國立臺灣大學工學院機械工程學研究所

碩士論文

Department of Mechanical Engineering

College of Engineering

National Taiwan University

Master's Thesis

利用前饋神經網路與數位雙生學習增強機器手臂的運動學校正演算法

Enhancing Robotic Arm Kinematic Calibration Algorithm with Feedforward Neural Network and Digital Twin Learning

詹妹臻

Mei-Chen Chan

指導教授: 李宇修 博士

Advisor: Yu-Hsiu Lee Ph.D.

中華民國 113 年 7 月

July, 2024

doi:10.6342/NTU202402735

摘要

本論文採用了一種創新的方法來解決機械手臂精度控制中的挑戰。通過結合雅可比矩陣 (Jacobian matrix) 校正和前饋神經網路 (Feedforward neural network, FNN)，我們提出了一個新型的平行架構模型，旨在提高機械手臂在現實世界中的運動學準確性。傳統的雅可比矩陣校正方法在捕捉機械手臂系統複雜性方面存在局限性，因此我們引入了數位雙生 (Digital twin) 技術和轉移學習 (Transfer learning) 的理念，以優化模型。透過這種平行架構，我們旨在識別出最貼近實際機械手臂行為的模型，從而提高精度控制，以滿足工業和醫療應用對精準度不斷增長的需求。

這一研究的貢獻在於，它不僅修正了傳統運動學模型的缺陷，還引入了數位雙生和轉移學習的思想，使得模擬環境中的演算法能夠直接應用到實際機械手臂的問題中。這不僅提高了機械手臂在現實應用中的準確性和效率，也拓展了機械手臂技術在工業和醫療領域的應用範疇。透過本研究提出的方法，可以更好地應對機械手臂精度控制面臨的挑戰，為相關領域的發展提供新的思路和解決方案。

關鍵字：機械手臂、運動學校正、前饋神經網路、數位雙生、轉移學習

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

Abstract

This paper adopts an innovative approach to address challenges in precision control of robotic arms. By combining Jacobian matrix correction and feed-forward neural networks, we propose a novel parallel architecture model aimed at enhancing the kinematic accuracy of robotic arms in real-world scenarios. Traditional Jacobian matrix correction methods have limitations in capturing the complexity of robotic arm systems; therefore, we introduce the concept of digital twin technology and transfer learning to optimize the model. Through this parallel architecture, we aim to identify models that closely mimic actual robotic arm behavior, thereby improving precision control to meet the increasing demands in industrial and medical applications.

The contribution of this research lies in its dual approach: correcting the shortcomings of traditional kinematic models and integrating the concepts of digital twins and transfer learning. This allows algorithms developed in simulation environments to be directly applied to real-world robotic arm challenges. Consequently, this approach not only enhances the accuracy and efficiency of robotic arms in practical applications but also broadens the scope of robotic arm technology in industrial and medical fields. The method proposed in this study provides a robust solution to the challenges in precision control of robotic arms and offers new perspectives and solutions for the advancement of related fields.

Keywords: Robotic arm, Kinematic calibration, Feedforward neural network, Digital twin, Transfer learning

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

目次

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

圖次

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

表次

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

符號列表

GE	幾何誤差 (Geometric error)
NGE	非幾何誤差 (Non-geometric error)
FK	順向運動學 (Forward kinematics)
FNN	前饋神經網路 (Feed forward network)
ANN	人工神經網路 (Artificial neural network)
θ_i、α_i、β_i	以 z_{i-1}、x_i、y_i 為軸的旋轉量
d_i、a_i	以 z_{i-1}、x_i 為軸的平移量
$\{F\}$	末端執行器座標系
$\{0\}$	機器人的世界座標系
${}^{i-1}_{i}T$	齊次轉換矩陣
$\vec{\theta}$	由各個關節輸入角度值組成之向量
M	物理模型手臂模型，在本論文即 Simulink 建模之物理手臂。

doi:10.6342/NTU202402735

$\vec{P}$	物理模型手臂末端姿態
M_{FK}	標稱手臂運動學模型
$\vec{P}_{FK}$	標稱手臂運動學模型末端姿態
$\vec{P}^*_{FK}$	更新後的手臂運動學模型末端姿態
M	手臂物理模型
$\Delta\vec{P}$	物理手臂末端姿態 $\vec{P}$ 與標稱手臂運動學模型末端姿態 $\vec{P}_{FK}$ 的誤差，由 $[\Delta x\ \Delta y\ \Delta z\ \delta x\ \delta y\ \delta z]^T$ 組成的向量
$\Delta x\ \Delta y\ \Delta z$	位置的誤差
$\delta x\ \delta y\ \delta z$	旋轉的誤差
J	雅可比矩陣
$\vec{q}$	由運動參數 θ、d、a、α、β 組成的向量
$\vec{\theta}$	由各個關節輸入角度值組成之向量
$\Delta\vec{P}_{NGE}$	非線性誤差，由 $[\Delta x\ \Delta y\ \Delta z\ \delta x\ \delta y\ \delta z]^T$ 組成的向量。
$\Delta\vec{P}_{pred}$	FNN 模型的輸出，由 $[\Delta x\ \Delta y\ \Delta z\ \delta x\ \delta y\ \delta z]^T$ 組成的向量，即預測的非幾何誤差。
$\vec{P}^*$	平行架構模型輸出，由 $[\Delta x\ \Delta y\ \Delta z\ \delta x\ \delta y\ \delta z]^T$ 組成的向量
$\Delta\vec{P}_{min}$	物理手臂末端姿態 $\vec{P}$ 與平行架構模型輸出 $\vec{P}^*$ 的差值，即為最小化的誤差，由 $[\Delta x\ \Delta y\ \Delta z\ \delta x\ \delta y\ \delta z]^T$ 組成的向量，同時為本論文之最終目標

doi:10.6342/NTU202402735

第一章　緒論

1.1　研究背景與動機

隨著工業 4.0 的發展，自動化產業越來越不可缺少機器手臂。這些機器手臂在生產線上的物料處理、組裝作業以及品質檢測中扮演著重要角色，它們不僅僅是提高生產力的工具，還能夠促進智能生產的實現 [1, 2]，並且廣泛應用於製造、組裝高精度的產品 [3]。除了精密製造以外，機器手臂也用於執行醫療手術，像是達文西手術系統 [4, 5]，能夠實現高精度、微創性的手術，對於複雜手術的執行提供了極大的幫助，使得手術風險大幅降低，效果更為可靠。然而，隨著對產品精度和醫療手術準確性的要求不斷提高，機器手臂在工業製造、測量儀器、居家照顧和醫療設備等多種領域面臨的挑戰也日益嚴峻。

機器手臂的定位精度受到零件尺寸誤差及裝配誤差的影響 [6]，尺寸誤差主要是源自於製造工序的公差，而裝配誤差則與導引定位的設計和裝配手法有關。尺寸誤差與裝配誤差造成標稱的運動學參數與實際設計尺寸有所出入，因此機器手臂末端位置的實際位置會與理想的預測位置有所不同。上述的問題主要屬於機器手臂的幾何誤差 (Geometric error, GE)[7]，可以透過運動學參數校正獲得解決。

不過在精密定位的製造或加工場合，還需考慮機器手臂的非幾何誤差 (Non-geometric error, NGE)[8]，例如機械撓性、背隙或皮帶滑差等，機器手臂的桿

doi:10.6342/NTU202402735

件並非完美的剛體，其運動時會因姿態與慣性力的影響發生變形。在靜態定位的範疇中，傳統校正方法主要假設零件轉矩撓性變化方向與旋轉接頭方向相同，透過虎克定律的線性模型假設可有效降低定位誤差。然而零件並非只有在旋轉接頭的方向具有撓性，在並聯與串並複合的結構下，現有演算法的衍伸不易進行直覺的撓性補償。

因此，為了改善機器手臂面臨的裝配誤差和機構非線性變化等問題，本論文旨在探討有效的校正和調整方法。首先，透過機器手臂的 DH 參數 (Denavit–Hartenberg parameters, DH) 校正對其順向運動學 (Forward kinematics, FK) 進行初步幾何補償。接著，應用機器學習技術來補償非線性誤差，藉此識別影響非線性變形的關鍵因素。由於機器學習需要大量數據，因此本論文引入數位雙生 (Digital twin) 模型進行訓練，並藉此識別模型網路的關鍵分支，從而進行學習轉移。在進行轉移學習之前，使用剪枝後的模型，以少量數據在真實系統上快速學習非幾何補償。通過這種方法，本研究期望為機器手臂的控制和調整提供新的思路和解決方案，以應對現代醫療和製造的挑戰。

1.2 文獻回顧

文獻回顧主要會分成三大部分，分別介紹傳統校正的方式、以及利用類神經網路的校正法，並說明如何從模擬到現實，將這些校正法在真實系統上使用，以縮小模擬和現實間的差距。

1.2.1 傳統校正法

傳統校正方法在機器手臂校正中起著基礎性作用，主要包括基於運動學參數的校正，這些方法通常依賴於機器手臂的幾何結構和運動學模型來進行校正。例

doi:10.6342/NTU202402735

如，1985 年 S. Hayati 等人提出的機器人幾何校準方法採用了一種創新的算法 [9]。該算法首先基於機器人的順向運動學模型，透過測量得到的實驗數據，利用最小平方法或者其他方式求解，來調整順向運動學的模型參數。這些參數包括關節長度、關節角度、和連接桿件的長度等。通過迭代的過程，該算法將不斷優化模型以最大程度地符合實際觀測到的機器人運動軌跡。最終，這個校正過程將使機器人的運動模型更準確，從而提高其運動精度和可靠性。這種方法具有廣泛的適用性，可應用於各種能夠類型的機器人，像是工業機器手臂、平行架構機器手臂 (Parallel robotic arm)。

1987 年，W Veitschegger 和 Chi-haur Wu 提出了兩種機器人運動學誤差補償算法 [10]。第一種方法假設在期望位置 $\mathrm{POS^d}$ 附近，且微分誤差變換不隨關節變量的小變化而顯著改變，先計算無誤差的實際位置 $\mathrm{POS^c}$，最後透過 $\mathrm{dPOS} = \mathrm{POS^c} - \mathrm{POS^d}$ 補償關節變量誤差。第二種則是使用 Newton-Raphson 法 [11]，先建立運動學的誤差方程 $\mathrm{POS^d} - \mathrm{POS^c} = \frac{\partial \mathrm{POS^c}}{\partial \theta_1}\Delta\theta_1 + \cdots + \frac{\partial \mathrm{POS^c}}{\partial \theta_6}\Delta\theta_6$，通過迭代計算修正量來更新關節變量，逐步減少位置誤差，最終達到期望的操作空間位置。

2015 年，A. Joubair 等人更是提出了如何將非幾何參數結合到傳統建模方式 [12]，主要針對影響機器人精度的非幾何因素進行校正，此誤差模型考慮了機器人關節齒輪箱的剛性，建立了一個靜態模式下的剛性模型，主要考慮連桿和末端執行器的重力影響，忽略了扭轉、拉伸和壓縮效應。該模型將每個關節齒輪箱 i 視為一個具有單一順應性參數 λ_i(compliance parameter) 的線性扭轉彈簧。考慮重力 f_j^i 與桿件重心 c_j^i，計算每個關節所受到的力矩 $\tau_{i,linkj} = f_j^i c_j^i$，並根據轉矩與轉角偏移的線性撓性模型計算偏差量 $\delta\theta_i = \lambda_i \tau_i$。通過將這些力矩參數納入運動學模型進行校正，能夠更準確地反映實際操作中的非線性變形。這種方法有效地提高了機器人的定位精度和運動穩定性。

doi:10.6342/NTU202402735

傳統校正方法的優點在於其理論基礎扎實，實施起來相對簡單，且在已知系統參數的情況下能夠達到較高的校正精度。然而，這些方法在處理非線性和複雜環境變化時存在一定的局限性，導致校正的準確性和速度有所不足

1.2.2 類神經網路的分類

在機器人運動學校正的背景下，傳統的運動學模型往往難以完全捕捉機械結構的變異、環境的變化以及製造過程中的誤差對系統的影響。為了提高手臂模型的準確性，引入神經網路可以從實際測量數據中學習。根據 Hornik 等人在 1989 年的研究指出 [13]，神經網路具有強大的函數近似能力，能有效處理複雜的非線性問題。因此這對於解決機器人手臂運動學模型中的非線性誤差具有顯著幫助。

在 Batta Mahesh 的《Machine Learning Algorithms - A Review》[14] 中，作者系統地回顧了不同的機器學習算法，討論了各種算法的基本原理、優缺點及其在不同領域的應用。以下是關於神經網路部分的重點摘要，類神經網路可以根據其學習策略和網路架構進行分類，這有助於理解和選擇適合不同應用的網路結構和算法。

根據學習策略分類

- **監督式學習網路 (Supervised learning network)**：監督式學習利用帶標籤的數據訓練模型，目標是學習一個將輸入映射到輸出的函數，以預測或估計輸出值。這類方法廣泛應用於圖像分類、語音識別和自然語言處理等領域，通常需要大量標註數據以實現良好的性能。

- **無監督式學習網路 (Unsupervised learning network)**：在沒有標籤的情況下，通過對數據的統計特性進行建模，發現數據中的潛在結構和規律。例如，聚

doi:10.6342/NTU202402735

類算法和自動編碼器就屬於無監督式學習的範疇。無監督式學習可用於數據降維和特徵提取等任務。

- **半監督式學習網路 (Semi-supervised learning network)**：結合了帶標籤和無標籤的數據進行訓練。這種方法在數據量較少但標籤費用較高的情況下很有用，例如在醫療診斷和金融預測中。半監督式學習通常利用未標註數據的信息來改善模型的性能。

- **強化學習網路 (Reinforcement learning network)**：通過與環境的交互來學習最優策略，以最大化期望的累積回報。這類網路在遊戲玩法、機器人控制和自動化交易等領域有廣泛應用。強化學習的目標是通過與環境的交互來學習最優的動作策略。

根據網路架構分類

- **前饋神經網路 (Feedforward neural network, FNN)**[15, 16]：架構如 Figure1.1表示，信息流從輸入層到輸出層，沒有循環反饋。這種網路結構適合用於靜態模式識別，如圖像分類和數據分類。前饋神經網路的層之間沒有回饋連接，每一層的輸出都是下一層的輸入。這種結構簡單、容易訓練，常用於基本的分類和回歸任務。

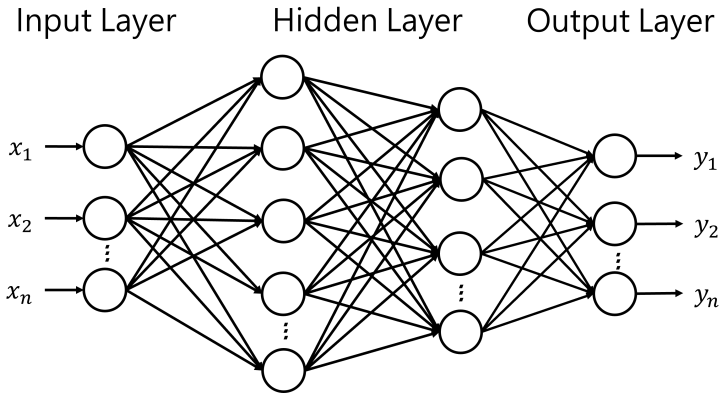


Figure 1.1: FNN 的架構

doi:10.6342/NTU202402735

- **循環神經網路 (Recurrent neural network, RNN)**[17]：架構如 Figure1.2表示，具有循環連接，能夠處理序列數據和時間關聯性，廣泛應用於語言建模、時間序列預測和機器翻譯等領域。RNN 在處理序列數據時能夠捕捉到時間上的依賴關係，但在長期依賴性和梯度消失問題上有限制 [17]。

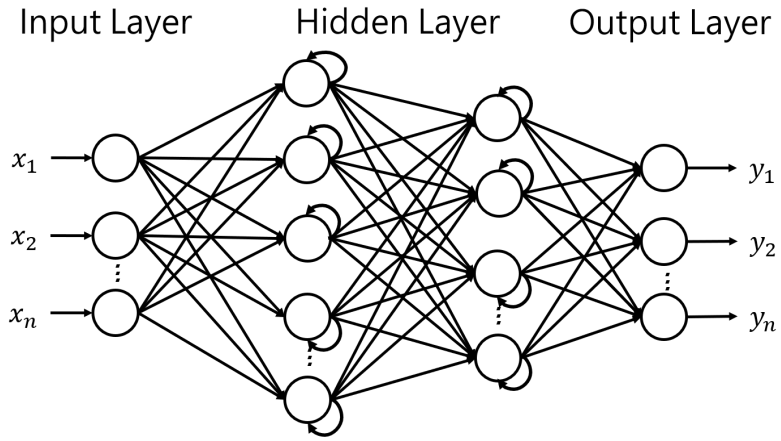


Figure 1.2: RNN 的架構

- **生成對抗網路 (Generative Adversarial Network, GAN)**[18]：由生成器和鑑別器組成，通過博弈的方式訓練。GAN 在圖像生成、風格轉換和影像合成等任務中取得了重大突破。

1.2.3 基於類神經網路的校正方法

近年來，機器學習和深度學習在的快速發展為機器手臂校正帶來了新的可能性。深度學習方法利用神經網路強大的學習和泛化能力，通過收集大量數據並進行訓練，機器可以自動學習校正過程中的模式和特徵。這些方法主要包括：

- **基於機器學習與深度學習的校正方法**：利用深度神經網路 (如卷積神經網路和遞歸神經網路) 來處理高維數據和非線性關係，提高校正精度。

- **視覺伺服校正方法**：通過攝像機等感測器收集環境數據，實現對機器手臂位

doi:10.6342/NTU202402735

置和姿態的精確校正。這種方法能夠自適應不同的環境條件，提高了機器手臂在實際應用中的可靠性和靈活性。

自 1990 年代以來，多位學者開始探索將類神經網路應用於機器手臂校正演算法中。這些研究利用神經網路的強大非線性逼近能力，旨在解決傳統校準方法中的不足之處。特別是在機器人和感測器校正領域，研究者們發現神經網路能有效處理複雜的非線性關係和多維數據，從而使校準結果更加準確和可靠。

放射狀基底函數網路 (Radial basis function network, RBFN) 是早期應用於校正問題的神經網路之一 [19]，RBFN 利用徑向基函數處理非線性關係，能在多維數據中擬合精確模型，特別適合於非幾何誤差的校正，RBFN 在非線性插值和逼近問題中具有強大的適應性。Jang 等人在 2001 年的研究 [8] 中表明，在靜態校正中，RBFN 通過學習非線性映射來補償系統性誤差，從而有效提高測量準確性。

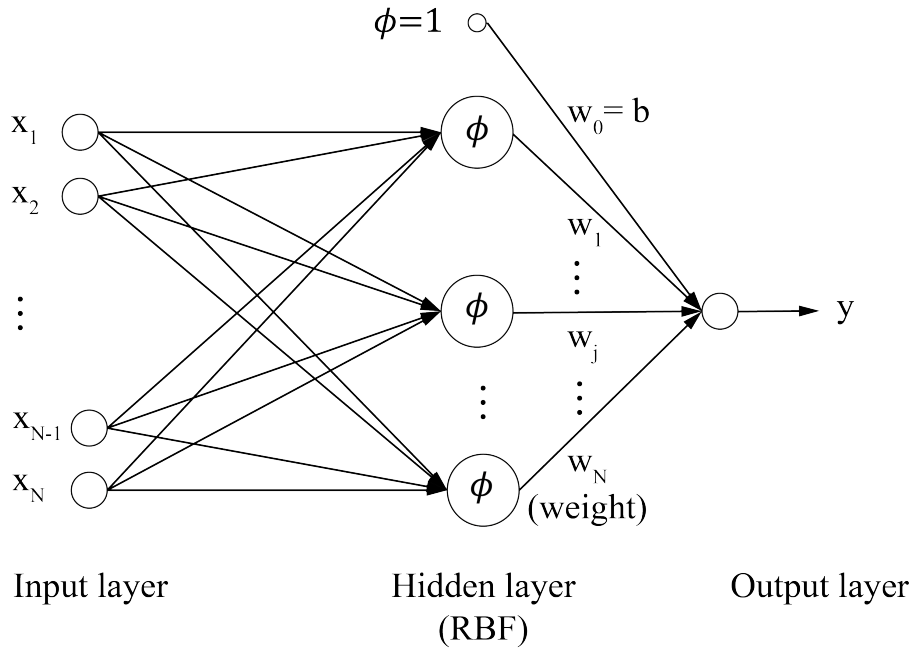


Figure 1.3: RBFN 的架構

RBFN 架構如 Figure 1.3所示，其中 ϕ 為徑向基函數，用於處理非線性插值和逼近問題。徑向基函數 ϕ 依賴於數據點之間的距離，將輸入映射到隱藏層節點並產生激活值，最常用的是高斯函數，即公式1.1：

doi:10.6342/NTU202402735

$$\phi\left(|x-x_i|\right)=\exp\left(-\frac{|x-x_i|^2}{\sigma^2}\right) \tag{1.1}$$

在工業機器人校準中，RBFN 被用於校正幾何和非幾何誤差，包括連桿長度、關節角度以及熱變形和彈性變形等。研究表明，相較於傳統最小二乘法，RBFN 提供更高的校準精度，並能適應不同的工業環境和操作條件。這些早期研究為神經網路在校準領域的應用奠定了基礎，展示了其在提高系統精度和可靠性方面的潛力。

機器學習和深度學習技術的快速發展為機器手臂校正帶來了新的可能性。深度學習方法利用神經網路強大的學習和泛化能力，通過收集大量數據並進行訓練，機器可以自動學習校正過程中的模式和特徵。這些方法主要包括：

H. Su 等人在 2019 年的研究 [5] 提出了一種利用人工神經網路 (Artificial neural network, ANN) 的方法來增強雙向遠程操作系統中的工具識別和校準。該方法利用 ANN 處理工具的位置、姿態及操作環境變量之間的非線性關係，顯著提高了系統的性能和精度。

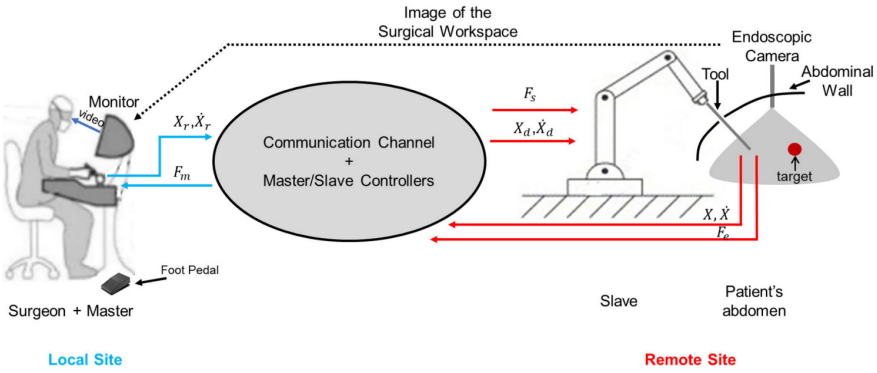


Figure 1.4: 雙向遠程操作系統
[5]

具體來說，研究者通過收集大量手術工具使用過程中的數據，包括工具的位置、姿態以及操作環境的變量。這些數據被用來訓練 ANN，以學習工具位置和姿

doi:10.6342/NTU202402735

態與校準參數之間的複雜非線性關係。訓練好的 ANN 可以被用來識別工具的當前狀態，並預測其理想狀態，從而進行準確的校準。這個方法不僅適用於補償工具的幾何誤差，如連桿長度和關節角度，還可以處理非幾何誤差，例如熱變形和彈性變形。最終，這項技術的應用顯示了基於 ANN 的方法對於提高手術機器人雙向遠程操作系統效率和可靠性的巨大潛力。

儘管基於機器學習的校正方法取得了顯著進展，但仍然存在一些挑戰和限制。例如，對大量數據的需求和模型訓練的時間成本可能會限制其在現實應用中的應用。底下介紹如何將機器學習與傳統校正方法相結合，以充分發揮其優勢。

未來的研究可以集中在數據高效的學習算法和混合校正方法兩個方向。首先，開發能夠在小數據集上有效學習的算法，以降低數據收集和標註的成本。其次，結合傳統物理模型和機器學習模型的優勢，開發更加高效和準確的校正演算法。通過這些研究和探索，本研究期待基於類神經網路的機器手臂校正方法能夠在更多實際應用中展示其潛力和優勢。

1.2.4 從模擬到現實：縮小差距的校正方法

在機器人校正中引入機器學習通常需要大量的物理實驗數據，蒐集資料集的過程相當耗時，而且物理實驗不僅成本高昂，在某些情況下很難實現，特別是當需要大量數據來訓練高性能的機器學習模型時。因此使用轉移學習，能夠先在虛擬環境中進行快速、低成本的模擬實驗，初步識別出網路結構的關鍵分支，並提供另一網路在物理實驗訓練時參數的初始值。轉移學習的核心精神，是透過將某任務中學到的知識應用到另一個相關任務中的方法 [20]。具體來說，可以在模擬環境中訓練一個模型，然後將其應用到現實環境中，這樣可以顯著減少物理實驗的數量。

doi:10.6342/NTU202402735

模擬環境中的校正效果往往與現實環境中存在差距，這種差距被稱為「現實差距」(Reality gap)，這是由於模擬環境無法完全複製現實中的各種變量。因此，學者們開始研究如何縮小這種差距，提出了一種從模擬到現實的校正方法。

Nguyen 等人 [21] 提出了一種從模擬到現實的校正方法，旨在縮小這一差距。他們通過改進模擬模型和校正算法，使得在模擬環境中得到的校正結果能夠更準確地應用到現實環境中。

Chen 等人的研究深入探討了如何量化和縮小機器手臂校正中的現實差距 [22]。他們指出，模擬環境中的理想化條件與現實世界中的複雜性之間的差異是一個挑戰。這些差異包括但不限於物理特性的不確定性、感測器噪聲和環境變化，這些因素可能導致在模擬中獲得的結果無法直接應用於現實情況中。

為了縮小這一差距，研究者們提出了多種方法。首先，改進模擬模型，使其更接近現實情況，包括引入更複雜的物理引擎和更精確的環境建模，以提高模擬結果的準確性。其次，通過在現實環境中收集大量數據來訓練校正算法，使其能夠更好地適應現實中的變化。

另一種方法是使用域隨機化 (Domain randomization) 技術 [23]，在訓練過程中隨機變化模擬環境中的各種參數，如光照條件、物體質量和摩擦係數等，以提高算法的泛化能力。這樣，訓練出的模型在面對現實世界中的不確定性時能夠保持較高的性能。

最後，混合現實 (Mixed reality) 技術也被用來縮小現實差距 [24]。通過將虛擬和現實環境結合，研究者可以在虛擬環境中模擬各種操作，同時在現實環境中進行驗證。這種方法不僅可以提高校正的精度，還能節省大量的實驗成本和時間。

縮小模擬與現實之間的差距是提高機器手臂校正效果的關鍵，研究者可以在

doi:10.6342/NTU202402735

模擬環境中獲得更準確的校正結果，並將其成功應用於現實世界中。這些方法在縮小現實差距方面取得了顯著成效，為機器手臂校正提供了新的思路和技術支持。

在本研究中，將應用轉移學習技術來進行機器手臂校正。首先在虛擬環境中進行快速、低成本的模擬實驗，識別網路結構的關鍵分支，並提供網路參數的初始值。這樣可以利用源任務學到的知識，根據目標任務的數據進行細微調整，從而提高模型在新任務上的性能。通過在現實環境中進行少量的物理實驗，對模型進行微調和驗證，期望能夠有效縮小現實差距，並將模擬環境中的校正效果成功應用到現實環境中。

doi:10.6342/NTU202402735

doi:10.6342/NTU202402735

第二章　手臂建模與校正算法

機器人運動學是用於描述機器手臂在三維空間中運動的一項技術，透過機器人運動學，可以使用參數和轉移矩陣來表示手臂各個關節之間的轉換關係，機器人運動學有助於了解機器手臂如何移動和定位。

2.1　機器人運動學

順向運動學的目標是給定機器人各關節的角度，計算出機器人末端執行器 (End-effector) 在空間中的位置和姿態，通常會使用 DH 參數 (Denavit–Hartenberg parameters) 建構出關節之間的轉換關係 [25]，運動學的描述方式可分為 Craig 和 Standard，兩者最大的差異在於定義關節轉換之間旋轉和平移的順序不同，導致齊次轉換矩陣 (Homogeneous transformation matrix) 的計算方式有所差異。由於 Standard DH 方法更加標準化，便於統一描述和分析機器人運動學，所以在研究和應用領域中更廣泛使用，因此本研究選用 Standard DH 進行說明。

由 Figure 2.1所表示，每個關節由 4 個參數表達，Standard DH 建立參數的方式如以下敘述，兩兩相鄰之坐標系分別依照 x 軸、z 軸為參考軸，定義出以 z_{i-1} 為參考軸，看 x_{i-1} 到 x_i 的旋轉 θ_i 和平移 d_i；以 x_i 為參考軸，看 z_{i-1} 到 z_i 的旋轉 α_i 和平移 a_i。

doi:10.6342/NTU202402735

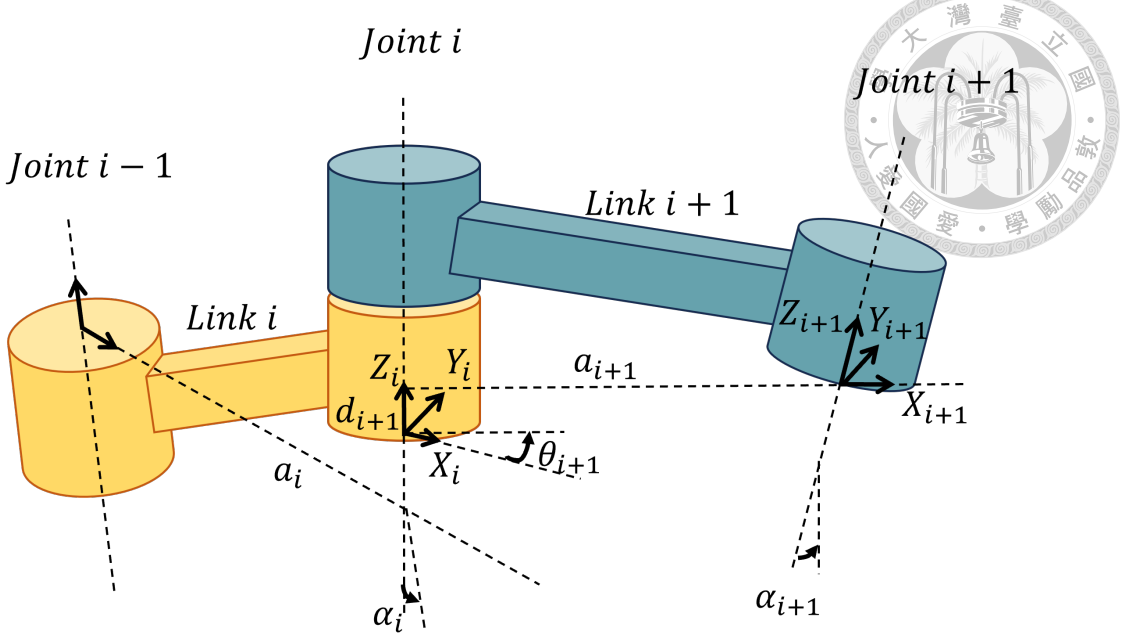


Figure 2.1: Standard DH 模型架構圖

按照 Standard DH 定義旋轉之方式和順序，可將齊次轉換矩陣之形式表為公式 (2.1)。

$$ {}^{i-1}_{i}T = \text{Rot}_{z_i}(\theta_i)\ \text{Trans}_{z_i}(d_i)\ \text{Trans}_{x_i}(a_i)\ \text{Rot}_{x_i}(\alpha_i) \tag{2.1} $$

其中，$\text{Rot}_{z_i}(\theta_i)$ 表示繞 z_i 軸旋轉 θ_i 角度的旋轉矩陣，$\text{Trans}_{z_i}(d_i)$ 表示沿 z_i 軸平移 d_i 距離的平移矩陣，$\text{Trans}_{x_i}(a_i)$ 表示沿 x_i 軸平移 a_i 距離的平移矩陣，$\text{Rot}_{x_i}(\alpha_i)$ 表示繞 x_i 軸旋轉 α_i 角度的旋轉矩陣，將公式 (2.1) 展開則得到齊次轉換矩陣公式 (2.2)

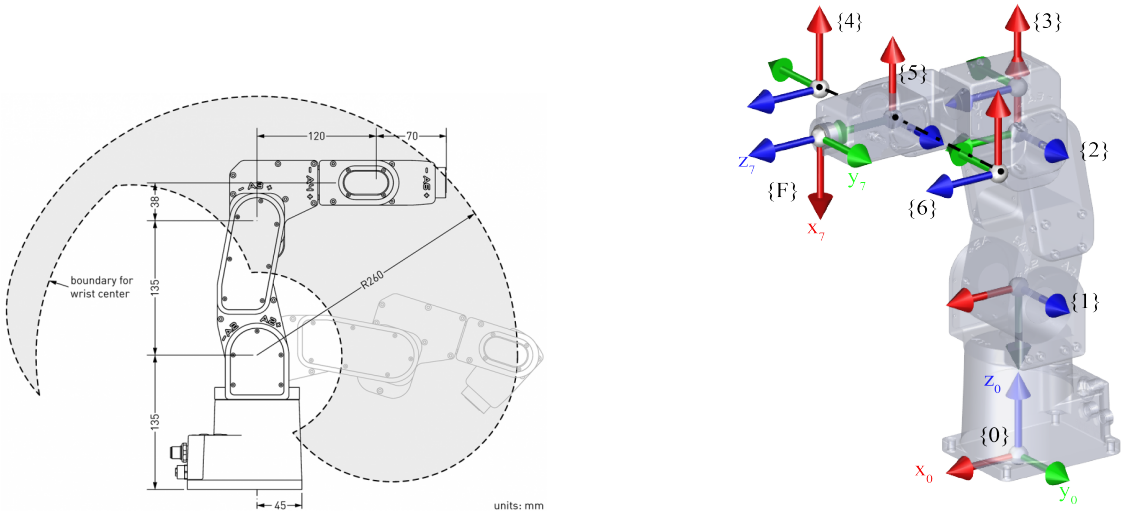
$$ {}^{i-1}_{i}T = \left[\begin{array}{ccc|c} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{2.2} $$

doi:10.6342/NTU202402735

以下使用六軸手臂 Meca500 作為範例，首先必須先建立手臂的幾何描述，先找出手臂的所有關節 (Joint) 以及連桿 (Link)，並且於關節處以馬達旋轉之方向為 z 軸建立每個關節的坐標系 (Frame)，Figure 2.2b為此描述法建立出的機器手臂架構圖。如此一來即可依照 Standard 定義之法則去描述兩關節之間的轉換關係，如 Table2.1。

Table 2.1: DH 表

i	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)
1	$\theta_1 - \frac{\pi}{2}$	135	0	$-\frac{\pi}{2}$
2	θ_2	0	135	0
3	θ_3	0	38	$-\frac{\pi}{2}$
4	0	120	0	0
5	θ_4	0	0	$\frac{\pi}{2}$
6	θ_5	0	0	$-\frac{\pi}{2}$
7	$\theta_6 + \pi$	70	0	0



(a) Meca500 規格和移動範圍 [26]　　(b) Meca500 座標圖

Figure 2.2: Meca500 (a) 規格和移動範圍 (b) 座標圖

在定義這些參數的同時，參照從 Meca500 的規格書 [26] 節錄的移動範圍 Figure 2.2a，可將 Table 2.1填好，並根據 Standard DH 的齊次轉換矩陣之形式 (2.2)，將 Table 2.1之參數按照此形式填入。

doi:10.6342/NTU202402735

對於 6 自由度的串聯機器手臂，其末端執行器座標系 $\{F\}$ 到機器人基底座 $\{0\}$ 的變換矩陣，將每個座標系的齊次轉換矩陣依序連乘如 (2.3)，即完成 Standard DH 之建構。

$$ {}^{0}_{F}T = {}^{0}_{1}T\ {}^{1}_{2}T\ {}^{2}_{3}T\ {}^{3}_{4}T\ {}^{4}_{5}T\ {}^{5}_{6}T\ {}^{6}_{F}T \tag{2.3} $$

2.2 校正演算法

在機器人製造過程中，裝配誤差、材料特性和物理特性的變異可能導致機器手臂的末端點偏移，進而影響其運動精度。傳統方法是利用雅可比矩陣 (Jacobian matrix) 進行運動學校正。然而，這種方法存在著無法完全捕捉到實際機器人系統複雜性的挑戰。製造和裝配過程中的偏移和誤差可能受到多種因素的影響，包括材料特性、製造過程中的變異、以及工作環境的影響等。因此，單純依靠雅可比矩陣的校正方法往往難以有效地處理這些複雜情況，導致校正結果的不準確性。

為克服這一限制，本研究提出了一個本論文提出了一種平行架構模型如 Figure 2.3所表示，來預測手臂末端的誤差，將傳統的雅可比矩陣校正與前饋式神經網路進行結合，使得模型能夠應對幾何與非幾何誤差。除此之外模型訓練還會將數位雙生技術和轉移學習相結合，詳細內容後續會在 2.2.4小節說明。

目標是通過這種方法找出最接近實際機器手臂行為的模型，從而改進機器手臂的精度控制。通過結合數位雙生技術和轉移學習，可以更好理解機器人系統的複雜性，並提出更有效的校正方法，以應對不斷增長的工業和醫療應用需求，進而實現更高水平的運動控制和精度。

doi:10.6342/NTU202402735

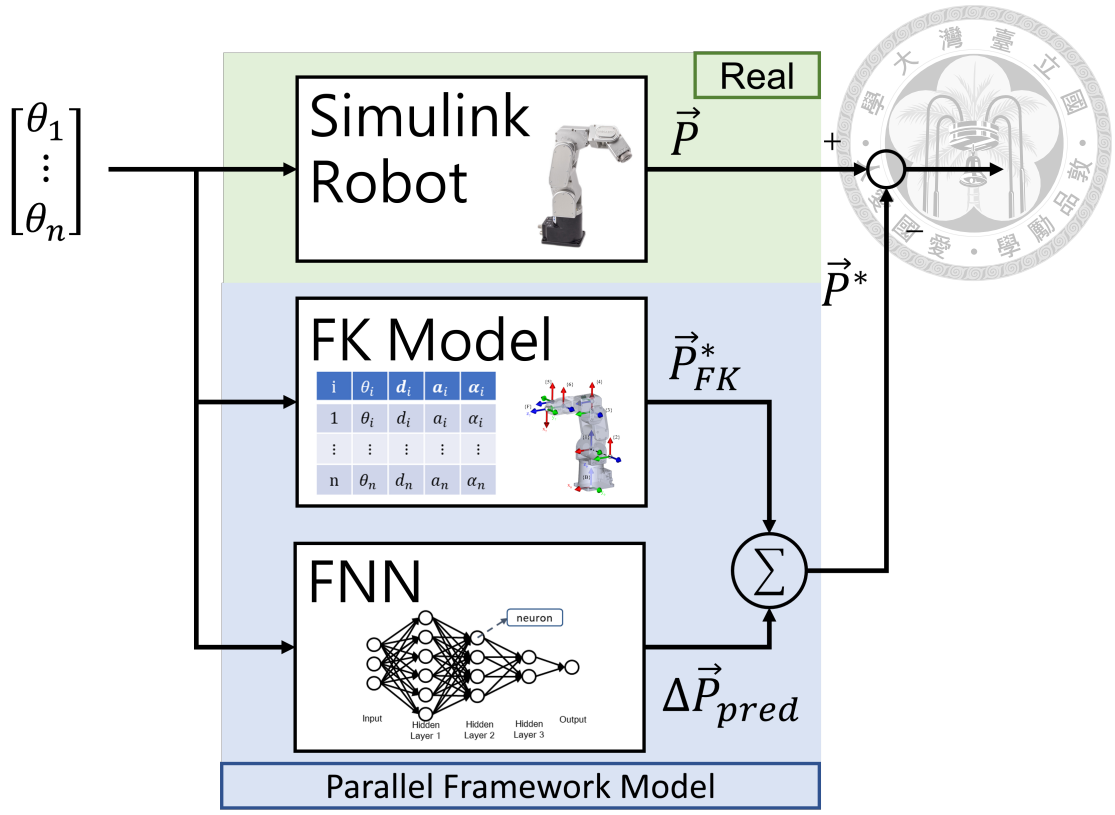


Figure 2.3: 平行架構模型

2.2.1 幾何校正

Figure2.4是本研究所使用的幾何校正流程圖，傳統的校正法，是透過修正章節2.1提及的 DH parameters，來讓模型可以更接近真實系統。首先，收集數筆實際手臂的末端姿態 $\vec{P}$。然後利用這些數據，即可透過雅可比算法對順向運動學模型 (FK Model) 的參數進行初步校正。

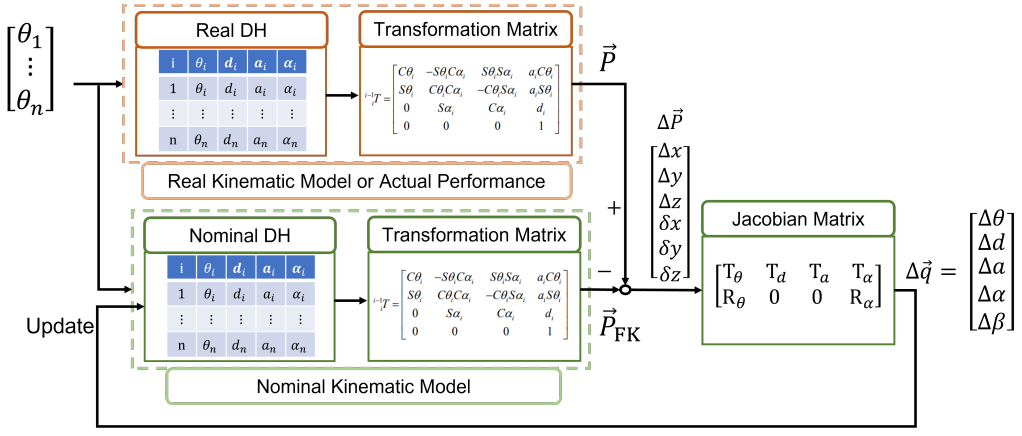


Figure 2.4: 手臂的 Jacobian 算法示意圖

doi:10.6342/NTU202402735

在運動學校正中，S. Hayati 提出一個問題 [9]，Standard DH 原有四組參數 θ、d、a、α，不適用於機器手臂包含兩個連續的平行或幾乎平行的關節校正，因為 z_i 軸對齊的微小誤差，會導致其 DH 參數的較大誤差。具體說明如 Figure2.5，假設在機器手臂的裝配過程中，若座標系 Z_i' 因裝配誤差未能與預設的座標系 Z_i 完全對齊，且此誤差表現為繞 Y_i 軸的旋轉角度 β_i，則會對機器手臂的運動學參數產生影響。根據 Standard DH 參數的定義，座標系 X_i 的方向應與 Z_i 和 Z_{i-1} 的公垂線一致。因此看到表2.2，β_i 的小誤差會導致關節角度 θ_i 和連桿偏移 a_i 產生較大誤差，甚至會有連桿長度 L_i 無法被明確描述的情況，這些誤差會使實際運動與預期設計之間存在差距，進而影響機器手臂的精確性和性能。

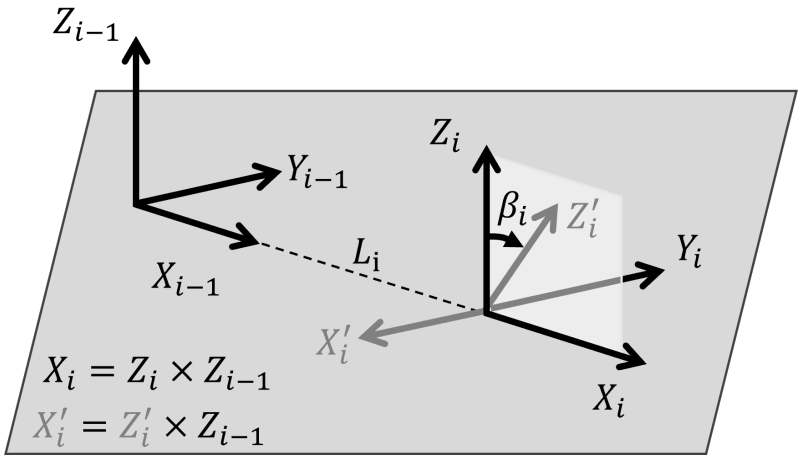


Figure 2.5: 兩連續平行之坐標系

Table 2.2: z_i 軸對齊所造成的參數大偏移

i	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)
Frame$\{i\}$	0	0	L_i	0
Frame$\{i\}'$	$-\frac{\pi}{2}$	0	0	β_i

因此 S. Hayati 提出了一種解決方法，改進 standard DH 的齊次轉換矩陣，即在 y 軸周圍對 β 進行微小的旋轉 $\text{Rot}_{y_i}(\beta_i)$ 如公式 (2.4)。

$$^{i-1}_{i}T = \text{Rot}_{z_i}(\theta_i)\ \text{Trans}_{z_i}(d_i)\ \text{Trans}_{x_i}(a_i)\ \text{Rot}_{x_i}(\alpha_i)\ \text{Rot}_{y_i}(\beta_i) \qquad (2.4)$$

doi:10.6342/NTU202402735

展開後的 Standard DH 齊次轉換矩陣之形式表示為 (2.5)，即可得知每個關相鄰關節的轉換關係。公式 (2.5) 中，使用了簡化符號來表示三角函數。其中，C 表示 cos，S 表示 sin。

$$
{}^{i-1}_{i}T = \left[\begin{array}{ccc|c}
C\theta_i C\beta_i - S\theta_i S\alpha_i S\beta_i & -S\theta_i C\alpha_i & C\theta_i S\beta_i + S\theta_i S\alpha_i C\beta_i & a_i C\theta_i \\
S\theta_i C\beta_i + C\theta_i S\alpha_i S\beta_i & C\theta_i C\alpha_i & S\theta_i S\beta_i - C\theta_i S\alpha_i C\beta_i & a_i S\theta_i \\
-C\alpha_i S\beta_i & S\alpha_i & C\alpha_i C\beta_i & d_i \\
\hline
0 & 0 & 0 & 1
\end{array}\right] \tag{2.5}
$$

將此旋轉操作添加到機器人的運動學模型中，用處理兩個平行關節或幾乎平行連續關節的情況。

含有誤差的機器手臂模型可表示為公式 (2.6)，將其展開，並且忽略較小的偏差項即可近似為每項都只有一個 ΔT。

$$
\begin{aligned}
{}^{0}_{N}T + \Delta {}^{0}_{N}T &= \prod_{n=1}^{N} ({}^{n-1}_{n}T + \Delta {}^{n-1}_{n}T) \\
&\approx \prod_{n=1}^{N} ({}^{n-2}_{n-3}T \ \dots \ {}^{n-1}_{n-2}T \ \Delta {}^{n-1}_{n}T \ {}^{n}_{n+1}T \ \dots \ {}^{n+1}_{N}T)
\end{aligned} \tag{2.6}
$$

將其對運動學參數 θ、d、a、α、β 的偏差展開為線性函數 (2.7)。

$$
\Delta {}^{n-1}_{n}T = \sum_{i=1}^{N} \left(\frac{\partial {}^{n-1}_{n}T}{\partial \theta_i}\Delta\theta_i + \frac{\partial {}^{n-1}_{n}T}{\partial d_i}\Delta d_i + \frac{\partial {}^{n-1}_{n}T}{\partial a_i}\Delta a_i + \frac{\partial {}^{n-1}_{n}T}{\partial \alpha_i}\Delta\alpha_i + \frac{\partial {}^{n-1}_{n}T}{\partial \beta_i}\Delta\beta_i\right) \tag{2.7}
$$

並忽略高階項，機器手臂末端執行器 F 相對於機器手臂基架 B 的微分可以近

doi:10.6342/NTU202402735

似為公式 (2.8)，其表示為各個運動學參數偏差的線性函數。

$$\begin{aligned}\Delta {}^{0}_{N}T &= \frac{\partial {}^{0}_{N}T}{\partial q_i}\Delta q_i \\ &= \sum_{i=1}^{N}\left(\frac{\partial {}^{0}_{N}T}{\partial \theta_i}\Delta\theta_i + \frac{\partial {}^{0}_{N}T}{\partial d_i}\Delta d_i + \frac{\partial {}^{0}_{N}T}{\partial a_i}\Delta a_i + \frac{\partial {}^{0}_{N}T}{\partial \alpha_i}\Delta\alpha_i + \frac{\partial {}^{n-1}_{n}T}{\partial \beta_i}\Delta\beta_i\right)\end{aligned} \tag{2.8}$$

其中，$\frac{\partial {}^{0}_{N}T}{\partial q_i}$ 如公式 (2.9)，每個轉移矩陣 $T_{4\times4}$ 都可以將其，拆分為位置 $M_{3\times1}$ 與旋轉矩陣 $R_{3\times3}$，而旋轉矩陣能轉換成旋轉向量。

$$\frac{\partial {}^{0}_{N}T}{\partial q_i} = {}^{0}_{1}T\ {}^{1}_{2}T\ \ldots\ \frac{\partial^{i-1}_{i}T}{\partial q_i}\ {}^{i}_{i+1}T\ \ldots\ {}^{N-1}_{N}T = \left[\begin{array}{c|c} R_{3\times3} & M_{3\times1} \\ \hline 0_{1\times3} & 1 \end{array}\right] \tag{2.9}$$

因此物理手臂的末端姿態 $\vec{P}$ 與還沒經過校正的標稱手臂運動學模型的末端姿態 $\vec{P}_{FK}$ 的誤差 $\Delta\vec{P}$ 和應該要補償的運動參數 $\vec{q}$ 的誤差方程可以將誤差模型改寫為矩陣格式，如公式 (2.10) 所示，其中矩陣 J 就是雅可比矩陣，它概述了每個運動學參數誤差如何影響機器人的位置和方向精度，透過最小平方法即可求解每個參數應該補償的量。

$$\begin{aligned}\Delta\vec{P} &= J\Delta\vec{q} \\ &= \begin{bmatrix} M_\theta & M_d & M_a & M_\alpha & M_\beta \\ R_\theta & 0 & 0 & R_\alpha & R_\beta \end{bmatrix}\begin{bmatrix}\Delta\theta & \Delta d & \Delta a & \Delta\alpha & \Delta\beta\end{bmatrix}^T\end{aligned} \tag{2.10}$$

doi:10.6342/NTU202402735

2.2.2 基於機器學習的非幾何校正

使用機器學習校正的主要特點是能夠處理前一小節2.2.1傳統校正的缺陷，因傳統校正只能針對幾何所造成的誤差，對運動學模型進行初步的校正。較難補償的非線性誤差，將利用機器學習來完成，藉此達到 Figure 中的2.3平行架構模型。

如何使用神經網路學習非幾何誤差的目標可以參考 Figure 2.6，$\vec{P}$ 為物理手臂之末端姿態，此處運動學模型的參數已經使用2.2.1小節的 Jacobian 算法更新完畢，並且算出更新後的末端姿態 $\vec{P}^*_{FK}$，將兩者相減得到的 $\Delta\vec{P}_{NGE}$ 即為那些無法被傳統校正彌補的非線性誤差。

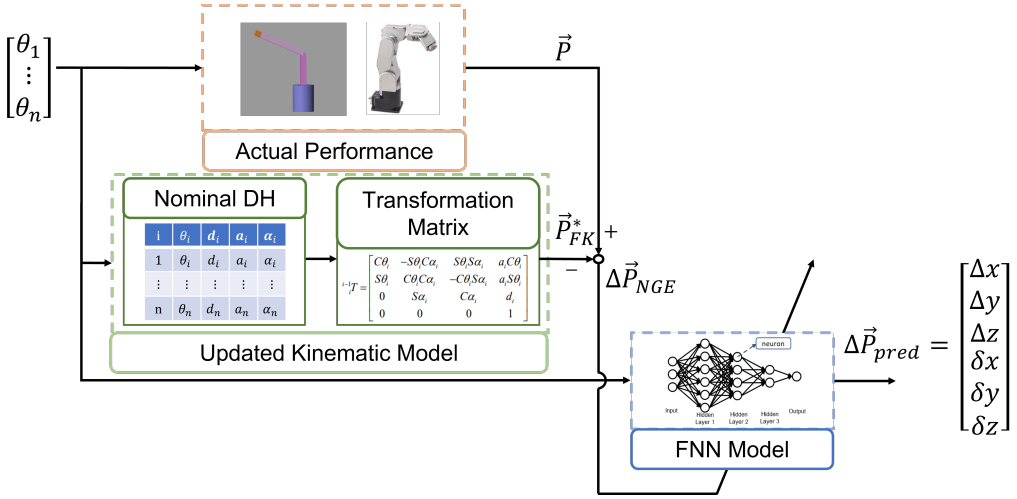


Figure 2.6: 手臂的機器學習流程示意圖

機器學習模型需要透過大量的數據來提高校正的準確度以及強健性。機器學習在機器人校正中的應用日益普及，本篇將介紹前饋神經網路 (Feedforward Neural Network, FNN)[13]，其在學習複雜的非幾何誤差有相當大的優勢。在機器手臂校正中，神經網路模型可以被用來學習關節角度與末端執行器位置之間的關係，具體實現過程如下：

1. **數據收集與預處理**：收集大量機器手臂的運動數據，包括關節角度、位置和對應的末端執行器位置。這些數據經過標準化處理，以確保輸入數據的一致性和穩定性。

doi:10.6342/NTU202402735

2. **網路結構設計**：設計一個適合機器手臂校正任務的神經網路結構。首先確定隱藏層的數量和每層神經元的數量。對於手臂校正問題，隱藏層的設計應考慮到網路的非線性表達能力，以便能夠精確地學習和校正手臂運動中的誤差。一般來說，較多的隱藏層和神經元可以提高模型的表達能力，但也會增加訓練的複雜性和計算成本。因此，需要在網路複雜性與訓練效率之間進行平衡。

3. **模型訓練**：使用預處理過的數據來訓練神經網路模型，目標是最小化模型輸出與實際末端位置之間的誤差。訓練過程中可以使用優化算法如 Adam，並通過調整超參數來提升模型性能。

4. **模型驗證與測試**：在獨立的測試數據集上驗證模型的校正效果。確保訓練出的網路模型在不同的手臂運動狀態下均能提供準確的校正結果。對模型進行評估，檢查其泛化能力和穩定性，並在必要時進行剪枝。

FNN 其結構如 Figure2.7，由多層神經元組成，包括輸入層、隱藏層和輸出層。每一層的神經元與下一層的神經元彼此相連，且信號僅在網路中前向傳播。FNN 適合用於靜態數據的處理和分析，能夠有效學習輸入角度和非幾何誤差之間的非線性映射關係。以下是 FNN 的結構、算法詳細介紹：

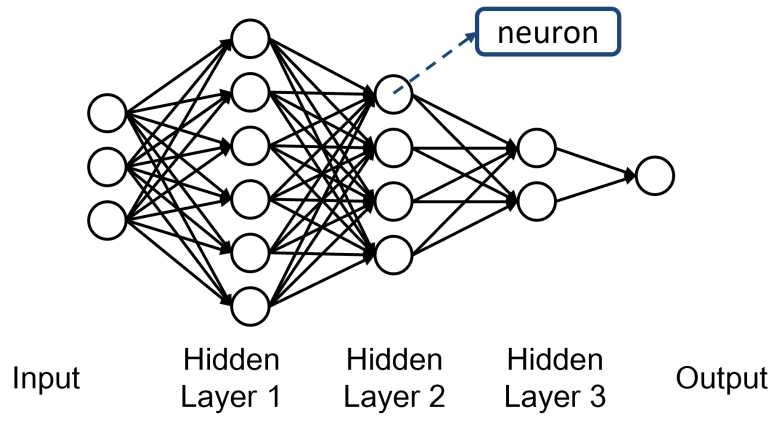


Figure 2.7: FNN 主要架構

doi:10.6342/NTU202402735

- **輸入層 (Input layer)**：輸入層的神經元接收資料集中的特徵或屬性，並將其傳遞給網路的其他層。輸入層的神經元數量必須與資料集中的特徵數量相匹配。

- **隱藏層 (Hidden layer)**：將輸入層和輸出層分開。根據模型的類型，隱藏層可以有多個，每層包含多個神經元。隱藏層的神經元在將輸入傳遞給下一層之前對其進行變換。

- **輸出層 (Output layer)**：輸出層根據構建的模型類型，生成預測的特徵。

神經元和權重

神經元是 FNN 中的基本單元，類似於生物神經元。它們有兩個主要功能：首先，計算加權輸入的總和，將總和通過激活函數轉換成輸出，如 Figure 2.8所示。

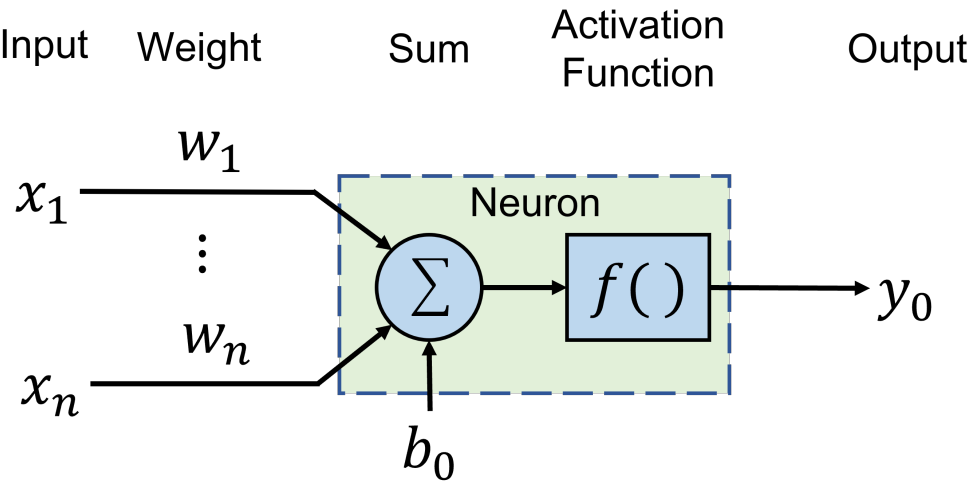


Figure 2.8: 神經元與權重

1. **神經元權重 (Neuron weights)**：神經元之間通過權重相連，每個神經元都是一個函數，這些權重測量其輸入信號的強度或幅度。權重類似於線性回歸中的係數，其在訓練過程中進行調整。權重的初始值通常設置在一個小範圍內(如接近零的隨機值)，並且會在學習過程中根據誤差進行更新。

doi:10.6342/NTU202402735

2. **激活函數 (Activation function)**：激活函數 $f(\cdot)$ 決定每個神經元的輸出。其可以是線性或非線性，以下介紹一些常見的激活函數。

(a) **Sigmoid**：將輸入值映射到 0 和 1 之間的輸出值，如公式 (2.11)。

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.11}$$

Sigmoid 函數的優點是輸出範圍固定在 0 到 1 之間，如此一來可將輸入值標準化。然而，在輸入值非常大或非常小時，Sigmoid 函數的梯度會變得非常小，導致梯度消失 (gradient vanishing) 問題，這使得網路訓練變得困難。

(b) **Tanh**：將輸入值映射到-1 和 1 之間的輸出值，如公式 (2.12)。

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.12}$$

相比於 Sigmoid 函數，Tanh 函數的輸出範圍更大且中心對稱，對於中心對稱的數據具有更好的性能。然而，Tanh 函數也存在梯度消失問題，尤其在輸入值非常大或非常小時。

(c) **ReLU**：ReLU 函數僅允許正值通過，將負值映射為 0，如公式 (2.13)。

$$\text{ReLU}(x) = \max(0, x) \tag{2.13}$$

ReLU 函數的優點是計算簡單且能夠緩解梯度消失問題，因為其導數在正值範圍內始終為 1。然而，ReLU 函數存在“死亡 ReLU”問題，即一些神經元可能會因為輸入值總是小於 0 而永遠不會被激活。

激活函數的選擇影響網路的學習能力和性能。ReLU 因其簡單性和有效性在機器學習中被廣泛使用。

doi:10.6342/NTU202402735

損失函數和最佳化演算法

1. **損失函數 (Loss function)**：衡量網路預測值 $\hat{y}_i$ 與實際值 y_i 之間的誤差，進而評估模型的性能。損失值越小，表明模型的預測越接近實際值，性能越好。以下是一些常用的損失函數：

   (a) **均方誤差 (Mean squared error, MSE)**：

   $$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.14}$$

   MSE 衡量的是預測值與實際值之間的平均平方差，它對於大的誤差比較敏感，因為誤差被平方了，值越小表示預測結果越精確。這個損失函數常用於回歸問題。

   (b) **平均絕對誤差 (Mean absolute error, MAE)**：

   $$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.15}$$

   MAE 衡量的是預測值與實際值之間的平均絕對誤差。相比 MSE，MAE 對於異常值 (Outliers) 更為強健，因為它對每個誤差給予同等的權重。

   (c) **Huber 損失函數 (Huber loss)**：

   $$\text{Huber loss} = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2, & \text{if } |y_i - \hat{y}_i| \le \delta \\ \delta(|y_i - \hat{y}_i| - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \tag{2.16}$$

   Huber Loss 是一種結合了 MAE 和 MSE 優點的損失函數。δ 是權衡要使用 MAE 或 MSE 的參數，當誤差小於 δ 時，它的行為類似於 MSE，否

doi:10.6342/NTU202402735

則類似於 MAE。這使得 Huber loss 對於異常值更加強健。

2. **最佳化演算法 (Optimization algorithm)**：在訓練過程中，模型的目標是最小化損失函數的值，這通常通過梯度下降算法來實現。權重更新的過程如下：

   (a) **梯度下降算法 (Gradient descent)**：用於最小化損失函數。在每一個訓練迭代中，計算損失函數對於權重的梯度，然後按照梯度的反方向更新權重。這樣，損失函數值會隨著時間逐漸降低，直到收斂到最小值或者接近最小值。梯度下降算法的公式如下：

$$w_t = w_{t-1} - \eta \frac{\partial L}{\partial w_{t-1}} \tag{2.17}$$

   其中，w 是權重，η 是學習率，L 是損失函數。

   (b) **動量法 (Momentum)**：動量法在更新權重時考慮了之前的更新方向，這樣可以加速收斂速度並且減少更新方向的變異性。具體來說，動量法引入了一個動量項，用於記錄之前更新的方向，並將其納入到當前的更新中。動量法的公式如下：

$$v_t = \gamma v_{t-1} - \eta \frac{\partial L}{\partial w_{t-1}} \tag{2.18}$$

$$w_t = w_{t-1} + v_t \tag{2.19}$$

   其中，γ 是動量參數，v 是之前的更新方向。

   (c) **Adam 優化算法 (Adaptive moment estimation)**：

   Adam 是一種結合了動量法和自適應學習率的優化算法。它基於梯度的一階動差函數 (First-order moment estimate) 平均梯度和二階動差函數 (Second-order moment estimate) 梯度的平方的平均值進行權重更新，具有自適應性和收斂速度快的特點，Adam 保有兩個移動平均的估計值，

doi:10.6342/NTU202402735

分別是梯度的一階動差函數和二階動差函數。

(d) **一階動差函數 (平均梯度)**：一階動差函數公式 (2.20) 計算梯度的指數移動平均，以捕捉梯度的動態特性。這個估計值可以看作是梯度的方向，用於更新動量。

(e) **二階動差函數 (梯度的平方的平均值)**：二階動差函數公式 (2.21) 計算梯度的平方的指數移動平均，以捕捉梯度的變化幅度。這個估計值可以看作是梯度的大小，用於調整學習率。

Adam 算法的更新公式如下：

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{2.20}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(g_t)^2 \tag{2.21}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{2.22}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{2.23}$$

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{2.24}$$

β_1 和 β_2 是控制一階和二階動差函數的衰減率。學習率 η 決定了更新的步長，而 ϵ 是一個微小常數，用於防止除以零。為了校正一階和二階動差函數的偏差，計算校正後的一階動差函數 $\hat{m}_t$ 和二階動差函數 $\hat{v}_t$。這些校正後的估計值用於計算權重的更新，確保在訓練過程中權重能夠有效地收斂。

模型訓練與驗證

訓練過程一般以隨機小值初始化權重和偏移。然後，對訓練資料集進行多次迭代，每次迭代都包括前向傳播、損失計算、反向傳播和權重更新。在每次迭代

doi:10.6342/NTU202402735

中，評估損失的變化，確保模型逐漸收斂到最優狀態。最後，使用測試資料集評估 FNN 模型的校正效果，以確保其能夠在不同運動狀態下提供準確的校正結果。

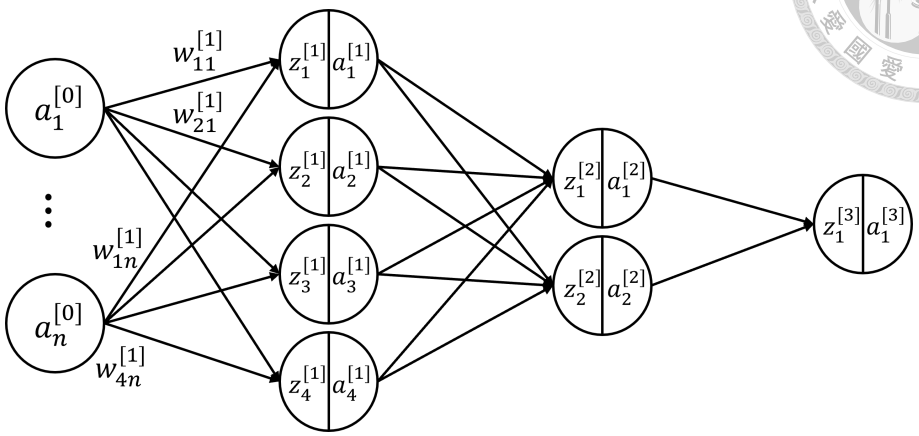


Figure 2.9: NN 示意圖

1. **前向傳播**：每層神經網路的節點運作如 Figure2.9 所示。前向傳播過程中，每個節點接收前一層節點的輸出，將其乘以相對應的權重並加上偏移，得到 z 如公式 (2.25)，再通過激活函數產生新的輸出 a。

   (a) 計算每層的輸入 $z^{(l)}$ 和激活值 $a^{(l)}$：

   $$z^{(l)} = W^{(l)}x^{(l)} + b^{(l)} \tag{2.25}$$

   $$a^{(l)} = f(z^{(l)}) \tag{2.26}$$

   (b) 計算損失：使用 MAE 公式 (2.15) 做為損失函數計算損失，其中，$\hat{y}_i = a^{(L)}$ 是輸出層的輸出值。

2. **反向傳播**：反向傳播用於計算每層的梯度，從而更新模型的參數 (包含權重和偏置)，通過損失函數對每個權重和偏置的梯度計算，從輸出層向輸入層進行反向傳播，然後根據這些誤差調整每一層的參數。

   (a) 計算輸出層的誤差項 $\delta^{(L)}$：

doi:10.6342/NTU202402735

輸出層的誤差項是對損失函數相對於輸出的梯度。使用均方根誤差 (RMS) 損失函數計算，其公式如下：

$$\delta^{(L)} = \frac{\partial L}{\partial a^{(L)}} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{2.27}$$

其中，$\delta^{(L)}$ 表示輸出層的誤差項，$\hat{y}_i$ 是模型預測的輸出，而 y_i 是實際值。

(b) 計算隱藏層的誤差項 $\delta^{(l)}$：

對於每一層 l(從輸出層往前遍歷到第一層)，需要計算每層的梯度 $\delta^{(l)}$：

$$\delta^{(l)} = (\delta^{(l+1)} \cdot W^{(l+1)T}) \odot f'(z^{(l)}) \tag{2.28}$$

在這裡，$\delta^{(l)}$ 是第 l 層的誤差項，$W^{(l+1)T}$ 是第 $l+1$ 層權重的轉置矩陣，$f'(z^{(l)})$ 是激活函數 $f(z)$ 在 $z^{(l)}$ 上的導數。$\odot$ 表示逐元素相乘 (Hadamard 乘積)。

(c) 計算權重和偏置的梯度：

對於每一層 l，權重 $W^{(l)}$ 和偏置 $b^{(l)}$ 的梯度分別為：

$$\frac{\partial L}{\partial W^{(l)}} = a^{(l-1)T}\delta^{(l)} \tag{2.29}$$

$$\frac{\partial L}{\partial b^{(l)}} = \delta^{(l)} \tag{2.30}$$

這些梯度將用於更新神經網路的權重和偏置，以最小化損失函數。

3. **更新權重**：在 Adam 最佳化演算法中，首先計算梯度，然後使用這些梯度來更新模型參數。使用公式 (2.20)~(2.24)Adam 優化算法來更新權重，公式

doi:10.6342/NTU202402735

(2.20) 和公式 (2.21) 內的 g_t 具體包含了模型的所有參數梯度：

$$g_t = \{\nabla W^{(1)}, \nabla b^{(1)}, \nabla W^{(2)}, \nabla b^{(2)}, \ldots, \nabla W^{(L)}, \nabla b^{(L)}\} \tag{2.31}$$

根據計算得到的梯度可透過帶入公式 (2.22) 與公式 (2.23)，即可自適應學習率進行調整：

$$w_{t+1} = w_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{2.32}$$

$$b_{t+1} = b_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{2.33}$$

總結來說，機器學習在機器手臂校正中的應用具有很大的潛力。通過合理的資料收集與預處理、精心設計的網路結構以及有效的訓練和驗證方法，模型能夠學習機器手臂的運動模式並提供準確的校正結果。

相比傳統方法，類神經網路在處理複雜非線性系統時展現出更高的精度和適應性。然而，類神經網路在應對實時性要求高的應用、資料量有限的情況下避免過擬合，以及處理噪聲和不確定性方面仍需進一步研究。未來的工作應集中於開發更強大穩健的神經網路結構，並探索更有效的訓練和優化方法，以提高其性能和可靠性。

2.2.3 機器學習中的模型優化

剪枝 (pruning) 技術 [27] 在機器學習中被廣泛應用，旨在通過減少模型中冗餘參數來提高運行效率和節省計算資源。權重剪枝能夠移除網路中被認為不必要的個別權重如 Figure2.10，這種技術基於對模型參數的分析，逐步剔除對模型預測性能貢獻有限的參數，從而達到優化模型的目的，具體實現方法如下：

doi:10.6342/NTU202402735

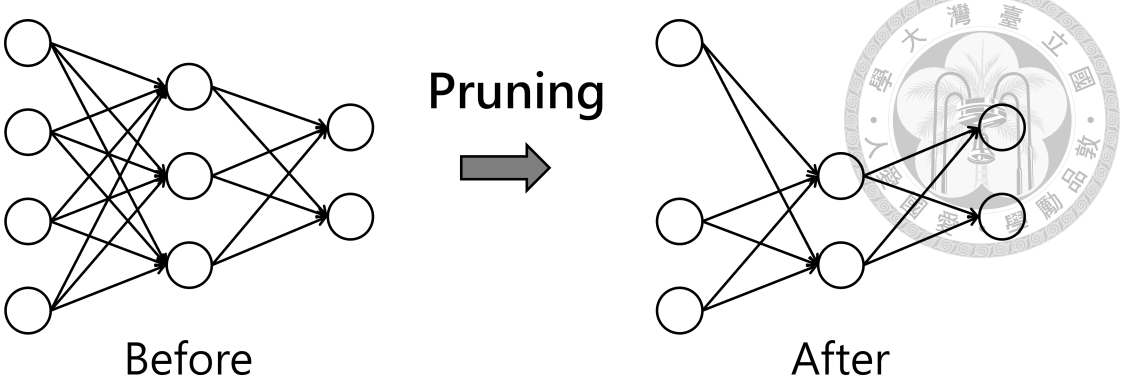


Figure 2.10: 剪枝模型示意圖

1. **選擇稀疏度衰減方程式**：首先，設定初始稀疏度 s_0 和最終稀疏度 s_f。這些參數決定了從何時開始剪枝以及剪枝到何種程度。並且選擇適合的稀疏度衰減方程式，以決定稀疏度 (剪枝率) $s(t)$ 隨時間 t 或訓練步數的變化。常見的方程式包括指數衰減和多項式衰減。

   - **指數衰減剪枝 (Exponential decay pruning)**：

$$s(t) = s_0 + (s_f - s_0) \cdot (1 - e^{-\eta \cdot t}) \tag{2.34}$$

   其中 s_0 是初始稀疏度，s_f 是最終稀疏度，η 是衰減速率，t 是訓練步數。

   - **多項式衰減剪枝 (Polynomial decay pruning)**：

$$s(t) = s_0 + (s_f - s_0) \left(\frac{t - t_0}{t_f - t_0} \right)^p \tag{2.35}$$

   其中，t_0 和 t_f 分別是剪枝開始和結束的步數，p 是多項式的次數，決定了衰減的速度和形狀。

2. **確定剪枝閾值**：修剪的決策法很多種，本處介紹兩種分別是基於按照全種大小修剪以及按照百分比修剪。前者設定一個固定的數值閾值 θ，低於此數值的權重將被剪枝；後者將權重按絕對值排序，剪枝 $s(t)$ 當前步伐百分比的最

doi:10.6342/NTU202402735

小權重，如公式 2.36計算出當前百分比相對應的閾值 θ，再利用公式2.37的原則進行剪枝。例如，如果設定剪枝 10% 的權重且閾值計算出來為 0.01，則移除權重絕對值小於 0.01 的前 10% 最小權重的參數。

$$\theta = Percentile(|W|, s) \tag{2.36}$$

$$w_i' = \begin{cases} 0, & \text{if } |w_i| < \theta \\ w_i, & \text{otherwise} \end{cases} \tag{2.37}$$

3. **計算結束步驟**：計算結束步驟 t_f，這個步驟指明了在多少個訓練步驟後，稀疏度達到設置的最終值。可以通過以下計算確定：

$$t_f = \frac{\text{Num samples} \times \text{Epochs}}{\text{Batch size}} \tag{2.38}$$

其中，Num samples 是訓練數據樣本數，Batch size 是批量大小，Epochs 是訓練輪數。

4. **應用剪枝到模型**：使用剪枝技術將選擇的稀疏度衰減方程式應用到預先訓練好的模型上。通過評估模型中的權重和神經元，識別出那些對模型輸出影響較小的冗餘參數，並將識別出的冗餘參數從模型中移除，重新計算稀疏程度，從而減少模型的大小和計算量。根據指定的稀疏度衰減方程式，自動調整模型中每層的權重，以實現稀疏化。

5. **重新編譯模型**：在應用了剪枝後，重新編譯模型，使用適當的優化器和損失函數。這確保了修剪後的模型可以有效地進行訓練和評估。

6. **訓練和評估**：最後，通過訓練和評估修剪後的模型，觀察其在驗證集上的性能表現，確保模型在減少參數的同時保持良好的預測能力。

doi:10.6342/NTU202402735

剪枝技術能有效精簡和優化預訓練模型，顯著減少模型的大小和計算負擔，使其在轉移學習中更適用於新任務。然而，若執行不慎，可能導致重要信息丟失和模型性能下降。因此，常需借助微調和再訓練等技術來修復潛在的準確性損失。這不僅提升了模型推理的速度，也減少了在資源受限環境中運行模型所需的成本，進一步擴展了轉移學習的實用性和應用範圍。

2.2.4 基於轉移學習的非幾何校正

數位雙生技術通過建立虛擬模型來模擬實際機器手臂的行為，更好地理解其運動特性。這種虛擬模型可以幫助預測和分析機器手臂在不同情況下的表現。另一方面，轉移學習則利用先前學到的知識來優化新任務或新環境中的性能，提高模型的泛化能力。根據 2.2.2 節所述，將機器學習引入機器人校正過程中通常需要大量的物理實驗數據。這些實驗不僅耗時，還會造成高昂的成本。為了提高效率並降低成本，可以採用轉移學習。轉移學習能夠減少實際物理實驗的需求，並加速機器學習模型在不同應用場景中的部署，從而有效提升整體性能和應用效果。Figure 2.11是轉移學習的基本流程，其可以有效利用已有的知識來解決新的問題。它包括以下幾個基本概念：

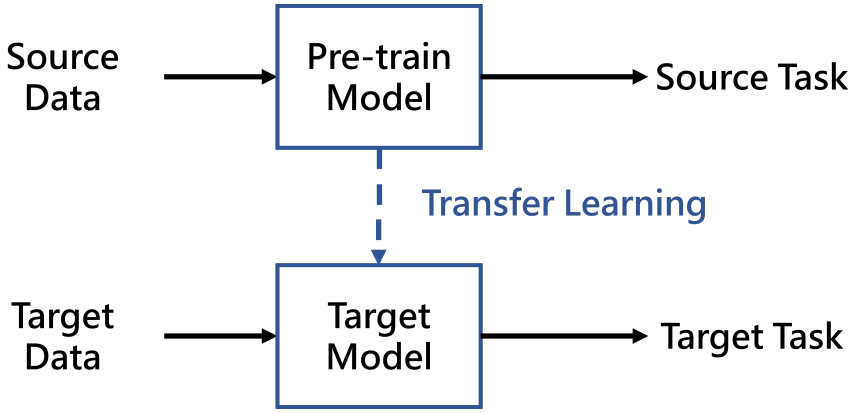


Figure 2.11: 轉移學習

- **源域與源任務 (Source domain and source task)**：源域是指包含大量標籤數

doi:10.6342/NTU202402735

據的領域，這些數據用於訓練初始模型，數據豐富且多樣化，涵蓋許多樣本和特徵。源任務是在源域上解決的具體任務，目的是提取有用的特徵和模式，這些特徵和模式可在目標任務中重用。

- **目標域與目標任務 (Target domain and target task)**：目標域是需要進行預測的領域，通常數據較少或標籤稀缺，雖然目標域數據與源域數據有所不同，但通常存在一定的相關性。目標任務是在目標域上解決的具體任務。由於目標域數據有限，直接訓練模型效果不佳。轉移學習旨在將源域中學到的知識應用到目標域，以提高目標任務的性能。

轉移學習的方法

轉移學習的基本概念是如何通過在一個領域中獲取的知識來增強另一個領域的學習過程。這種方法在解決現實世界中的多種問題時，特別是在數據稀缺的情況下，表現出了極大的潛力。通過適當的源域選擇和有效的轉移策略，可以大幅提升機器學習模型在各種應用場景中的表現，轉移學習的方法可以分為幾個主要類別：

1. **特徵轉移 (Feature transfer)**：這種方法旨在通過源域的數據學習一個有效的特徵表示，這些特徵能夠在目標任務中直接應用。通常使用預訓練模型的特徵提取層，這些層在大規模數據集上訓練過，能夠捕捉到普遍性的特徵。對於目標任務，這些特徵可用來進行分類或其他學習任務，無需從頭開始訓練模型。

2. **參數微調 (Fine-tuning)**：在已經訓練好的預訓練模型基礎上，然後在目標任務上進行參數微調的方法。具體做法是保持預訓練模型的大部分層凍結，只對最後幾層進行訓練，不過也有部分做法是，不凍結任何層。這樣可以利用

doi:10.6342/NTU202402735

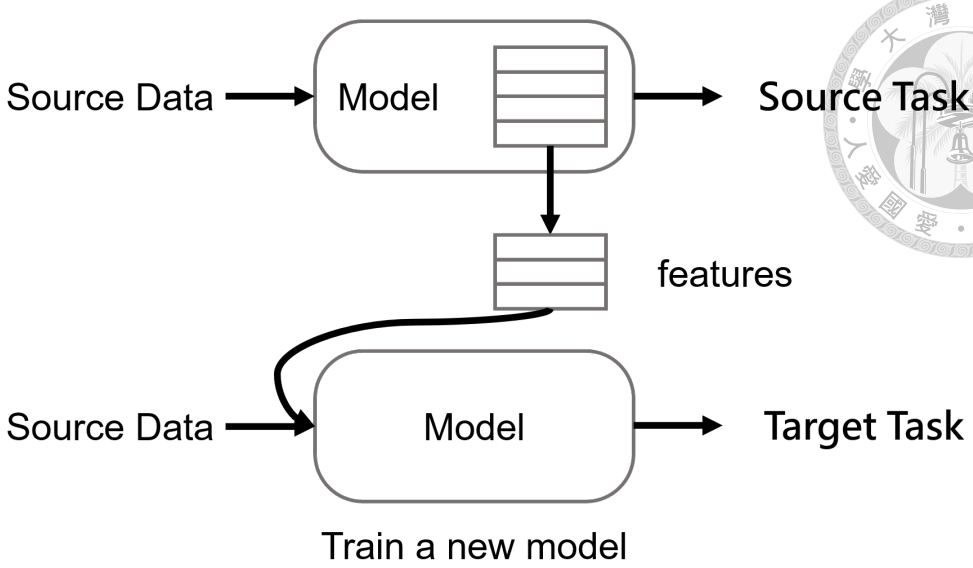


Figure 2.12: 特徵轉移

源任務學到的知識，並根據目標任務的數據進行細微的調整，從而提高模型在新任務上的性能。

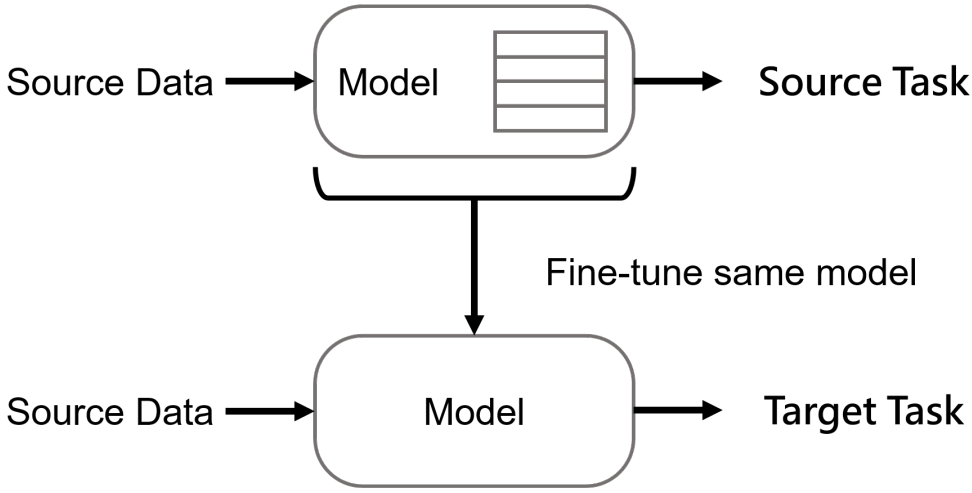


Figure 2.13: 參數微調

3. **模擬到現實 (Sim-to-Real transfer)**：模擬到現實的方法是首先在模擬環境中訓練模型，然後將其應用於現實世界的任務中。這種方法可以顯著減少在現實環境中收集數據的成本和時間，因為模擬環境是可控且無風險的，適合快速迭代和實驗，因為現實世界的數據收集可能涉及高時間與金錢成本。

4. **多領域學習 (Multi-domain learning)**：同時訓練多個相關任務的模型，通過

doi:10.6342/NTU202402735

共享特徵表示來提高所有相關任務的性能。這種方法旨在利用不同領域的數據，提高模型的泛化性。通過多任務學習，模型能夠學到更具代表性的特徵，從而在不同的任務中取得更好的結果。

基於本研究收集大量實際數據較難限制，本研究選擇了剪枝、微調和模擬到現實這三種技術來達成預測手臂末端非幾何誤差的目標。剪枝技術減少模型中的冗餘參數，提高效率和運行速度，使校正和預測更快速。微調利用預訓練模型中學到的特徵，針對具體應用場景進行調整，提高準確性和適用性。模擬到現實的轉移方法，在模擬環境中進行大量訓練和測試，降低數據收集成本和時間，並將訓練好的模型應用於現實，提升校正效率和準確性。這些技術的結合有效解決了數據稀缺問題，縮短了訓練時間，並提高了模型性能，實現了精確預測機械手臂末端非幾何誤差的目標。

轉移學習提高了效率並降低了成本，使機器人校準在實際應用中更可行。轉移學習能夠在不同應用場景中更快地部署機器學習模型，提高整體性能。通過利用源任務的豐富數據和特徵，轉移學習解決了數據稀缺問題，縮短了訓練時間，並提升了模型性能。

doi:10.6342/NTU202402735

第三章　RR 手臂示範演算法流程

為驗證算法的可行性，因此本章節3將著重在把算法應用在 RR 手臂上進行模擬的驗證，本研究會先進行兩種模擬實驗：

1. 證實利用 Jacobian 能夠校正手臂末端姿態
2. 證實利用機器學習校正手臂末端姿態的可行性

3.1　步驟說明

本次實驗將分為四個主要階段。Figure3.1為第一階段，使用運動學模型建構標稱手臂運動學模型 M_{FK}，基於此標稱手臂運動學模型，建構另一模型含有非幾何變化，稱其為手臂物理模型 M。這一步驟涉及詳細描述手臂的運動學參數和物理特性，確保模型準確反映實際情況。

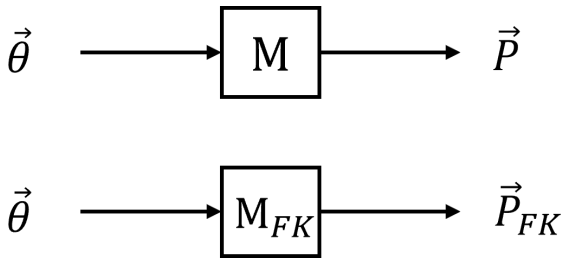


Figure 3.1: 手臂末端資料蒐集

Figure3.2為第二階段，進行 DH 參數的校正。將使用 Jacobian 演算法，利用標稱 DH 參數以及從物理模型蒐集到的末端姿態數據 $\vec{P}$、$\vec{P}_{FK}$ 進行參數校正，從而精確調整手臂模型 M_{FK}，使之更貼近真實手臂的運動特性。

doi:10.6342/NTU202402735

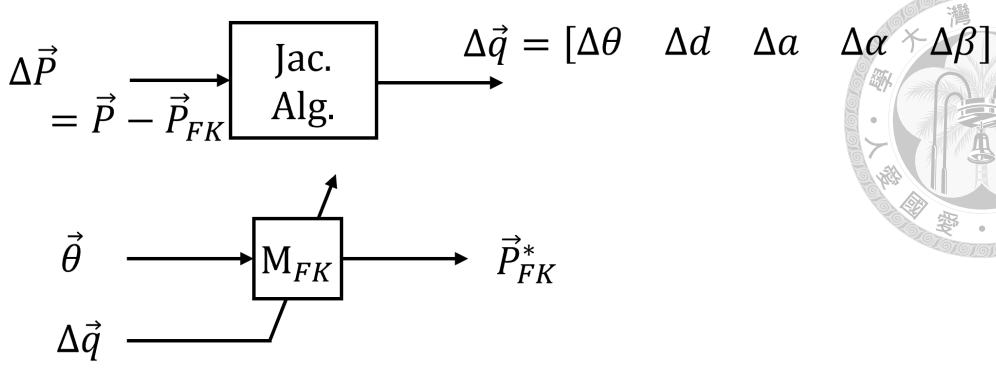


Figure 3.2: Jacobian 算法

Figure3.3為第三階段，會以校正後的運動學模型及物理模型的誤差作為機器學習的學習目標。具體而言，將利用這些誤差數據訓練機器學習模型，使其能夠預測並校正手臂運動中的偏差。

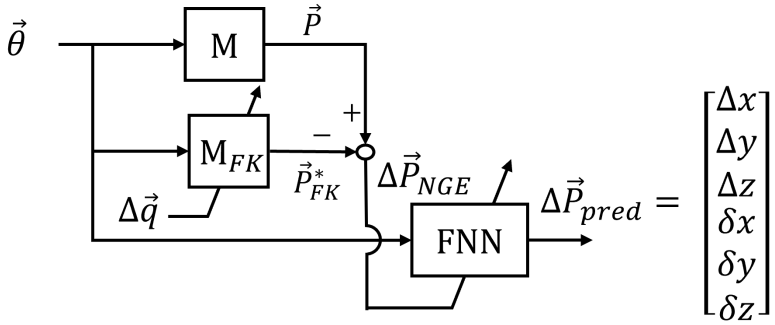


Figure 3.3: 模型訓練

Figure3.4為第四階段，將把機器學習模型與校正後的運動學模型結合，進行綜合評估，觀察整合模型後的誤差是否有所縮小，從而驗證整體方法的有效性。

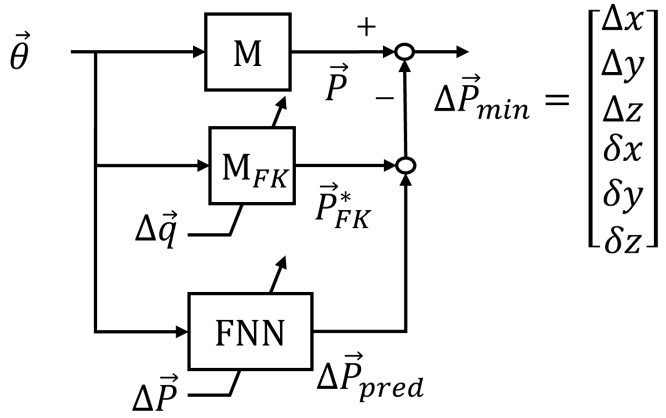


Figure 3.4: 驗證流程圖

doi:10.6342/NTU202402735

3.2 RR 手臂的模型建置

為了驗證算法，必須對手臂做兩種建模，分別為標稱手臂運動學模型以及含有誤差的 3D 的手臂物理模型。

3.2.1 運動學模型

首先，建立出由兩個旋轉關節與兩個連桿組成的手臂，稱其為 RR 手臂，延續章節 2.1的運動學建模方式，依照 Standard 定義之法則定義出架構 Figure 3.5。

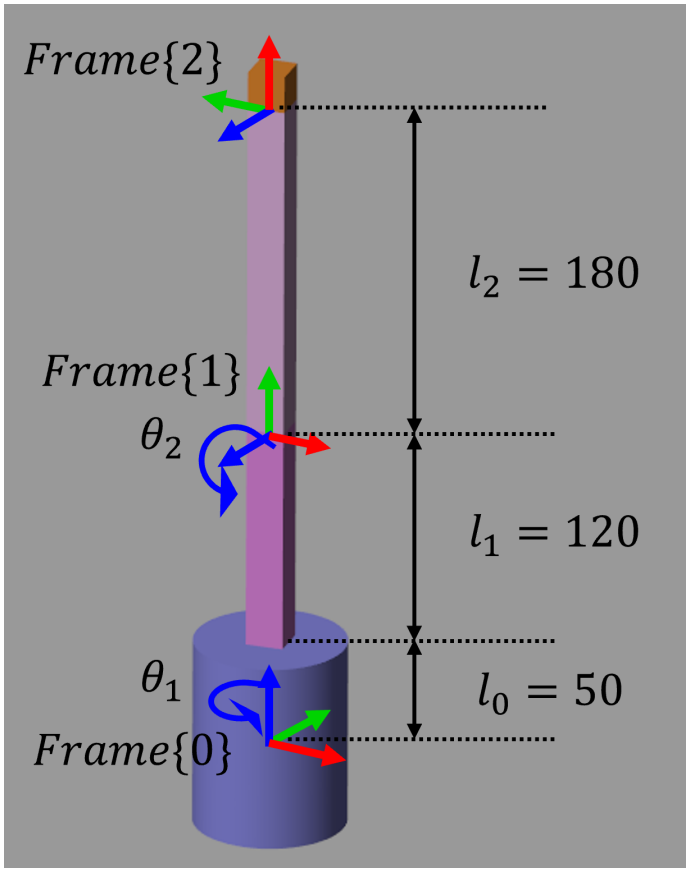


Figure 3.5: RR 手臂架構圖

依照 Figure 3.5之規格，即可描述 RR 手臂的 DH table，此理論值描述的運動學模型，稱其為 Nominal DH table，假設誤差 DH error 為 Table3.1，將此誤差加在 Nominal DH table 作為 Actual DH table，即完成目標要識別的運動學模型。

doi:10.6342/NTU202402735

Table 3.1: RR 手臂的 Nominal DH 表

i	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)	β_i (rad)
1	θ_1	0	$50+120$	$\frac{\pi}{2}$	0
2	$\theta_2+\frac{\pi}{2}$	180	0	0	0

Table 3.2: RR 手臂的 Actual DH 表

i	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)	β_i (rad)
1	$\theta_1+0.001$	0.16	170.17	1.572	0
2	$\theta_2+1.575$	180.12	0.14	0.017	0

Table 3.3: RR 手臂的誤差表

i	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)	β_i (rad)
1	0.001	0.16	0.17	0.002	0
2	0.005	0.12	0.14	0.017	0

3.2.2 物理模型

除了運動學模型以外，為了能夠更真實地模擬手臂的非幾何誤差，本研究選擇在 Simulink 中建立了一個 3D 模型 Figure 3.6。這個模型利用了多種 Simulink 中的模組和方塊來實現：

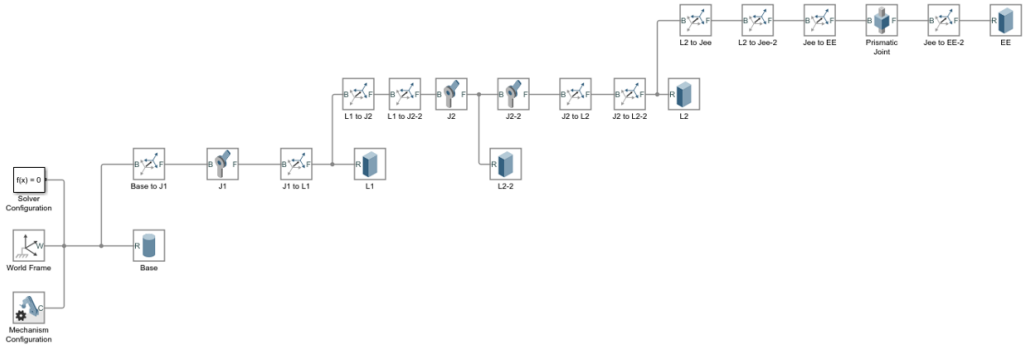


Figure 3.6: RR 手臂 Simulink 的剛體樹模型

- **剛體變換模塊 (Rigid transform block)**：

  實現從一個參考座標系到另一個座標系的剛體變換。在機器手臂模擬中，它用來表示機器手臂中各個關節之間的連接關係，以及末端執行器相對於基座的位置和姿態。

doi:10.6342/NTU202402735

- **旋轉關節模塊 (Revolute joint block)**：

  模擬機器手臂的旋轉關節。可以設置旋轉關節的初始角度、運動範圍、運動速度等參數，以模擬真實機器手臂的關節運動。此模塊也被使用來設定彈簧係數、阻尼係數，來模擬手臂在真實世界中的非幾何誤差。

- **GetTransform**：

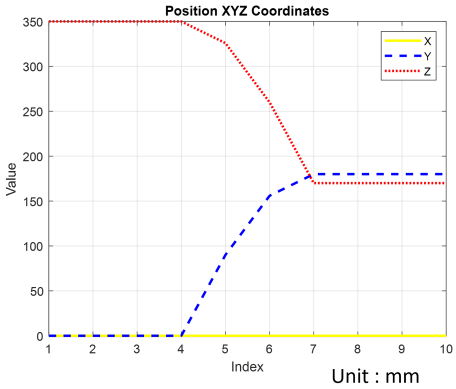
  - **功能**：GetTransform 模塊主要用於從 Simulink 模型中獲取剛體變換 (rigid body transform) 的數據。這些數據可以包括機器手臂各個關節或末端執行器的位置 (Position) 和旋轉姿態 (Rotation)。
  - **Rigid body 的使用**：在 Simulink 中，可以通過配置 GetTransform 模塊來設定需要獲取的剛體變換。例如，可以指定特定的關節或末端執行器，並指定相應的基座 (Base) 和目標座標系 (Target)。
  - **Base 和 Target 的關聯性**：在設置 GetTransform 模塊時，需要明確指定 Base 和 Target。Base 是參考座標系，通常是機器手臂基座的座標系；Target 則是相對於 Base 而言的目標座標系，可以是機器手臂末端執行器的座標系或是某個關節的座標系。GetTransform 模塊會計算並輸出 Base 到 Target 的姿態。

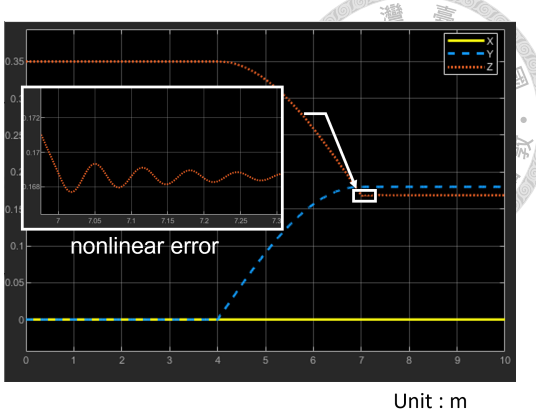
- **ToWorkspace**：

  將 Simulink 模型中的數據輸出到 MATLAB 工作區 (Workspace) 中。這使得模擬數據可以進行後續分析、可視化或保存，進而協助手臂校正的計算。

這些 Simulink 模塊不僅使得機器手臂動態模擬更直觀和可控，還能夠有效進行非幾何誤差的模擬，Figure 3.7 和 Figure 3.8分別是末端點位置以及旋轉的輸出。

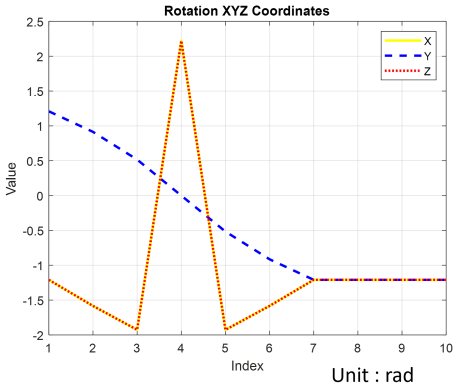
doi:10.6342/NTU202402735

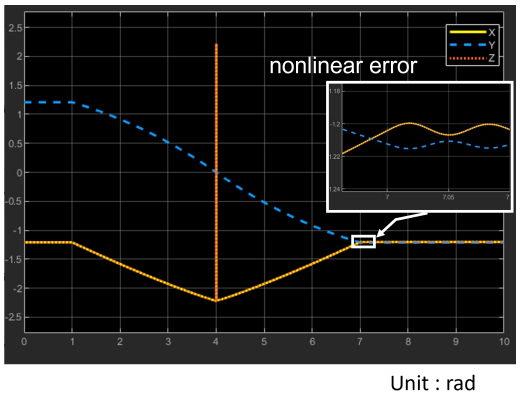




(a) 運動學輸出　　(b) Simulink 輸出

Figure 3.7: RR 手臂末端位置座標 (a) 運動學輸出 (b) Simulink 輸出





(a) 運動學輸出　　(b) Simulink 輸出

Figure 3.8: RR 手臂末端姿態座標 (a) 運動學輸出 (b) Simulink 輸出

將 Simulink 的 Scope 放大觀察，可以發現 Figure 3.7b 在 z 軸方向上有震盪，此現象同時也可以在 Figure 3.8b裡面觀察到。這證實了在使用旋轉關節模塊 (Revolute joint block) 時，設定的彈簧係數和阻尼係數成功地造成了系統的不穩定，從而模擬非幾何誤差。因此，透過上述介紹的模組和方塊結合使用，Simulink 可以實現對機器手臂非幾何誤差的高度精確建模，使模擬結果更加真實和可靠。

3.3　基於平行架構的校正

以下將介紹，如何在 RR 手臂上進行，平行架構演算法的模擬，先透過傳統校正法將幾何誤差補償到標稱手臂運動學模型 M_{FK}，再將剩餘的非幾何誤差透過 FNN 模型訓練出來，以找到一個平行架構模型能夠最小化 $\Delta\vec{P}_{min}$ 的任務。

doi:10.6342/NTU202402735

3.3.1 DH 參數校正

本節使用了2.2.1節所介紹的 Jacobian 校正演算法來進行運算，不過此處並未加入 $\Delta\beta$，僅使用原有的 $[\Delta\theta \quad \Delta d \quad \Delta a \quad \Delta\alpha]$，Jacobian 校正的流程圖如 Figure 2.4。其中 Actual DH 即為表3.2，Nominal DH 則為表3.1，分別將兩個運動學的模型代入齊次轉換矩陣算出手臂物理模型的輸出 $\vec{P}$ 以及標稱手臂運動學模型的輸出 $\vec{P}_{FK}$，兩者相減算出 $\Delta\vec{P}$，即可將公式 (2.10) 重新寫成公式 (3.1)。

$$\begin{aligned}\Delta\vec{P} &= \vec{P} - \vec{P}_{FK} \\ &= J\Delta\vec{q} \\ &= \begin{bmatrix} M_\theta & M_d & M_a & M_\alpha \\ R_\theta & 0 & 0 & R_\alpha \end{bmatrix} [\Delta\theta \quad \Delta d \quad \Delta a \quad \Delta\alpha]^T\end{aligned} \tag{3.1}$$

經過 Jacobian 校正，即可得知應該要被更新的參數 $\Delta\vec{q}$ 的值分別為多少，並依序將 $[\Delta\theta \quad \Delta d \quad \Delta a \quad \Delta\alpha]^T$，更新至原本的 Nominal DH 表3.1，並再重新進行運算即可不斷得知各個更新的參數值，本研究做了三次的更新實驗，結果如表3.4、表3.5。由實驗果可以知道所模擬出的數值越來越接近誤差值，也將每次迭代後更新的值做紀錄，並且且輸出成以下兩組，分別為 Figure 3.9迭代 10 次與 Figure 3.10迭代 100 次，其中較細的虛線為要趨近的值，較粗的虛線則為每次迭代後更新的值。

Table 3.4: 經過 Jacobian 校正後第一組參數更新值

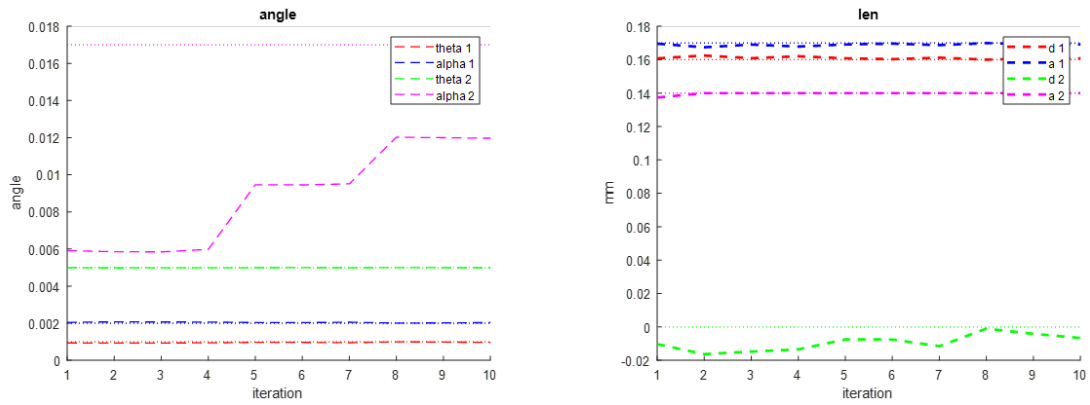
Iteration	θ_1 (rad)	d_1 (mm)	a_1 (mm)	α_1 (rad)
	0.001	0.16	0.17	0.002
1	0.000942	0.1608	0.1696	0.002
10	0.000971	0.1609	0.1691	0.002
100	0.001	0.16	0.17	0.002

可以發現原本經過 Jacobian 算法一次運算特別不準的 a_2、d_2，透過迭代的方

doi:10.6342/NTU202402735

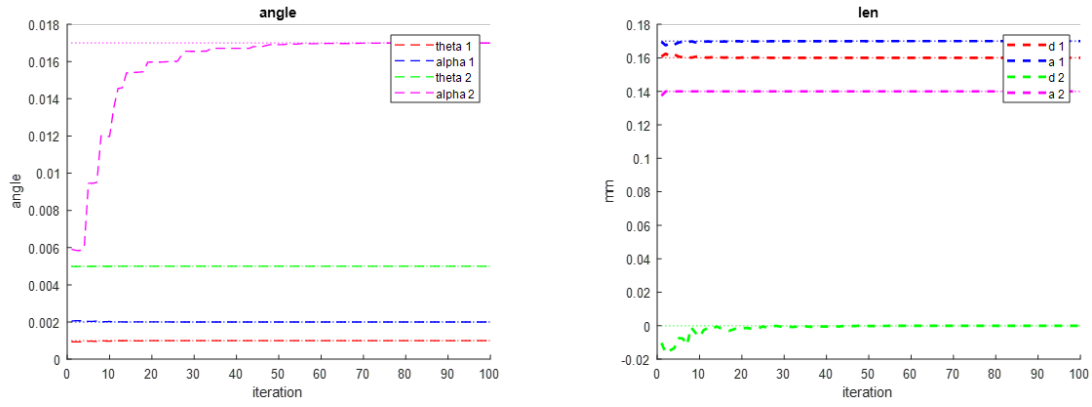
Table 3.5: 經過 Jacobian 校正後第二組參數更新值

Iteration	θ_2 (rad)	d_2 (mm)	a_2 (mm)	α_2 (rad)
	0.005	0	0.14	0.017
1	0.005	-0.0103	0.1374	0.0059
10	0.005	-0.0066	0.14	0.012
100	0.005	-0.00005	0.14	0.017



(a) 角度　　(b) 長度

Figure 3.9: 經過 10 次迭代參數應該更新的值 (a) 角度 (b) 長度
(粗虛線：真實值；細實線：迭代的計算值)



(a) 角度　　(b) 長度

Figure 3.10: 經過 100 次迭代參數應該更新的值 (a) 角度 (b) 長度
(粗虛線：真實值；細實線：迭代的計算值)

式都有更加趨近應該要估算出的誤差值，而且大概經過 50 次迭代後，每一項誤差值已經趨近穩定。

doi:10.6342/NTU202402735

3.3.2 類神經網路校正

本節使用了2.2.2節所介紹的機器學習校正演算法來驗證機器學習可以用來校正手臂的非幾何誤差，流程圖如 Figure 3.11，機器學習的步驟包括數據收集、數據預處理、選擇模型、訓練模型、評估模型和進行預測。

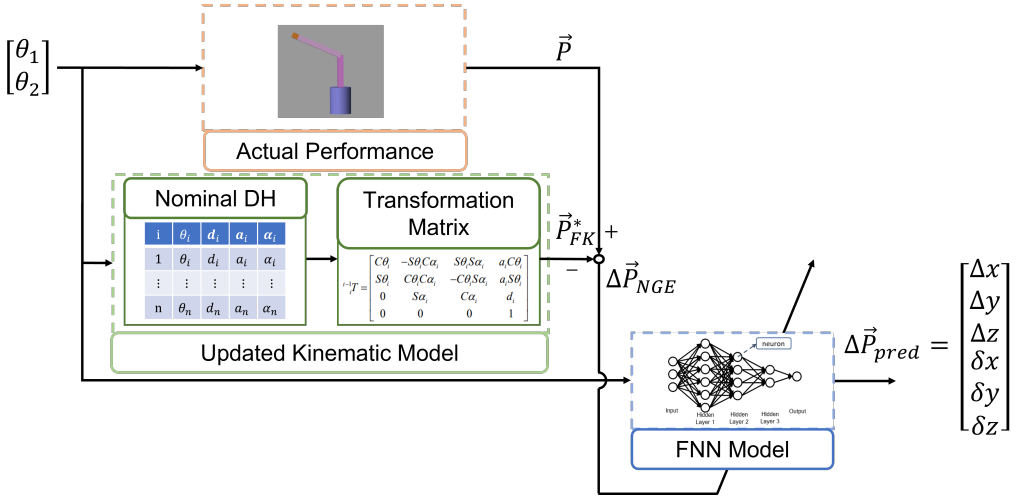


Figure 3.11: RR 手臂的機器學習流程示意圖

首先資料收集與處理的部分按照 Figure 3.11中的做法，先收集由3.2.2節建立的 Simulink 物理模型之姿態 $\vec{P}$，此處假設運動學模型的參數已經被更新完畢，並且算出被更新過後的末端姿態 $\vec{P}^*_{FK}$，將兩者相減得到的 $\Delta\vec{P}_{NGE}$ 即為神經網路模型要學習的目標任務。

- **環境設置和數據加載**

  首先，導入所需的 Python 函式庫，包括 Pandas、NumPy、Matplotlib、Scikit-learn 和 Keras。這些函式庫提供了強大的數據處理、數據可視化和機器學習建模功能。使用 np.loadtxt 函數從 CSV 文件中載入數據集。數據集包括關節角度 θ、末端執行器位置和旋轉的誤差量 $\Delta\vec{P} = [\ \Delta x \quad \Delta y \quad \Delta z \quad \delta x \quad \delta y \quad \delta z\ ]$。

doi:10.6342/NTU202402735

- **數據處理**

將數據集分割為不同部分：關節角度 (q)、末端執行器位置 (Position) 和末端執行器姿態 (Rotation)。使用 Scikit-learn 中的 `train_test_split` 函數將數據集分為訓練集、驗證集和測試集。訓練集用於訓練模型，驗證集用於在訓練過程中評估模型的性能，測試集則用於最終評估模型的泛化能力。

Table 3.6: RR 手臂 A 訓練資料集切割後資料數

Dataset (type)	Input (num of samples, dimensions)	Output (num of samples, dimensions)
Training	(7960, 2)	(7960, 6)
Validation	(995, 2)	(995, 6)
Testing	(995, 2)	(995, 6)

- **模型構建**

本研究構建了一個前饋神經網路 (FNN)，這是一種常見的神經網路架構，適用於回歸和分類問題，模型架構如 Figure3.12，其包含三個隱藏層，每層分別有 128、64 和 32 個神經元，並使用 ReLU 作為激活函數，這有助於解決非線性問題。輸出層包含 6 個神經元，對應於末端執行器的位置和姿態參數。模型使用平均絕對誤差 (MAE) 作為損失函數，並選擇 Adam 優化器進行優化，Adam 優化器能夠自動調整學習率，提高模型的收斂速度和效果。

```
Model: "sequential_16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_64 (Dense)            (None, 128)               384

 dense_65 (Dense)            (None, 64)                8256

 dense_66 (Dense)            (None, 32)                2080

 dense_67 (Dense)            (None, 6)                 198

=================================================================
Total params: 10918 (42.65 KB)
Trainable params: 10918 (42.65 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 3.12: RR 手臂 FNN 架構

doi:10.6342/NTU202402735

- **模型訓練**

在訓練過程中，將模型在訓練集上進行 50 個訓練週期 (Epochs)，每個批次 (Batch) 包含 32 個樣本。在每個訓練週期結束後，使用驗證集來評估模型的性能。這樣做的目的是為了監控模型的訓練過程，防止過擬合現象發生。通過觀察訓練損失和驗證損失的變化，可以了解模型的學習狀況，並及時調整參數。

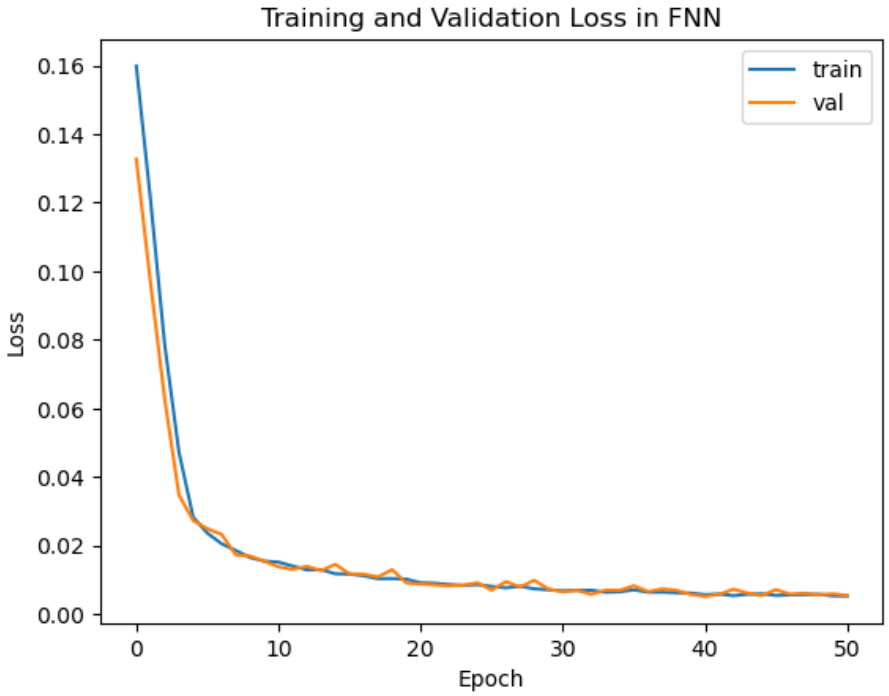
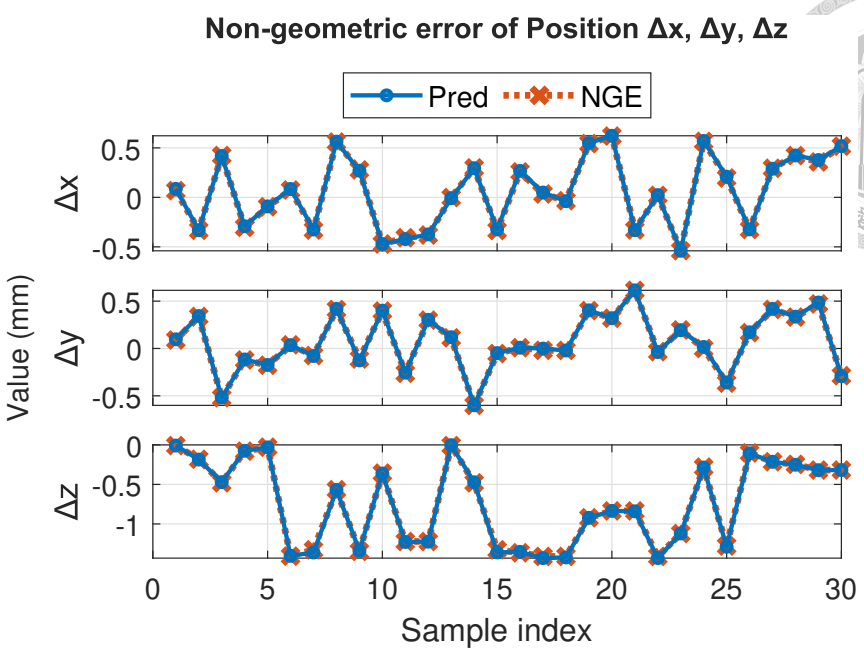


Figure 3.13: RR 手臂學習曲線

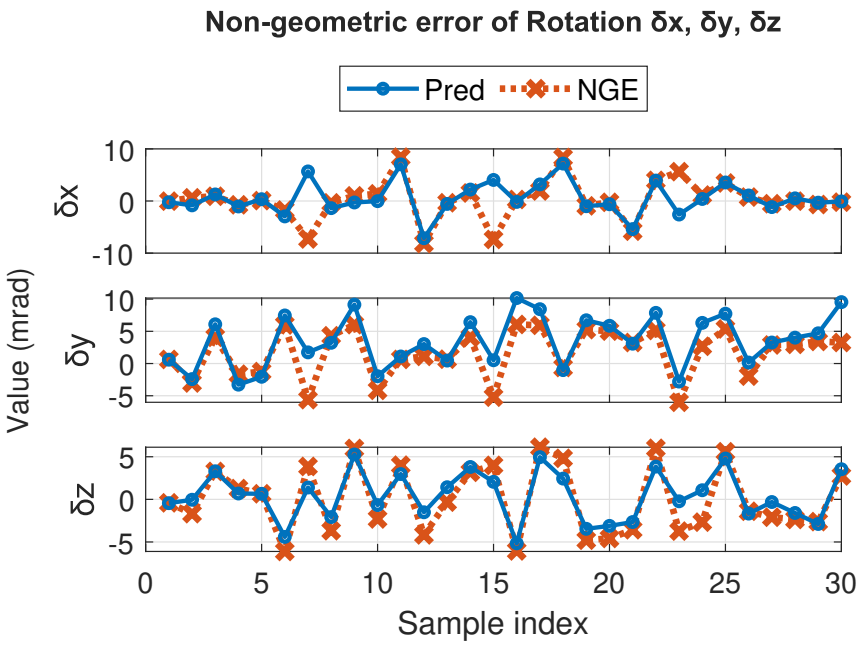
- **模型評估與分析**

訓練完成後，在驗證集上評估模型的性能，驗證損失。使用 Matplotlib 庫將訓練損失和驗證損失的變化過程可視化，可以直觀地觀察模型的收斂情況。如果驗證損失顯著高於訓練損失，可能意味著模型存在過擬合問題。Figure3.14是將非幾何誤差 (NGE) 以及後模型預測的誤差放在一起比較，可以發現擬和的相似度非常高。

doi:10.6342/NTU202402735



(a) 位置



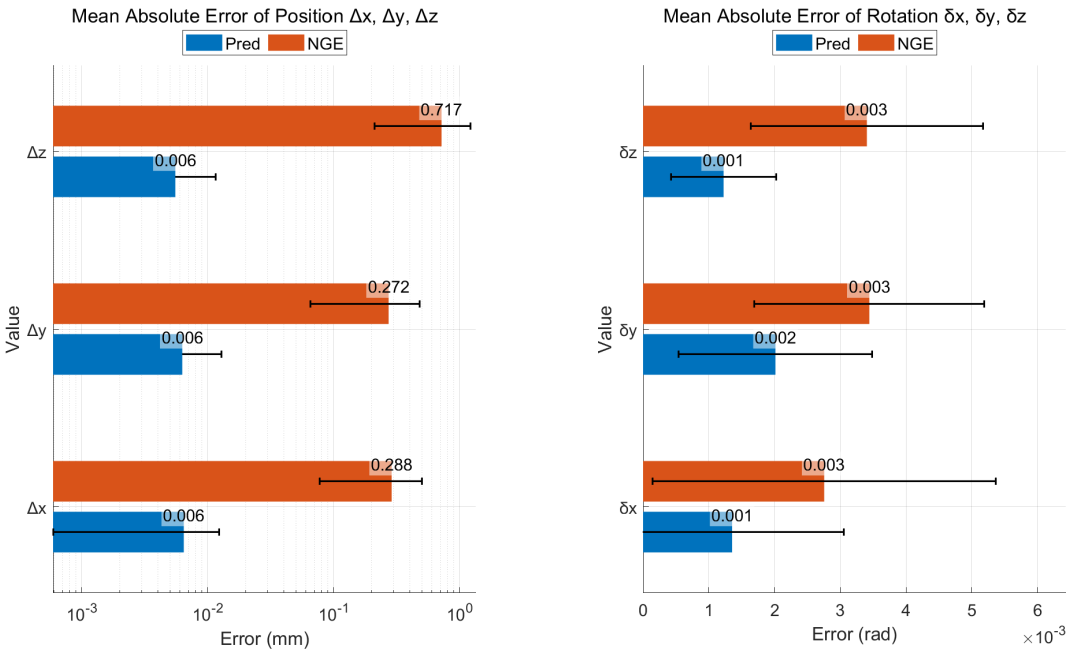
(b) 旋轉

Figure 3.14: 隨機取樣 30 個點之預測值和真實值 (a) 位置 (b) 旋轉

- **誤差分析**

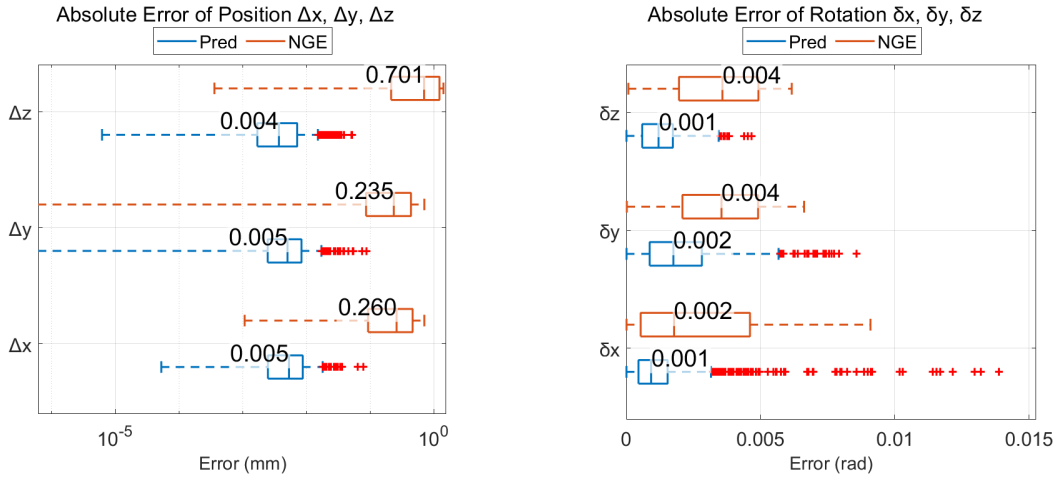
接下來，將使用測試集的輸入對模型進行預測，並計算平均絕對誤差 (MAE)。MAE 是一種常用的誤差指標，用於衡量模型預測結果的精度，Figure 3.15即為非幾何誤差與預測值的 MAE，其中黑色的線條為誤差棒 (Error bar)，其長度等同於該項數據的標準差。

doi:10.6342/NTU202402735



(a) 位置 (b) 旋轉

Figure 3.15: 測試資料集預測值和真實值的平均絕對誤差 (a) 位置 (b) 旋轉



(a) 位置 (b) 旋轉

Figure 3.16: 測試資料集預測誤差和真實值的箱形圖 (a) 位置 (b) 旋轉

Figure3.16為箱形圖 (Box plot)，將資料的散佈狀況顯示出來，其中箱子本體標示了依序為數字由小到大排列後，分別為 25%、50%、75% 的數字，其名稱依序稱作 Q_1、Q_2、Q_3。上下界線則是將 Q_1、Q_3 分別減去和加上 1.5 倍四分位距 (InterQuartile Range, IQR) 的值，IQR 又為 $Q_1 - Q_3$。而紅色十字表示離群值 (Outliers) 即超過 1.5 倍四分位距的數據點。通過這些誤差指標，可以全面評估模型的性能，確保其在實際應用中的可靠性。

doi:10.6342/NTU202402735

這些步驟展示了使用 Python 和 TensorFlow 進行機器學習的完整流程，包括數據加載、處理、模型構建、訓練、評估和誤差分析。這些操作不僅有助於驗證和優化機器手臂的控制策略，還能提高機器手臂在複雜工作環境中的表現。

3.3.3 使用轉移學習進行的數位雙生模型校正

參照3.3.2節所訓練出來的模型，將其稱為 Model A，且此模型使用的資料集來自於物理模型手臂 A。為了展現如何使用 Pruning 與 Fine-Tune 技術，同樣使用3.2.2節的物理模型，並且稍微調整其關節上的彈簧係數以及阻尼係數，使其成為手臂 B，詳細參數如表3.7，並且收集較少的資料集來做訓練。實際轉移學習流程如 Figure 3.17所示。

Table 3.7: 手臂物理模型的參數設定

手臂	彈簧係數 (N·m/rad)	阻尼係數 (N·m/(deg/s))	資料集數量
Meca500 A	10	2e-4	36572
Meca500 B	5	1e-4	1528

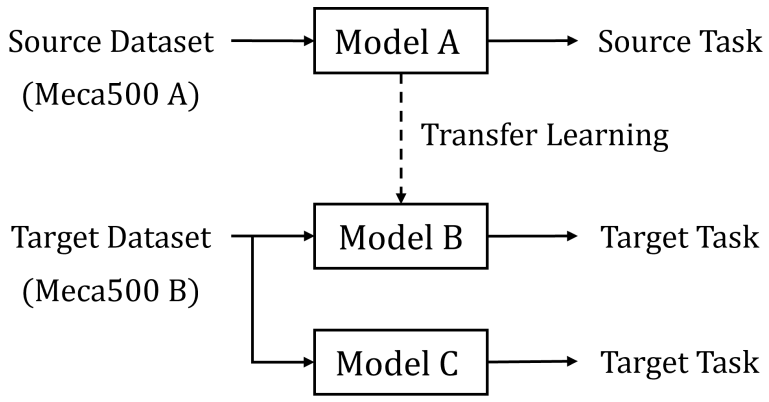


Figure 3.17: 轉移學習示意圖

- **數據預處理**：同樣按照3.3.2節使用 `np.loadtxt` 函數從事先準備好的 CSV 文件中加載數據集。這些數據包括手臂 B 的關節角度和末端執行器的位置及姿態信息，這些數據將用於訓練的模型。

doi:10.6342/NTU202402735

Table 3.8: RR 手臂 B 訓練資料集切割後資料數

Dataset (type)	Input (num of samples, dimensions)	Output (num of samples, dimensions)
Training	(321, 2)	(312, 6)
Validation	(39, 2)	(39, 6)
Testing	(39, 2)	(39, 6)

- **預訓練基礎模型**：作為基礎模型，本研究選擇3.3.2節所訓練出來的 Model A。這個模型已經通過大量數據訓練並且具有適應各種運動學校正任務的能力。使用 Keras 的 `load_model` 函數，載入這個預訓練模型，確保其網路架構和權重與先前訓練的結果一致。

- **參數修剪**：

  為了進一步優化模型的性能和效率，引入了 TensorFlow Model Optimization 工具包中的參數修剪技術。這項技術廣泛應用於深度學習模型，旨在減少 Model A 中冗餘參數的數量，以實現模型瘦身和加速。

  首先，設置初始稀疏度為 5% 和最終稀疏度為 30%，這些值是基於模型性能和稀疏度之間的權衡考慮設定的。接著，使用函數設置稀疏度的變化規則此處選擇 `tfmot.sparsity.keras.PolynomialDecay` ，這是一種多項式衰減函數，能夠根據訓練步驟動態調整稀疏度。Figure 3.18則為剪枝過程的函數。

  最後，利用 `tfmot.sparsity.keras.prune_low_magnitude` 函數將參數修剪應用到預先訓練的模型上。這個函數在每個訓練步驟中根據設置的稀疏度進行修剪，從而產生一個瘦身後的模型。觀察 Figure 3.19可以發現修剪前的手臂模型參數較多，這是因為在修剪時，多給模型多一層的遮罩，這些遮罩用來記錄權重是否要被更新，Figure 3.20中可以觀察到原本所有位置都有權重值，但經過剪枝，被修改為 0。

doi:10.6342/NTU202402735

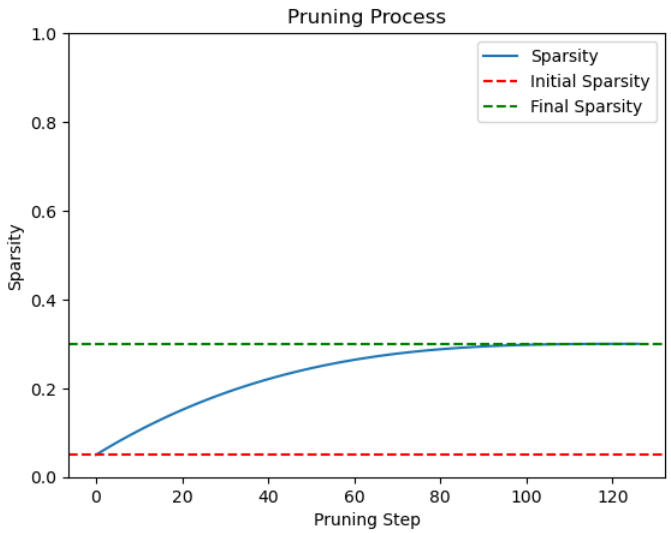


Figure 3.18: RR 手臂 A 剪枝過程

```
 Layer (type)                Output Shape              Param #
=================================================================
 prune_low_magnitude_dense_  (None, 128)               642
 64 (PruneLowMagnitude)

 prune_low_magnitude_dense_  (None, 64)                16450
 65 (PruneLowMagnitude)

 prune_low_magnitude_dense_  (None, 32)                4130
 66 (PruneLowMagnitude)

 prune_low_magnitude_dense_  (None, 6)                 392
 67 (PruneLowMagnitude)

=================================================================
Total params: 21614 (84.45 KB)
Trainable params: 10918 (42.65 KB)
Non-trainable params: 10696 (41.80 KB)
_________________________________________________________________
Pre-pruning non-zero weights percentage: 100.00%
```

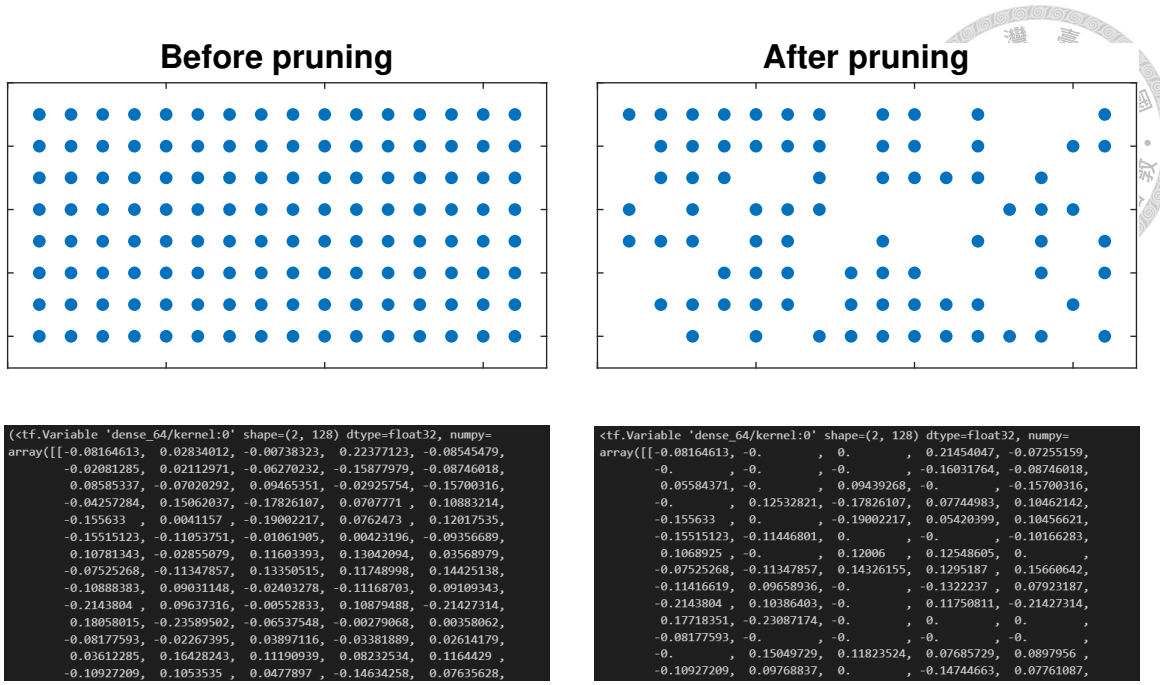
(a) 剪枝前的手臂 A

```
 Layer (type)                Output Shape              Param #
=================================================================
 dense_64 (Dense)            (None, 128)               384

 dense_65 (Dense)            (None, 64)                8256

 dense_66 (Dense)            (None, 32)                2080

 dense_67 (Dense)            (None, 6)                 198

=================================================================
Total params: 10918 (42.65 KB)
Trainable params: 10918 (42.65 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Post-pruning non-zero weights percentage: 70.21%
```

(b) 剪枝後的手臂 A

Figure 3.19: RR 手臂剪枝模型 (a) 剪枝前的手臂 A (b) 剪枝後的手臂 A

doi:10.6342/NTU202402735



(a) 剪枝前　　(b) 剪枝後

Figure 3.20: Model A 剪枝前後模型第一層第一列的權重分布與其值
(a) 剪枝前 (b) 剪枝後

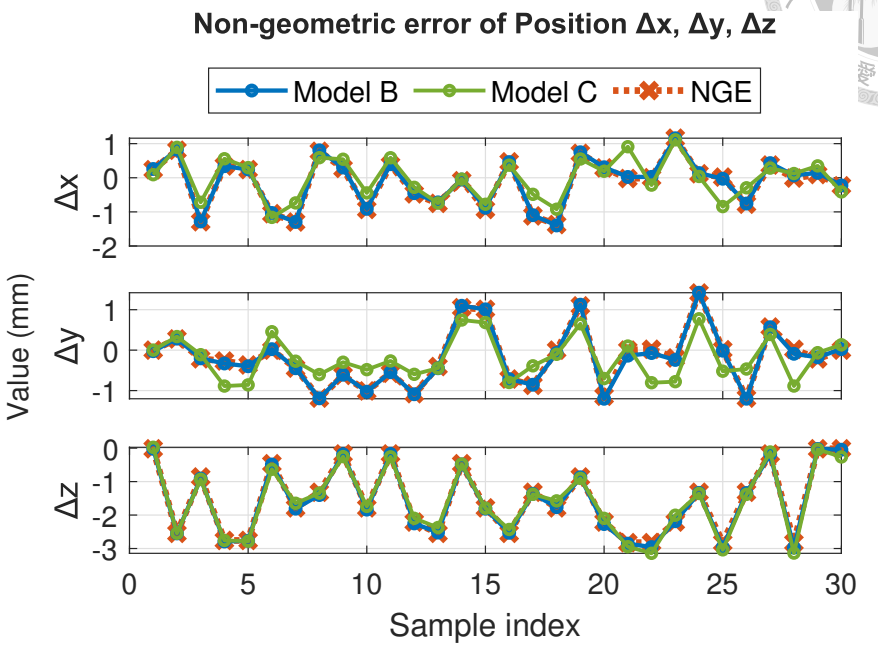
- **修剪後模型的訓練**：在對修剪後的模型重新編譯後，使用訓練集和驗證集來訓練模型 B。在訓練過程中，同時利用回調函數來更新修剪步驟並監控模型的性能。

- **效能評估和結果分析**：在訓練過程結束後，本研究進一步使用試誤法（Trial and error），重新使用損失函數 Huber loss 和 RME 對模型進行訓練，以期望在目標任務上取得更優異的表現。針對 Huber loss 中的權衡參數 δ，採用經驗法則和交叉驗證的方法進行選擇。根據經驗法則，選取數據標準差的 $\frac{1}{5}$ 和 $\frac{1}{10}$ 作為 δ 的候選範圍。隨後，通過交叉驗證在驗證集上測試不同 δ 值的效果，並依據驗證損失挑選最佳的 δ。結果顯示，使用 Huber loss 比 MAE 和 RMS 取得了更好的表現，證明了其在處理離群點和提高模型精度方面的優勢。最後得到的 Model B，即為使用 Huber loss 進行訓練的模型。

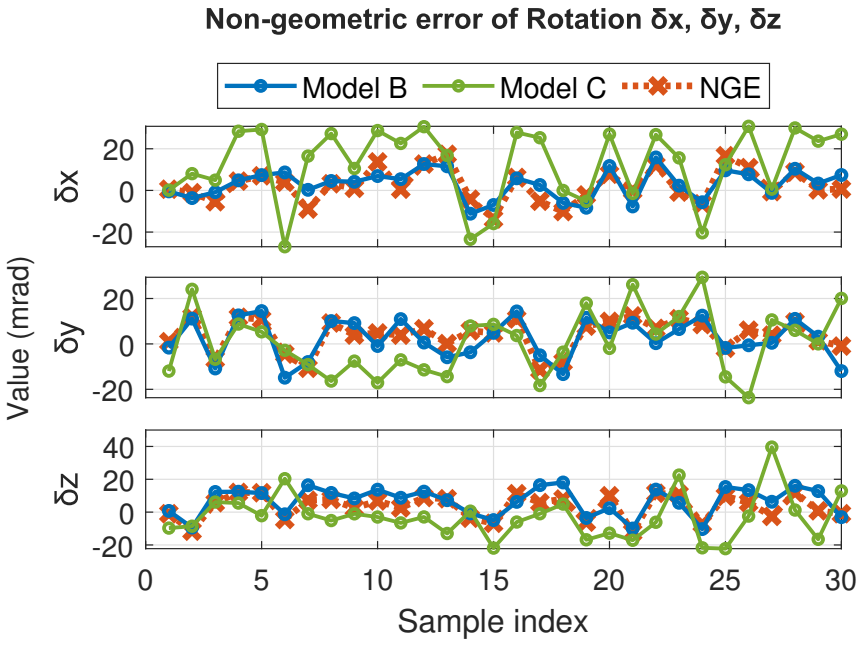
為了驗證此 Model B 的性能，額外在相同的模型架構底下，訓練了另一個 Model C 但此模型並沒有使用轉移學習。觀察 Figure 3.21，可以發現對於位

doi:10.6342/NTU202402735

置的部分兩個模型都預測的不錯，但 model B 因為有使用轉移學習，所以在旋轉的部分預測的表現比 model C 還要好。



(a) 位置

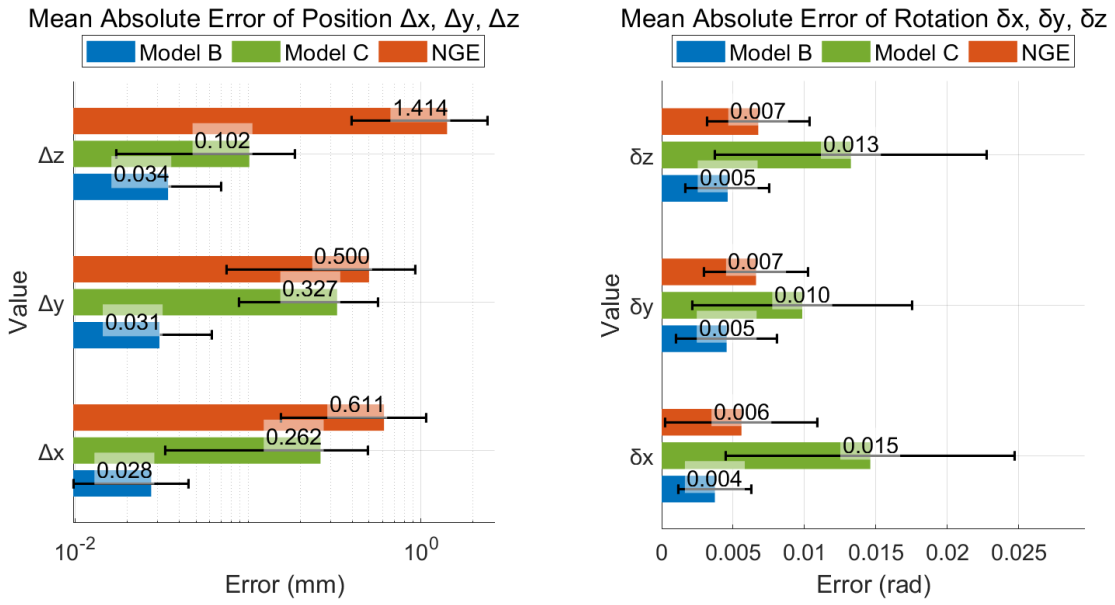


(b) 旋轉

Figure 3.21: Model B 和 Model C 隨機取樣 30 個點之預測值 (a) 位置 (b) 旋轉

doi:10.6342/NTU202402735

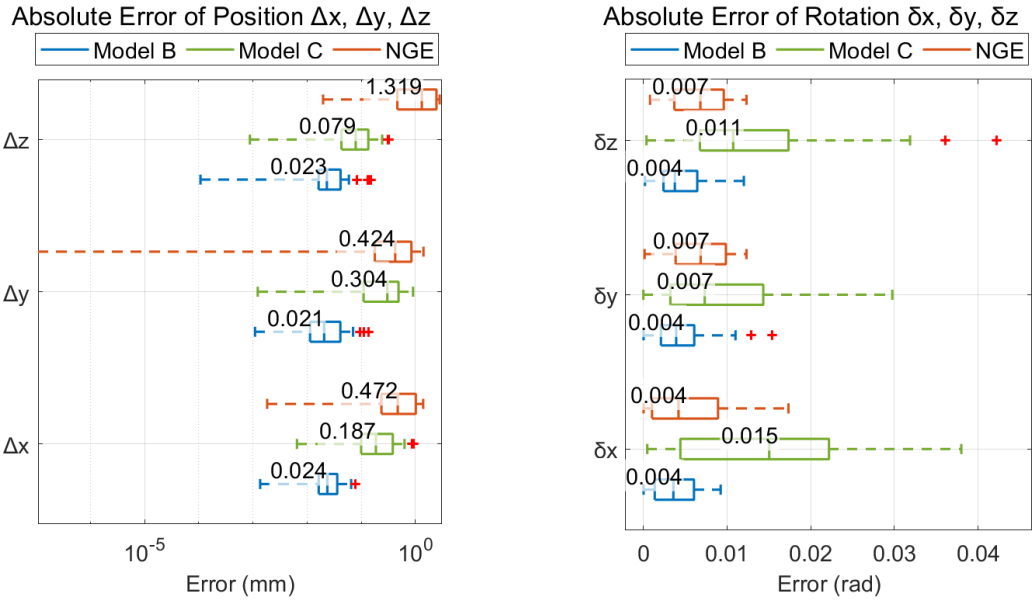
另外將 Model B 和 Model C 對所有測試資料集的平均絕對誤差進行比較，如 Figure 3.22，結果顯示，Model B 在預測這些資料時的表現優於 Model C。



(a) 位置　　(b) 旋轉

Figure 3.22: Model B 和 Model C 在測試資料的平均絕對誤差 (a) 旋轉 (b) 位置

同時也將箱形圖拿 Figure 3.23來做比較，反映出兩個模型在預測誤差時，每一筆數據的分布情況，也會發現 Model B 在預測這些資料時能夠降低整個誤差的分散程度，同時也降低了最大值的位置。



(a) 位置　　(b) 旋轉

Figure 3.23: Model B 和 Model C 在測試資料的預測值之箱形圖 (a) 位置 (b) 旋轉

doi:10.6342/NTU202402735

總結來說，這些步驟不僅展示了如何利用轉移學習和參數修剪技術來優化深度學習模型的性能，還提供了一個清晰的方法來理解研究方法的具體細節及其在工程應用中的重要性。這對於本研究的讀者來說，提供了一個實際且有效的方法來改進機器人運動學校正模型的精度和效率。

3.4 與轉軸方向不同的扭轉彈簧

參考3.2.2節的 Figure 3.6，由於在第二個關節的 Revolute Joint Block 上所設定的彈簧係數是跟該轉軸的方向相同之扭轉彈簧 (torsion spring)，如同 Figure 3.25a所表示，其受到之扭矩 τ_{z_i} 與轉軸 z_i 相同，且扭矩受到彈簧係數 K 與阻尼係數 C 影響，而此彈簧造成的扭矩也已經有傳統的校正法能夠進行彌補 [12]，因此為了驗證機器學習強大的非線性函數擬和能力，可以面對來自不同方向的扭矩，如同 Figure 3.25b所表示，扭矩受到三組彈簧係數與阻尼係數的影響，分別是 τ_{x_i}、τ_{y_i}、τ_{z_i} 三個方向的扭矩，這些係數影響了非幾何誤差。3.4節基於前面 RR 手臂的架構，分別在第二個關節的座標系之 R_x、R_y 上額外的扭轉彈簧，如 Figure 3.24。

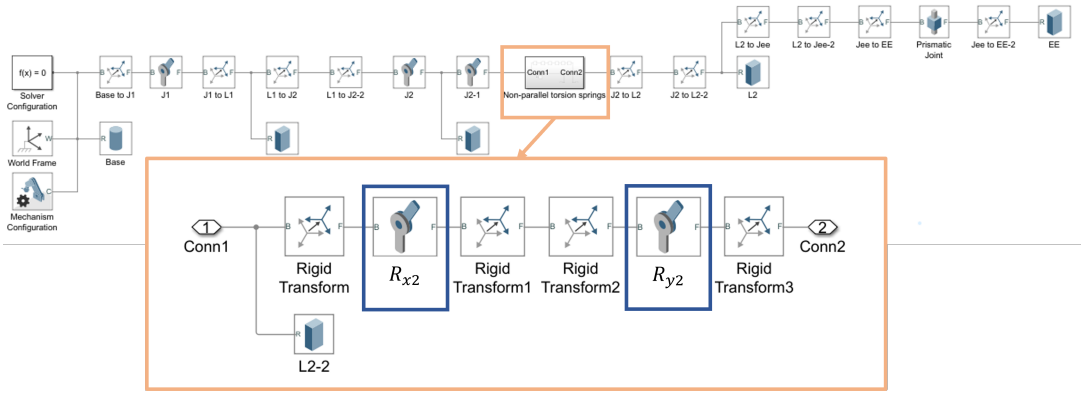
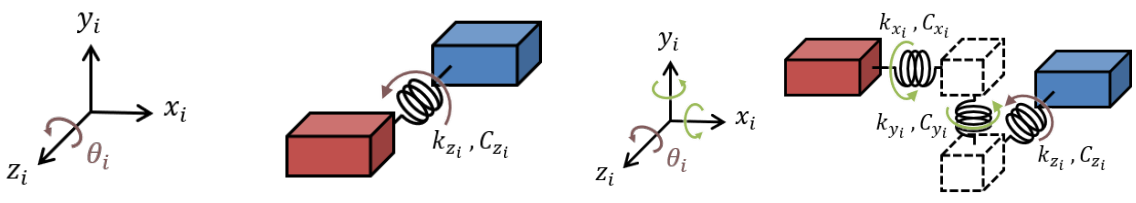


Figure 3.24: 加上非轉軸方向的 RR 手臂 Simulink 模型的連接圖

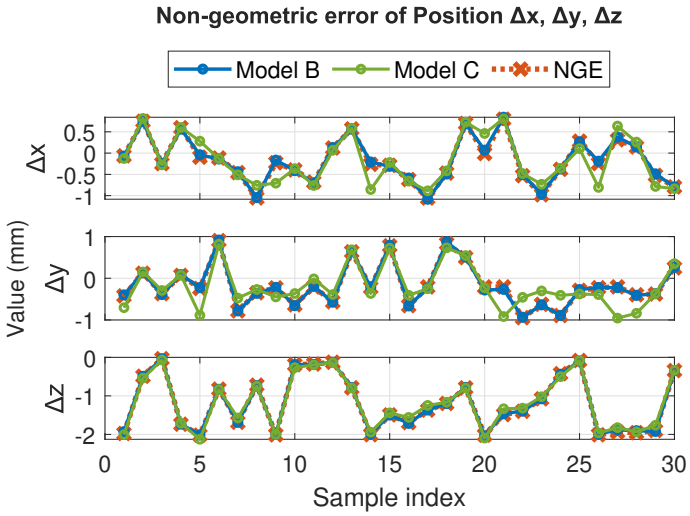


(a) 與轉軸同向　　(b) 與轉軸非同向

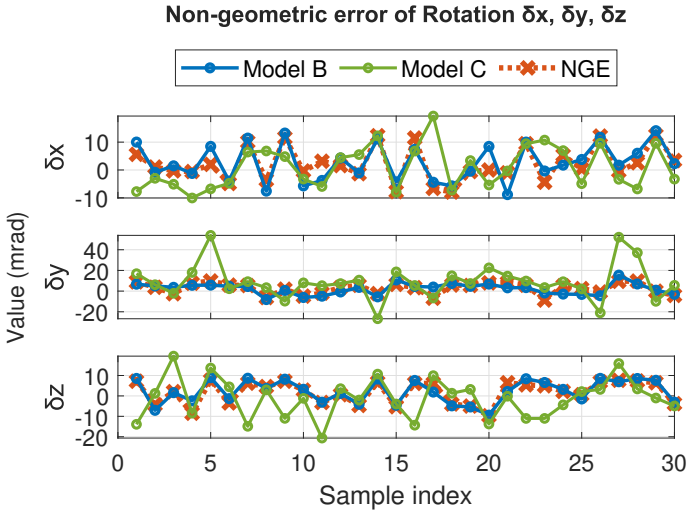
Figure 3.25: 扭矩示意圖 (a) 與轉軸同向 (b) 加上與轉軸非同向

doi:10.6342/NTU202402735

數據處理、模型建構、模型訓練的所有流程、參數設定都與3.3.2節和3.3.3相同，以下是針對 Model A 訓練完成，並對其進行剪枝後，將剪枝模型的參數轉移到 Model B 的評估結果，同樣為了與其比較這邊一樣訓練了 Model C。由 Figure3.26可以觀察到對於位置的部分兩個模型都預測的不錯，但 model B 因為有使用轉移學習，所以在旋轉的部分預測的表現比 model C 還要好上更多。



(a) 位置



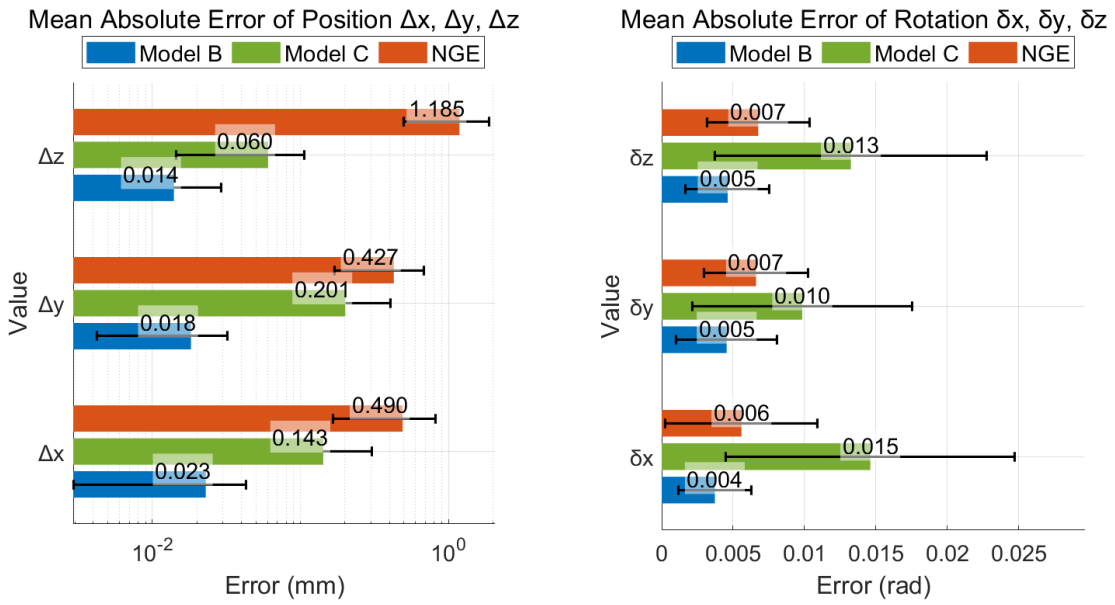
(b) 旋轉

Figure 3.26: Model B 和 Model C 隨機取樣 30 個點之預測值 (a) 位置 (b) 旋轉

將 Model B 和 Model C 對所有測試資料集的平均絕對誤差進行比較，如 Figure 3.27，結果顯示，Model B 在預測這些資料時的表現優於 Model C。Figure 3.28的箱形圖也能反映出 Model B 在預測這些資料時能夠降低整個誤差的分散程

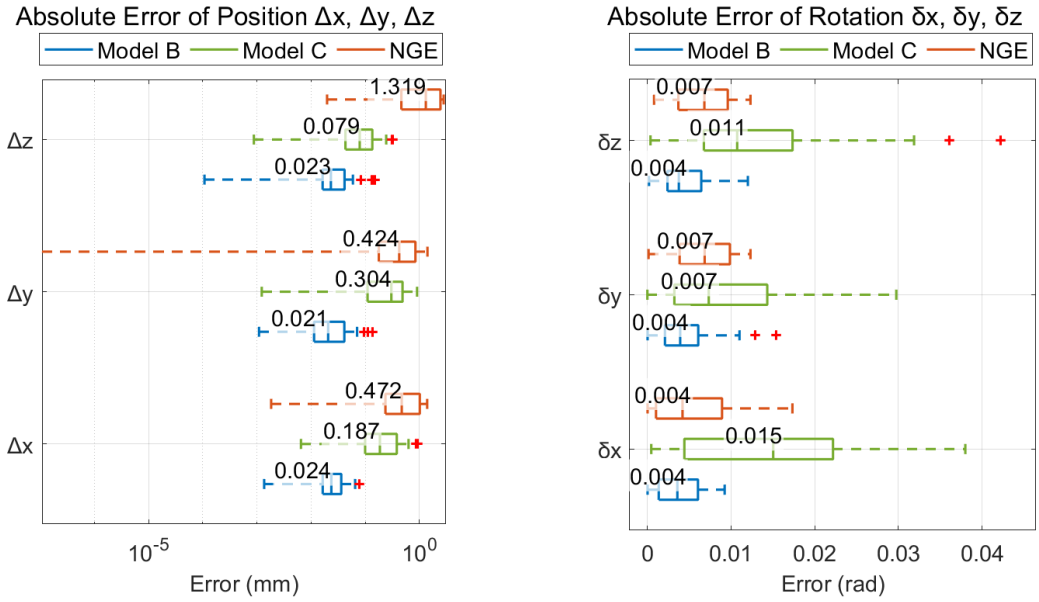
doi:10.6342/NTU202402735

度，同時也降低了最大值的位置。



(a) 位置　　(b) 旋轉

Figure 3.27: Model B 和 Model C 在測試資料的平均絕對誤差 (a) 旋轉 (b) 位置



(a) 位置　　(b) 旋轉

Figure 3.28: Model B 和 Model C 在測試資料誤差分佈的箱形圖 (a) 位置 (b) 旋轉

此實驗證明了機器學習的應用不僅限於傳統校正方法所能處理的彈性係數，還能憑藉其強大的非線性學習能力，在更廣泛的條件下準確預測誤差，包括扭轉彈簧與關節非平行的情況。

doi:10.6342/NTU202402735

第四章　串聯式工業機器人的校正模擬

為驗證本論文所提出之平行架構算法，因此本章節4將著重在把算法應用在所選用之串列式工業機器人 Meca500 上進行模擬的驗證。

4.1　機械手臂的模型

4.1節會介紹所選用之 Meca500 手臂建模方式，分別為標稱手臂運動學模型以及目標要識別的手臂物理模型，該物理手臂會在 Simulink 上進行建模，並同時建構幾何與非幾何誤差在此手臂上以進行模擬。

4.1.1　運動學模型

建立手臂的運動模型，才能夠獲得各個關節之間的轉換關係，並且藉此獲得世界座標與末端執行器的相對位置與姿態。參照章節 3.2的運動學建模方式，本章節最後同樣使用 Standard DH 建構 Meca500 手臂。

Table 4.1: Meca500 的 Nominal DH 表

i	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)	β_i (rad)
1	$\theta_1 - \frac{\pi}{2}$	135	0	$-\frac{\pi}{2}$	0
2	θ_2	0	135	0	0
3	θ_3	0	38	$-\frac{\pi}{2}$	0
4	0	120	0	0	0
5	θ_4	0	0	$\frac{\pi}{2}$	0
6	θ_5	0	0	$-\frac{\pi}{2}$	0
7	$\theta_6 + \pi$	70	0	0	0

doi:10.6342/NTU202402735

Table 4.2: Meca500 的 Actual DH 表

i	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)	β_i (rad)
1	$\theta_1 - 1.5708$	137	0	-1.5708	0
2	$\theta_2 - 0.1$	0	136.4	-0.06	0
3	$\theta_3 - 0.15$	0	40.3	-1.5668	0
4	0.06	120.35	0	-0.004	0
5	$\theta_4 - 0.15$	0.16	0	1.5708	0
6	θ_5	-0.027	0	-1.5708	0
7	$\theta_6 + 3.1416$	70	0	0	0

Table 4.3: Meca500 的誤差 DH 表

i	$\Delta\theta_i$ (rad)	Δd_i (mm)	Δa_i (mm)	$\Delta\alpha_i$ (rad)	$\Delta\beta_i$ (rad)
1	0	2	0	0	0
2	-0.1	0	1.4	-0.06	0
3	-0.15	0	2.3	0.004	0
4	0.06	0.35	0	-0.004	0
5	-0.15	0.16	0	0	0
6	0	-0.027	0	0	0
7	0	0	0	0	0

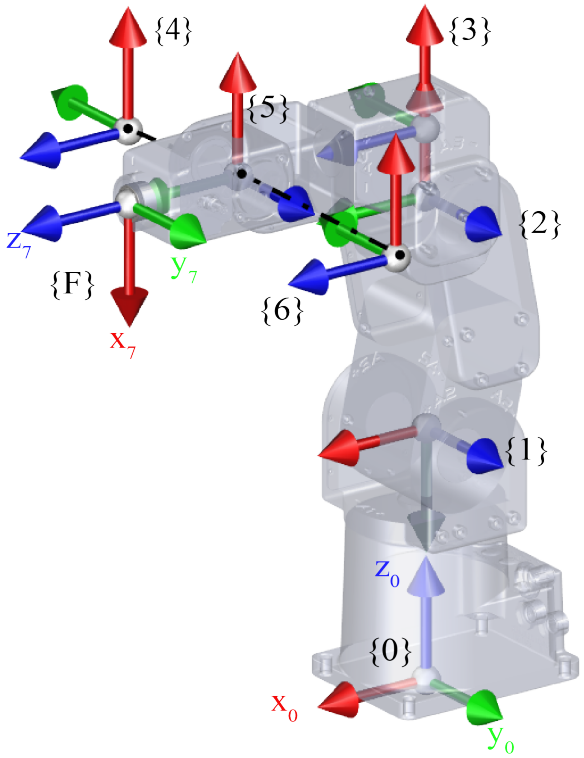


Figure 4.1: Meca500 Standard DH 座標圖

參照 Figure 4.1為描述法 Table 4.1建立出的機械手臂架構圖，將其帶入 Standard DH 的齊次轉換矩陣之形式 (2.4)，即可得知每個關相鄰關節的轉換關係。

doi:10.6342/NTU202402735

4.1.2 物理模型

為了能夠更真實地模擬機器手臂的非幾何誤差，一樣使用了 Simulink 建立物理模型。首先，構建 Meca500 的剛體樹模型 (Rigid Body Tree)Figure 4.2，引入 Workspace 已經算好的齊次轉換矩陣到連接各關節的 Rigid Transform 模塊，並在這個模型第二個和第三個旋轉關節後方，分別添加了一個串聯的額外關節。這些額外的關節不是實際機器手臂的一部分，而是用來模擬非幾何誤差。

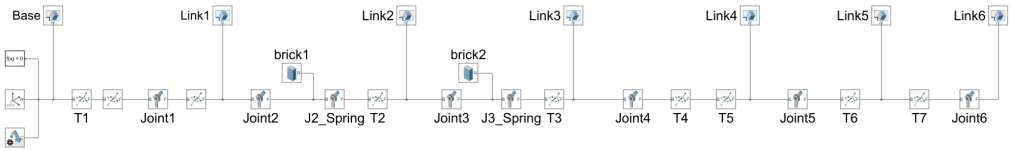


Figure 4.2: Meca500 在 simulink 的剛體樹模型

為了模擬這些誤差，對這些額外的關節施加了彈簧係數和阻尼係數。彈簧係數模擬了由於柔性結構或裝配公差引起的機器手臂關節的彈性變形，而阻尼係數則模擬了關節運動中的摩擦和能量損耗。這些參數的設置使得模型能夠更精確地反映實際情況中的機器手臂行為，特別是在受力或運動過程中出現的誤差。

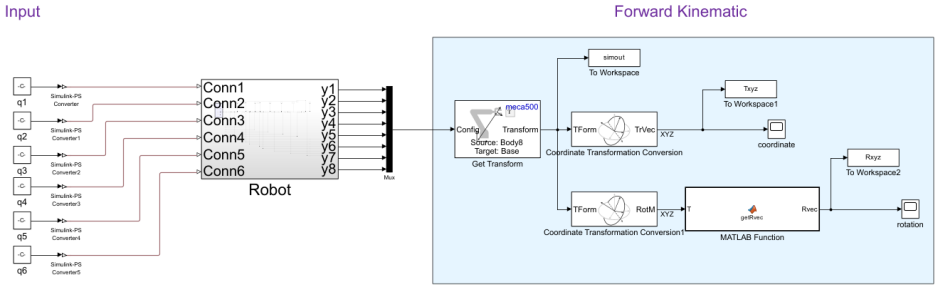


Figure 4.3: Meca500 在 simulink 的連接圖

完成剛體樹模型的構建後，串接上輸入與輸出如 Figure 4.3所示，使用 Get Transform 模塊來獲得末端執行器的姿態，此模塊將計算並回傳 Base 對末端的齊

doi:10.6342/NTU202402735

次轉換矩陣，由此矩陣可以分析出手臂的末端位置以及旋轉姿態，Figure 4.4為手臂角度 $\theta_1 \sim \theta_6$ 依序為 $[10,\ 20,\ -20,\ 10,\ 0,\ 0]$ 時，剛體樹模型在 Mechanics Explorer 可視化狀態。

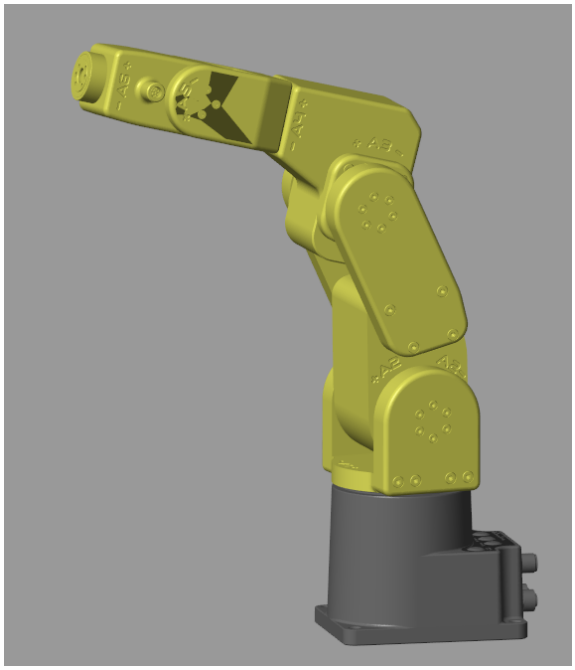
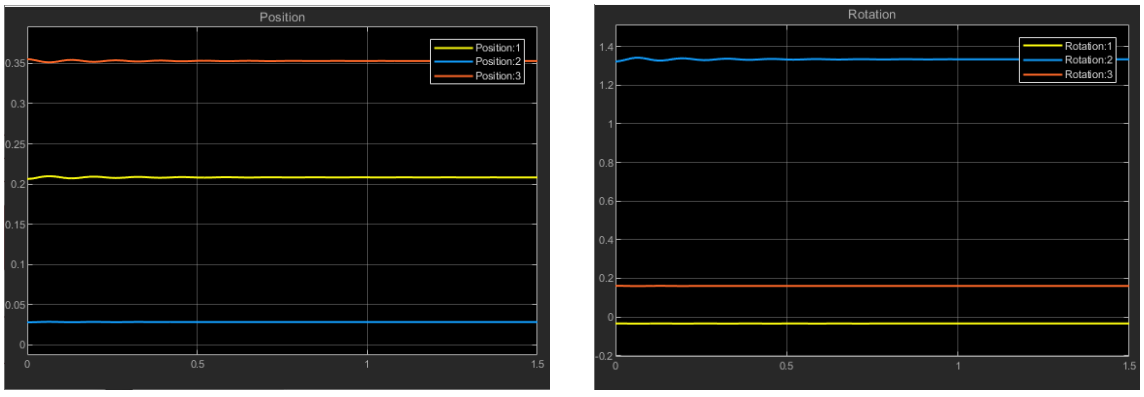


Figure 4.4: Meca500 在 simulink 的物理模型

Figure 4.5 則為此狀態的輸出圖，可以觀察到確實有震盪的情況在末端座標上產生，證實串接的關節確實有模擬到非幾何的誤差。



(a) 位置 (unit:mm)[26]　　(b) 旋轉向量

Figure 4.5: Meca500 在 simulink scope 輸出圖 (a) 位置 (b) 旋轉向量

doi:10.6342/NTU202402735

在生成物理模型末端姿態的部分為了講究實驗的真實性，參考了 Meca500 的規格書 [26] ，利用其馬達可轉動角度範圍表4.4來定義 Meca500 做動範圍是否合乎規格，使用 q_range $=$ `linspace`$(x1, x2, n)$ 函數將這些範圍切割至想要的份數，其中：$x1$ 是範圍的起始值、$x2$ 是範圍的終止值、n 是要分割的份數。

Table 4.4: Meca500 關節活動範圍

Joint	Range of motion (degrees)
Joint 1	$[-175°, 175°]$
Joint 2	$[-70°, 90°]$
Joint 3	$[-135°, 70°]$
Joint 4	$[-170°, 170°]$
Joint 5	$[-115°, 115°]$
Joint 6	$[-36,000°, 36,000°]$

後續將使用到轉移學習，因此設計兩個 Meca500 手臂，詳細設定將於下一小節介紹，將模擬所用的相應的角度範圍寫成表4.5如下：

Table 4.5: 角度範圍與分割數量

變數	定義範圍 (度數)	Meca500 A 分割數量	Meca500 B 分割數量
$q1\_range$	$[-175°, 175°]$	10	5
$q2\_range$	$[-60°, 60°]$	10	5
$q3\_range$	$[-120°, 60°]$	10	5
$q4\_range$	$[-160°, 160°]$	10	6
$q5\_range$	$[-90°, 90°]$	10	5
$q6\_range$	$[0°]$	1	1

同時透過 `ndgrid` 生成相對應的輸出角度網格：

$$[q_1\_range, q_2\_range, q_3\_range, q_4\_range, q_5\_range, q_6\_range] = \text{ndgrid}(q_1\_range, q_2\_range, q_3\_range, q_4\_range, q_5\_range, q_6\_range) \tag{4.1}$$

此過程確保獲得覆蓋指定範圍的完整角度網格，並將每個角度範圍分割成所需的份數。藉此降低需要大量巢狀迴圈運算的時間複雜度，依照每組輸出角度輸入至模型，檢查末端點是否合乎 Meca500 的運動範圍 Figure2.2a。

doi:10.6342/NTU202402735

4.2 基於平行架構的校正

4.2節將著重於如何透過傳統校正法將幾何誤差補償到標稱手臂運動學模型，再將剩餘的非幾何誤差透過 FNN 模型訓練出來，以找到一個平行架構模型能夠最小化 $\Delta\vec{P}_{min}$ 的任務。

4.2.1 DH 參數校正

為了方便後續描述轉移學習，將更詳細定義接下來會用到的代號，Figure 3.3的 FNN 實際訓練流程如 Figure 4.6所示。

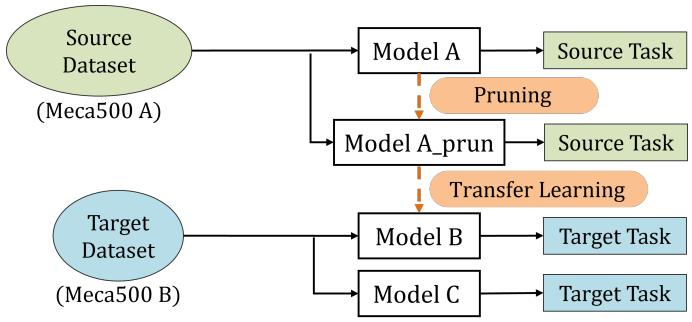


Figure 4.6: 轉移學習示意圖

首先分別定義出兩組相同架構，但彈簧係數和阻尼係數不同的 Meca500 A 和 Meca500 B，詳細參數定義內容可參見表 4.6，並且為了使用轉移學習所以在 Meca500 B 生成較少的資料集。目標任務是 Meca500 B 生成之末端姿態與經過 Jacobian 校正後的運動學輸出誤差，源任務則是 Meca500 A 生成的末端姿態同樣經過 Jacobian 校正後的運動學輸出誤差。

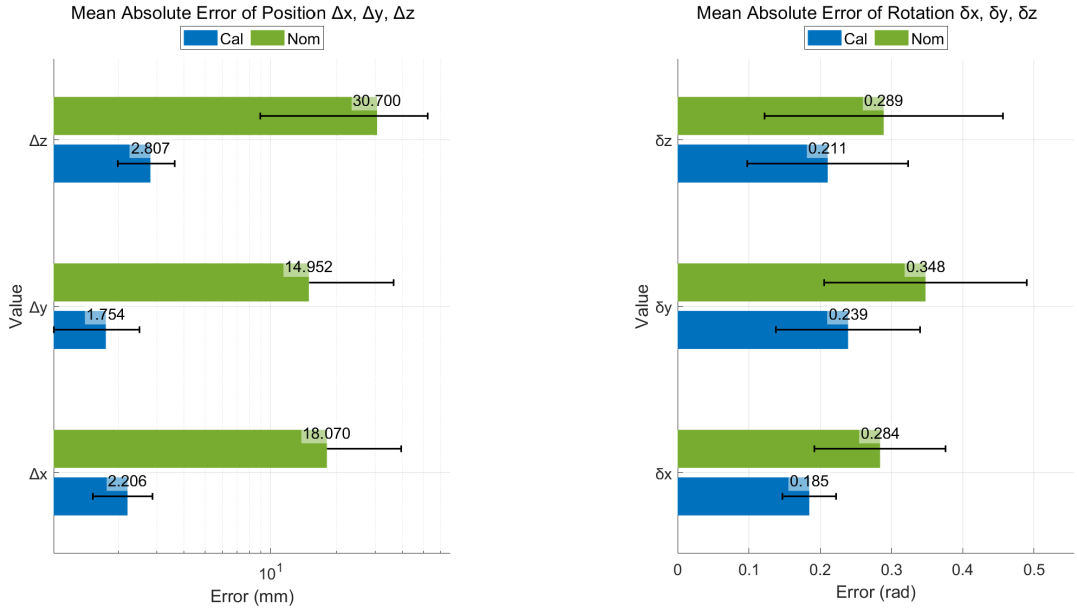
Table 4.6: 手臂物理模型的參數設定

手臂	彈簧係數 (N·m/rad)	阻尼係數 (N·m/(deg/s))	資料集數量
Meca500 A	2	4e-3	36572
Meca500 B	1	3e-3	1528

以下是將 Jacobian 校正法應用在 Meca500 B 的結果，隨機取樣 100 個點進行校正。計算標稱和校正後的手臂末端姿態的平均絕對誤差，如 Figure4.7在位置部

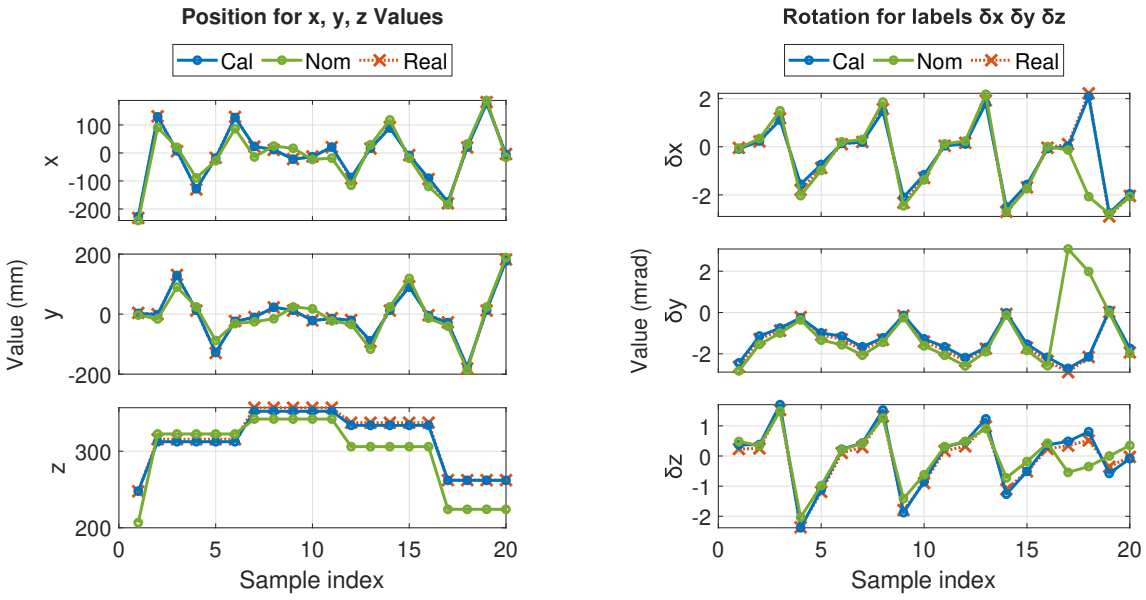
doi:10.6342/NTU202402735

分的誤差降低了 90%，旋轉部分也降低了約 30%。由 Figure4.8可知，原本較不準確的標稱順向運動學模型經過校正後，末端姿態更接近實際值。



(a) 位置　　(b) 旋轉

Figure 4.7: 測試資料集預測值和真實值的平均絕對誤差 (a) 位置 (b) 旋轉



(a) 位置　　(b) 旋轉

Figure 4.8: Meca500 B 取樣 20 個點之預測值和真實值 (a) 位置 (b) 旋轉

doi:10.6342/NTU202402735

4.2.2 類神經網路校正

類神經網路主要用來補償那些沒辦法被幾何校正的非幾何誤差，因此機器學習的目標是那些經過 Jacobian 校正後的運動學輸出與實際末端末端姿態之差值。

預訓練模型使用的是 Meca500 A 所生成的資料集，模型架構如 Figure 4.9。此模型使用 36572 筆資料來進行資料的切割，其中表4.7為訓練、驗證、測試資料集的數量，比例大概是 7 : 1.5 : 1.5。

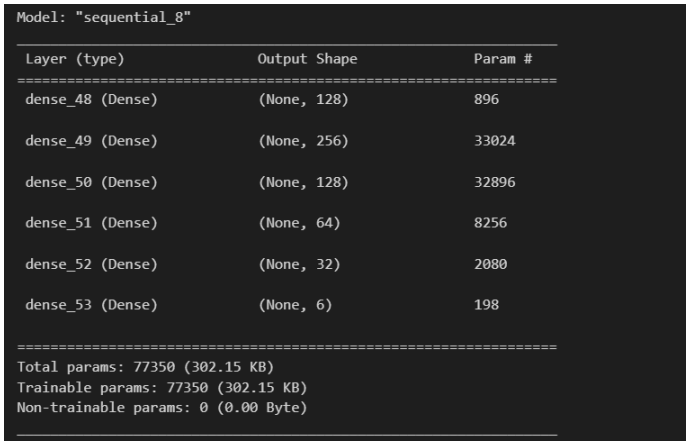


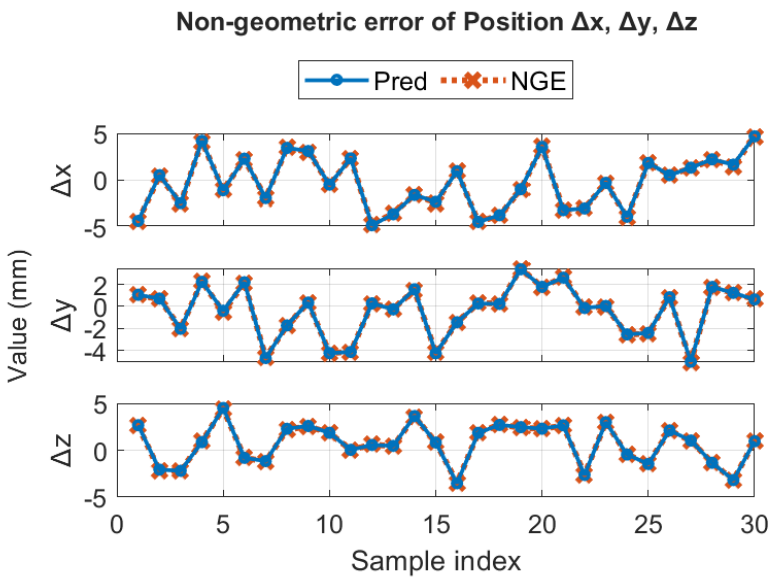
Figure 4.9: Model A 模型架構

Table 4.7: Meca 手臂訓練資料集切割後資料數

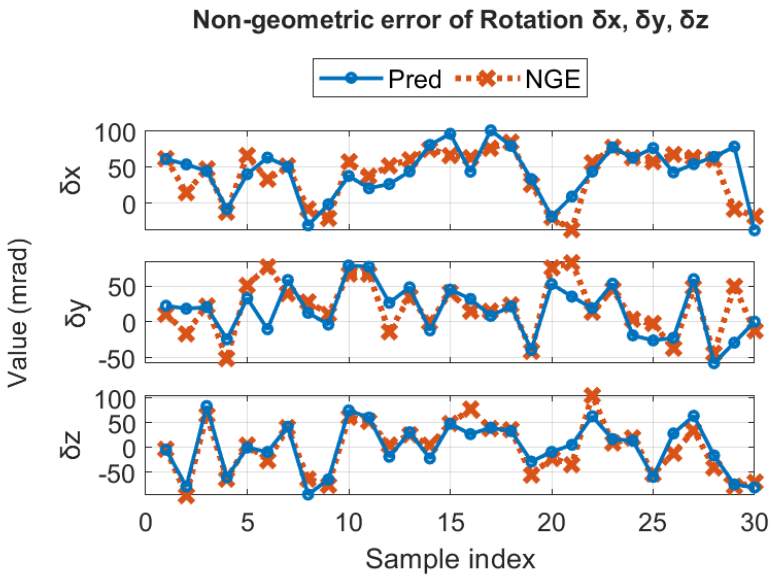
Dataset (type)	Input (num of samples, dimensions)	Output (num of samples, dimensions)
Training	(25600, 6)	(25600, 6)
Validation	(5486, 6)	(5486, 6)
Testing	(5486, 6)	(5486, 6)

以下是模型訓練的結果，由 Figure4.10可以觀察出，透過使用大量的資料集進行模型訓練，可以顯著提升神經網路的表現。不論是在位置預測上，還是在旋轉預測上，都能達到良好的效果。

doi:10.6342/NTU202402735



(a) 位置

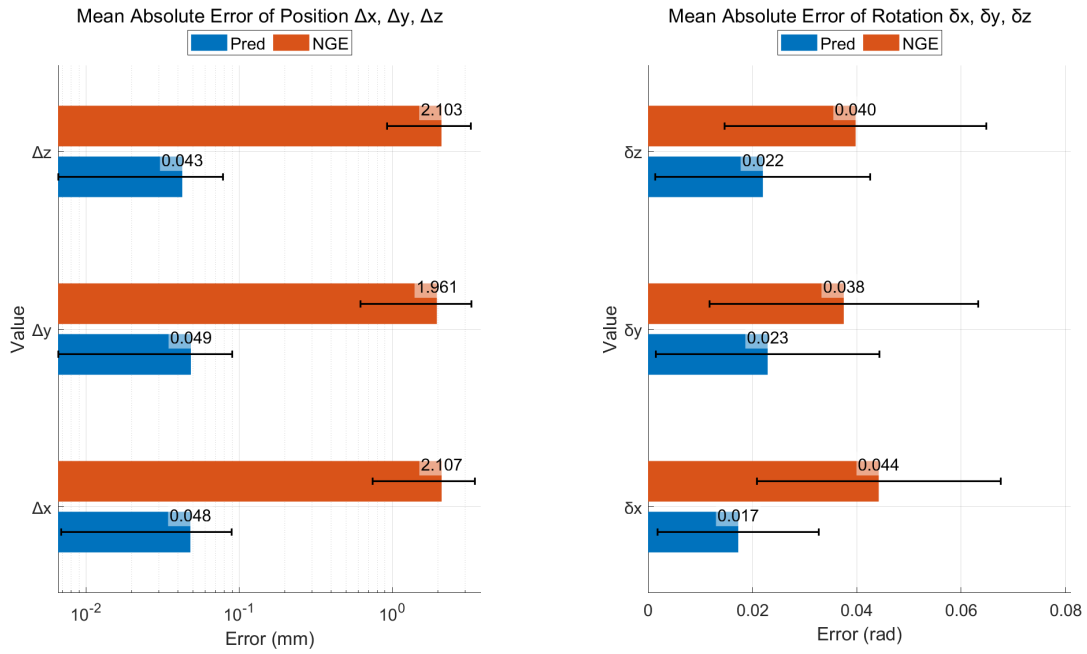


(b) 旋轉

Figure 4.10: Model A 取樣 30 個點之預測值和真實值 (a) 位置 (b) 旋轉

同樣將實際的末端姿態作為基準，把非幾何誤差當作標準，算出平均絕對誤差如 Figure4.11，可以發現在位置的部分誤差降低了 95%，而在旋轉的部分也降低了 40% 左右。

doi:10.6342/NTU202402735



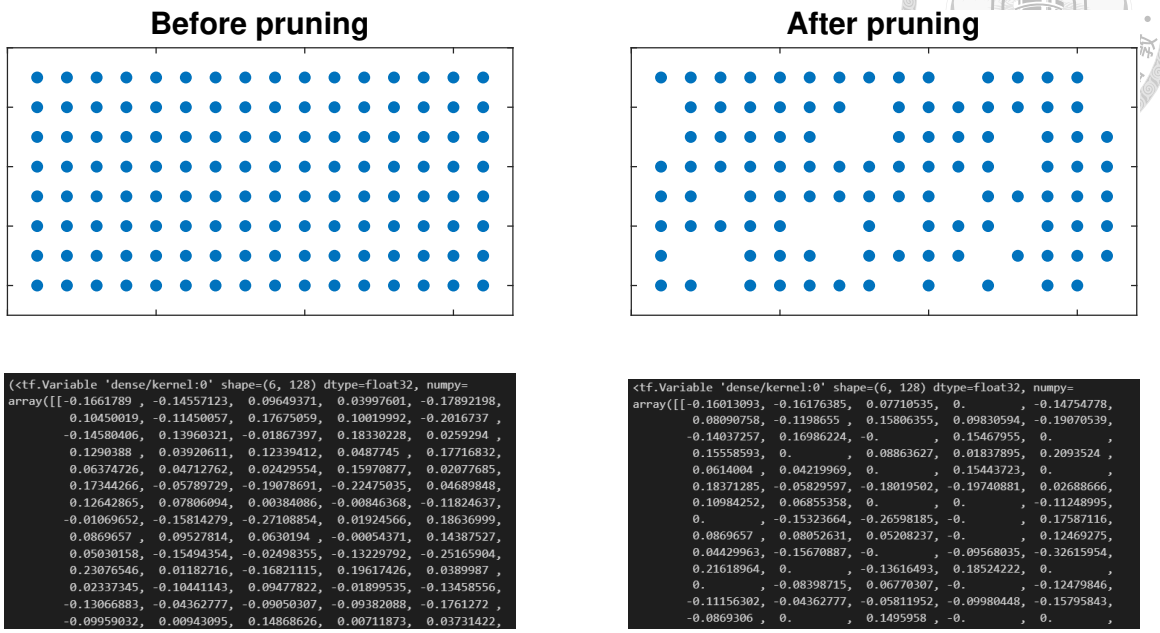
(a) 位置　　(b) 旋轉

Figure 4.11: Model A 測試資料集預測值和真實值的平均絕對誤差 (a) 位置 (b) 旋轉

同時，為了增加後續轉移學習上的表現，並且降低 Model A 的計算複雜度對其進行剪枝。如此可以有效減少模型參數數量，提升模型的運算效率，並在保持模型精度的同時，減少過擬合的風險。此外，通過剪枝後的模型可以更容易適應不同的任務和數據集，從而在轉移學習中表現得更為出色。Figure 4.12為剪枝過後模型的參數，將模型的最終剪枝率設為 20%，因此剪枝過後有部分的參數被修剪為 0。

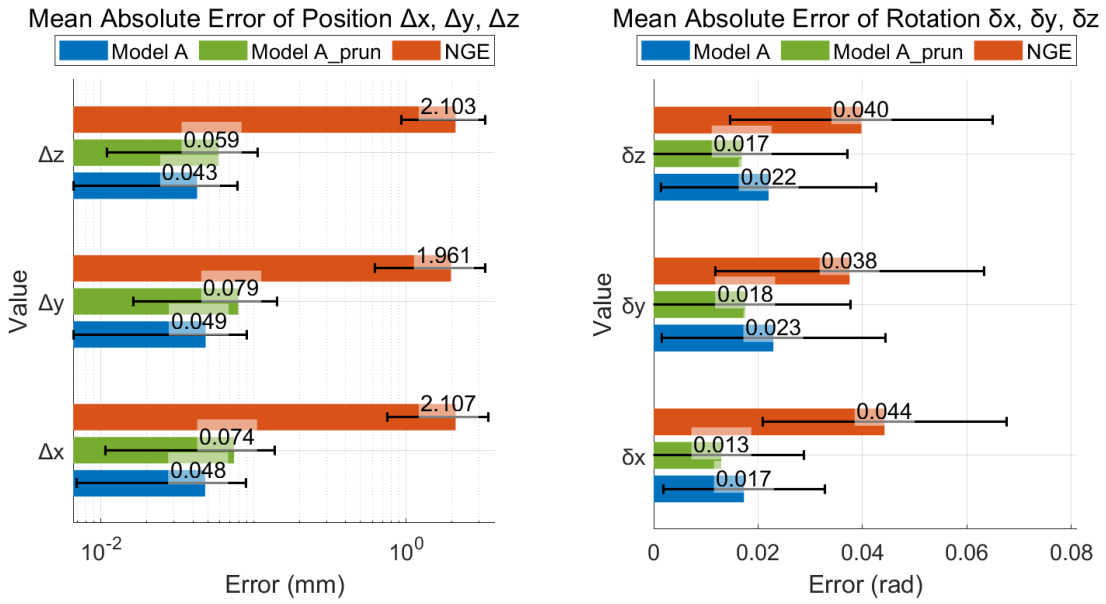
Figure 4.13為原本的非幾何誤差、ModelA 和的 Model A 剪枝後的平均絕對誤差，可以發現兩個模型性能上沒有太大的差異，甚至是經過剪枝後的模型在旋轉的部分反而預測得更好一些。

doi:10.6342/NTU202402735



(a) 剪枝前　　(b) 剪枝後

Figure 4.12: Model A 剪枝前後模型第一層第一列的權重分布與其值
(a) 剪枝前 (b) 剪枝後



(a) 位置　　(b) 旋轉

Figure 4.13: Model A 剪枝前後測試資料集的平均絕對誤差 (a) 位置 (b) 旋轉

doi:10.6342/NTU202402735

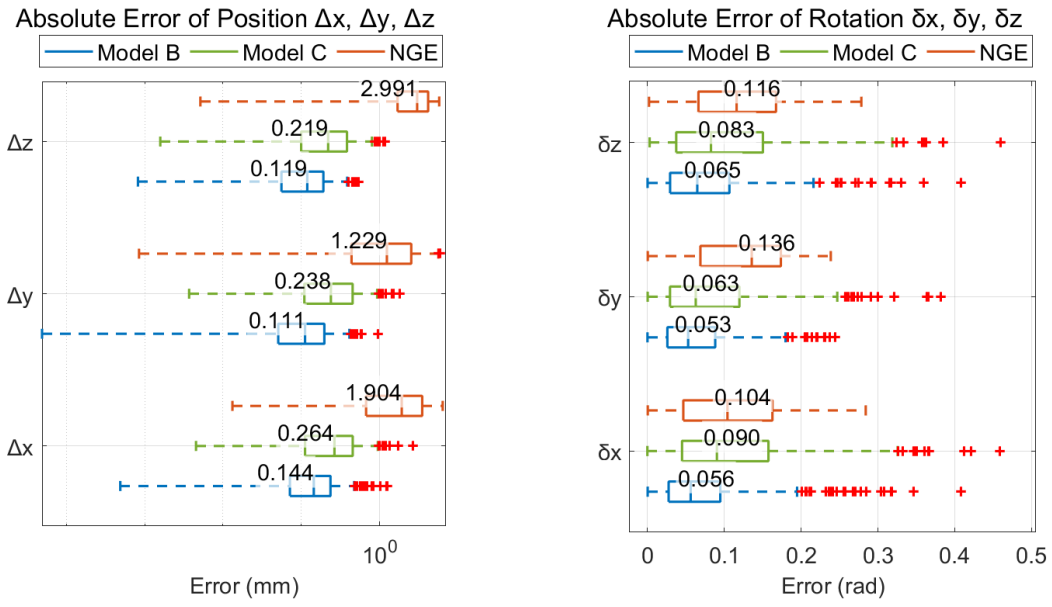
4.2.3 使用轉移學習進行的數位雙生模型校正

如同 Figure 4.6 所示，最終的目標任務是要識別 Meac500 B 的幾何誤差，所以延續4.2.1節的校正結果，模擬真實資料集較難以收集，此章節必須用少量的1528 筆資料集完成模型學習，表4.8為切割後的資料數量比例為 5 : 2.5 : 2.5。

Table 4.8: Meca 手臂 B 訓練資料集切割後資料數

Dataset (type)	Input (num of samples, dimensions)	Output (num of samples, dimensions)
Training	(764, 6)	(764, 6)
Validation	(382, 6)	(382, 6)
Testing	(382, 6)	(382, 6)

Model C 直接用 Figure 4.9架構訓練，Model B 則是基於經過剪枝的 Model A 的參數來做 Fine-Tune。將經過校正運動學的末端姿態作為基準，把非幾何誤差當作標準。首先觀察箱形圖 Figure 4.14比較兩者資料分布狀態，反映出兩個模型在預測誤差時，發現 Model B 在預測這些資料時，相較 Model C 更能夠降低整個誤差的分散程度。

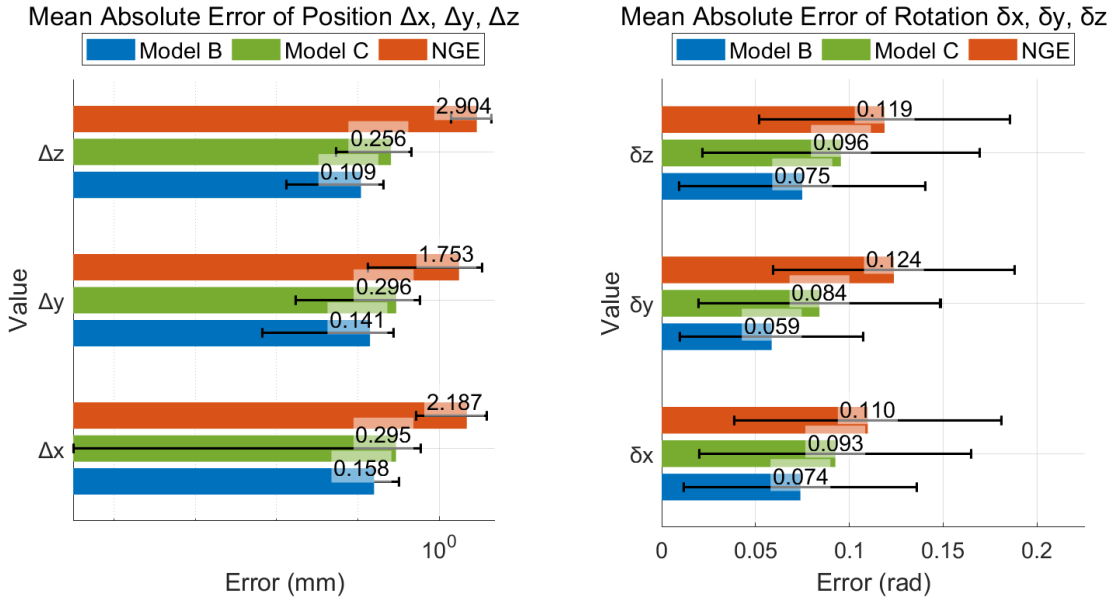


(a) 位置　　(b) 旋轉

Figure 4.14: Model B 和 Model C 在測試資料誤差分佈的箱形圖 (a) 位置 (b) 旋轉

doi:10.6342/NTU202402735

同時，算出 Model B 和 Model C 對所有測試資料集的平均絕對誤差如 Figure 4.15進行比較，Model B 的表現，發現在位置的部分誤差降低了 95~97%，而在旋轉的部分也降低了 30~50%。相較之下 Model C 的表現略微遜色，在位置的部分誤差降低了 80~90%，旋轉的部分僅減少了 8~28%，可見在 Model B 在預測這些資料時的表現優於 Model C。

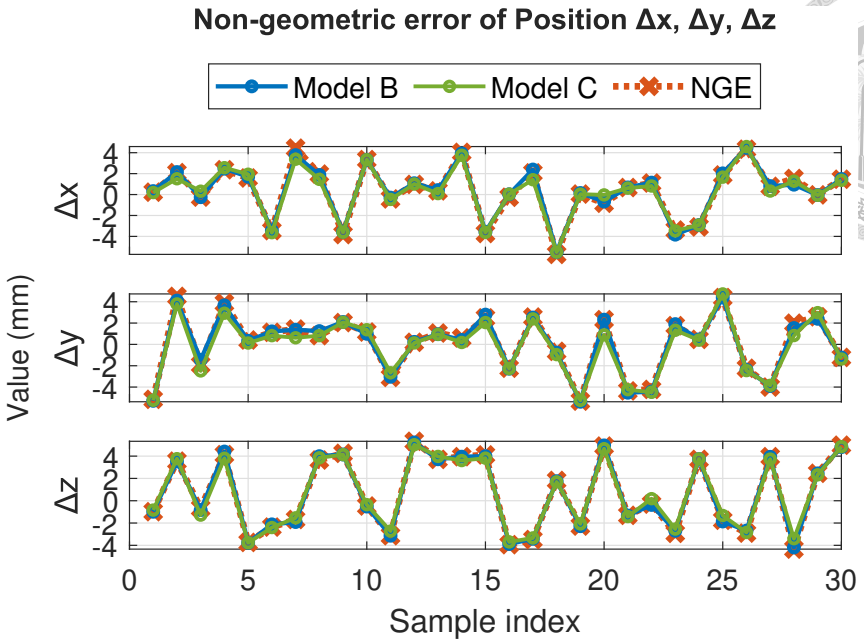


(a) 位置　　(b) 旋轉

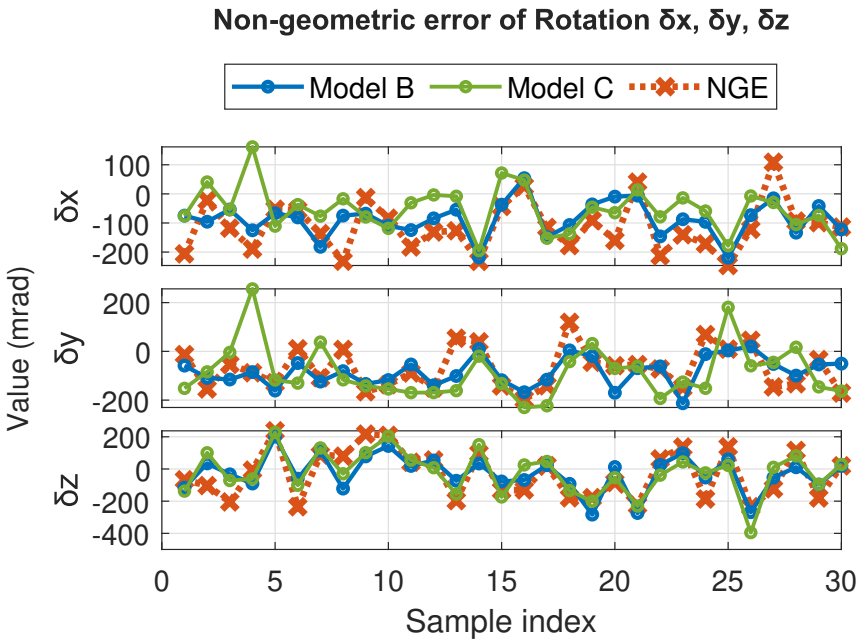
Figure 4.15: Model B 和 Model C 在測試資料的平均絕對誤差 (a) 位置 (b) 旋轉

也可以將真實幾何誤差與模型的預測放在折線圖進行比較，Figure 4.16 可以發現對於位置的部分兩個模型都預測的不錯，但 model B 因為有使用轉移學習，所以在位置的部分預測幾乎能夠完全預測到。旋轉的部分兩個模型雖然都有符合趨勢，雖然看折線圖沒辦法詳細的確定哪個模型表現較好，但從 Figure4.15 和 Figure4.14 可以得知，旋轉部分的 MAE 結果和散布情形還是 model B 的性能較優。

doi:10.6342/NTU202402735



(a) 位置



(b) 旋轉

Figure 4.16: Model B 和 Model C 隨機取樣 30 個點之預測值 (a) 位置 (b) 旋轉

doi:10.6342/NTU202402735

第五章　結論

5.1　整體校正效能

本論文提出一種平行架構的手臂校正方法，基於傳統的 Jacobian 校正方式先補償幾何誤差，而剩餘的非幾何透過機器學習來補償，同時利用轉移學習的方式來強化當實際資料集較少的情況下，能夠保持模型之強健性，進而識別出一個最接近真實手臂之模型。後續可以利用反向運動學計算每個關節的輸入值，實現預期的末端位置和旋轉姿態。此平行架構算法在第一階段的幾何校正，於位置部分降低了 90% 左右的誤差量，第二階段的非幾何校正，基於第一階段已經校正好的幾何誤差，利用機器學習與轉移學習，再降低了 95% 的誤差量，最後誤差能夠從 30~20 mm 減少到 0.1 mm 左右；旋轉部分段的幾何較正則是降低 30%，非幾何校正再降低 40%，最後誤差能夠從 0.3~0.2 rad 減少到 0.01 rad 左右，由此可知不管是幾何還是非幾何，誤差都有相當顯著的下降。

5.2　未來展望

目前實驗主要在模擬環境中進行，未來可以進一步拓展以下幾個方向，以期望提升校正效能並推廣其應用：

1. **真實環境測試**：將所提出的校正方法應用於實際的機械手臂系統，尤其是非幾何變形較嚴重的系統，如平行架構式的連桿機器手臂和軟性機器手臂，

doi:10.6342/NTU202402735

驗證其在真實環境下的性能和穩健性。如 Figure 5.1，本研究預計在真實的 Meca500 手臂上進行的實驗，選擇動作捕捉器與感光球來做為感測器量測末端實際姿態，同時設計了超過 Meca500 負載的質量塊讓實際手臂產生非幾何的偏移。

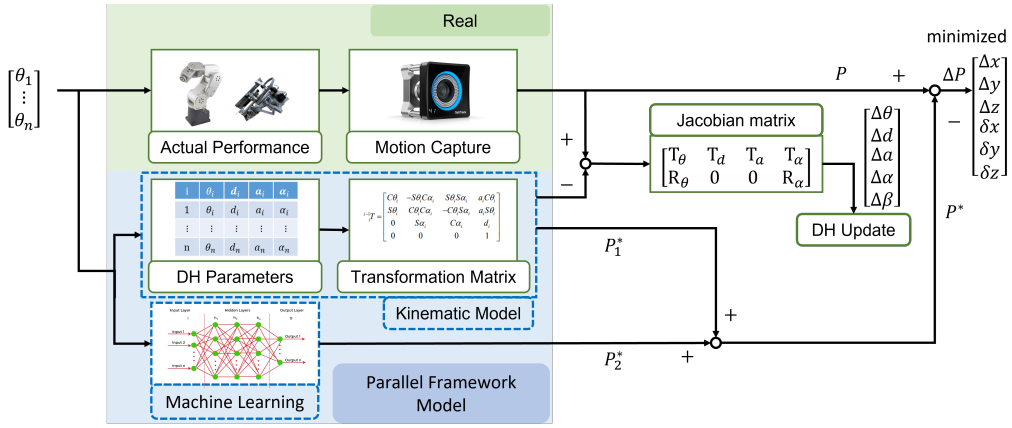


Figure 5.1: 未來實驗規劃

2. **算法優化**：進一步優化平行架構算法，特別是機器學習部分。可以嘗試不同的學習結構，如集成學習 (Ensemble learning) 將多種模型合併。此外，可以加入其他非線性因素，如背隙 (backlash) 等，來提升校正精度和訓練效率。尤其是針對旋轉姿態校正較難的問題，探索將非幾何變形函數納入網路中，以改進擬合效果的物理學引導神經網路 (Physics-guided Neural Networks，PGNN)，甚至是近一步考慮使用長短期記憶模型 (Long Short-Term Memory，LSTM) 加入動態的預測。

3. **資料集擴展與多樣化**：增加更多實際數據，並包含在各種操作條件下的數據如增加手臂負載等，以進一步提高模型的泛化能力。可以考慮收集來自不同型號和應用場景的機械手臂數據，以全面覆蓋各種可能的情況。

通過這些方向的深入研究和實踐，不僅可以進一步提升所提出方法的性能和應用價值，還可以為機械手臂的校正技術發展提供新的思路。

doi:10.6342/NTU202402735

參考文獻

[1] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, and Rajiv Suman. Substantial capabilities of robotics in enhancing industry 4.0 implementation. Cognitive Robotics, 1:58–75, 2021.

[2] Ruchi Goel and Pooja Gupta. Robotics and industry 4.0. A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development, pages 157–169, 2020.

[3] Martin Hägele, Klas Nilsson, J Norberto Pires, and Rainer Bischoff. Industrial robotics. Springer handbook of robotics, pages 1385–1422, 2016.

[4] Orhan Özgüner, Thomas Shkurti, Siqi Huang, Ran Hao, Russell C Jackson, Wyatt S Newman, and M Cenk Çavuşoğlu. Camera-robot calibration for the da vinci robotic surgery system. IEEE Transactions on Automation Science and Engineering, 17(4):2154–2161, 2020.

[5] Hang Su, Chenguang Yang, Hussein Mdeihly, Alessandro Rizzo, Giancarlo Ferrigno, and Elena De Momi. Neural network enhanced robot tool identification and calibration for bilateral teleoperation. IEEE Access, 7:122041–122051, 2019.

[6] ZVIS Roth, Benjamin Mooring, and Bahram Ravani. An overview of robot calibration. IEEE Journal on Robotics and Automation, 3(5):377–385, 1987.

doi:10.6342/NTU202402735

[7] Long Qian, Jie Ying Wu, Simon P DiMaio, Nassir Navab, and Peter Kazanzides. A review of augmented reality in robotic-assisted surgery. IEEE Transactions on Medical Robotics and Bionics, 2(1):1–16, 2019.

[8] Joon Hyun Jang, Soo Hyun Kim, and Yoon Keun Kwak. Calibration of geometric and non-geometric errors of an industrial robot. Robotica, 19(3):311–321, 2001.

[9] Samad Hayati and M Mirmirani. Improving the absolute positioning accuracy of robot manipulators. Journal of robotic systems, 2(4):397–413, 1985.

[10] W Veitschegger and Chi-haur Wu. A method for calibrating and compensating robot kinematic errors. In Proceedings. 1987 IEEE International Conference on Robotics and Automation, volume 4, pages 39–44. IEEE, 1987.

[11] Donald Lee Pieper. The kinematics of manipulators under computer control. Stanford University, 1969.

[12] Ahmed Joubair and Ilian A Bonev. Non-kinematic calibration of a six-axis serial robot using planar constraints. Precision Engineering, 40:325–333, 2015.

[13] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

[14] Batta Mahesh. Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9(1):381–386, 2020.

[15] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5:115–133, 1943.

[16] Christian Bauckhage and Daniel Speicher. Lecture notes on machine learning neurons with non-monotonic activation functions. cal, 5:4, 1943.

doi:10.6342/NTU202402735

[17] Paul Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. PhD thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA, 1974.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.

[19] Jooyoung Park and Irwin W Sandberg. Universal approximation using radial-basis-function networks. Neural computation, 3(2):246–257, 1991.

[20] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359, 2009.

[21] Phuong DH Nguyen, Tobias Fischer, Hyung Jin Chang, Ugo Pattacini, Giorgio Metta, and Yiannis Demiris. Transferring visuomotor learning from simulation to the real world for robotics manipulation tasks. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6667–6674. IEEE, 2018.

[22] Jack Collins, David Howard, and Jurgen Leitner. Quantifying the reality gap in robotic manipulation tasks. In 2019 International Conference on Robotics and Automation (ICRA), pages 6706–6712. IEEE, 2019.

[23] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pages 23–30. IEEE, 2017.

doi:10.6342/NTU202402735

[24] Jared Alan Frank, Sai Prasanth Krishnamoorthy, and Vikram Kapila. Toward mobile mixed-reality interaction with multi-robot systems. IEEE Robotics and Automation Letters, 2(4):1901–1908, 2017.

[25] Jacques Denavit and Richard S Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. 1955.

[26] Mecademic. User Manual Original instructions, 2021. Meca500 (R3) Robot Firmware: 8.3 Document Revision: A February 17, 2021.

[27] Russell Reed. Pruning algorithms-a survey. IEEE transactions on Neural Networks, 4(5):740–747, 1993.

doi:10.6342/NTU202402735