

**Southern Taiwan University of Science
and Technology**

Graduate School of Electrical Engineering

Ph.D. Dissertation

**FPGA Realization of Forward Kinematics and
Inverse Kinematics for Five-Axis Articulated
Robot Arm**

Graduate Student: Bui Thi Hai Linh

Advisor : Ying-Shieh Kung

July, 2015

i



Acknowledgments

I would like to express my deepest gratitude to all of my teachers in Department of Electrical Engineering, Southern Taiwan University of Technology and Science, especially Prof. Ying-Shieh Kung for his patience and guidance throughout my research since 2009 when I studied the Master degree until now when I have studied the Doctoral degree. His guidance and inspiration have provided an invaluable experience that will help me in my career.

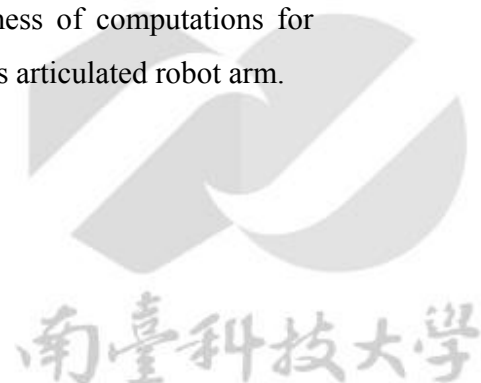
I would also like to thank my lab-mates for their help and advice. They are also the people who make me have unforgettable time and sweet memories in Taiwan.

Finally, I am grateful to my family for their constant love and support, and encouragement.



Abstract

This dissertation presents a study of the forward and inverse kinematics for a five-axis articulated robot arm based on Field Programmable Gate Array (FPGA) technology. Some trigonometric functions using Look-Up Table (LUT) and Taylor series method are used in hardware implementation to speed up of tracking the motion trajectories applied for forward kinematics and inverse kinematics for five-axis articulated robot arm. Firstly, the forward kinematics and inverse kinematics of five-axis articulated robot arm are derived. Secondly, the computations algorithms and its hardware implementation are described. Thirdly, Very high speed integrated circuits Hardware Description Language (VHDL) is applied to describe the overall hardware behavior of forward and inverse kinematics. Additionally, Finite State Machine (FSM) is applied for reducing the hardware resource usage. Further, to verify the correctness of the forward and inverse kinematics for five-axis articulated robot arm, a co-simulation work is constructed by Modelsim and Matlab Simulink. The forward and inverse kinematics hardware is run by Modelsim and a test bench which generates stimulus to Modelsim and displays the output response that is taken in Simulink. Under this design, the combination of the forward and inverse kinematics simulation for tracking the motion trajectories is adopted. Fourthly, the design of forward and inverse kinematics IPs for five-axis robot arm is implemented by a single FPGA. Additionally, a Nios II processor can be embedded into FPGA to construct a System on a Programmable Chip (SoPC) developing environment. Programs in Nios II processor are coded in C language and IPs digital hardware is described by VHDL. The Man-Machine Interface (MMI) developed by Visual Basic language which displays the results of computations kinematics in FPGA into decimal number for easy checking the correctness of results. Therefore, the digital hardware/software co-design based on the SoPC is suitable for the development of the forward and inverse kinematics for five-axis articulated robot arm. Finally, an experiment system has been built up as well as some experimental results have been demonstrated to verify the effectiveness and correctness of computations for forward and inverse kinematic which is applied to the real five-axis articulated robot arm.



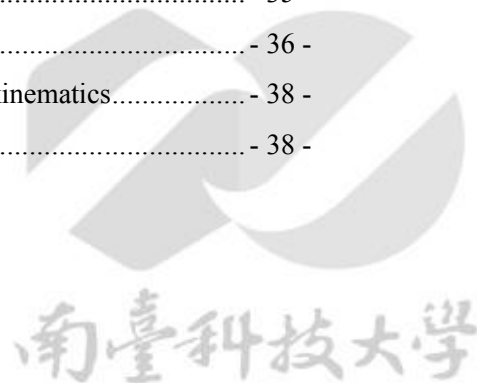
摘要

本文基於 FPGA（現場可程式邏輯閘陣列）技術提出了前向和逆向運動學的五軸模組化機械臂之研究。首先，推導五軸關節型機械手臂的前向運動學和逆向運動學。其次，針對演算法和硬體實現進行了描述。第三，超高速積體電路硬體描述語言（VHDL）被用於描述前向和反向運動學的整體硬體行為。此外，運用有限狀態機器（FSM）以減少硬體資源的使用。為了驗證五軸關節型機械手臂的前向和逆向運動學的正確性，將結合 Modelsim 和 Matlab Simulink 進行模擬。前向和逆向運動學的硬體是由 Modelsim 執行，而 Modelsim 測試平台產生的輸入訊號及輸出響應將顯示在 Simulink 中。根據這樣的設計，前向及逆向運動學及運動軌跡追蹤可以在數微秒內完成。第四，五軸機械臂的前向和逆向運動學 IP 設計將由單顆 FPGA 實現。此外，Nios II 處理器可以嵌入到 FPGA 中以建構 SoPC（在系統可編程片）開發環境。在 Nios II 處理器的應用程式以 C 語言撰寫，而 VHDL 將用於描述前向和逆向運動學的數位硬體電路。另外，本論文也利用 visual basic 開發一套人機介面(MMI)，此將呈現前向和逆向運動學 IP 計算後的結果。因此，基於所述，SoPC 將適合發展五軸機械手臂的前向和逆向運動學的硬體/軟體共同設計環境。最後，將建立一個實驗系統並有實驗結果來證實應用於五軸模組化機械手臂的前向及逆向運動學計算的有效性和正確性。



Table of Content

Acknowledgments	i
Abstract	iii
摘要	iv
Table of Content	v
List of Figures.....	viii
List of Tables.....	xi
List of Symbols.....	xii
Chapter 1 Introduction	- 1 -
1.1 Research background & literature survey.....	- 1 -
1.2 The motivation of the study	- 6 -
1.3 The structure of thesis	- 7 -
Chapter 2 Mathematical description of kinematics and motion trajectories planning	- 8 -
2.1 Introduction of five-axis articulated robot arm	- 8 -
2.2 Review of kinematics	- 10 -
2.2.1 Rotating coordinate system	- 12 -
2.2.2 Homogeneous coordinates	- 13 -
2.2.3 Coordinates architecture	- 15 -
2.2.4 Rotary joint coordinates architecture	- 16 -
2.3 Robot kinematics of five-axis articulated robot arm	- 18 -
2.3.1 Forward kinematics	- 20 -
2.3.2 Inverse kinematics	- 24 -
2.4 The computation of point-to-point motion control	- 29 -
2.4.1 Five axes trajectory planning	- 32 -
2.4.2 The formulas of motion trajectories.....	- 33 -
2.4.2.1 Linear motion trajectory.....	- 34 -
2.4.2.2 Circular motion trajectory	- 34 -
2.4.2.3 Star motion trajectory.....	- 35 -
2.4.2.4 Window motion trajectory.....	- 36 -
Chapter 3 Hardware implementation of forward kinematics and inverse kinematics.....	- 38 -
3.1 Introduction and literature review	- 38 -



3.2 Review of VHDL and Q-format design.....	- 42 -
3.3 An example of Sum of Product.....	- 44 -
3.4 Trigonometric functions	- 49 -
3.4.1 Computation algorithm of <i>Sine</i> and <i>Cosine</i> functions	- 49 -
3.4.2 Computation algorithm of <i>Arctangent</i> function using Taylor series expansion.....	- 50 -
3.4.3 Computation of <i>Arccosine</i> function using Taylor series expansion method.....	- 54 -
3.5 Design of hardware implementation for forward kinematics and inverse kinematics.....	- 57 -
3.5.1 Forward kinematics and inverse kinematics design in VHDL using Q-format.....	- 57 -
3.5.2 FSM for forward kinematics and inverse kinematics	- 58 -
Chapter 4 Modelsim/Simulink co-simulation of forward/inverse kinematics for five-axis articulated robot arm	- 64 -
4.1 Introduction of Modelsim/Simulink co-simulation	- 64 -
4.2 Co-Simulation cases using Modelsim/ Simulink	- 68 -
4.2.1 Sum of Product simulation results.....	- 69 -
4.2.2 Sine and cosine functions co-simulation results.....	- 70 -
4.2.3 Arctangent and arccosine functions co-simulation results.....	- 73 -
4.3 Modelsim/Simulink co-simulation of forward/inverse kinematics	- 75 -
4.4 Simulation results in Modelsim/Simulink of tracking motion trajectories	- 80 -
4.4.1 Linear motion trajectory.....	- 81 -
4.4.2 Circular motion trajectory	- 83 -
4.4.3 Star motion trajectory	- 83 -
4.4.4 Window motion trajectory	- 86 -
Chapter 5 FPGA realization of forward/inverse kinematics for five-axis articulated robot arm..	- 89 -
5.1 Introduction.....	- 89 -
5.2 Description of SoPC builder design	- 90 -
5.2.1 DE2 115 board.....	- 90 -
5.2.2 Nios II embedded processor.....	- 93 -
5.3 FPGA implementation of forward kinematics and inverse kinematics	- 96 -
5.4 Applying to real robot arm.....	- 99 -
5.4.1 Hardware implementation system	- 101 -
5.4.1.1 CAN bus interface.....	- 101 -
5.4.1.2 Overall hardware system	- 103 -
5.4.2 Experimental results	- 105 -



5.4.2.1 Linear motion trajectory.....	- 106 -
5.4.2.2 Circular motion trajectory.....	- 107 -
5.4.2.3 Star motion trajectory.....	- 107 -
5.4.2.4 Window motion trajectory.....	- 108 -
Chapter 6 Conclusion and future works.....	- 113 -
6.1 Conclusion.....	- 113 -
6.2 Future works.....	- 114 -
References.....	- 115 -
Biography.....	- 123 -
Academic Publications.....	- 124 -



List of Figures

Figure 2.1 Electrical · Rotary Actuators · Universal Rotary Actuators	- 9 -
Figure 2.2 The sectional diagram	- 9 -
Figure 2.3 The five-axis articulated robot arm	- 10 -
Figure 2.4 The definition of standard Denavit-Hartenberg link parameters [37]	- 11 -
Figure 2.5 The end-effector	- 13 -
Figure 2.6 The coordinate architecture	- 16 -
Figure 2.7 The location of three-vectors n, o, a	- 16 -
Figure 2.8 Robot coordinates indicate	- 17 -
Figure 2.9 The relationship coordinates between two joints	- 17 -
Figure 2.10 The schematic representation of forward and inverse kinematics	- 19 -
Figure 2.11 The link coordinate system of a five-axis articulated robot arm (general)	- 19 -
Figure 2.12 The link coordinate system of a five-axis articulated robot arm (details)	- 20 -
Figure 2.13 The base and first link coordinate schematic	- 21 -
Figure 2.14 The first link and the second link coordinate schematic	- 21 -
Figure 2.15 The second link and the third link coordinates schematic	- 22 -
Figure 2.16 The third link and the fourth link coordinates schematic	- 22 -
Figure 2.17 The fifth link and the fourth link coordinates schematic	- 23 -
Figure 2.18 LFPB trajectory (a) velocity; (b) acceleration	- 31 -
Figure 2.19 Minimum-time trajectory (a) velocity; (b) acceleration	- 32 -
Figure 2.20 Circular motion trajectory tracking	- 34 -
Figure 2.21 Star motion trajectory tracking	- 36 -
Figure 2.22 Window motion trajectory tracking	- 37 -
Figure 3.1 Parallel processing using three multipliers and two adders execute by one step	- 47 -
Figure 3.2 Sequential processing using one multiplier and one adder execute by five step	- 47 -
Figure 3.3 VHDL code for computing the sum of product	- 48 -
Figure 3.4 Three cases for computing the sum of product	- 48 -
Figure 3.5 FSM for computing the <i>sine</i> function	- 50 -
Figure 3.6 FSM for computing the <i>cosine</i> function	- 50 -
Figure 3.7 Compute $\theta = a \tan 2(y/x)$ of each region in X-Y coordinates	- 53 -
Figure 3.8 The diagram to compute $a \tan 2(y/x)$ function	- 54 -
Figure 3.9 FSM to compute $\tan^{-1}(x)$ function	- 54 -
Figure 3.10 Computing range of $\theta = \cos^{-1}(x)$	- 55 -
Figure 3.11 Hardware implementation diagram for <i>arccosine</i> function	- 56 -
Figure 3.12 FSM for computing the $\cos^{-1}(x)$ function (range from 45° to 90°)	- 57 -
Figure 3.13 Block diagram for (a) forward kinematics and (b) inverse kinematics	- 58 -
Figure 3.14 FSM for computing the Forward kinematics	- 61 -
Figure 3.15 FSM for computing the Inverse kinematics	- 62 -

Figure 3.16 Forward kinematics computations time in FPGA	- 63 -
Figure 3.17 Inverse kinematics computations time in FPGA	- 63 -
Figure 4.1 Project flow.....	- 67 -
Figure 4.2 Simulation flow working library.....	- 68 -
Figure 4.3 Modelsim co-simulation of Sum of Product with 16bit Q0.....	- 69 -
Figure 4.4 Modelsim co-simulation of Sum of Product with 16bit Q15	- 70 -
Figure 4.5 Modelsim co-simulation of Sum of Product with 16bit Q24.....	- 70 -
Figure 4.6 The co-simulation in Modelsim/Simulink of <i>Sine</i> function.....	- 72 -
Figure 4.7 The co-simulation in Modelsim/Simulink of <i>Cosine</i> function.....	- 72 -
Figure 4.8 The co-simulation in Modelsim/Simulink of <i>Arctangent</i> function	- 74 -
Figure 4.9 The co-simulation in Modelsim/Simulink of <i>Arccosine</i> function.....	- 74 -
Figure 4.10 Setting complier	- 76 -
Figure 4.11 Setting the “220pack.vhd” and “220model.vhd” compile to library “lpm”	- 76 -
Figure 4.12 Compiled all files successfully and run Modelsim.....	- 77 -
Figure 4.13 Co-simulation architecture of forward kinematics using Modelsim/Simulink	- 77 -
Figure 4.14 Co-simulation architecture of inverse kinematics using Modelsim/Simulink	- 78 -
Figure 4.15 Forward kinematics and Inverse kinematics co-simulation in Modelsim/Simulink..	- 81 -
Figure 4.16 The results of linear motion trajectory co-simulation in Modelsim/Simulink	- 82 -
Figure 4.17 Error of linear motion trajectory co-simulation in Modelsim/Simulink	- 82 -
Figure 4.18 The results of circular motion trajectory co-simulation in Modelsim/Simulink	- 84 -
Figure 4.19 Error of circular motion trajectory co-simulation in Modelsim/Simulink.....	- 84 -
Figure 4.20 The results of star motion trajectory co-simulation in Modelsim/Simulink	- 85 -
Figure 4.21 Error of star motion trajectory co-simulation in Modelsim/Simulink	- 85 -
Figure 4.22 The results of window motion trajectory simulation in Modelsim/Simulink	- 86 -
Figure 4.23 Error of window motion trajectory simulation in Modelsim/Simulink	- 87 -
Figure 5.1 The architecture system of hardware implementation.....	- 90 -
Figure 5.2 The DE2-115 board (top view) [76].....	- 92 -
Figure 5.3 The Nios embedded processor	- 93 -
Figure 5.4 The Nios Development Tool Flow.....	- 94 -
Figure 5.5 A Nios II system implemented on the DE2 board [76]	- 95 -
Figure 5.6 Embedded components information flow.....	- 96 -
Figure 5.7 Architecture of forward kinematics IP and inverse kinematics IP base on FPGA.....	- 97 -
Figure 5.8 Design of forward kinematics IP, inverse kinematics IP and Nios II processor	- 98 -
Figure 5.9 Nios II IDE environments.....	- 98 -
Figure 5.10 The man-machine interface programmed in Visual Basic.....	- 99 -
Figure 5.11 Five-axis articulated robot arm by PowerCube.....	- 100 -
Figure 5.12 CAN-bus adaptor	- 101 -
Figure 5.13 Block-circuit diagram of CAN-USB-Mini module.....	- 102 -
Figure 5.14 Physical Connection for CAN bus [77].....	- 103 -
Figure 5.15 Block of system architecture.....	- 104 -
Figure 5.16 Five-axis articulated robot arm and MMI.....	- 105 -

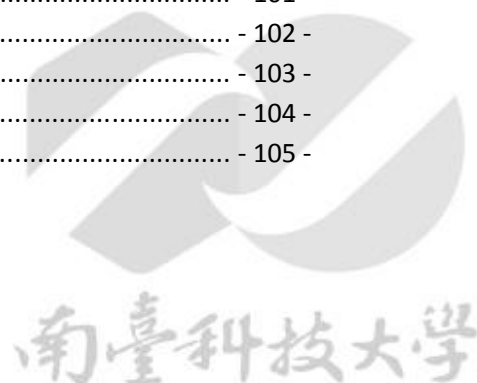


Figure 5.17 The feedback of window motion trajectory on MMI	- 105 -
Figure 5.18 The tracking response of linear motion trajectory by PowerCube robot arm	- 106 -
Figure 5.19 Tracking error of linear motion trajectory	- 107 -
Figure 5.20 The tracking response of circular motion trajectory by PowerCube robot arm	- 108 -
Figure 5.21 Tracking error of circular motion trajectory	- 109 -
Figure 5.22 The tracking response of star motion trajectory by PowerCube robot arm	- 109 -
Figure 5.23 Tracking error of star motion trajectory	- 110 -
Figure 5.24 The tracking response of window motion trajectory by PowerCube robot arm.....	- 110 -
Figure 5.25 Tracking error of window motion trajectory.....	- 111 -

