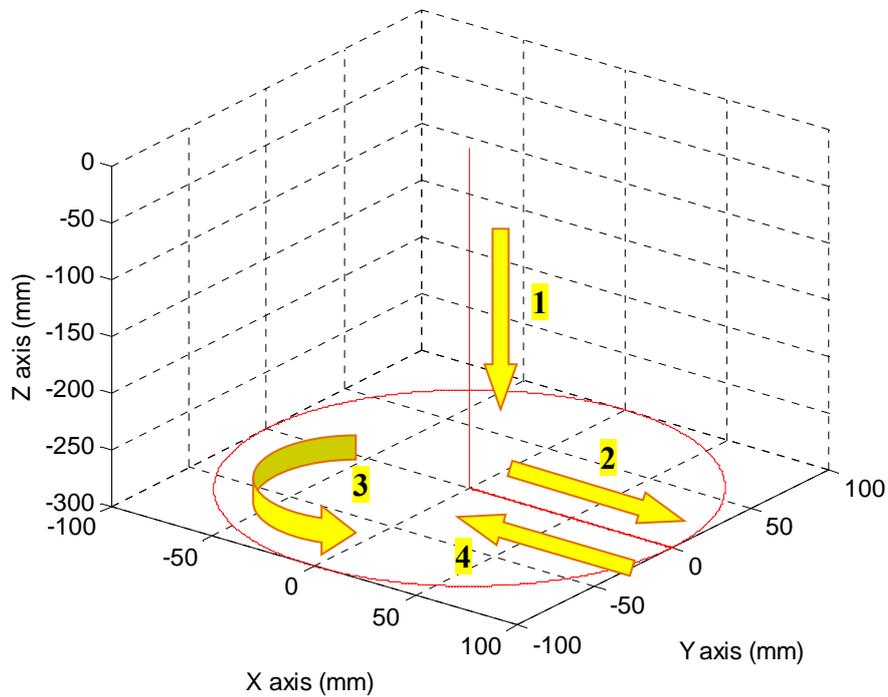


Fig. 6.2 Designed circle-shape line trajectory for end-effector

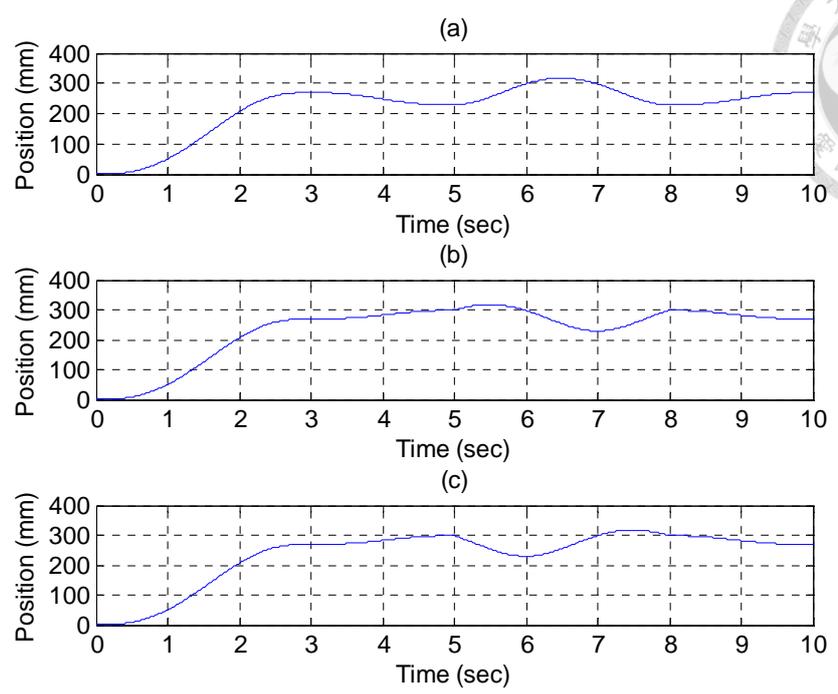
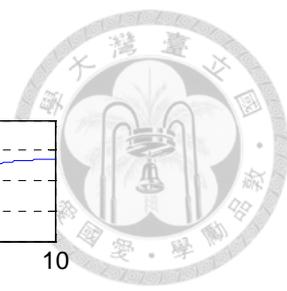


Fig. 6.3 Calculated joint space trajectories via inverse kinematics for the circle-shape line trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator

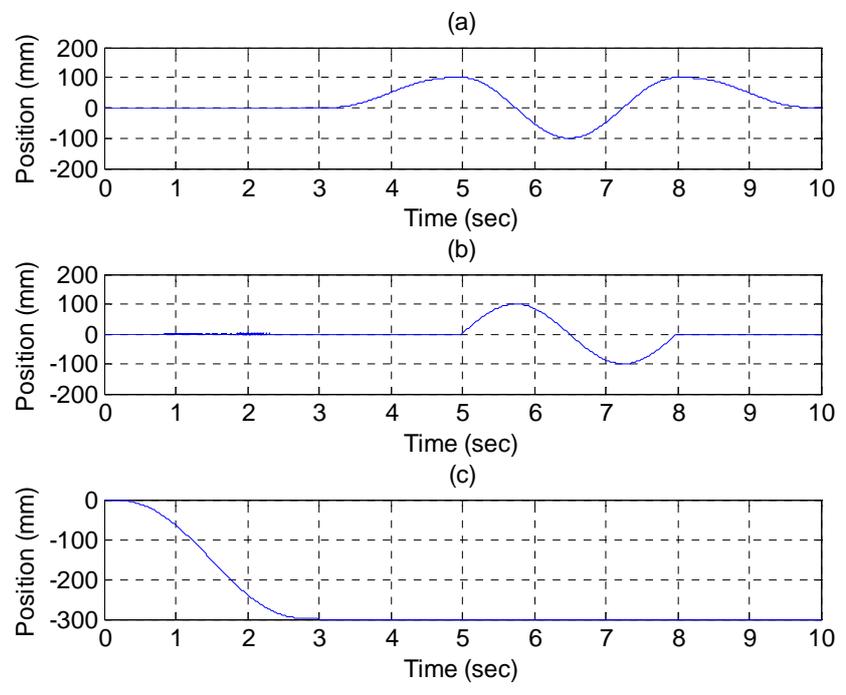


Fig. 6.4 Calculated task space trajectories via forward kinematics for the circle-shape line trajectory: (a) X-axis (b) Y-axis (C) Z-axis

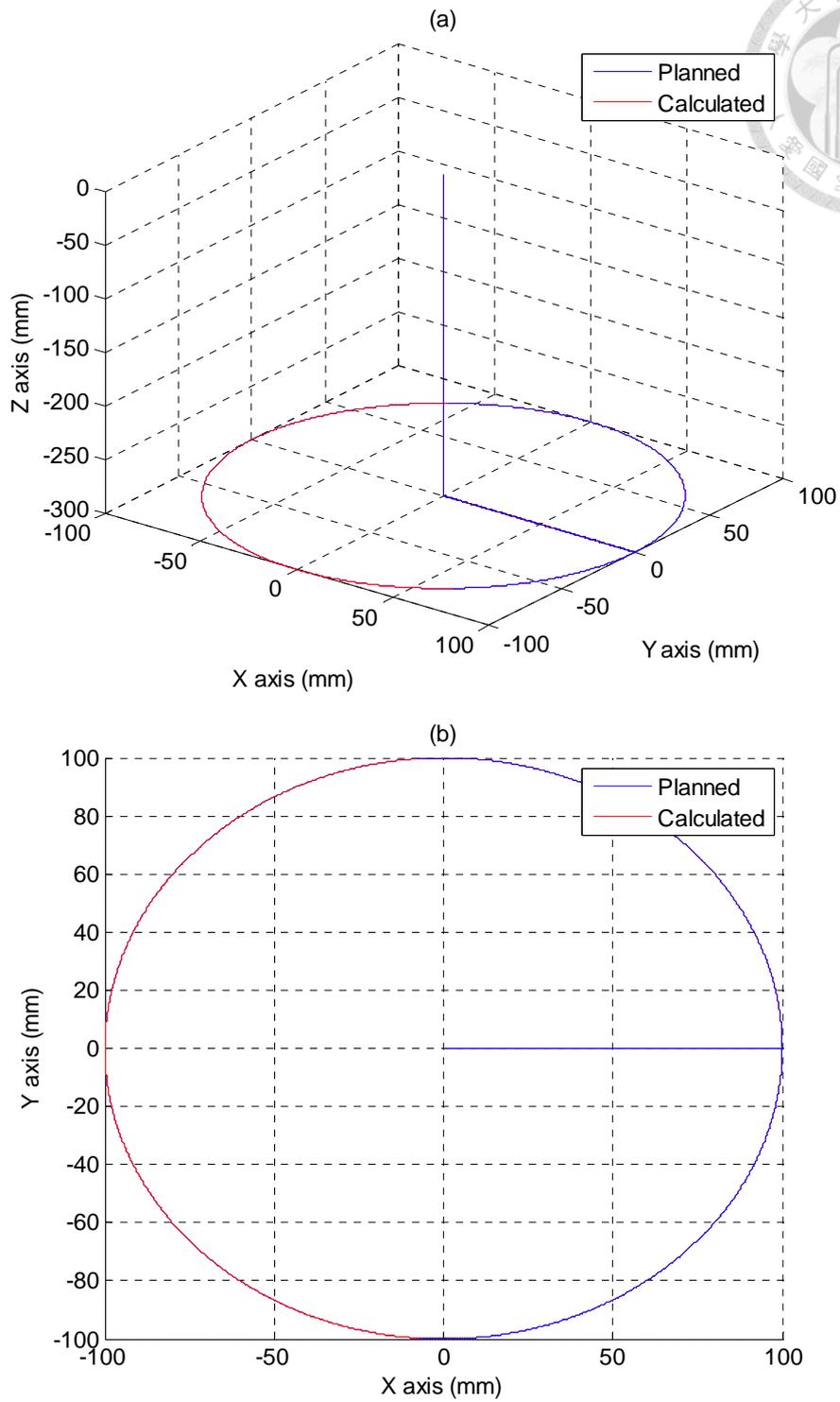
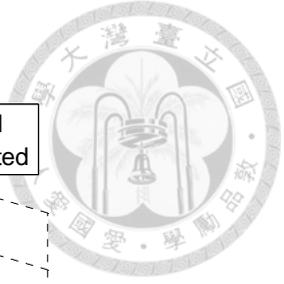


Fig. 6.5 Comparison between the planned and the calculated end-effector for the circle-shape line trajectory: (a) front view (b) top view

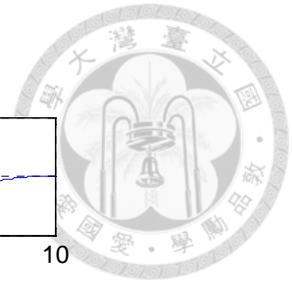
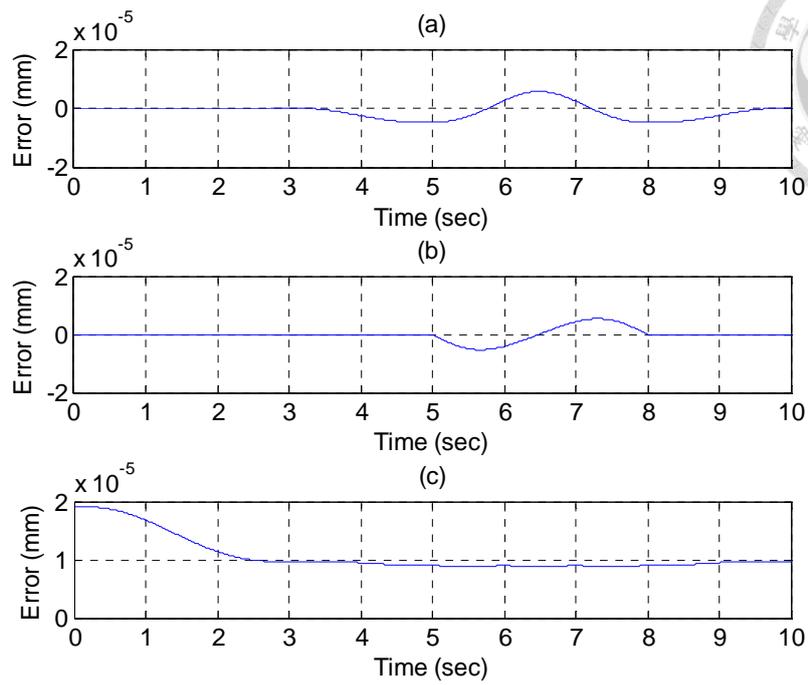


Fig. 6.6 Numerical error between the planned and the calculated end-effector for the circle-shape line trajectory: (a) X-axis (b) Y-axis and (C) Z-axis



6.1.2 Sphere-shape Line Trajectory

The designed sphere-shape line trajectory is shown in Fig. 6.7 and has two segments of the set path.

The corresponding joint trajectories calculated from the inverse kinematics are shown in Fig. 6.8, where Fig. 6.8(a), (b) and (c) represent the output joint trajectories for axes A, B and C, respectively.

Then, the end-effector trajectory based on the calculated actuated joint trajectories is computed from the forward kinematics and compared to the designed end-effector trajectory. The verification results are shown in Fig. 6.9 and Fig. 6.10. The numerical calculation error between the planned and the calculated end-effector trajectories are presented in Fig. 6.11. The calculation errors on each axis are below $\pm 2.0 \times 10^{-5}$ mm.

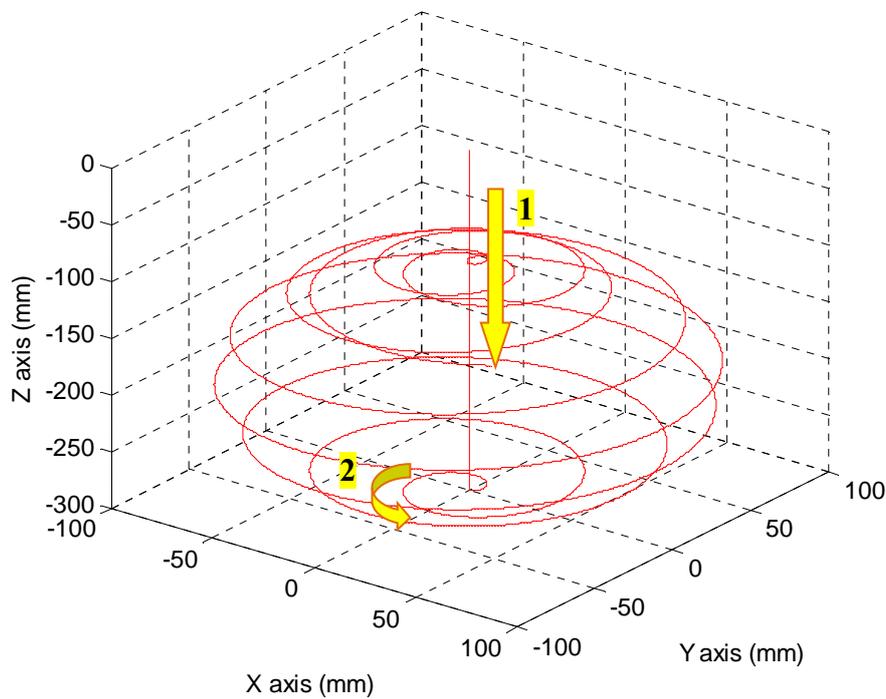


Fig. 6.7 Designed sphere-shape line trajectory for end-effector

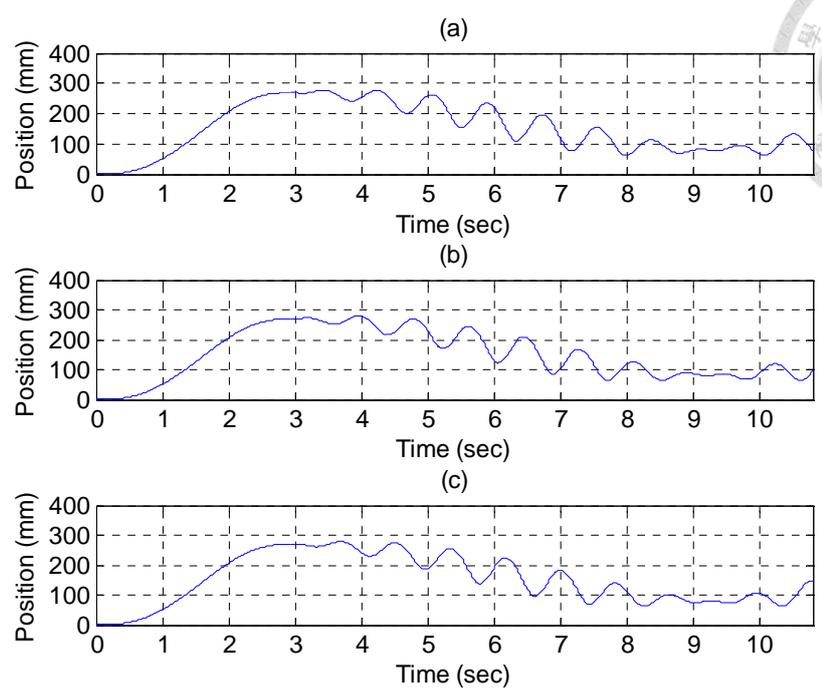
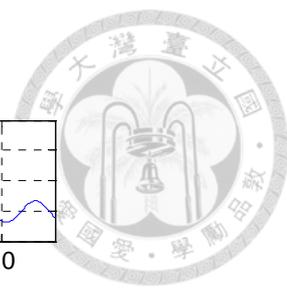


Fig. 6.8 Calculated joint space trajectories via inverse kinematics for the sphere-shape line trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator

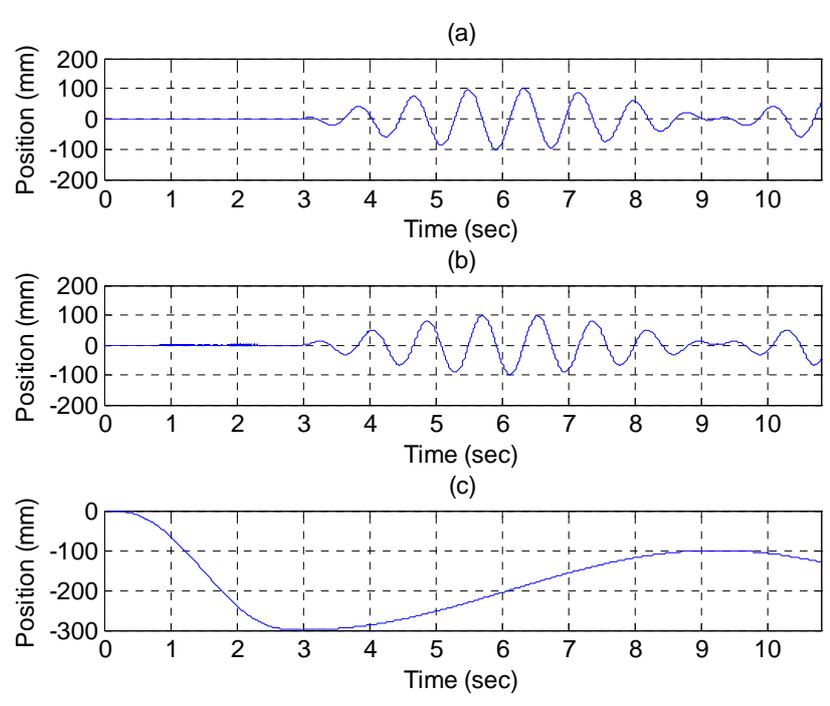


Fig. 6.9 Calculated task space trajectories via forward kinematics for the sphere-shape line trajectory: (a) X-axis (b) Y-axis (c) Z-axis

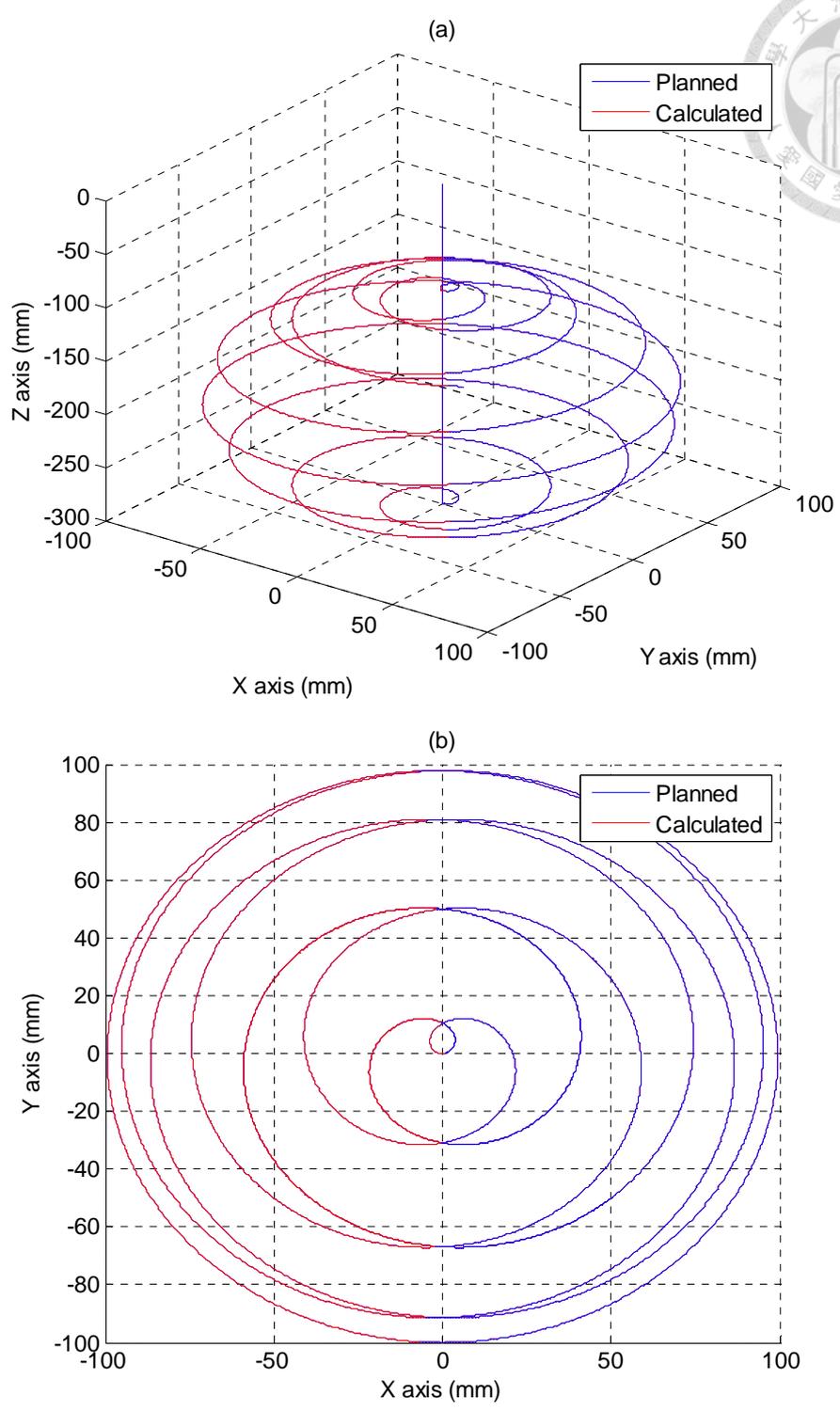
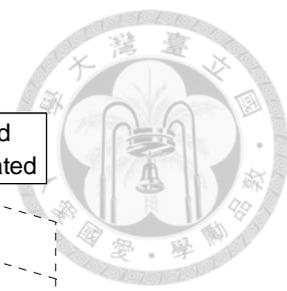


Fig. 6.10 Comparison between the planned and the calculated end-effector for the sphere-shape line trajectory: (a) front view (b) top view

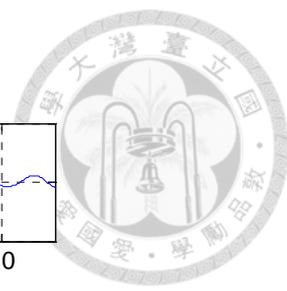
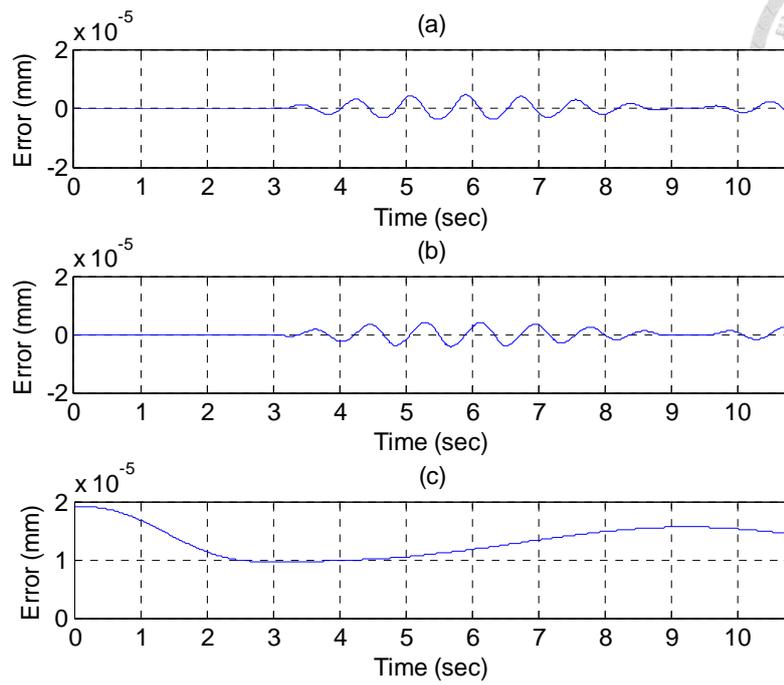


Fig. 6.11 Numerical error between the planned and the calculated end-effector for the sphere-shape line trajectory: (a) X-axis (b) Y-axis and (C) Z-axis

6.1.3 Solid Cuboid Trajectory

The desired printing object of solid cuboid is shown in Fig. 6.12. In order to build the model, the object is sliced into 3 layers and consists of 75 points presenting in Fig. 6.13(a). After using path planning algorithms introduced in Chapter 3, the overall path of the model is established and shown in Fig. 6.13(b). The details on how the algorithms working are depicted in Fig. 6.14, Fig. 6.15 and Fig. 6.16.

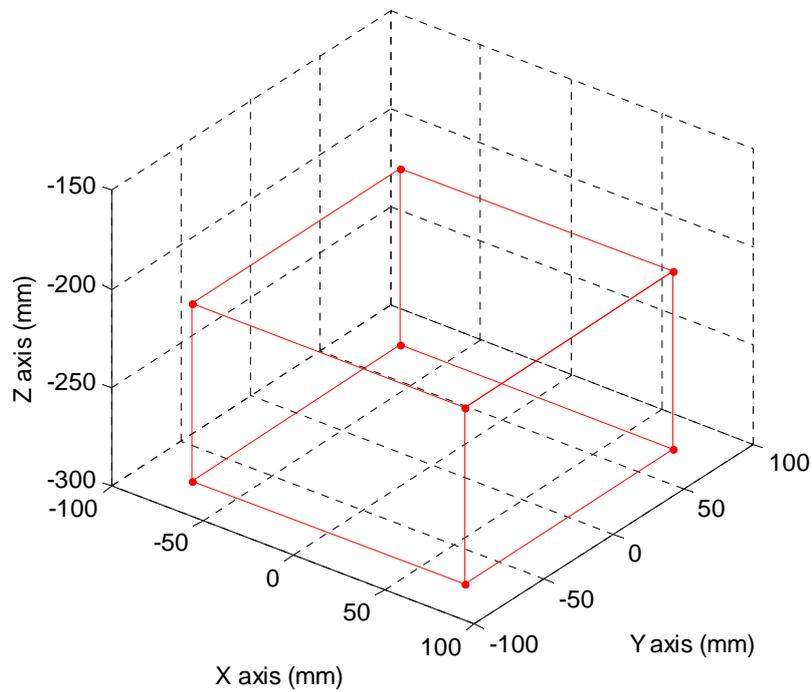


Fig. 6.12 Desired printing object of solid cuboid

Fig. 6.14(a) has 2 sub-paths, \overline{AB} and \overline{CD} , after using the modified-DFS. To calculate minimum distance of linking, the modified-GA is used and shown in Fig. 6.14(b). The linking direction is \overline{BC} and the path planning of 1st layer is accomplished.

In Fig. 6.15(a), continuing to use the last point of 1st-layer path as a start, there is only 1 sub-path, \overline{EF} , after using modified-DFS. No other sub-paths, the minimum distance is 0 and shown in Fig. 6.15(b) as same as in Fig. 6.15(a). After that, the path planning of 2nd layer is finished.

In Fig. 6.16(a), continuing to use the last point of 2nd-layer path as a start, there are 2 sub-paths, \overline{GH} and \overline{IJ} , after using the modified-DFS. To calculate minimum distance of linking, the modified-GA is used and shown in Fig. 6.16(b). The linking direction is \overline{HI} and the path planning of 3rd layer is accomplished, also the last layer.

The corresponding joint trajectories calculated from the inverse kinematics are shown in Fig. 6.17, where Fig. 6.17(a), (b) and (c) represent the output joint trajectories for axes A, B and C, respectively.

Then, the end-effector trajectory based on the calculated actuated joint trajectories is computed from the forward kinematics and compared to the designed end-effector trajectory. The verification results are shown in Fig. 6.18 and Fig. 6.19. The numerical calculation error between the planned and the calculated end-effector trajectories are presented in Fig. 6.20. The calculation errors on each axis are below $\pm 2.0 \times 10^{-5}$ mm. The details on each layer are shown in Fig. 6.21, Fig. 6.22 and Fig. 6.23.

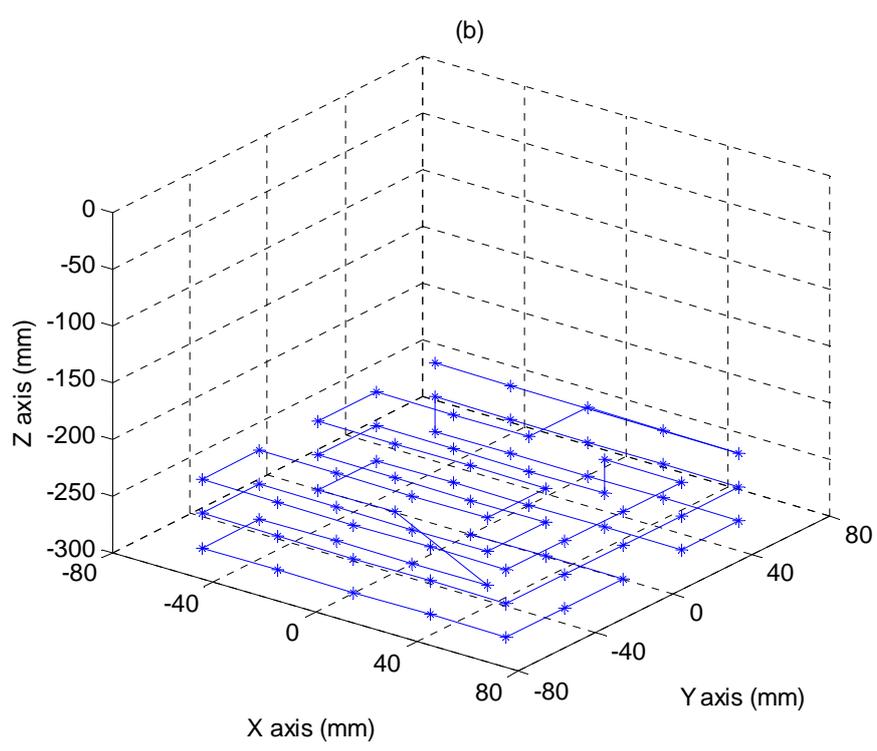
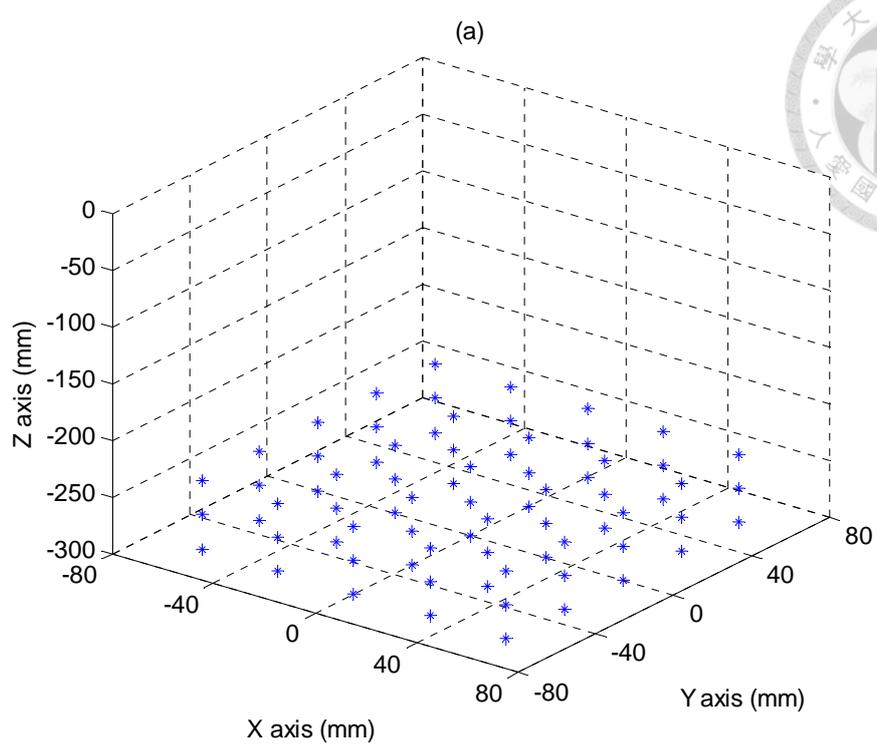


Fig. 6.13 Desired printing object model of solid cuboid:

(a) all points (b) after path planning

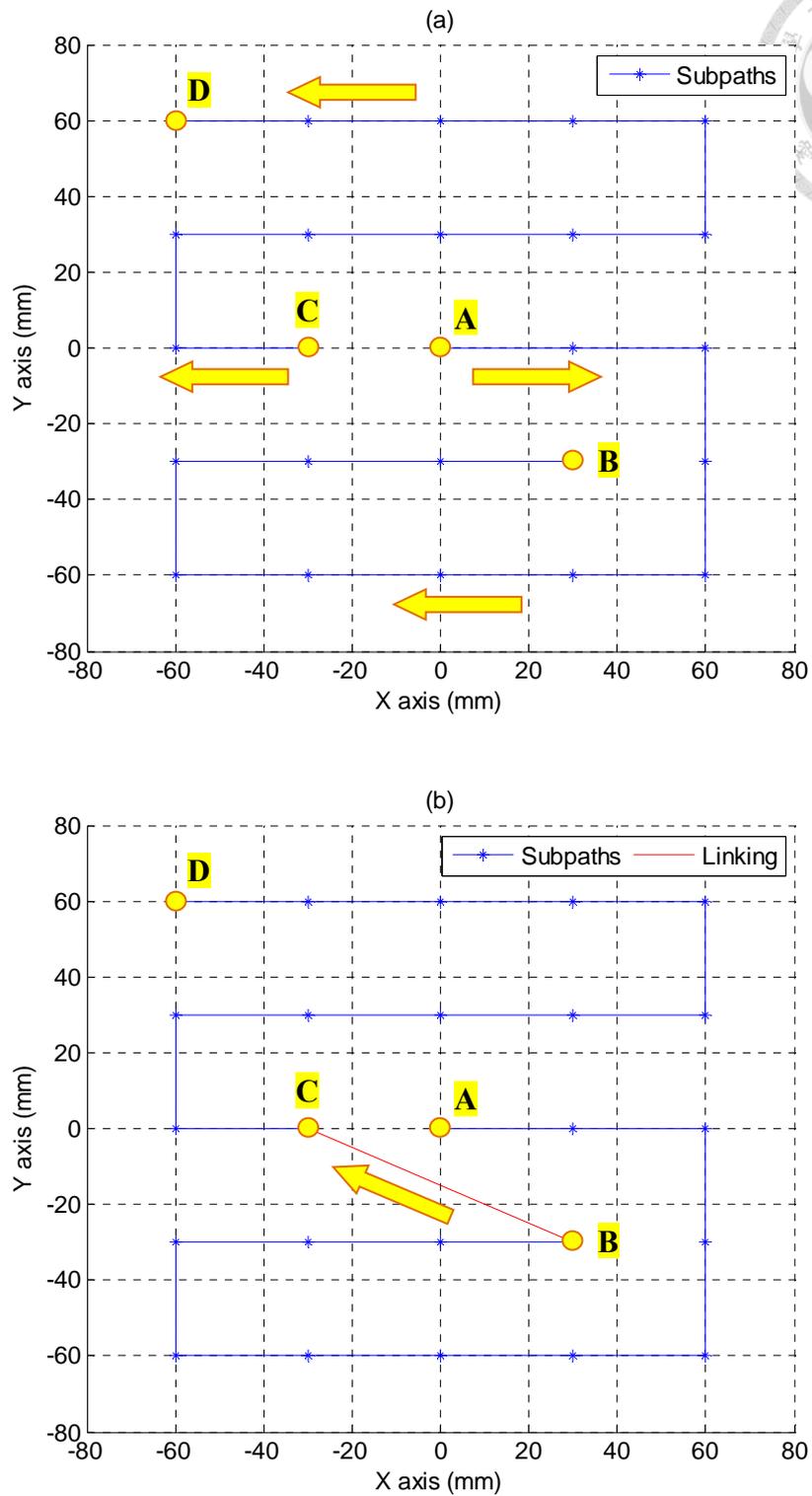
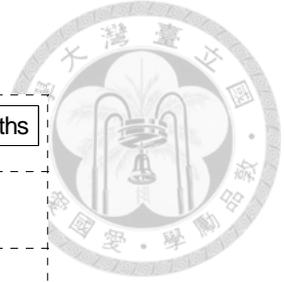


Fig. 6.14 Path planning for solid cuboid trajectory on the 1st layer ($z = -300$ mm):

(a) after modified-DFS (b) after modified-GA

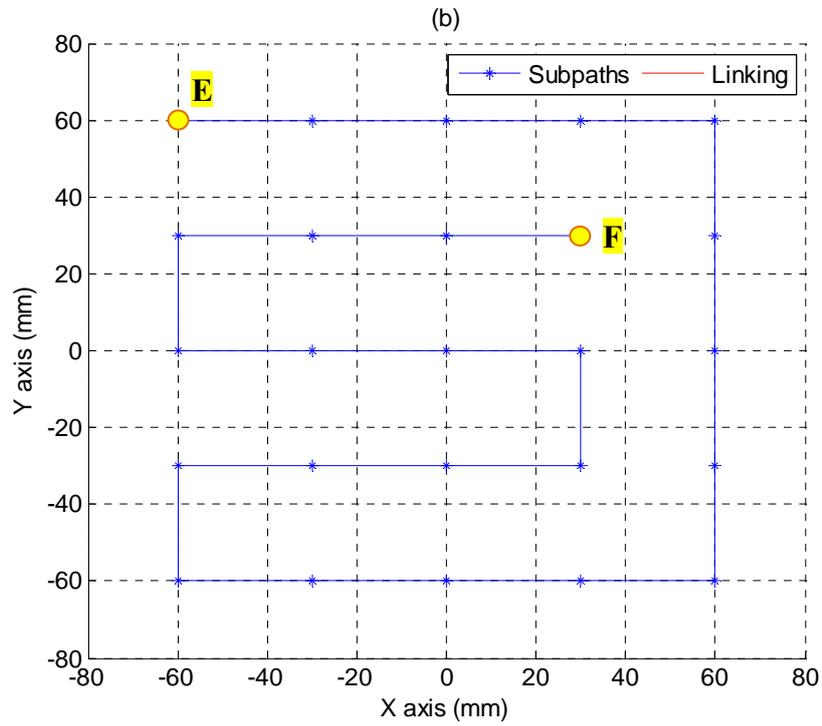
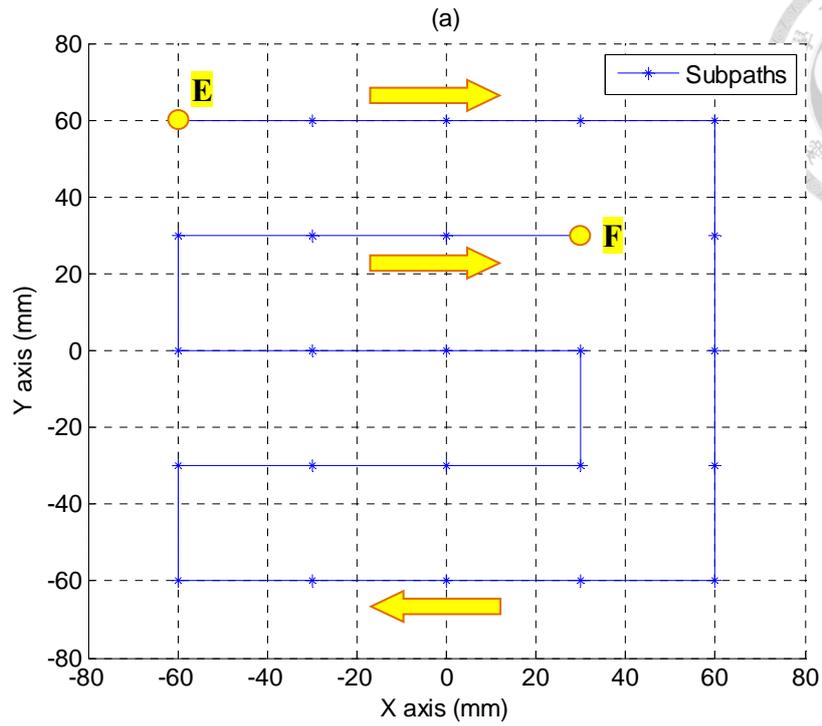
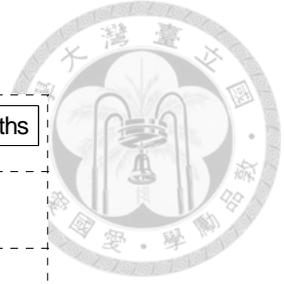


Fig. 6.15 Path planning for solid cuboid trajectory on the 2nd layer ($z = -270$ mm):

(a) after modified-DFS (b) after modified-GA

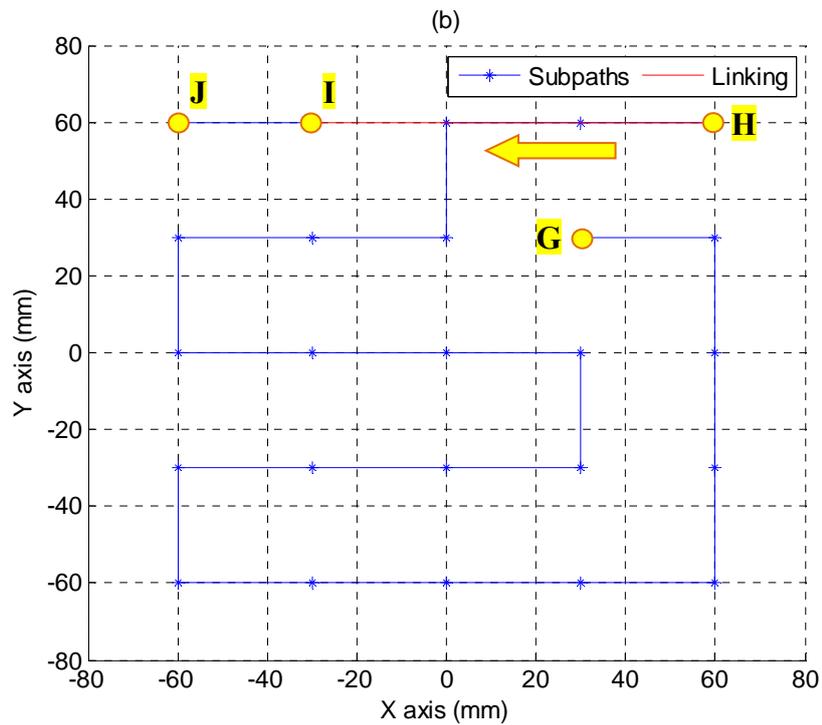
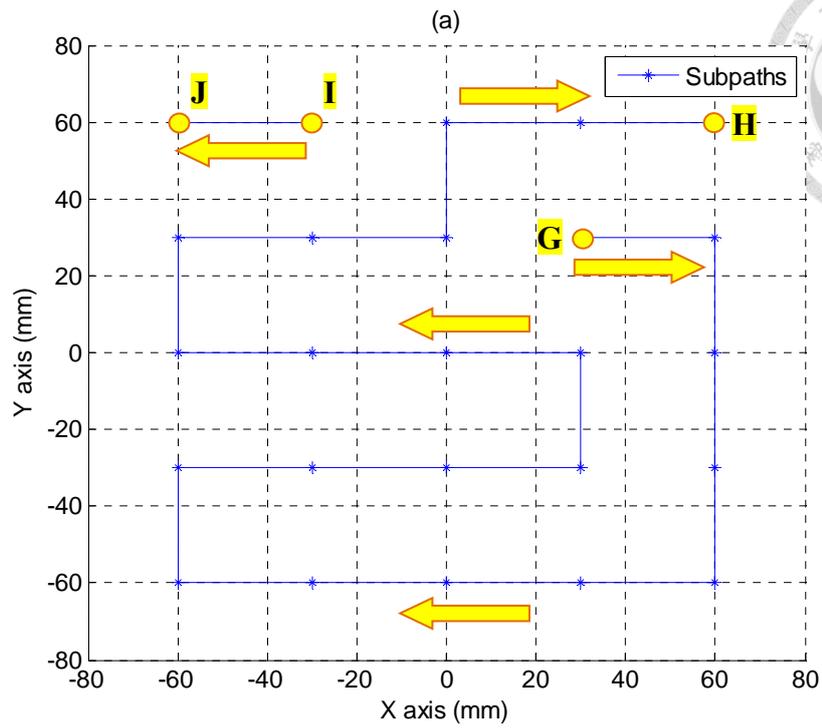
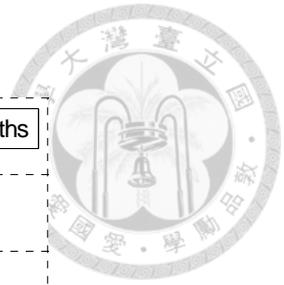


Fig. 6.16 Path planning for solid cuboid trajectory on the 3rd layer ($z = -240$ mm):

(a) after modified-DFS (b) after modified-GA

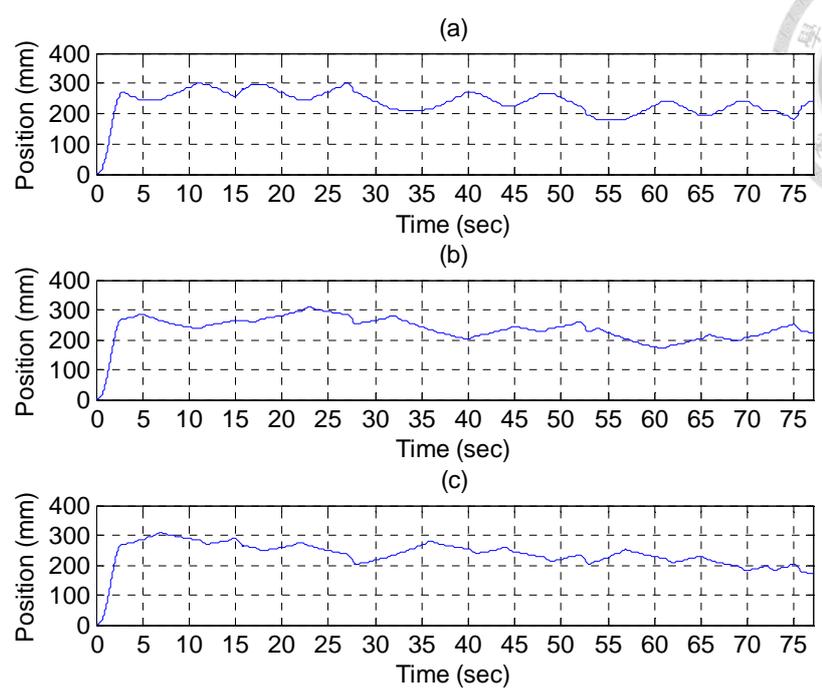
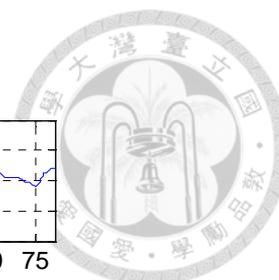


Fig. 6.17 Calculated joint space trajectories via inverse kinematics for the solid cuboid trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator

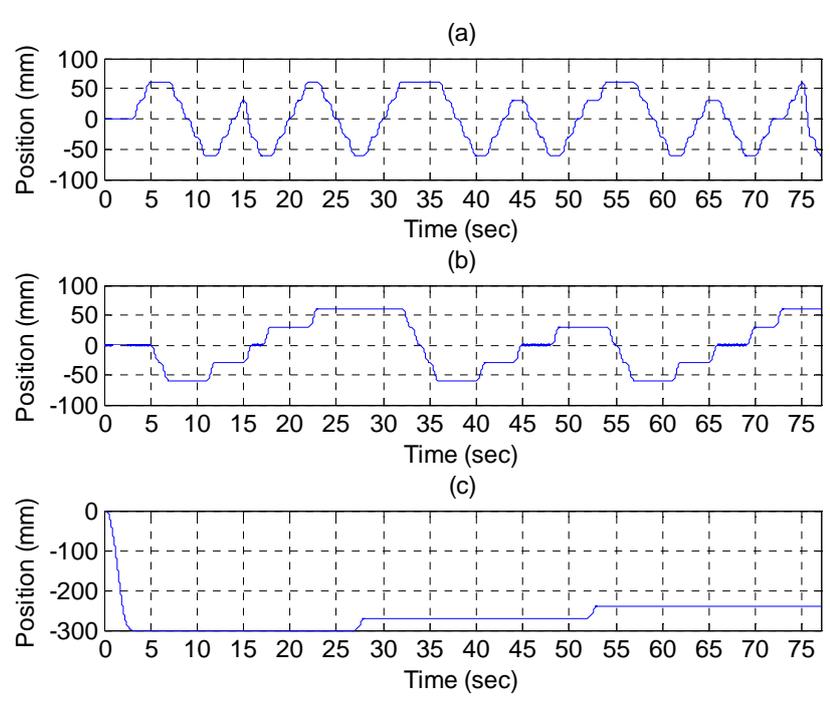


Fig. 6.18 Calculated task space trajectories via forward kinematics for the solid cuboid trajectory: (a) X-axis (b) Y-axis (C) Z-axis

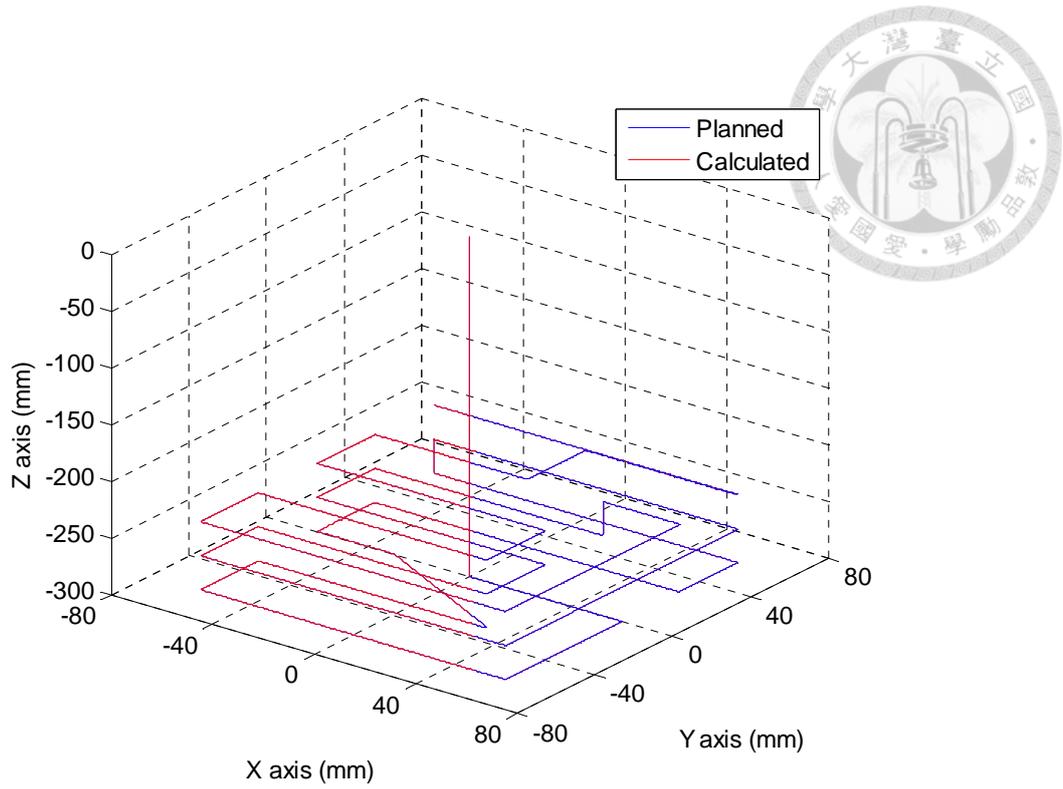


Fig. 6.19 Comparison between the planned and the calculated end-effector for the solid cuboid trajectory.

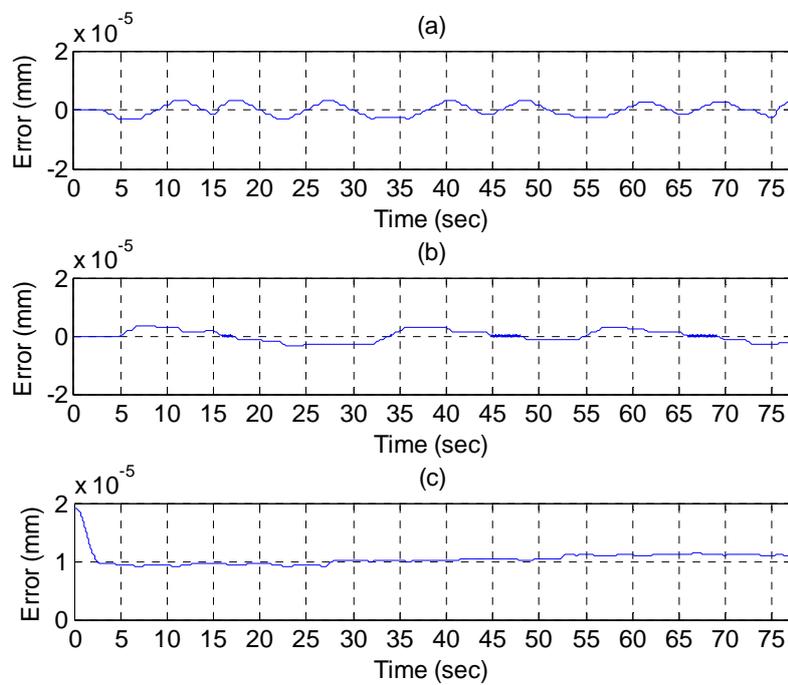


Fig. 6.20 Numerical error between the planned and the calculated end-effector for the solid cuboid trajectory: (a) X-axis (b) Y-axis and (C) Z-axis

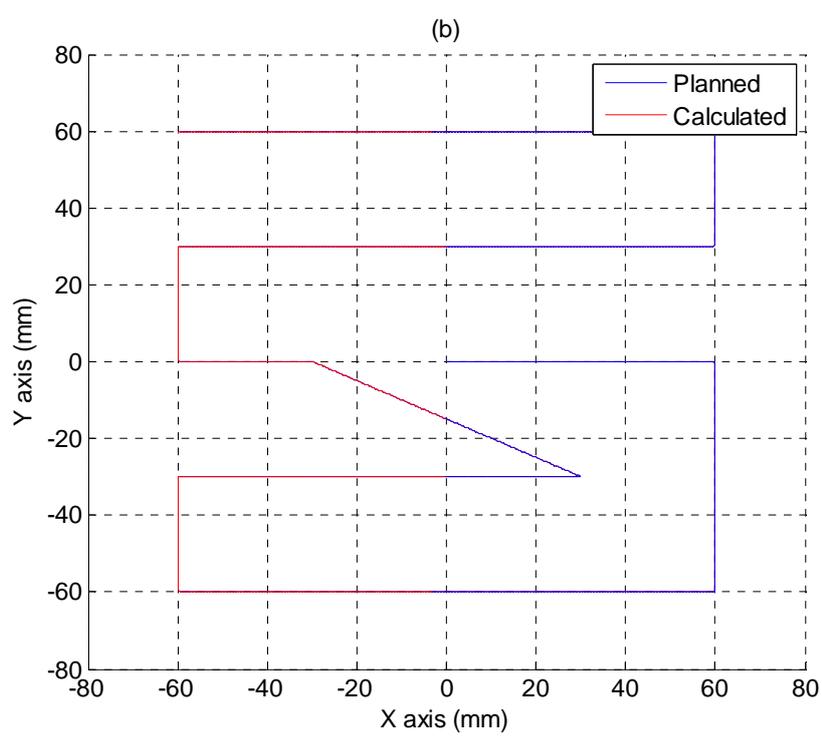
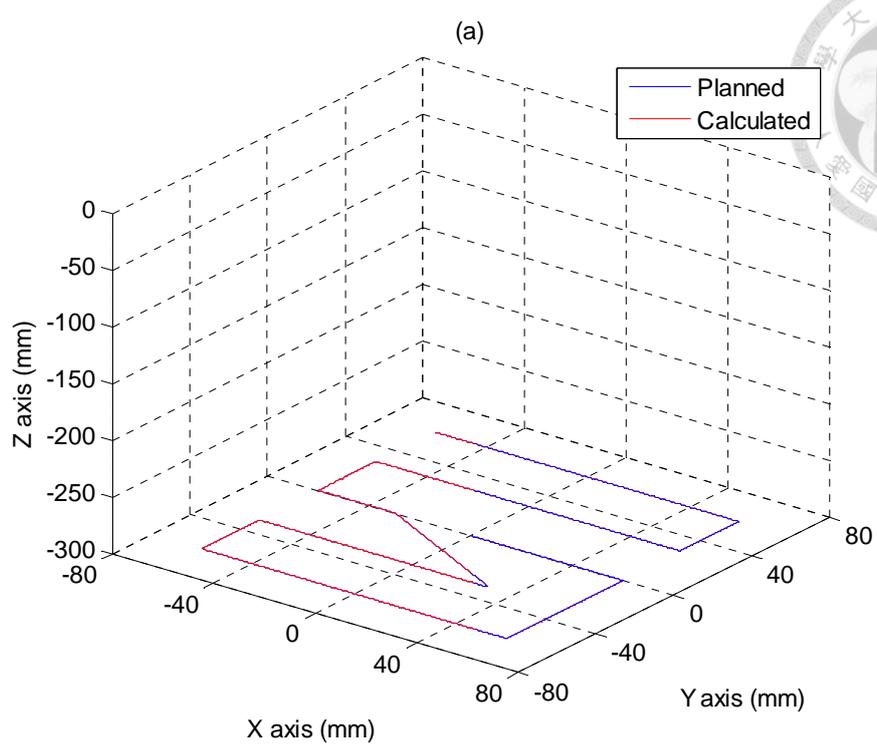
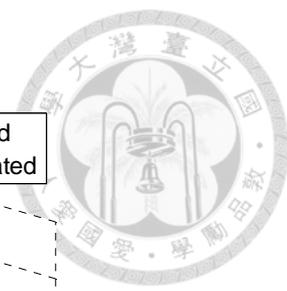


Fig. 6.21 Comparison between the planned and the calculated end-effector for the solid cuboid trajectory of the 1st layer ($z = -300$ mm): (a) front view (b) top view

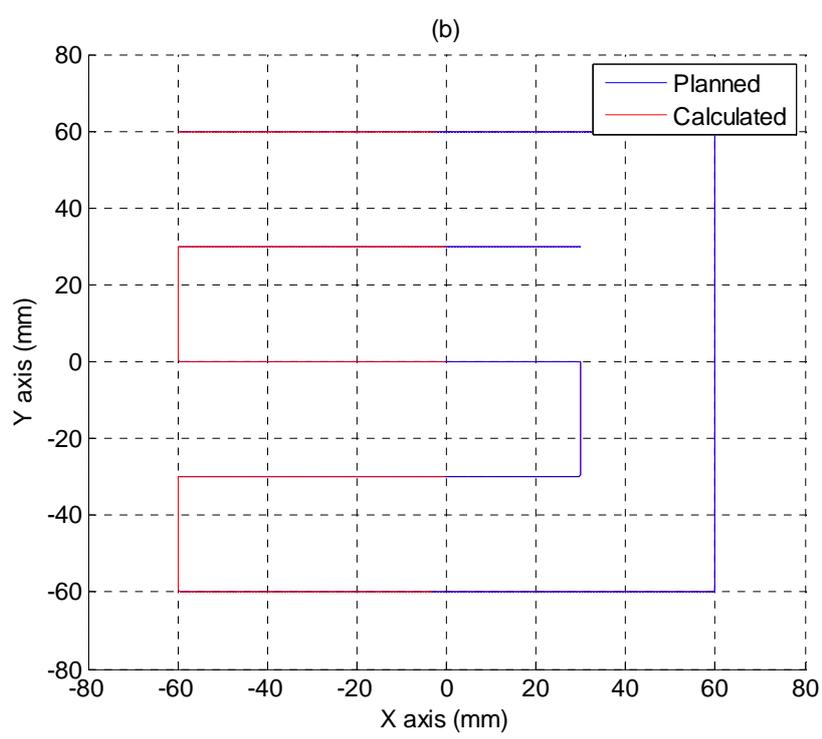
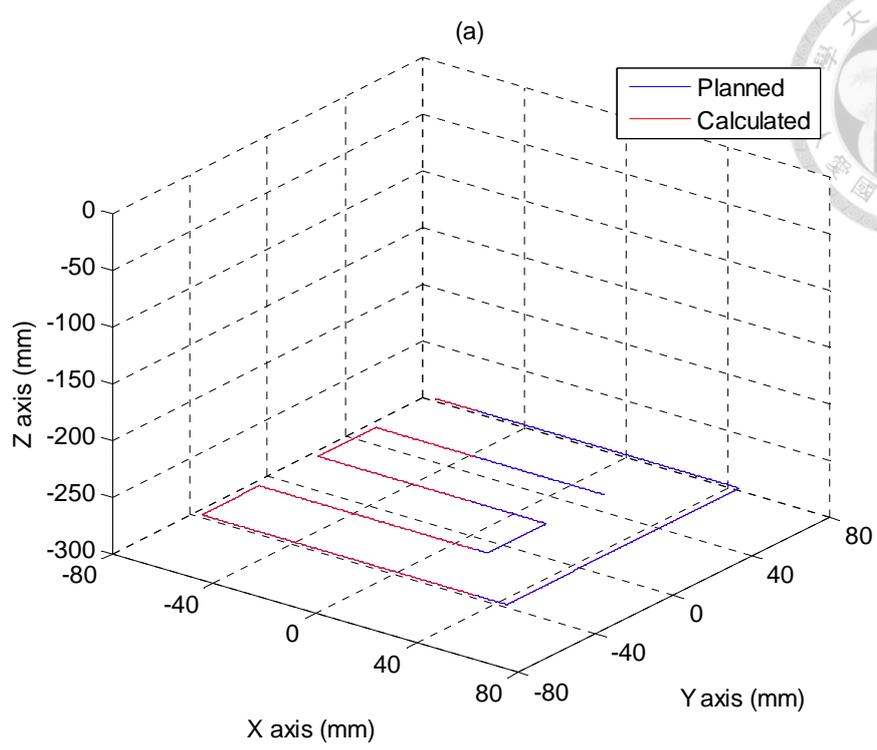
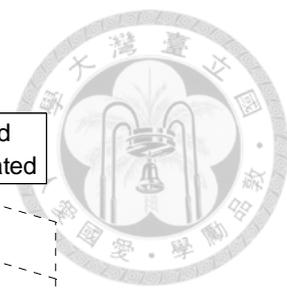


Fig. 6.22 Comparison between the planned and the calculated end-effector for the solid cuboid trajectory of the 2nd layer ($z = -270$ mm): (a) front view (b) top view

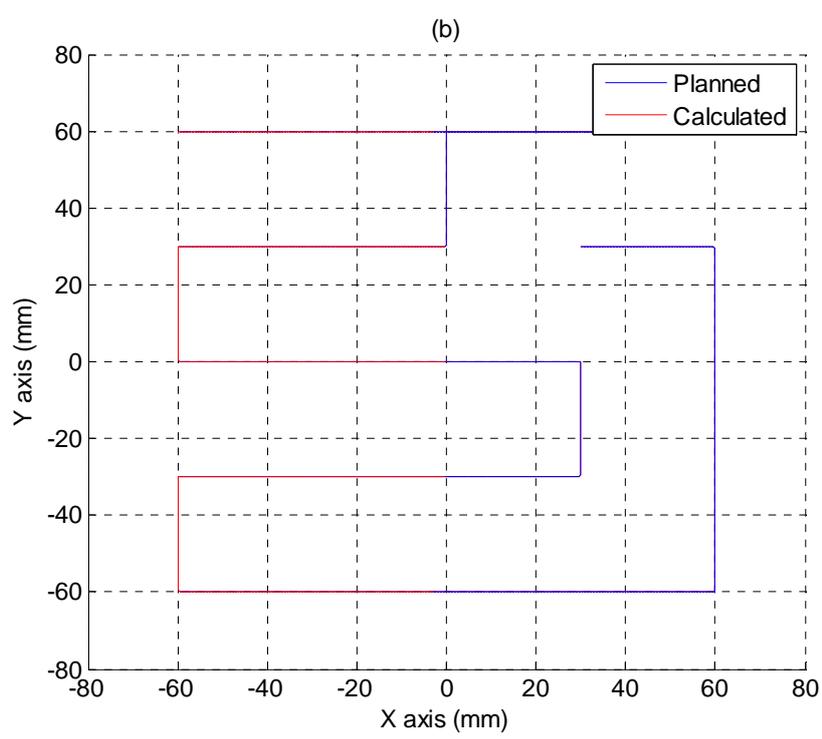
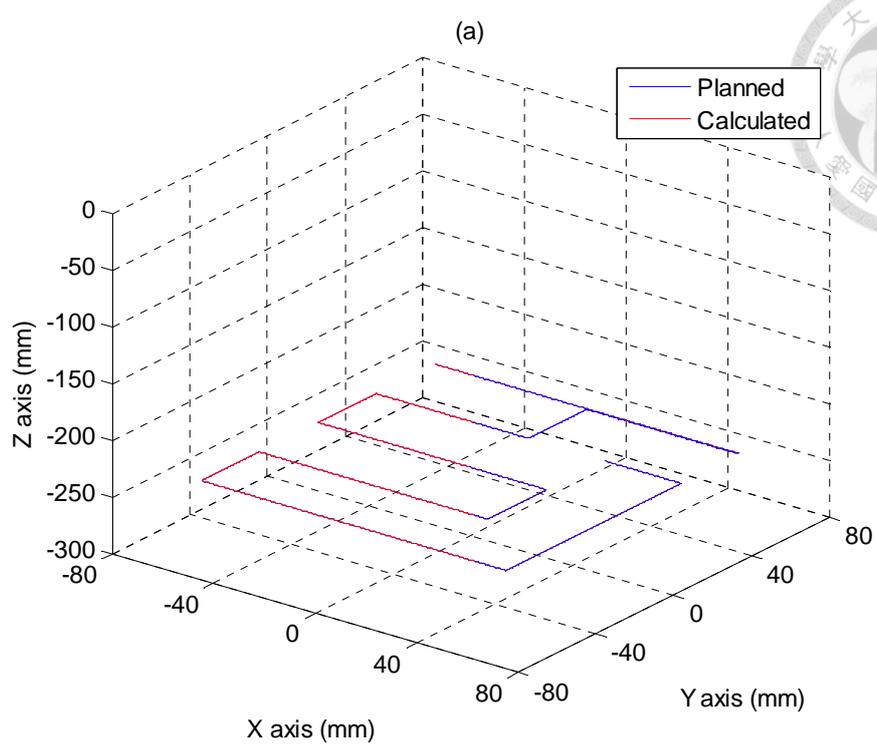
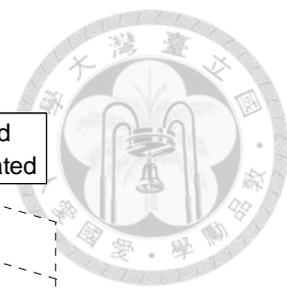


Fig. 6.23 Comparison between the planned and the calculated end-effector for the solid cuboid trajectory of the 3rd layer ($z = -240$ mm): (a) front view (b) top view

6.1.4 Solid Polyhedron Trajectory

The desired printing object of solid polyhedron is shown in Fig. 6.24. In order to build the model, the object is sliced into 3 layers and consists of 160 points presenting in Fig. 6.25(a). After using path planning algorithms introduced in Chapter 3, the overall path of the model is established and shown in Fig. 6.25(b). The details on how the algorithms working are depicted in Fig. 6.26, Fig. 6.27 and Fig. 6.28.

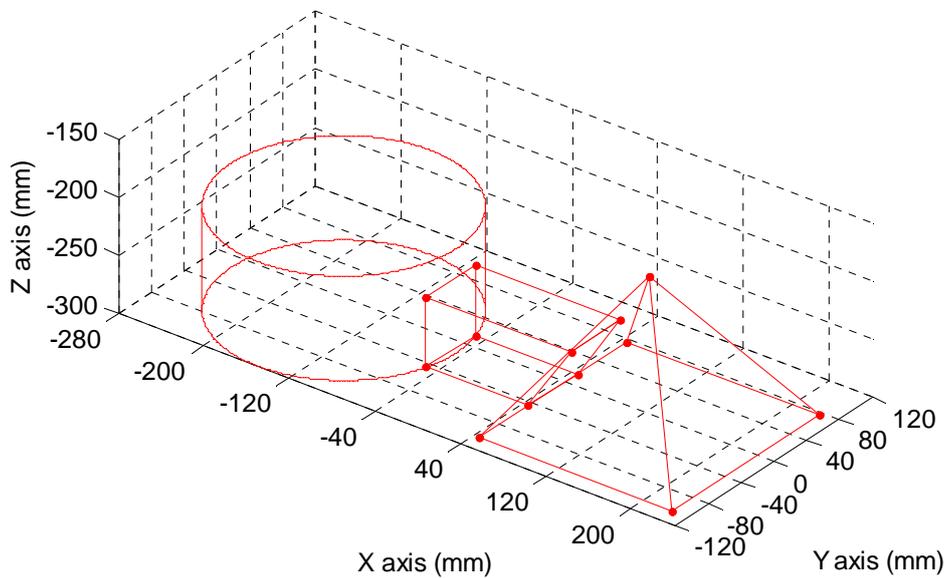


Fig. 6.24 Desired printing object of solid polyhedron

In Fig. 6.26(a), there are 4 sub-paths, \overline{AB} , \overline{CD} , \overline{EF} and \overline{GH} , after using the modified-DFS. To calculate minimum distance of linking, the modified-GA is used and shown in Fig. 6.26(b). The linking directions are \overline{BC} , \overline{DE} and \overline{FH} . One direction of sub-paths, \overline{GH} , is reversed. Finally, the path planning of 1st layer is accomplished.

In Fig. 6.27(a), continuing to use the last point of 1st-layer path as a start, however, it does not exist on 2nd layer and chooses the nearest point I as a start. There are 4 sub-paths, \overline{IJ} , \overline{KL} , \overline{MN} and \overline{OP} , after using modified-DFS. To calculate minimum distance of linking, modified-GA is used and shown in Fig. 6.27(b). One direction of sub-paths, \overline{IJ} , is reversed and breaks into a point I and a sub-path \overline{QJ} . The linking directions are \overline{IO} , \overline{PQ} , \overline{JK} and \overline{LM} . After that, the path of 2nd layer is finished.

In Fig. 6.28(a), continuing to use the last point of 2nd-layer path as a start, there are 3 sub-paths, \overline{RS} , \overline{TU} and \overline{VW} , after using modified-DFS. To calculate minimum distance of linking, modified-GA is used and shown in Fig. 6.28(b). The linking direction are \overline{ST} and \overline{UV} . At the end, the path planning of 3rd layer is accomplished and also the last layer.

The corresponding joint trajectories calculated from the inverse kinematics are shown in Fig. 6.29, where Fig. 6.29(a), (b) and (c) represent the output joint trajectories for axes A, B and C, respectively.

Then, the end-effector trajectory based on the calculated actuated joint trajectories is computed from the forward kinematics and compared to the designed end-effector trajectory. The verification results are shown in Fig. 6.30 and Fig. 6.31. The numerical calculation error between the planned and the calculated end-effector trajectories are presented in Fig. 6.32. The calculation errors on each axis are below $\pm 2.0 \times 10^{-5}$ mm. The details on each layer are shown in Fig. 6.33, Fig. 6.34 and Fig. 6.35.

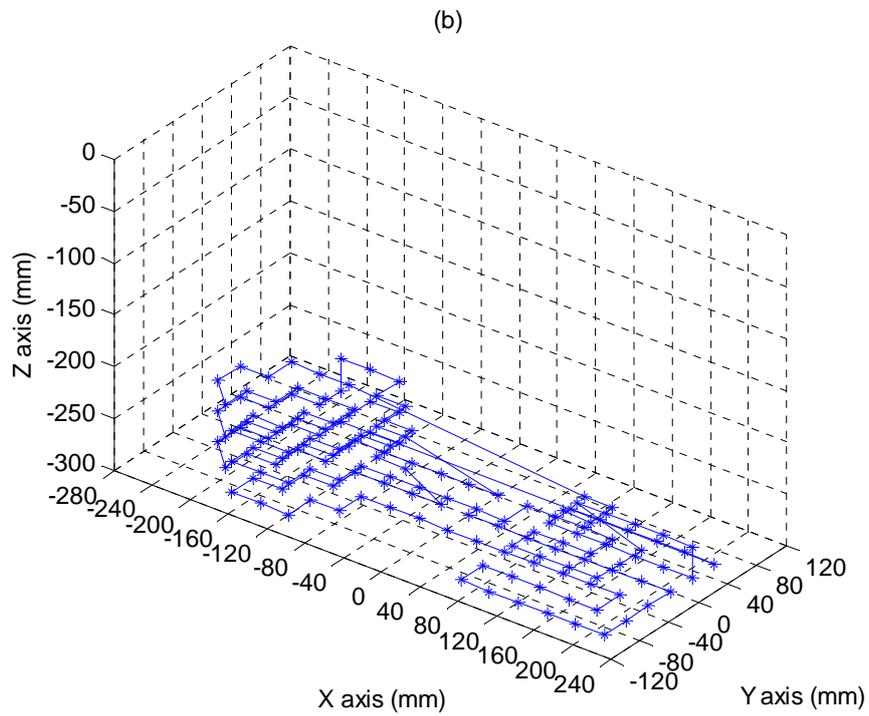
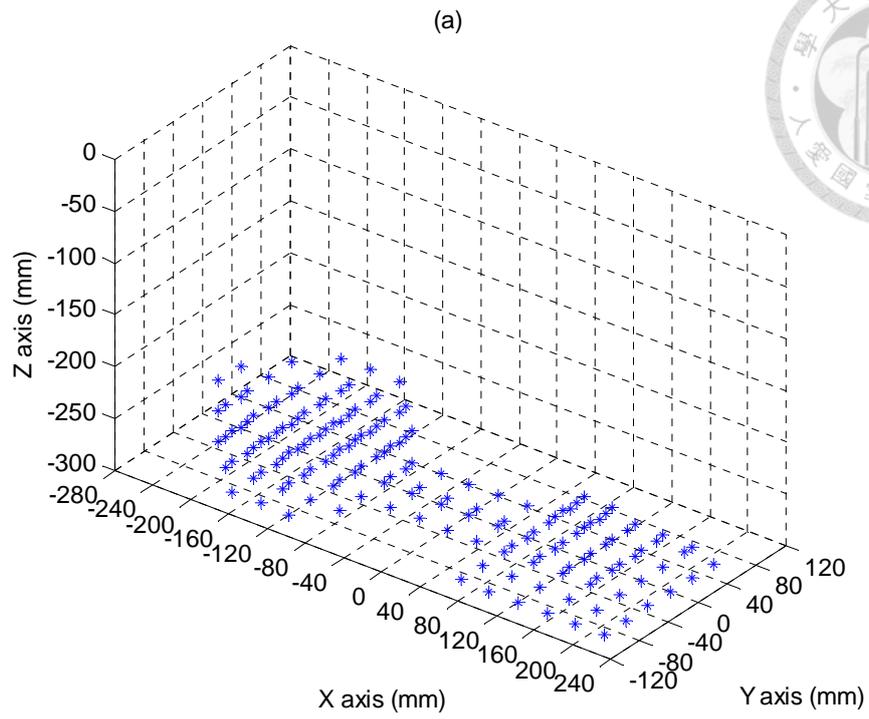


Fig. 6.25 Desired printing object model of solid polyhedron:

(a) all points (b) after path planning

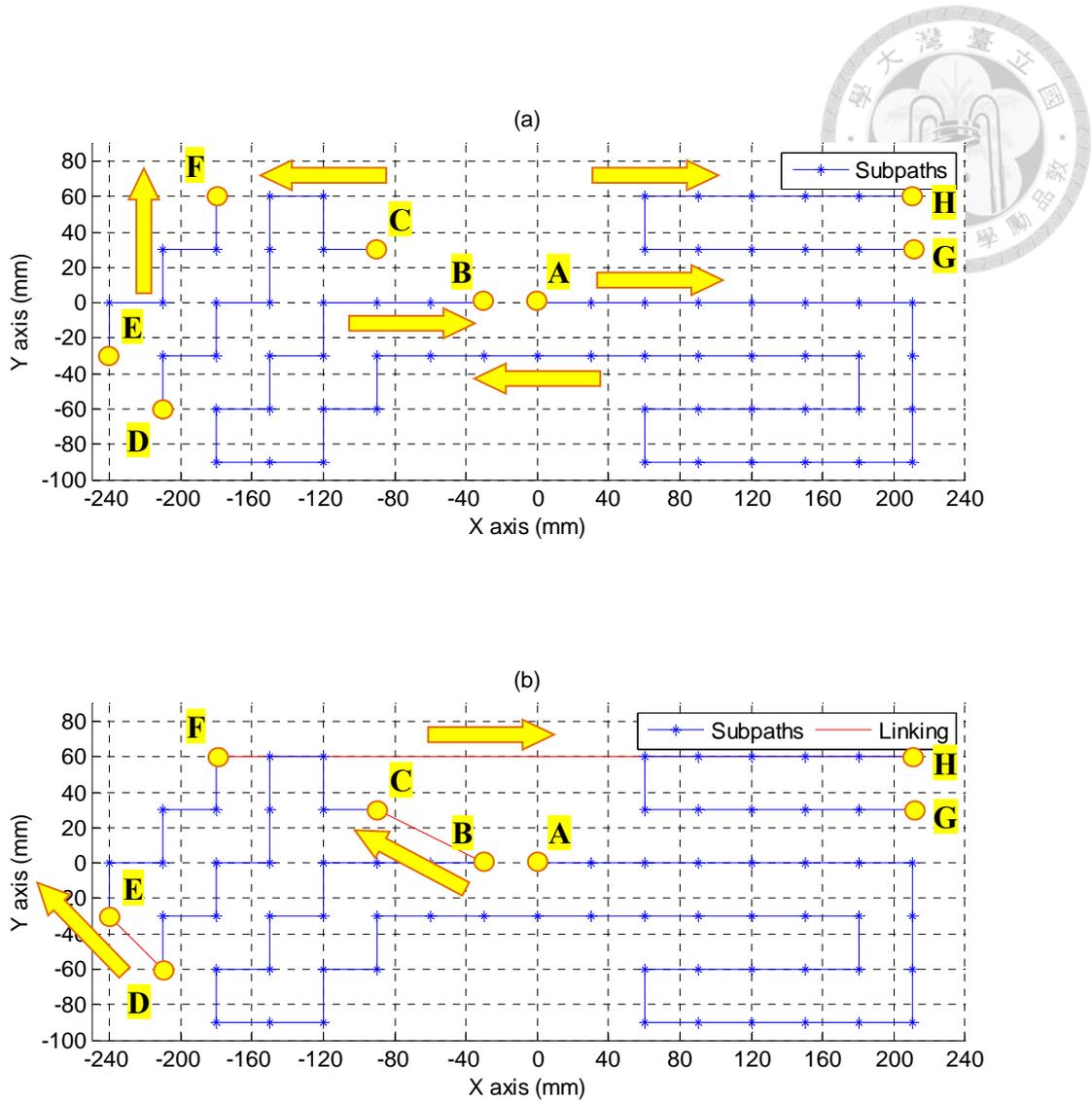


Fig. 6.26 Path planning for solid polyhedron trajectory on the 1st layer ($z = -300$ mm):

(a) after modified-DFS (b) after modified-GA

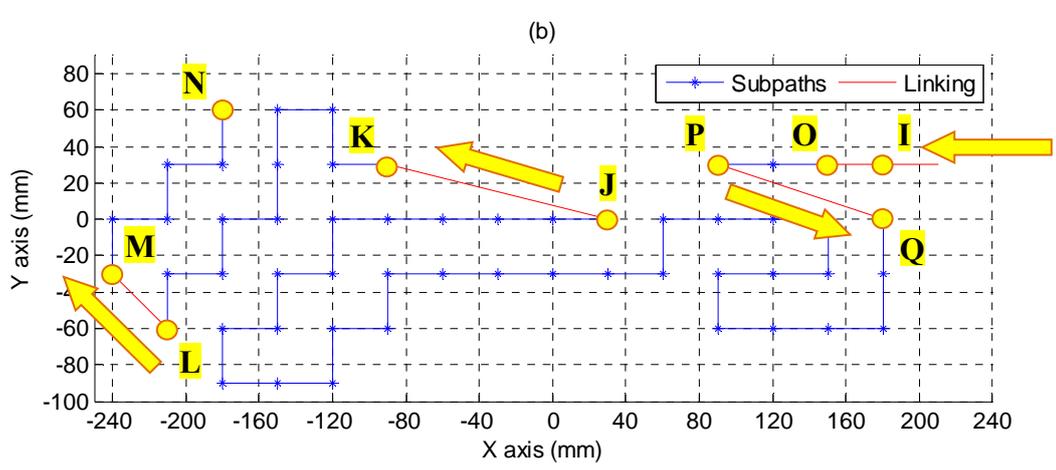
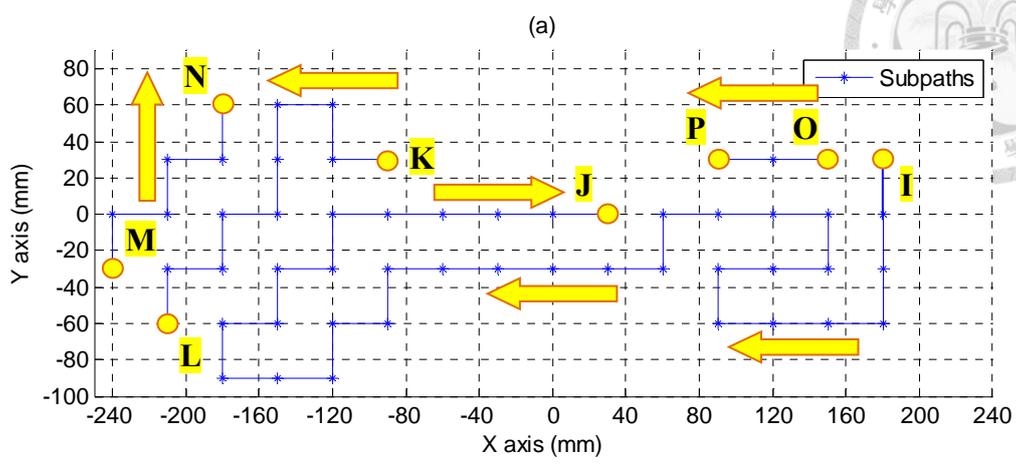
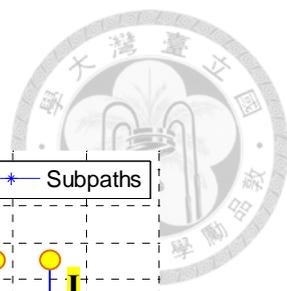


Fig. 6.27 Path planning for solid polyhedron trajectory on the 2nd layer ($z = -270$ mm):

(a) after modified-DFS (b) after modified-GA

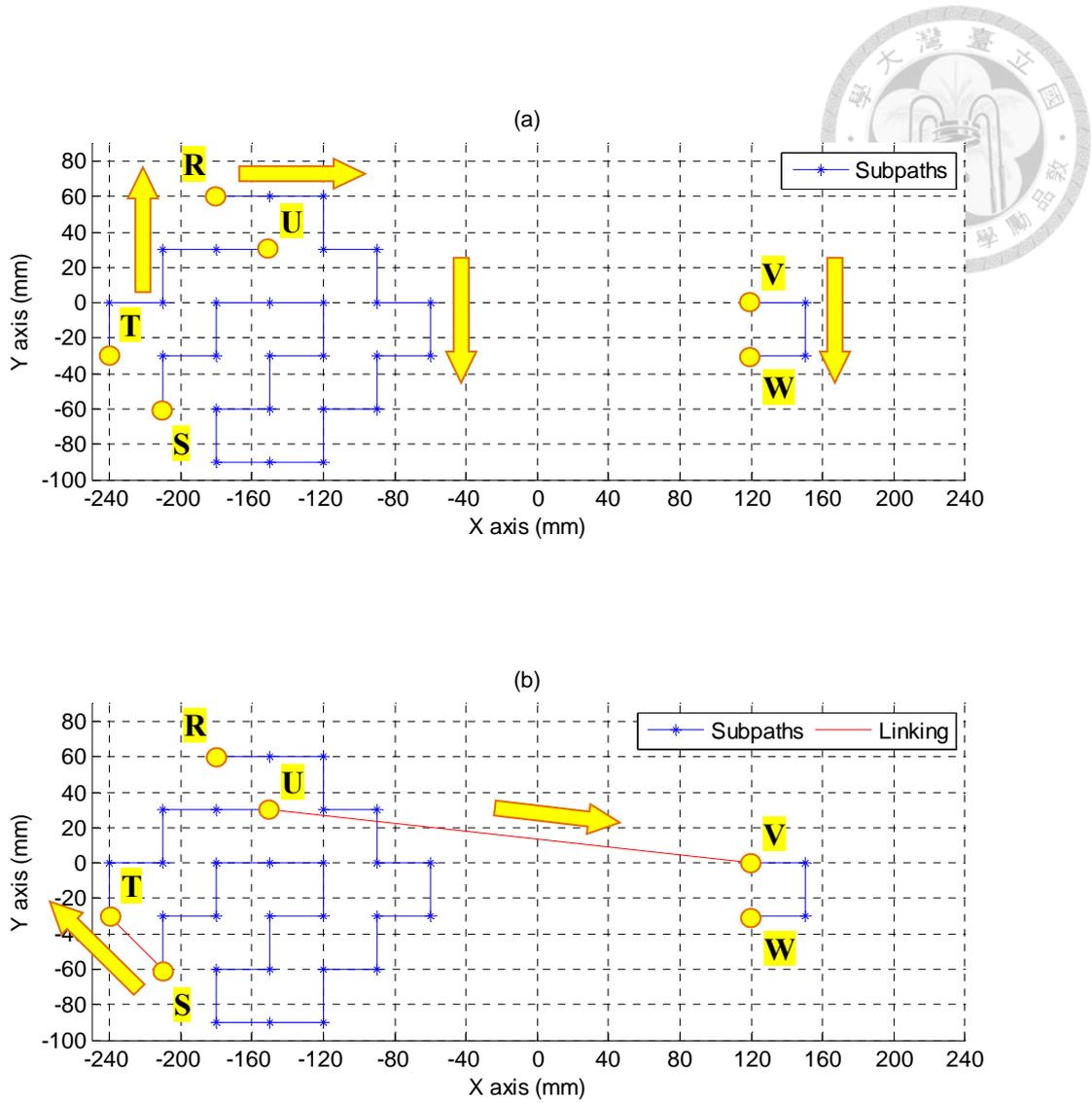


Fig. 6.28 Path planning for solid polyhedron trajectory on the 3rd layer ($z = -240$ mm):

(a) after modified-DFS (b) after modified-GA

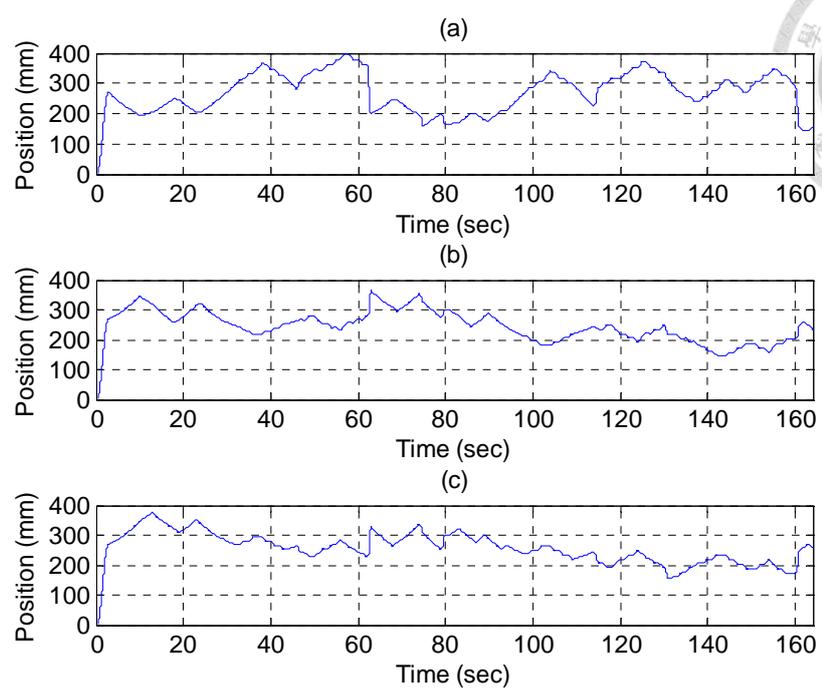
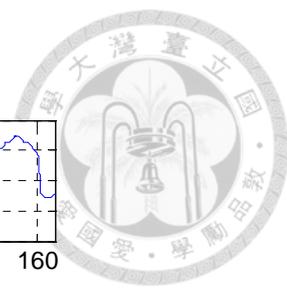


Fig. 6.29 Calculated joint space trajectories via inverse kinematics for the solid polyhedron trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator

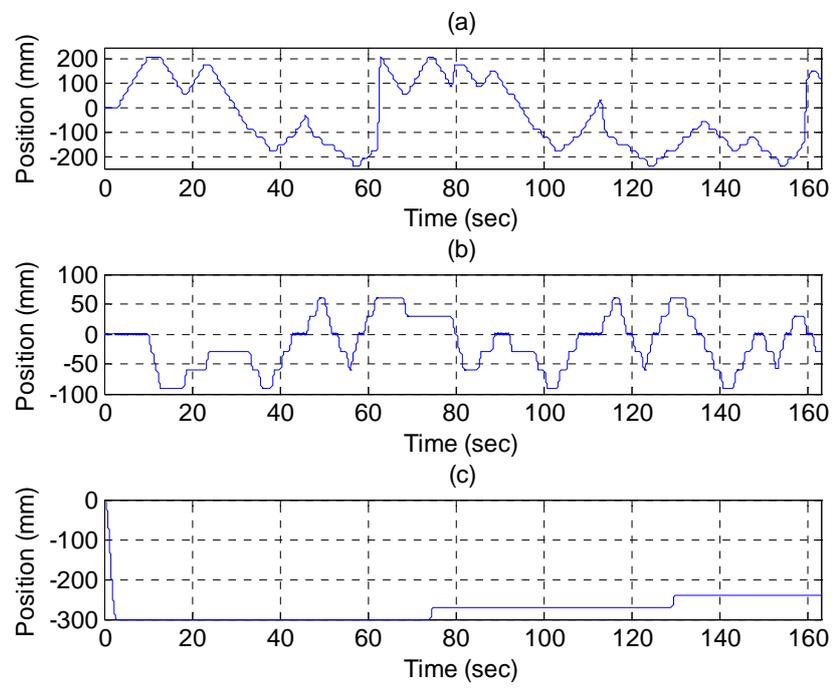


Fig. 6.30 Calculated task space trajectories via forward kinematics for the solid polyhedron trajectory: (a) X-axis (b) Y-axis (c) Z-axis

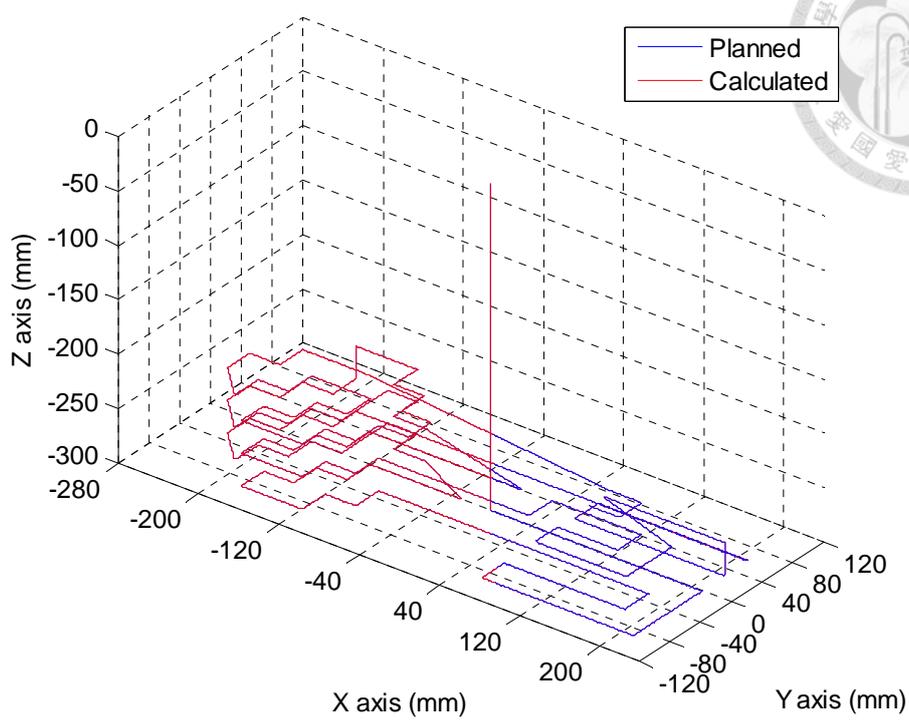


Fig. 6.31 Comparison between the planned and the calculated end-effector for the solid polyhedron trajectory.

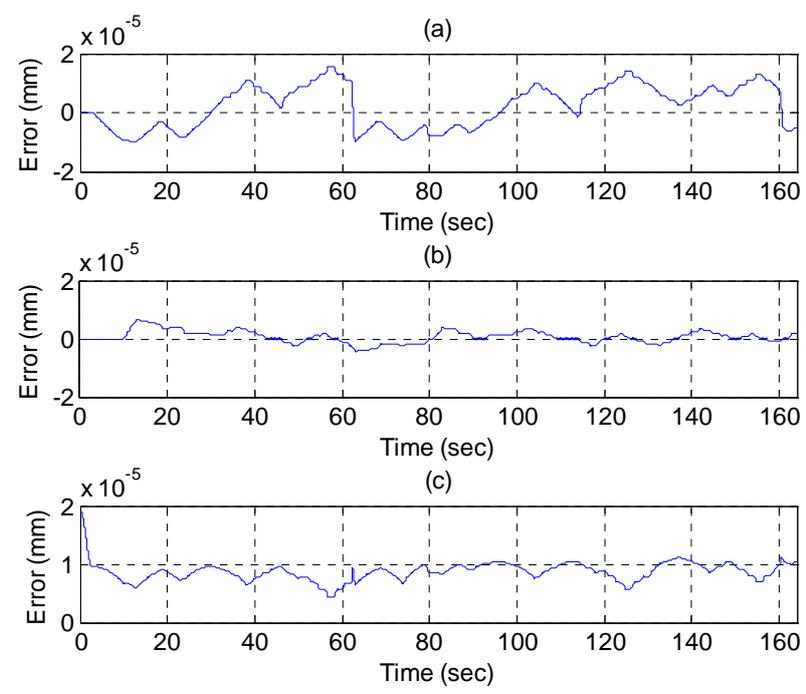


Fig. 6.32 Numerical error between the planned and the calculated end-effector for the solid polyhedron trajectory: (a) X-axis (b) Y-axis and (C) Z-axis

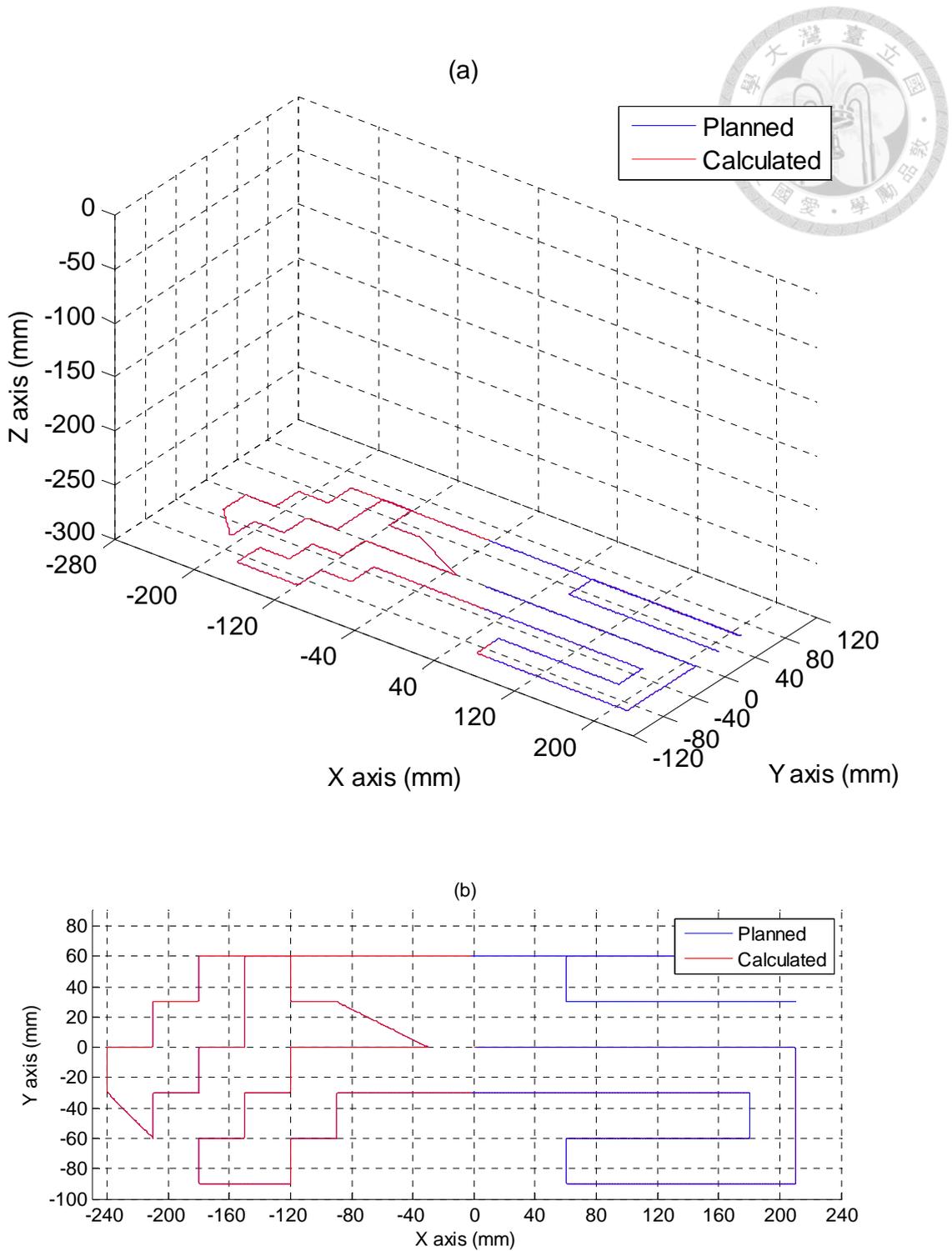


Fig. 6.33 Comparison between the planned and the calculated end-effector for the solid polyhedron trajectory of the 1st layer ($z = -300$ mm): (a) front view (b) top view

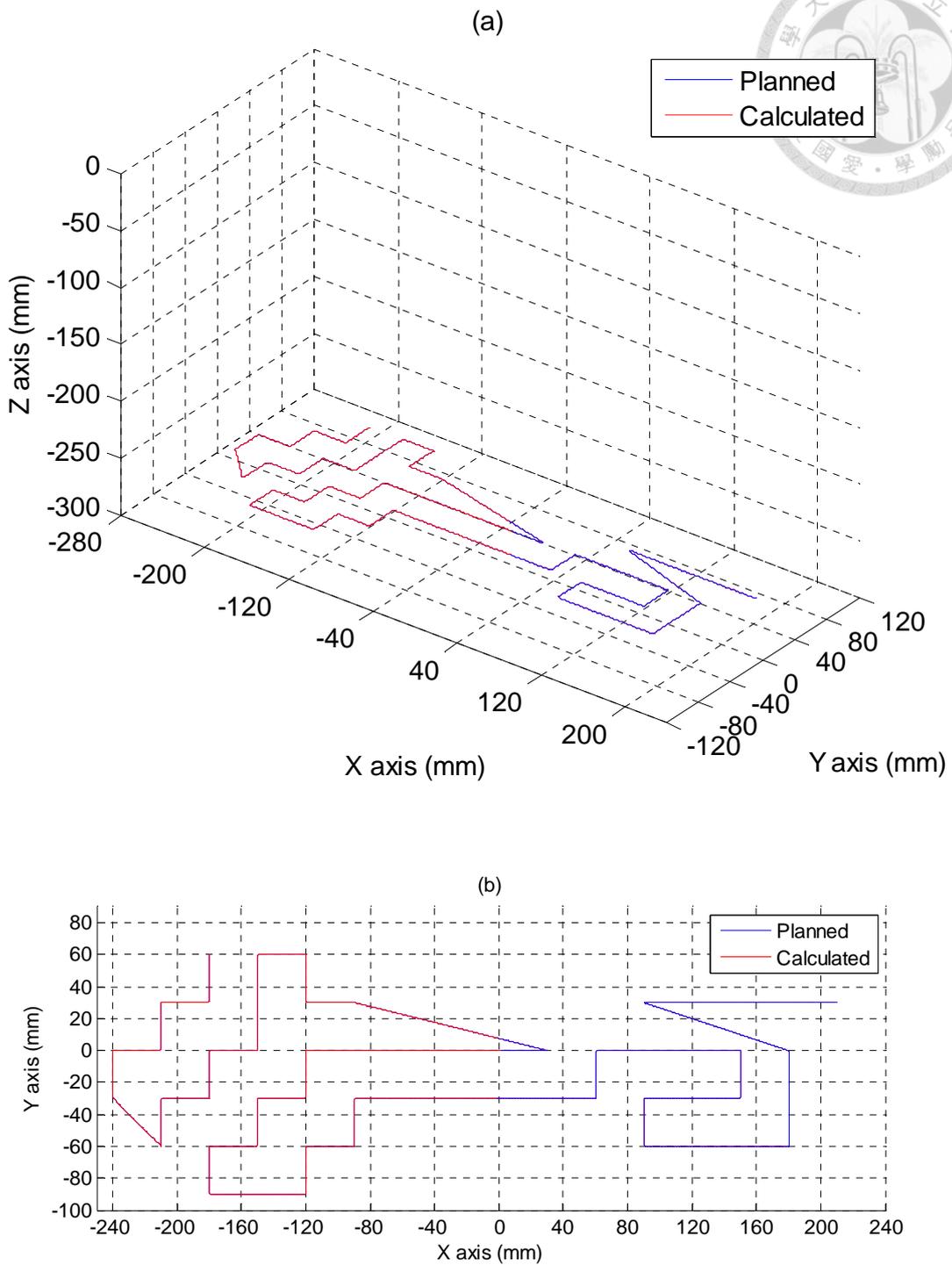


Fig. 6.34 Comparison between the planned and the calculated end-effector for the solid polyhedron trajectory of the 2nd layer ($z = -270$ mm): (a) front view (b) top view

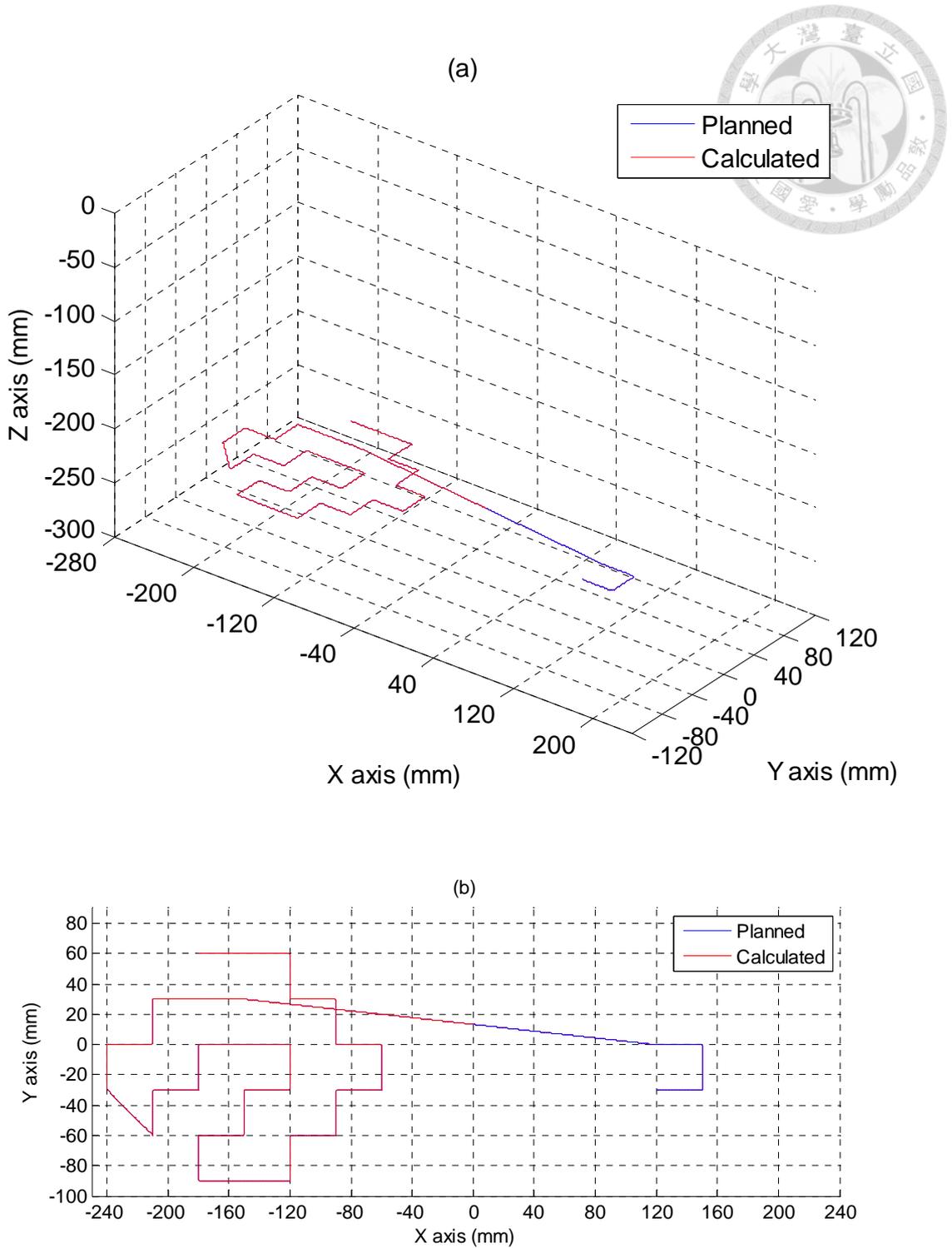


Fig. 6.35 Comparison between the planned and the calculated end-effector for the solid polyhedron trajectory of the 3rd layer ($z = -240$ mm): (a) front view (b) top view

6.1.5 Solid Complex 3D Trajectory

The desired printing object of solid complex 3D is shown in Fig. 6.36, and the joint and end-effector trajectories are shown in Fig. 6.37 and Fig. 6.38. The comparison between the calculated and the designed are shown in Fig. 6.39. The numerical calculation errors are presented in Fig. 6.40 and all are below $\pm 2.0 \times 10^{-5}$ mm.

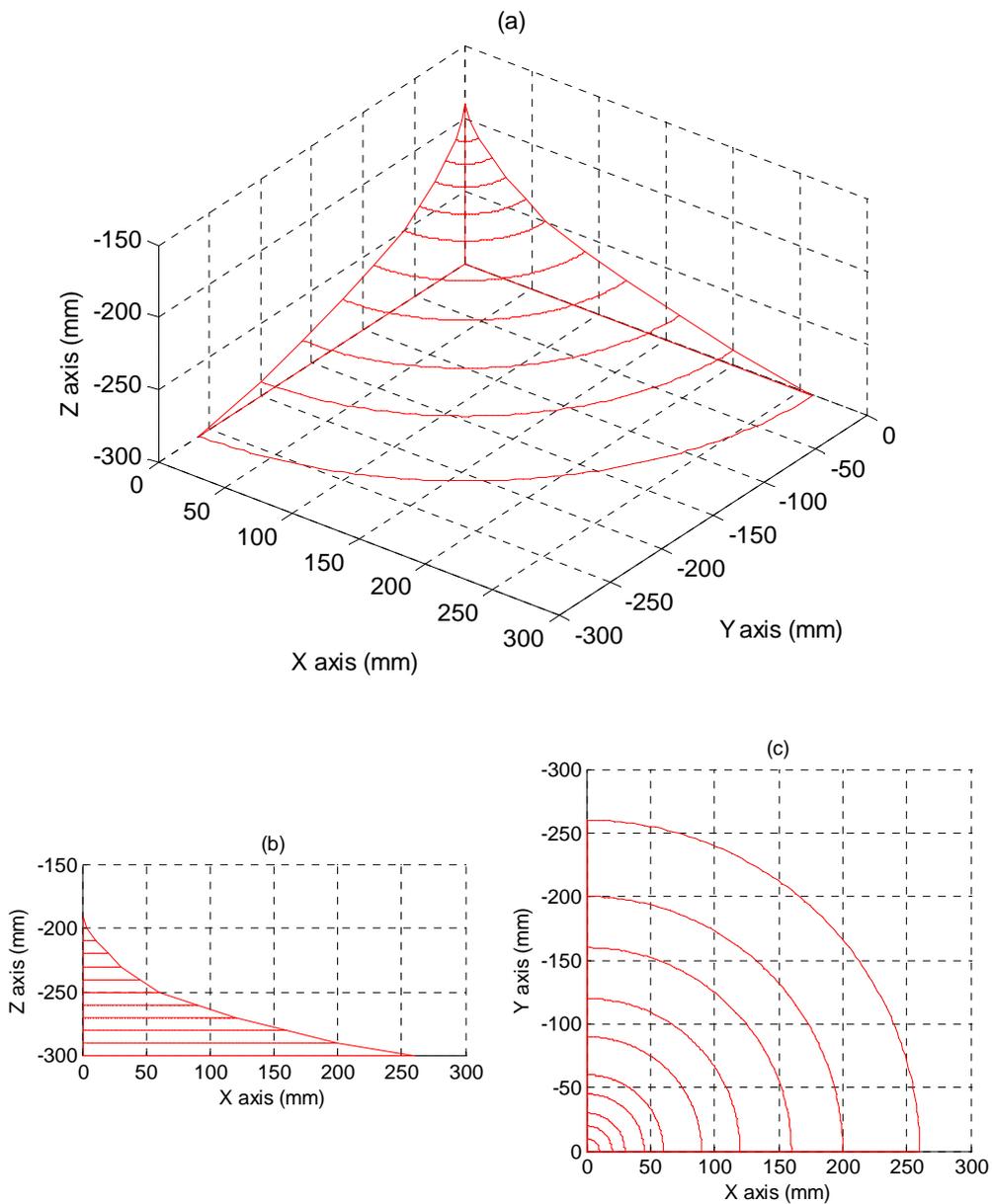


Fig. 6.36 Desired printing object of solid complex 3D:

(a) front view (b) side view (c) top view

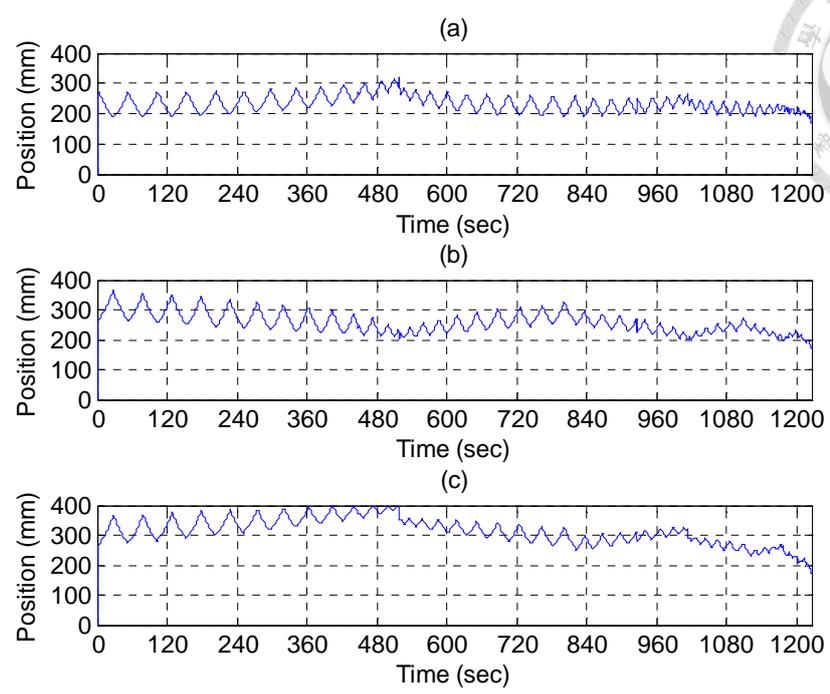
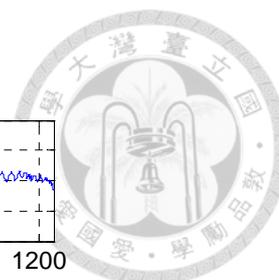


Fig. 6.37 Calculated joint space trajectories via inverse kinematics for the solid complex 3D trajectory: (a) A-axis actuator (b) B-axis actuator (c) C-axis actuator

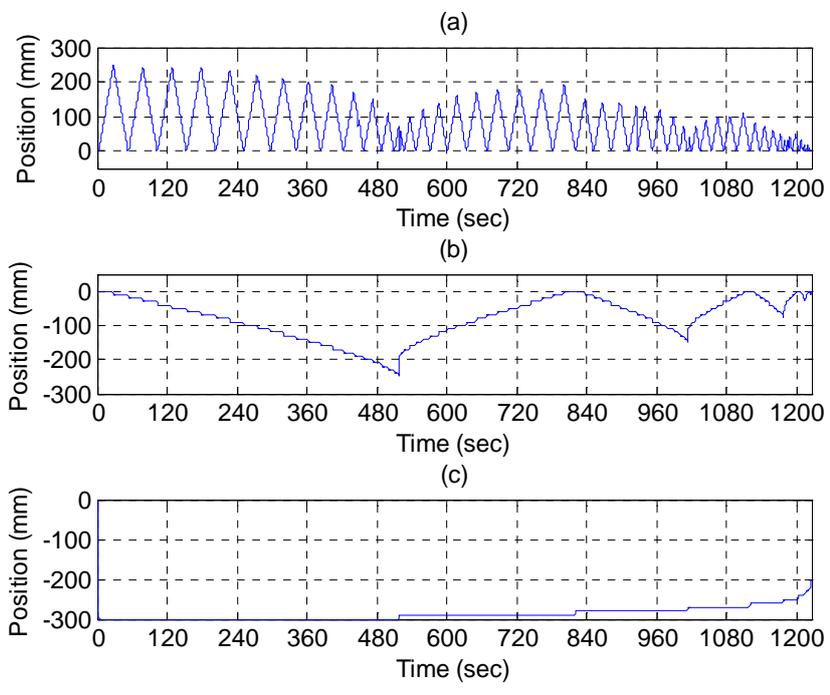


Fig. 6.38 Calculated task space trajectories via forward kinematics for the solid complex 3D trajectory: (a) X-axis (b) Y-axis (c) Z-axis

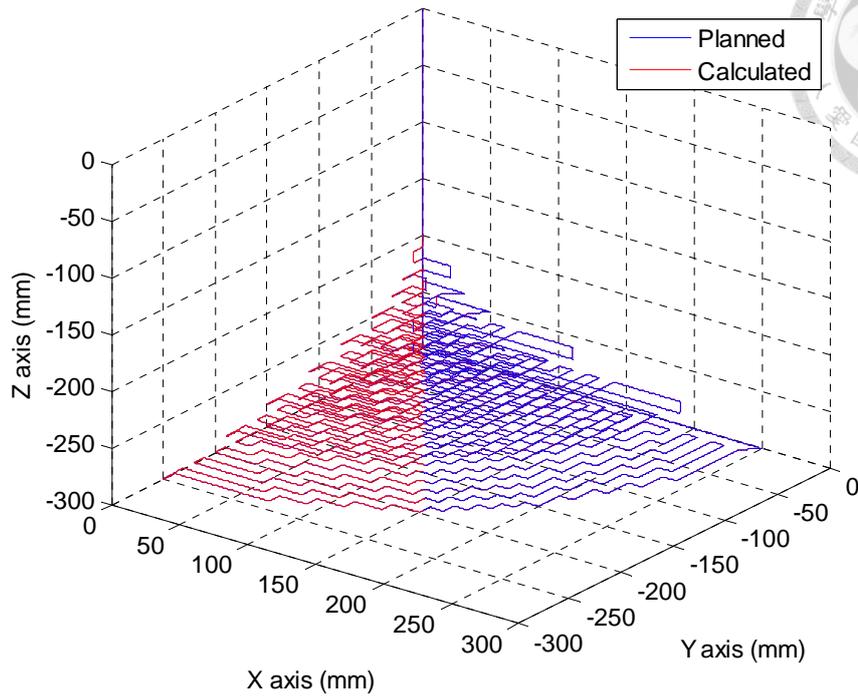
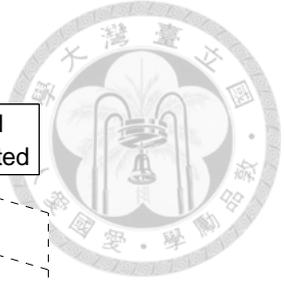


Fig. 6.39 Comparison between the planned and the calculated end-effector for the solid complex 3D trajectory.

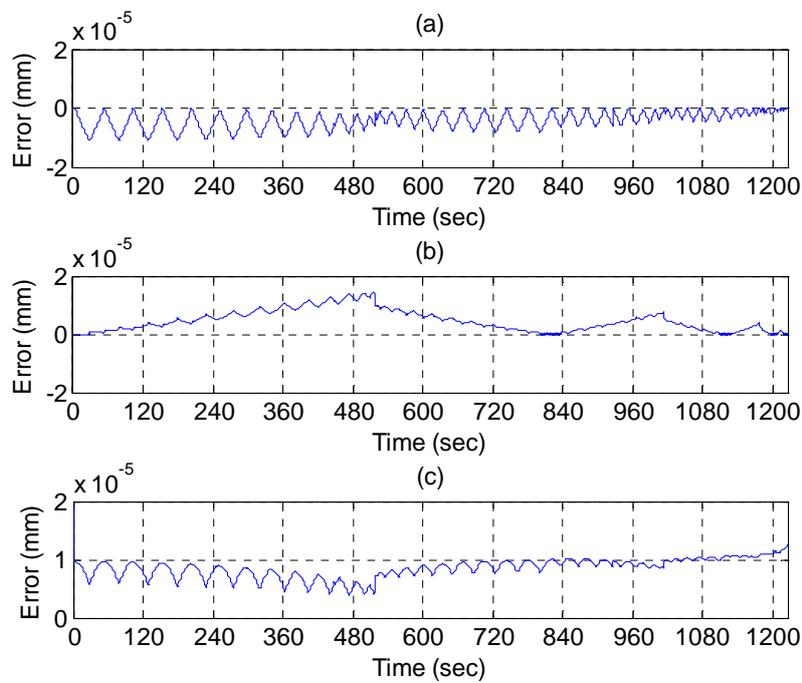
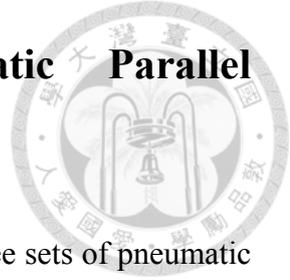


Fig. 6.40 Numerical error between the planned and the calculated end-effector for the solid complex 3D trajectory: (a) X-axis (b) Y-axis and (c) Z-axis

6.2 Simulations of Three-Axial Pneumatic Parallel Manipulator by ADAMS and SIMULINK



The manipulator simulations include a ADAMS model and three sets of pneumatic system and controller. In Fig. 6.41, the designed trajectory of end-effector E_d will transfer to the actuator trajectories A, B and C axis respectively via the inverse kinematics. Each pressure difference of pneumatic cylinders will generated force corresponding to A, B and C axis as an input of ADAMS model and the position feedback will compensate the tracking error. The SIMULINK model is shown in Fig. 6.42.

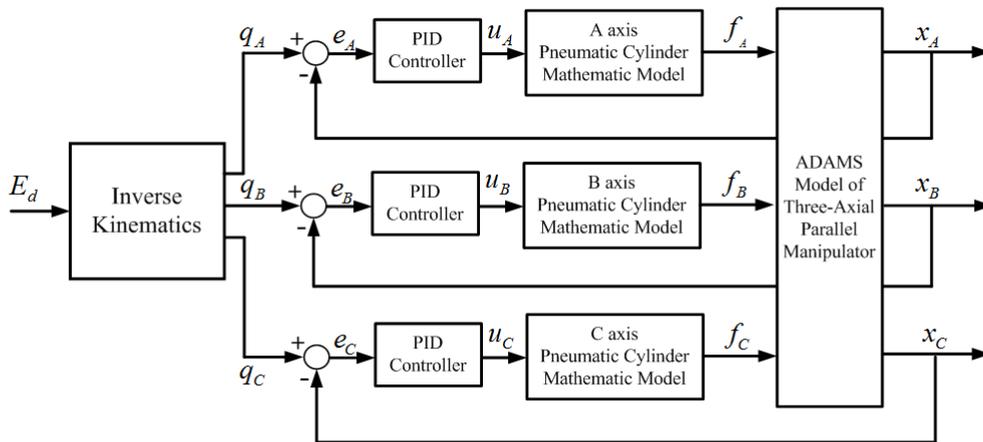


Fig. 6.41 Schematic diagram of the three-axial closed-loop path tracking control

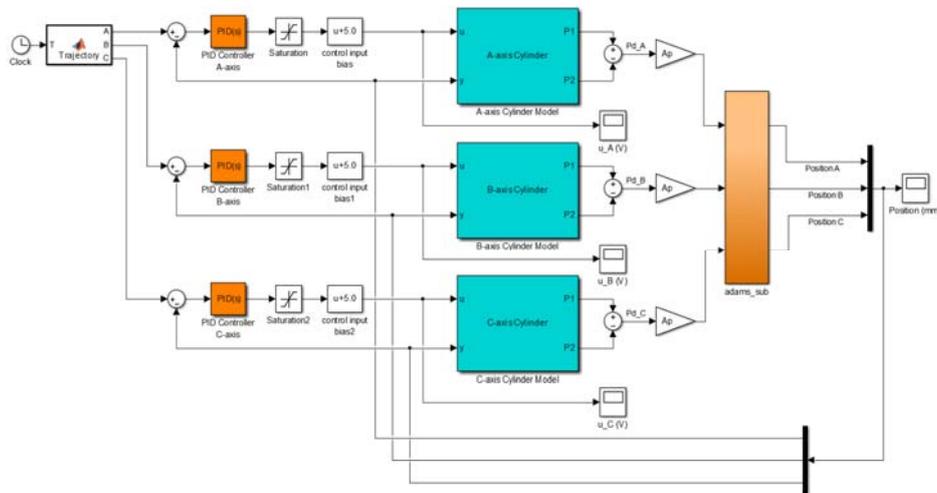
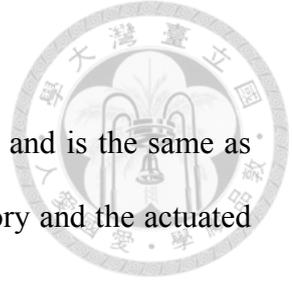


Fig. 6.42 SIMULINK model of the three-axial closed-loop system



6.2.1 Simulation of Circle-shape Line Trajectory

The designed circle-shape line trajectory is shown in Fig. 6.43 and is the same as that in Section 6.1.1. The details of the end-effector position trajectory and the actuated position trajectory have been also discussed in Section 6.1.1.

The simulation results of path tracking control for A, B and C axis are shown in Fig. 6.44, Fig. 6.45, and Fig. 6.46. The simulation position errors of the manipulator actuator for A, B and C axis are below ± 2.0 mm. The control signals oscillate in the low velocity region and motion direction change of the cylinder. In addition, the estimated end-effector positions are calculated by the forward kinematics from the actuated joint positions of three cylinders and shown in Fig. 6.47. The maximum estimated error of the end-effector is about 0.6 mm at 1.5 sec.

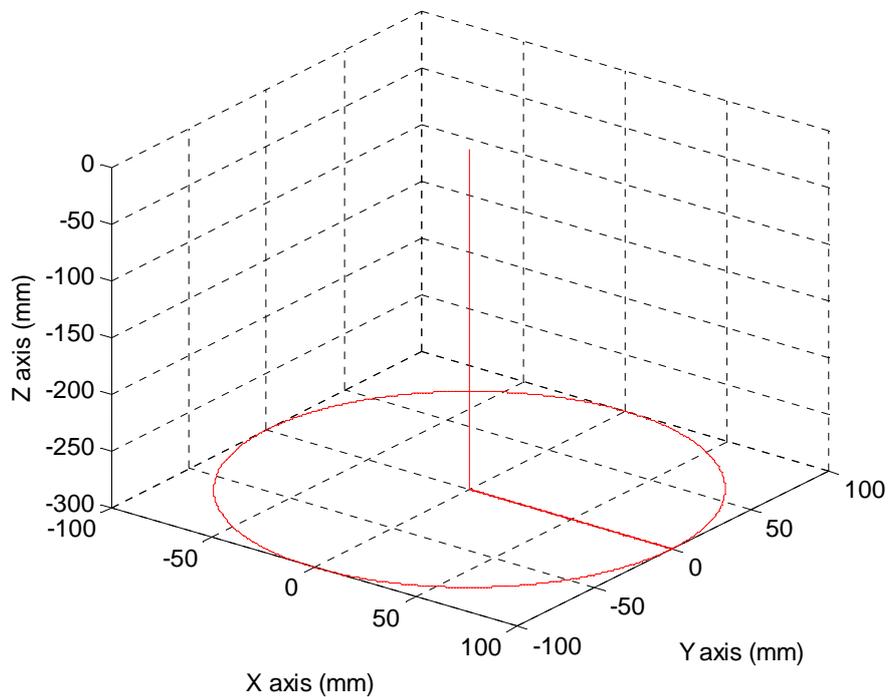


Fig. 6.43 Designed circle-shape line trajectory for end-effector

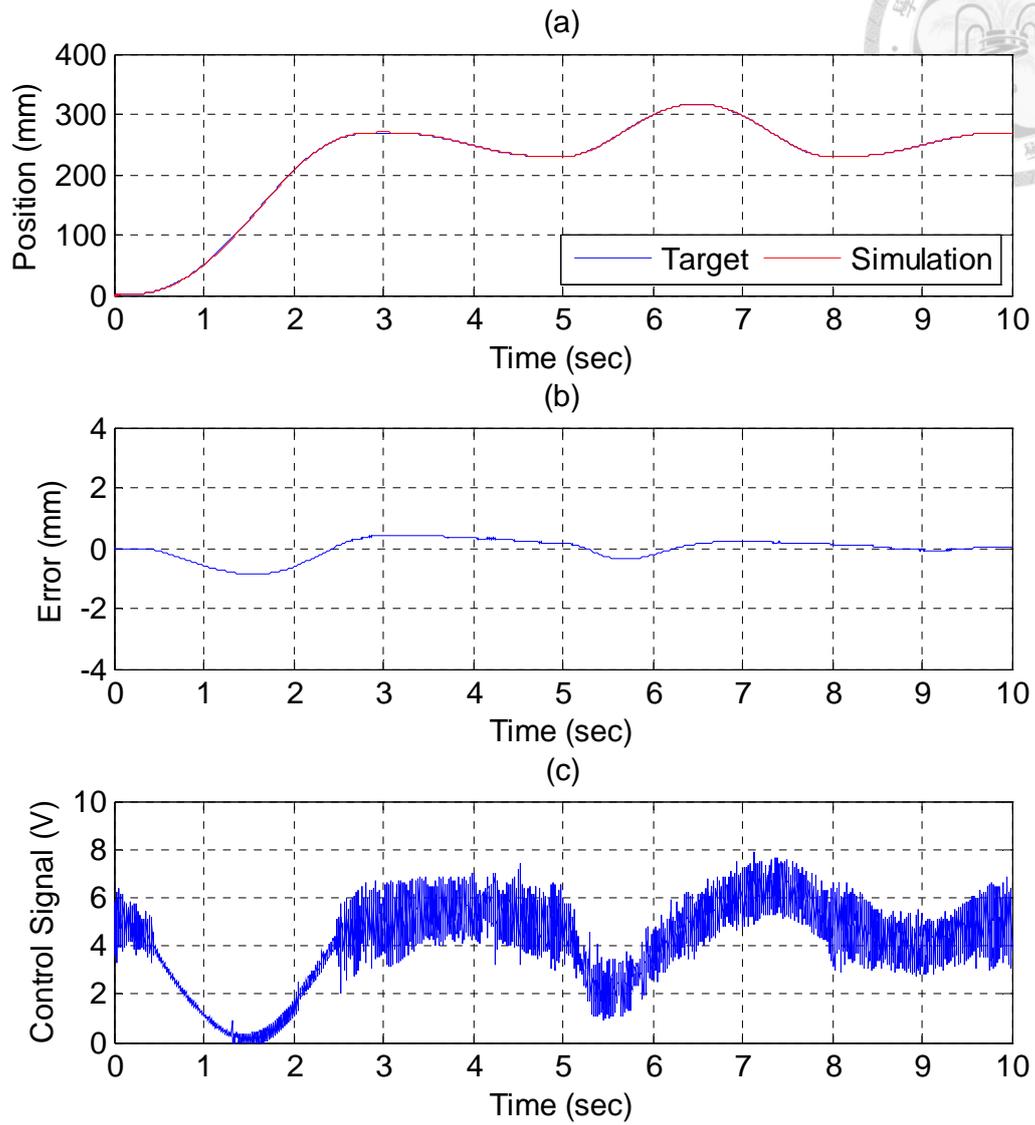


Fig. 6.44 Simulation results of end-effector path tracking control for

A-axis cylinder via a circle-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

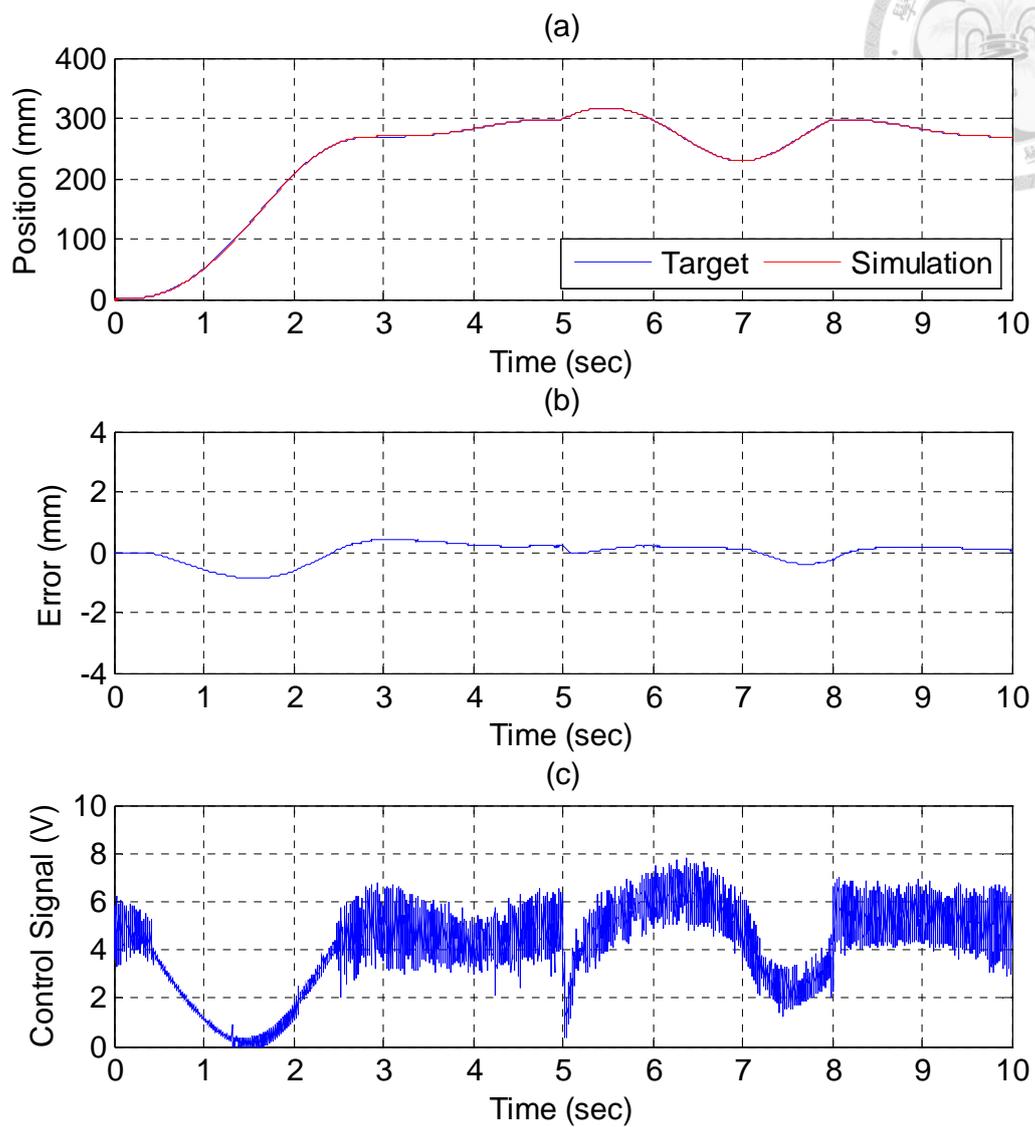


Fig. 6.45 Simulation results of end-effector path tracking control for

B-axis cylinder via a circle-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

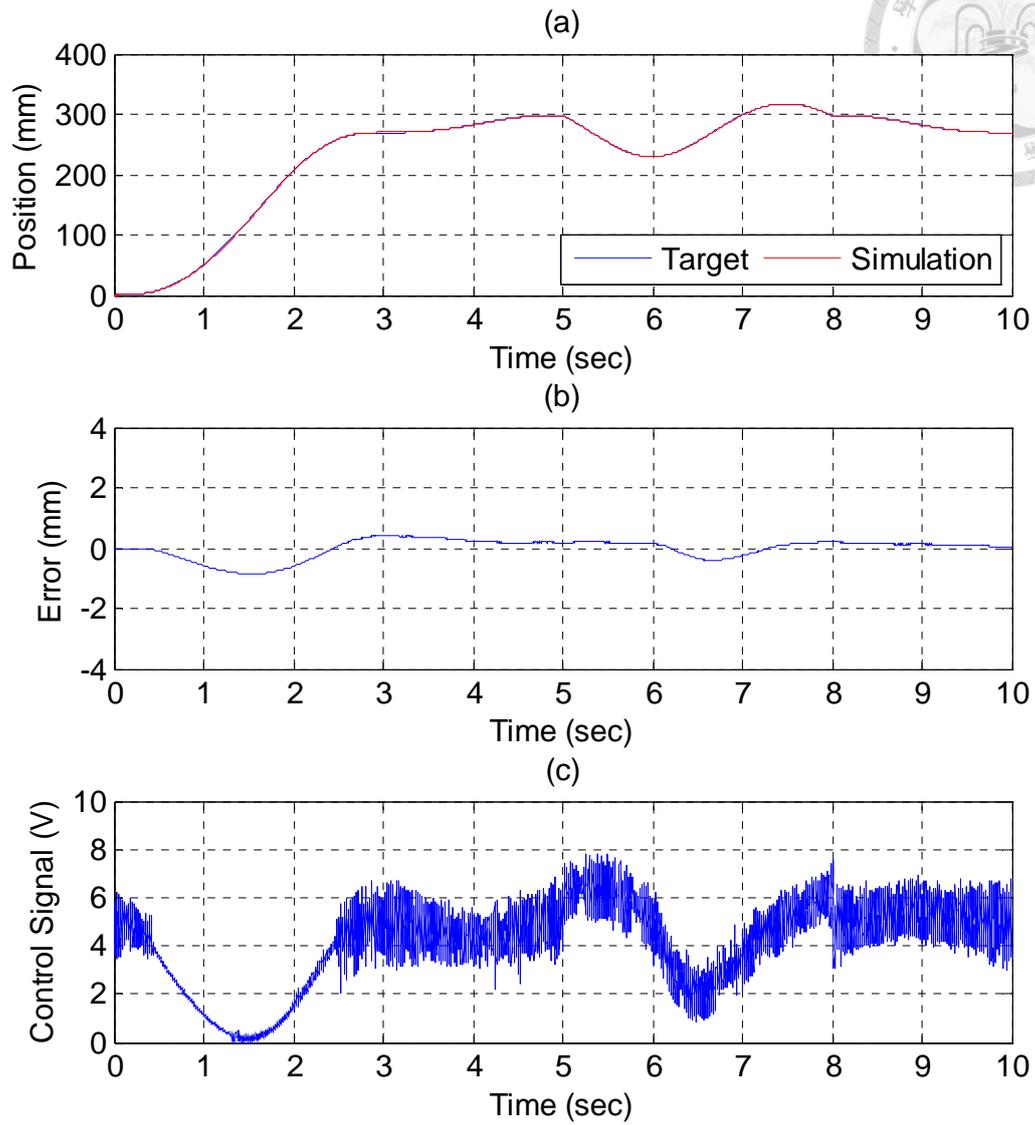


Fig. 6.46 Simulation results of end-effector path tracking control for

C-axis cylinder via a circle-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

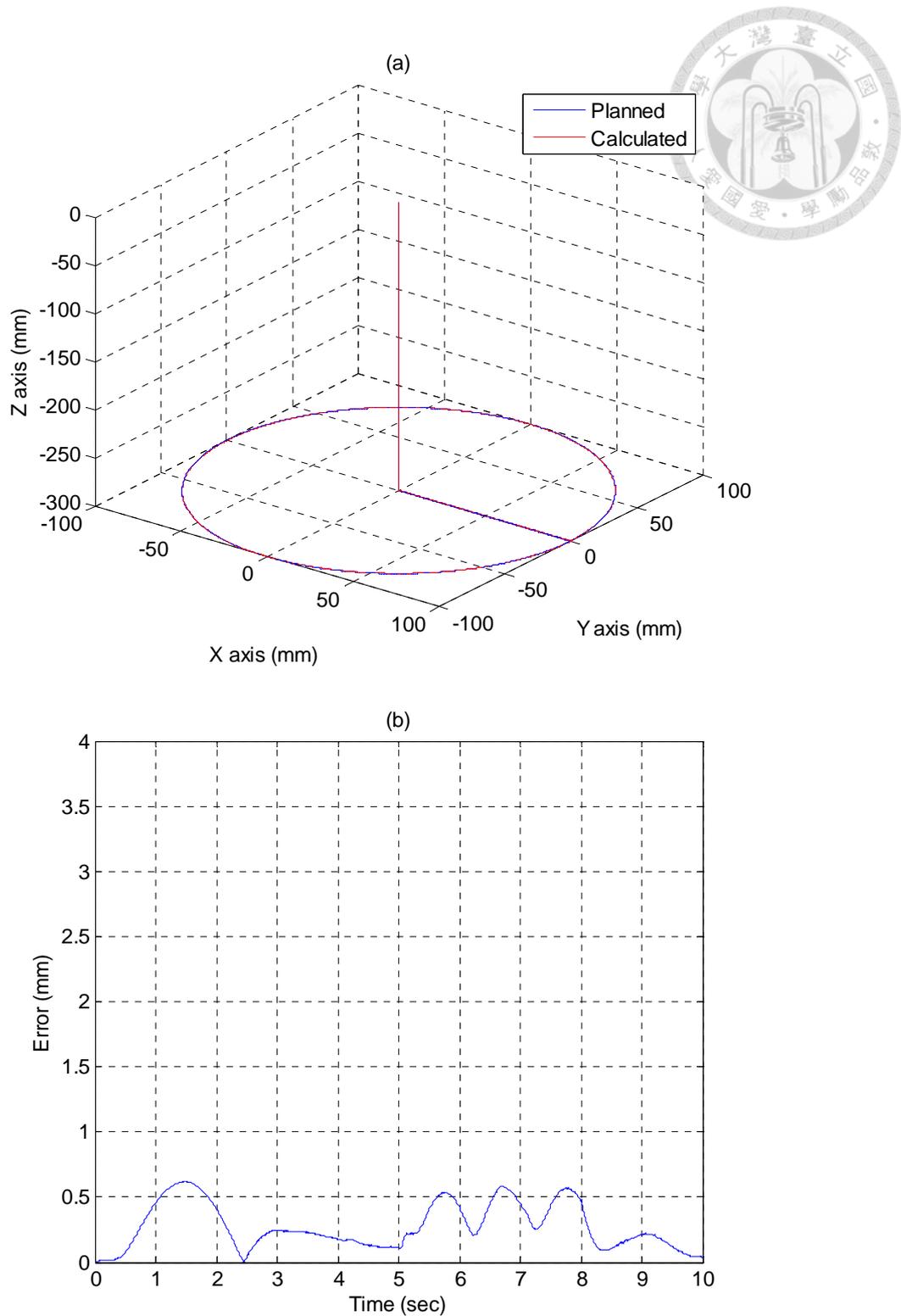
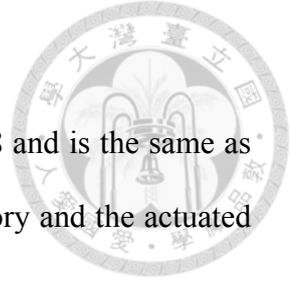


Fig. 6.47 Estimated end-effector position tracking response for a circle-shape line trajectory: (a) calculated end-effector position
(b) calculated end-effector position error

6.2.2 Simulation of Sphere-shape Line Trajectory



The designed sphere-shape line trajectory is shown in Fig. 6.48 and is the same as that in Section 6.1.2. The details of the end-effector position trajectory and the actuated position trajectory have been also discussed in Section 6.1.2.

The simulation results of path tracking control for A, B and C axis are shown in Fig. 6.49, Fig. 6.50, and Fig. 6.51. The simulation position errors of the manipulator actuator for A, B and C axis are below ± 2.0 mm. The control signals oscillate in the low velocity region and motion direction change of the cylinder. In addition, the estimated end-effector positions are calculated by the forward kinematics from the actuated joint positions of three cylinders and shown in Fig. 6.52. The maximum estimated error of the end-effector is about 1.0 mm at 5.5 sec.

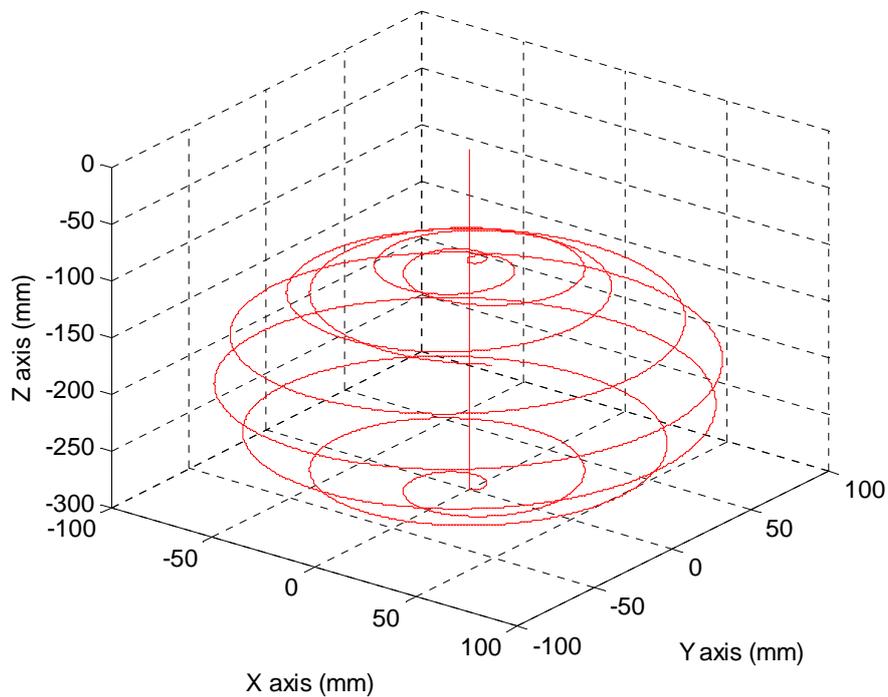


Fig. 6.48 Designed sphere-shape line trajectory for end-effector

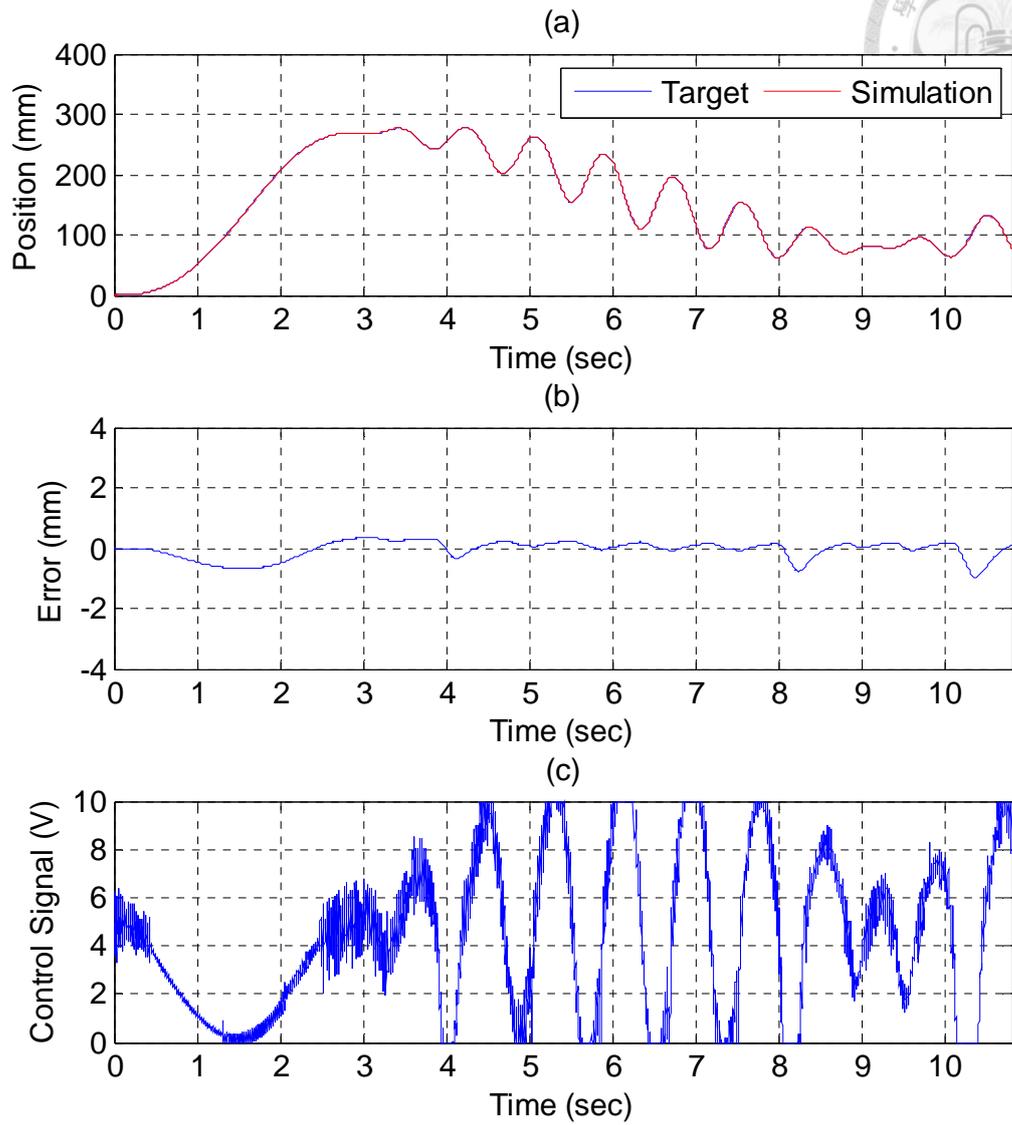


Fig. 6.49 Simulation results of end-effector path tracking control for

A-axis cylinder via a sphere-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

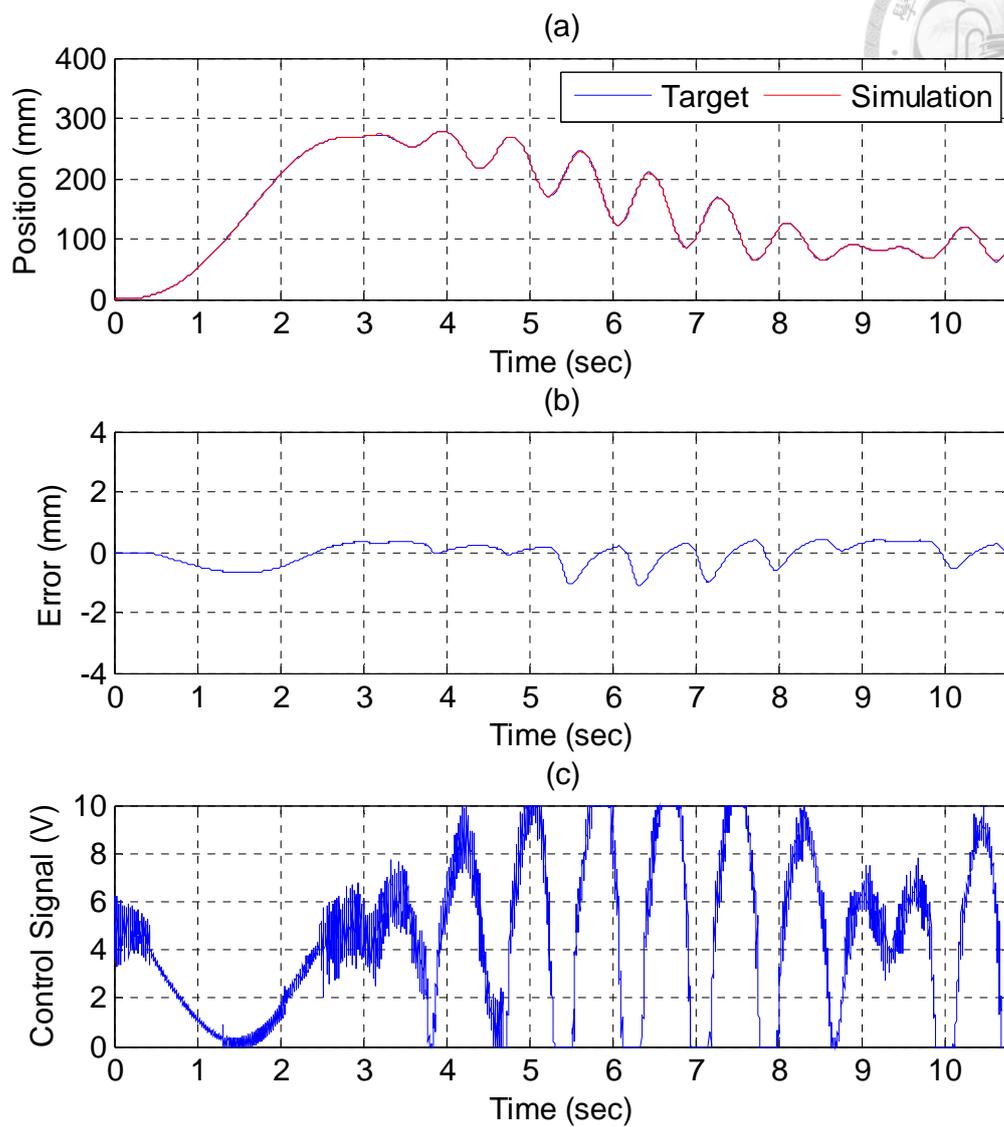


Fig. 6.50 Simulation results of end-effector path tracking control for

B-axis cylinder via a sphere-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

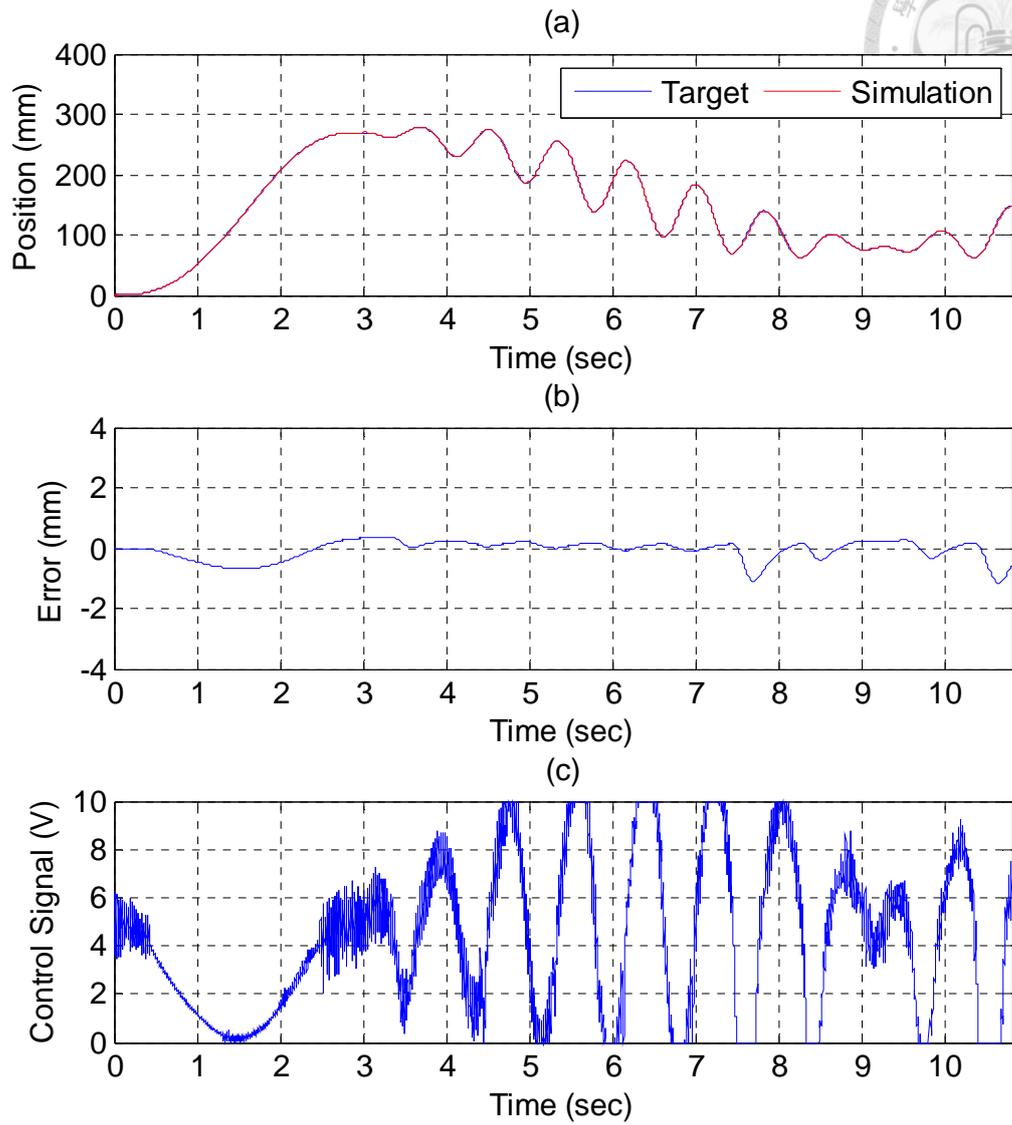


Fig. 6.51 Simulation results of end-effector path tracking control for

C-axis cylinder via a sphere-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

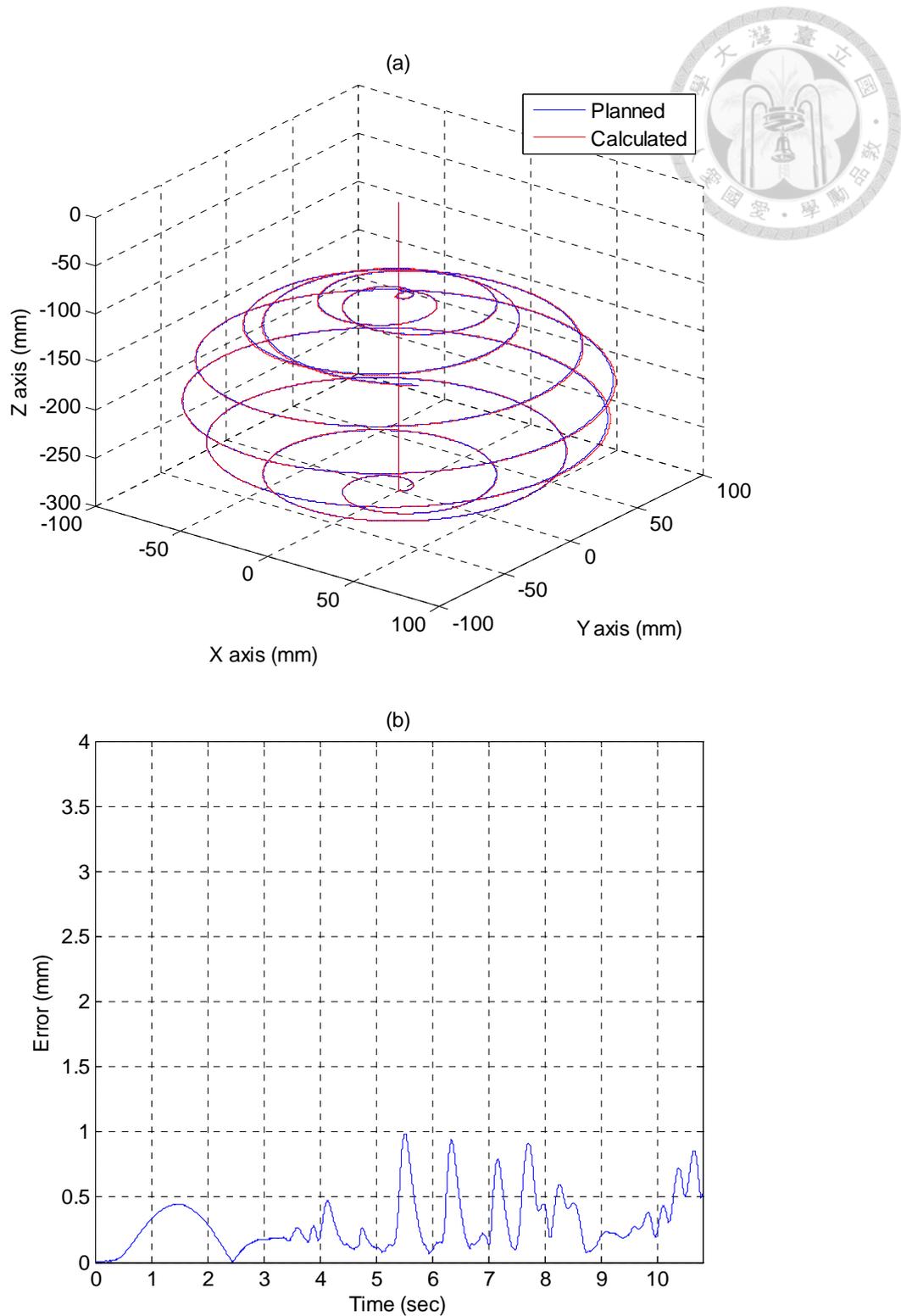
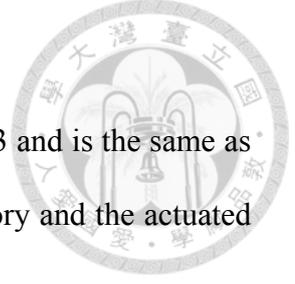


Fig. 6.52 Estimated end-effector position tracking response for a sphere-shape line trajectory: (a) calculated end-effector position
(b) calculated end-effector position error

6.2.3 Simulation of Solid Cuboid Trajectory



The desired printing object of solid cuboid is shown in Fig. 6.53 and is the same as that in Section 6.1.3. The details of the end-effector position trajectory and the actuated position trajectory have been also discussed in Section 6.1.3.

The simulation results of path tracking control for A, B and C axis are shown in Fig. 6.54, Fig. 6.55, and Fig. 6.56. The simulation position errors of the manipulator actuator for A, B and C axis are below ± 2.0 mm. The control signals oscillate in the low velocity region and motion direction change of the cylinder. In addition, the estimated end-effector positions are calculated by the forward kinematics from the actuated joint positions of three cylinders and shown in Fig. 6.57. The maximum estimated error of the end-effector is about 0.6 mm at 2.0 sec.

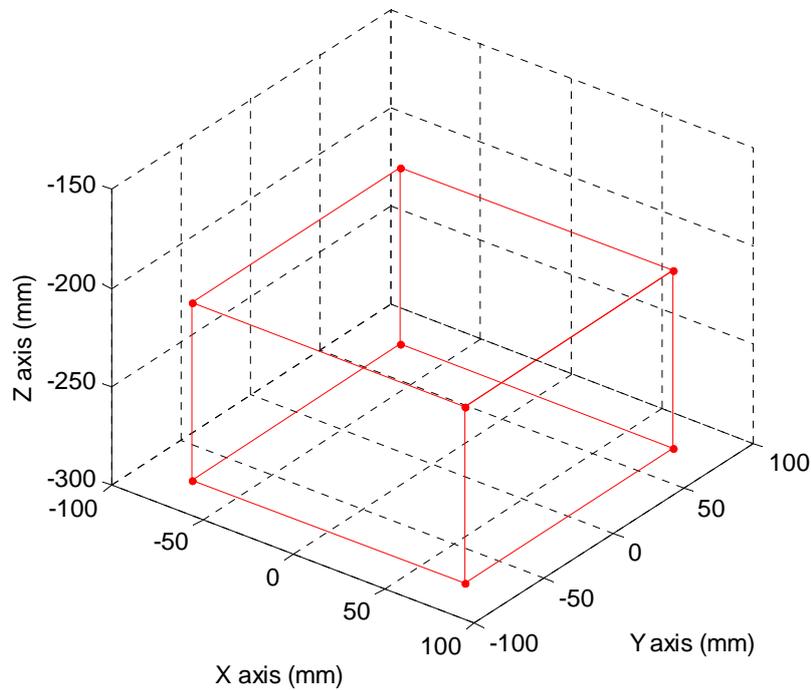


Fig. 6.53 Desired printing object of solid cuboid

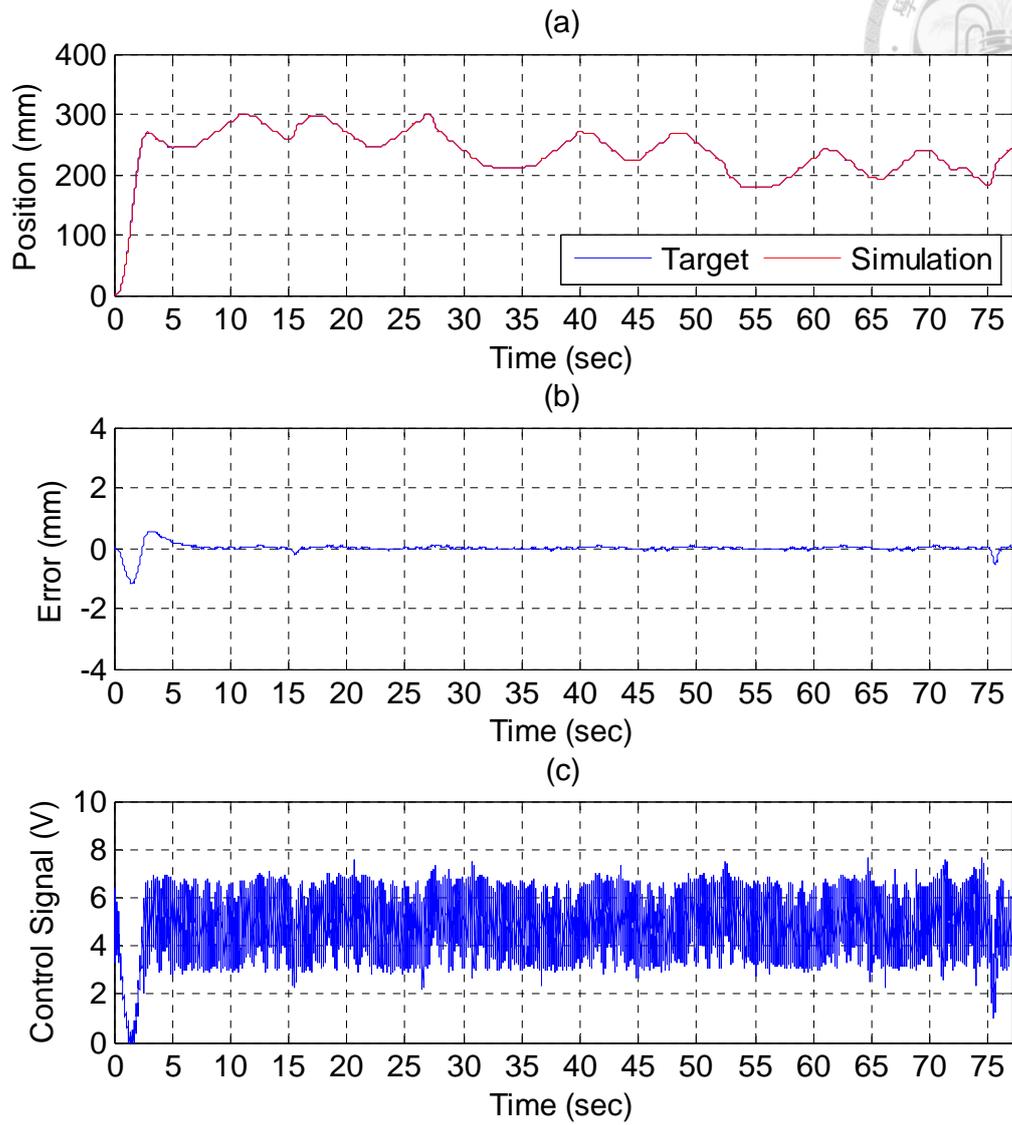


Fig. 6.54 Simulation results of end-effector path tracking control for

A-axis cylinder via a solid cuboid trajectory:

(a) position tracking response (b) position tracking error (c) control signal

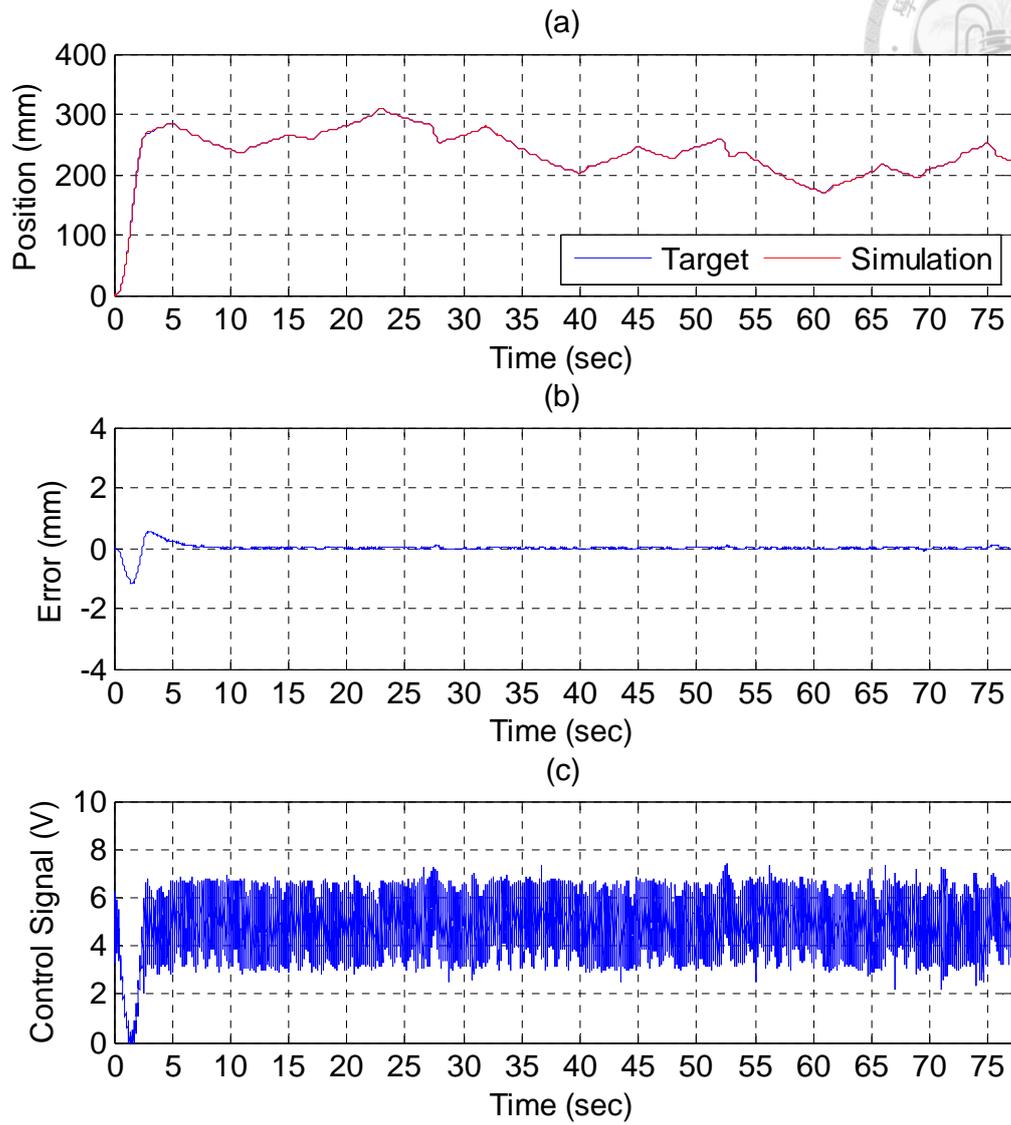


Fig. 6.55 Simulation results of end-effector path tracking control for

B-axis cylinder via a solid cuboid trajectory:

(a) position tracking response (b) position tracking error (c) control signal

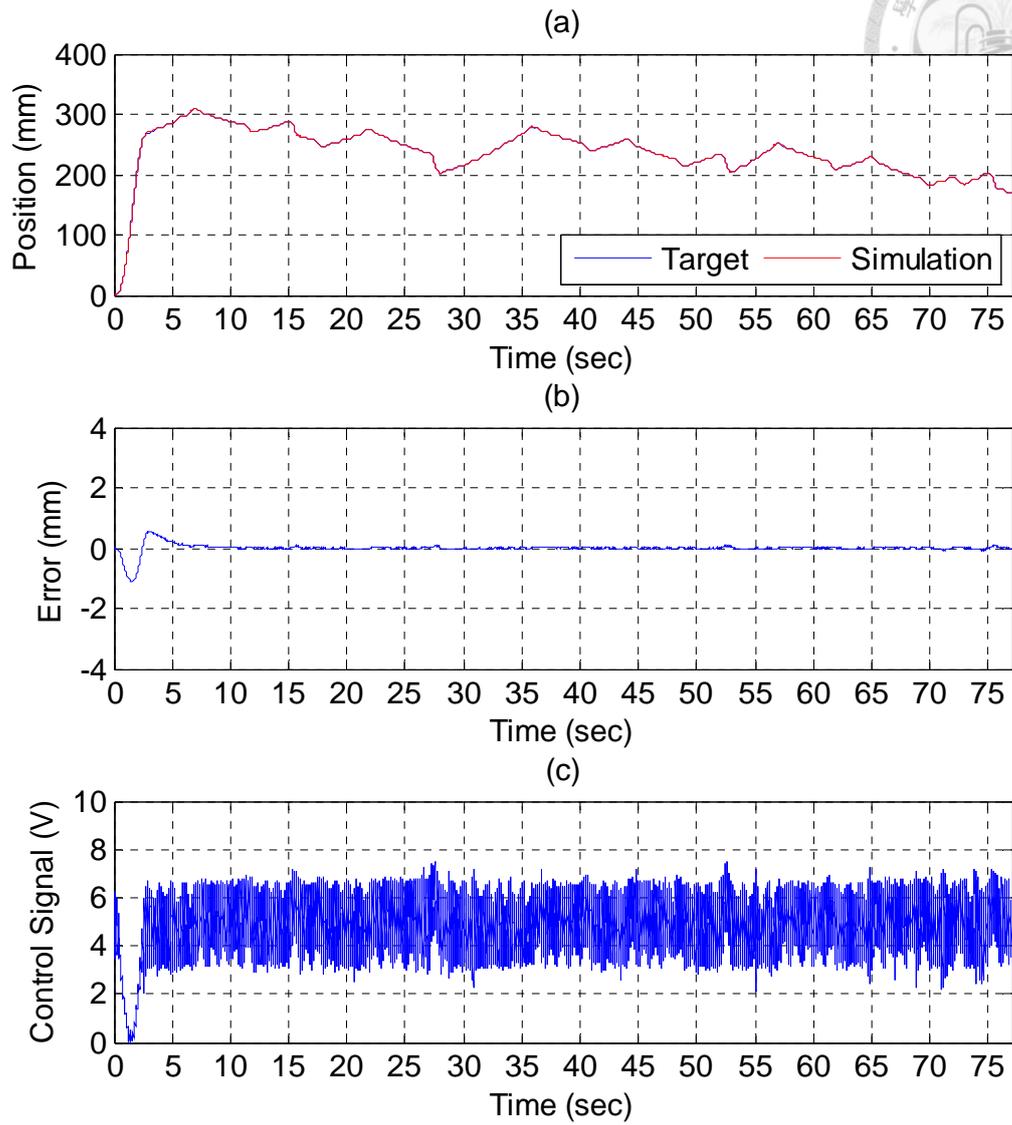


Fig. 6.56 Simulation results of end-effector path tracking control for

C-axis cylinder via a solid cuboid trajectory:

(a) position tracking response (b) position tracking error (c) control signal

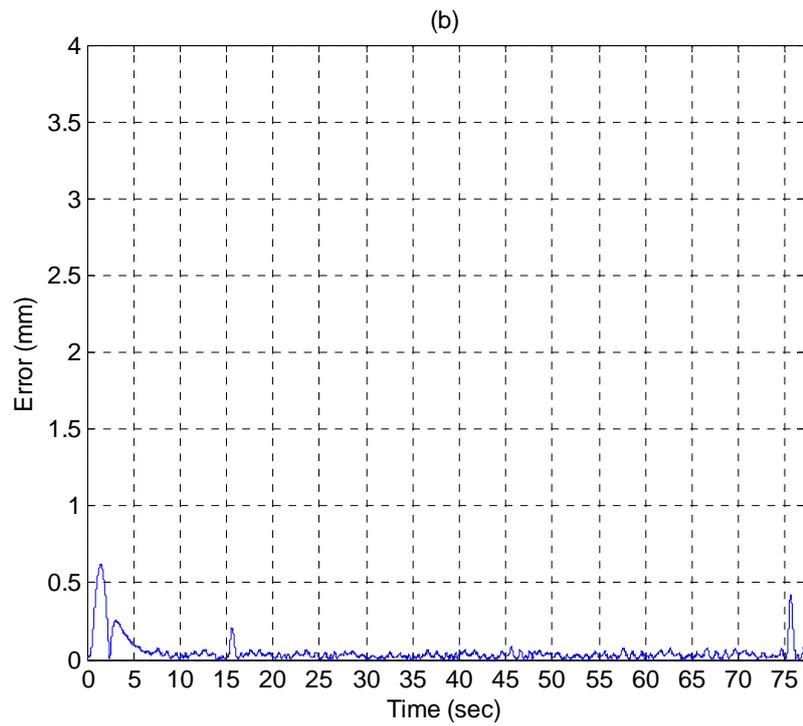
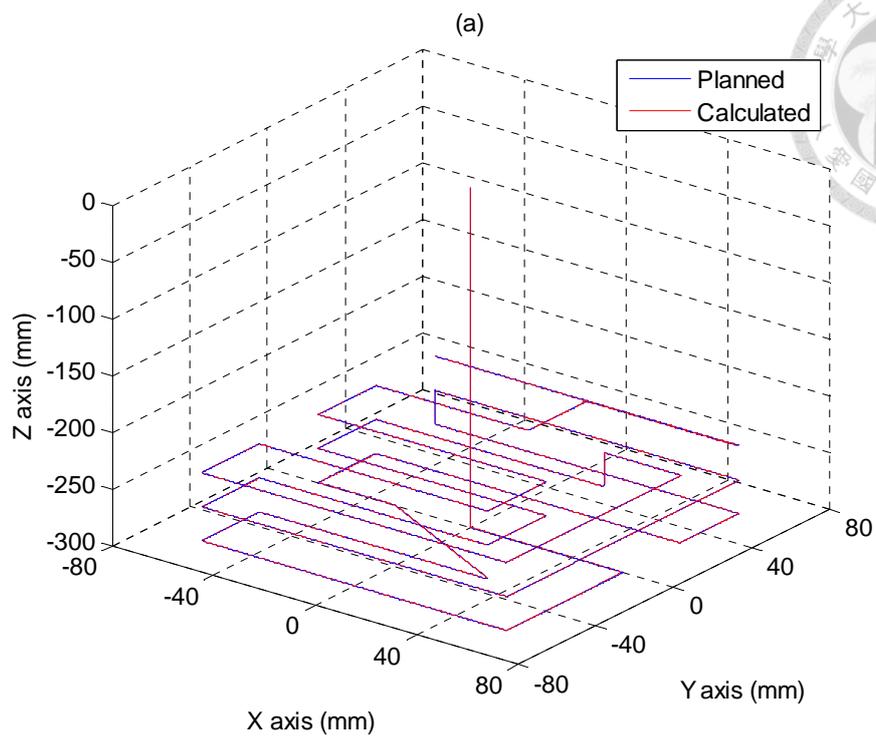
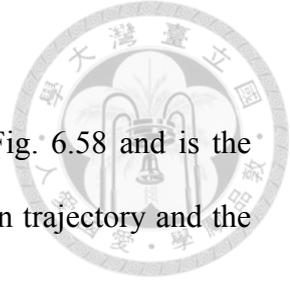


Fig. 6.57 Estimated end-effector position tracking response for a solid cuboid trajectory: (a) calculated end-effector position
(b) calculated end-effector position error

6.2.4 Simulation of Solid Polyhedron Trajectory



The desired printing object of solid polyhedron is shown in Fig. 6.58 and is the same as that in Section 6.1.4. The details of the end-effector position trajectory and the actuated position trajectory have been also discussed in Section 6.1.4.

The simulation results of path tracking control for A, B and C axis are shown in Fig. 6.59, Fig. 6.60, and Fig. 6.61. The simulation position errors of the manipulator actuator for A, B and C axis are below ± 2.0 mm. The control signals oscillate in the low velocity region and motion direction change of the cylinder. In addition, the estimated end-effector positions are calculated by the forward kinematics from the actuated joint positions of three cylinders and shown in Fig. 6.62. The maximum estimated error of the end-effector is about 1.3 mm at 62 sec.

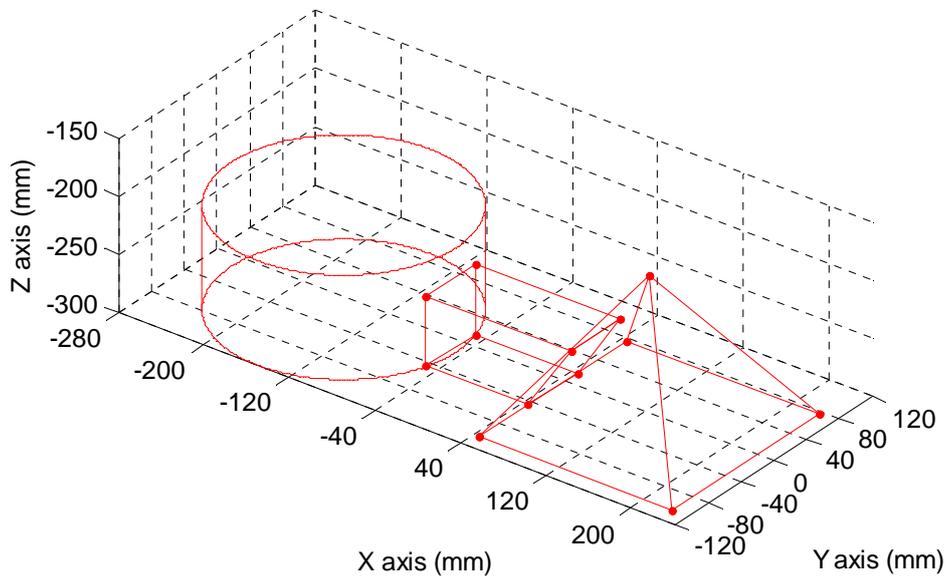


Fig. 6.58 Desired printing object of solid polyhedron

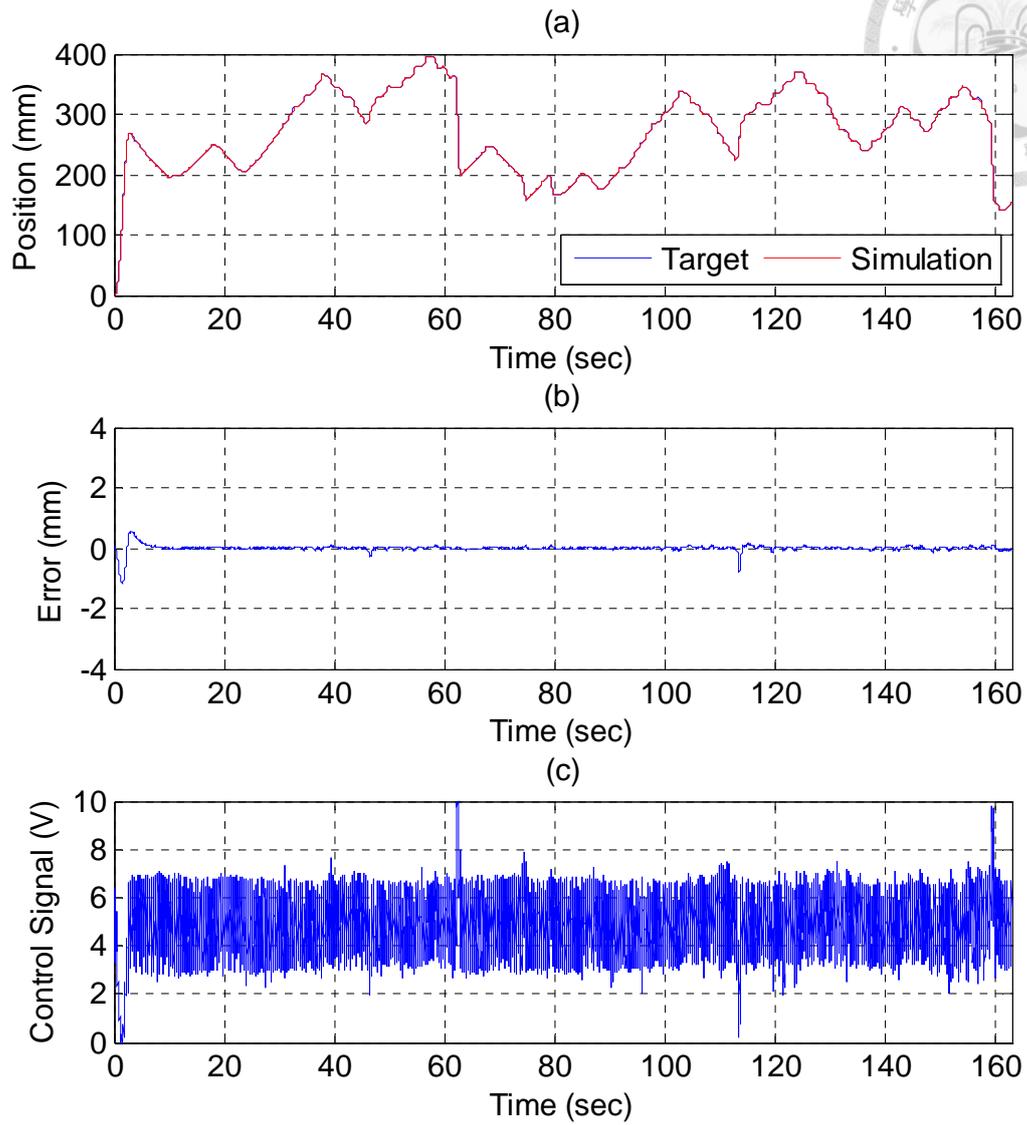


Fig. 6.59 Simulation results of end-effector path tracking control for

A-axis cylinder via a solid polyhedron trajectory:

(a) position tracking response (b) position tracking error (c) control signal

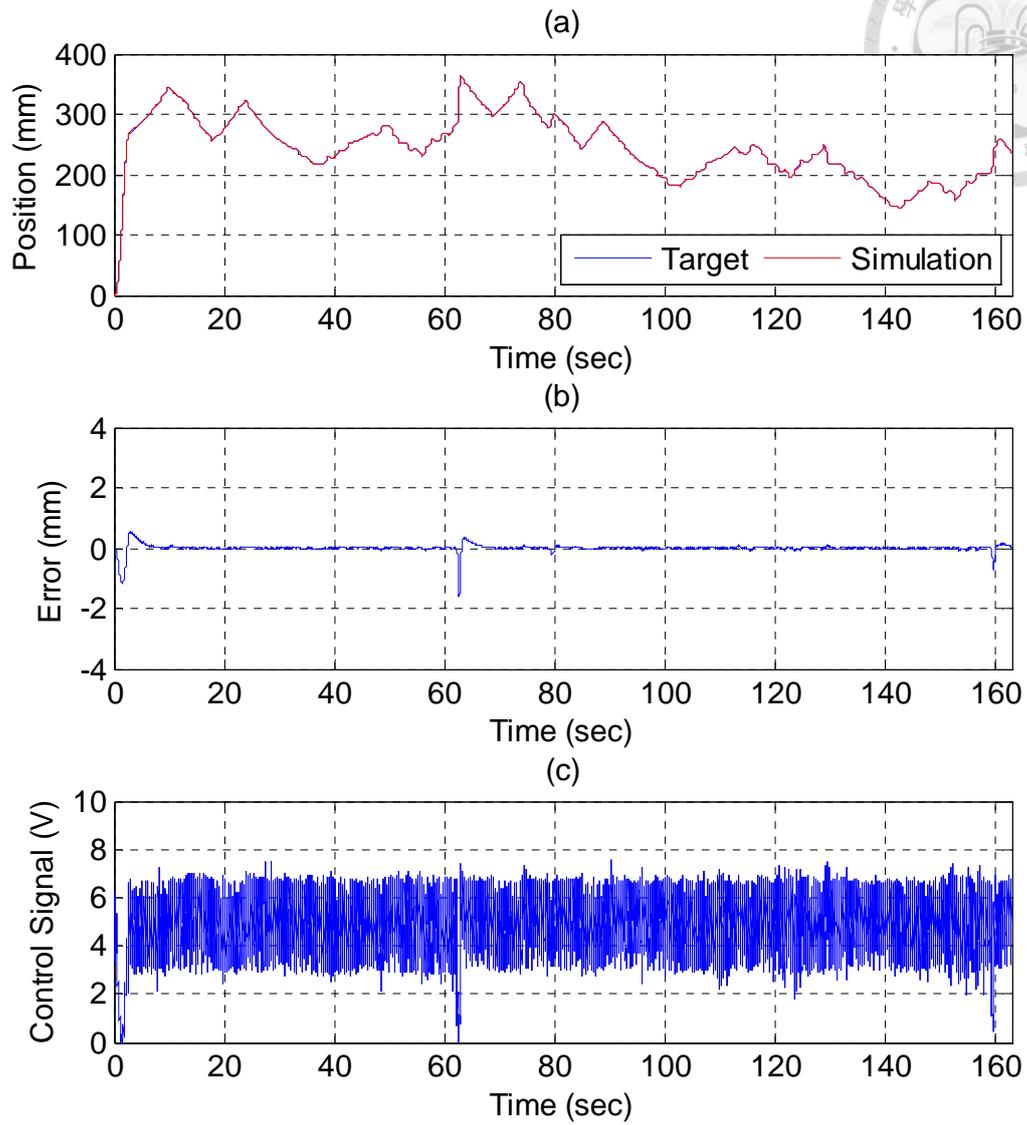


Fig. 6.60 Simulation results of end-effector path tracking control for

B-axis cylinder via a solid polyhedron trajectory:

(a) position tracking response (b) position tracking error (c) control signal

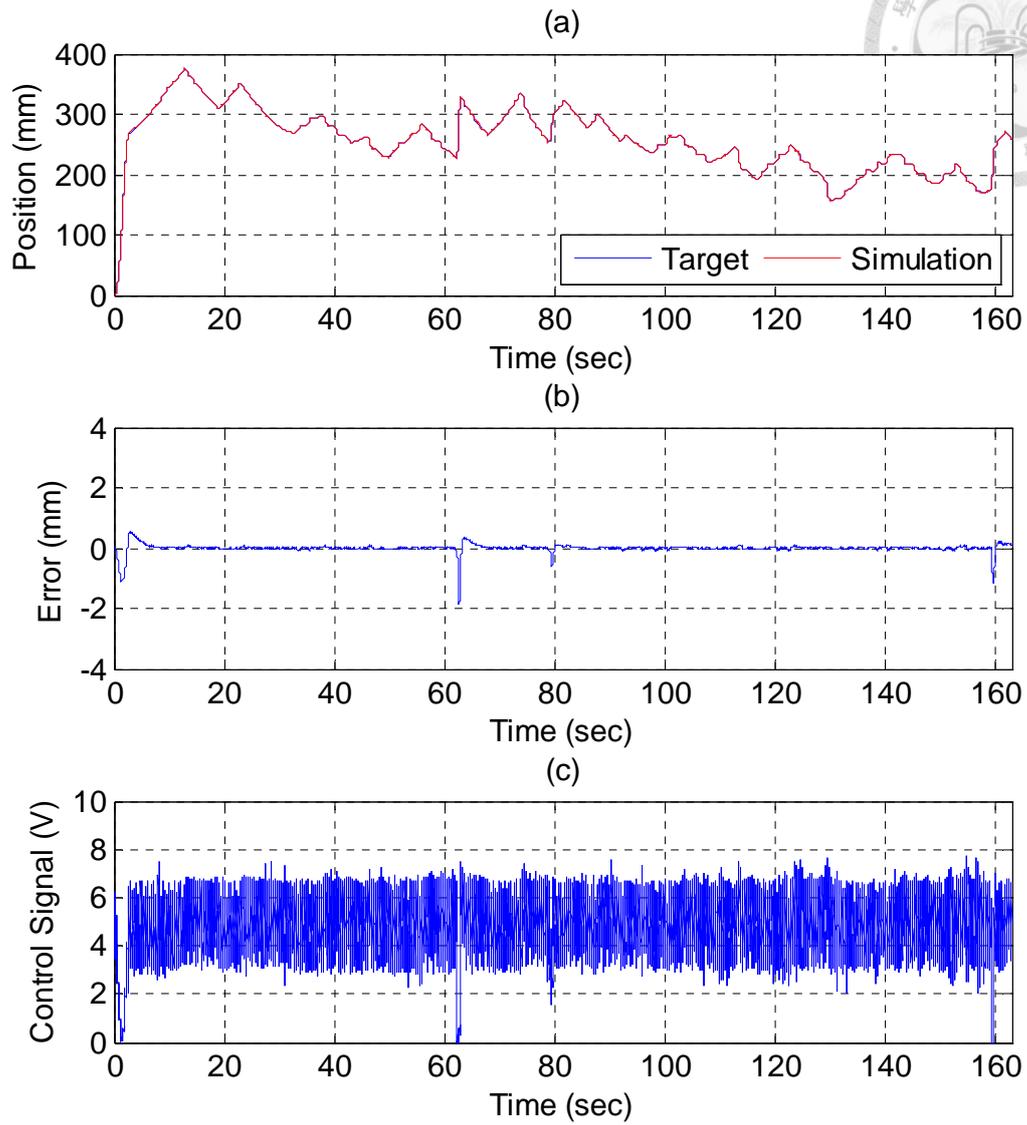


Fig. 6.61 Simulation results of end-effector path tracking control for

C-axis cylinder via a solid polyhedron trajectory:

(a) position tracking response (b) position tracking error (c) control signal

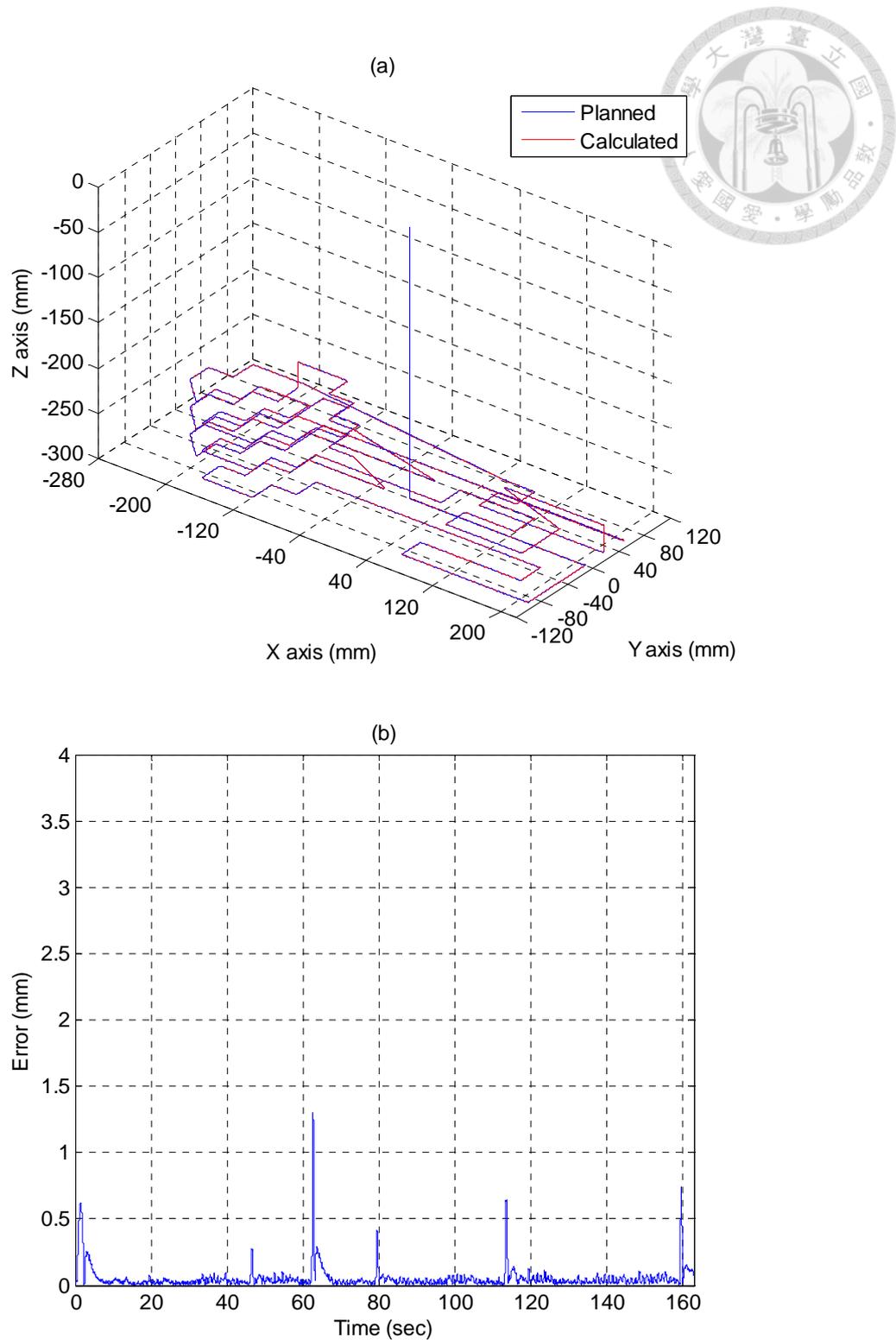
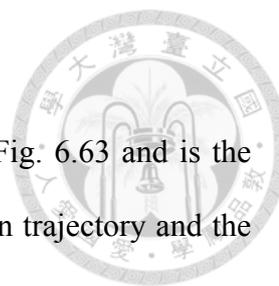


Fig. 6.62 Estimated end-effector position tracking response for a solid polyhedron trajectory: (a) calculated end-effector position
(b) calculated end-effector position error



6.2.5 Simulation of Solid Complex 3D Trajectory

The desired printing object of solid complex 3D is shown in Fig. 6.63 and is the same as that in Section 6.1.5. The details of the end-effector position trajectory and the actuated position trajectory have been also discussed in Section 6.1.5.

The simulation results of path tracking control for A, B and C axis are shown in Fig. 6.64, Fig. 6.65, and Fig. 6.66. The simulation position errors of the manipulator actuator for A, B and C axis are below ± 2.0 mm. The control signals oscillate in the low velocity region and motion direction change of the cylinder. In addition, the estimated end-effector positions are calculated by the forward kinematics from the actuated joint positions of three cylinders and shown in Fig. 6.67. The maximum estimated error of the end-effector is about 1.2 mm at 1.0 sec.

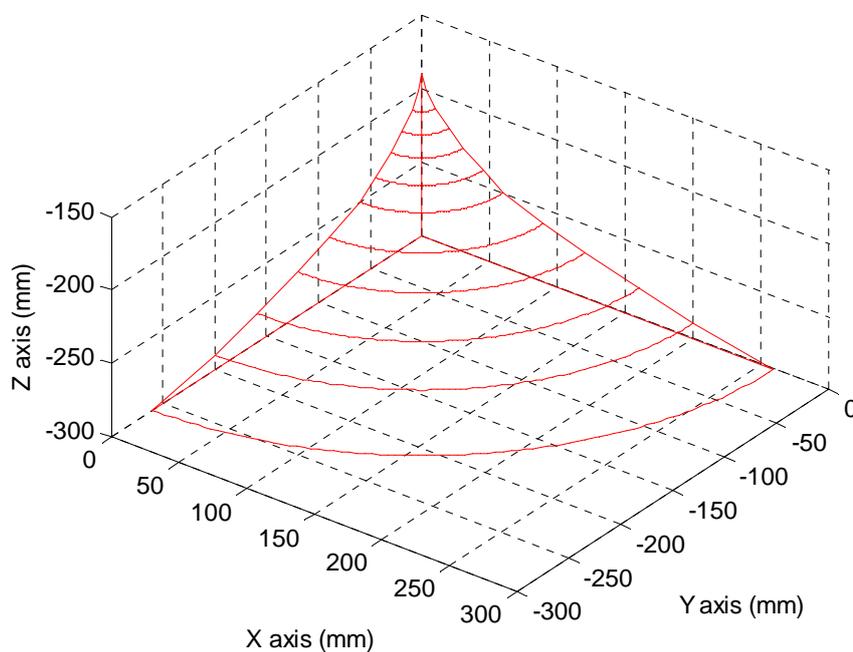


Fig. 6.63 Desired printing object of solid complex 3D

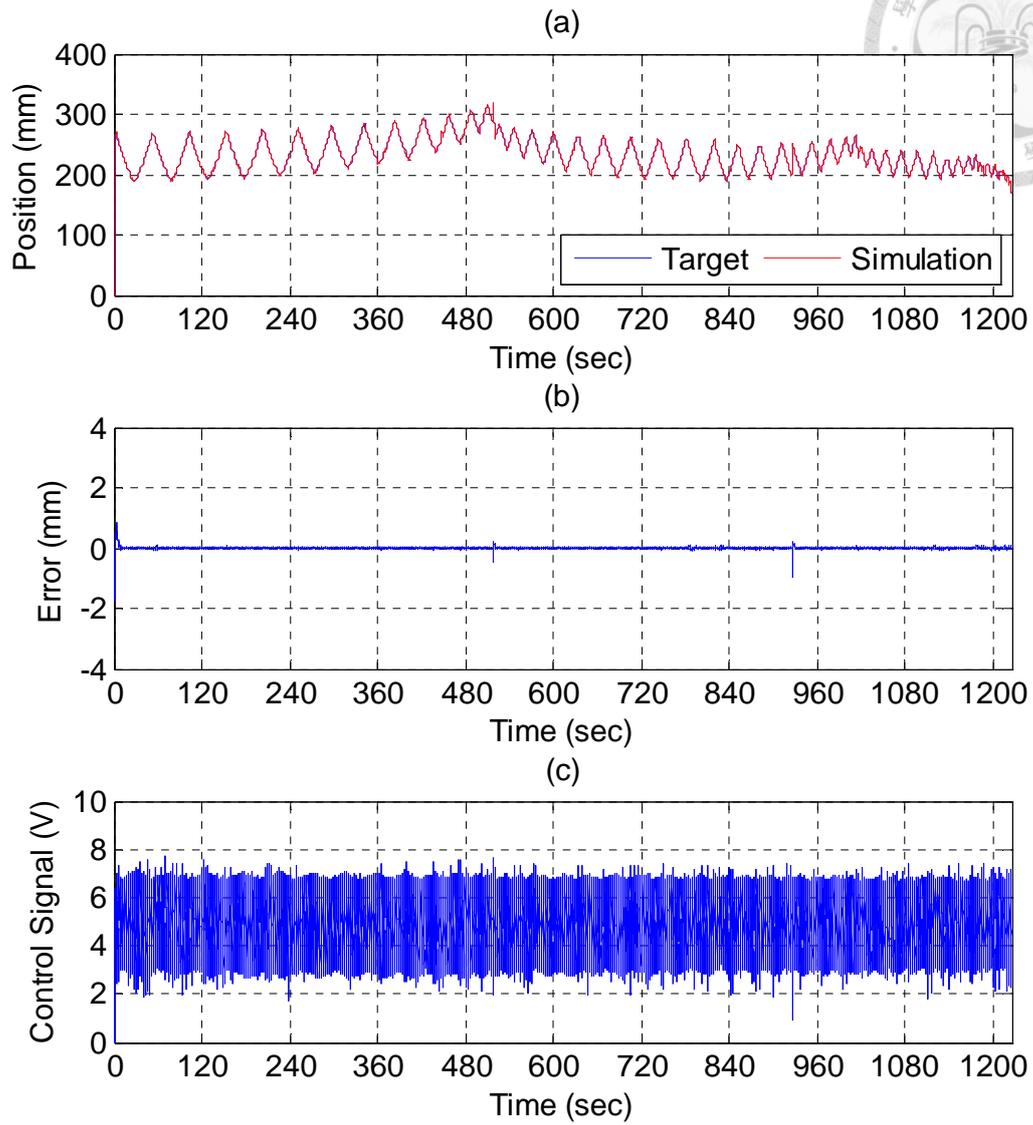


Fig. 6.64 Simulation results of end-effector path tracking control for

A-axis cylinder via a solid complex 3D trajectory:

(a) position tracking response (b) position tracking error (c) control signal

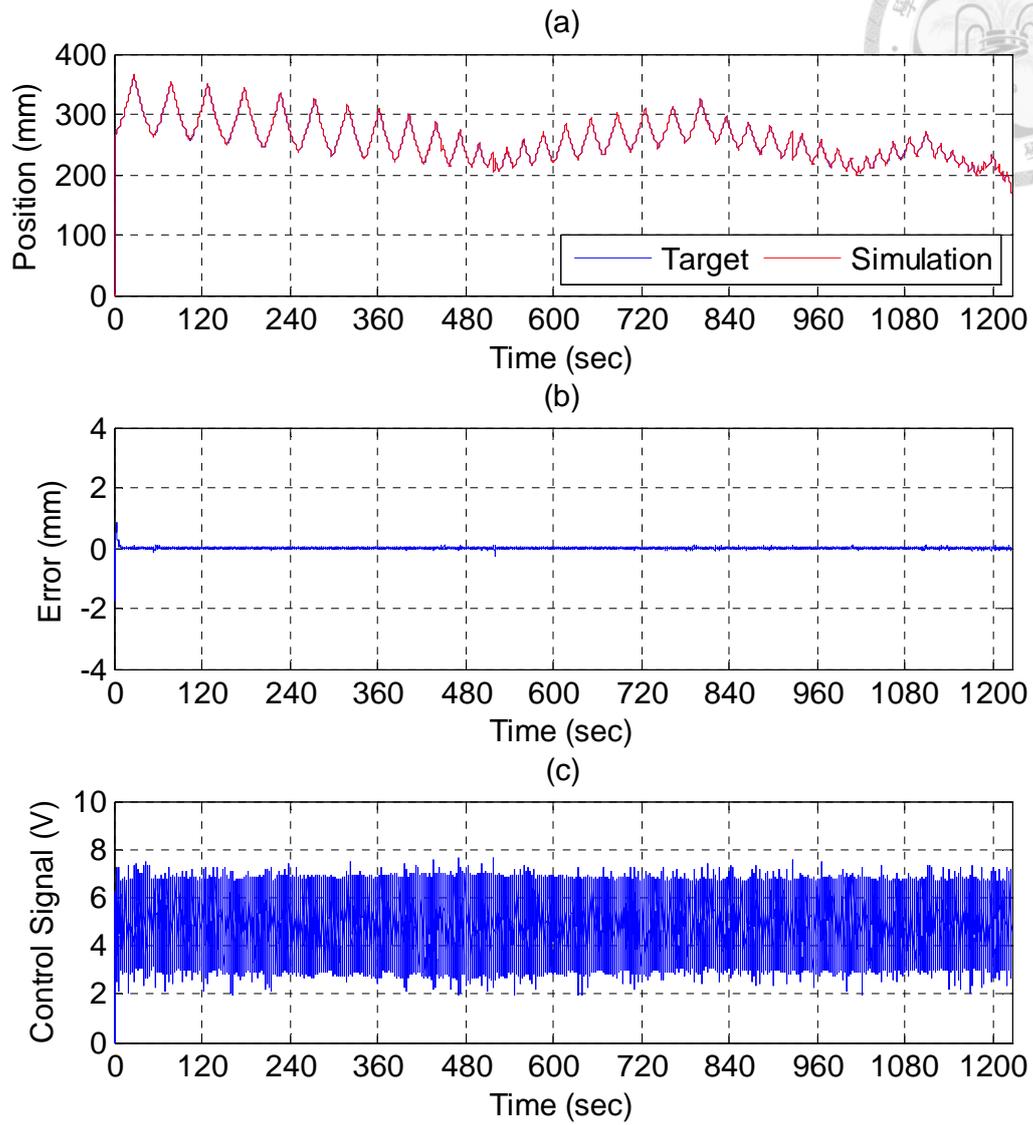


Fig. 6.65 Simulation results of end-effector path tracking control for

B-axis cylinder via a solid complex 3D trajectory:

(a) position tracking response (b) position tracking error (c) control signal

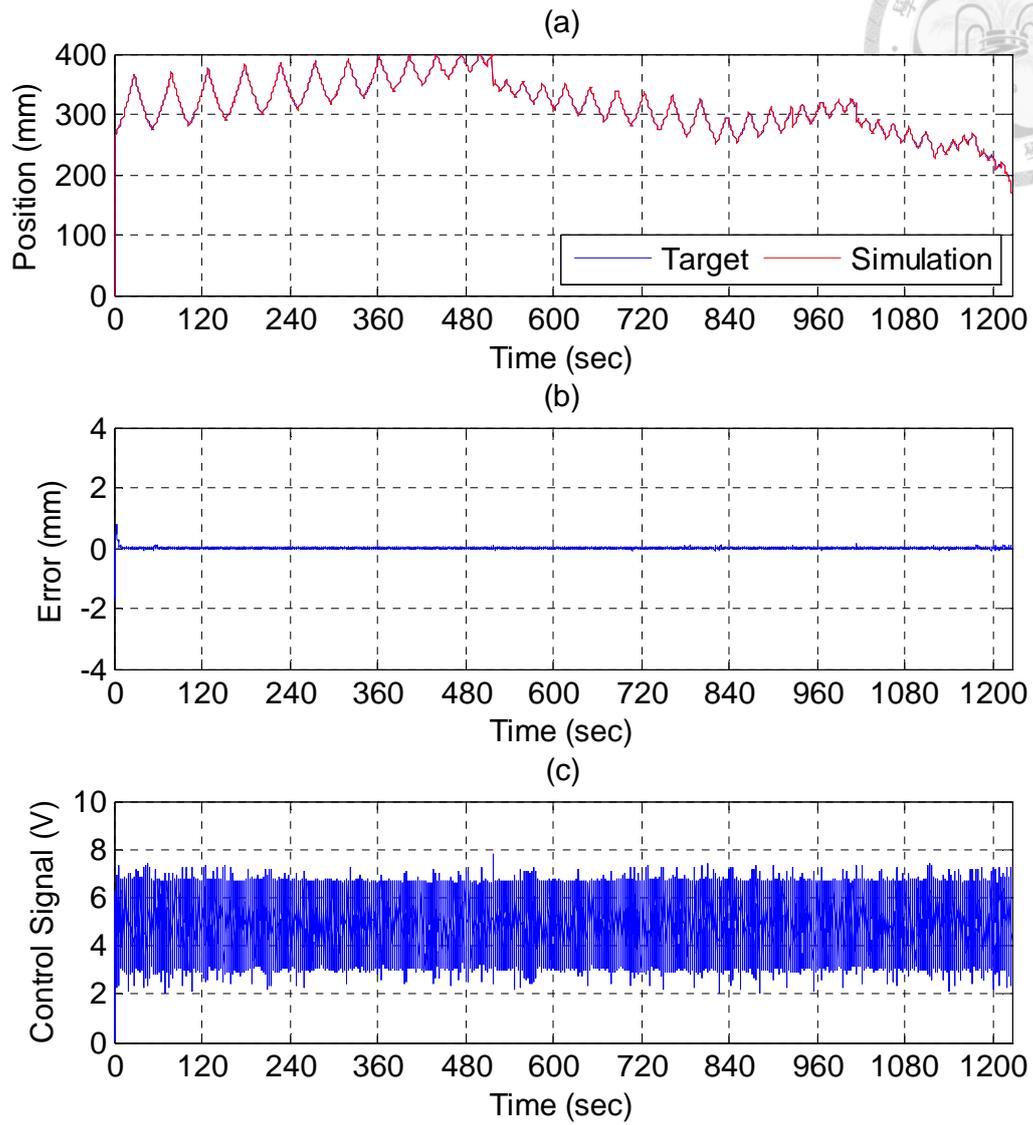


Fig. 6.66 Simulation results of end-effector path tracking control for

C-axis cylinder via a solid complex 3D trajectory:

(a) position tracking response (b) position tracking error (c) control signal

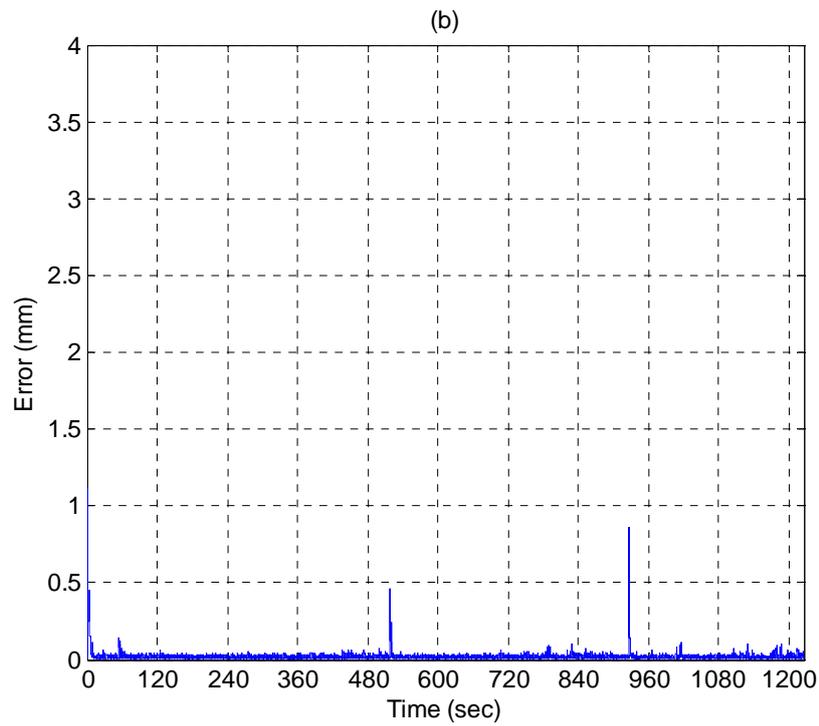
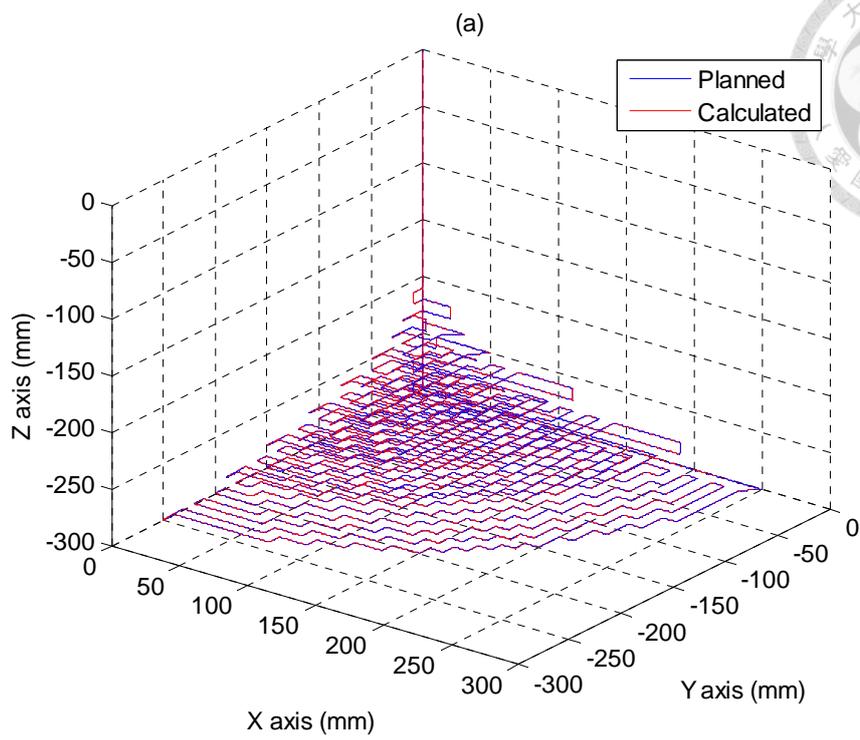


Fig. 6.67 Estimated end-effector position tracking response for a solid complex 3D trajectory: (a) calculated end-effector position
(b) calculated end-effector position error

6.3 Experiments of Three-Axial Pneumatic Parallel Manipulator by Path Tracking Control of End-Effector

In this section, the experiments of the proposed three-axial pneumatic parallel manipulator with the path tracking control of the manipulator end-effector are achieved. A number of the reference 3-D trajectories of end-effector are first designed and set up in the control program. Then, the designed end-effector trajectory is transformed into the actuated joint trajectories for the three cylinder actuators using the inverse kinematic model derived in Chapter 4. By controlling the positions of the three cylinder pistons according to the corresponding joint trajectories, the path tracking control of the end-effector can be accomplished.

The control scheme of three-axial manipulator is based on the proposed overall control system introduced in Chapter 5 and shown in Fig. 6.68. The output trajectories of three cylinder pistons from experiments are used to calculate the estimated position of end-effector by forward kinematics. The position errors of end-effector are also estimated from the tracking error of individual actuated joints.

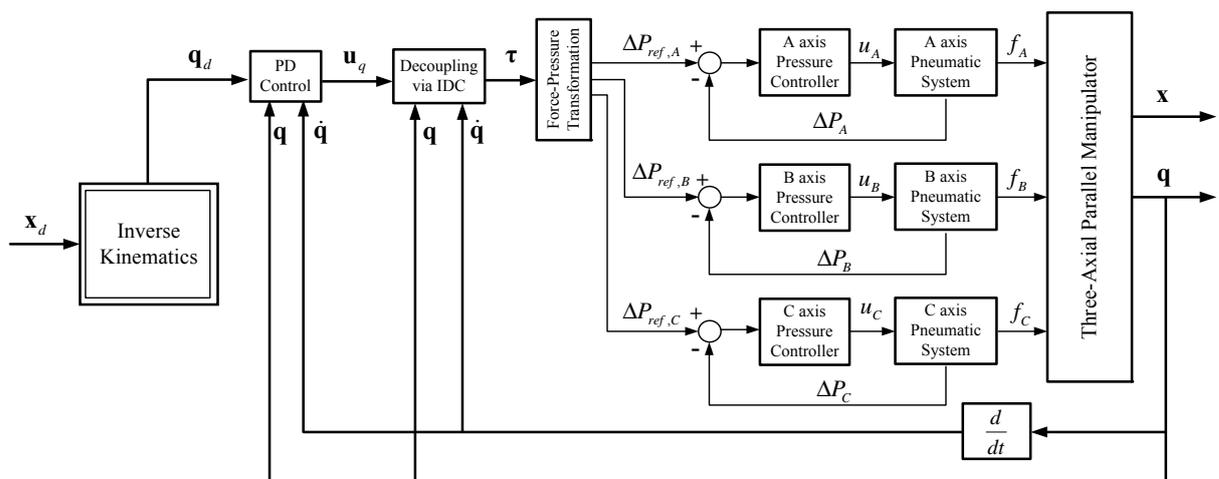
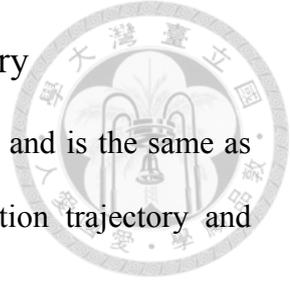


Fig. 6.68 Schematic diagram of the overall control system

6.3.1 Experimental Results of Circle-shape Line Trajectory



The designed circle-shape line trajectory is shown in Fig. 6.69 and is the same as that in Section 6.1.1 and 6.2.1. The details of end-effector position trajectory and actuated position trajectory have been also discussed in Section 6.1.1.

The experimental results of cylinder response on each axis are shown in Fig. 6.70, Fig. 6.71, and Fig. 6.72. In addition, the estimated end-effector position during the path tracking is calculated by the forward kinematics from the output actuated joint positions of three cylinders and shown in Fig. 6.73(a). The estimated position error of the manipulator end-effector is also calculated from the position error of the actuated joint and shown in Fig. 6.73(b). The top three significant errors during the path tracking control occur at the changing points of different trajectory segments as $t = 3.1, 3.4$ and 4.2 sec. The maximum estimated error of the end-effector is about 2.8 mm at 4.2 sec.

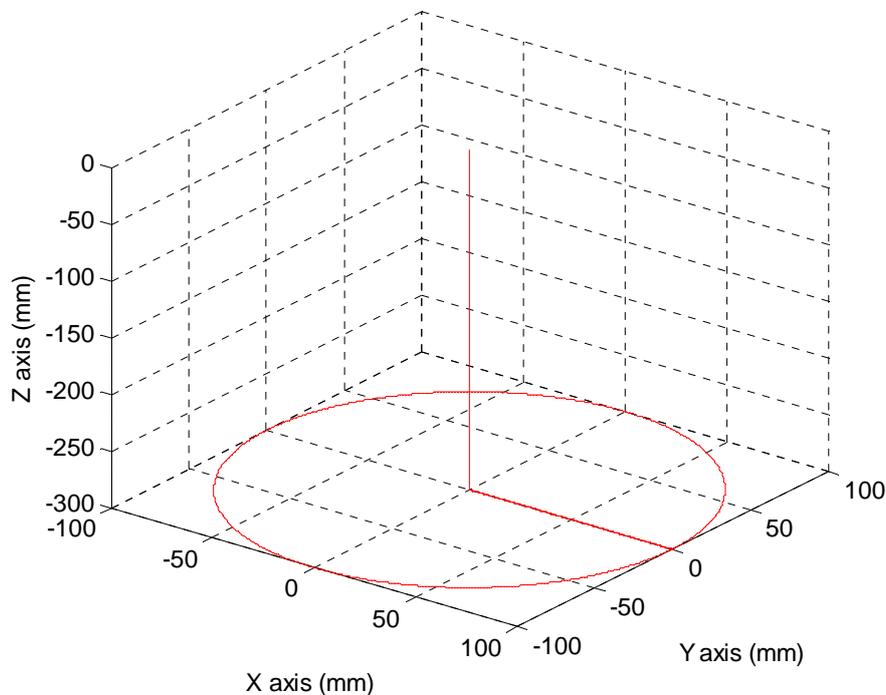


Fig. 6.69 Designed circle-shape line trajectory for end-effector

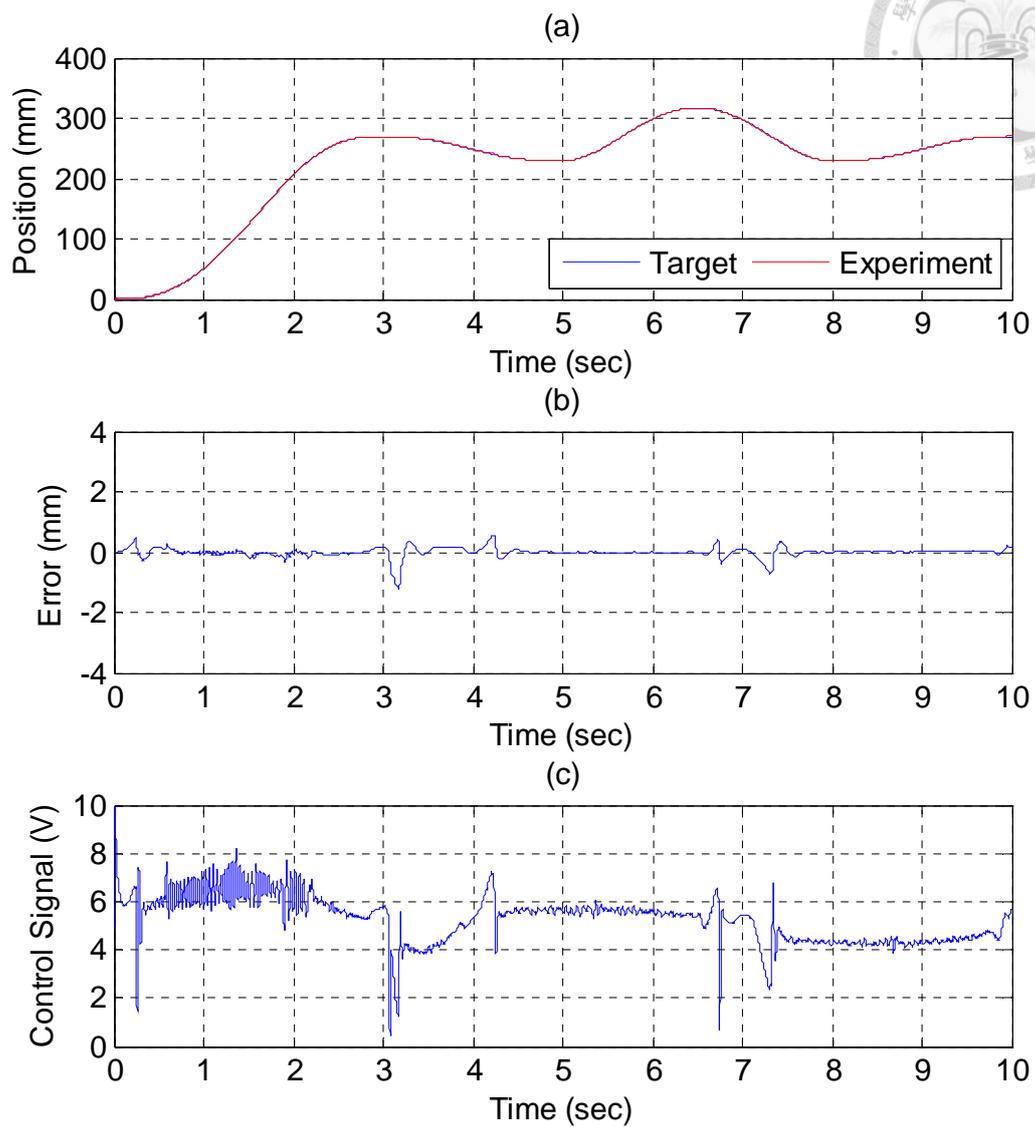


Fig. 6.70 Experimental results of end-effector path tracking control for

A-axis cylinder via a circle-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

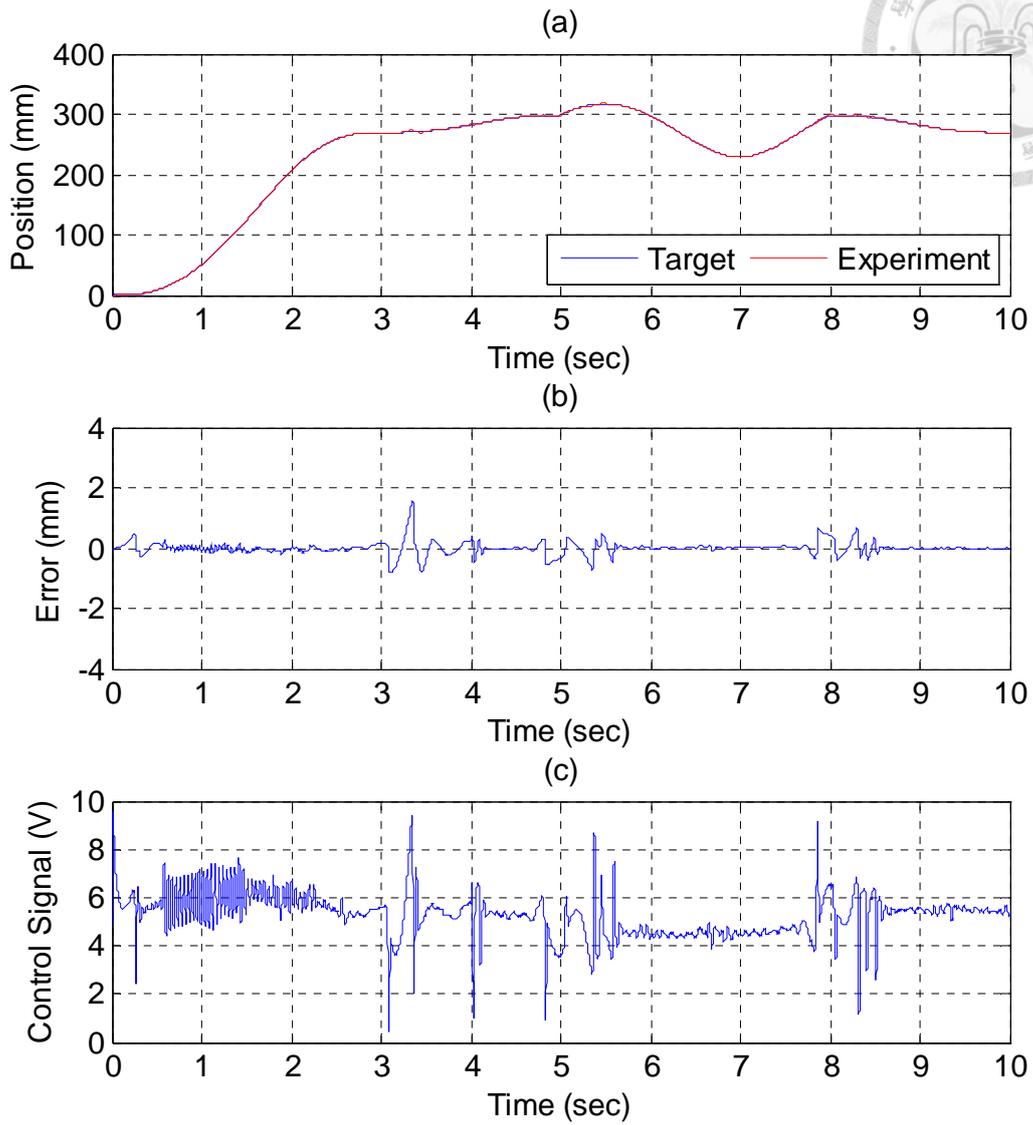


Fig. 6.71 Experimental results of end-effector path tracking control for

B-axis cylinder via a circle-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

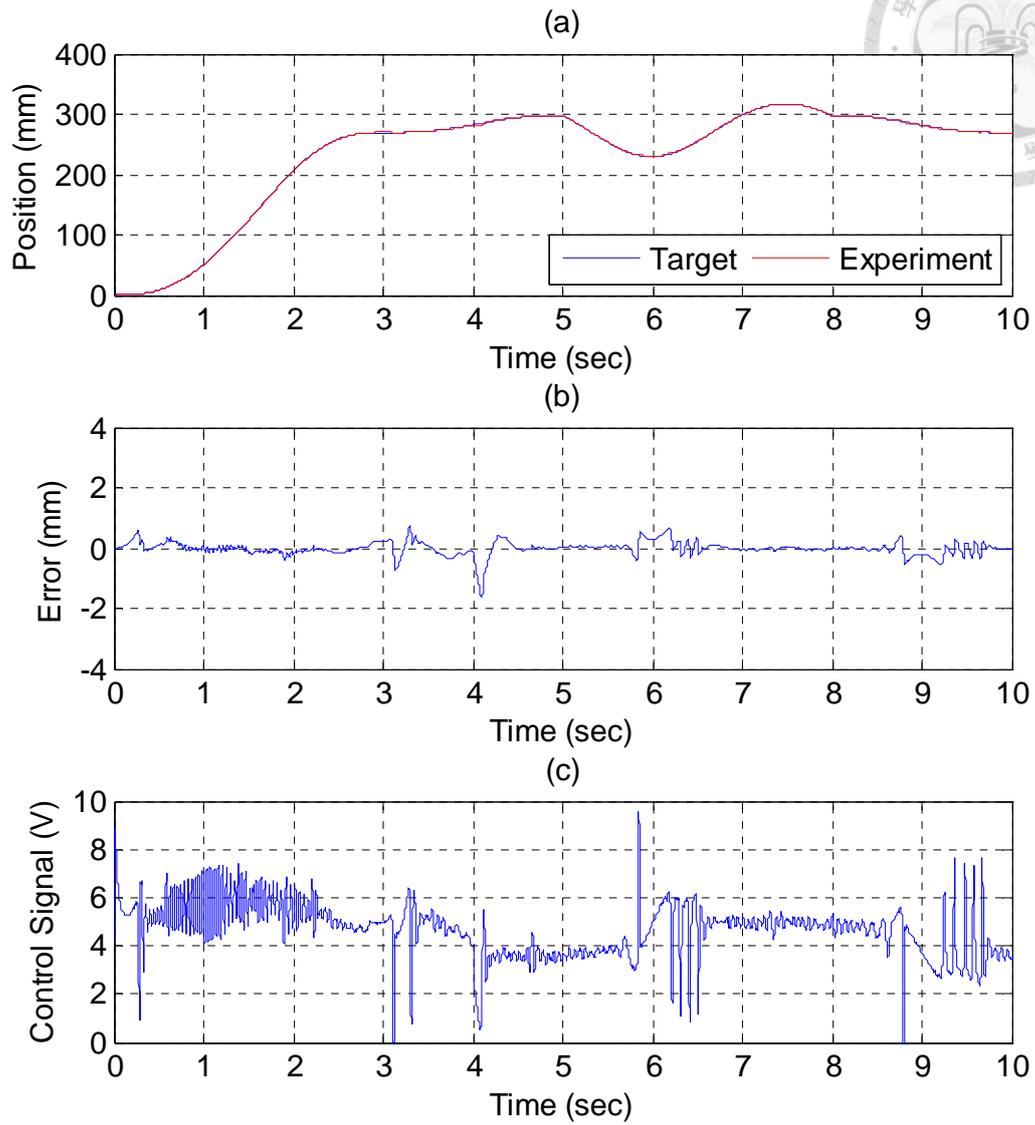


Fig. 6.72 Experimental results of end-effector path tracking control for

C-axis cylinder via a circle-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

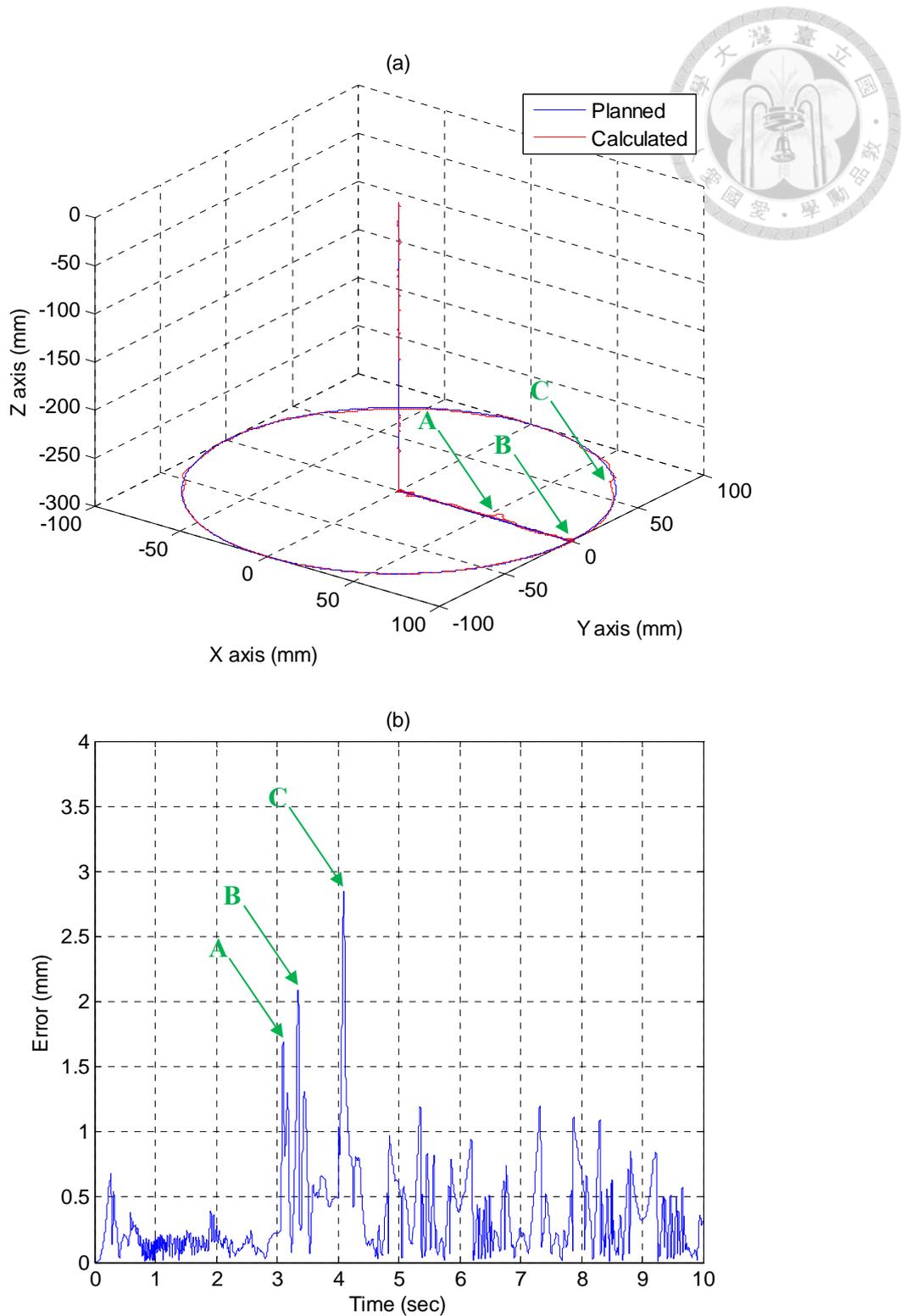


Fig. 6.73 Estimated end-effector position tracking response for a circle-shape line trajectory: (a) calculated end-effector position
(b) calculated end-effector position error

6.3.2 Experimental Results of Sphere-shape Line Trajectory

The designed sphere-shape line trajectory is shown in Fig. 6.74 and is the same as that in Section 6.1.2 and 6.2.2. The details of end-effector position trajectory and actuated position trajectory have been also discussed in Section 6.1.2.

The experimental results of cylinder response on each axis are shown in Fig. 6.75, Fig. 6.76, and Fig. 6.77. In addition, the estimated end-effector position during the path tracking is calculated by the forward kinematics from the output actuated joint positions of three cylinders and shown in Fig. 6.78(a). The estimated position error of the manipulator end-effector is also calculated from the position error of the actuated joint and shown in Fig. 6.78(b). The top three significant errors during the path tracking control occur at the changing points of different trajectory segments as $t = 3.4, 4.4$ and 6.4 sec. The maximum estimated error of the end-effector is about 3.0 mm at 4.4 sec.

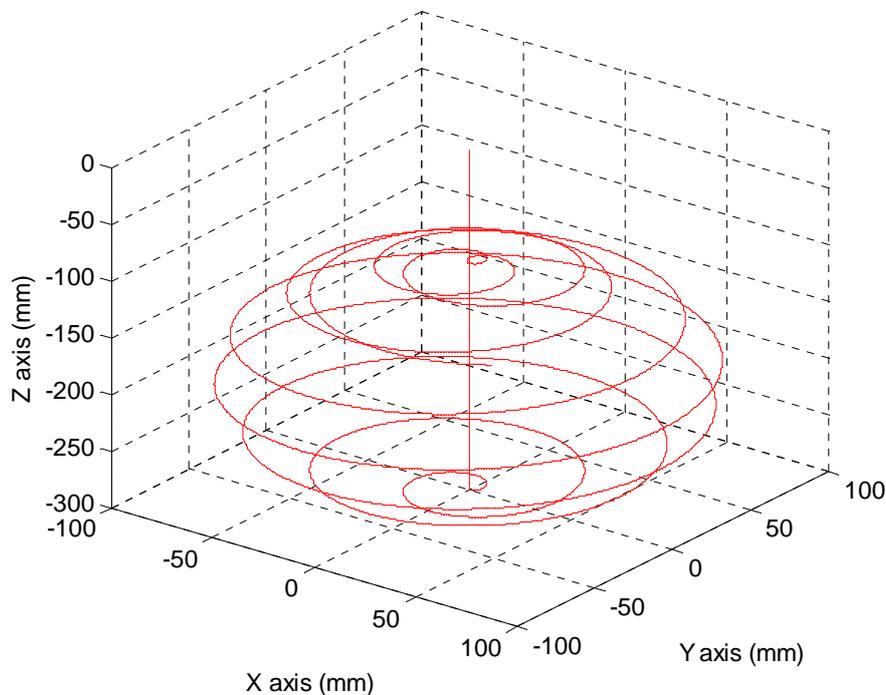


Fig. 6.74 Designed sphere-shape line trajectory for end-effector

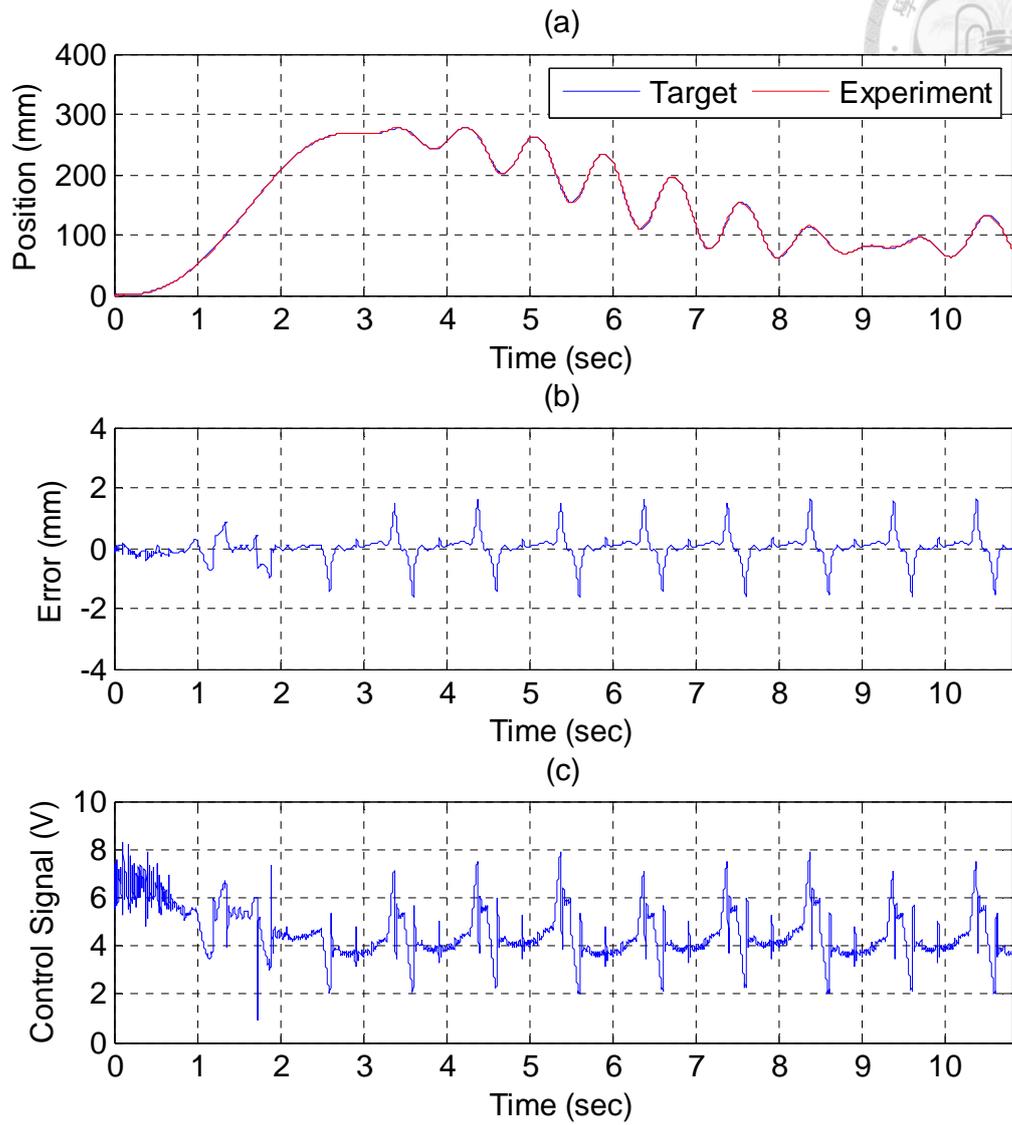


Fig. 6.75 Experimental results of end-effector path tracking control for

A-axis cylinder via a sphere-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

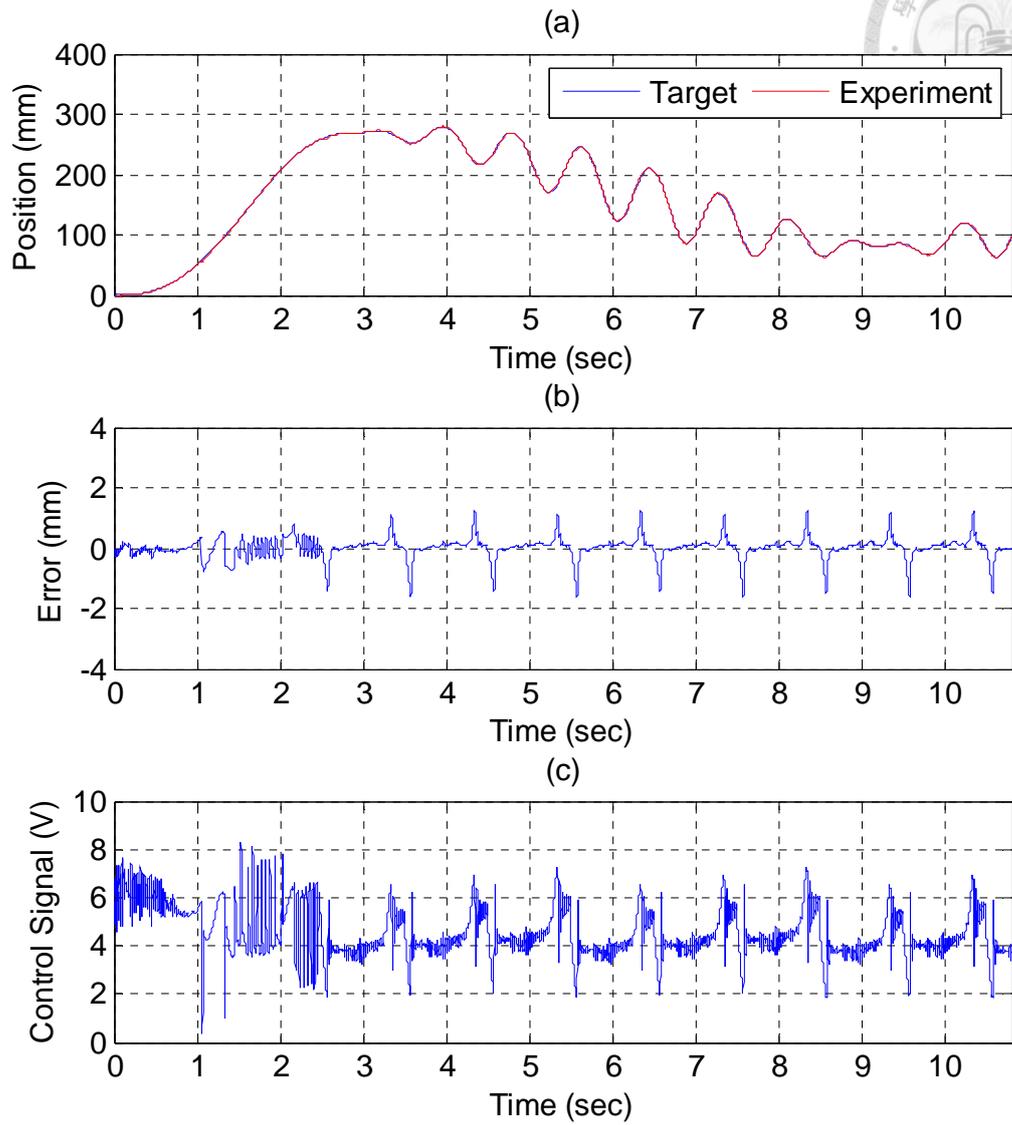


Fig. 6.76 Experimental results of end-effector path tracking control for

B-axis cylinder via a sphere-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

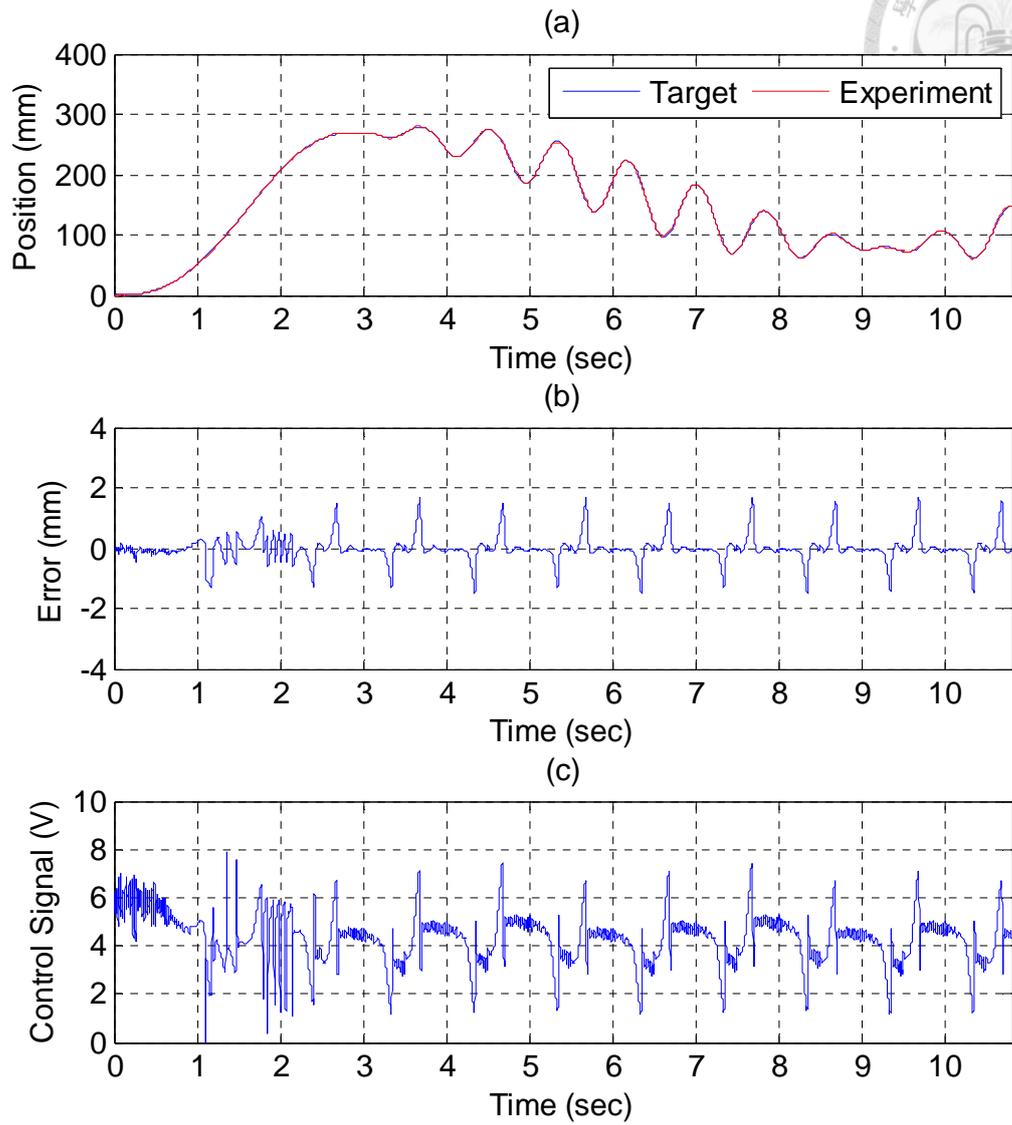


Fig. 6.77 Experimental results of end-effector path tracking control for

C-axis cylinder via a sphere-shape line trajectory:

(a) position tracking response (b) position tracking error (c) control signal

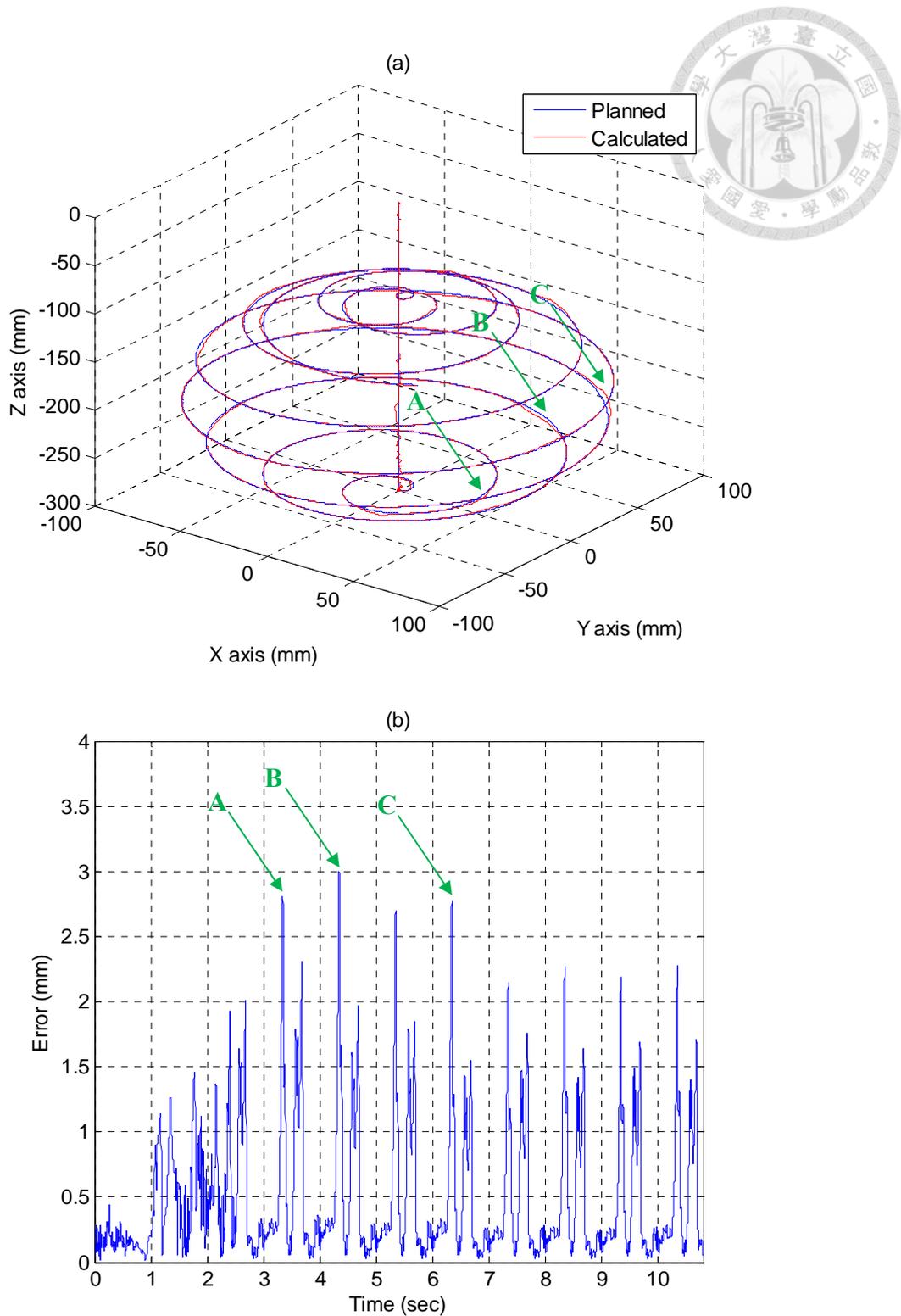
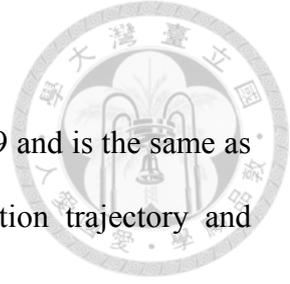


Fig. 6.78 Estimated end-effector position tracking response for a sphere-shape line trajectory: (a) calculated end-effector position
(b) calculated end-effector position error

6.3.3 Experimental Results of Solid Cuboid Trajectory



The desired printing object of solid cuboid is shown in Fig. 6.79 and is the same as that in Section 6.1.3 and 6.2.3. The details of end-effector position trajectory and actuated position trajectory have been also discussed in Section 6.1.3.

The experimental results of cylinder response on each axis are shown in Fig. 6.80, Fig. 6.81, and Fig. 6.82. In addition, the estimated end-effector position during the path tracking is calculated by the forward kinematics from the output actuated joint positions of three cylinders and shown in Fig. 6.83(a). The estimated position error of the manipulator end-effector is also calculated from the position error of the actuated joint and shown in Fig. 6.83(b). The top three significant errors during the path tracking control occur at the changing points of different trajectory segments as $t = 47, 64$ and 67 sec. The maximum estimated error of the end-effector is about 2.9 mm at 47 sec.

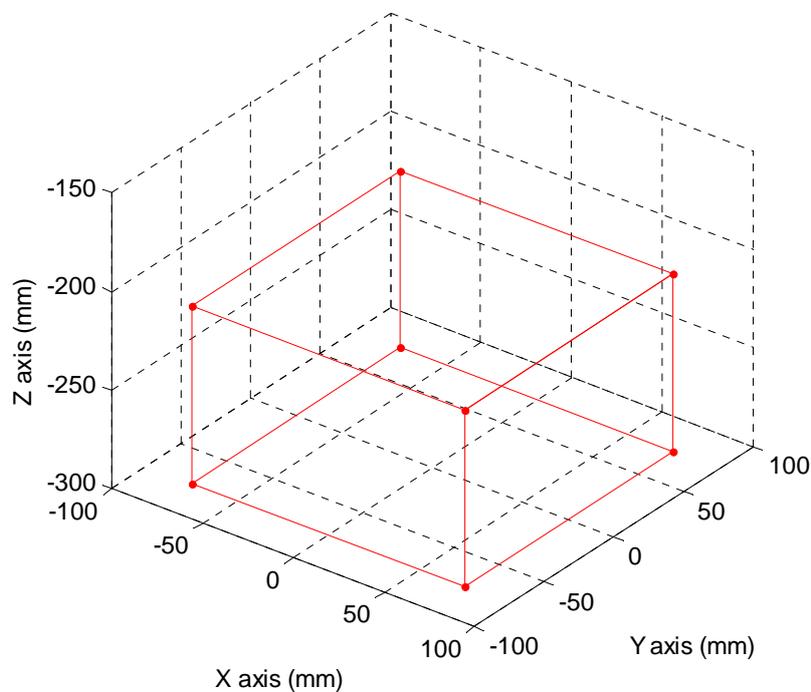


Fig. 6.79 Desired printing object of solid cuboid

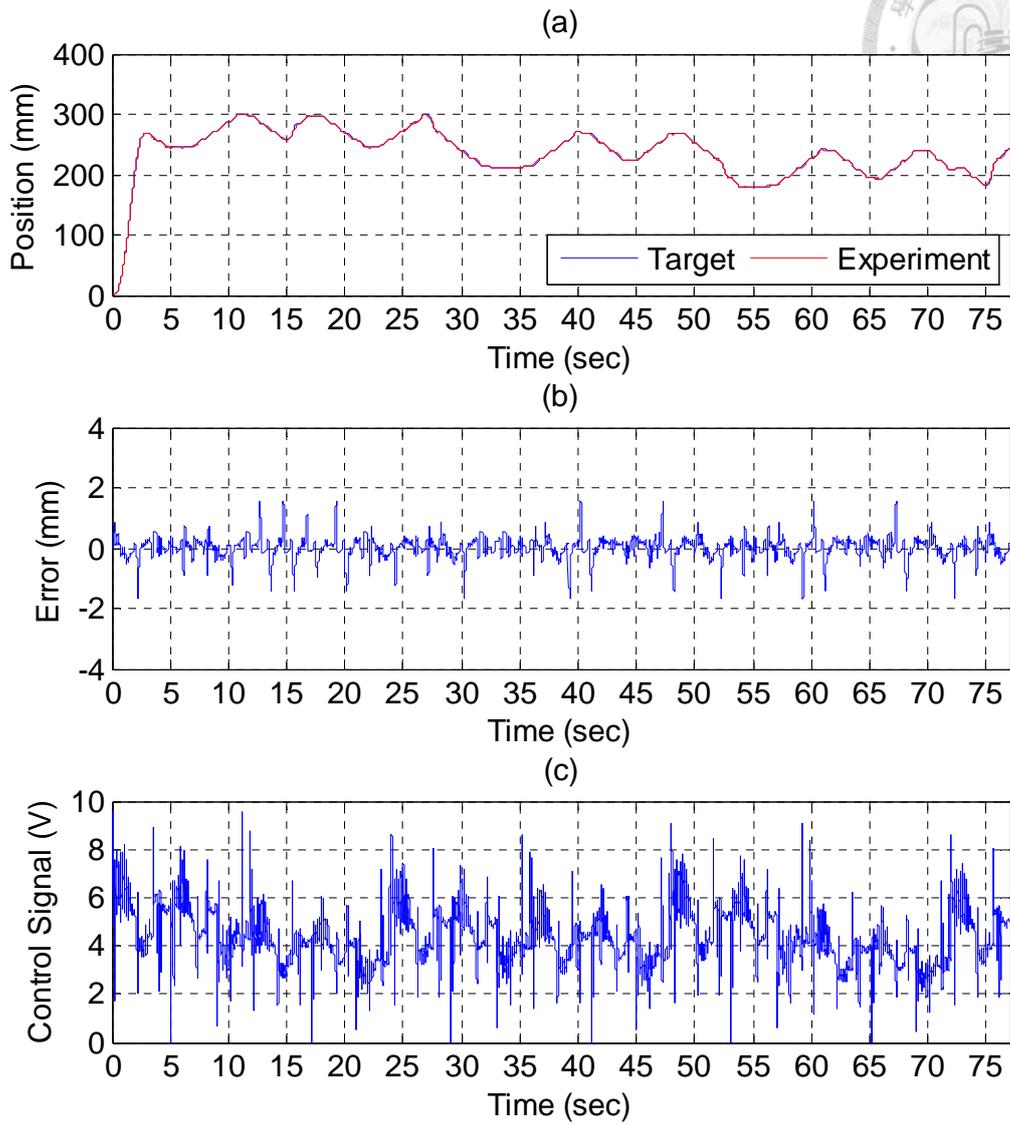


Fig. 6.80 Experimental results of end-effector path tracking control for

A-axis cylinder via a solid cuboid trajectory:

(a) position tracking response (b) position tracking error (c) control signal

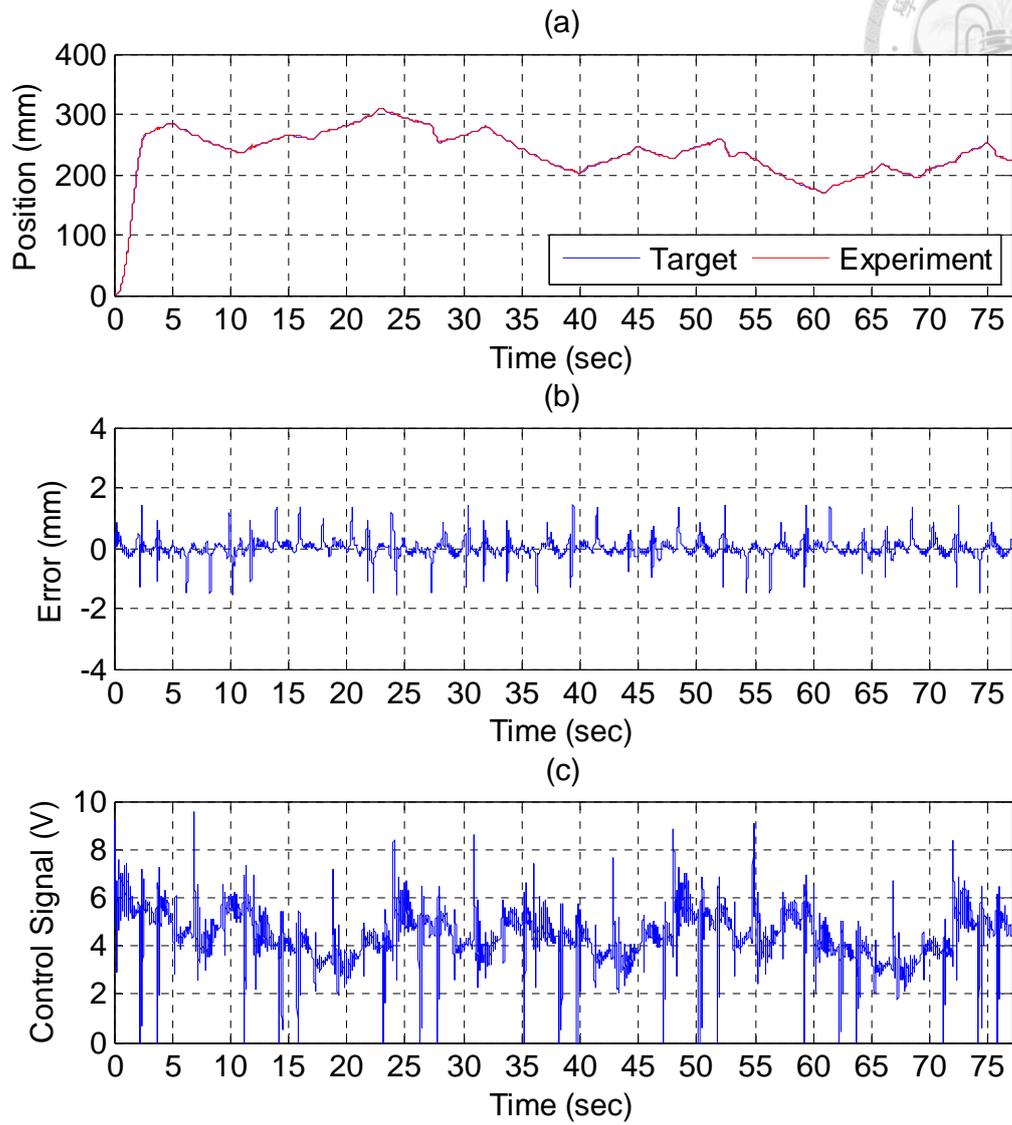


Fig. 6.81 Experimental results of end-effector path tracking control for

B-axis cylinder via a solid cuboid trajectory:

(a) position tracking response (b) position tracking error (c) control signal

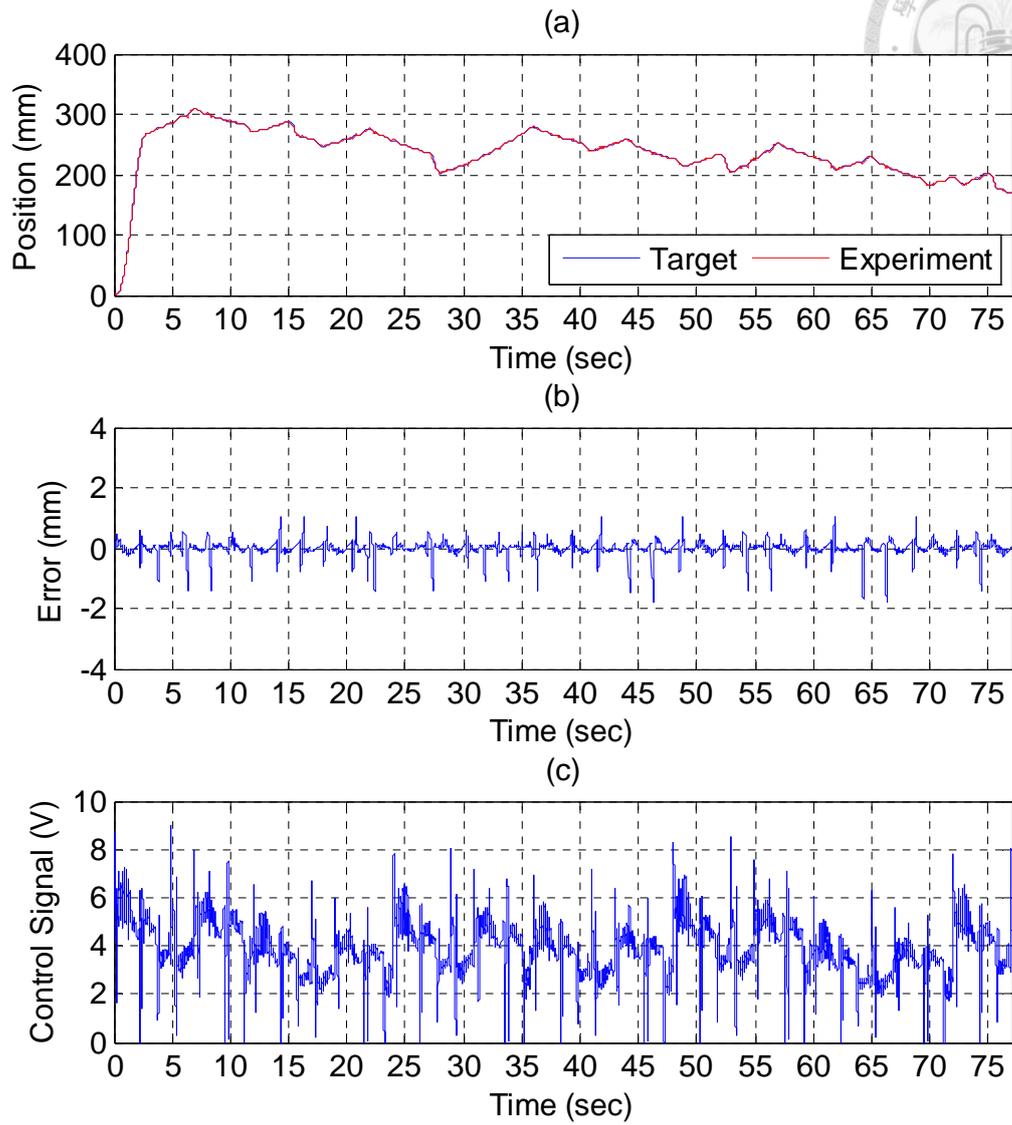


Fig. 6.82 Experimental results of end-effector path tracking control for

C-axis cylinder via a solid cuboid trajectory:

(a) position tracking response (b) position tracking error (c) control signal

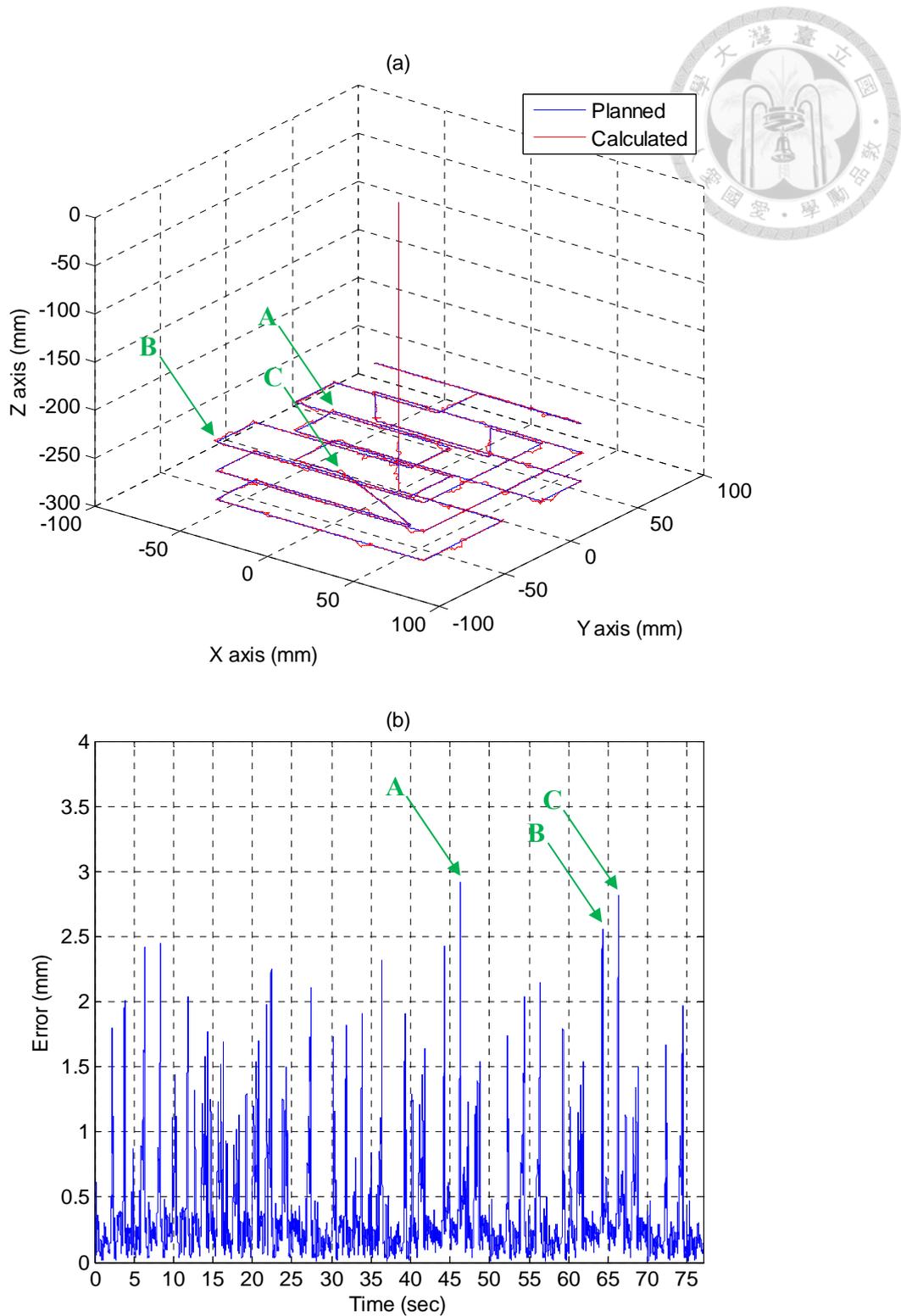


Fig. 6.83 Estimated end-effector position tracking response for a solid cuboid trajectory: (a) calculated end-effector position
(b) calculated end-effector position error

6.3.4 Experimental Results of Solid Polyhedron Trajectory

The desired printing object of solid polyhedron is shown in Fig. 6.84 and is the same as that in Section 6.1.4 and 6.2.4. The details of end-effector position trajectory and actuated position trajectory have been also discussed in Section 6.1.4.

The experimental results of cylinder response on each axis are shown in Fig. 6.85, Fig. 6.86, and Fig. 6.87. In addition, the estimated end-effector position during the path tracking is calculated by the forward kinematics from the output actuated joint positions of three cylinders and shown in Fig. 6.88(a). The estimated position error of the manipulator end-effector is also calculated from the position error of the actuated joint and shown in Fig. 6.88(b). The top three significant errors during the path tracking control occur at the changing points of different trajectory segments as $t = 46, 64$ and 68 sec. The maximum estimated error of the end-effector is about 2.8 mm at 64 sec.

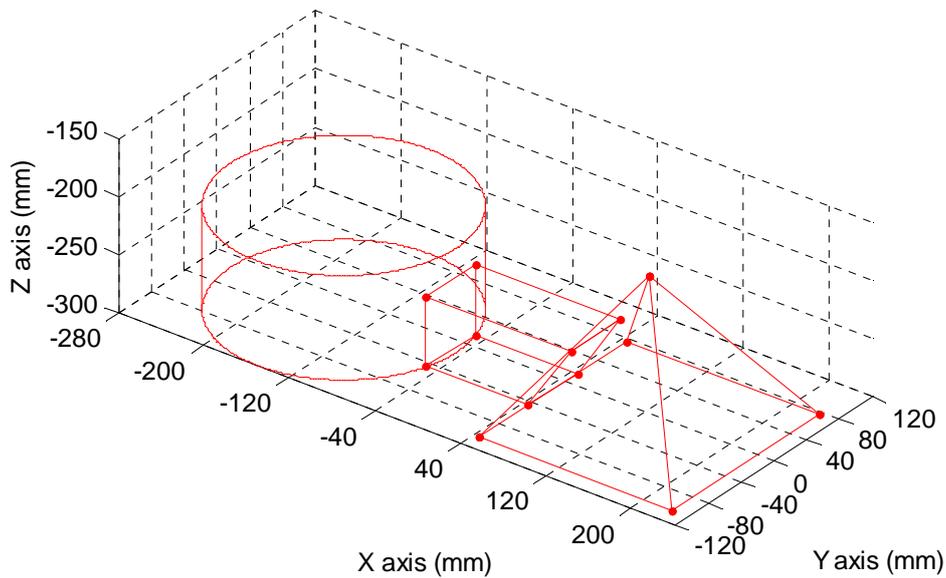


Fig. 6.84 Desired printing object of solid polyhedron

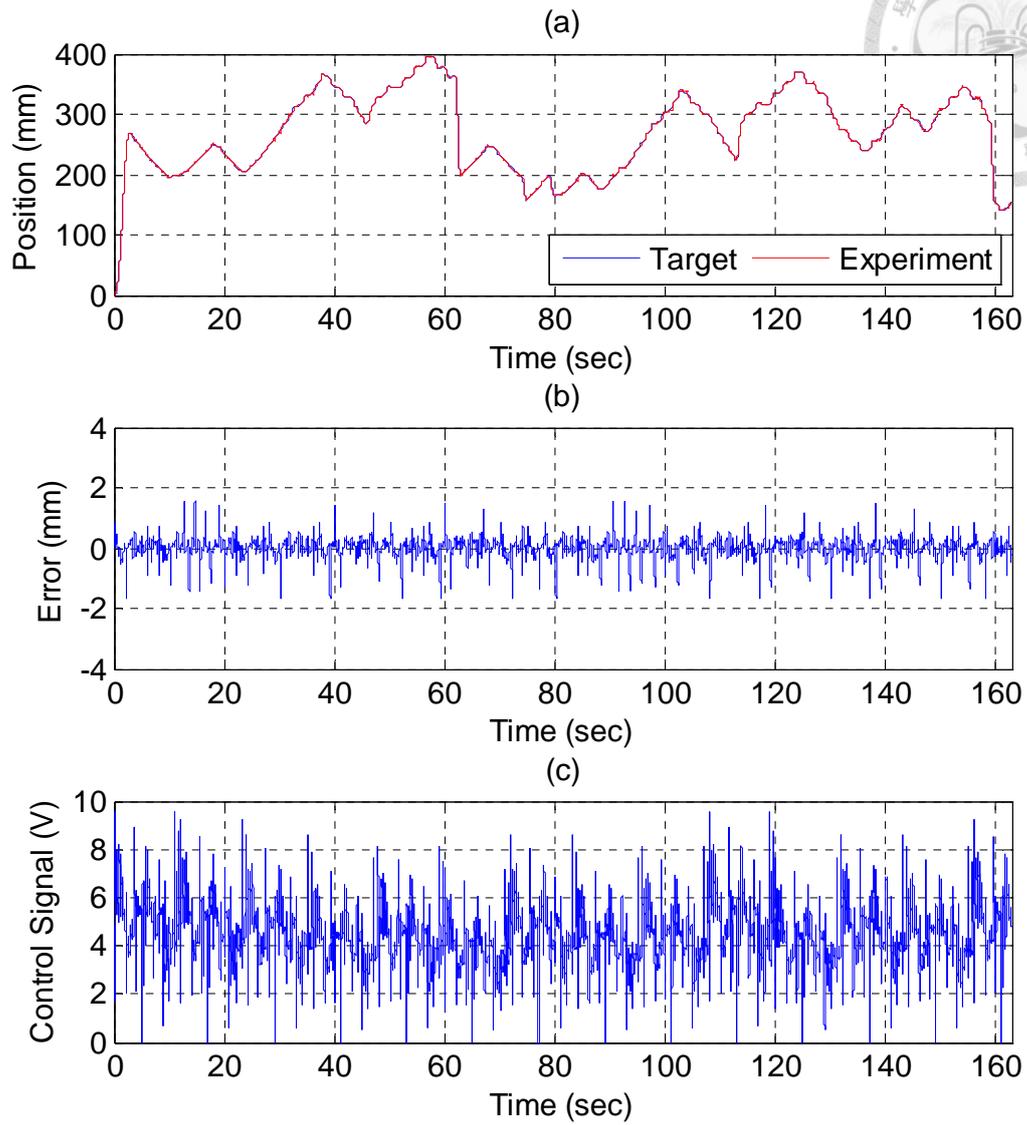


Fig. 6.85 Experimental results of end-effector path tracking control for

A-axis cylinder via a solid polyhedron trajectory:

(a) position tracking response (b) position tracking error (c) control signal

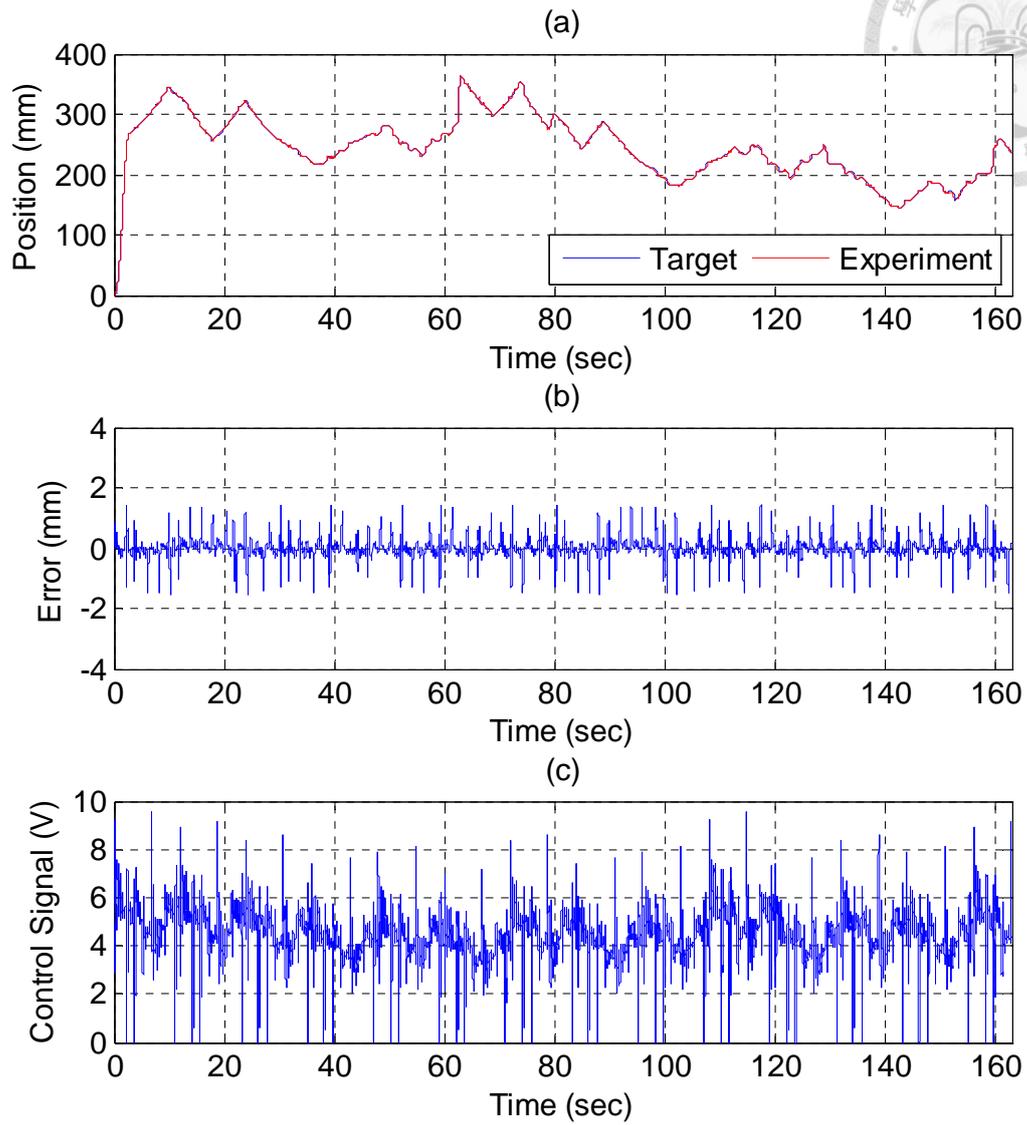


Fig. 6.86 Experimental results of end-effector path tracking control for

B-axis cylinder via a solid polyhedron trajectory:

(a) position tracking response (b) position tracking error (c) control signal

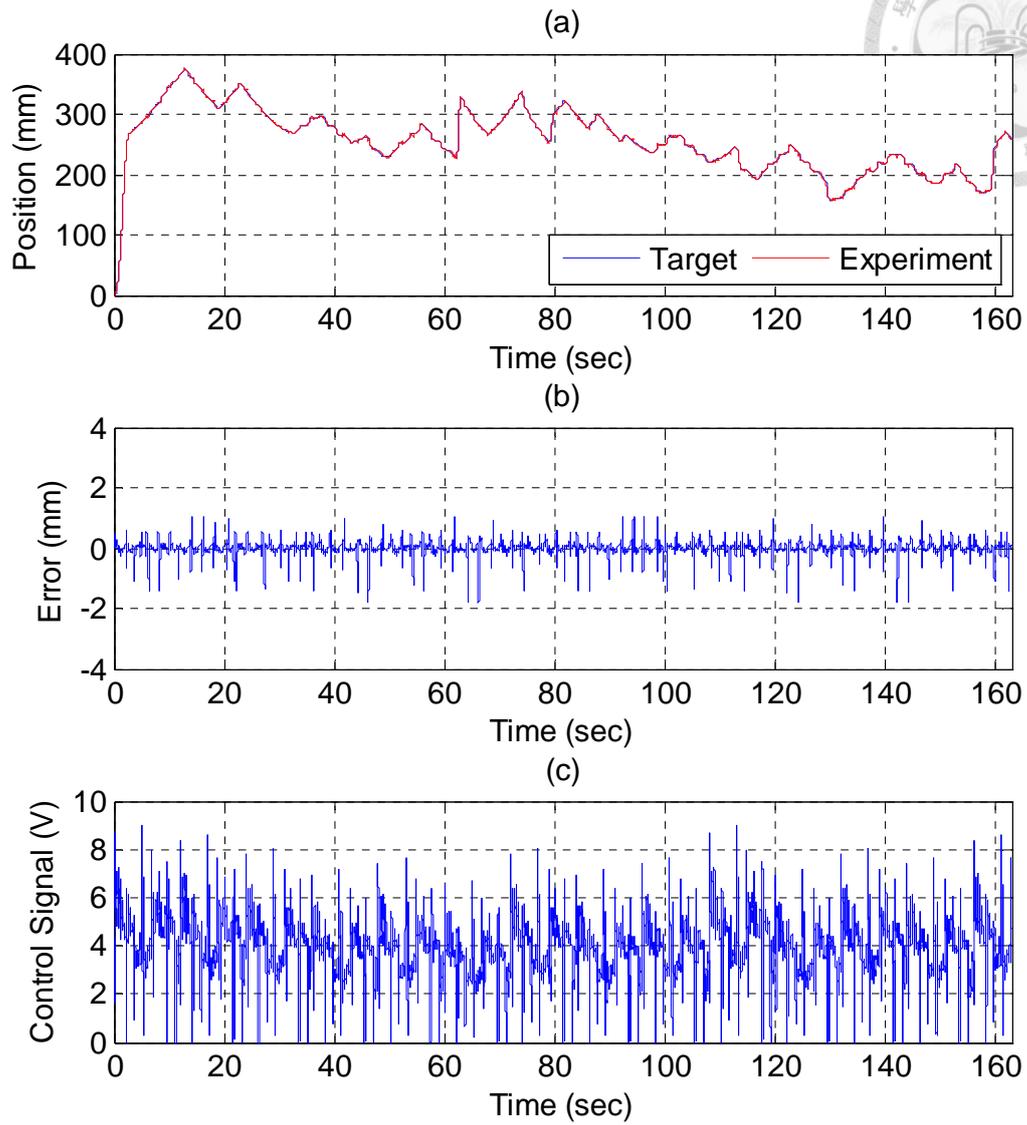


Fig. 6.87 Experimental results of end-effector path tracking control for

C-axis cylinder via a solid polyhedron trajectory:

(a) position tracking response (b) position tracking error (c) control signal

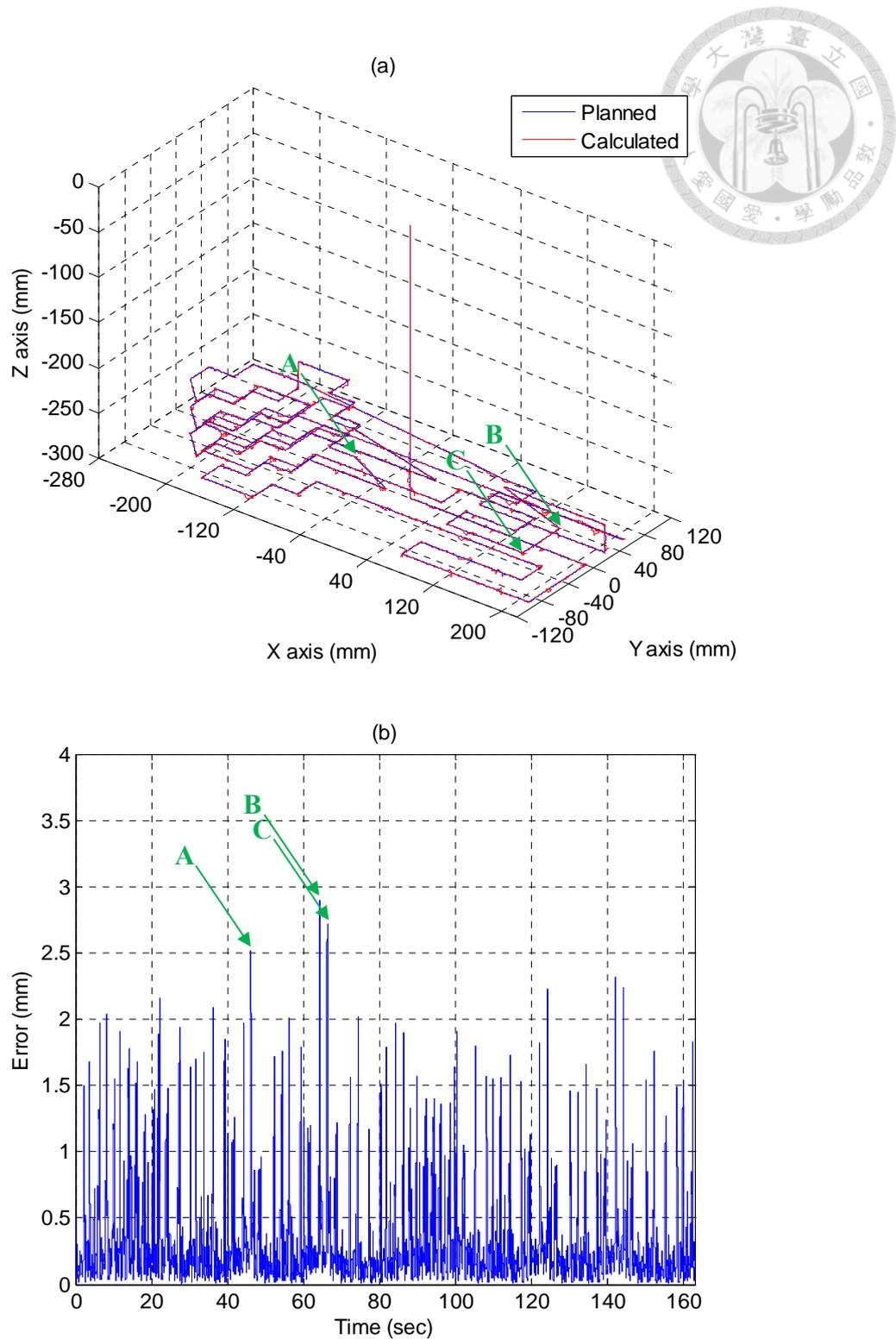


Fig. 6.88 Estimated end-effector position tracking response for a solid polyhedron trajectory: (a) calculated end-effector position
(b) calculated end-effector position error

6.3.5 Experimental Results of Solid Complex 3D Trajectory

The desired printing object of solid complex 3D is shown in Fig. 6.89 and is the same as that in Section 6.1.5 and 6.2.5. The details of end-effector position trajectory and actuated position trajectory have been also discussed in Section 6.1.5.

The experimental results of cylinder response on each axis are shown in Fig. 6.90, Fig. 6.91, and Fig. 6.92. In addition, the estimated end-effector position during the path tracking is calculated by the forward kinematics from the output actuated joint positions of three cylinders and shown in Fig. 6.93(a). The estimated position error of the manipulator end-effector is also calculated from the position error of the actuated joint and shown in Fig. 6.93(b). The top three significant errors during the path tracking control occur at the changing points of different trajectory segments as $t = 122, 460$ and 470 sec. The maximum estimated error of the end-effector is about 2.9 mm at 470 sec.

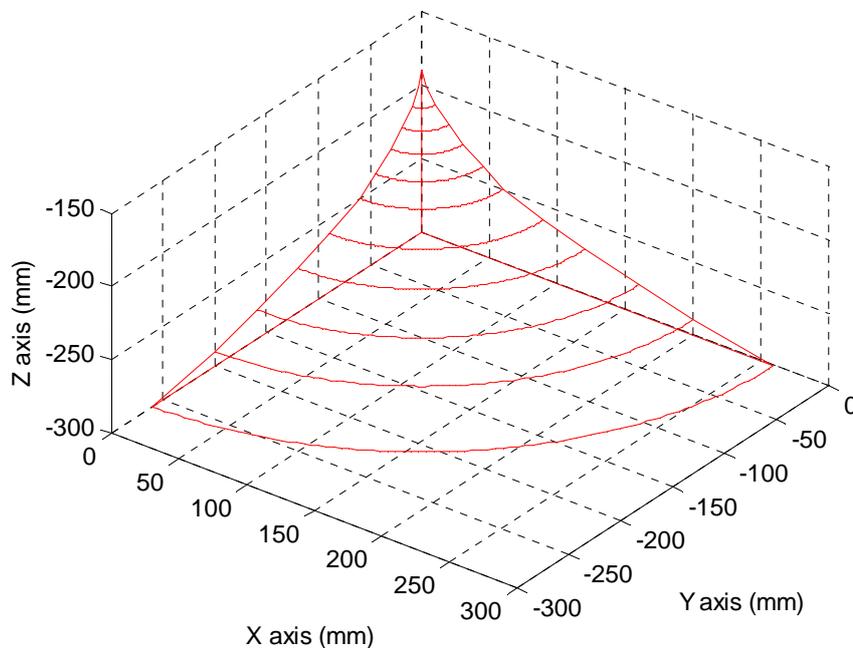


Fig. 6.89 Desired printing object of solid complex 3D

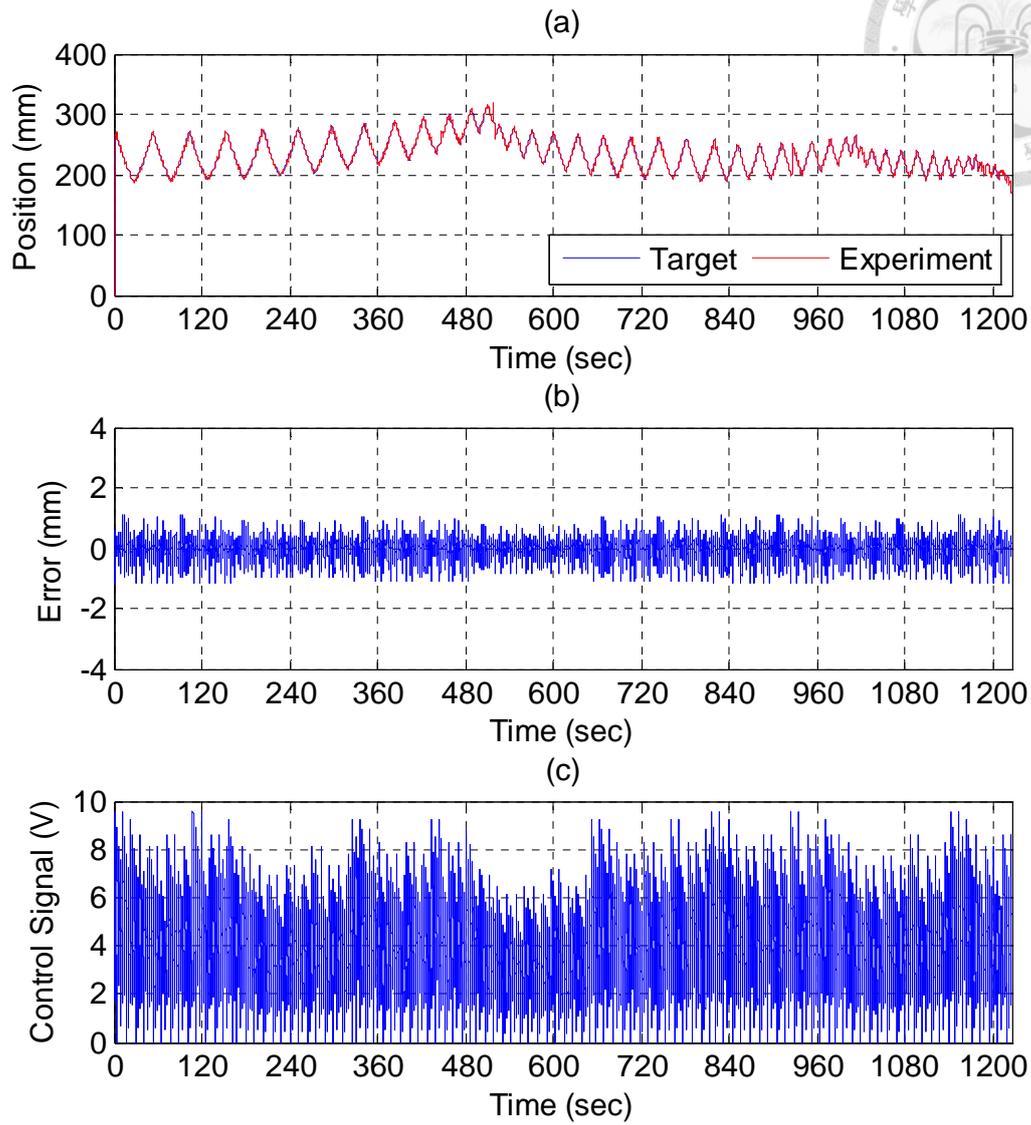


Fig. 6.90 Experimental results of end-effector path tracking control for

A-axis cylinder via a solid complex 3D trajectory:

(a) position tracking response (b) position tracking error (c) control signal

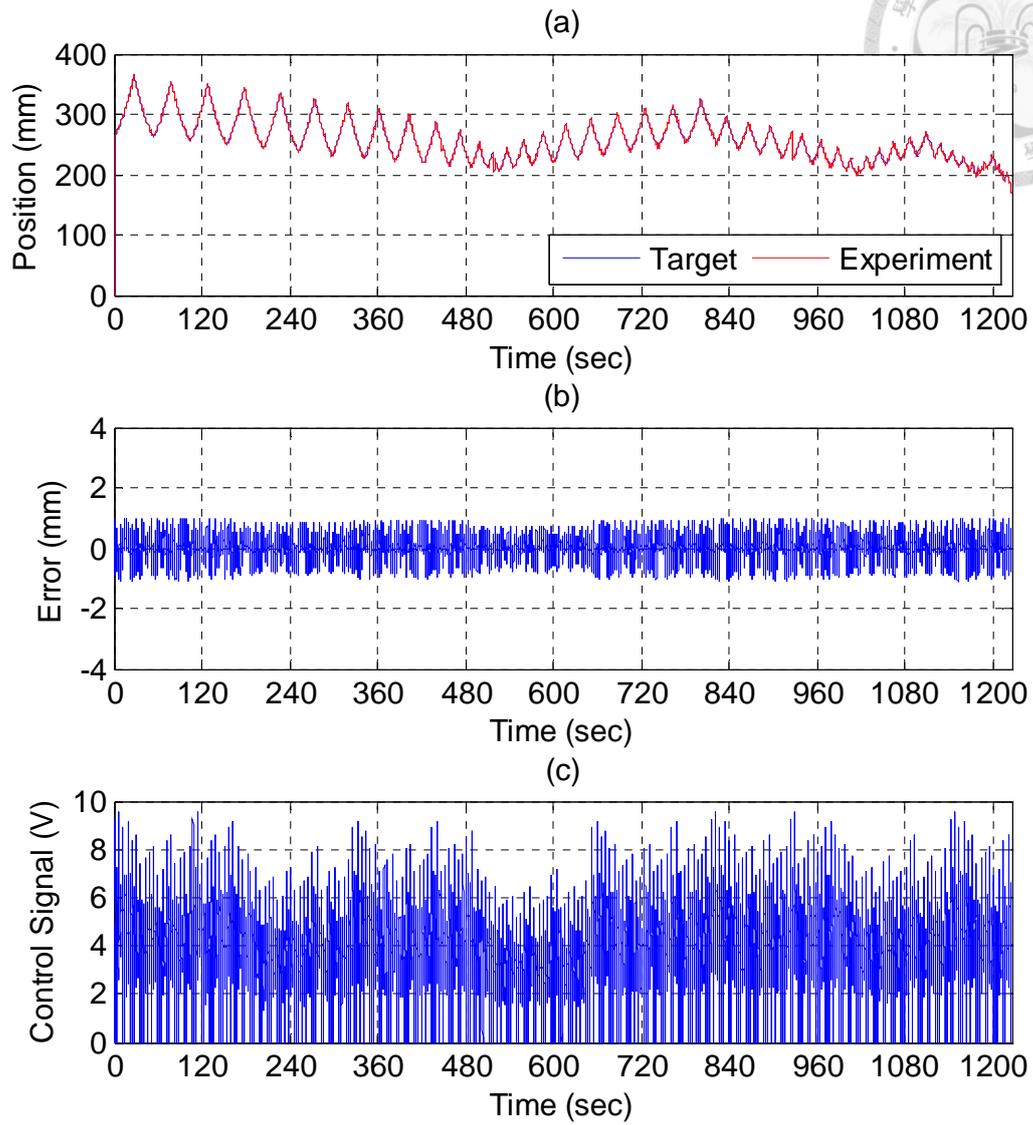


Fig. 6.91 Experimental results of end-effector path tracking control for

B-axis cylinder via a solid complex 3D trajectory:

(a) position tracking response (b) position tracking error (c) control signal

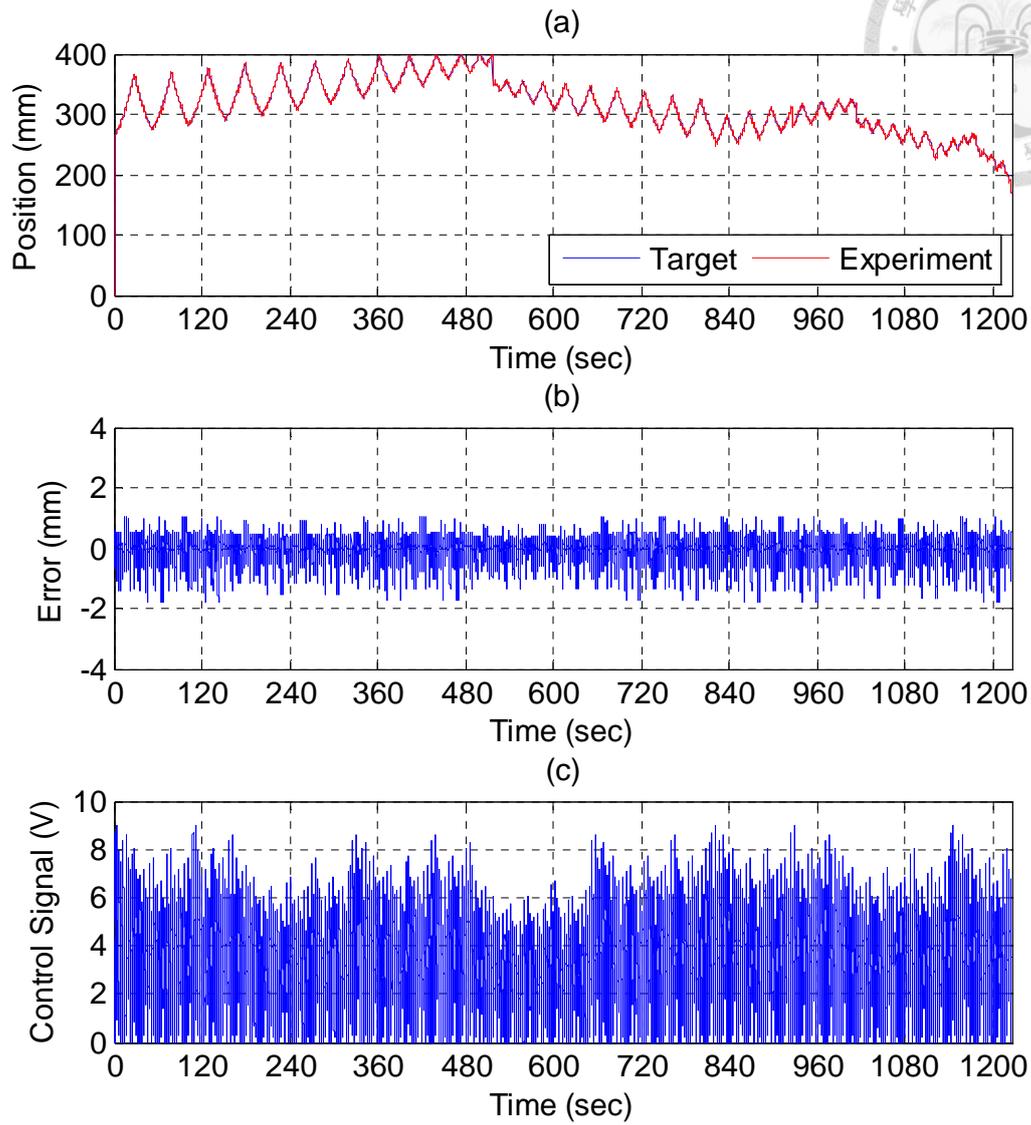


Fig. 6.92 Experimental results of end-effector path tracking control for

C-axis cylinder via a solid complex 3D trajectory:

(a) position tracking response (b) position tracking error (c) control signal

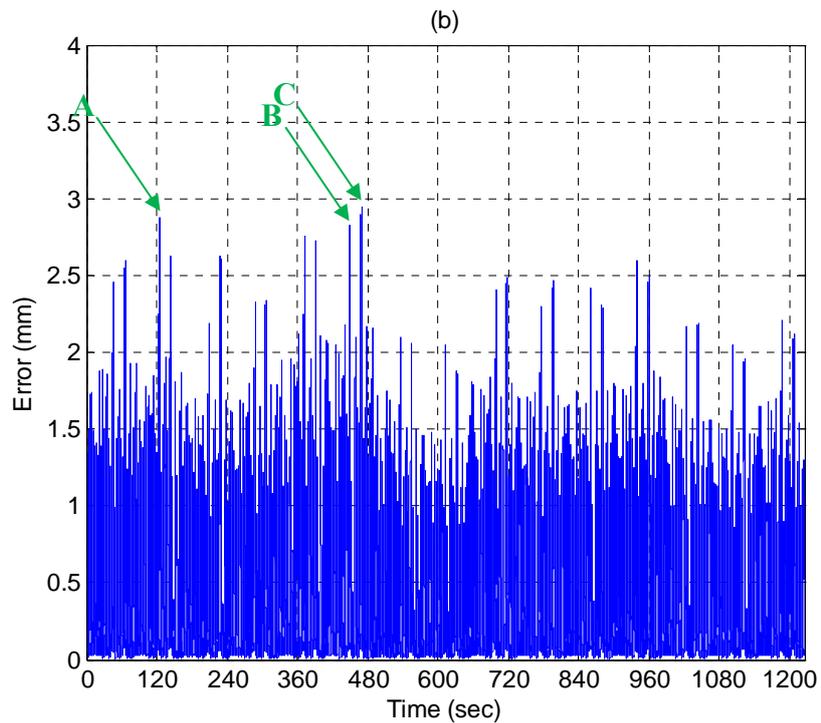
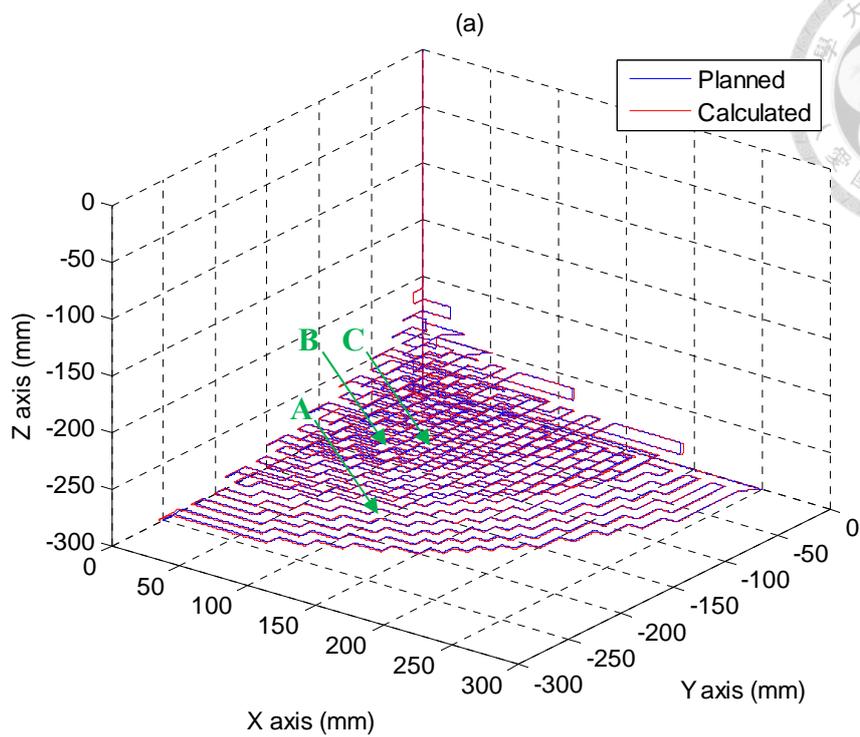


Fig. 6.93 Estimated end-effector position tracking response for a solid complex 3D trajectory: (a) calculated end-effector position (b) calculated end-effector position error

Chapter 7 Conclusions



The goal of this study is to develop efficient path planning algorithms and utilize on three-axial pneumatic parallel manipulator as a 3D printer.

To build desired-printing object model, directed graph is introduced. Before path planning, each point in the same layer has double-directions vectors to link each other and one-direction vectors to link between layers from bottom to top. The modified-DFS solves the case when a group of points has no links to others in the same layer, and the modified-GA makes sure the start point in a layer would not change since algorithms start. Besides, for cascading all layers, the algorithm is to choose the last point of x-y axes of a layer as a start point to next layer, and automatically create this point if not existing. Repeating above methods, the overall path planning of an object is accomplished and each point in the same layer only has one-direction vectors to link each other.

Analyses of kinematics have been implemented to illustrate the kinematic relationship and behavior of manipulator system. Through the inverse and forward kinematics, specific positions of actuator and end-effector can be calculated and transformed mutually. The control scheme of overall manipulator system, developed in AFPCL, contains an inner pressure controller and an outer position controller, called dual-loop feedback control. The inner pressure control loop applies a PI controller with anti-windup function to regulate pressure differences of cylinder, and a PD controller is used to compensate position errors of cylinder piston in the outer position control loop.

The realistic trajectories of end-effector position in simulations and experiments cannot be measured directly, and the alternative is to use the forward kinematics on the actuators to predict the end-effector position, i.e. the numerical-calculation results. Thus,

the purpose of Section 6.1 is to demonstrate the alternative is feasible. After inputting five different end-effector trajectories, the numerical errors between the ideal and the calculated end-effector position are all below $\pm 2.0 \times 10^{-5}$ mm, which is an acceptable result; therefore, this method is used in the simulations and experiments in this study.

In the simulations, PID controllers have good control responses and the errors between the ideal and the calculated end-effector position can be achieved with satisfactory results. Each trajectory only has a little oscillation when entering the low velocity region or occurring at motion direction change of the cylinder.

However, in the experiments, the control responses of dual-loop feedback control is not as good as that in the simulations and the errors between the ideal and the calculated end-effector position are far behind the expectations. Each trajectory has obvious severe oscillation, solid complex 3D trajectory especially. The reason is that the distance between two adjacent points in the same layer is the shortest among these five trajectories. When an actuator tries to move from a point to another point, the system is in the transient response which causes some oscillation at the start and end, but if two points are too close, the system would never being in the steady-state response. That is why the trajectories look always being oscillating. On the other hand, the differential controller in dual-loop feedback control would magnify the noises, which brings the oscillation much more obviously. According to above mechanical limitation, a possible way is to re-think of the control strategy of three-axial pneumatic parallel manipulator.

REFERENCES



- [1] D. Stewart, "A platform with six degrees of freedom," *Proceedings of the institution of mechanical engineers*, vol. 180, pp. 371-386, 1965.
- [2] R. Clavel, "Delta, a fast robot with parallel geometry," in *Proceedings of the International Symposium on Industrial Robots, 1988*, 1988, pp. 91-100.
- [3] F. Pierrot, C. Reynaud, and A. Fournier, "DELTA: a simple and efficient parallel robot," *Robotica*, vol. 8, pp. 105-109, 1990.
- [4] A. Codourey, "Dynamic modeling of parallel robots for computed-torque control implementation," *The International Journal of Robotics Research*, vol. 17, pp. 1325-1336, 1998.
- [5] L.-W. Tsai, G. C. Walsh, and R. E. Stamper, "Kinematics of a novel three DOF translational platform," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, 1996, pp. 3446-3451.
- [6] J. Wang and X.-J. Liu, "Analysis of a novel cylindrical 3-DoF parallel robot," *Robotics and Autonomous Systems*, vol. 42, pp. 31-46, 2003.
- [7] V. Lukanin, "Inverse Kinematics, Forward Kinematics and working space determination of 3DOF parallel manipulator with SPR Joint Structure," *Mechanical Engineering*, vol. 49, pp. 39-61, 2005.
- [8] D. Chablat and P. Wenger, "Architecture optimization of a 3-DOF translational parallel mechanism for machining applications, the orthoglide," *Robotics and Automation, IEEE Transactions on*, vol. 19, pp. 403-410, 2003.
- [9] S. Huda, Y. Takeda, and S. Hanagasaki, "Kinematic design of 3-URU pure rotational parallel mechanism to perform precise motion within a large workspace," *Meccanica*, vol. 46, pp. 89-100, 2011.
- [10] H.-S. Choi, C.-S. Han, K.-y. Lee, and S.-h. Lee, "Development of hybrid robot for construction works with pneumatic actuator," *Automation in Construction*, vol. 14, pp. 452-459, 2005.
- [11] M.-H. Chiang and H.-T. Lin, "Development of a 3D parallel mechanism robot arm with three vertical-axial pneumatic actuators combined with a stereo vision system," *Sensors*, vol. 11, pp. 11476-11494, 2011.
- [12] H.-T. Lin, "Design, analysis and path tracking servo control of 3-PUU pneumatic parallel manipulators with a stereo vision measurement system," Ph.D. dissertation, National Taiwan University, Taipei, Taiwan, 2012.
- [13] M.-H. Chiang, H.-T. Lin, and C.-L. Hou, "Development of a stereo vision measurement system for a 3D three-axial pneumatic parallel mechanism robot arm," *Sensors*, vol. 11, pp. 2257-2281, 2011.
- [14] D. Ben-Dov and S. E. Salcudean, "A force-controlled pneumatic actuator," *Robotics and Automation, IEEE Transactions on*, vol. 11, pp. 906-911, 1995.
- [15] E. Richer and Y. Hurmuzlu, "A high performance pneumatic force actuator system: Part I—Nonlinear mathematical model," *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, pp. 416-425, 1999.
- [16] J. Wang, D. Wang, P. R. Moore, and J. Pu, "Modelling study, analysis and robust servocontrol of pneumatic cylinder actuator systems," *IEE Proceedings-Control Theory and Applications*, vol. 148, pp. 35-42, 2001.
- [17] J. E. Bobrow and B. W. McDonell, "Adaptive tracking control of an air powered

- robot actuator," *Journal of dynamic systems, measurement, and control*, vol. 115, pp. 427-433, 1993.
- [18] J. E. Bobrow and B. W. McDonell, "Modeling, identification, and control of a pneumatically actuated, force controllable robot," *Robotics and Automation, IEEE Transactions on*, vol. 14, pp. 732-742, 1998.
- [19] A. Moran, S. Nakadai, and M. Nagai, "Analysis of computer-controlled pneumatic servo system for robotic applications," in *Intelligent Robots and Systems '90. Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, 1990, pp. 897-902 vol.2.
- [20] S. Liu and J. E. Bobrow, "An analysis of a pneumatic servo system and its application to a computer-controlled robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 110, pp. 228-235, 1988.
- [21] T. Noritsugu and M. Takaiwa, "Robust positioning control of pneumatic servo system with pressure control loop," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, 1995, pp. 2613-2618 vol.3.
- [22] H. K. Lee, G. S. Choi, and G. H. Choi, "A study on tracking position control of pneumatic actuators," *Mechatronics*, vol. 12, pp. 813-831, 2002.
- [23] B. W. McDonell and J. E. Bobrow, "Adaptive tracking control of an air powered robot actuator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 427-433, 1993.
- [24] K. Tanaka, Y. Yamada, T. Satoh, A. Uchibori, and S. Uchikado, "Model reference adaptive control with multi-rate type neural network for electro-pneumatic servo system," in *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, 1999, pp. 1716-1721 vol. 2.
- [25] K. Tanaka, Y. Yamada, A. Shimizu, and S. Shibata, "Multi-rate adaptive pole-placement control for pneumatic servo system with additive external forces," in *Advanced Motion Control, 1996. AMC '96-MIE. Proceedings., 1996 4th International Workshop on*, 1996, pp. 213-218 vol.1.
- [26] L. Baoren, L. Zhuangyun, and X. Yaoming, "Study on adaptive control for a pneumatic position servo system," *Advances in Modelling & Analysis C*, vol. 49, pp. 21-28, 1997.
- [27] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control* vol. 199: Prentice hall New Jersey, 1991.
- [28] E. Richer and Y. Hurmuzlu, "A high performance pneumatic force actuator system: Part II—Nonlinear controller design," *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, pp. 426-434, 1999.
- [29] G. M. Bone and N. Shu, "Experimental comparison of position tracking control algorithms for pneumatic cylinder actuators," *Mechatronics, IEEE/ASME Transactions on*, vol. 12, pp. 557-561, 2007.
- [30] A.-C. Haung and K.-K. Liao, "FAT-based adaptive sliding control for flexible arms: Theory and experiments," *Journal of Sound and Vibration*, vol. 298, pp. 194-205, 11/22/ 2006.
- [31] M.-H. Chiang, L.-W. Lee, C.-F. Kuo, and Y.-N. Chen, "A H^∞ tracking-based adaptive sliding-mode controller for nonlinear pneumatic path tracking systems via a functional approximation approach," *Journal of the Chinese Society of Mechanical Engineers*, vol. 30, pp. 83-94, 2009.
- [32] H. Kodama, "A scheme for three-dimensional display by automatic fabrication of 3three-dimensional model," *IEICE Transactions on Electronics (Japanese Edition)*,

- vol.J64-C, No.4, pp.237-241, 1981.
- [33] H. Kodama, "Automatic method for fabricating a three-dimensional plastic model with photo-hardening polymer," *Review of Scientific Instruments*, Vol. 52, No. 11, pp. 1770-1773, 1981.
- [34] C. W. Hull, "Apparatus for production of three-dimensional objects by stereolithography," U. S. Patent No. 4575330, March 11, 1986
- [35] S. Crump, "Apparatus and method for creating three-dimensional objects," U. S. Patent No. 5121329, October 30, 1992
- [36] S. Crump, "Modeling apparatus for three-dimensional objects," U. S. Patent No. 5340433, August 23, 1994
- [37] E. M. Sachs, J. S. Haggerty, M. J. Cima, and P. A. Williams, "Three-dimensional printing techniques," U. S. Patent No. 5204055, April 20, 1993
- [38] T. Anderson and J. F. Brecht, "Three dimensional printing materials system and method of use," Europe Patent No. 0925169, December 19, 2001
- [39] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, and A. Bowyer, "Reprap - the replicating rapid prototype," *Robotica*, vol. 29, pp. 177-191, 2011.
- [40] J. M. Pearce, C. M. Blair, K. J. Laciak, R. Andrews, A. Nosrat, and I. Zelenika-Zovko, "3-D printing of open source appropriate technologies for self-directed sustainable development," *Journal of Sustainable Development*, vol. 3, pp. 17-29, 2010.
- [41] S. C. Park and B. K. Choi, "Tool-path planning for direction parallel area milling," *Compute Aided Design*, vol. 32, pp. 17-25, 2000.
- [42] R. T. Farouki, T. Koenig, K. A. Tarabanis, J. U. Korein, and J. S. Batchelder, "Path planning with offset curves for layered fabrication processes," *Journal of Manufacturing Systems*, vol. 14, pp. 355-368, 1995.
- [43] H. Wang, P. Janq, and J. A. Stori, "A metric-based approach to two dimensional tool-path optimization for high-speed machining," *Journal of Manufacturing Science and Engineering*, pp. 127-133, 2005.
- [44] J. Balic and M. Korosec, "Intelligent tool path generation for milling of free surfaces using neural networks," *International Journal of Machine Tools & Manufacture*, vol. 42, no. 10, pp. 1171-1179, 2002.
- [45] J. Balic, A. Nestler, and G. Schulz, "Prediction and optimization of cutting conditions using neural networks and genetic algorithm," *Journal of Mechanical Engineering, Assoc. Mech. Eng. Tech. Slovenia*, pp. 192-203, 1999.
- [46] P. K. Wah, K. G. Murty, A. Joneja, and L. C. Chiu, "Toolpath optimization in layered manufacturing," *IEE Trans.*, vol.34, pp. 335-347, 2002.
- [47] L.-W. Tsai and S. Joshi, "Kinematics and optimization of a spatial 3-UPU parallel manipulator," *Journal of Mechanical Design*, vol. 122, p. 439, 2000.
- [48] M. Bouri and R. Clavel, "The Linear Delta: Developments and Applications," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1-8, 2010.
- [49] N. Deo, *Graphs theory with applications to engineering and computer science*, Prentice-Hall, 1974.
- [50] G. L. Alexanderson, "Euler and Königsberg's bridges: a historical view," *Bulletin of the American Mathematical Society*, vol. 43, pp. 567-573, 2006.
- [51] J. Hopcroft and R. Tarjan, "Efficient planarity testing," *Journal of the Association for Computing Machinery*, vol. 21, no. 4, pp. 549-568, 1974.
- [52] J. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1975.
- [53] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators:*

- Wiley-Interscience, 1999.
- [54] K. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA, 2005.
- [55] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*: John Wiley & Sons New York, 2006.

