

# 運用 Simulink 機電整合工具開發多指機械人手掌

## Development of a Multi-Finger Hand for Humanoid Robot using Simulink Mechatronics Tool Chain

陳傳生<sup>1</sup>, 張家榮<sup>2</sup>

元智大學機械工程研究所

mecch@saturn.yzu.edu.tw

### 摘要

機械手掌是人型機械人與環境及使用者互動的必要工具。多數達到人手靈活程度的機械手掌，必須容忍過大的體積與重量；還需要極複雜的電腦控制器才能操縱高度靈活的動作。複雜的機構與電腦軟硬體對於機構與嵌入式軟硬體設計，構成高難度的挑戰。

為達到高靈活度與高結構剛性，本文的機械手掌具有4根手指，採用連桿式驅動設計。共有14個可動關節，由10組線性伺服直流馬達控制，所有控制器均鑲嵌於機械手掌結構。全部重量僅有500公克，體積與人手相同。

手掌採用10個控制模組構成的分散式網路控制系統。10個控制模組都以16位元單晶片Microchip dsPIC33FJ128MC202 微控制器為核心。每一模組具有full-bridge PWM 馬達驅動，位置伺服控制，路徑規劃，力量控制與I2C及UART通訊功能。控制系統本身即可獨立執行基本動作，如握手，拾取物品。使用者僅需透過串列(RS232或USB)界面，下達高階指令，由控制系統中兼任Master的模組將高階指令透過預存的劇本，換算為個別模組的速度及位置指令，再透過I2C網路傳送到各控制器中

在開發機械手掌的過程中，我們使用Solidworks設計手掌機構，並將3D電腦模型導入Simmechanics進行運動模擬。而手掌實體是以快速成型技術直接從CAD模型製作。至於嵌入控制器設計則透過Matlab/Simulink工具箱進行系統參數鑑別與控制器設計。所有控制器程式都是使用Simulink與Stateflow的方塊建立，再透過Real time Workshop embedded Coder及Stateflow coder與dsPIC的介面方塊，產生C程式。開發控制器時進行的參數鑑別，控制器設計模擬也是使用相關的toolboxes完成。

由於手掌運動必須協調各手指的運動。為避免直接編寫手指動作的不便，以及考慮運動控制系統對手指運動時間的影響，作者以Matlab的人機介面開發

程式(GUIDE)開發機械手的3D模擬與動作編輯程式。它結合機械手控制系統與機械運動的Simulink模型，VR Toolbox建立的3D虛擬機械手掌，以及與USB介面上的實體機械手指的通訊功能而成。在GUI模擬的任何細微動作與順序，都能立刻顯示在3D手掌；並可將動作內容儲存到手掌的記憶體作為後續的動作劇本。由於此3D虛擬手掌的Simulink動態模型，包含馬達動態響應，輸出飽和與路徑規畫的模型，所以模擬的反應時與實際手掌的動作時間相差極微。

機械手掌與機械人是結合機械，電子與電腦程式的複雜機電整合系統。開發此複雜系統必須使用現代的電腦輔助工具，才能掌握跨越不同領域的眾多設計參數，我們運用機械設計與控制設計的Rapid Control Prototyping工具，在極短的時內，就完成整個機械手掌的開發工作。

**關鍵字：**機械手掌，嵌入式系統，人機介面。

### 1. 介紹

開發機械手掌的目的不僅是賦予人形機器人與人類相同外觀的手掌，更提升機械人手部動作的靈巧度，能夠進行複雜而纖細的服務工作，如握手，取用物品，使用工具等。機械手也做為殘障人士的電子義肢，提供遠比傳統義肢優越的性能與外觀。欲達到上述目標，機械手掌必須具備體積小、重量輕，外型逼真與高靈活度的特點。

隨著機械人發展的趨勢，機械手掌的研發近來有大幅進展。德國航太總署(DLR)研發的DLR Hand II具有5根手指，18個獨立馬達控制的齒輪驅動關節；指尖配備微力量/力矩感測器，而尺寸約為人手的1.5倍。但是不計入主控制器時，重量已達3.5Kg[1, 2]。日本早稻田大學的Twenty-one具有4隻手指，13個自由度，但尺寸更大，重量更重。兩者雖具有極高的

---

<sup>1</sup> Correspondent author;

<sup>2</sup> graduate student;

靈活度，但龐大的尺寸與重量勢必提高手掌與機械人結合的困難，遑論符合電子義肢所需的節能與體積要求。其他諸如美國 Barret Technology 的 Whole Arm Manipulator 所用機械手與 Shadow Robot Company 的 Shadow hand。前者採用纜繩驅動的3爪設計；後者採用 air muscle 的驅動設計。兩者雖然手掌本身的體積接近人手，但是驅動器與控制器的體積卻遠高於手掌。

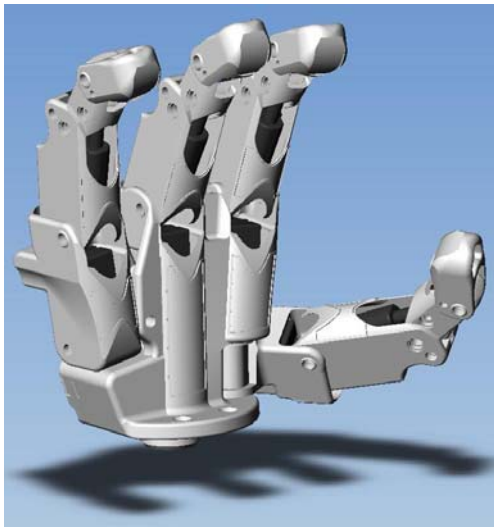


圖1. 機械手掌外觀

本研究的目的是針對中小型機械人與電子義肢，開發高靈活性，小體積，低重量的機械手，透過創新的機構設計與高性能的嵌入式控制系統，整合機構設計與嵌入式系統已達到此目標。

由於必須徹底整合機構設計與控制系統才能達到體積小的目標，我們嘗試透過最新的快速開發技術進行整個手掌的機電系統設計。後續文章中，我們首先介紹本機械手掌的機構設計特點。其次介紹嵌入式控制系統的硬體架構。然後說明如何使用 Simulink 的機電整合工具，進行從物理建模，系統參數鑑別，到控制器設計模擬，產生 C 語言控制程式，參數調整，以及以 Matlab 的 GUIDE 開發工具建立 3D 虛擬控制界面的工作過程。

## 2. 機械手掌機構設計

人類手掌具有五隻手指，每隻手指末端關節的靈活度較其餘的關節低。每隻手指的運動主要是由根部與中央關節控制，尤其是最靠近掌心的根部關節，它具有左右與俯仰兩個運動自由度。在五隻手指中，小指的靈活度較其餘手指低。考慮上述觀察與機械設計的難度，本機械手掌採用省略小指的4隻手指設計(圖1)。除拇指根部關節增加一個旋轉的自由度外，其餘手指的根部關節僅有一個自由度。每根手指都具有3個關節。但是末端關節與中央關節是以連桿相連，並

非獨立控制(圖2)。

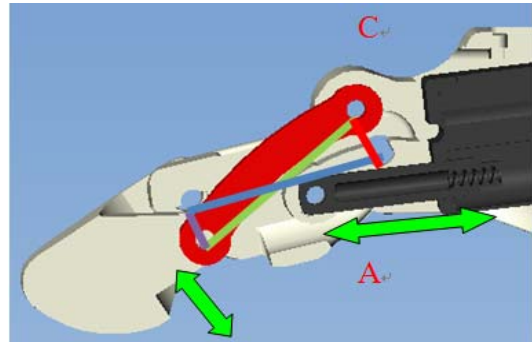


圖2. 指尖機構

機械手掌設計總計有14個關節，10個自由度，分別以10顆直線性伺服馬達，以連桿的驅動關節旋轉。為盡量降低體積與重量，手掌的外型貼近驅動機構及馬達的輪廓，以減低體積與美化造型。

我們採用最近的快速成型(Rapid Prototyping, 簡稱 RP)製作技術，以ABS工程樹脂做為成型材料。ABS工程樹脂的機械強度與剛性較前一代的壓克力材質優良。所有的電路板都鑲嵌在機械手掌的結構內，以降低總體重量與體積(圖3)。電路板的尺寸都是根據手指關節的外型而設計，總共有3種尺寸，分別位於手指的關節與手背。

完成後的機械手掌全長僅有95mm；包含馬達及控制電路板在內的重量則只有500克，無論重量及尺寸都與成人手掌相同。

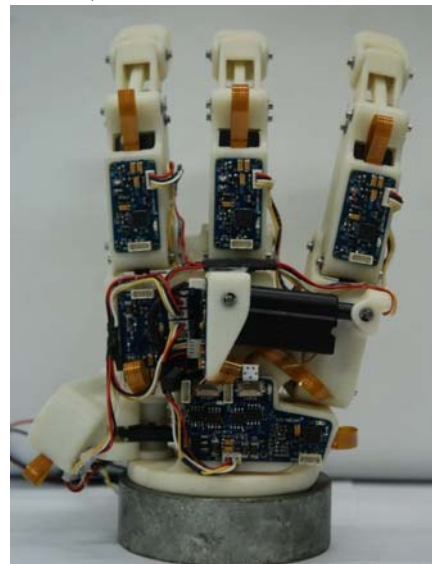


圖3. 機械手掌背部

## 3. 分散式控制系統

由於需要控制的馬達數目眾多，唯有使用分散式的控制架構，才能降低線路數目與體積。I2C網路雖然傳輸速度最高只有400 kHz的位元速度，但I2C的匯

流排架構，僅需2條信號線即可連結所有模組，形成一個分散式控制系統。所有模組都串聯在5Volts電源及I2C構成的匯流排(圖4)。

各馬達控制模組皆為獨立的子系統，透過I2C網路接收來自Master模組的位置、速度及控制模式等命令。Master模組是由分散系統中的一組控制模組兼任，負責傳達命令到其他9組Slave控制模組。Master模組配備有USB介面能與個人電腦或其他系統通訊，接收來自使用者(人型機械人)的命令。此外Master模組還有1MB的EEPROM，儲存預先設定的複雜動作指令，如握手，取物等。

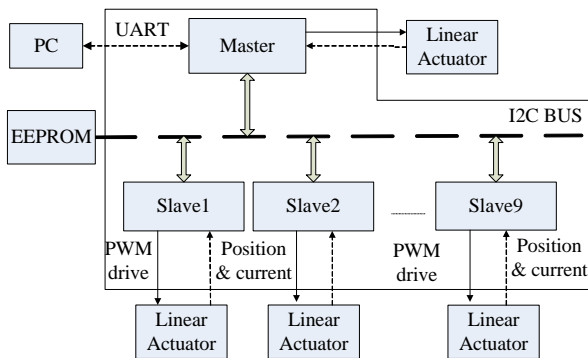


圖4 控制系統架構示意圖

每個馬達控制模組均以 16 位元微控制器 Microchip dsPIC33FJ128MC202 為核心，擁有一組額定連續電流 5A 的 20 kHz 的 Full H-Bridge PWM MOS-FET 驅動電路，馬達電位計電壓 AD 電路，馬達電流測量濾波電路，Universal Asynchronous Receiver Transmitter (簡稱 UART)與 I2C 埠各一組(圖5)。

UART 埠在開發程式時可與 PC Host 通訊，即時提供 PC Host 有關嵌入控制系統的電流，位置資料；以及調整控制器參數的用途。而主控制模組 (Master)，可透過 UART 埠接收上游電腦(機械人)的動作命令。

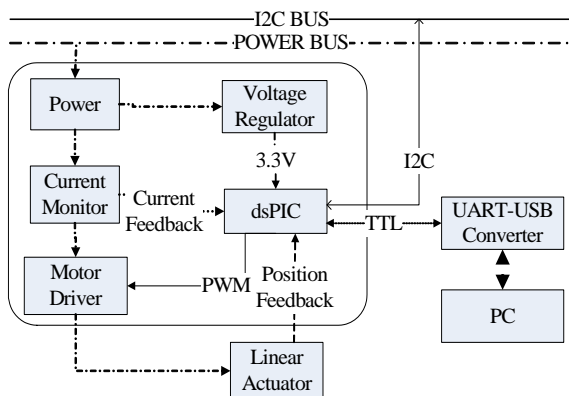


圖5. 伺服控制模組架構

#### 4. 伺服控制程式設計與實現

伺服馬達控制功能全由 Simulink 與 Stateflow 方塊圖建立(圖 6)。依照功能可以分為:路徑產生器 (motion profile generator)，RST 伺服控制及力量控制三個部份。

路徑產生器根據從 I2C 傳來的終點位置(target position)及速度命令，依照梯形速度(trapezoidal velocity profile)產生連續的二次多項式路徑，並在每一個取樣時間，計算下達給伺服控制器的位置命令(commanded position)。

力量控制是測量串聯馬達的 0.1 Ω 精密電阻的電壓降，經過低通濾波放大及 AD 轉換器，計算馬達的電流。根據電流的變化值，判定手指是否抓取到物體時，如判斷為已經達到預定的施力，則通知路徑產生器將目標位置設為目前手指關節的位置，停止進一步的移動，直到接收到新的位置命令為止。

RST 控制器負責控制 PWM 的 duty cycle，控制馬達的位置。採用包含 two-degree-of freedom 的控制架構[3]。

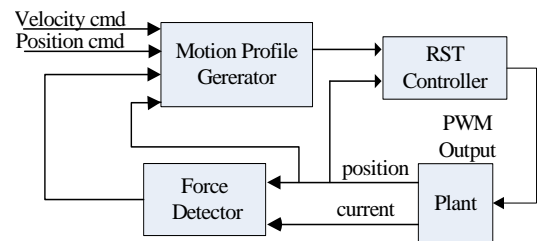


圖6. 伺服控制功能

常見的 Simulink Rapid Control Prototyping 工具以 xPC Target 最為常見。但是，PC-based 系統體積過大，無法應用於此機械手掌。目前支援 Microchip™ dsPIC 的 Simulink blockset 已有多家廠商。如 Microchip dsPIC blockset for Matlab。我們結合購買的 dsPIC 控制器硬體介面方塊，與自行以 Simulink 及 Stateflow 開發的控制函數與邏輯，自動完成產生 C 程式，編譯與下載微控制器的工作。

然後透過 UART 埠連結各控制模組，以 Matlab 即時調整控制參數，獲得最佳的控制效果。

##### 4.1 伺服控制設計

作者運用此系列的工具，進行系統參數鑑別，求出伺服馬達系統的 PWM Duty Cycle 輸入與馬達位置輸出之間的 Transfer function。

馬達的位置訊號是以馬達內部的線性電位計，在 200 Hz 取樣頻率下，以 10-bit AD 轉換。參數鑑別得到的伺服馬達 Transfer function  $G(s)$  為：

$$G(s) = \frac{2.2}{s(0.070s + 1)}$$

伺服控制採用 RST 控制理論設計[3]，包含 Feedforward、Feedback 的功能，控制器也有積分控制與 anti windup 的功能。閉路系統的性能等於頻寬為  $\omega_n = 13.3 \text{ rad/sec}$ ，damping ratio  $\zeta = 0.8$  的二階動態系統

$$G_c(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}。$$

為測試伺服控制器在定位控制與積分飽和的反應，控制器執行一個全程達 1/2 運動範圍的 step command (圖 7)。當控制器輸出達到輸出上限時，anti-windup 能夠避免 windup 造成輸出動盪的現象。而積分控制具有消除穩態誤差的作用。

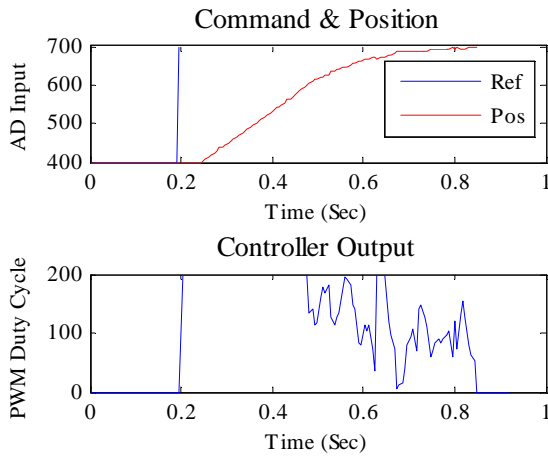


圖 7. 定位控制實驗結果

#### 4.2 路徑產生器

透過關節的運動路徑規劃，手指可以平緩地運動，避免劇烈的加減速導致振動。作者採用等加減速的梯形速度曲線路徑產生器 (trapezoidal velocity motion profile generator)，產生二次多項式的位移命令 (圖 8)。

梯形速度曲線分為等加速  $V_1 = a \cdot n \cdot h$ ，等速  $V_2 = V_{\max}$  與等減速  $V_3 = V_2 - a \cdot n \cdot h$  三段，其中  $h, n$  分別是取樣周期與取樣點的順序編號。

在預設的最大速度  $V_{\max}$  與固定加速度  $a$  下，路徑產生器判斷位移量  $S$  是否超過以等加速度  $a$  移動時，到達最高速度  $V_{\max}$  位移量的兩倍， $S \geq V_{\max}^2 / a$ 。當位移足夠時，加入一段等速度行程  $V_2 = V_{\max}$ ；否則只有  $V_1$  與  $V_3$  行程，在 1/2 行程處變換等加速度與等減速度。三段速度曲線  $V_{1..3}$  對應的位移量  $S_{1..3}$  是  $V_{\max}$  及  $a$  與全部位移量  $S$  的函數，可以預先計算加速度改變的位移處  $S_1, S_2, S_3$ 。

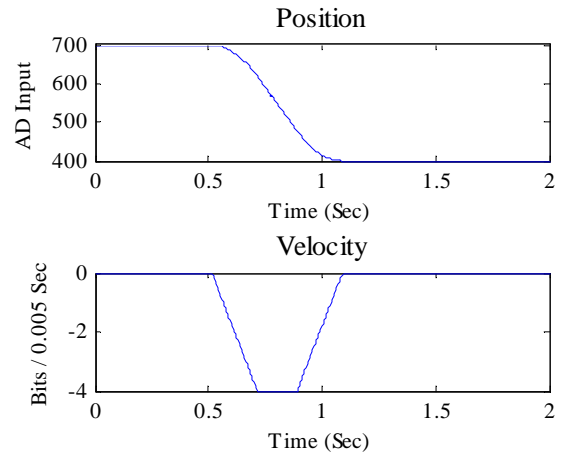


圖 8. 路徑規劃輸出

路徑產生器以積分器型式在 Simulink 建立：

$$V_{k+1} = V_k + a_k \cdot h$$

$$S_{k+1} = S_k + \frac{V_k + V_{k+1}}{2} \cdot h$$

而加速度  $a_k$  根據速度曲線的區域改成  $a$ ，0 及  $-a$ 。整個路徑產生器是以 Simulink 與 Stateflow 建立。執行伺服控制的典型結果顯示在圖 9。

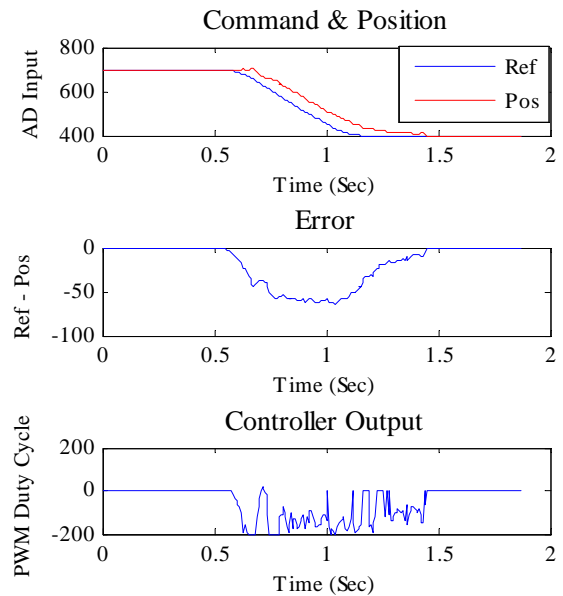


圖 9. 路徑控制實驗結果

#### 4.3 手指力量控制

機械手指的控制策略與工業伺服控制不同。工業伺服控制要求高定位精度以及迅速補償負荷的能

力；而手指控制對定位精度要求較低，但對環境需有敏銳的反應。在許多應用上需要力量控制或是順應控制(compliance control)[4]。力量控制常以測量手指力量作為控制回授，但是考慮伴隨的高製作成本，作者採用回授馬達電流的方式，以減低成本。透過串連馬達的電流測量精密電阻，將電阻電壓降經過放大與低通濾波處理與 AD 轉換後，傳回伺服控制器。路徑產生器根據電流(施力)回授信號，修正控制器的位置命令值，以達到近似順應控制的功能。

如以手掌抓取未知形狀的物品為例。當手指接觸到物體，電流對位移量的斜率  $\delta I / \delta x$  會改變，根據此斜率變化可判斷手指是否接觸到物體以及接觸物體的近似剛性值。路徑產生器會依照預設的邏輯，將控制器的位置命令更改為目前位置，停止進一步地移動。



圖10. 抓取新鮮雞蛋實驗

加入力量控制的手指在抓取物品時，可以使手掌的控制更為簡易。使用者毋須費心設定各關節位移，手指能自行適應不同形狀與軟硬度的物體，自動調整各關節的位移，毋須事先設定。

以抓取易碎的新鮮雞蛋為例。僅須對機械手掌下達閻掌的指令，當任一手指碰觸到蛋體時，力量控制將停止馬達運動。而其餘關節的馬達則繼續進行，直到所有手指都接觸到蛋體為止(圖10)。

#### 4.4 控制系統Simulink模型

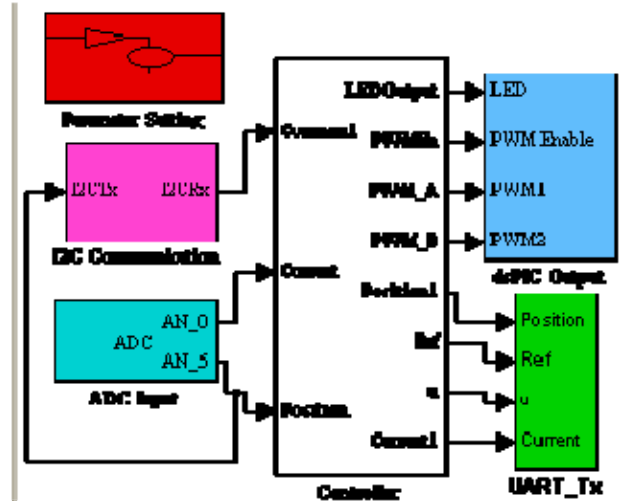


圖11. Slave Simulink Model

所有在嵌入式控制模組執行的程式都由 Simulink 方塊圖建立後，透過 Real Workshop Embedded Coder (簡稱 RTWEC)產生最後的執行程式。在控制手指關節的 Slave 控制模組 Simulink 模型(圖11)包含

- 硬體介面的方塊: slave I2C 與 UART 通訊；以及 AD，PWM 與驅動 LED 顯示燈的數位輸出入介面。其中的 UART 輸出方塊提供即時的內部資料以供偵錯之用。部分硬體介面方塊是自行以 C 程式開發。
- 控制器:包含伺服控制，力量控制與路徑產生器的部分，全是自行開發；
- 微控制器參數設定:設定微控制器的參數，例如 clock rate 及 watchdog timer。

在 Simulink 建立的模型，經過 RTWEC 產生 C 程式與 Makefile 後，啟動 Microchip 的 C30 編譯器產生 object code 後，使用 In-Circuit Debugger 下載到微控制器的 Flash ROM 中執行。

Master 控制模組(圖12)兼負接收上游控制電腦的高階指令與翻譯(translate)高階指令為單一關節運動指令的工作。所以其 Simulink 模組增加 UART 接收的介面與指令翻譯方塊(圖12中央 Zero Order Hold 方塊)與傳送關節指令的 I2C 方塊(I2C Master)。

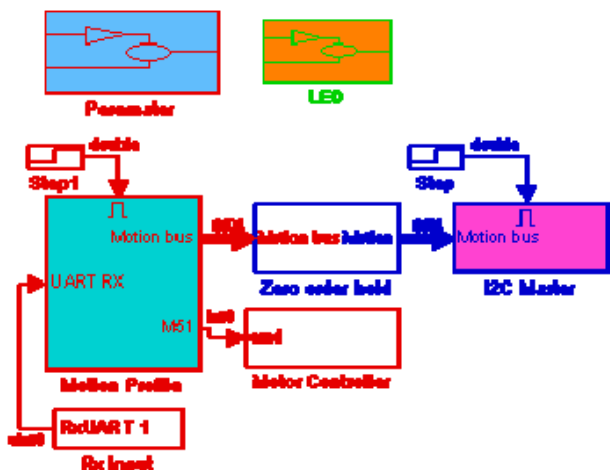


圖 12. Master Simulink Model

以 Stateflow 建立的高階指令翻譯方塊中，每個預先儲存的高階動作都有獨立的編號，其關節動作的詳細內容，所謂的字典，都儲存在 Master 的 Flash ROM 中。它採用 Stateflow 中 case-switch 框架，一旦 Master 模組從 UART 接收到高階動作的指令時，Master 模組即根據接收到的動作編號，查詢字典，再

將查詢所得的詳細的關節位移，速度與力量的指令下達到 slave 模組。

### 5. 手掌模擬器與人機介面

由於機械手掌的自由度數目眾多。即使簡單的握拳動作，也須逐一調整各個關節的角度，執行的速度與順序才能完成。雖然透過如 data glove 等 haptic devices 的教導方式也能完成，但是機械手掌與人手的結構並不相同，兩者的關節並非一對一的對應關係，所以仍須自行開發軟體將所需的動作，轉換為機械手掌的關節運動指令。

為簡化機械手掌動作的程式工作，作者設計一套 3D 的虛擬手掌控制介面。透過此介面的 3D 立體模型，使用者可以直覺地控制 3D 虛擬手掌任一關節的動作，並立即檢視手指的空間關係是否符合所需的手掌功用。此外，實際手掌的動作，除須考慮關節的最終角度之外，各手指關節的運動間，存在時間上的同步或協調的關係。優良的模擬器必須準確地模擬手指運動時的時間關係(time sequence)，並且能在模擬器上編寫手指關節運動的時間順序，建立手掌的動作劇本(script)。

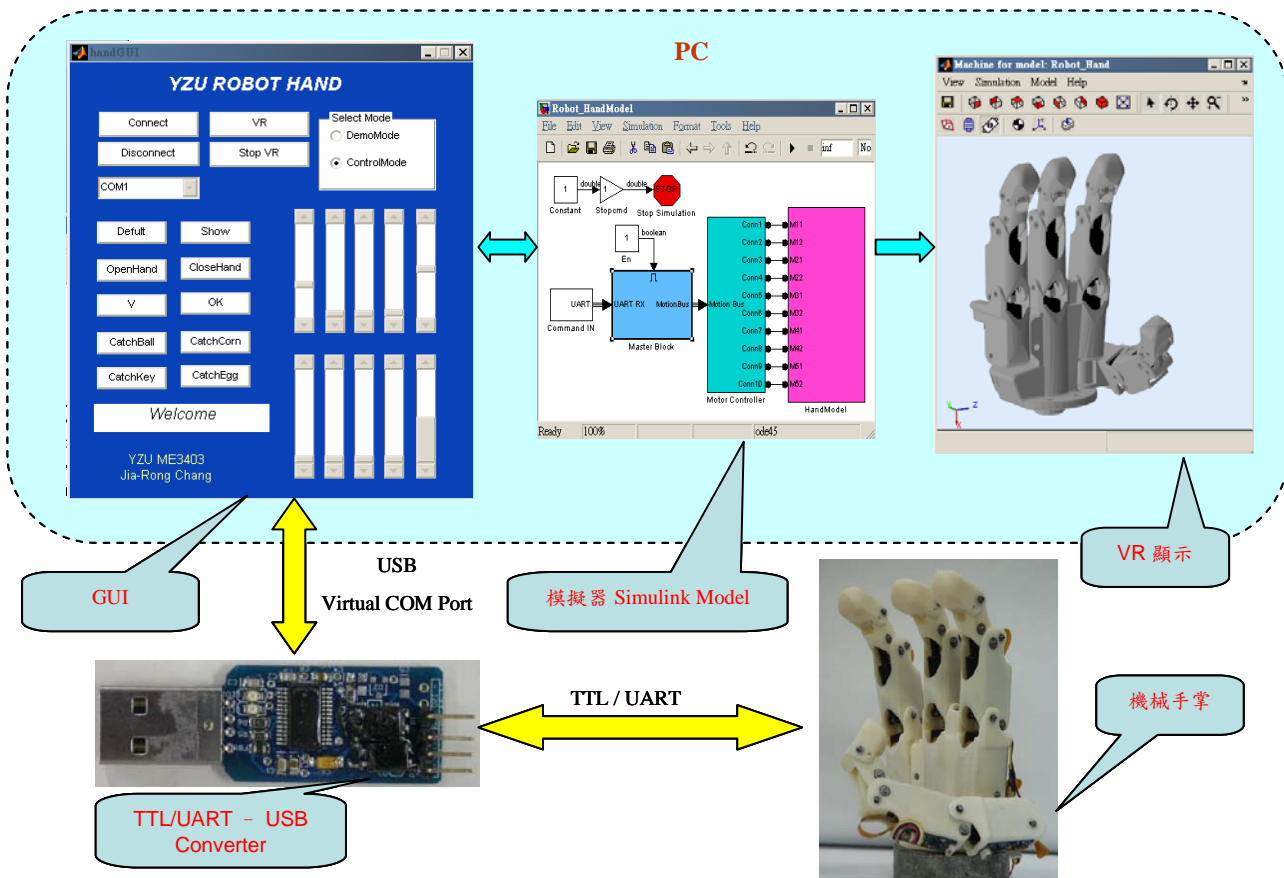


圖 13. 人機介面整體架構

此外，機械手指的實際運動時間是由閉路控制系統與機械手指共同決定。若希望虛擬手掌準確地模擬實際手掌的運動，則必須考慮運動控制系統的影響(dynamics)。因為馬達動態性能，路徑產生以及閉路控制器輸出飽和等因素，都會影響手指的移動時間。模擬器必須包含上述細節，才能得到逼真的結果。

由於馬達的動態模型，路徑產生與閉路控制系統的動態模型，在設計控制器時，都已建立 Simulink 模型，直接擴充現有模型成為模擬器是最簡易的途徑。

作者使用 Solidworks to Simmechanics translator 將原有 Solidworks 的機械手掌 CAD 模型轉成 Simmechanics 的 3D 模型。然後，結合運動控制系統與 Simmechanics 的模型，成為完整的機電整合模型(mechatronics model)。此模型包含馬達的動態模型，機構運動模型以及嵌入伺服控制系統的運動控制，輸出飽和與路徑規劃的非線性模型。

模擬器人機介面(簡稱 GUI)採用 Matlab 的 GUIDE 開發程式。GUI 整合 Serial Communication 與 3D 模擬器，在模擬時可同步控制實體的機械手掌。

使用者在 3D 機械手掌模擬器可用滑鼠直接修改各關節的角度，並立刻顯示手掌的運動結果。3D 模擬器可以減少設計劇本的時間。模擬器之輸入及輸出與實際系統相同，模擬器可從 UART/USB 介面，同步發出手掌(指)信號，控制實體的手掌。

以 GUIDE 建立所需 GUI 介面兩種工作模式：

- 動作編輯: 直接點選 3D 虛擬手指模擬器的手指關節，控制其運動位置，預視手指的動作。結合手指動作成為劇本。

- 劇本執行: 執行預先儲存的動作劇本。

3D 虛擬手掌模擬器的輸入指令與傳送給實體機械手掌的指令格式相同。兩者可以單獨或是同時動作(圖 14)。

## 6. 結論

本研究完成一具採用分散式控制系統的人型機械手掌。具有 4 指，10 個自由度。包含 10 組控制模組的分散式控制系統與機械手掌融為一體，總重量僅有 500 公克，而體積與人手相似。具有劇本與力量控制能力，可獨立完成複雜動作及抓取任意形狀物體的能力。

控制機械手掌所有的程式均以 Matlab/Simulink 建立，並透過 Real Time Workshop Embedded Coder 與 cross compiling 產生執行碼。每個控制模組都包含路徑產生、力量控制，伺服控制及 I2C 網路通訊等功能。

為簡化控制機械手掌的工作，作者以 Matlab

GUIDE 設計一項獨特的 3D 虛擬手掌模擬器。模擬器是以原有的伺服控制器，馬達動態等 Simulink 模型為基礎，結合從 Solidworks CAD 模型轉換 Simmechanics 模型，加上 GUIDE 開發的人機介面，而成為高真實度的機械手掌模擬系統，它不但顯示 3D 模型，也能直接控制連接的機械手掌，同時可以建立機械手掌的動作劇本，儲存在手掌內部的 Flash ROM 記憶體。

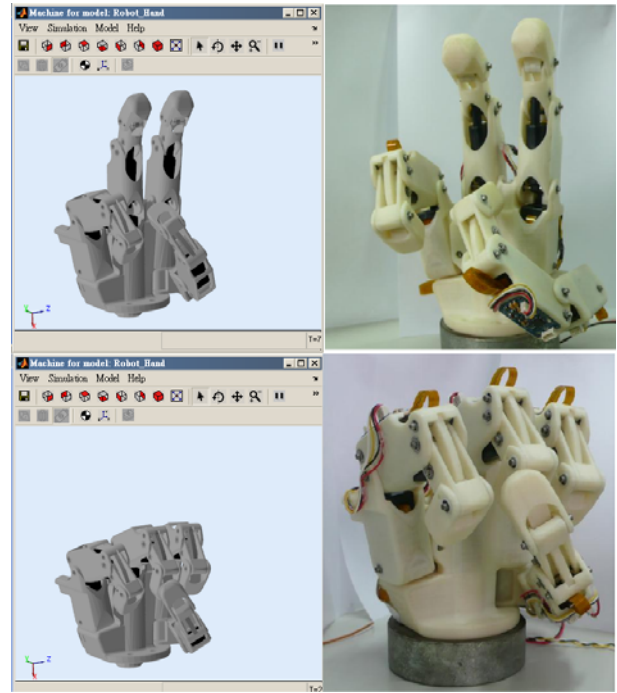


圖 14. 在模擬器同步控制 3D 模型與手掌

## 7. 參考文獻

- [1] H. Liu, P. Meusel, G. Hirzinger, M.H. Jin, Y.W. Liu, and Z.W. Xie, "The modular multisensory DLR-HIT-Hand: Hardware and software architecture," *Ieee-Asme Transactions on Mechatronics*, vol. 13, (no. 4), pp. 461-469, Aug 2008.
- [2] H. Liu, P. Meusel, N. Seitz, B. Willberg, G. Hirzinger, M.H. Jin, Y.W. Liu, R. Wei, and Z.W. Xie, "The modular multisensory DLR-HIT-Hand," *Mechanism and Machine Theory*, vol. 42, (no. 5), pp. 612-625, May 2007.
- [3] K.J. Åström and B. Wittenmark, *Computer-controlled systems (3rd ed.)*: Prentice-Hall, Inc., 1997.
- [4] H. Kazerooni, A. Chu, and R. Steger, "That Which Does Not Stabilize, Will Only Make Us Stronger," *The International Journal of Robotics Research*, vol. 26, (no. 1), pp. 75-89, 2007.

