

國立交通大學

電機與控制工程學系

碩士論文

可攜性力感系統之研發與應用

Development of a Portable Force Reflection System and its
Application



研究生：洪榮聰

指導教授：楊谷洋 博士

中華民國九十五年七月

可攜性力感系統之研發與應用

Development of a Portable Force Reflection System and its Application

研究生：洪榮聰

Student : Jung-Tsung Hung

指導教授：楊谷洋

Advisor : Kuu-Young Young



A Thesis

Submitted to Department of Electrical and Control Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

可攜性力感系統之研發與應用

研究生： 洪榮聰

指導教授: 楊谷洋教授

國立交通大學電機與控制工程學系

摘要

隨著網際網路的蓬勃發展，網路上的相關應用相當多樣化，人與人間的互動也變得不再因為距離而受到限制。在此情狀下，發展出許多的遠端應用，而對於目前的遠端呈現，聲音與影像都已相當普遍且技術上有很多的突破與創意，但對於力感的直覺式呈現方面，卻還相當的少，極大多數的力感呈現方式還停留在有向或無向的固定式震動居多。而本論文內容為建置一個可攜性的力感系統，連結現有的力回饋裝置、將所架構的系統藉由 TCP / IP 網路通訊協定於遠端及近端相互傳遞資訊，並以雙執行緒的運作方式使得回饋影像和回饋力感分開並同時進行；此外，討論在所建置系統運作的虛擬世界裡，如何對虛擬物體施力、與力感如何回饋等。而在整體系統架構與軟體設計流程建立，並建置完成系統後，將在虛擬世界中進行合力搬運應用，且對該應用過程中遇到的問題，提出解決的方式，以展現系統成效。

Development of a Portable Force Reflection System and its Application

Student: Jung-Tsung Hung

Advisor: Dr. Kuu-Young Young

Department of Electrical and Control Engineering

National Chiao Tung University

Abstract

The logo of National Chiao Tung University is a circular emblem. It features a gear-like outer border. Inside the circle, there is a stylized building or structure with the letters 'ES' and 'A' on it. Below the building, the year '1896' is inscribed. The entire logo is rendered in a light blue color.

As the Internet is getting more popular, the interaction of people is no longer limited by the distance. Meanwhile, there have been many researches in teleoperation and telepresence. For telepresence, techniques in sound and image are now more mature. By contrast, the sense of force is not yet well developed; mostly it is implemented on fixed patterns. Therefore, in this thesis, we develop a portable force reflection system, using existing force feedback joysticks. The system is composed of a server and many clients which are located in various remote sites. We use the Internet as a command transmission medium via the TCP/IP protocol, and use the double threads to deal with the image and force simultaneously. Key issues for system implementation are discussed, including system structure, signal flow, and force management. Experiments on join forces for object movement, are performed to demonstrate the effectiveness of the proposed system.

誌 謝

首先感謝我的指導教授----楊谷洋博士，在這兩年的研究期間，由於他熱心的指導，使我的研究工作得以順利完成。同時，感謝口試委員們：蘇順豐教授、宋開泰教授及陳永平教授撥冗參與論文口試，並給予許多寶貴的指導與建議，使我獲益良多。另外，謝謝許多學長給予我在研究上的討論與建議，還有其他在「人與機器實驗室」的夥伴們：木政、一元、豪宇學長，政龍、猷長、少麟同學以及又勳、李政、哲儒、怡康、博翔學弟們，在這兩年的實驗室研究生活中，由於你們的陪伴讓我的生活更多采多姿。最後要感謝的是我的家人以及我的朋友們，你們的關懷與支持使我能心無旁騖的完成學業。很高興自己在邁向人生另一階段的關鍵時刻，有這麼多美好的回憶陪伴著我，我會將它好好珍藏起來，讓它成爲我最美麗的回憶，謝謝大家!



目 錄

| | |
|-------|-----|
| 中文摘要 | I |
| 英文摘要 | II |
| 誌 謝 | III |
| 目 錄 | IV |
| 表 目 錄 | VI |
| 圖 目 錄 | VII |

| | |
|---------------------|----|
| 1. 導論..... | 1 |
| 2. 可攜性力感系統..... | 5 |
| 2.1 力互動..... | 5 |
| 2.2 系統架構..... | 6 |
| 2.2.1 Client 端..... | 7 |
| 2.2.2 Server 端..... | 9 |
| 2.3 網路傳輸..... | 12 |
| 2.3.1 時間延遲..... | 12 |
| 2.3.2 通訊協定..... | 14 |
| 2.4 多執行緒..... | 17 |

| | |
|-------------------------|-----------|
| 3. 力感復現..... | 19 |
| 3.1 力回饋裝置..... | 19 |
| 3.2 觸覺頻率..... | 23 |
| 3.3 力感復現..... | 24 |
| 3.3.1 搖桿系統模型..... | 24 |
| 3.3.2 對虛擬物體的力感復現模型..... | 28 |
| 4. 合作搬運應用..... | 32 |
| 4.1 合作搬運過程解析..... | 33 |
| 4.2 合作搬運應用之系統調整..... | 35 |
| 4.3 系統實踐..... | 38 |
| 5. 實驗..... | 40 |
| 5.1 實驗設計..... | 40 |
| 5.2 實驗結果..... | 45 |
| 5.3 遠距實驗..... | 48 |
| 6. 結論與未來工作..... | 53 |
| 6.1 未來工作..... | 53 |
| 參考文獻..... | 55 |

表 目 錄

| | | |
|-------|-----------------------|----|
| 表 3-1 | 力回饋搖桿規格比較..... | 22 |
| 表 3-2 | 系統模擬時視覺/觸覺所需相關參數..... | 24 |



圖目錄

| | | |
|-------|--|----|
| 圖 1.1 | 典型的遠端操作系統示意圖..... | 2 |
| 圖 1-2 | 隔洋握手操作示意圖..... | 3 |
| 圖2-1 | Server-Client 架構圖..... | 7 |
| 圖2-2 | 穿透物體時的搖桿位置與虛擬機械臂位置..... | 8 |
| 圖2-3 | Client 端接收與傳送資訊示意圖..... | 8 |
| 圖2-4 | Server 端系統決策流程圖..... | 9 |
| 圖2-5 | Server 端系統示意圖..... | 10 |
| 圖2-6 | 系統架構的資訊流..... | 11 |
| 圖 2-7 | 通訊延遲示意圖..... | 12 |
| 圖 2-8 | 遠端操作往返一趟所需時間示意圖..... | 13 |
| 圖 2-9 | 雙向遠端操作系統訊號流程圖..... | 13 |
| 圖2-10 | WinSock 應用程式介面..... | 15 |
| 圖2-11 | 連接導向式 (TCP) 的 Client-Server 程式流程圖..... | 16 |
| 圖2-12 | 單執行緒與多執行緒的執行順序和時間關係..... | 17 |
| 圖2-13 | Server 端雙執行緒程式流程圖..... | 18 |
| 圖 3-1 | 力感操控系統控制流程..... | 19 |
| 圖 3-2 | 力回饋裝置示意圖..... | 20 |
| 圖3-3 | (a) Laparoscopic Impulse Engine (b) Impulse Engine 2000..... | 21 |
| 圖 3-4 | 系統阻抗圖：真實世界和虛擬環境..... | 26 |
| 圖 3-5 | 物體受力與反應力方向..... | 28 |
| 圖 3-6 | 以距接觸面切線距離為主的力感模型..... | 29 |
| 圖 3-7 | 以距最近表面距離為主的力感模型..... | 30 |
| 圖 3-8 | 以實際位置和接觸位置距離為主的力感模型..... | 31 |

| | | |
|--------|---|----|
| 圖 4-1 | 合作搬運示意圖..... | 32 |
| 圖 4-2 | A、B 同時對物體施力解析..... | 33 |
| 圖 4-3 | 造成滑動的示意圖..... | 35 |
| 圖 4-4 | 修改合力後推動示意圖..... | 36 |
| 圖 4-5 | 多人時的合力方式..... | 37 |
| 圖4-6 | 針對合力搬運因應調整的系統資訊流通架構..... | 39 |
| 圖 5-1 | 實驗設備與進行環境..... | 41 |
| 圖 5-2 | 測定搖桿輸出力..... | 42 |
| 圖 5-3 | 量測得的輸出力值與理想的輸出力值..... | 42 |
| 圖 5-4 | 搬運測試 (a) 水平方向 (b) 垂直方向 (c) 圓形軌跡..... | 44 |
| 圖 5-5 | 水平向搬運的誤差與完成時間..... | 45 |
| 圖 5-6 | 垂直向搬運的誤差與完成時間..... | 46 |
| 圖 5-7 | 圓軌跡搬運的誤差與完成時間..... | 46 |
| 圖 5-8 | 水平向搬運的誤差與完成時間 (針對合力應用)..... | 46 |
| 圖 5-9 | 垂直向搬運的誤差與完成時間 (針對合力應用)..... | 47 |
| 圖 5-10 | 圓軌跡搬運的誤差與完成時間 (針對合力應用)..... | 47 |
| 圖 5-11 | 原系統與調整後系統 (a)水平向 (b)垂直向 (c)圓形軌跡 搬運..... | 48 |
| 圖 5-12 | 實驗室測得的網路延遲狀況..... | 49 |
| 圖 5-13 | 家用網路所測得的網路延遲狀況..... | 50 |
| 圖 5-14 | 實驗室網路進行水平向搬運過程中虛擬臂的位置..... | 50 |
| 圖 5-15 | 家用網路進行水平向搬運過程中虛擬臂的位置..... | 51 |
| 圖 5-16 | 家用網路進行水平向搬運過程中施力的狀況..... | 52 |

第一章

導論

在今日電腦網路盛行的時代，網際網路的蓬勃發展對人們的日常生活造成一定的影響，而網路上的相關應用也相當地多樣化，使得人與人間的互動不再因為距離而受到限制，例如網頁瀏覽 (web browsing) 或者電子郵件 (e-mail) 等，幾乎成為現代人們生活中不可或缺的一部分。而在其多樣化的應用中，遠端應用即是其中的一個重要部分，例如為了讓身在遠端的機器人和人一樣可以適時做出適當的決策，Goertz 與 Thompson 於 1954 年提出遠端操控 (teleoperation) [14]，將人類與機器人兩者的優點結合，既可讓人類的感測與操控能力延伸到遠端，又可以使操縱者處於安全的地方來進行操作；另外，在 1986 年由 Jaron Lanier (the founder of VPL Corporation) 首先提到的虛擬實境 (Virtual Reality, VR) [10]，其發展也從實驗室走入一般大眾的日常生活與工業應用，例如 3D 的網路遊戲、房屋仲介的線上看屋系統、玩具製造公司的建模模擬等。

當進行遠端應用時，需要將各種資訊回傳到主控端給操縱者，像是視覺、聽覺、觸覺，甚至是嗅覺及味覺等感覺資訊，使操縱者可以感受到遠端環境的變化情形，將遙不可見的遠距景物清晰的呈現在使用者的面前，此項技術稱為遠端呈現 (Telepresence) [3]。遠端呈現這個名詞最早在 1983 年由 Akin 等人 [3] 定義，其目的是用於太空科技上，以幫助太空人在外太空作檢修的工作，其定義如下：“在執行任務端，操縱器 (manipulator) 具有靈活性，能讓使用者完成一般人類的動作。而在控制端，使用者接收足夠感覺資訊，並提供一種彷彿如實際身在現場的感覺”。而所謂的足夠感覺資訊，就人主要的感官

感覺來討論，一般以視覺、聽覺、以及觸覺為主，所以主要的人機介面裝置也是用來處理影像、聲音、與力量方面的資訊。虛擬實境發展之初以立體影像 (stereoscopic image) 為主，接著加入環場音效 (surround sound) 以增加臨場效果；而發展較晚的觸覺回饋，早期應用在遠端操作 (teleoperation) 上，用來復現遠端機器人在順應性工作下的力資訊情況，協助操作員控制遠端機器人完成任務 [15,5]。但為了讓人們能單純地從獲得刺激到更進一步發展與虛擬的環境互動，以期營造出更具真實的虛擬世界，由於力感是最直接與環境互動的介面，所以觸覺回饋裝置漸漸受到重視，在 1960 年代後即被大量應用於虛擬實境的回授刺激上 [1,7]。而對於加入力覺或觸覺回饋裝置後的典型遠端操作系統，呈現於圖 1-1 中，圖中人類操作者於近端對力回饋搖桿進行操作、搖桿將獲得的資訊交給電腦運算處理，電腦運算後即再將控制命令下達給遠端機械臂的驅動裝置、以驅使機械臂依循控制命令執行動作，而機器臂上的感測裝置不斷地把目前機械臂的位置與感受到的受力傳回電腦處理，電腦將其一部份轉換成視覺資訊顯示於螢幕、另一部分交由力回饋裝置施行力感回饋。

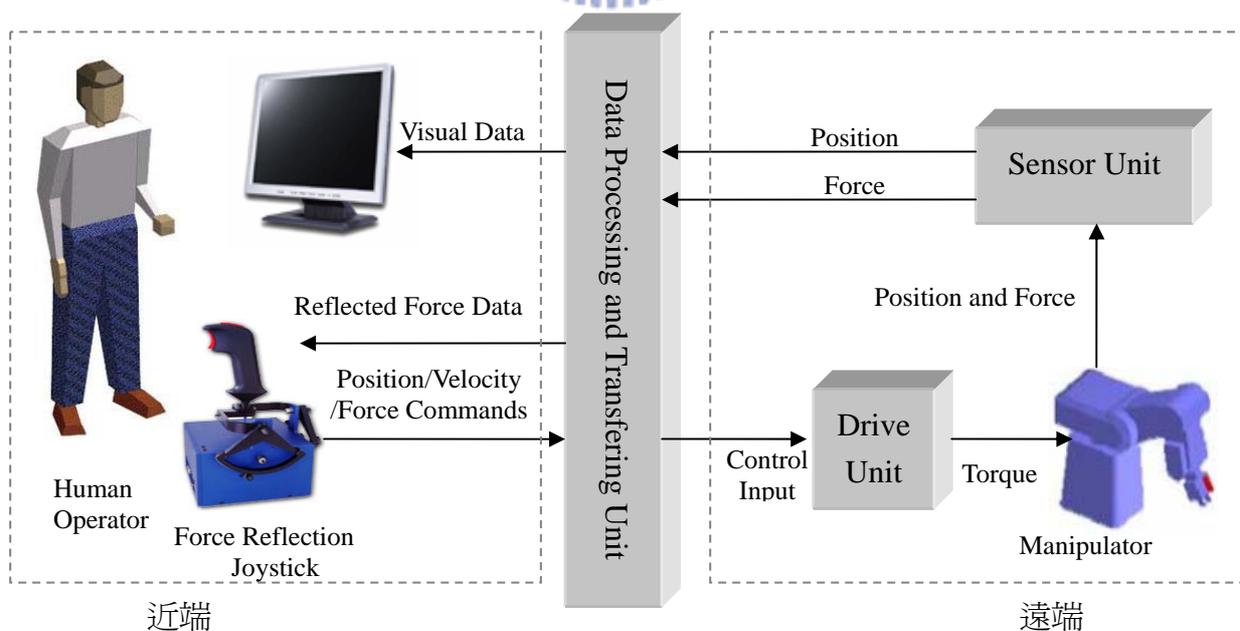


圖 1-1 典型的遠端操作系統示意圖

在目前的遠端呈現，聲音與影像已經發展多時且技術上有很多的突破，然而對於力感的直覺式呈現方面，除了工業界或是需要精密操作的遠端環境外，一般大眾常接觸到的力感、或說是觸感的呈現方式，極大多數仍停留在有向或無向的固定式震動居多，其力感、觸感表現大多很粗糙，而原因是因為取得設備的價格昂貴，使得普及推廣有一定的難度；但隨著科技的不斷進步，各式資訊裝置的成本已逐漸低廉，在預期硬體可以大眾化的情況下，未來力感的精緻化與普及化將會是必然的潮流，像是立體雙聲道幾乎已完全取代了單聲道一樣。例如在 2002 年，英國倫敦大學學院與美國麻省理工學院這兩地的科學家，透過網際網路成功示範的「隔洋握手」 (Virtual hands reach across ocean) [19]，在未來相信會是很一般、直覺式的網路互動行為，所謂的「隔洋握手」如圖 1-2 所示，即雙方研究員透過小型機器臂 Phantom 隔洋合力把一個電腦驅動的方型盒子抬起來，能夠清楚感受到對方的推拉力道，並從電腦螢幕了解彼此是如何移動方形盒子。但上述的隔洋握手提及到的力資訊與觸覺的流通，是建立在每秒 10Mb 的網路傳輸速度上，所用的是光纖或高速寬頻網路，而非一般家庭用的網際網路，但在可預期的今後，一般家用網路傳輸必定會朝比此更高的傳輸速度邁進。

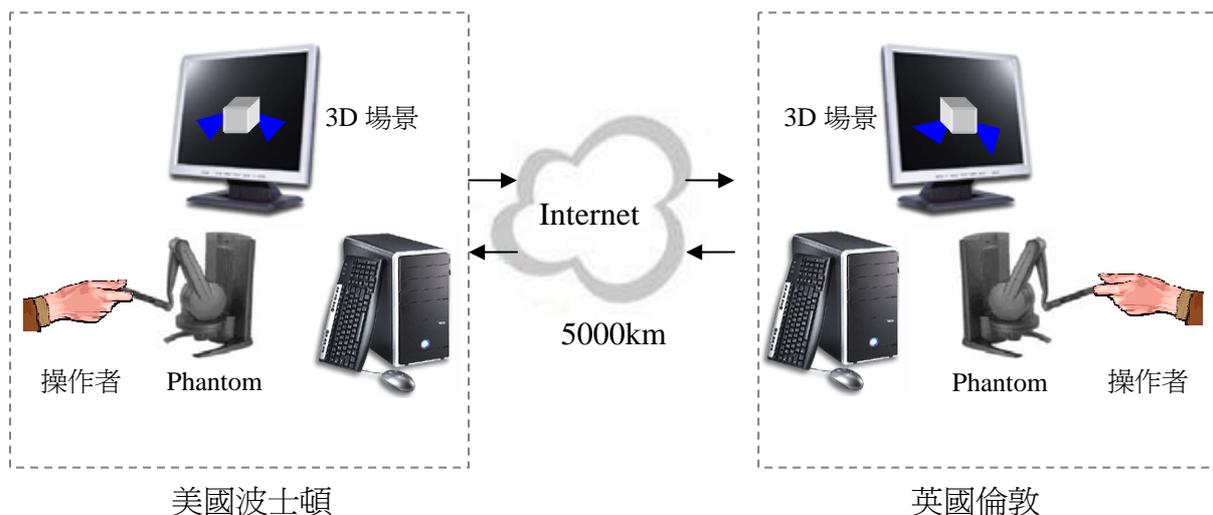


圖 1-2 隔洋握手操作示意圖

如前述的互動行爲，若欲推廣至大眾，勢必要讓軟體與硬體皆可輕易的安裝。故本論文的目的，爲發展一可攜性的力感系統，讓使用者可以輕易地帶著力感裝置至不同的地方安裝，經由電腦網路與在遠方的其他使用者即時互動，並且能操作各自的力回饋裝置，一起在同一個虛擬環境中進行活動。而對於目前已發展的設備，諸如力回饋搖桿 (force-reflection joystick)、感應手套 (Sensing glove)、空間球 (Spaceball)、3D 滑鼠 (3D mouse) 以及感測衣 (Body suit) 等，這些雖都可呈現細膩、正確方向與大小的觸感、力感，但以可以爲大眾輕易取得、操作易懂、建置容易、且技術較成熟爲前提下，本文中將選用力回饋搖桿作爲發展實現力感的工具。而在決定好了力回饋裝置與系統的建置後，我們也著手進行了合作搬運的測試，了解對於合作搬運此一應用上，可能會遇到的問題，並就遇到的問題提出討論且解決之。



本論文將於第二章介紹所建置的可攜性力感系統，說明本實驗所架構的互動式遠端操作系統，包含基本架構與系統組成；第三章依據說明如何利用系統中的力回饋裝置，建置並復現虛擬世界中的力感；第四章則是針對在合力搬運的應用上所做的調整；第五章利用所發展的力回饋系統進行合力搬運實驗；最後，第六章提出結論以及未來發展的方向。

第二章

可攜性力感系統

本章目的在於發展一建構於網路之上的通用性力感系統。首先說明針對於一個多人的互動式遠端力感系統，根據力的合作、對抗的數種可能性，探討其互動關係，據以建立系統的基本架構；接下來討論完整的系統架構，並描述相關的軟體設計。

2.1 力互動模式



力的互動方式大致可以分爲兩大類：合作與對抗。合作就像是雙手合力一起搬運某個物件，其好處是減輕單手施力的負擔，爲的是更易於完成工作；對抗就好比兩個人在比腕力，爲的可能是樂趣、也有可能是爲了阻止對方的危險、不當地動作。合作的議題一直是被廣泛討論的，像是以一支機器手臂去控制多支機械臂進行合作搬運的控制方法 [6]，或像是常用在人與機器手臂的互動上的 Impedance control [17] 與 Compliance control [13]等，每當有不同的任務需求，就會發展出不同的控制方式因應。在以“合作”爲主要目的的前提下，當然希望彼此的施力不會產生互相對抗的情況，因此前面提到的控制方式，其作法大多是盡量去削減可能會受到的對抗力量；但在直覺情形下，如果兩個人位於一物體的不同側，其中一人施力於物體使其推向另一人，另一側的人應很自然地感受到物體被推過來的壓力，而如果經過前面的控制方式修改，可能另一側人感受到的推力會不自然，因此在建立系統時，無論是爲了以後的發展性，還是現階段的通用性，

應該將模擬的環境依自然界的物理法則建置，如此則可以隨意切換進行力的對抗或合作。

另外，假使擁有多個力回饋裝置，可能達成的合作就更多了，像是對一持物體的機械手臂，可以以一個力回饋裝置進行手臂空間位置的控制，另一個力回饋裝置進行機械臂手腕端持物的角度與動作控制；而控制這多個力回饋裝置的方式，可以是一個人自己以雙手同時進行，或是多個使用者各自控制各個裝置進行。而我們在此發展的系統所做到的即是透過網路連結多個力回饋裝置於同一個環境，以進行各種可能的合作。

2.2 系統架構



在系統的建置中，因為要連結來自不同地方的力回饋裝置，使其置於同樣的模擬環境中，故需透過網路來達成。而對於各種常見的遠端系統，Server-Client 架構是最廣泛運用的網路通訊模型，又稱之為 Master-Slave 主從式架構，其概念為 Client 先對 Server 端提出要求，而後 Server 對 Client 所提出的要求提供適當的服務，接著將結果再傳回 Client 端；如圖 2-1 所示，當多個 Client 連結至一個 Server，由 Server 端進行各 Client 端提出的運算要求，再將結果送回 Client 端，此架構的好處是，可以確保 Server 端所在的模擬環境是針對所有的資訊進行處理，此模擬環境可以維持穩定。而如果採取分散式的系統架構，即沒有明顯的 Server 與 Client 之分，任一台主機（操作端）都是 Server、也同時是 Client，則整體系統的設計難度將會提高很多，要達成各操作端的同步性是更加困難，因此像現在的網路遊戲，主要還是採用 Server-Client 架構。而本論文也將以 Server-Client 架構為發展，在 Server 端建置虛擬的環境，接受 Client 端的連線，使各個 Client 端操作者可以藉力回饋裝置活動於該虛擬環境。而 Server

與 Client 端所各自司責的工作，以下敘述。

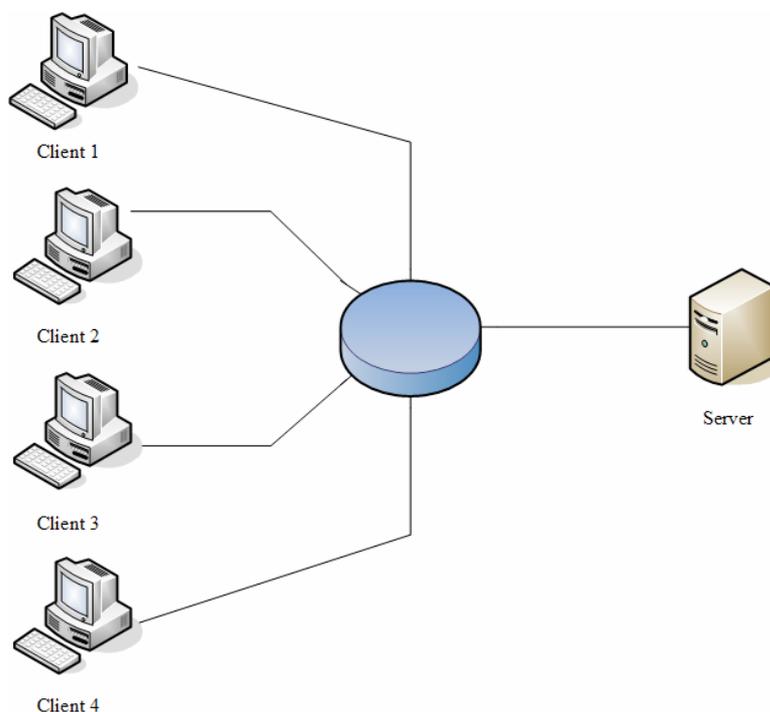


圖2-1 Server-Client 架構圖



2.2.1 Client 端

Client 端的操作者對搖桿進行操作，並對 Server 端提出連線要求，發送 Client 端目前搖桿的位置資訊至 Server 端，而在 Server 端有一對應於 Client 端搖桿位置的虛擬機械臂 (Manipulator)，當其接觸虛擬物件時，會對虛擬物件產生作用力，然而虛擬機械臂的位置並非一直與搖桿位置相同，其位置在未接觸物體時是與搖桿相同，但接觸時、搖桿位置要穿透物體表面以計算對物體的施力，然而此時虛擬機械臂必須停留在搖桿穿透入物體瞬間時的接觸點上，否則操作者將難以判斷施力方向與施力點，圖2-2顯示當搖桿位置穿透物體表面時，搖桿實際位置與虛擬機械臂位置的差異。

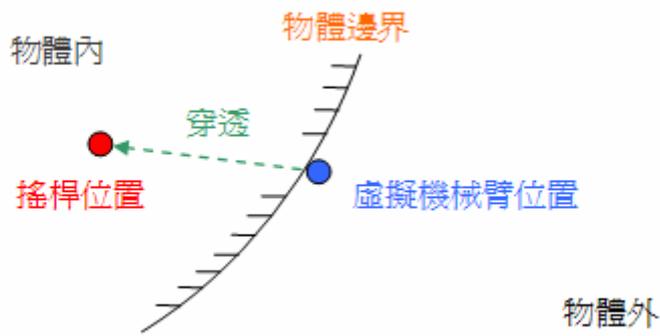


圖2-2 穿透物體時的搖桿位置與虛擬機械臂位置

因此，爲了利於操作者進行判斷決策，必須在把搖桿目前位置傳送至 Server 端、進行與模擬環境互動的同時，不斷地從 Server 端接收自身的虛擬手臂位置與其他 Client 操作者的虛擬手臂位置，以及虛擬物件目前位置，還有執行從 Server 端傳回、要 Client 端力回饋裝置執行的力資訊，所有 Client 端所需要接收與發送的的資訊，皆標示於圖2-3，如上所述。



圖2-3 Client 端接收與傳送資訊示意圖

2.2.2 Server 端

在 Server 端方面，就架構功能而言，負責自身任務要求需要的運算與接受來自 Client 端的運算要求。在系統中，依據遠端 Client 端所傳來的位置資訊進行判斷是否有對模擬物件產生接觸事件，一旦發生接觸時立即進行力資訊運算，得出作用於物體的力資訊與應回饋於 Client 端的回饋力，且將回饋力傳回 Client 端，並同時負責計算施與物體之力資訊對物體產生的反應，其決策流程如圖2-4所示。另外，因為也在 Server 端電腦裝置了力回饋搖桿，使得 Server 端電腦本身亦扮演了 Client 端的角色，換言之，也同時要處理來自 Server 端操作者的要求，圖2-5為 Server 端系統示意圖，顯示了要接收處理與要發送的資訊。因為 Server 端必須不斷地傳遞資料至各 Client 端、還有進行力資訊和模擬環境的運算，在此情況下，系統本身與網路的負荷皆會較重，要以比較高速的電腦擔任此項任務。

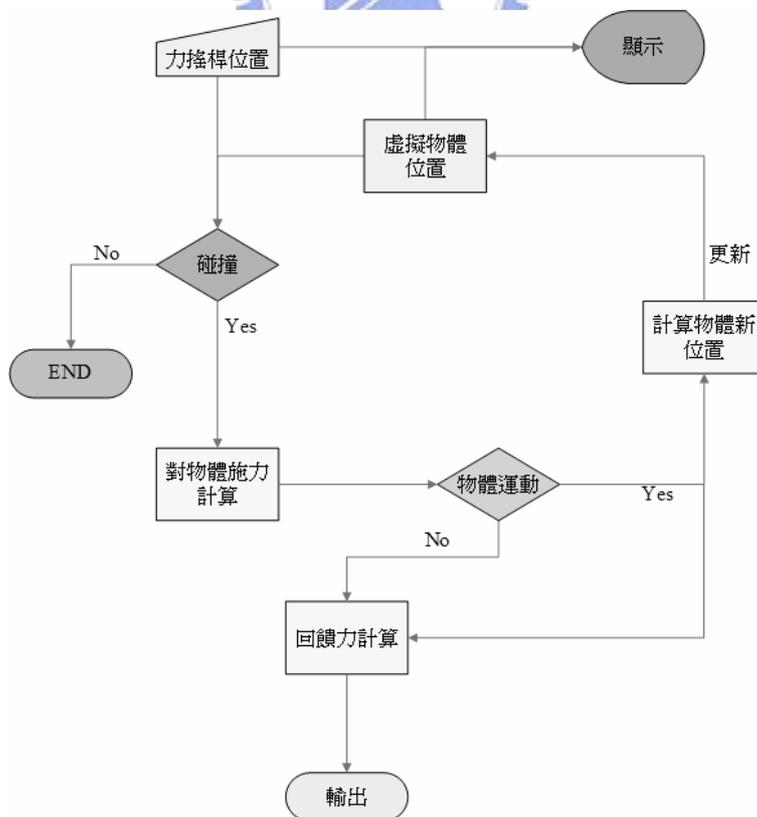


圖2-4 Server 端系統決策流程圖

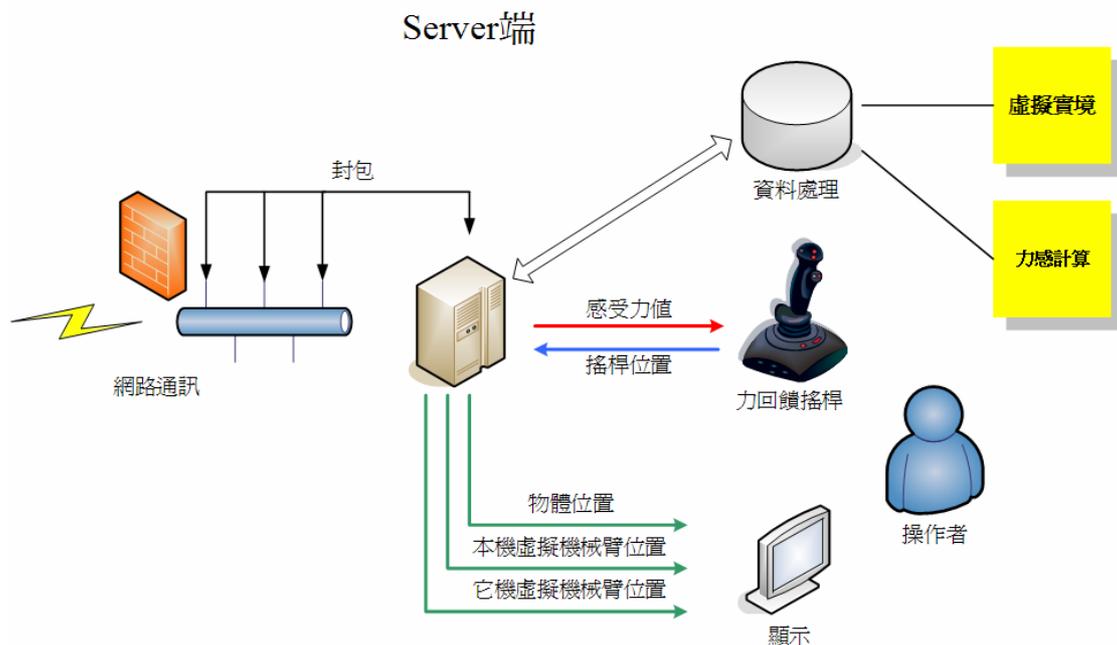


圖2-5 Server 端系統示意圖

我們所建構的系統是基於力回饋搖桿位置來進行系統的控制模擬，所以綜合上述的 Server 端與 Client 端系統，可以圖2-6簡單地呈現出實踐系統完整的系統資訊流通（在此圖中省去了 Client 與 Server 端的網路通訊部分），在圖中 Server 表示 Server 端操作者（力回饋搖桿），Client 表示 Client 端操作者（力回饋搖桿），M-Server 指對應於 Server 端力回饋搖桿的虛擬機械臂，M-Client 指對應於 Client 端力回饋搖桿的虛擬機械臂， \vec{f}_s 表示 Server 端力搖桿對虛擬環境的施力， \vec{f}_c 表示 Client 端力搖桿對虛擬環境的施力， \vec{f}_{rs} 表示由虛擬環境回傳欲使 Server 端操作者感受的回饋力， \vec{f}_{rc} 表示由虛擬環境回傳欲使 Client 端操作者感受的回饋力， \vec{x}_s 表示 Server 端力搖桿的位置， \vec{x}_c 表示 Client 端力搖桿的位置， \vec{x}_s' 表示由環境回傳的 Server 端虛擬機械臂的位置， \vec{x}_c' 表示由環境回傳的 Client 端虛擬機械臂的位置。圖中的資訊流動，

以 Server 端例，Server 端力搖桿的位置 \vec{x}_s 在輸入進虛擬環境後，即轉換成 M-Server 虛擬機械臂的位置 \vec{x}_s ，當在未接觸虛擬物體前，兩者位置同步，然而一旦接觸虛擬物後，因為搖桿實際位置穿透物體表面、但虛擬臂卻因要吻合物理現象位置而不可穿越，所以虛擬環境要給予 M-Server 虛擬機械臂應停留的表面位置 \vec{x}_s' ，M-Server 虛擬機械臂的位置會變成這個新的位置 \vec{x}_s' ，直到不在接觸物體時才會回復到 \vec{x}_s ；根據 \vec{x}_s' 與 \vec{x}_s 的差值再乘上一個放大比例 K ，就可以獲得欲使 Server 端操作者感受的回饋力 \vec{f}_{RS} ，而其反向也就是 Server 端力搖桿對虛擬環境的施力 \vec{f}_s 。

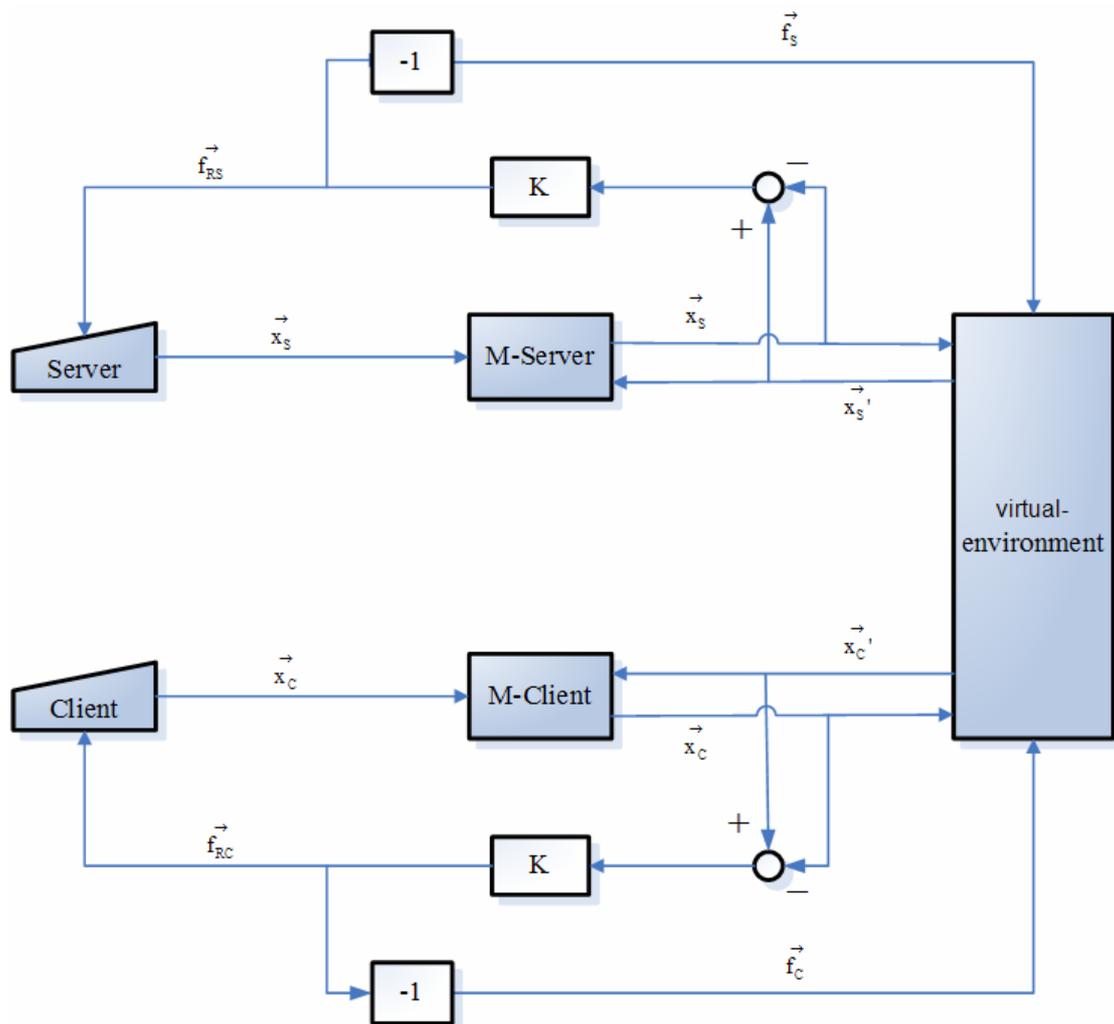


圖2-6 系統中的資訊流通

2.3 網路傳輸

本節首先討論一遠端系統在網路上會遇到的訊號延遲問題，接著介紹實作系統網路部份時，所使用的通訊協定，以了解此系統的傳輸方法。

2.3.1 時間延遲

資料在網路上傳輸會有時間延遲的現象，典型通訊延遲主要由三種部份所組成，第一部份為傳輸時間 (transmission time)，第二部份為傳播延遲 (propagation delay)，也就是信號從電路上一點傳播到另一點所需的時間，又稱網路延遲 (network latency)，第三部份為處理延遲 (processing delay)，也就是從請求信號發出到請求信號到處理設備接收到請求信號所需的時間，又稱排隊延遲 (queuing delay)，如圖 2-7 所示。

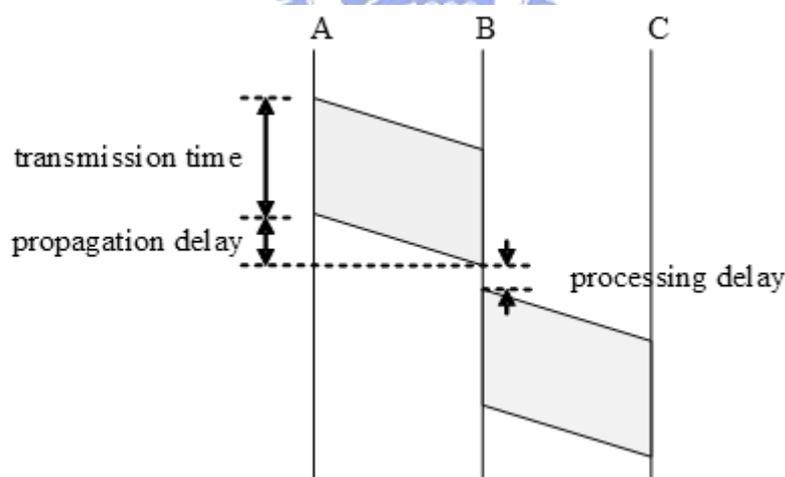


圖 2-7 通訊延遲示意圖

而遠端操作一個完整來回的操作所需花費的時間，包含了以下四個部份，分別為由遠端系統傳回給近端操作者的網路通訊延遲 t_1 ，操作者作決定所造成的時間延遲 t_2 ，將控制命令傳給遠端系統的網路通訊延遲 t_3 ，以及遠端系統執行

動作所花的時間 t_4 ，將 $t_1 + t_2 + t_3 + t_4$ 就是一個迴圈所需的全部時間，如圖 2-8 所示。

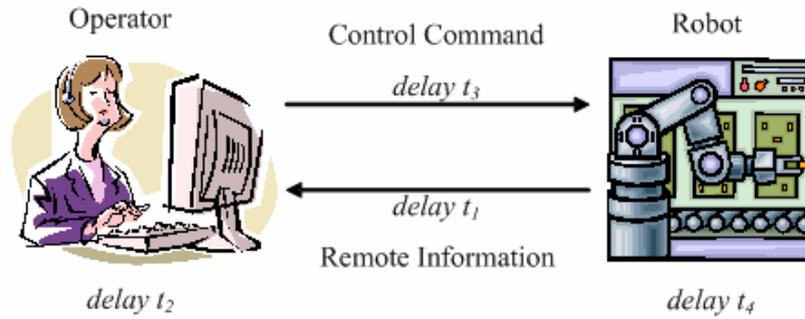


圖 2-8 遠端操作往返一趟所需時間示意圖

例如典型遠端操控系統雙向訊號流程圖，如圖 2-9 所示，圖中下標 m 代表 Local site， d 代表 delay， h 代表操作者， e 代表遠端環境， x 為位置資訊， v 為速度資訊， f 為力資訊， t 為時間， $T_1(t)$ 和 $T_2(t)$ 則分別代表由 Local site 端傳到 Remote site 端以及 Remote site 端傳到 Local site 端的傳輸時間延遲函式。各訊號之間的關係，如下所示：

$$x_{md}(t) = x_m(t - T_1(t)) \quad (2.1)$$

$$v_{md}(t) = v_m(t - T_1(t)) \quad (2.2)$$

$$f_{hd}(t) = f_h(t - T_1(t)) \quad (2.3)$$

$$f_{ed}(t) = f_e(t - T_2(t)) \quad (2.4)$$

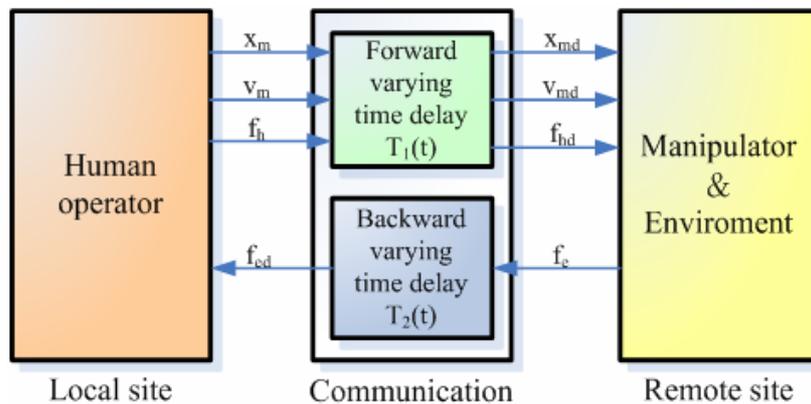


圖 2-9 雙向遠端操作系統訊號流程圖

訊號流程首先 Client 端透過網路傳送位置、速度或加速度等命令，操控 Server 端系統，藉由 Server 端操控器與環境間的互動，傳回接觸時的力到 Client 端。另外，網路上因存在路徑選擇等問題，也有可能造成資料遺失的現象，且網路上時間延遲的現象是時變的，造成雙向延遲不對稱，使得系統在穩定與分析上更加困難。

2.3.2 通訊協定

實作裡使用 WinSock 來建構網路通訊介面，再藉由 TCP/IP 通訊協定與 Ethernet 來達成 Server 端 Client 端的連接。而 TCP/IP 通訊協定並不是只有 TCP/IP，實際上它是很多協定的組合，TCP/IP 協定家族是由 TCP、UDP、IP、ARP、RARP、ICMP 等所組成的，以下稍微對 IP、TCP 與 UDP 作介紹：

IP (Internet Protocol) / 網際網路協定：

IP 位於網際網路層，主要工作是切割封包與選擇封包傳送的路徑，由於 IP 只負責將資料傳送到目的地，而不作任何錯誤檢查與控制，因此為非可靠性傳輸，需要靠上層的 TCP 作偵錯的動作。

TCP (Transmission Control Protocol) / 傳輸控制協定：

TCP 位於傳輸層，採用連接導向模式，具有資料接收確認回應與兩端相互維護封包順序號碼之特性，為可靠性傳輸，適合用在需要可靠傳輸而不希望資料傳輸錯誤的地方。

UDP (User Datagram Protocol) / 使用者資料傳施協定：

UDP 位於傳輸層，採用無連接模式，對於發送端送出的資料封包並不具有順序號碼，接收端也不會有所回應產生，雖然較不可靠，但與 TCP 比較起來具有較少的額外負擔 (overhead)，此外，UDP 還有可作多點投射 (multicasting) 與廣播 (broadcasting) 的好處。

雖然在許多的論文提到的實驗環境裡 UDP 因其快速性被作為主要的傳輸協定，但由於我們希望此系統可以達到一定的通用性與穩定性，以便可以在任意一台電腦與一般的家庭網路上建立此模擬環境，不希望在過程中發生資料丟失的情形，所以不採用 UDP，而採用 TCP 為主要協定。

因為實現網路通訊的平台是在 Microsoft 公司的 Windows 作業系統上進行，所以使用 WinSock 此公開在 Microsoft Windows 下發展網路程式的應用程式介面 (Application Programming Interface, API) 進行實作，其概念如圖2-10，它是上層連結了使用 WinSock 資源的應用程式，下層則對於傳輸的核心協定進行溝通。

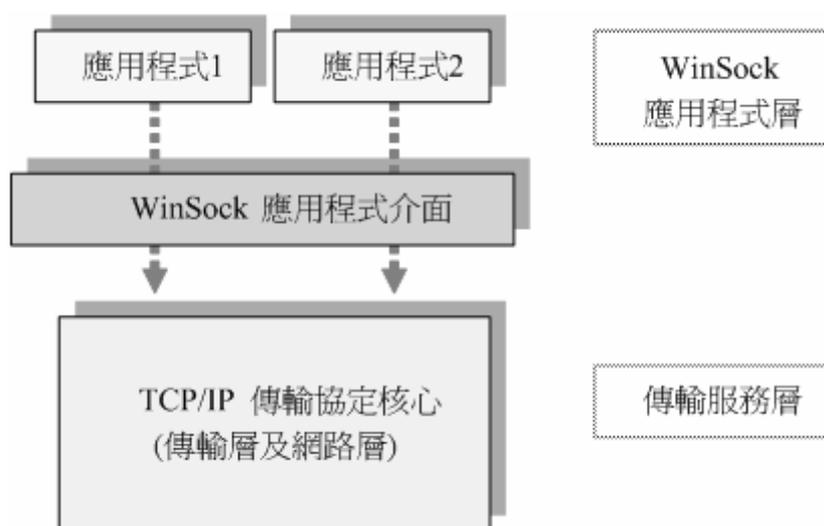


圖2-10 WinSock 應用程式介面

採用的連接導向 (TCP) Server-Client 模式，其執行流程如圖2-11所示。首先呼叫 WSAStartup() 函式向作業系統要求使用 WinSock 的動態函式庫，接著在兩端各開啓一個 socket，socket 可抽象地想像成資料的傳輸口，利用這個傳輸口和遠端的機器互相傳送資料，並指定本地 IP 位址和 port number 給未定名的 socket，等一切設定都指定就緒後，Server 端進行 listen() 等待 Client 端進行 connect() 連結，接受後雙方即可互相傳輸 socket，Server 端接收 socket 後，

將所收到的資料進行計算與處理，再透過 socket 將資料傳回給 Client 端，Client 端將接收到的資料再次進行計算與處理，如此即完成一次完整的傳送與接收迴圈，之後按照這樣的迴圈如此反覆下去，直到操作結束，而當要中斷兩端點的連線狀態時，Server 端與 Client 端均需利用 closesocket() 函式關閉 socket，並且呼叫 WSACleanup() 函式註銷使用 WinSock 的動態函式庫。

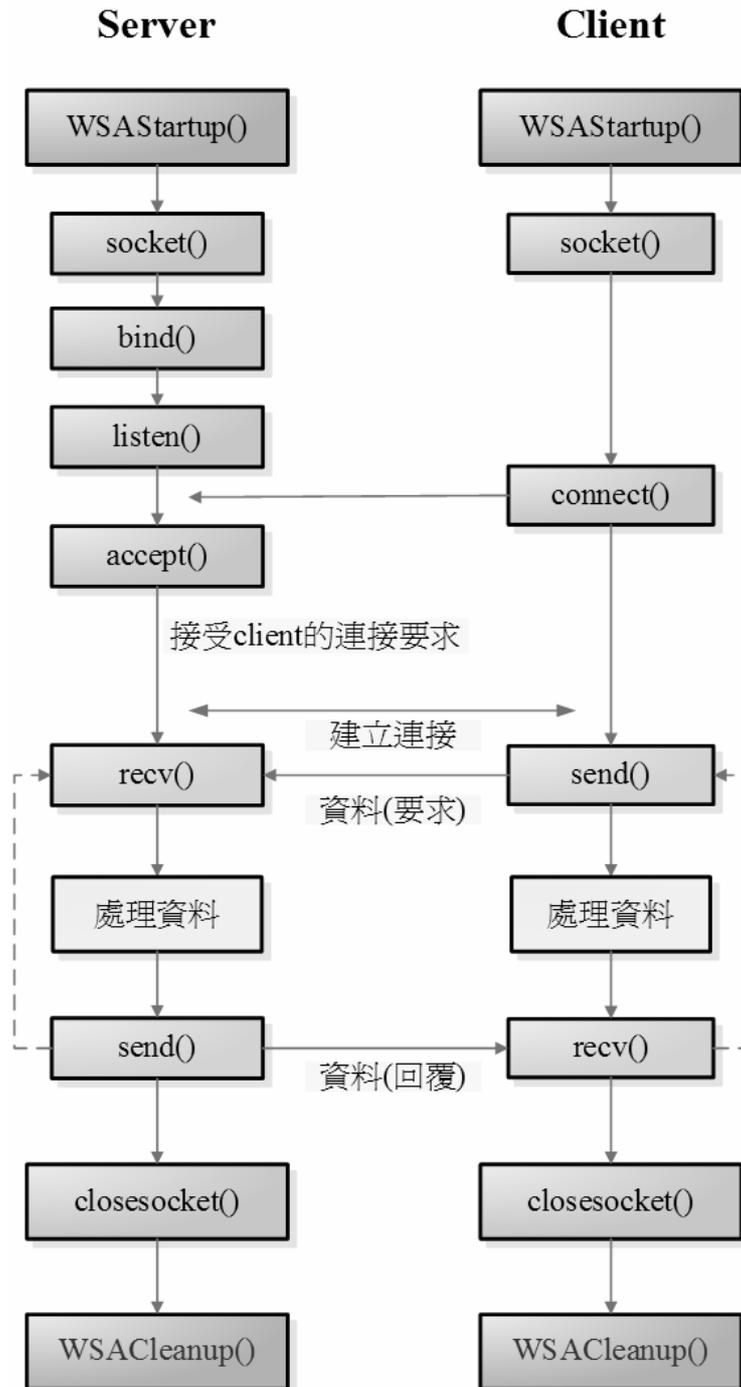


圖2-11 連接導向式 (TCP) 的 Client-Server 程式流程圖

2.4 多執行緒

力的更新頻率與視覺的更新頻率有所不同，而如果將力的更新頻率與視覺的更新頻率調為相同，即將視覺更新頻率拉高到與觸覺相同，則會因圖形處理耗去電腦硬體的大量計算，使得電腦執行效能大幅下降；所以不把電腦的圖形處理和觸覺處理在同一個執行緒裡頭運行，而以多執行緒 (multithread) 的方式進行，將兩者分別獨立運作。

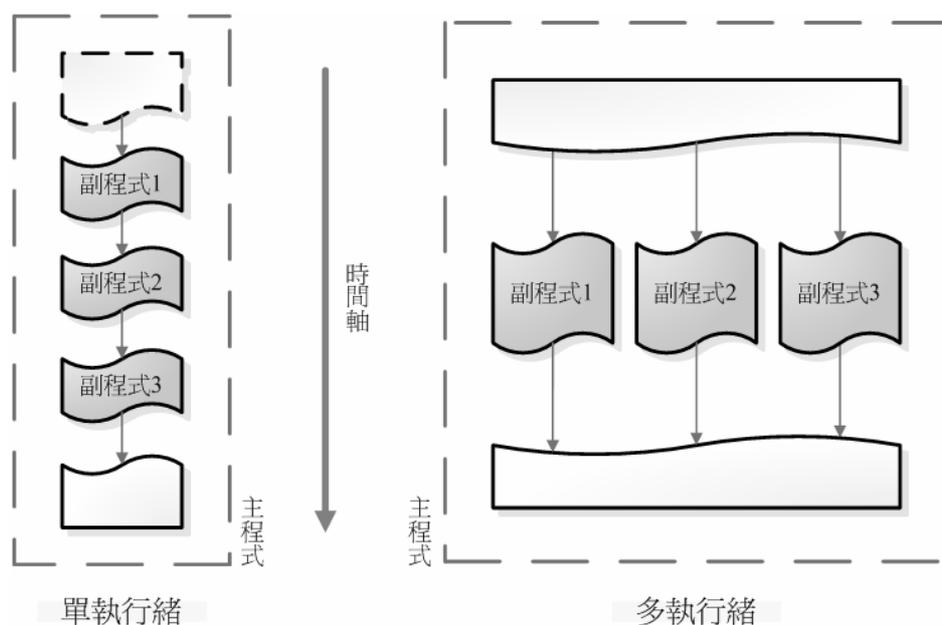


圖2-12 單執行緒與多執行緒的執行順序和時間關係

多執行緒能夠同時提供多個執行緒 (thread) 的功能，通常執行緒所指的是在一個程式中的副程式，多重執行緒將一個程式分割成幾個小模組，每個模組都在一個特定的執行緒當中執行，這就宛如平行處理一般，但是卻利用了多工的能力，表面上使用者並未感覺到多重管線的運作方式，但是實際上程式執行的效率卻提高了。舉例來說，單執行緒就好像大型量販店只有一個結帳出口，如果顧客採買的東西只有一點點，那麼就可以很快速的結帳，但如果採買的東西很多，則結帳的時間就可能很久，而其它欲結帳的顧客就必須等待，而多執行緒就好像有

很多的結帳出口，就算一個出口停頓了很久，也不會影響到其它的出口，如此可大幅的提升運作效益。圖2-12顯示一般程式設計（單執行緒）對照多執行緒的執行順序和時間關係。此外，利用多執行緒的技術來設計程式，不但可提升效率，更可以簡化程式架構，在本論文中，即採用雙執行緒，整個系統的流程如圖2-13所示（在此以 Server 端為例），區分為兩個運作迴圈，分別處理力回饋與視覺呈現，利用此雙執行緒來作為程式內部的運作方式。

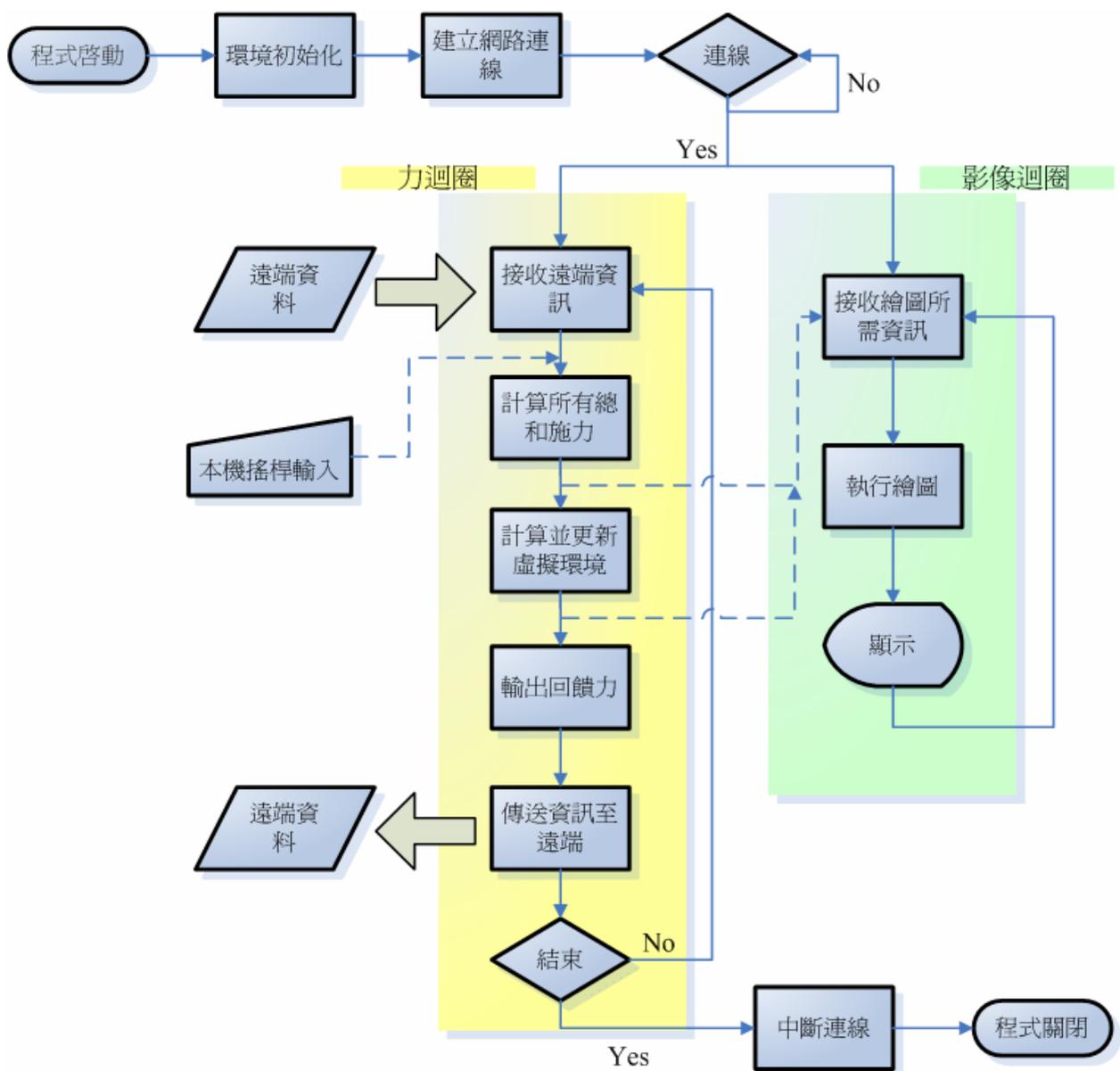


圖2-13 Server 端雙執行緒程式流程圖

第三章

力感復現

在二章所提計劃建構的系統中，有一個不可或缺的環節，亦即操作者用來進行操控的力回饋裝置，本章首先介紹所使用的力回饋裝置，及其與系統溝通方式，接著討論如何使用力回饋裝置以達到力感復現，再決定所要使用的力感復現模型。

3.1 力回饋裝置



計劃建構的力回饋系統其運作的流程如圖 3-1 所示，系統接受來自使用者的施力，經電腦運算後決定送至虛擬場景的運動指令，再不斷地將遠端感受到的受力，經由運算單元得到控制訊號，送至力回饋裝置，再回傳給使用者。

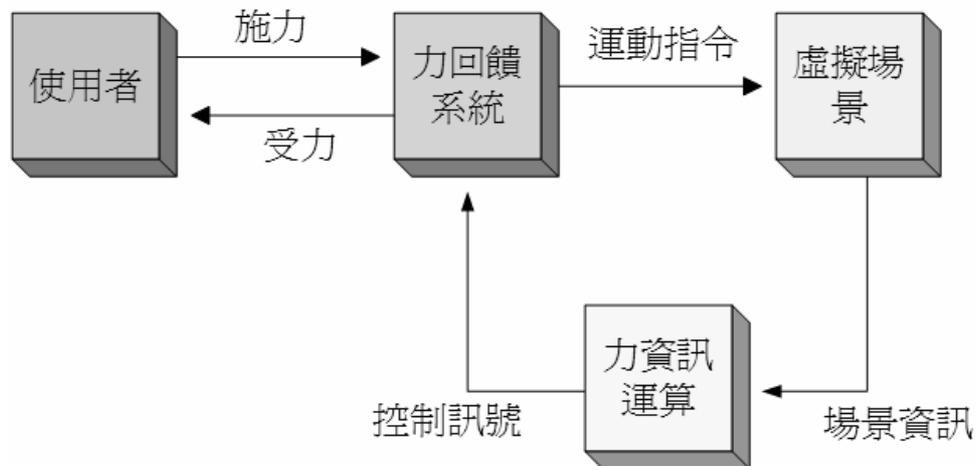


圖 3-1 力感操控系統控制流程

而對於直接和使用者進行互動的力回饋裝置，其基本組成如圖 3-2 所示，在力回饋裝置內有著不斷感測目前資訊的感測元件，還有驅使裝置本體（如搖桿本體）將力量傳回操作者以達致力回饋功能的動力元件。

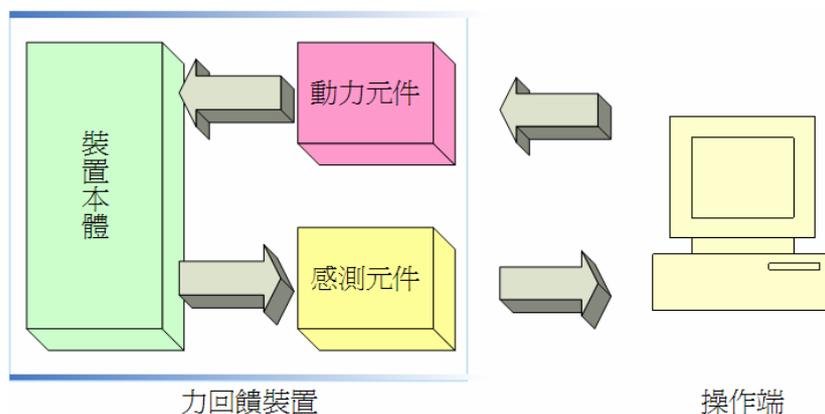


圖 3-2 力回饋裝置示意圖

根據不同的力回饋裝置以及其感測裝置，大致可分為主動式和被動式兩種，兩者皆是當環境有反應力回傳時，會把力感回饋給操作者感受，所以就回饋力感的功能性而言，兩者無異，其最大的不同，是在於施力方面，主動式力回饋裝置隨時偵知操作者肢體的出力情況，而被動式裝置卻僅在與環境的物體接觸後，才會在該環境中產生施力，換句話說，主動式力回饋裝置能主動感知操作者施力大小和方向，就算施力者的施力肢體處於自由運動狀態、或是與環境處於靜止對抗狀態（比如推物體推不動，但其實手還是在輸出推力的情況）皆能知道施力者的施力狀況，而要達到如此的隨時主動探知，其裝置上的力感測元件必定要隨時與施力的肢體保持接觸才行，整體的概念是利用現今發展的肌電訊號直接量測肌肉出力大小，再搭配專致力回饋的力回饋裝置來構成整套系統。而至於被動式的力裝置，相較於主動式裝置的複雜性，大多只含有位置的感測裝置，只能感知裝置的輸出位置，或是移動速度，而不能隨時感知施力者究竟出了多大的力量，也因此僅在操作者接觸遠端物體後，才能決定究竟施了多少力在遠方環境，但如果在被動式的裝置上加上力感測裝置 (force sensor)，也可以量測到施力靜止對抗狀態

時的出力資訊。整體來說，被動式裝置因為建置較為容易，且精確度容易控制，如果再搭配力感測裝置、或是加速度感測裝置就可以滿足大部分的使用需求。而本論文以實驗室現有的兩個被動式力回饋裝置做為發展工具。

使用的兩個裝置分別是 Immersion 公司的 Laparoscopic Impulse Engine (圖 3-3(a)) 與 Impulse Engine 2000 (圖 3-3(b))；其中 Laparoscopic Impulse Engine 是五個自由度的操控器，而 Impulse Engine 2000 則是兩個自由度的操控器。而其硬體規格與一般目前世面上的低價力搖桿的比較表列於表 3-1，由於 Laparoscopic Impulse Engine 的架構與自由度較不同，以不同方式列於表 3-1 之右。



圖 3-3(a) Laparoscopic Impulse Engine

圖 3-3(b) Impulse Engine 2000

| 項 目 | | Impulse Engine 2000 | 市 售 Microsoft & Logitech | Laparoscopic Impulse Engine |
|----------------------|-----|----------------------------------|--------------------------------|--|
| 活動範圍 (θ) | X 軸 | 65 deg | 45 ~ 55 deg | 自由度 5 (可動部份) 3 (力回饋部分) |
| | Y 軸 | 65 deg | 35 ~ 50 deg | |
| 總高度(L3) | | 28 cm | 18 ~ 30 cm | 工作空間 5" x 9" x 9" (12.7 x 22.8 x 22.8cm) |
| 握把長度(L3-L1) | | 18.5 cm | 13 ~ 20 cm | |
| 擺動距離(R) | | 18 cm | 10 ~ 16 cm | 搖桿精確度 0.0009" (1100 dpi) (0.023 mm) |
| 最大額定輸出力矩 | | 8.9 N | not available | 最大額定輸出力矩 2lbs. (8.9 N) |
| 搖桿精確度 | | 0.0009" (1100 dpi) (0.023 mm) | not available | 摩擦力 < 0.5 oz. (0.14N) |
| 電腦連接介面 | | dedicated card | USB、game port | 工作頻寬 650Hz (線性軸) 120 Hz (旋轉軸) |
| 定位方式 | | free | fixed | |
| 工作頻寬 | | 650 Hz | not available | |
| 控制方式 | | PC | μ -processor | |
| 價位 | | 180000 NT | 1000 ~ 5000 NT | |

表 3-1 力回饋搖桿規格比較

3.2 觸覺頻率

在討論如何復現力感前，我們先討論關於人的特性以及力感與觸感兩者的不同，以利在建立力感模擬系統時可以知道力回饋裝置所需要的工作頻寬，如此所復現的力感，感覺才會連續。當人類與物體發生碰觸時，首先遍及皮膚表面的感覺受體會感受到接觸力，藉由神經系統將這感覺資訊傳回給大腦，讓大腦得知碰觸的實際情況後，人腦可對肌肉下達指令以作出適當的反應。而人體感覺器官接收外界刺激的能力，在視覺方面頻率約為 50 Hz [16]，而 [20] 中提到能讓使用者感到互動流暢的 3D 電腦動畫最低標準為 15 fps (frames per second)，電影般的程度約為 24 fps，電腦動畫標準門檻值為 30 fps，但這些值在某些場合還是不夠的，在多數軍用或商用的飛行駕駛模擬應用程式中，要達到真正的流暢需達到 60 fps，一但超過了 60 fps 以上，肉眼已經很難分辨出之間的差異；而人耳朵聽到的頻率約在 20 – 20k Hz 之間 [16]。



人的觸覺 (haptics) 大概可以分為力感 (force) 及觸感 (touch/tactile) 兩個部分，力感部分的感覺器官 (receptor) 較低頻約為 20 – 30 Hz，位在人體內較深處，而力感對力解析度約為 0.5 N [4]；觸感部分的感覺器官較為高頻約 320 Hz [16]，位於接近皮膚表面；一般而言，力感感覺器官處理低頻且高振幅的力資訊，而觸感感覺器官處理高頻寬低振幅的觸感資訊。

頻寬及解析度等參數值除因各文獻量測的對象及方法會有些差異外，還會隨使用者運動的方式不同而有所改變，比如說人的觸感系統在精神集中的情形下，可以感覺得到的振動有時可以高達 1k Hz [7]，甚至可高達 10k Hz [4]，皆遠比先前提及的 320 Hz 要高出很多，很明顯地這些參數會隨應用環境而有所變動，所以在發展觸覺回饋 (haptic feedback) 裝置時，要依其應用場所需的規格來設計，

例如醫生開刀所需的觸感回饋 (touch feedback) 裝置需要高達 1k Hz 才能提供足夠的觸感 [2]，隨操作者的手越快速移動則需要更高的更新頻率 [18]。故在大多數的狀況下，力更新頻率遠較影像的更新頻率 30 – 60 Hz 高出很多，此時若觸覺和影像使用相同的更新頻率，則在復現力感上會有不真實與不連續感。歸納視覺與觸覺所需的更新頻率至表 3-2，依此在建置系統時，參照目前使用的硬體，決定採用 650 Hz 的觸覺更新率與 100 Hz 的圖形更新率。

| 項目 | 電影播放 | 電腦動畫 | 飛行模擬 |
|--------|-------|-------|-------|
| 視覺更新頻率 | 24fps | 30fps | 60fps |

| 項目 | 力感 | 觸感 |
|--------|------|-----------|
| 觸覺更新頻率 | 30Hz | 1k~10k Hz |

表 3-2 系統模擬時視覺/觸覺所需相關參數

3.3 力感復現

對於力感復現，應該先了解搖桿的系統模型，才能知道如何決定搖桿的施力大小，而介紹完搖桿系統模型後，接著要討論決定在模擬環境裡，要選擇何種對虛擬物件產生施力與反應力的模型，區別這些模型間的不同。

3.3.1 搖桿系統模型

操作者所感受到的力量主要是靠力回饋搖桿馬達輸出的機械功率所賦予，不同的力覺感受係由不同的力量大小、方向與頻率等因素組合而成，當使用者操縱

搖桿時，施予搖桿力量的手會與搖桿產生互動，其中包含位置、速度、加速度與力量彼此間的動態關係，可以以阻抗來表示，Hogan 在 1987 年提出阻抗控制理論 (Impedance Control Theory) [11,12]，用來處理機器手臂碰觸物體表面所發生的狀況，我們以此為基礎，將其概念應用在力回饋搖桿與遠端環境兩者雙向之間的互動力感關係上。

首先定義在操作者所處的真实世界中，搖桿系統本身的運動性質，令系統本身的彈性係數為 k ，阻尼係數為 b ，質量係數為 m ，搖桿的位置為 x ，速度為 v ，加速度為 a ，系統的摩擦阻力為 f_{fr} ，使用者施加的力量為 F_h ，馬達出力為 F_m ；接著相對應於真實世界的搖桿，要定義用在電腦模擬的虛擬世界中，代表虛擬搖桿的物理性質，令模擬物件的彈性係數為 K ，阻尼係數為 B ，質量係數為 M ，搖桿的位置訊號偵測值為 X ，速度為 V ，加速度為 A ，虛擬的摩擦阻力為 F_{fr} ，使用者感受的力量與施加的力量同為 F_h ，根據 [8,9] 中的系統模型，在真實操作端中的搖桿存在以下的阻抗關係式：

$$F_h + F_m = ma + bv + kx + f_{fr} \frac{v}{\|v\|} \quad (3.1)$$

而在虛擬世界裡我們期待能達成類似(3.1)式的虛擬阻抗關係式：

$$F_h = MA + BV + KX + F_{fr} \frac{V}{\|V\|} \quad (3.2)$$

圖 3-4 即是顯示此兩種關係的示意圖，左邊是真實世界操作端搖桿的系統組抗圖，可以看出當施力於搖桿上時，所施的力量與代表搖桿本體的物體質量 m 間會有一彈性彈簧與阻尼模擬接觸時的情形；而在電腦端運行的虛擬世界中，即圖右邊，也有相當的關係存在於施力者和虛擬端的搖桿質量之間。假設量測到的物理量與實際值相等，即： $x = X, v = V, a = A$ ，結合 (3.1) 和 (3.2)

兩式，可得到馬達輸出力的方程式：

$$F_m = (m - M)a + (b - B)v + (k - K)x + f_{fr} \frac{v}{\|v\|} - F_{fr} \frac{V}{\|V\|} \quad (3.3)$$

其中 m 、 b 、 k 和 f_{fr} 為系統常數，理論上先輸入需要模擬的虛擬質量 M ，虛擬阻尼係數 B ，虛擬彈性係數 K ，以及虛擬摩擦力 F_{fr} ，只要能夠量測到位置 x ，速度 v ，以及加速度 a ，就可藉由不同的阻抗元件及其組合，調變馬達的輸出力 F_m ，模擬各種物理元件。

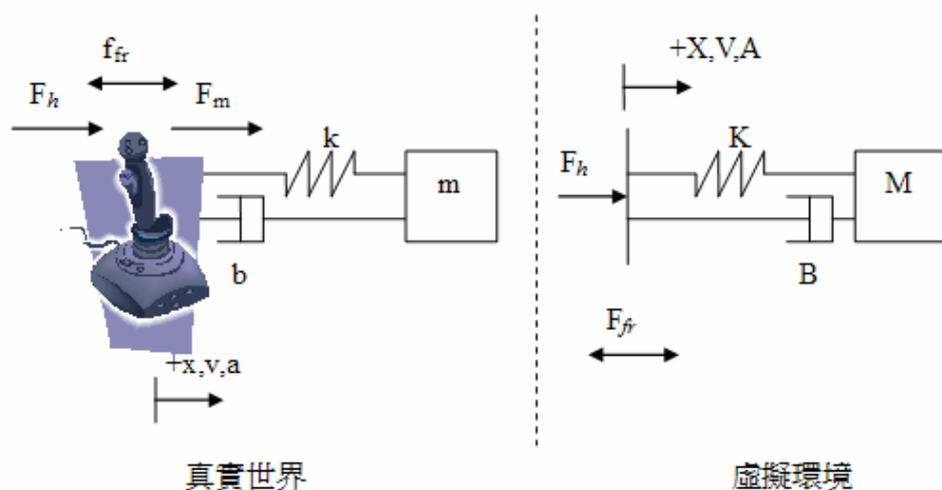


圖 3-4 系統阻抗圖：真實世界和虛擬環境

另外，若加速度不易量得（需要高精度的加速規），也可利用加裝在搖桿上的力感測器，直接量得的搖桿與手之間的作用力 F_h ，則馬達出力會有如下關係式：

$$F_m = \left\{ \left(\frac{m}{M} - 1 \right) F_h + bv + kx + f_{fr} \frac{v}{\|v\|} \right\} - \left(\frac{m}{M} \right) \left(Bv + Kx + F_{fr} \frac{v}{\|v\|} \right) \quad (3.4)$$

若是在模擬中，欲直接以真實端之搖桿去感受模擬物體的特性，可將模擬物件的質量直接設定成搖桿系統的自然等效質量 ($M=m$)，代入 (3.4) 式：

$$F_m = \left\{ bv+kx+f_{fr} \frac{v}{\|v\|} \right\} - \left\{ Bv+Kx+F_{fr} \frac{v}{\|v\|} \right\} \quad (3.5)$$

則力感測值並不會影響馬達的力量輸出，只單純的分成真實世界的物理力量與虛擬世界的虛擬力需求的差值。值得一提的是，假設把 (3.5) 式後半部設為零，相當於希望搖桿模擬處在一完全不受力的虛擬環境之情況，使用者將感受到搖桿平順無阻力的移動，此時馬達需額外出力以克服搖桿本身阻力，其出力為

$$F_m = \left\{ bv+kx+f_{fr} \frac{v}{\|v\|} \right\} \quad (3.6)$$

但因為在此我們所使用的力回饋裝置，其摩擦力並不高，而阻尼也不明顯，所以可以忽略掉 (3.5) 式前半部，即只要專注於 (3.5) 式後半部即可模擬出接觸的物理元件性質。

3.3.2 對虛擬物體的力感復現模型

爲了實現力回饋系統中的模擬環境，讓操作者可以充分感受，像是搬動遠端物體、或是撞到牆面會有的反應力，所以要建立遠端虛擬物件合宜的力感模型。在遠端物件端產生施力與力回饋的概念如圖 3-5 所示，爲了產生施力，所以搖桿位置實際上要如圖一樣進入接觸物體內，而就接觸面、接觸位置、搖桿進入後的實際位置三者的關係，可以決定出如何對物體施力，以及決定施力方向的力感模型，而對物體的施力與物體對施力者的反應力應該相反，如此才符合直覺感受。

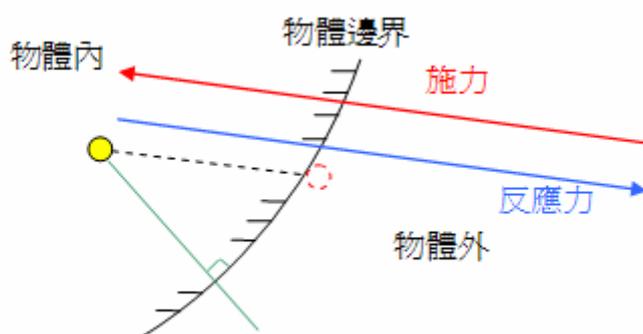


圖 3-5 物體受力與反應力方向

在開始討論力感模型前，先討論所希望感受到的力覺反應。以位於力回饋裝置端的操作者來說，很自然地、會希望在遠端的模擬物體未被推動時，可以沿著物體的表面，像是用摸的感覺，感受出其整個輪廓。換言之，如果有一個接觸點在虛擬的環境中代表操作者的指尖，此接觸點要能沿著物體表面移動，而在想要對物體施力時則可以直接朝表面、向物體內面推去，就像是對一個牆面施壓一樣的方式去推所欲推動之物體。接著開始以接觸面、接觸位置、以及搖桿進入後的實際位置，三者的關係來討論力感模型，而使用 **impedance control** 可以得的結果和第二種模型很類似，但考量物體表面幾何形狀可能較爲複雜，我們還是以三種

模型做為基礎的選擇對象，並加入時間 t_1 、 t_2 以方便解說，其中 t_1 是搖桿穿透入虛擬物表面後的第一瞬間，而 t_2 則是 t_1 的下一瞬時，在圖 3-7、圖 3-8、與圖 3-9 中的虛線，紅色虛線是接觸點沿著物體表面的移動軌跡，綠色虛線則是實際搖桿位置的移動軌跡，三種模型分述如下。

(1) 以距接觸面切線距離為主的力感模型

此一模型可用圖 3-6 解說，假設在時間 t_1 ，搖桿穿透後，以「實際移動到的位置距當時接觸面切線的距離」，即圖中的灰色字所標示之距離，為計算施力的依據，如此所計算出的瞬間出力與反應力，儘管可以吻合 t_1 瞬間接觸位置對物體表面的施力情況，然而對依 t_1 時搖桿實際位置所算出的下一瞬間 t_2 的接觸位置而言，會直覺受到的表面反應力並不吻合，但如果將此模型應用在「和物體接觸後，接觸位置會停留住而不沿表面移動」如此情形下，此方法非常合適。

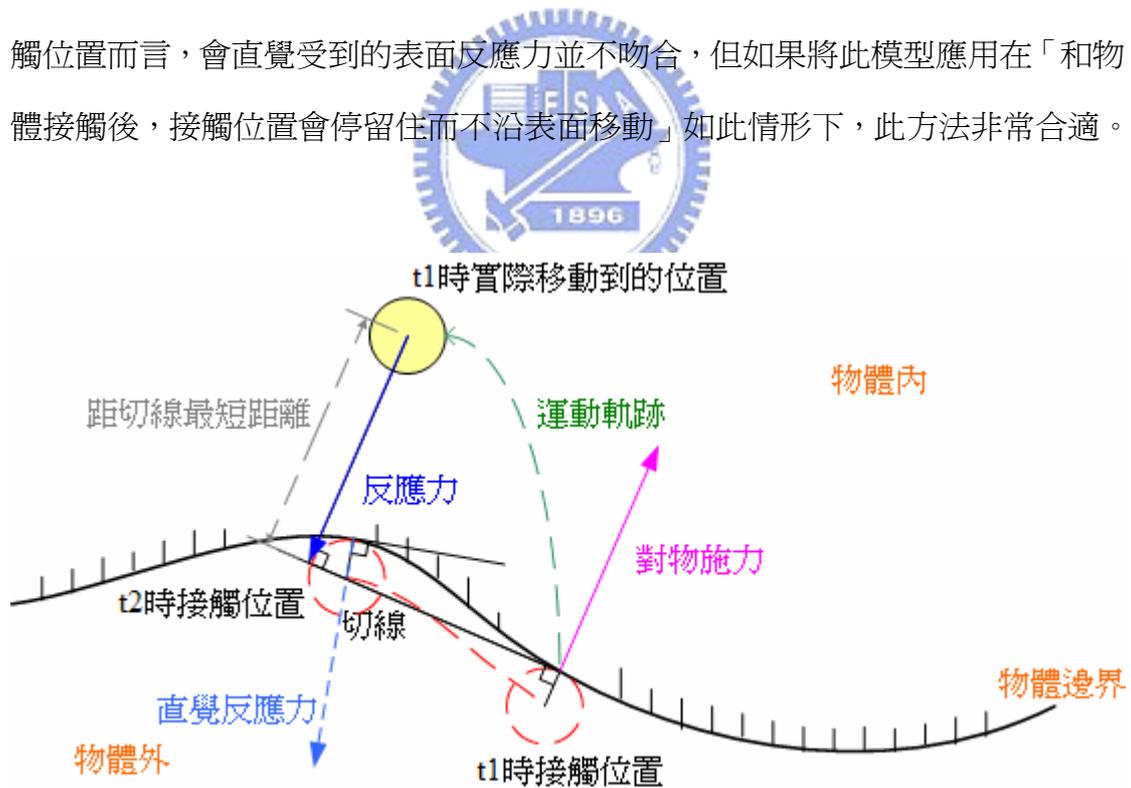


圖 3-6 以距接觸面切線距離為主的力感模型

(2) 以距最近表面距離為主的力感模型

此模型如圖 3-7，不論何時，以當時搖桿實際位置距最近表面的最短距離來決定對物體的施力作用，其產生的力是不斷地把搖桿往最近表面推去，直到推出物體外，如此的方式與圖 3-7 的方式相比，可發現將可以符合 t_1 下一瞬間、 t_2 時物體表面對搖桿的直覺反應力方向，因為隨時皆吻合最新的位置接觸位置與物體表面關係，其對於觸摸表面的觸覺和視覺位置聯想最為直覺。

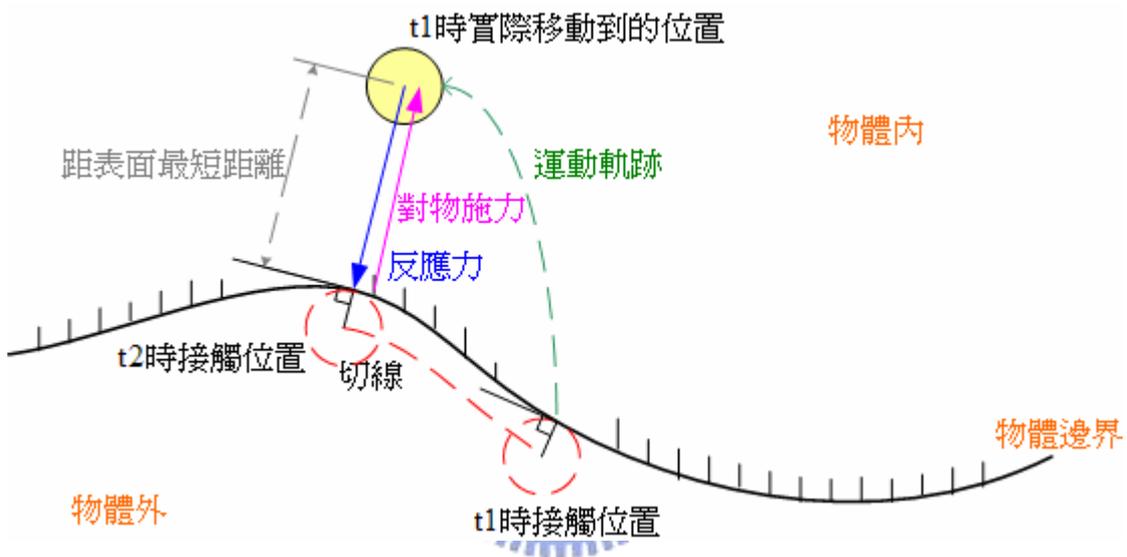


圖 3-7 以距最近表面距離為主的力感模型

(3) 以實際位置和接觸位置距離為主的力感模型

如圖 3-8 所示，第三種方式為不管何種狀況，隨時都以瞬時搖桿實際位置和瞬時接觸位置的距離差為計算施力的依據，如此會使得不管搖桿位置在哪只要在物體內，都朝著 t_1 瞬時的接觸位置移動，這樣的方式會使得在對物體的施力過程變得不直覺，因為不論是在 t_1 瞬間或 t_2 瞬間，對新、舊接觸位置來說，物體表面的直覺反應力和計算出的施力方向，並不一定相同，但此類方式卻適合用於決定在用夾子夾住物體後、所要移動的施力方向與施力大小。

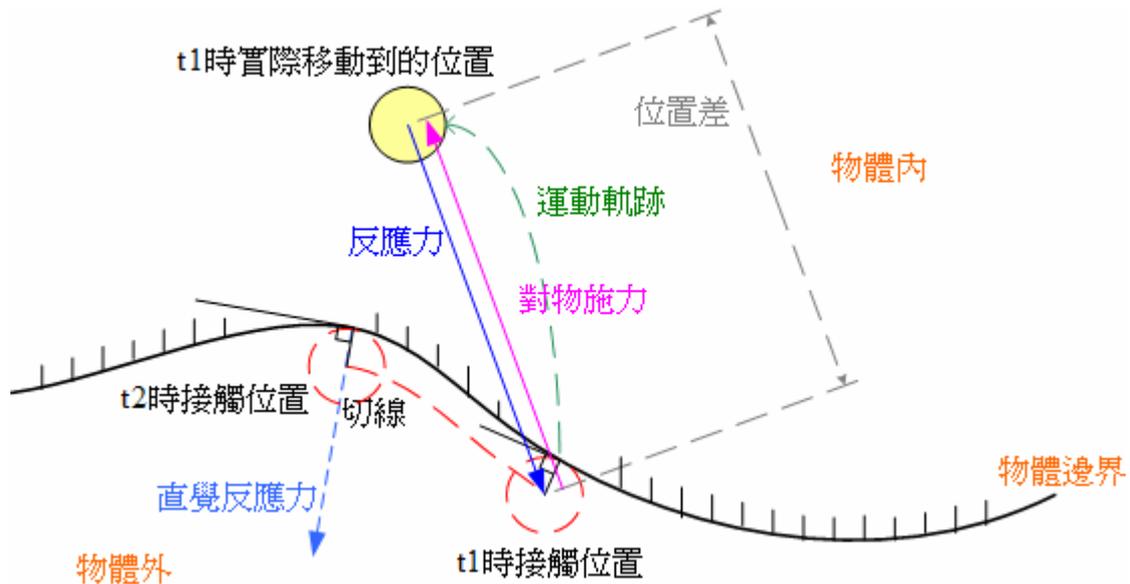


圖 3-8 以實際位置和接觸位置距離為主的力感模型

在上述三種方式裡，本文對於發展的虛擬物採用的是第 (2) 種力感模型，圖 3-7 所示。因為系統整體的運作順序，於產生接觸後，依搖桿目前所在位置會同時算出新的接觸位置和對物體的施力，而當使用者以視覺看到新的接觸位置時，將會直覺的認定新位置與物體表面的反應力，才是正確的力覺感受。在此所選定的此一模型，遇當 A、B 兩人置於被搬物旁，若 A 不動，B 推欲搬動物體，一旦物體被 B 推動的瞬間，電腦算出物體的新表面因為會涵蓋 A 的原來位置，換句話說 A 的實際位置已經侵入了物體表面內，所以 A 端會立刻復現應有的感受力和對物體的施力。而如果 B 推物的力量未超過靜摩擦力，則物體本身就不會被推動，而另一端的 A 搖桿位置自然也不會侵入物體表面，所以 A 此時是不會感受到力量的。

第四章

合作搬運應用

根據第二章所建構的力回饋系統以及第三章中的力感復現，在本章我們將進行系統的實際應用。由於所建構的系統著重的是力資訊的溝通，合適的應用可以是力對抗或是力合作等，而在多操作者的情況下，最常遇到的應用之一就是合作搬運，這就像是日常生活中常會遇到的狀況，當發現一個人的力量不足時會很自然的需要其他人給予支援，而多人合作的方式也很多，像是一同控制一個機器人，有的人負責控制行走姿態，有的人負責上半身的手部行為。在此，我們將重點放在討論各別使用者對單一物體，同時進行推力使其移動的合作搬運應用，如圖4-1所示，因為這是最基本的合力搬運應用型態，可能是兩個人一起努力推動物體，或是一個人用了兩隻手用力推動物體。在此章將詳細討論，在同時施力狀況下，物體依力感模形產生運動的過程，並描述此過程中會遇到的合力搬運問題，針對此合力應用搬運議題，我們也將所發展系統進行因應調整，並增添些許功能，使其在進行此應用上，更加地輕鬆容易。



圖 4-1 合作搬運示意圖

4.1 合作搬運過程解析

圖 4-2 顯示兩個搖桿在原架構系統中，進行合力搬運時的情形，此圖中假定 A、B 從虛擬臂位置移至搖桿實際位置時是沿垂直表面的方向行進，圖中 \vec{F}_a 、 \vec{F}_b 分別是 A 和 B 的對物體施力， \vec{F}_{ar} 是物體對 A 搖桿的反應力、 \vec{F}_{br} 是物體對 B 搖桿的反應力， \vec{F}_t 是 \vec{F}_a 與 \vec{F}_b 的總合力，而其餘的標示可以以數學關係式表示：

$$\vec{F}_{ac} = \vec{F}_a \cdot \vec{P}_{ab} \cdot \vec{P}_{ab} / \|\vec{P}_{ab}\| \quad (4.1)$$

$$\vec{F}_{bc} = \vec{F}_b \cdot \vec{P}_{ab} \cdot \vec{P}_{ab} / \|\vec{P}_{ab}\| \quad (4.2)$$

$$\vec{F}_{as} = \vec{F}_a - \vec{F}_{ac} \quad (4.3)$$

$$\vec{F}_{bs} = \vec{F}_b - \vec{F}_{bc} \quad (4.4)$$

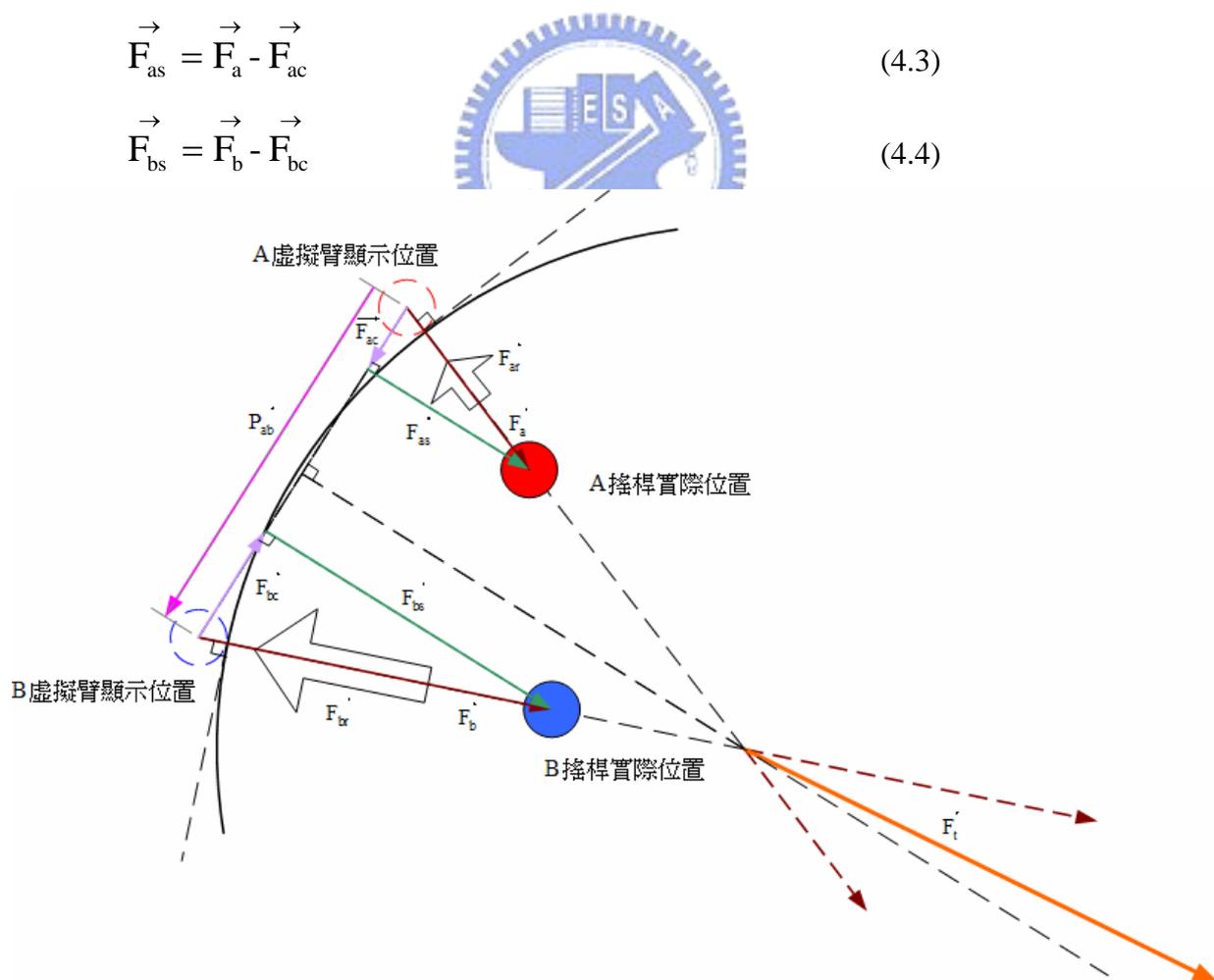


圖 4-2 A、B 同時對物體施力解析

由圖 4-2 中可知，只要 A、B 同時對物體施力，則物體受到的是兩者的合力 \vec{F}_t ，而物體也會依此受力運動，但實作上發現了一個問題，一旦合力推動了物體後，因為物體運動後新位置與舊位置的不同，新位置產生的力 \vec{F}_a 、 \vec{F}_b 與舊位置產生的力落差很容易導致施力者偏差了原施力方向，而產生對物體表面的「滑動現象」。造成滑動現象的原因是因為只要侵入物體表面，則依力感模型即會產生一抵抗力，然而因個別操作者原來施力時必定是以較大的力量推動物體，就算物體移動後瞬間的接觸面變淺，操作者仍將以較大的力量慣性朝原施力方向施力，然而個別操作者的施力方向因與合力方向的不同，所以瞬間物體的運動方向必與個別操作者的施力方向有所不同，圖 4-2 簡單顯示此概念，再加上因為依據力感模型，搖桿對新物體表面位置進入產生的微小力量，很容易就把原施力慣量朝旁邊推去，此部份可以從圖 4-3 看出，藍色是前一刻的慣性施力，綠色是新表面對當時實際位置的反應力，而橘色就是最後產生的合力，可看出合力後的橘色力確實是傾向滑過新物體的表面。

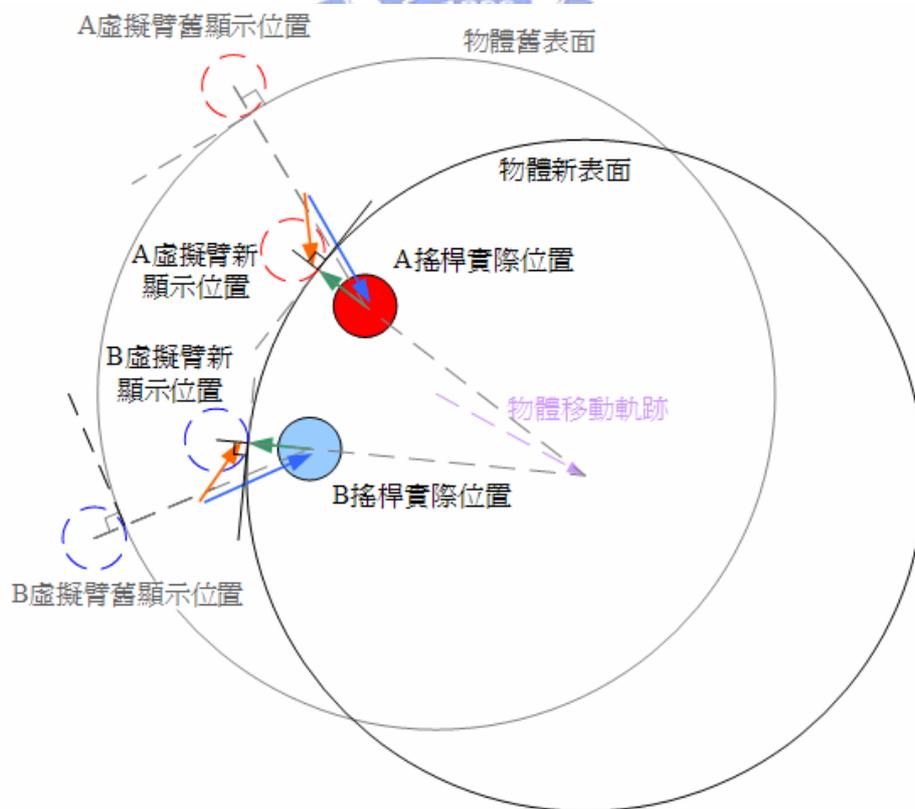


圖 4-3 造成滑動的示意圖

4.2 合作搬運應用之系統調整

造成施力容易滑動主要原因在於合力後物體的運動方向與運動後施力方向的不同所致，所以如果讓施力時的方向與瞬間合力後的方向相同的話，則推動物體後，就應該仍可以保持施力方向的穩定。依此方式，我們新設計一合力系統架構，當有遇到同時施力的狀況，在該瞬間系統只保留 \vec{F}_{as} 和 \vec{F}_{bs} 對質心的施力，即將兩者原施力 \vec{F}_a 和 \vec{F}_b 對位置差向量 \vec{P}_{ab} 的投影去除，則可以保證新得到的合力方向必定與施力 \vec{F}_{as} 、 \vec{F}_{bs} 相同，而如此推動後的物體也必朝此方向前進，就不會滑動，可以很穩定地施力在物體上，在圖 4-4 中可以很容易的看出此成果，因為物體產生的運動方向與物體移動後對搖桿新產生的反應力 (\vec{F}_a' 和 \vec{F}_b') 方向在同一個軸上，所以不會有滑動的情形。



此外，因為我們省略了 \vec{F}_{ac} 和 \vec{F}_{bc} ，所以新的系統架構合力與原系統略有落差，此一情形在當 A、B 距離很接近時並不明顯，但一旦 A、B 距離拉開後其差值會比較大，然而新的合力方式其造成的物體運動方向對操作者而言，仍是相當直覺地，故操作上不太能區分其差異。但考量有可能 A、B 在一瞬間某個人不想朝目前方向推進物體，而想要朝不同的方向施力時，必須要有一個判斷機制來達成每個人當下的自由意願，因此需要對其限制，只在滿足某些條件下，才可進行提出的合力機制。而要滿足的兩個條件：(1) 要圖 4-2 中的 \vec{F}_{ac} 和 \vec{F}_{bc} 反向，才使用合力機制，因為反向代表 A、B 兩者間有相互作用的力存在。(2) 瞬間的 \vec{F}_{as} 和 \vec{F}_{bs} 要分別大於或等於 \vec{F}_{ac} 和 \vec{F}_{bc} ，此代表有共同作用方向的傾向，所以還要吻合此狀況，才判定要使用合力作用之。

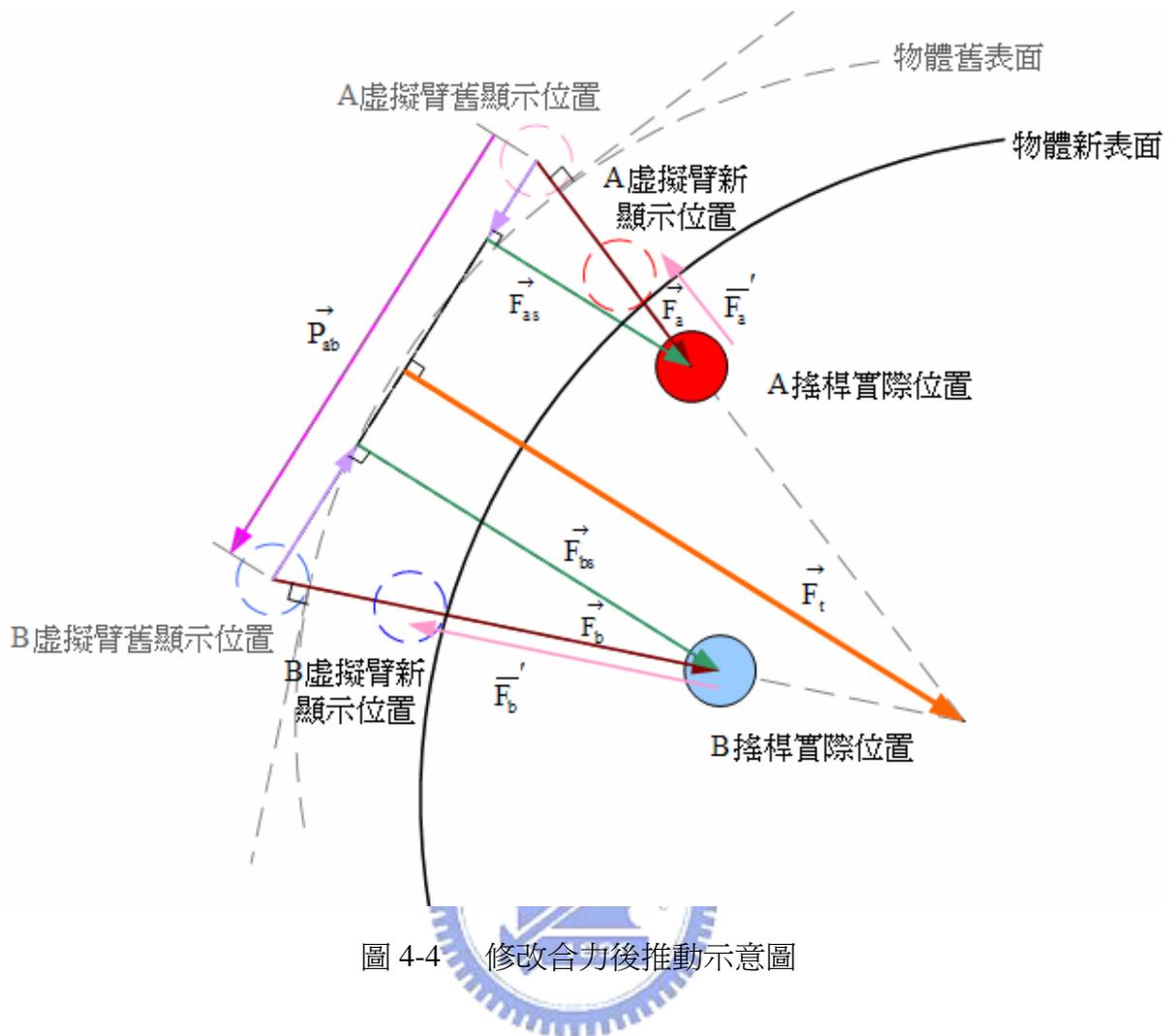


圖 4-4 修改合力後推動示意圖

對於多加入其他操作者的情況，也可以同樣的概念推導出如何合力而不容易滑動。只要把圖 4-2 稍作修改，改為如圖 4-5，圖中新增一個平均位置 P ，整個數學關係式修改如下：

$$P = (P_1 + P_2 + P_3 + P_4 + \dots + P_n) / n \quad (4.5)$$

$$\vec{P}_a = P - P_1, \vec{P}_b = P - P_2, \dots, \vec{P}_n = P - P_n \quad (4.6)$$

再將 (4.1) 式和 (4.3) 式合併修改後可得，

$$\vec{F}_{as} = \vec{F}_a - \vec{F}_a \cdot \vec{P}_a \cdot \vec{P}_a / \|\vec{P}_a\|, \dots, \vec{F}_{ns} = \vec{F}_n - \vec{F}_n \cdot \vec{P}_n \cdot \vec{P}_n / \|\vec{P}_n\| \quad (4.7)$$

如此可以決定出多人時物體質心應受到的合力 \vec{F}_t ：

$$\vec{F}_t = \vec{F}_{as} + \vec{F}_{bs} + \vec{F}_{cs} + \dots + \vec{F}_{ns} \quad (4.8)$$

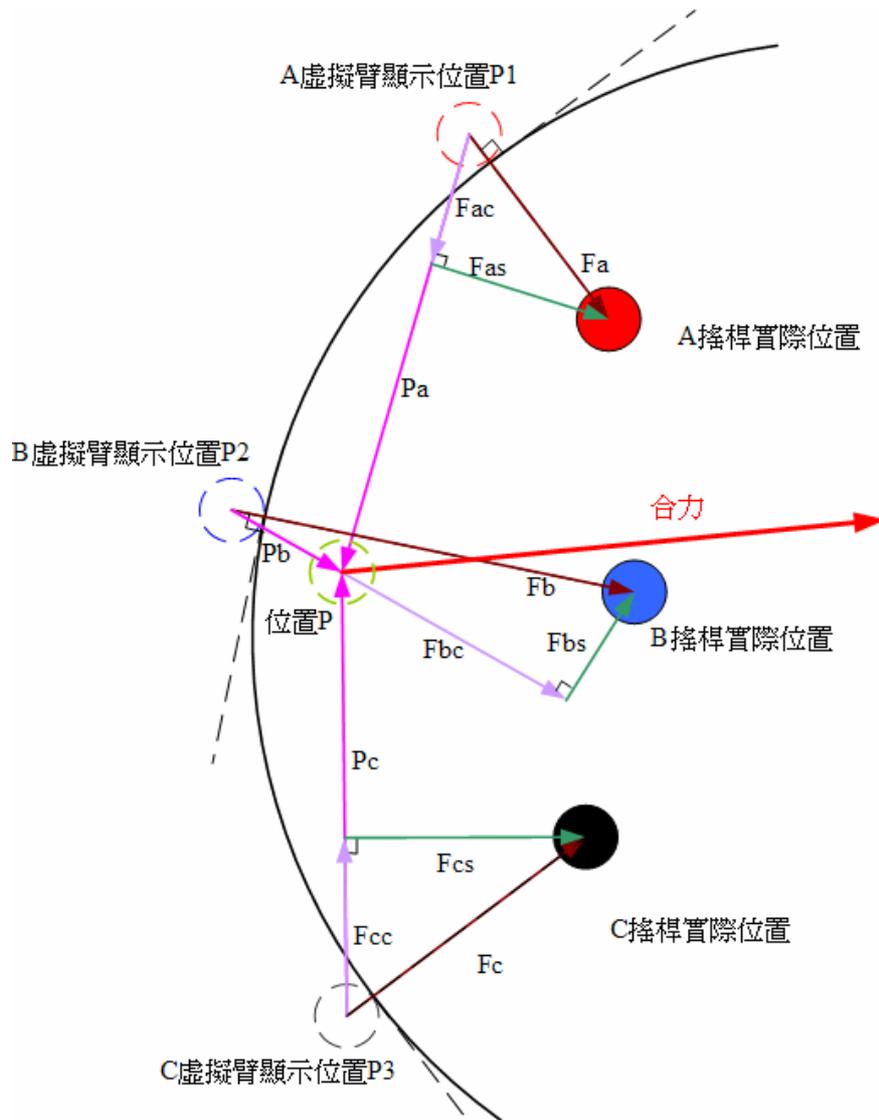


圖 4-5 多人時的合力方式

此外，判斷是否合力的機制也稍作修改，但其基本概念仍然相同：(1) 檢查 \vec{F}_{ac} 是否和 \vec{P}_a 同向、 \vec{F}_{bc} 是否和 \vec{P}_b 同向，同向的才會列入合力機制裡，因為那才代表有對彼此進行力作用的傾向，而例如：若是 \vec{F}_{cc} 與 \vec{P}_c 反向，則 \vec{F}_c 將不加入合作施力的機制裡。(2) 第二個條件是個別施力的投影向量還要大於對角投影向量，即 $\vec{F}_{as} > \vec{F}_{ac}$ 、 $\vec{F}_{bs} > \vec{F}_{bc}$ 、 $\vec{F}_{cs} > \vec{F}_{cc}$ 、...等，如此才使用合力機制。

4.3 調整後的系統實踐

根據 4.2 節所述，依據新的合力方式，對建構的系統進行一簡單的調整，將資訊流進行更多的運算，即先計算出接觸點的平均幾何位置後，再分別對每個接觸點求位置差向量，接著將各自的施力對個別的位置差向量求對角投影，如此個別所得的投影力總合即形成合力，因此可將系統的架構調整如圖4-6，在圖中顯示了資訊流通的方向與進行的運算，新增的資訊流則分別是： \vec{e}_c 是 Client 端虛擬臂位置對平均虛擬臂位置的向量差， \vec{e}_s 是 Server 端虛擬臂位置對平均虛擬臂位置的向量差， \vec{f}_{RSe} 表示由虛擬環境回傳欲 Server 端操作者感受的回饋力對 \vec{e} 的投影向量， \vec{f}_{RCe} 表示由虛擬環境回傳欲 Client 端操作者感受的回饋力對 \vec{e} 的投影向量，另因為只有兩個操作者，所以 $n = 2$ 。資訊流的下標大 R 即表示是回饋力，而新增的 \vec{e}_c 、 \vec{e}_s 皆是為了能計算出新的合力而存在。上述可以下面的數學形式來表示：

$$\vec{e}_c = \left(\left(\frac{\vec{x}_s' + \vec{x}_c'}{n} \right) - \vec{x}_c' \right) \quad (4.9)$$

$$\vec{e}_s = \left(\left(\frac{\vec{x}_s' + \vec{x}_c'}{n} \right) - \vec{x}_s' \right) \quad (4.10)$$

$$\vec{f}_{RSe} = \left(\frac{\vec{f}_{RS} \cdot \vec{e}}{|\vec{e}|^2} \right) \cdot \vec{e} \quad (4.11)$$

$$\vec{f}_{RCe} = \left(\frac{\vec{f}_{RC} \cdot \vec{e}}{|\vec{e}|^2} \right) \cdot \vec{e} \quad (4.12)$$

而在圖 4-6 中，黃色框起來的部份，即是代表一個 Client 端，當有 n 個 Client 端時，整個系統架構就要有 n 個黃色框部分。

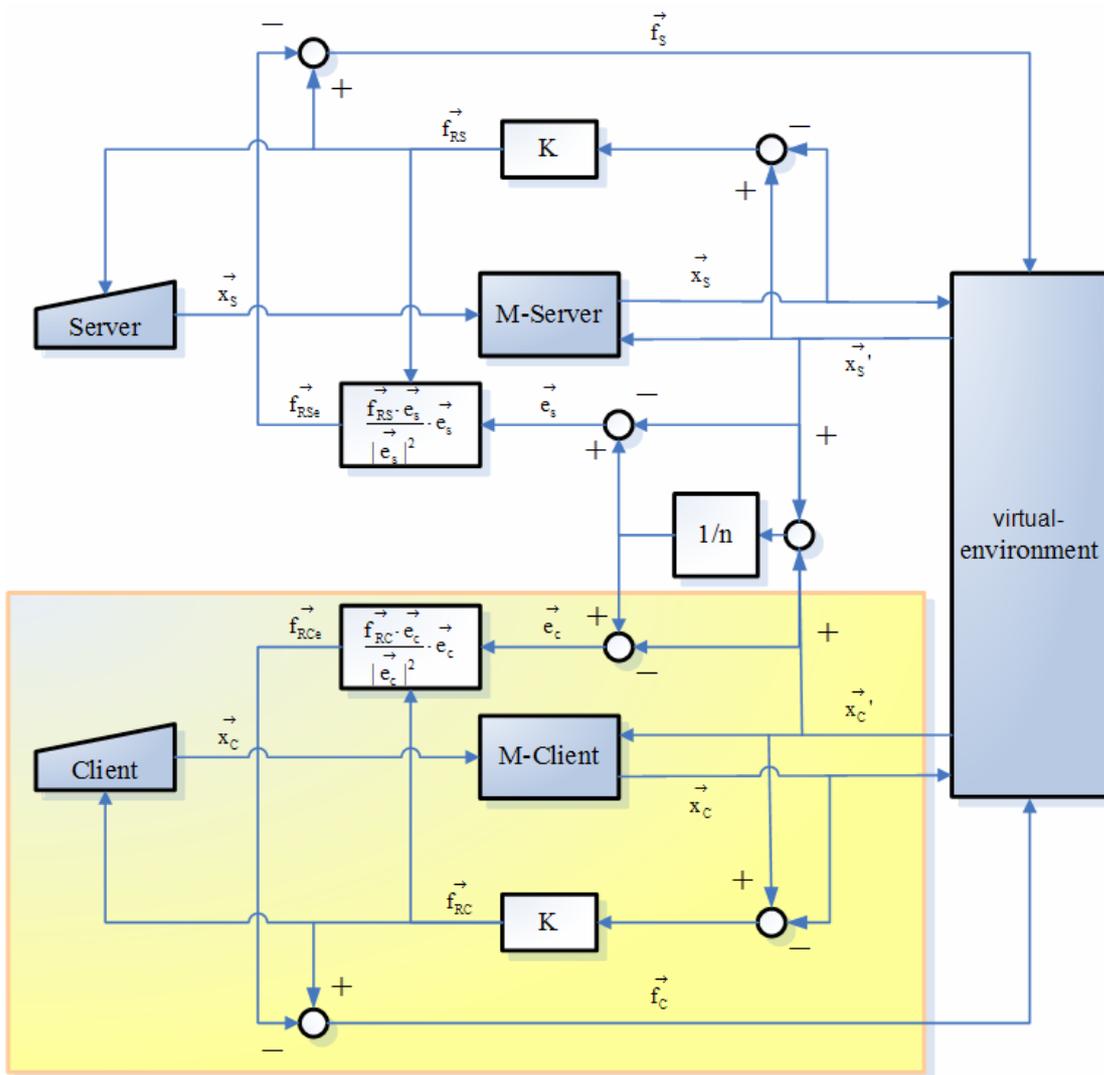


圖4-6 針對合力搬運因應調整的系統資訊流通架構

第五章

實驗

我們針對建構的力回饋系統進行測試，以了解其對於力合作與力對抗的資訊是否能執行無誤，所以要透過網路連結置於不同地方的力回饋裝置，在同一場景中進行對虛擬物件的施力測試，藉此映證力回饋系統的可行性，而對合作搬運應用所作的系統調整，也進行相同的測試，以確認是否提升了合作搬運的效率。本章首先說明實驗的設計方式，以了解實驗的進行，接著對建置的力回饋系統開始進行試驗，列舉實驗結果，然後再進行調整後系統之相關實驗，作為對照。為進一步了解網路延遲對力回饋系統可能造成的影響，我們也進行了實驗室的網路環境與一般家用網路環境的比對，以確實掌握力資訊的流通是否連續無誤。

5.1 實驗設計

實驗進行的環境如圖 5-1 所示，以一台電腦連接力回饋裝置 Laparoscopic Impulse Engine 作為 Client 端，一台電腦連接另一力回饋裝置 Impulse Engine 2000 作為 Server 端，而為使兩力回饋裝置的自由度皆能相同，操作是在同一個 2D 場景中進行，Server 端操作者所操作的虛擬臂以一個小紅圓點來代表，Client 端操作者所操作的則是小藍圓點，兩者合力搬動一個圓形的橘色物體（以 3D 的觀點來看，實際上應為一立於地面的圓柱狀物體），而此一圓形物體表面極為光滑，所以施力將會很容易沿著表面滑動。當操作的虛擬臂接觸到欲搬動物體時，即可以感受到接觸面傳來的反應力，而搬動時也可以確實地感受到搬動時會

傳回抵抗力，整個感受的力量依第三章的力感模型決定。系統中對於模擬環境所設定的數據如下：欲搬動的圓柱重 1.52 N，靜摩擦力係數與動摩擦係數為 3 m/sec² 和 2.5 m/sec²，此外力模型中物體表面產生力的彈力係數設定為 0.2834 N/mm，物體表面是不會產生形變，但根據調整物體彈力係數可以改變產生施力的大小，又物體本身的運動會受到摩擦力的影響，當推其運動時、物體的反應是像我們用手指去推動一個重的圓柱一般，而回傳的力感也是和現實裡一樣；發展的平台是採用非強制性即時多作業系統的 Microsoft Windows XP。另外，在操作過程中，螢幕上代表各操作者的小圓點在彼此互相接觸時並不會有力感作用，但在同時接觸到欲搬動物體時，可以感受到任一個操作者推動物體後，因物體運動而對其他操作者所產生的反應力；如此當有兩個操作者時，只要把兩個虛擬臂置於欲搬動物體的兩側，就可以進行像是比腕力這樣的互動。



圖 5-1 實驗設備與進行環境

在實驗開始前，對於實驗進行中要使用的力搖桿，先簡單地量測其力量值是否依其規格呈線性，我們利用數顆砝碼進行簡單測試。而由於力矩會造成測定上的錯誤，所以擺放測試的位置點亦非常重要，試驗的結果是，將砝碼置於如圖 5-2 中所示位置，也就是接近搖桿最低點時，會與理論的規格數值接近，測試結

果示於圖 5-3。由圖 5-3 可見搖桿的馬達確實是線性出力，而對於微小的誤差、也有可能是量測位置所造成，而且就實際操作的感覺，其實無法明顯地分辨。



圖 5-2 測定搖桿輸出力

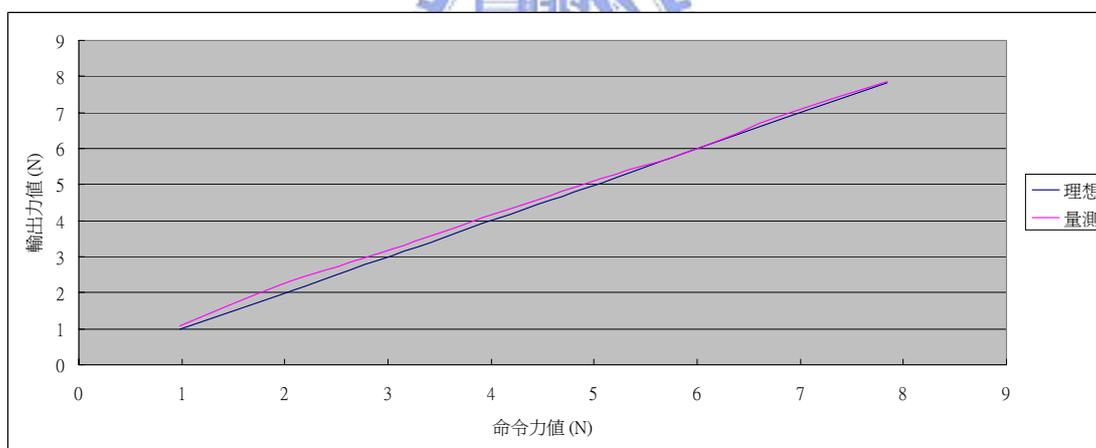


圖 5-3 量測得的輸出力值與理想的輸出力值

在實驗進行方面，首先檢驗對於水平方向與垂直方向搬運的困難度，分別量測偏離預想軌跡的誤差量和實驗達成時間，然後再作一圓形軌跡的的搬運測試，一樣量測偏離預想軌跡的誤差量和實驗達成時間。像是在進行水平向測試時，物體並不會受限於只能水平向運動，而施力者也不會被受限只能對水平向施力，所以是兩端的操作著有先達成共識，了解目前是要依水平向進行運動，也因此，過

程中任一使用者也可能會發生對另一使用者反向施力的情形，其餘的垂直向和圓形軌跡也是一樣的進行法。圖 5-4(a) 是對水平方向的量測，圖 5-4(b) 則是針對垂直方向，如果合力搬運時的範圍超出了所畫的框線就會記錄下偏離框線的距離大小，像水平就是要由框線左邊搬到框線右邊，垂直的則是由下搬到上。雖然進行水平和垂直搬運時，操作者已經是進行二維方向的施力，但所驗證的是對於單一方向性軌跡的準確性，所以最後進行圓形軌跡搬運以作為綜合測試，因為其可以同時驗證水平與垂直向施力方向的準確性，如圖 5-4(c) 所示，又在實驗進行中，我們以像素(pixel)為顯示距離的單位，因為真實世界的比例被轉換到螢幕上，如果依原模擬尺寸則可能會很容易超過畫面的顯示範圍或是看起來太小，在此實驗中，真實比例 0.23 mm 換算為螢幕上的一個 pixel。而在經過實驗後，因為針對此合作搬運應用調整後的系統，完成合力搬運的效率較高，也將以這個版本進行遠距的網路實驗，即不是以單純的區域網路進行實驗，而會移到一般家用網路進行測試。此外，此程式設定的觸覺部分為了達到 650Hz 的取樣頻率，所以執行力感部份是約每隔 1.54ms 進行一次更新。



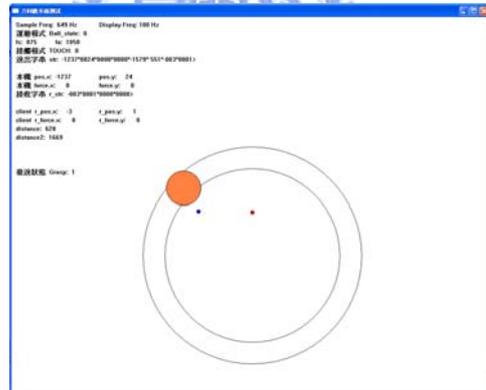
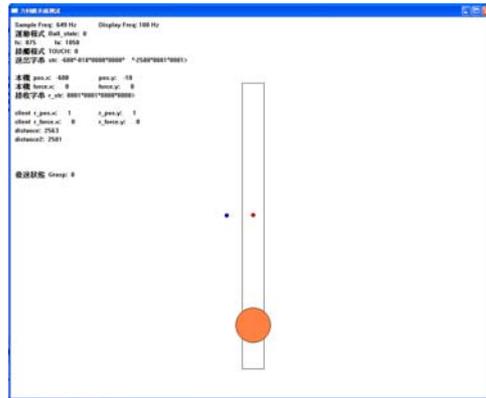
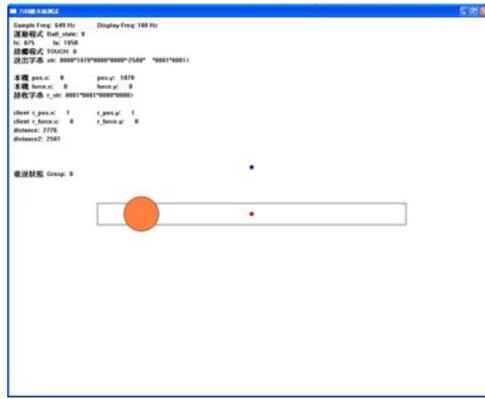


圖 5-4 搬運測試：(a) 水平方向(上)；(b) 垂直方向(中)；(c) 圓形軌跡(下)

5.2 實驗結果

對於預期的實驗結果，不論是對圓形軌跡，還是水平、垂直搬運來說，應是以被搬物體位置偏離軌跡量最小，同時所花費時間最少為最佳，而實驗進行中對各式搬運軌跡皆進行了 10 次取樣，進行搬運測試所得最好的實驗結果，列舉在圖 5-5 ~ 圖 5-7 中，圖中縱軸是代表偏差軌跡的量值，而橫軸則是代表完成此次搬運所花費的時間。此外，對於 10 次的取樣是以如下的方式進行判別取樣結果最佳者，分別對所取樣的資料進行偏移軌跡量的平均值計算，再設計一評估用的誤差圓 (error circle)，此圓半徑以下式決定：

$$(\text{error circle 半徑})^2 = (\text{平均軌跡偏移量})^2 + (\text{完成工作之秒數})^2 \quad (5.1)$$

因半徑越小者，代表花費的時間與偏移量皆最少，可知其表示結果較佳。以此圓評估有一個特點，因為完成時間(秒)對於圓半徑的影響會低於偏移量的像素(每單位 pixel)，所以對於急於把物體推到終點而不去管位置誤差的情形，將會因較大的位置誤差、而得到較大的圓半徑，如此就成了較差的結果，利用此關係，操作者必定會去謹慎控制偏移量。另在實驗進行過程中，可以發現，進行垂直向搬運比水平向要困難很多，應和人手對於伸離身體的動作，較難控制精準度有關。

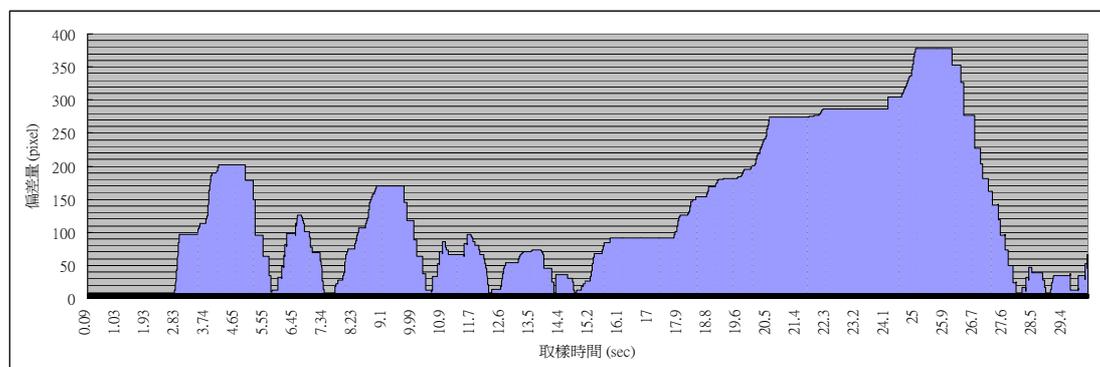


圖 5-5 水平向搬運的誤差與完成時間

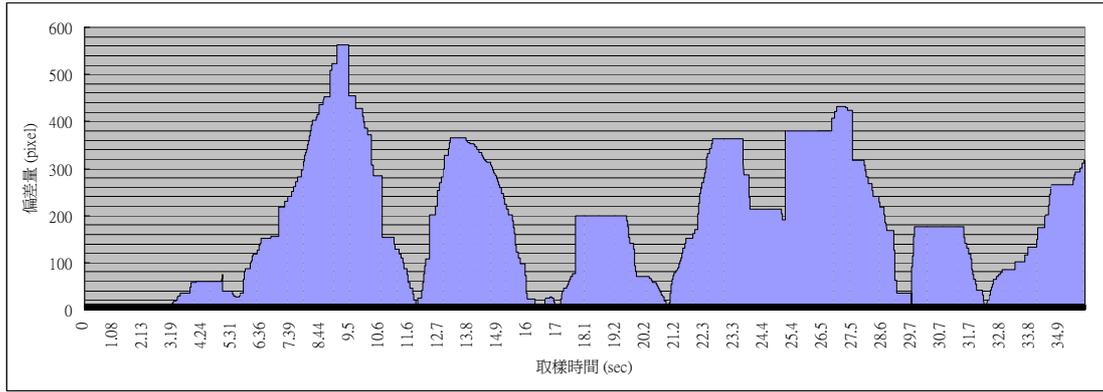


圖 5-6 垂直向搬運的誤差與完成時間

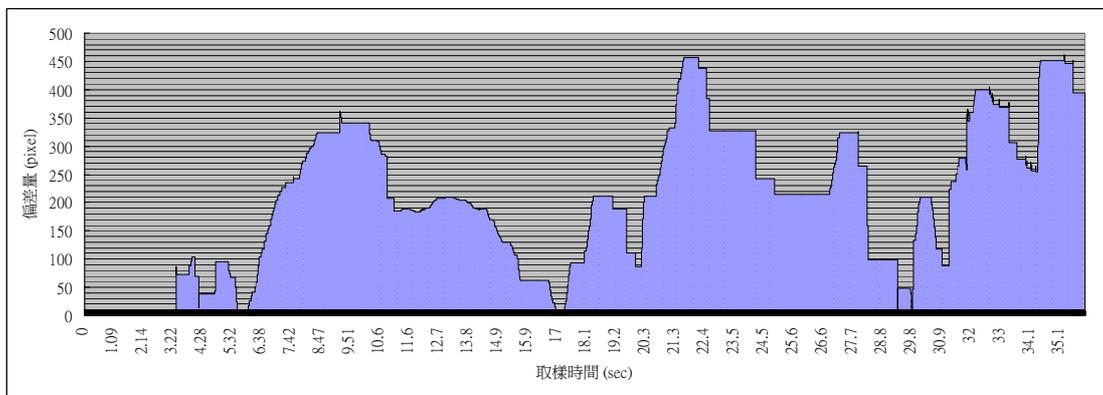


圖 5-7 圓軌跡搬運的誤差與完成時間

此外，針對合作應用調整後的系統，也對其進行 10 次取樣，並列出最好的結果，如圖 5-8~圖 5-10。比對調整前之系統，很明顯地看出，調整後系統的軌跡偏移量較低，且工作完成時間也都更短。

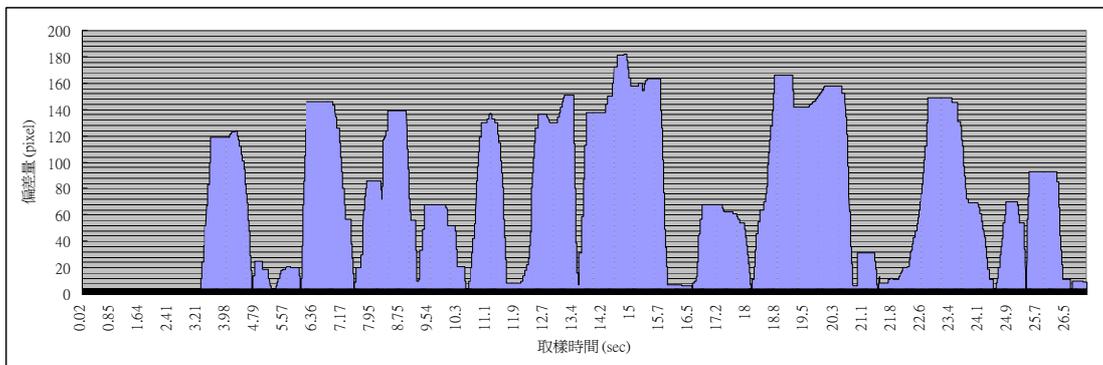


圖 5-8 水平向搬運的誤差與完成時間 (針對合力應用)

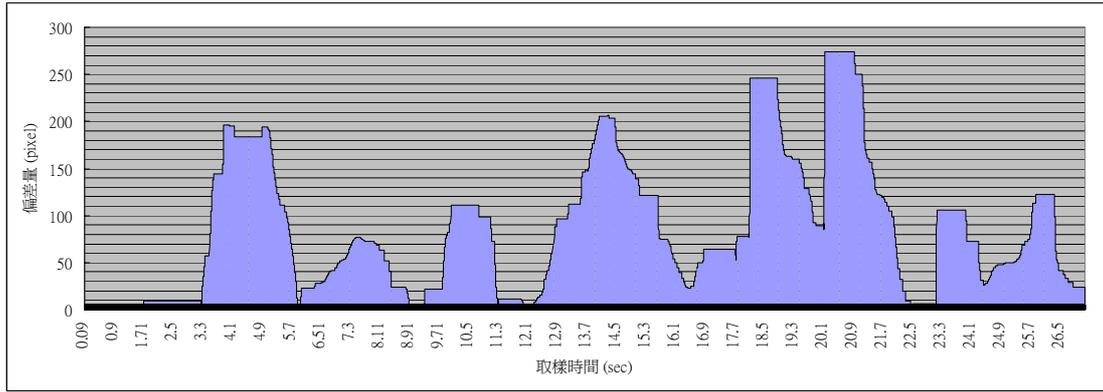


圖 5-9 垂直向搬運的誤差與完成時間 (針對合力應用)

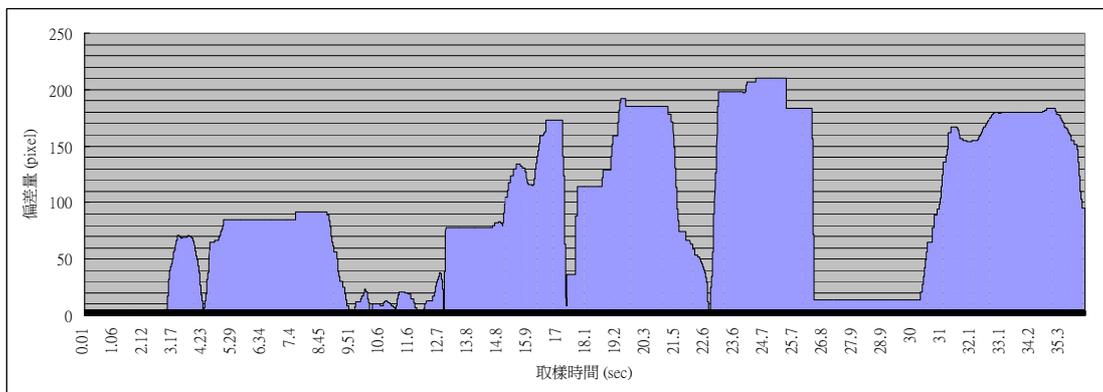


圖 5-10 圓軌跡搬運的誤差與完成時間 (針對合力應用)

圖 5-11(a)~(c)，分別是原力回饋系統與調整後系統，針對 10 次取樣的偏差位置作平均，再進行兩者的誤差圓半徑比較，綠色是原系統、藍色是針對合作應用的，而不論是圓軌跡、水平向、或是垂直向測試，調整後系統所形成的圓半徑都比較小，印證了利用虛擬物體表面對操作者的反應力方向不變，可以很有效率的改善搬動的難易度，完成對表面光滑物體的流暢搬運，而原系統也算是成功地提供了其基本的力資訊流通管道。

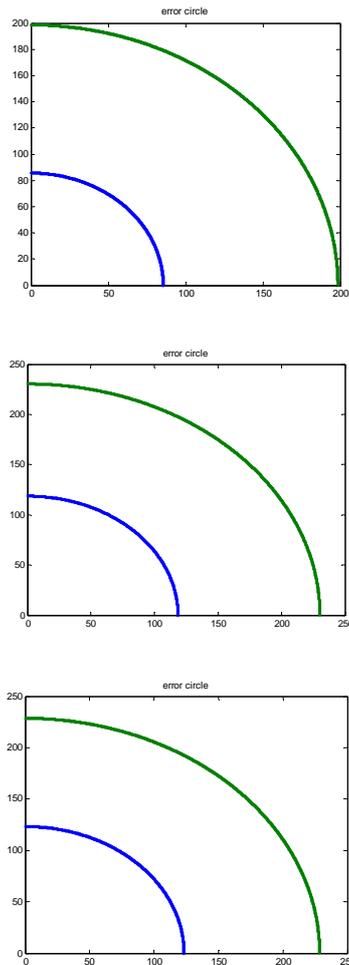


圖 5-11 原系統與調整後系統(a)水平向(上) (b)垂直向(中) (c)圓形軌跡(下) 搬運

5.3 遠距實驗

首先對於網路延遲的狀況，我們測試家用網路和實驗室中的網路延遲現象是否差異很大，進行的測試環境同樣是在非強制性即時作業系統，前面所建構的實驗即是在此情形下進行，同樣採用 Microsoft 的 Windows XP。方法為利用一台電腦作為 Client、不斷地對另一台電腦 Server 發送資料，而 Server 端接收到

資料後即發送訊息給 Client，如此可以得到資訊來回所花費的時間，連續進行 40000 筆測試結果，在實驗室的環境中進行的測試結果如圖 5-12 所示，可知操作過程中使用者可以感受到的延遲大約會在 0.2 ms 左右，而因為大部分的延遲還是在低於 1 ms 的狀態下，可知雖有少許時刻會有較大的延遲（高達 7 ms），不過大多數的網路延遲還是在低於我們實驗所設定的每次力更新時間間隔 1.54 ms，所以實驗上幾乎沒有影響，可以在力命令執行前完成傳輸。

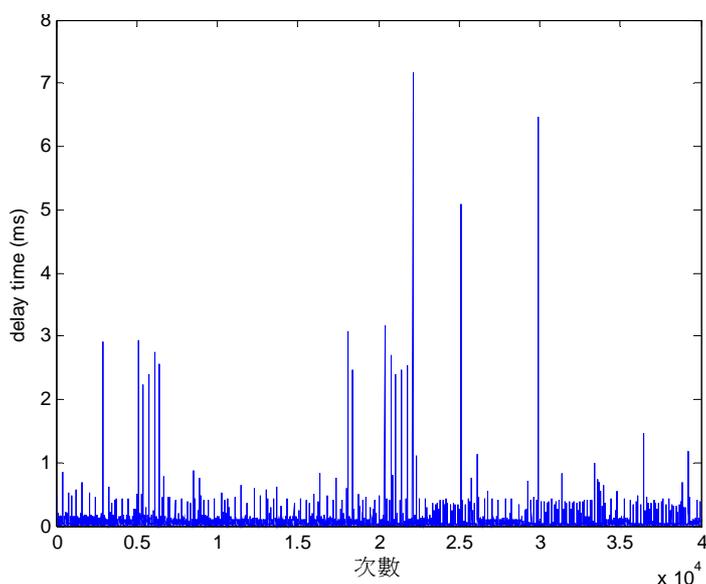


圖 5-12 實驗室測得的網路延遲狀況

但是在一般的家用網路，撥接寬頻 ADSL-2M 測試所得的網路狀況，與實驗室所用的網路傳輸速度差了近 10 倍，如圖 5-13 所示，一般情況下可以感受的網路延遲在 40 ms 上下，換言之，其能達到的力更新頻率應該在 25 Hz 左右，然而如此的力更新頻率根本不夠，會造成明顯的力不連續現象，甚至連視覺所需的更新頻率都沒有辦法滿足。

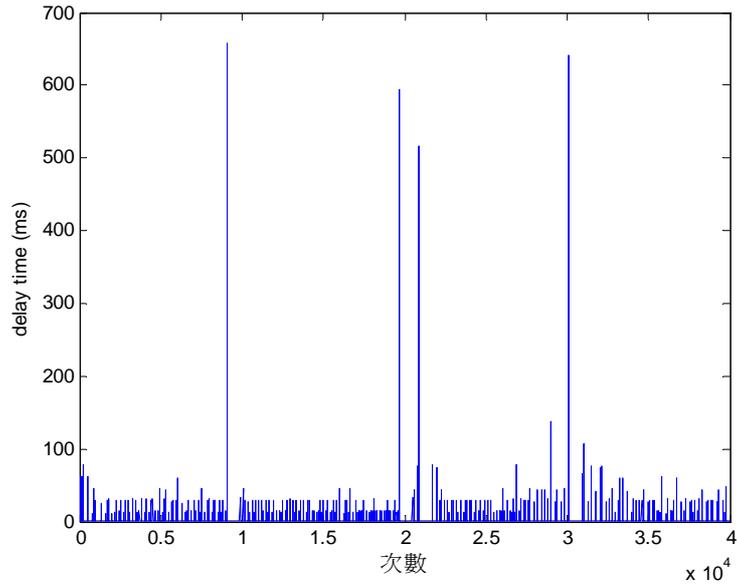


圖 5-13 家用網路所測得的網路延遲狀況

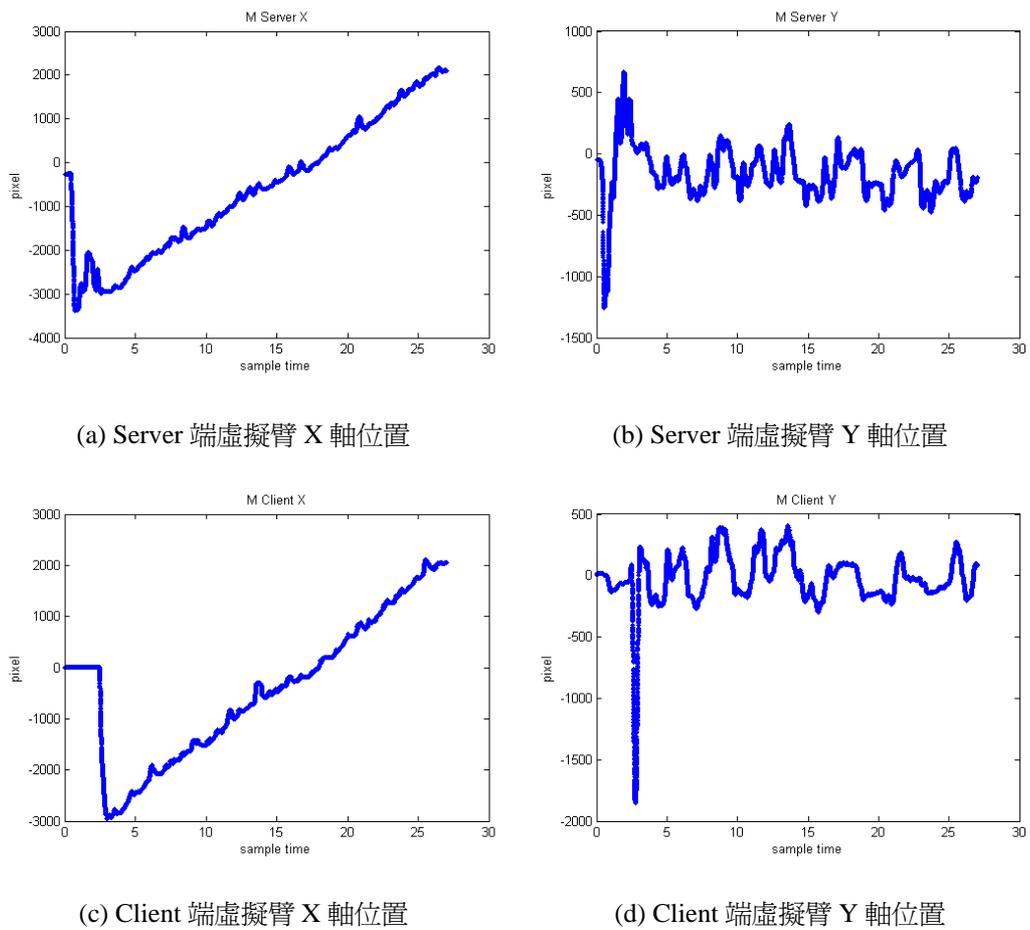
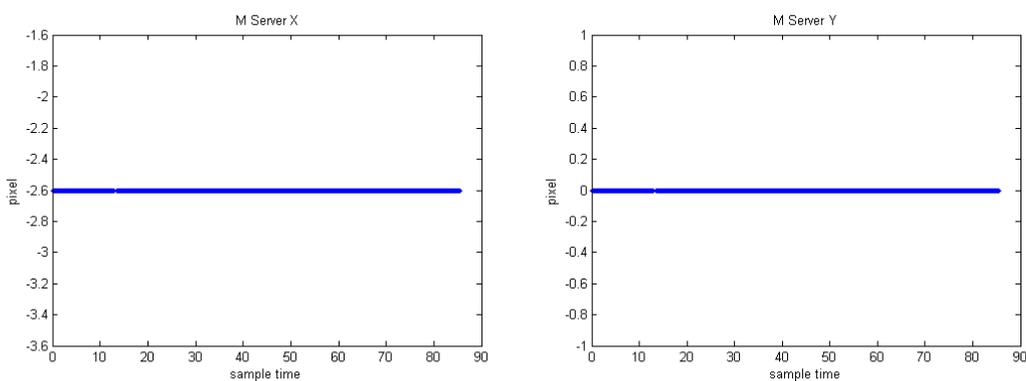


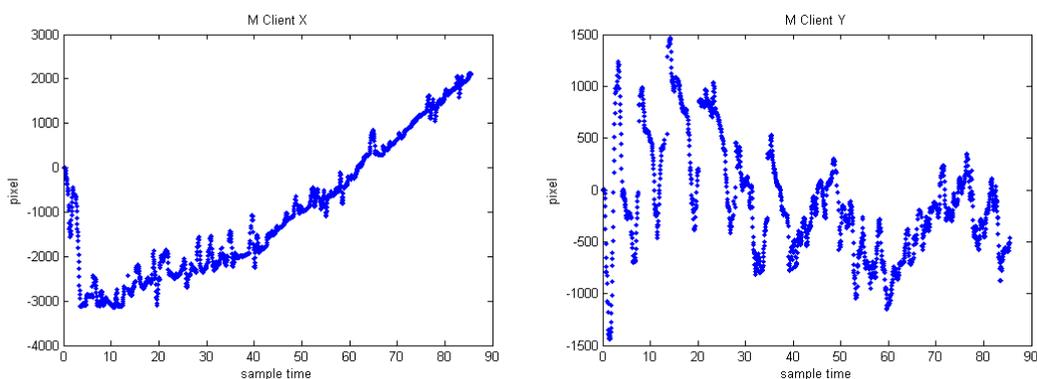
圖 5-14 實驗室網路進行水平向搬運過程中虛擬臂的位置

由於力更新頻率根本不夠，因此於家用環境中測試的結果，光是單人搬運就遭遇很大的問題，因為不連續性與資料傳遞過慢，造成力跳躍的情形，將從圖 5-14~5-16 中結果可得到驗證。圖 5-14 是實驗室所測水平向搬運最好的一次取樣的虛擬臂位置圖，而圖 5-15 和圖 5-16 則是在家用環境中進行水平向搬運的虛擬臂位置圖與施力圖。



(a) Server 端虛擬臂 X 軸位置

(b) Server 端虛擬臂 Y 軸位置



(c) Client 端虛擬臂 X 軸位置

(d) Client 端虛擬臂 Y 軸位置

圖 5-15 家用網路進行水平向搬運過程中虛擬臂的位置

從圖 5-15(c)~(d) 中很明顯的看出虛擬臂位置軌跡有跳躍發生，即產生斷點，相較之下圖 5-14 就幾乎沒有斷點的產生，而且在圖 5-15 中，還因為 Server 端的操作者並未移動搖桿 (所以圖 5-15(a)~(b) 中，代表 Server 端虛擬臂位置的 M-Server X 和 M-Server Y 皆保持在原位沒有移動，也因為 Server 端並未接觸

物體，所以圖 5-16(a)~(b) 中受力皆為零)，僅 Client 端進行搬運操作，理論上應該可以較穩定的朝單一方向施力，但從圖 5-16(c)~(d) 可看出，Client 端感受到的 X 軸和 Y 軸力都震盪的相當厲害，可以知道因為網路延遲位置傳送，造成系統回應太慢，而使得力不連續。再進行了實際操作後，整體的結論是，因為網路延遲造成相當於力更新頻率與繪圖頻率被拉低的形況，所以操作上相當的不順暢，連單人操作都顯得困難，更何況是合力操作的情形，所以要在網路上順暢的傳遞細膩力感，目前還是需要依靠寬頻專線，否則就是引入其它的預測技術、利用補插間等等的方式，以達到力的連續性。

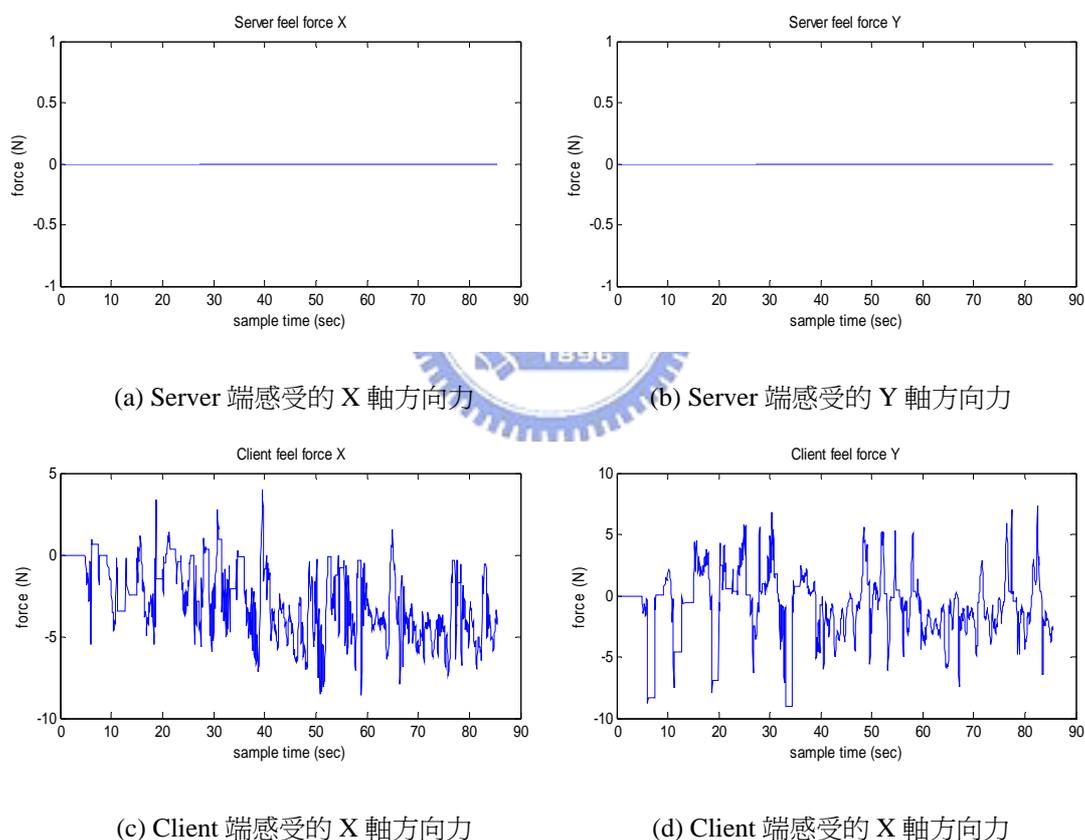


圖 5-16 家用網路進行水平向搬運過程中施力的狀況

第六章

結論與未來工作

本論文建立了一可攜性的力感系統，讓使用者可以即使身處不同地點，只要帶著力回饋裝置與安裝軟件，就可以在所建置的虛擬環境中進行力資訊的應用。而對於多人連線進行力互動的價值，除了在遊戲中可以帶來樂趣，如果可以將虛擬臂連結至真實的機械手臂，進而處理真實世界中的物件；當然，還需要克服真實物體轉換至虛擬物體的步驟，和其他像是座標轉換、夾取控制等等，但完成後，理想上就可以用多個機械手臂對物體進行合力搬運，而如此對於工業上的實際應用將可以順利發展。所發展的系統，因為是用 C 語言完成的，所以除了有執行快速的優點外，只要將原始碼經過轉譯，就算不同作業系統也可以執行，如此更達到了力可攜地目的。而對於在本文所發展的系統上，針對合力應用而做的調整，也確實地發揮了作用，可以確信此系統的力資訊流沒有問題，此後也可以再加入其它應用。

6.1 未來工作

為朝向能模擬更複雜的真實環境，以及克服網路上的障礙，還有增進系統的應用性，未來還有很多可以努力的空間，如以下所列：

1. 網路方面：為使力感能連續，可對網路延遲狀況以預測未來的力值方式進行，當超過一定延遲時即進行預測，為了達成此種目的，要進程式改寫，

使程式執行時再多一個執行緒同步運行，如此網路、力感與視覺三部份將可分開，分別運行。而預測也不只對於力量的大小和方向，也應對於移動路徑進行補償，總之使整個系統可以進行地像連續系統。

2. 高維度化：為使感覺更加真實，以及增加操作的維度，勢必要將模型復現至更高維度，而此同時也需要使用更多自由度的操作工具使其空間操作成為可能，此外復現時，因為隨著模型的精密度與維度提高，勢必會使系統效能降低，故對於 3D 空間的力感模型與視覺呈現要如何進行，可以再作研究。
3. 實體連結：可以進行虛擬世界與真實世界的連結，例如將實驗中，虛擬臂的位置連接指定為真實機械臂的末端位置，並推動真實的物體，然後同樣藉由目前的力資訊架構，感受力回饋資訊，如此要連結外部的力感測裝置，作多維度的方向與力量判別，而在此方面，整合力回饋裝置以外的實體裝置進入系統後，對其參數的調校，以及採用何種智慧控制，要再依當時應用場合而定。

參考文獻

- [1] D. C. Ruspini, K. Koralov, and O. Khatib, “Haptic interaction in virtual environments,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 128 –133, 1997.
- [2] F. Vahora, B. Temkin, T. M. Krummel, and P. J. Golman, “Development of real-time reality haptic application: real-time issues”, *12th IEEE Symposium on Computer-Based Medical Systems*, pp. 290 – 295, 1999.
- [3] J. G. Webster and D. G. Hanger, “Telepresence for touch and proprioception in teleoperator system,” *IEEE Trans. On System, Man and Cybernetics*, Vol. 18, No. 6, pp. 1020-1023, 1989.
- [4] K. B. Shimoga, “A survey of perceptual feedback issue in dexterous telemanipulation: Part I. Finger force feedback”, *IEEE Virtual Reality Annual International Symposium*, pp. 263 – 270, 1993.
- [5] L. D. Joly and C. Andriot, “Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism”, *IEEE International Conference on Robotics and Automation*, pp. 357-362, 1995.
- [6] L. Dongjun, O. Martinez-Palafox, and M. W. Spong, “Bilateral Teleoperation of Multiple Cooperative Robots over Delayed Communication Networks: Theory”, *IEEE International Conference on Robotics and Automation*, pp. 360-365, 2005.
- [7] M. A. Srinivasan and C. Basdogan, “Haptics in virtual environments: taxonomy, research status, and challenges”, *Computers and Graphics*, Vol. 21, No. 4, pp. 393-404, 1997.
- [8] M. Ouh-young, J.-R. Wu, W.-N. Tsai, T.-J. Yang, and C.-H. Huang, “A force feedback joystick and its use in PC video games,” *IEEE International Conference on Consumer Electronics*, pp. 326-327, 1995.
- [9] M. Ouh-young, W. N. Tsai, M. C. Tsai, J.R. Wu, C. H. Huang, and T. J. Yang, “A low-cost force feedback joystick and its use in PC video games”, *IEEE Transactions on Consumer Electronics*, Vol 41, Issue 3, pp. 787 -794, 1995.
- [10] M. S. Yoh, “The Reality of Virtual Reality”, *Seventh International Conference on Virtual Systems and Multimedia*, pp. 666-674, 2001.
- [11] N. Hogan, “Controlling impedance at the man/machine interface”, *IEEE International Conference on Robotics and Automation*, Vol.3, pp. 1626-1631, 1989.
- [12] N. Hogan, “Stable Execution of Contact Tasks Using Impedance Control”, *IEEE*

- International Conference on Robotics and Automation*, pp. 1047-1054, 1987.
- [13] O. M. Al-Jarrah and Y. F. Zheng, “Arm-manipulator coordination for load sharing using variable compliance control”, *IEEE International Conference on Robotics and Automation*, pp. 895-900, 1997.
- [14] R. C. Goertz and R. Thompson, “Electronically controlled manipulator”, *Nucleonics*, pp. 46-47, Nov. 1954.
- [15] S. G. Hong, J. J. Lee, and S. Kim, “Generating artificial force for feedback control of teleoperated mobile robots”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1721–1726, 1999.
- [16] T. L. Brooks, “Telerobotics response requirements”, *IEEE International Conference on Systems, Man and Cybernetics*, pp. 113 – 120, 1990.
- [17] T. Tsumugiwa, R. Yokogawa, and K. Hara, “Variable impedance control with virtual stiffness for human-robot cooperative peg-in-hole task”, *IEEE International Conference on Intelligent Robots and Systems*, pp. 1075-1081, 2002.
- [18] Y. Adachi, T. Kumano, and K. Ogino, “Intermediate representation for stiff virtual objects”, *IEEE Virtual Reality Annual International Symposium*, pp. 203 – 211, 1995.
- [19] *Virtual hands reach across ocean*,
<http://news.bbc.co.uk/1/hi/technology/2371103.stm>
- [20] *Open GL 超級手冊*, 大新資訊譯, 碁峰, 台北市, 2000.