

國立臺灣海洋大學
資訊工程學系
碩士學位論文

指導教授：翁世光

即時水下無人載具運動模擬器
Real-Time ROV Motion Simulator

研究生：周杰士 撰
中華民國 108 年 6 月

即時水下無人載具運動模擬器

Real-Time ROV Motion Simulator

研究 生：周杰士

Student : Chieh-Shih Chou

指 導 教 授：翁世光

Advisor : Shyh-Kuang Ueng

國 立 臺 灣 海 洋 大 學
資 訊 工 程 學 系
碩 士 論 文

A Thesis

Submitted to the Department of Computer Science and Engineering

College of Electrical Engineering and Computer Science

National Taiwan Ocean University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

In

Computer Science and Engineering

June 2019

Keelung, Taiwan, Republic of China

中華民國 108 年 6 月

致謝

如白駒過隙的兩年，努力不懈，不知不覺中也到了這時間，並且完成了論文，在這段時間內，遇到了許多的問題，也有非常多的人給予幫助。首先，感謝翁世光教授這兩年內對於我論文的諸多指導，他常常鼓勵我們努力向上並且出國看看世界，尋求更好的環境與待遇。同時也感謝黃志鵬教授與盧東華教授，抽空前來擔任我的口試委員，對於論文提供了很多寶貴的意見與修改方式，使我的論文能更加合理與完整。

除了感謝教授們的指導，也感謝圖學實驗室的所有研究同學、夥伴們；感謝陳陸剛學長，對於我程式能力的諸多不足給予幫助，造成他諸多困擾與麻煩，希望他可以諒解。感謝張智惟同學，此篇論文為我與他共同合作討論的結晶，對於實作的諸多細節與方法，他的意見非常寶貴。感謝黃信誠、黃炫愷同學，同樣也有對於論文提出很多有用的參考方法與意見。另外感謝同窗的好友們，任思堯、邱垂邦、何冠杰、歐陽飛鴻等，在這兩年的於海洋大學的生活，有了你們這群好夥伴，生活多了許多樂趣。

最後，感謝我的母親，感謝她在這兩年內能接受我不太常回家，雖然她還是會一直問我為何不回家，希望他能體諒到生為研究生的苦衷。再一次感謝於這段時間內，所有不論是指導、幫助與陪伴我的人，對於這段時間內學會的寶貴知識與人生哲學，我會銘記在心，並將他們帶往我下一個人生階段，謝謝。

周杰士 謹誌 於

國立台灣海洋大學資訊工程學系

中華民國 108 年 7 月

摘要

ROVs(Remotely Operated Underwater Vehicles)被廣泛用於執行水下任務，在深海的環境中，ROV 會受到多種環境因素影響，譬如海流、地形障礙、視野不佳等等，當操作員無法熟悉這些突發狀況，會使的任務難以執行，且因 ROV 器材昂貴，執行一次任務所需要的成本過高，為避免器材的損傷與減少時間成本，操作員必須更加熟悉任務流程與操作方式，因此，本研究目的為以遊戲程式設計(Game Programming)的方式建立一個包含人工智慧引擎(Artificial Intelligence Engine)、物理引擎(Physics Engine)與繪圖引擎(Graphics Engine)三大引擎的即時水下無人載具運動模擬器(Real-Time ROV Motion Simulator)。

在人工智慧引擎的部分，為了實現一套能從指令轉換為動作的操作系統，本研究利用 PID 控制器(Proportional – Integral – Derivative controller)，將使用者的指令或搖桿的訊號以參數的方式輸入，再透過逆運動學(Inverse Kinematic)計算多個推進器之間的力，轉換為一個「推進器向量(Propeller Vector)」。

在物理引擎的部分，為了計算 ROV 的質量與體積分布並用於計算水下靜力(Hydrostatic)，本研究使用體素化(Voxel-based)的方法來估算物體的質量與體積分布，水下動力(Hydrodynamic)的部分，利用人工智慧引擎算出的推進器向量，總和其推進器的力與力矩(Moments)，並根據公式計算 ROV 所受到的阻力，再透過運動學公式來更新 ROV 讓其運動。

最後透過繪圖引擎與人機介面，將上述內容顯示給使用者，並透過此模擬器，提供操作員操作訓練模擬與行前的任務演練等用途。

關鍵字：遠端遙控載具、運動模擬、人工智慧引擎、PID 控制器、物理引擎、水下動力。

Abstract

Remotely Operated Underwater Vehicles(ROVs) are widely used to perform underwater missions. In deep sea environments, ROVs are subject to a variety of environmental factors, such as currents, terrain obstacles, poor visibility, etc. When operators are not familiar with these, the situation can make the task difficult to perform. Furthermore, the ROV equipment is too expensive, the cost of performing a task is too high. To avoid equipment damage and reduce time costs, operators must be more familiar with the task procedure and operation. Therefore, the purpose of this thesis is to use Game Programming method to build a real-time ROV motion simulator that includes the three engines - Physics Engine, AI Engine and Graphics Engine.

In the part of the artificial intelligence engine. In order to realize an operating system that can convert instruction to action, this study uses the Proportional – Integral – Derivative controller to input the user's command or joystick signal as a parameter. Then using Inverse Kinematic to calculate the force between multiple thrusters and converts them into a "Propeller Vector".

In the part of physics engine. First, we use voxel based method to calculates hydrostatics force parameter such as mass, volume, and tensor of inertia. Second, to compute hydrodynamics, this paper use the propeller vector calculated by artificial intelligence engine to sums force and torque. Then using empirical formula to compute drag. In the end of physics engine, using kinematics formulas to update ROV's parameter and make ROV animated.

Finally, through the graphics engine and the human-machine interface, the above contents are displayed to the user. With this simulator, we can provide users for operating training simulations, pre-departure exercise, and any other purpose.

Keyword: Remotely Operated Underwater Vehicles, Motion Simulator, Artificial Intelligence Engine, PID Controller, Physics Engine, Hydrodynamics.

目次

摘要.....	I
Abstract.....	II
目次.....	III
圖目次.....	VI
表目次.....	VIII
第 1 章 緒論	1
第 1-1 節 前言	1
第 1-2 節 相關研究.....	2
第 1-3 節 研究方法與論文結構.....	2
第 2 章 背景知識	4
第 2-1 節 ROV 的參考中心.....	4
第 2-1-1 節 重量中心(Gravity Center)	4
第 2-1-2 節 浮力中心(Buoyancy Center)	4
第 2-1-3 節 定傾中心(Meta Center)	4
第 2-2 節 座標系統.....	6
第 2-2-1 節 世界座標系(World Coordinate System).....	6
第 2-2-2 節 身體座標系(Body Coordinate System)	6
第 2-2-3 節 運動座標系(Motion Coordinate System)	6
第 2-3 節 六軸自由度運動(6 Degree of Freedom Motion)	7
第 2-3-1 節 線性(平移)運動(Linear Motion)	7
第 2-3-2 節 旋轉運動(Angular Motion)	7
第 2-4 節 水下力學	8
第 2-4-1 節 水下靜力學(Hydrostatics)	8
第 2-4-2 節 水下動力學(Hydrodynamics).....	8
第 2-5 節 穩定力矩	9

第 3 章 系統架構	11
第 3-1 節 虛擬 ROV 架構.....	13
第 3-2 節 ROV 的基本物理量.....	14
第 3-2-1 節 質量、質心、體積與浮心的計算	14
第 3-2-2 節 慣性張量的計算	15
第 3-2-3 節 可變物理量的計算	15
第 3-3 節 ROV 的推進器	16
第 3-3-1 節 推進器的位置	16
第 3-3-2 節 推進器的推力方向	16
第 4 章 人工智慧引擎	17
第 4-1 節 PID 控制器.....	17
第 4-1-1 節 PID 控制器的運用	18
第 4-2 節 推進器向量	19
第 4-2-1 節 推進器向量的運用	19
第 4-2-2 節 計算推進器向量	20
第 5 章 物理引擎	21
第 5-1 節 物理引擎流程	21
第 5-2 節 計算力與力矩	22
第 5-2-1 節 重力與浮力	22
第 5-2-2 節 推進器推力	22
第 5-2-3 節 阻力	23
第 5-3 節 運動學計算(Kinematics)	25
第 5-3-1 節 線性運動	25
第 5-3-2 節 旋轉運動	26
第 5-4 節 轉換矩陣(Transformation Matrix)的建立	27
第 5-4-1 節 線性運動的影響	27
第 5-4-2 節 旋轉運動的影響	28
第 5-4-3 節 平移矩陣 Tb	29
第 5-4-4 節 旋轉矩陣 Rb	29

第 5-4-5 節 轉換矩陣 Mb	29
第 6 章 繪圖引擎與人機介面	30
第 6-1 節 虛擬實境(Virtual Reality)	30
第 6-2 節 ROV 歐拉角的視覺化與計算.....	32
第 6-3 節 操作介面.....	35
第 6-3-1 節 指令模式	35
第 6-3-2 節 搖桿模式	36
第 7 章 實作結果	38
第 7-1 節 基本物理量	38
第 7-2 節 穩定測試.....	39
第 7-3 節 自動控制測試.....	40
第 7-4 節 手動操作測試.....	45
第 8 章 結論與未來發展	46
參考文獻.....	47
附錄.....	49

圖目次

圖 1-1 ROV 模擬器總覽	3
圖 2-1 水面船舶定傾中心	4
圖 2-2 水下船舶定傾中心	5
圖 2-3 ROV 模擬器的三個座標系統	6
圖 2-4 線性運動	7
圖 2-5 旋轉運動	7
圖 2-6 水下靜力示意圖	8
圖 2-7 水下動力示意圖	8
圖 2-8 ROV 水面穩定力矩	9
圖 2-9 ROV 水下負穩定力矩	9
圖 2-10 ROV 水下正穩定力矩	10
圖 3-1 系統架構	11
圖 3-2 ROV Investigator 90	13
圖 3-3 海洋大學系工系 ROV	13
圖 3-4 虛擬 ROV 模型	13
圖 3-5 ROV 多邊形模型	14
圖 3-6 浮塊多邊形模型	14
圖 3-7 浮塊體素模型	14
圖 3-8 推進器的位置	16
圖 3-9 推進器的施力方向	16
圖 4-1 ROV 控制器	17
圖 4-2 PID 反饋迴路	17
圖 4-3 推進器向量	19
圖 5-1 物理引擎流程	21
圖 5-2 形狀阻力與摩擦阻力	23
圖 5-3 切線速度	24
圖 5-4 身體原點的線性位移	27
圖 5-5 身體原點的旋轉位移	28
圖 6-1 模擬器主畫面	30
圖 6-2 ROV 攝影機配置	31
圖 6-3 六軸平行投影鏡頭	31
圖 6-4 第三人稱自由鏡頭	32
圖 6-5 ROV 歐拉角羅盤	32
圖 6-6 歐拉角的計算-橫搖角	34
圖 6-7 資訊(Information)欄	35

圖 6-8 功能(Function)欄	35
圖 6-9 第一搖桿-ROV 控制搖桿	36
圖 6-10 一~三節機械手臂自由度	36
圖 6-11 四~六節機械手臂自由度	37
圖 6-12 第二搖桿-機械手臂控制搖桿	37
圖 7-1 質心與浮心的位置	38
圖 7-2 考慮機械手臂水下靜力平衡	39
圖 7-3 無考慮機械手臂水下靜力平衡	39
圖 7-4 自動穩定平衡姿態	39
圖 7-5 穩定位置測試	40
圖 7-6 基本運動-Turning Circle	45
圖 7-7 基本運動-Helix	45

表目次

表 3-1 ROV Investigator 90 參數.....	13
表 3-2 推進器的位置	16
表 3-3 推進器的施力方向	16
表 4-1 六軸運動於運動座標系的推進器向量範例(質心於物體中央).....	19
表 5-1 阻力係數	23
表 7-1 ROV 基本物理量	38
表 7-2 自動控制測試-位置維持(Positioning).....	41
表 7-3 自動控制測試-姿態維持(橫擺 Yaw-航向)	42

第1章 緒論

第1-1節 前言

ROV 遠端遙控載具屬於水下無人載具 (Unmanned Underwater Vehicles, UUVs) 的一種，泛指所有水下無載人的潛航器與水下機器人，水下無人載具誕生於 50 年代，第一台 UUV 為法國人於 1953 年發明，屬於 ROV，經歷數十年後，各國對於 UUV 的技術能力逐漸重視。最早將 UUV 用於軍事用途的是美國海軍，水下專用研究載具 (Special Purpose Underwater Research Vehicle, SPURV)，其最大下潛深度達到 3600 公尺深，而工作深度為 3000 公尺，動力為電池，每次可續航四至五小時，屬於自主式無人水下載具 (Autonomous Underwater Vehicles, AUVs)，主要用於海底水文資料調查。SPURV 一直被使用到 1970 年代，才由 SPURV II 取代。另外，水下滑翔翼 (Underwater glider) 亦屬於 AUVs 的分支之一，雖然其續航力可達到數千公里遠的優勢，但因為負重能力差，且造價昂貴，僅有少數國家研究，在 2016 年 12 月於南海被中國捕獲的美國 UUV 屬於此種類[24]。

進入 21 世紀後，隨著人類開發海洋，UUV 在從石油開採、海底礦藏調查、水下搜尋與打撈作業、海洋科學探勘及研究，到海底管線、水下結構物、海底電纜、鑽油平台、港灣碼頭等水下設施檢測作業，都發揮重要的功用。因此，作為海洋國家的台灣，近年來不只是潛艦國造，對於探索海洋與爭奪國際海底資源的重要工具 UUV 也更趨於重視。

根據上述可知，UUV 主要分為兩種，一種為有纜線 ROV，而另一種為無纜線的 AUV。ROV 靠著纜線與水面上的母船連結來傳輸資訊與獲得電力，因此對於電力來源不用太過擔心，但也限制其水下工作的深度。因 1970 年發生的石油危機所帶動的海底探油開採需求而蓬勃發展，至今已逐漸取代潛水員成為水下作業的主流，再加上能夠加裝機械手臂執行水底樣本採集、水下電鋸接管施工、水下打撈、水雷清除等工作，ROV 就如同一具水下機器人。而 AUV 不需仰賴水面母船所提供的電力與操作，靠著自身內建的電池與導航系統來執行水下的工作，通常被用於特殊環境的水下調查、搜尋與監測。

ROV 需要由母船上的人員來操作，但由於在深海的環境中，ROV 會受到多種環境因素影響，譬如海流、地形障礙、視野不佳等等。操作員必須受到良好的訓練與熟悉任務流程，因此本研究目為建立一個即時水下無人載具運動模擬器，以提供操作員操作訓練模擬與行前的任務演練等用途。

本模擬器擁有逼真的虛擬實境(Virtual Reality)、真實的 ROV 操作介面、豐富的參數視覺化界面與人工智慧引擎的自動駕駛系統，提供使用者進行訓練與學習；並且，本模擬器為一個具彈性、可擴張性的系統，使用者可擴充或更改模擬環境與模擬對象 ROV。在創新部分，本研究考慮了機械手臂對於水下靜力的影響、內部計算歐拉角的方法也不同於以往，並且建立一套推進器向量(Propeller Vector)系統，計算 ROV 的目標推力與各個推進器之間的關係，以此來實現逆運動學(Inverse Kinematics)。

第1-2節 相關研究

如前言所敘，水下無人載具已經有近 50 年的歷史，期間內累積了不少研究成果，透過閱讀相關的文獻資料，從中學習與實驗各種水下無人載具所需的知識，並透過這些來幫助本研究完善模擬系統。

在[14]這篇論文中，Evens 與 Nahon 對一個最簡易的、只有船體與水翼的 AUV 進行模擬並研究與驗證。在[23]這篇論文中，Prestero 開發了一個包含六軸非線性運動的 AUV 模擬器，模擬器不僅包含了水下靜力、水下動力的升力、阻力，還有 Added mass、推進器等等參數。在[12]這篇論文，Deng 與 Han 為了討論 ROV 的空間機動性，其利用泰勒級數分析水下動力，並著重於三個運動上：垂直運動、旋轉運動與 Z 型運動。另外[6, 15, 18, 27]這幾篇論文都將整個系統流程清楚的說明，並且均利用 VR(Virtual Reality)模擬水下載具的運動與駕駛。

對於水下物體的模擬，都需要計算水下運動學，因此剛體運動的知識非常重要，透過[2]、[3]兩篇說明剛體運動的模擬流程與數學模型，建立水下物體的運動知識基礎，對於水下靜力、水下動力與阻力，在上述論文中都有充分的研究與測試，而最主要差距的部份為如何統合控制 ROV 的各個推進器，主要可以分成兩派方法，一種使用變結構控制(Variable Structure Control, VSC)，譬如[7-9]，在[8]這篇論文中，Corradini 與 Orlando 使用變結構控制來完成多輸入多輸出(MIMO)的多個推進器 ROV 系統，而另一派如[10, 19, 20, 25, 28]使用 PID 控制器，在[19]這篇論文，Geder 等人為了解決近岸和沿海地區任務所需的低速和高機動性能，使用 Fuzzy Logic PID Based 來控制水翼的變化，以達到穩定且高機動性的運動，同樣的，Khodayari 與 Balochian 在[20]這篇論文中，也使用 Fuzzy Logic PID Based 的技術來完成一個非線性 MIMO 架構的 AUV，而在 Tehrani 等人提出的[25]這篇論文內，他們使用 PID 來完成深度控制(Depth Control)與穩定分析。

除了相關的文獻，網路上也有很多 ROV 的模擬程式譬如[21, 26]，而在[21]中較著重於機械手臂的運動模擬，機械手臂作動時物理量的變化不明顯，但主 ROV 運動模擬的流程非常流暢，並輔以一些任務關卡，使模擬器能達到訓練操作員的目的，而本研究為了將機械手臂對於 ROV 本體的影響也加入考量，透過即時的對 ROV 的物理量作更新來達到此目的，在[5, 16, 17]內討論的以 Articulated-body Inertia 來計算包含機械手臂的相關運動，但是其探討的主要是立於地面的機器人，當其機械手臂運動時所產生的內力關係，而非這裡本研究所要討論的物理量變化。

第1-3節 研究方法與論文結構

UUV 雖分為 ROV 與 AUV，但其模擬器均可視為水下物體之運動模擬，而 AUV 大多為滑翔翼或具水翼的潛艇狀結構，其提供動力輸出的推進器不多，多為使用水翼與舵來進行操控，ROV 則利用多個推進器來達到六軸運動，且能進行機械手臂的安裝以進行更多水下工作，因此本研究以較多推進器與加裝機械手臂的 ROV 模擬為目標來進行設計開發。

在整個系統架構中，本研究參考遊戲設計的方法，將整個模擬程序分為三大部分，第一部分為負責操作 ROV 運動的人工智慧引擎，由第 1-2 節可知有 VSC 與 PID 兩種方法，而本研究使用 PID 控制器的方法，利用 PID 將使用者輸入的訊號轉換為一個用於統一推進器的系統—推進器向量，第二部分則是負責統合所有力與力矩並進行如[8, 14, 23]運動學計算的物理引擎，力與力矩又可以根據來源將其分為水下靜力與水下動力，在計算水下靜力之前，必須計算 ROV 的物理量如質量、體積、質心、浮心與慣性張量，而舊有的研究大多使用[22]的切片理論(Strip Theory)來計算，本研究在此使用體素化(Voxelization)的方法，水下動力的部分則根據人工智慧引擎得出的推進器向量來統合推進器的力，並利用公式計算 ROV 所受到的阻力，之後透過牛頓第二運動定律(Newton's Second Law)與歐拉第二定律(Euler's second law)的運動學公式計算 ROV 新的位置與姿態。最後一部份則為提供使用者控制、輸入與視覺化 ROV 參數的繪圖引擎與人機介面，在人機介面本研究提供使用者可以直接下達命令，如前往至某目標地，或是透過搖桿來操作 ROV，並且將物理引擎內重要的資訊計算後透過人機介面顯示與畫出 ROV 的運動狀態，如圖 1-1 ROV 模擬器總覽。

本論文的整體結構如下：第二章將介紹各種相關的背景知識，包括水下力學、六軸自由度運動與 ROV 在進行運動學計算時需要注意的參考座標中心與空間座標系統，第三章為本系統的架構，會將整個模擬系統的模組分別描述並講解系統流程，並計算整個模擬系統的前置作業，第四、第五與第六章分別為物理引擎、人工智慧引擎、繪圖引擎與人機介面，詳細描述了各個引擎內部的運作與其計算方式，第七章為實作結果，呈現使用本模擬器進行一些基本測試與基本的操作結果，最後，在第八章進行總結，並提出未來展望與後續研究方向。

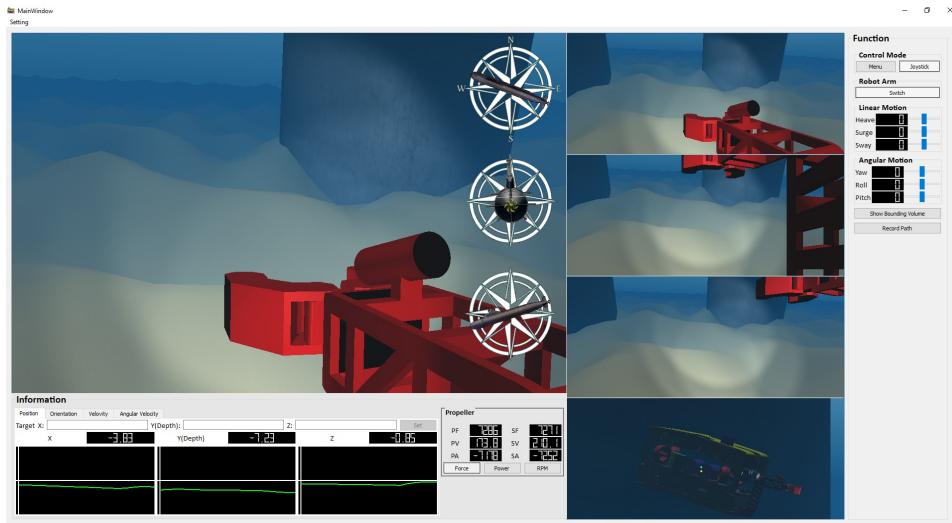


圖 1-1 ROV 模擬器總覽

第2章 背景知識

第2-1節 ROV 的參考中心

在運動模擬時，首先就是訂一好物體的參考點與參考座標系統，對於 ROV 而言，必須先談到三個參考點：重量中心(Gravity Center)、浮力中心(Buoyancy Center)、定傾中心(Meta Center)。

第2-1-1節 重量中心(Gravity Center)

ROV 重量中心也是其質量中心，稱其為 G，由 ROV 的質量分布情形決定，而 ROV 的重力即作用於重量中心。大多數 ROV 都有一至兩軸對稱，以確保重量中心於物體中心軸上。

第2-1-2節 浮力中心(Buoyancy Center)

浮心稱為 B，其位於 ROV 排開水的體積中心，而浮力即作用於浮心上。浮體處於平衡時，浮力的大小等於浮體的重力。

第2-1-3節 定傾中心(Meta Center)

當 ROV 於水面傾斜時，因其水下體積發生變化，使原本平衡的浮力與重力作用於不同平面，因此產生力矩使 ROV 旋轉。而旋轉的參考中心，稱其為定傾中心，又稱為穩心。定傾中心的位置為以平衡狀態時的重心G作一鉛垂線L1，同時L1通過G與B，並與船體傾斜後的新浮心B'其鉛垂線 L2 相交，交點即為定傾中心 M。

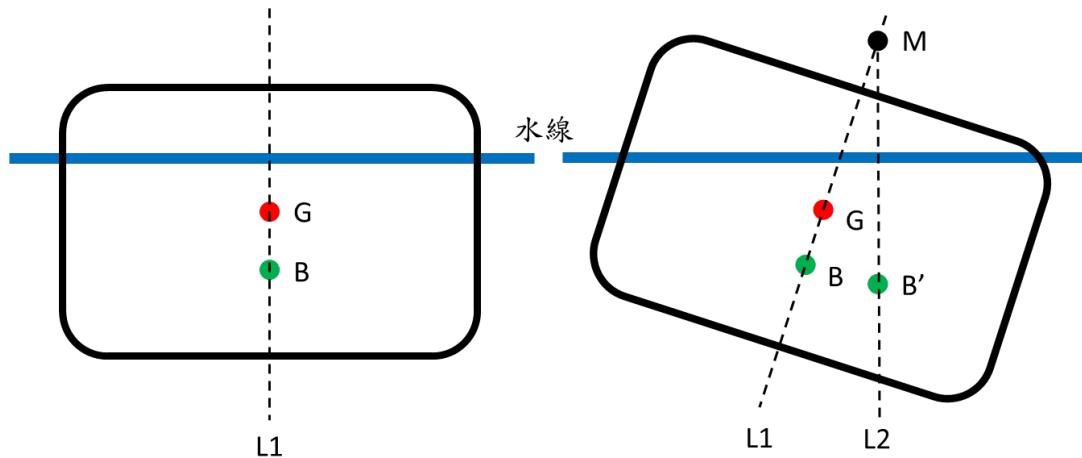


圖 2-1 水面船舶定傾中心

水面上的 ROV 其定傾中心會依據其水下體積而改變，但對於水下的 ROV 而言，卻並非如此，因為水面下的 ROV 不論其如何旋轉，其體積均固定，浮心也不會因此而改變，因此可以發現，水面下 ROV 其定傾中心即為其浮心。

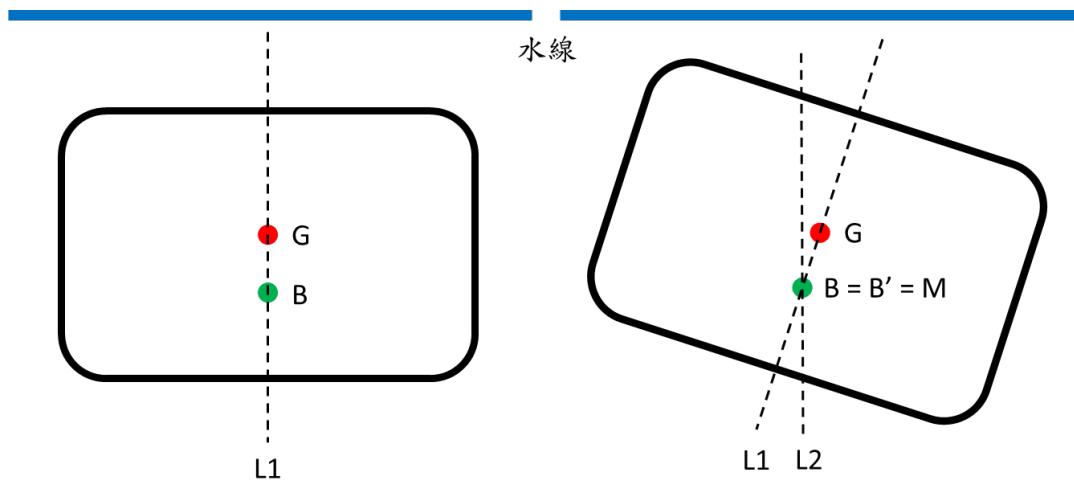


圖 2-2 水下船舶定傾中心

第2-2節 座標系統

本研究因物理引擎與繪圖引擎的需求，必須考慮三個座標系統，世界座標系(World Coordinate System)、身體座標系(Body Coordinate System)、運動座標系(Motion Coordinate System)，這三種座標系統如圖 2-3 ROV 模擬器的三個座標系統所示。

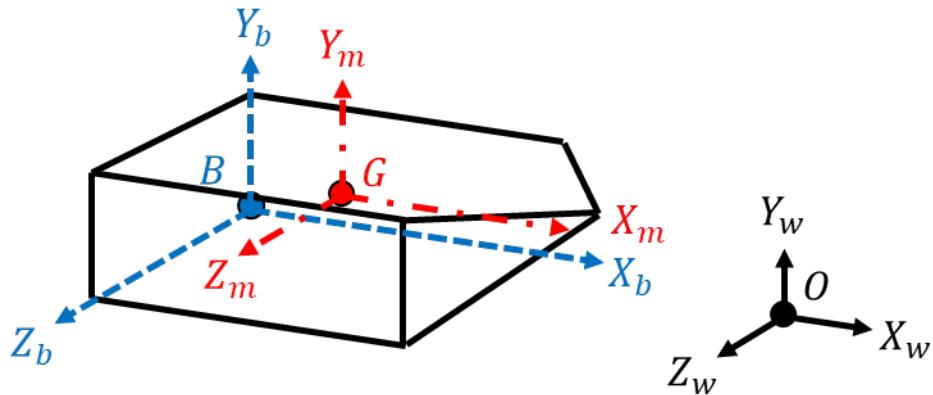


圖 2-3 ROV 模擬器的三個座標系統

第2-2-1節 世界座標系(World Coordinate System)

世界座標系是以場景中心 O 為原點，水平張出 Z_w 及 X_w 兩軸，並以 Y_w 軸指向天空，如圖 2-3 中以黑色座標系統所表示的座標系。世界座標系中定義了所有物件的實際位置，用以計算物件之間的碰撞與場景渲染，如海床、地形、障礙物等，所以整個場景由世界座標系統定義。

第2-2-2節 身體座標系(Body Coordinate System)

身體座標系其原點是模型之原點，三軸即為建模時所定義之三軸，如圖 2-3 圖 2-3 ROV 模擬器的三個座標系統中的藍色座標系統所示。ROV 內的所有物件的位置均利用此座標系定義；譬如推進器的位置、推進器的施力方向、探照燈的位置、探照燈的角度、機械手臂的位置、每一節機械手臂的旋轉角度等等，所以 ROV 的所有元件的幾何資訊，例如位置與法向量，均以身體座標系定義。

第2-2-3節 運動座標系(Motion Coordinate System)

運動座標系其參考點以重心 G 為原點，此座標系的三軸會平行於身體座標系的三軸，如圖 2-3 中紅色座標系統所示。本座標系統主要用來定義六軸運動，沿 X_m 軸平移即為縱移(Surge)，沿 Z_m 橫平移即為橫移(Sway)，而沿 Y_m 軸上下平移即為起伏(Heave)。而以重心為旋轉中心，依 X_m 軸為旋轉軸旋轉稱其為側傾(Roll)，依 Z_m 軸為旋轉軸旋轉稱其為縱搖(Pitch)，而依 Y_m 軸為旋轉軸旋轉稱其為橫搖(Yaw)。

第2-3節 六軸自由度運動(6 Degree of Freedom Motion)

六軸自由度運動是指剛體在三維空間中的運動自由度。具體來說，ROV 可以在三個垂直軸上自由改變前進/後退，上/下，左/右平移，並通過繞三個垂直軸旋轉的方向變化。

第2-3-1節 線性(平移)運動(Linear Motion)

線性運動分別對應物體的三軸：

在X軸前後移動的縱移(Surge)

在Z軸左右移動的橫移(Sway)

在Y軸上下移動的起伏(Heave)

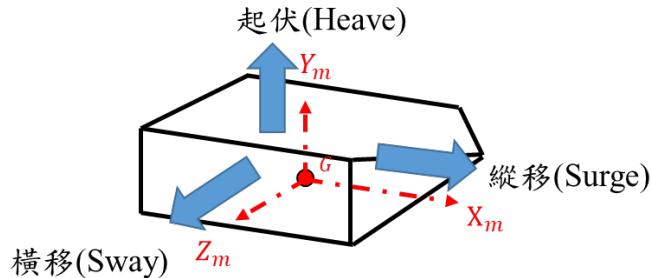


圖 2-4 線性運動

第2-3-2節 旋轉運動(Angular Motion)

旋轉運動依物體的三軸做旋轉：

依X軸旋轉的側傾(Roll)

依Z軸旋轉的縱搖(Pitch)

依Y軸旋轉的橫搖(Yaw)

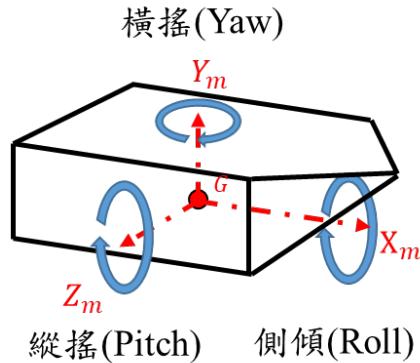


圖 2-5 旋轉運動

第2-4節 水下力學

水下力學主要是研究水下載具受力與運動的關係，可分為水下靜力學(Hydrostatics)和水下動力學(Hydrodynamics)，前者研究 ROV 處於水下時受到的重力與浮力作用，產生運動與姿態的改變；後者研究螺旋槳推力對於 ROV 運動的影響。

第2-4-1節 水下靜力學(Hydrostatics)

在 ROV 運動速度低於 3 節(Knot)時，其運動方式由水下靜力主導，水下靜力主要包含 ROV 的浮力與重力。當 ROV 處於水下時，其所受的浮力作用於浮心，而重力作用於重心，力的方向為垂直方向，如圖 2-6 所示；此兩種力產生線性運動，改變 ROV 的深度。而這兩種力對應於重心所產生的力矩會造成旋轉運動，改變 ROV 的姿態。

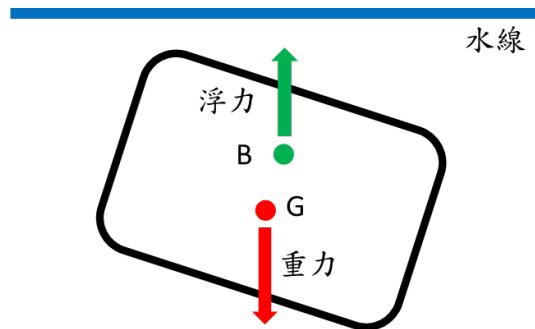


圖 2-6 水下靜力示意圖

第2-4-2節 水下動力學(Hydrodynamics)

ROV 的運動速度高於 3 節時，其運動方式由水下動力主導。對於 ROV 而言，其水下動力的來源為推進器所產生的推力與其運動時所產生的阻力(Drag)，如圖 2-7 所示。水下動力能產生六軸運動，包含三種線性運動與三種旋轉運動，這六種運動在第 2-3 節節中均有所提及。

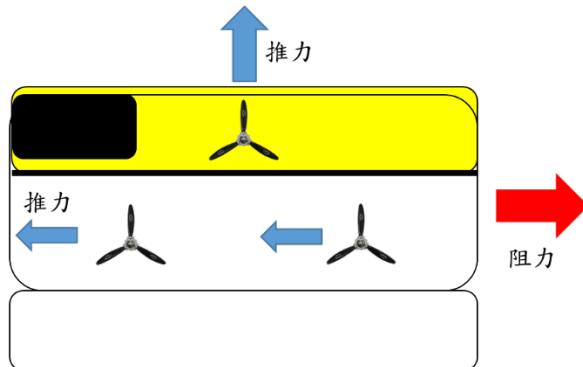


圖 2-7 水下動力示意圖

第2-5節 穩定力矩

當重心與浮心作用於不同平面所因此產生力矩使物體旋轉，此力矩會使船體恢復穩定狀態，因此稱此力矩為穩定力矩，穩定力矩的計算與船體的重力中心與浮力中心有關。

在水面上的 ROV，若重心位於浮心上方，當產生穩定力矩時，就能使 ROV 回復至穩定狀態，如圖 2-8，穩定力矩會以定傾中心做為旋轉中心進行逆時鐘旋轉。

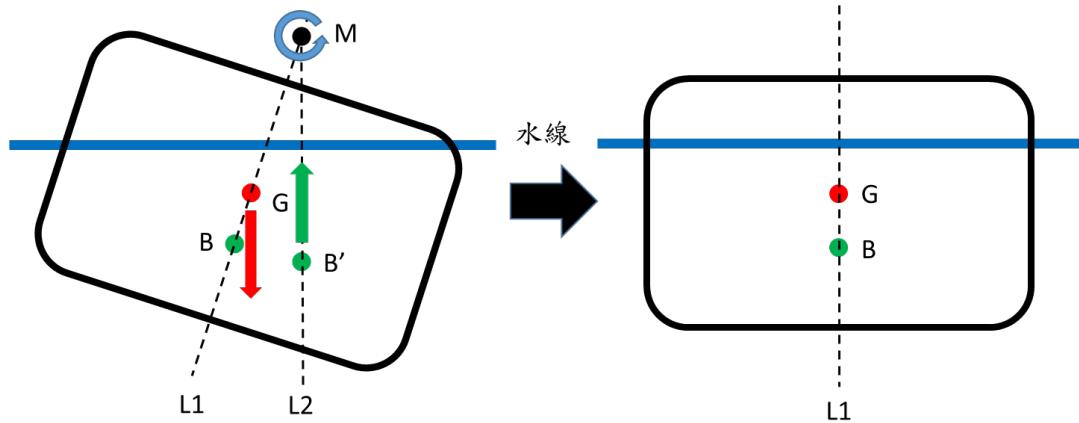


圖 2-8 ROV 水面穩定力矩

而水下的 ROV 不同於水面，若重心位於浮心之上，所產生的力矩並無法使其回復穩定，反而會使潛艇翻覆。如圖 2-9 所示，重力與浮力所產生之合力矩使物體依穩心順時鐘旋轉，造成傾角增大。反之如圖 2-10，若重心位於浮心下方，所產生之穩定力矩依穩心逆時鐘旋轉物體，使其回復平衡。

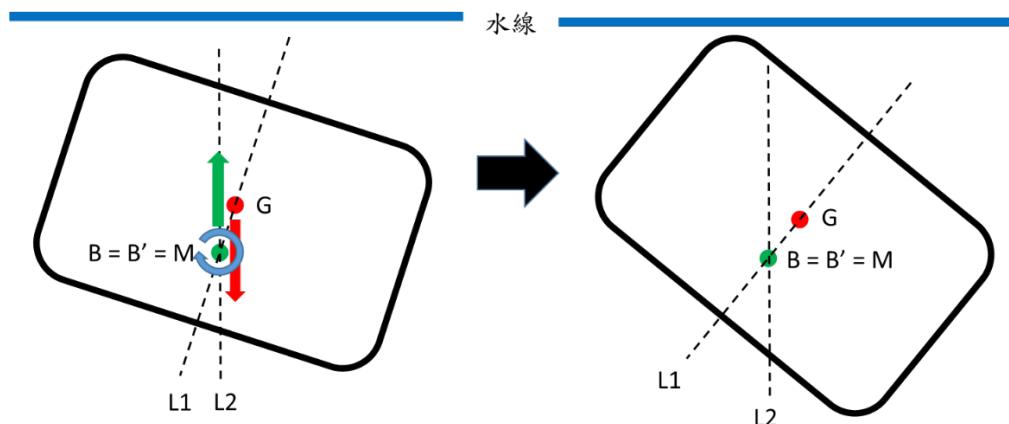


圖 2-9 ROV 水下負穩定力矩

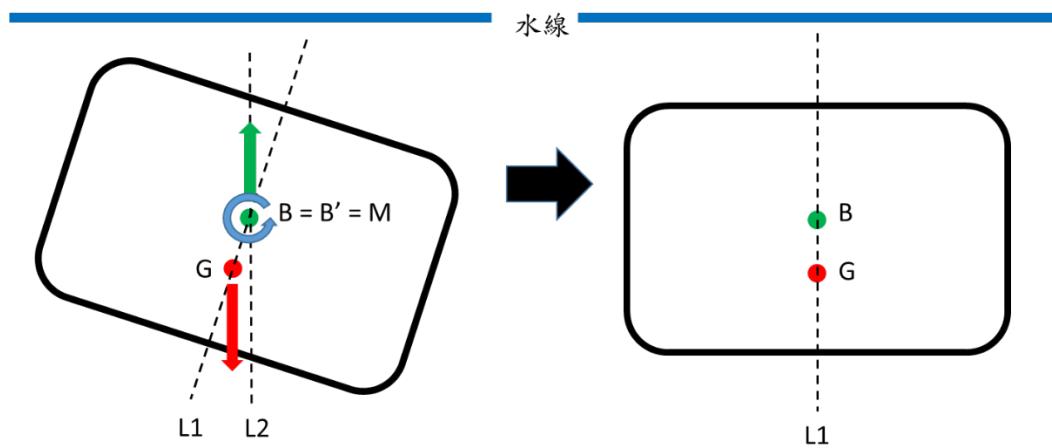


圖 2-10 ROV 水下正穩定力矩

第3章 系統架構

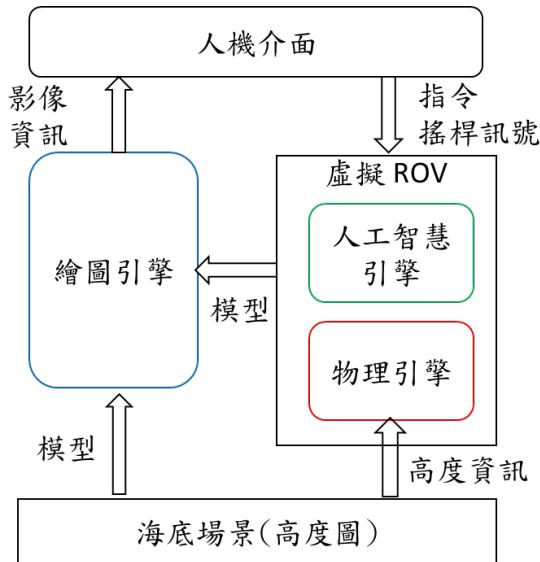


圖 3-1 系統架構

本系統分為六部分，其中包含下面兩種虛擬物體：

一、虛擬 ROV(Virtual ROV)

本研究利用幾何建模工具建造一部虛擬 ROV，做為模擬平台；除了 ROV 的幾何結構與座標，虛擬 ROV 還包含下列參數：ROV 位置、姿態、質量、質心、體積、浮心、慣性張量等物理量；另外還有推進器(Propeller)的施力方向、推進器位置。依據不同 ROV 特性更改上述參數。這些參數都在身體座標系下定義，身體座標系請參考圖 2-3。詳細的虛擬 ROV 架構會在第 3 章的後半段繼續敘述。

二、海底場景(Underwater Scene)

本研究提供兩種海底場景模擬環境，均利用高度圖(Height Map)建立。其一為淺海的沙地地形，水深設定為 10 公尺，其二為深海的岩地地形，水深設定為 50 公尺。兩者均擁有障礙物並根據深度降低可見光亮度，以模擬 ROV 於不同模擬環境中的狀況。

整個模擬器的計算與繪圖模組依功能可分為下面四個子系統：

一、 人工智慧引擎(Artificial Intelligence Engine)

本子系統目的為簡化使用者的操作，透過將使用者輸入的指令、命令或搖桿訊號轉換為實際操作 ROV 的資訊，執行複雜的操作控制。詳細的人工智慧引擎設計與實作會在第 4 章中作介紹。

二、 物理引擎(Physics Engine)

本子系統計算 ROV 所受到的作用力與力矩，並根據運動學公式更新 ROV 的速度與角速度，計算 ROV 的位移與角位移，最後建立轉換矩陣(Transformation Matrix)改變 ROV 的姿態與位置，同時執行碰撞偵測(Collision Detection)。詳細的物理引擎設計與實作會在第 5 章中作介紹。

三、 繪圖引擎(Graphics Engine)

本子系統繪製模擬場景與虛擬 ROV，顯示模擬過程；同時繪製曲線圖顯示各項參數的變化。此子系統為模擬器的核心，提供使用者浸入式(Immersive)的模擬經驗；本模擬器使用第一人稱與第三人稱的鏡頭來繪製模擬過程，同時也會顯示 ROV 各個鏡頭所觀測到的場景，並於多視窗的系統下顯示結果。

四、 人機介面(Human–Computer Interaction)

人機介面主要作為輸入與輸出的功能，本模擬器的輸入裝置含：鍵盤、滑鼠、搖桿與功能選單，讓使用者輸入命令、參數值與功能選項。人機介面的輸出主要為繪圖引擎的輸出畫面、參數視覺化結果與記錄檔。

第3-1節 虛擬 ROV 架構

本研究所參考的 ROV 為 DWTEK ROV, Investigator 90，為台灣的玉豐海洋科儀股份有限公司所製造，其參數透過其網站[13]獲得，並如表 3-1 所述：

表 3-1 ROV Investigator 90 參數

	資訊	被註
尺寸	110(長)x68(寬)x49(高)(公分)	不包含額外裝置
重量(Air)	~132KG	不包含額外裝置
作業深度	500 公尺	1000 公尺(選擇)
動力配置	無刷直流馬達	四顆水平向量&兩顆垂直向量
特殊功能	自動定深、自動定向	
額外裝置	機械手臂	
酬載	27 公斤	



圖 3-2 ROV Investigator 90



圖 3-3 海洋大學系工系 ROV

模擬器內的虛擬 ROV 透過參考並測量本校系工系的 ROV 建立模型，而內部參數包含：

1. 基本物理量：尺寸、質量、質心、體積、浮心、慣性張量。
2. 推進器：推進器位置、推進器角度。
3. 攝影機：攝影機位置、攝影機角度。
4. 機械手臂：機械手臂位置、機械手臂控制角度。

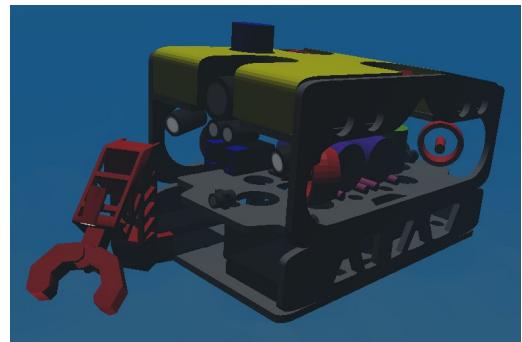


圖 3-4 虛擬 ROV 模型

第3-2節 ROV 的基本物理量

在此節要計算 ROV 的基礎物理量，包括其質量、質心、體積、浮心與慣性張量，為了有根據的計算出 ROV 的基礎物理量，本研究採取一種體素化的方法，透過實際測量並建立多邊形模型，再將其體素化後，依照部件材料給予每個體素不同密度。

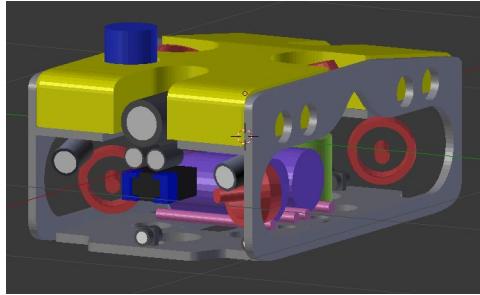


圖 3-5 ROV 多邊形模型

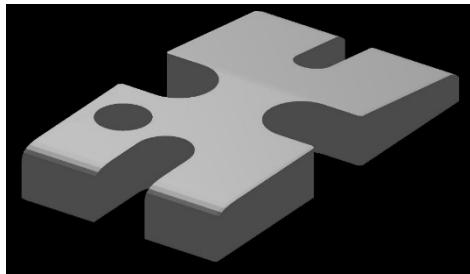


圖 3-6 浮塊多邊型模型

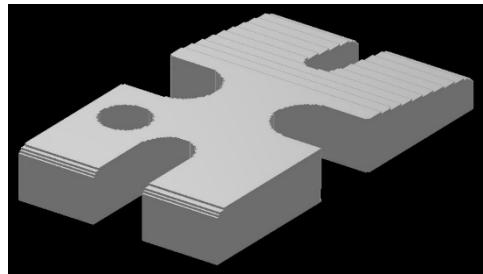


圖 3-7 浮塊體素模型

對所有部件進行體素化並將體素值設定為密度後，再依每個部件其位置，依序將所有部件塞入同一份三為體素資料中，最終得到一份體素資料，每個體素值所代表的即為模型該位置的材質密度。

第3-2-1節 質量、質心、體積與浮心的計算

在完成模型的體素化與整合以後，依序利用以下公式計算其質量M、質心 \bar{G} 、體積V與浮心 \bar{B} ：

$$M = \sum m_i \quad (3-1)$$

$$\bar{G} = \frac{\sum m_i \bar{r}_i}{\sum m_i} \quad (3-2)$$

$$V = \sum v_i \quad (3-3)$$

$$\bar{B} = \frac{\sum v_i \bar{r}_i}{\sum v_i} \quad (3-4)$$

v_i 為體素化時定義的最小體素大小，本研究使用 $0.2^3 cm^3$ 。

$m_i = v_i * \rho$ ， ρ 為體素值。

\bar{r}_i 為該體素的位置向量。

第3-2-2節 慣性張量的計算

慣性張量在旋轉運動中如同線性運動中的質量一般，其代表的是物體的質量分布情形，在經由體素化後，可以清楚的根據體素位置與密度來計算物體的慣性張量，慣性張量 \bar{I} 為一 3×3 矩陣，其中的各個分量計算利用以下公式獲得：

$$\begin{aligned}\bar{I} &= \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (3-5) \\ I_{xx} &= \sum (y_i^2 + z_i^2)m_i, \quad I_{yy} = \sum (x_i^2 + z_i^2)m_i, \quad I_{zz} = \sum (x_i^2 + y_i^2)m_i \\ I_{xy} &= \sum x_i y_i m_i, \quad I_{xz} = \sum x_i z_i m_i, \quad I_{yz} = \sum y_i z_i m_i\end{aligned}$$

第3-2-3節 可變物理量的計算

當機械手臂運動時，ROV 的總質心、總浮心與其慣性張量都會因此而改變，模擬器必須重新更新其數據。在質心與浮心的部分，利用公式(3-2) 質心公式與公式(3-4) 浮心公式來重新計算，此時公式(3-2)的 m_i 為主 ROV 與各個機械手臂的質量，而 \bar{r}_i 則為主 ROV 與機械手臂的於身體座標系的質心位置。浮心的更新也是同樣模式，將公式(3-4)中的 v_i 改為主 ROV 與機械手臂的體積，而 \bar{F}_i 則為其浮心位置。慣性張量因為其計算方式與物體形狀與質量分佈有著密切關係，無法使用平均的方式做更新，根據[1, 11]平行軸定理(Principle of Parallel Axis Theorem)，使用以下公式來計算包含機械手臂的慣性張量：

$$\bar{I} = \bar{I}_{rov} + \sum_{i=0}^n (\bar{I}_i + m_i * \bar{d}^2) \quad (3-6)$$

本研究將 ROV 機械手臂分 6 節以建立模型，並有各自的慣性張量，因此 n 為 6

\bar{I}_i 為每一節機械手臂依其運動座標系所計算出來的慣性張量

m_i 為每一節機械手臂的質量

\bar{d} 為 \bar{I}_i 與 \bar{I}_{rov} 之間的位移量矩陣

第3-3節 ROV 的推進器

此研究參考的 ROV Investigator 90 擁有六個推進器，其中四個推進器負責 ROV 水平方向出力，而另外兩個主要負責鉛直方向。根據推進器的位置分別給予推進器不同的代號，推進器依其位於 ROV 位置先分為左舷 Port side (P) 與右舷 Starboard side (S)，再分為前置 Front (F)、後置 After (A) 與垂直 Vertical (V)；例如位於 ROV 左舷並靠前的推進器則稱其為 PV。

第3-3-1節 推進器的位置

推進器位置以 ROV 的身體座標系定義，記為 $\bar{P}_{propeller}$ 。推進器的位置可參考圖 3-8，數值如表 3-2。

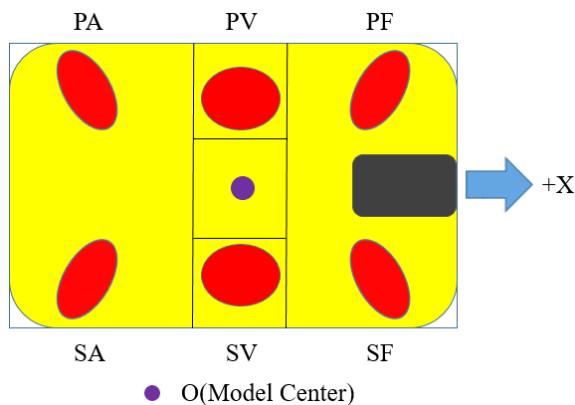


表 3-2 推進器的位置

\bar{P}_{PF}	$< 0.375, -0.085, -0.2 >$
\bar{P}_{PV}	$< 0.0, 0.16, -0.2 >$
\bar{P}_{PA}	$< -0.375, -0.085, -0.2 >$
\bar{P}_{SF}	$< 0.375, -0.085, 0.2 >$
\bar{P}_{SV}	$< 0.0, 0.16, 0.2 >$
\bar{P}_{SA}	$< -0.375, -0.085, 0.2 >$

單位：公尺

圖 3-8 推進器的位置

第3-3-2節 推進器的推力方向

推進器的推力方向記為 $\vec{f}_{propeller}$ ，其方向定義於身體座標系，同時也平行於運動座標系。水平的四個推進器之中，前置的兩個推進器以面向正 X_m 軸內傾 30 度，而後置的兩個推進器則為面向正 X_m 軸外傾 30 度。位於 ROV 中間的兩個垂直推進器則面相正 Y_m 軸內傾 15 度。推進器方向可參考圖 3-9，其數值如表 3-3。

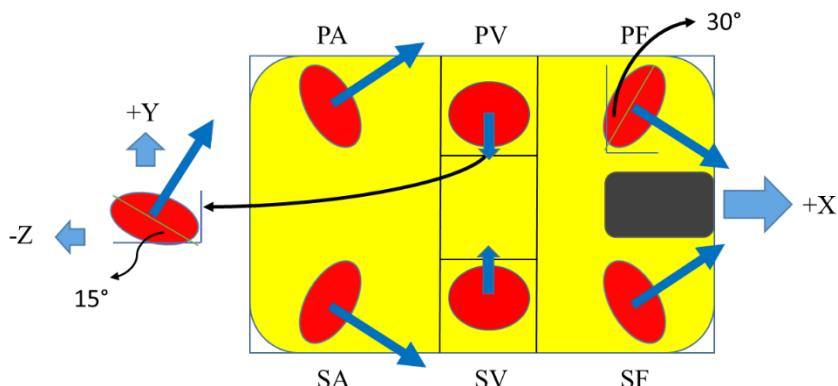


表 3-3 推進器的施力方向

\vec{f}_{PF}	$< 0.866, 0.0, -0.5 >$
\vec{f}_{PV}	$< 0.0, -0.966, 0.258 >$
\vec{f}_{PA}	$< 0.866, 0.0, 0.5 >$
\vec{f}_{SF}	$< 0.866, 0.0, 0.5 >$
\vec{f}_{SV}	$< 0.0, -0.966, -0.258 >$
\vec{f}_{SA}	$< 0.866, 0.0, -0.5 >$

單位：公尺

圖 3-9 推進器的施力方向

第4章 人工智慧引擎

人工智能引擎在本研究內扮演的角色為將使用者下達的指令，譬如圖 4-1 內的自動定深(Auto Depth)功能，另外還有像自動定向(Auto Heading)的功能；經由人工智能引擎，轉換為系統實際運作的操作參數，而為了模擬船長直接下達命令與實際操作員操作 ROV 的兩種不同操作模式，提供了兩種控制模式：指令模式(Command Mode)與搖桿模式(Joystick Mode)。為了實現此功能，人工智能引擎利用「PID 控制器」與「推進器向量」來進行命令的轉換。



圖 4-1 ROV 控制器

PID 控制器是一個在工業控制應用中常見的反饋迴路部件。如圖 4-2，整個反饋迴路的流程，首先把收集到的數據和一個目標值(Target Values)進行比較，然後把這個差距(Errors)透過 PID 控制器計算出控制訊號，並將控制訊號輸入相應的處理模組(Actuator)執行後，會使整個系統趨近並保持於目標值，而本模擬器的處理模組即為物理引擎。

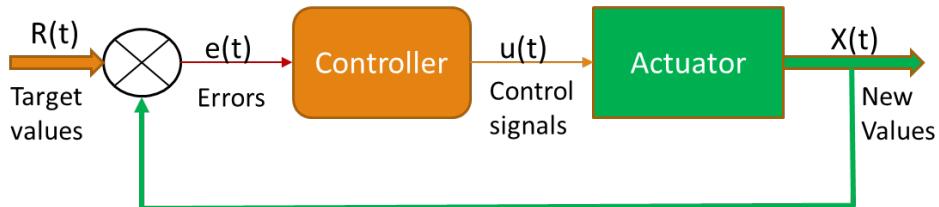


圖 4-2 PID 反饋迴路

PID 控制器的全名為 Proportional – Integral – Derivative controller (PID Controller)，是利用由比例單元(P)、積分單元(I)和微分單元(D)組成，而其方程式可以表示成：

$$u(t) = K_p e(t) + K_i \int_{t-100}^t e(s) ds + K_d \frac{d}{dt} e(t) \quad (4-1)$$

將其離散化後則為：

$$u(t) = K_p e[t] + K_i \sum_{n=t-100}^t e[n] + K_d (e[t] - e[t-1])/t \quad (4-2)$$

$K_p e[t]$ ：比例控制考慮的是當前的誤差。

$K_i \sum_{n=t-100}^t e[n]$ ：一定時間內的誤差值總和，本模擬器每 0.01 秒紀錄一次誤差，故積分項考慮會慮過去 1 秒內的誤差總和。

$K_d (e[t] - e[t-1])/t$ ：最近兩次誤差的一階導數(斜率)。

第4-1-1節 PID 控制器的運用

在 PID 的運用上，本模擬器提供 12 個可控制參數，如下列項目：

- a. ROV 的位置，含三個座標(P_x, P_y, P_z)：

ROV 於世界座標系的位置，將其轉換至運動座標系，並計算需透過第 2-3-1 節小節所提之線性運動以達到目的地。

- b. ROV 的姿態，含三個角度(ψ, θ, ϕ)：

ROV 的歐拉角，分別為第 2-3-2 節所提之側傾角、縱搖角與橫搖角，利用此來計算旋轉運動大小。

- c. ROV 的線性速度，含三個分量(V_x, V_y, V_z)：

ROV 於世界座標系內的線性速度，將其轉為運動座標系方向的速度，再計算相對應的線性運動大小。

- d. ROV 的角速度，含三個角速率($\omega_x, \omega_y, \omega_z$)：

利用 ROV 的三個角速率來計算相對應的旋轉運動大小。

上述 ROV 的位置與姿態，透過人機介面，直接輸入目標值進行控制；而速度與角速度則以搖桿模式進行輸入。

利用 12 個基本參數組合出高階的指令，譬如：

深度保持(Auto Depth)

位置 (free, Target Depth, free)，x 與 z 座標不變，y 座標設定為目標值。

線性速度 (free, 0, free)，此時 ROV 的垂直方向速度為零。

必須同時控制 ROV 於世界座標系的深度值與其於 Y_w 軸的速度，而其他不參與的參數可以忽視。

在確定好目標值以後，經由 PID 控制器的運算，得到的控制訊號 $u(t)$ ，要將控制訊號再轉換為 ROV 推進器所出的力，即為下一節所提的推進器向量。

第4-2節 推進器向量

為了將 ROV 推進器的力做一個系統性的整合，本研究利用推進器向量來統合 ROV 上的六個推進器，推進器向量計做 \vec{PV} ，其每個分量代表的是每一個推進器所需出力的大小，正負號代表的是推進器正槳或者是逆槳，透過推進器向量可以直接表明進行運動時每個推進器所需出的力。

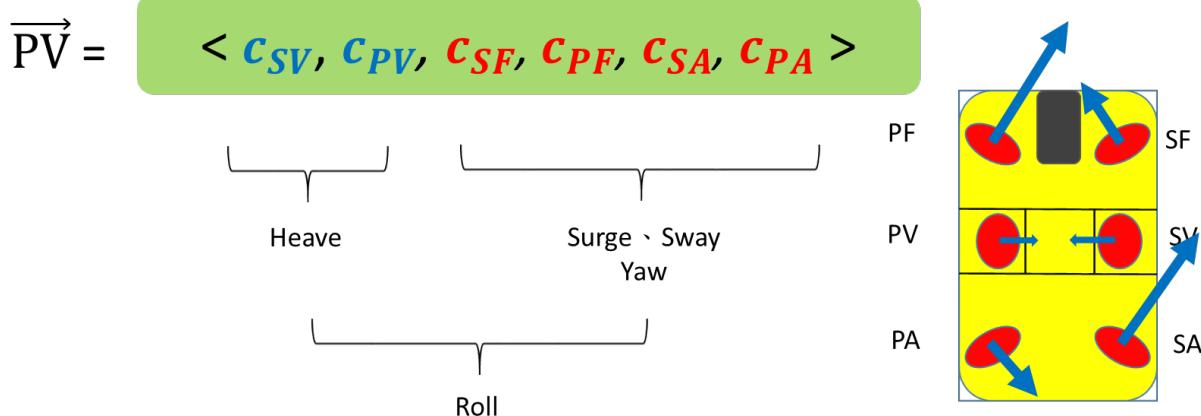


圖 4-3 推進器向量

第4-2-1節 推進器向量的運用

若質心位於 ROV 的中心，可以由圖 4-3 推進器向量知道，起伏只靠 SV 與 PV 兩個推進器就能完成，而縱移、橫移與橫搖只靠 SF、PF、SA 與 PA 推進器來完成，側傾方面因為需要 PV 與 SV 兩推進器依一正槳一逆槳的方法來施力，但此時也會產生橫移方面的推力，因此需要加入 SF、PF、SA 與 PA 推進器來抵銷橫移方面的推力；最後縱搖在質心位於中心的情況下，推進器的相對位置無法產生相對的力矩。如表 4-1，為質心於物體中央時六軸運動對應的推進器向量值。

表 4-1 六軸運動於運動座標系的推進器向量範例(質心於物體中央)

	c_{SV}	c_{PV}	c_{SF}	c_{PF}	c_{SA}	c_{PA}
\vec{PV}_{Heave}	1	1	0	0	0	0
\vec{PV}_{Surge}	0	0	1	1	1	1
\vec{PV}_{Sway}	0	0	-1	1	1	-1
\vec{PV}_{Yaw}	0	0	1	-1	1	-1
\vec{PV}_{Roll}	1	-1	0.259	-0.259	-0.259	0.259
\vec{PV}_{Pitch}	NULL	NULL	NULL	NULL	NULL	NULL

第4-2-2節 計算推進器向量

在經過 PID 控制器的運算後，可以獲得與目標值的差距值 $u(t)$ ，利用此一差距作為 ROV 的目標運動，並根據本身推進器的位置 \bar{P} 與 ROV 的質心位置 \bar{G} 來計算每個推進器所需出力與力矩，其計算出的推進器向量則為此目標運動每個推進器所需出的力，本研究透過建立一線性系統，並利用高斯消去法(Gaussian Elimination)來解決這個問題。

$$\sum_{i=0}^5 c_i * \vec{f}_i = \vec{f}_{motion} \quad (4-3)$$

$$\sum_{i=0}^5 c_i * \vec{\tau}_i = \vec{\tau}_{motion} \quad (4-4)$$

以於運動座標系內的起伏為例，起伏於運動座標系內的施力 \vec{f}_{heave} 為 $<0, 1, 0>$ ，而其力矩 $\vec{\tau}_{heave}$ 為 $<0, 0, 0>$ ，因此系統表示成：

$$\begin{bmatrix} f_{0x} & f_{0x} & f_{0x} & f_{3x} & f_{4x} & f_{5x} \\ f_{0y} & f_{0x} & f_{0x} & f_{3y} & f_{4y} & f_{5y} \\ f_{0z} & f_{1z} & f_{2z} & f_{3z} & f_{4z} & f_{5z} \\ \tau_{0x} & \tau_{1x} & \tau_{2x} & \tau_{3x} & \tau_{4x} & \tau_{5x} \\ \tau_{0y} & \tau_{1y} & \tau_{2y} & \tau_{3y} & \tau_{4y} & \tau_{5y} \\ \tau_{0z} & \tau_{1z} & \tau_{2z} & \tau_{3z} & \tau_{4z} & \tau_{5z} \end{bmatrix} * \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

f_i 為表 3-3 推進器的施力方向

$$\tau_i = r_i \times f_i$$

$$r_i = (\bar{P}_i - \bar{G})$$

\bar{P}_i 為表 3-2 推進器的位置

\bar{G} 為第 3-2-1 節質心位置

在解此線性系統時，會發現系統為線性相依，原因是 ROV 的六個推進器彼此對稱，因此，本研究的解決方案是在其中幾個推進器的位置加上很小的誤差，以此來讓此線性系統是可以解的。

第5章 物理引擎

第5-1節 物理引擎流程

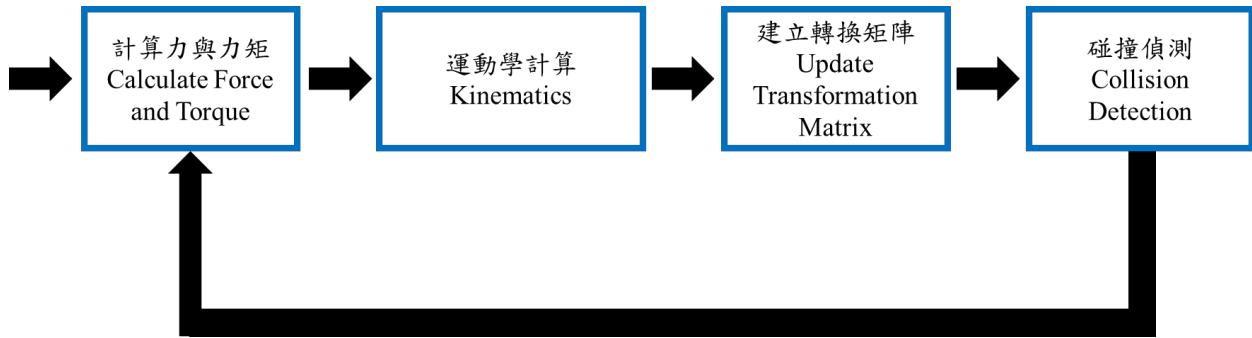


圖 5-1 物理引擎流程

在物理引擎的第一部分為計算 ROV 所受到的總施力與總力矩，在這部分又可以根據來源分為兩類：

一、水下靜力：

水下靜力主要是由物體的重力與浮力所決定，透過第 3-2 節的方法對 ROV 模型體速化後計算 ROV 的質量、質心、體積與浮心，並透過這些參數轉換為其所受重力 \vec{f}_w 、浮力 \vec{f}_b 、重力力矩 $\vec{\tau}_w$ 與浮力力矩 $\vec{\tau}_b$ 。

二、水下動力：

水下動力又可分為兩部分，分別為透過 ROV 的推進器所產生的推力 f_p 與由線性運動與旋轉運動所產生的阻力 \vec{f}_d (Drag force)、 \vec{f}_r (Resistance force)。

物理引擎的第二部分為利用運動學公式如牛頓第二定律(Newton's Second Law)與歐拉第二定律(Euler's Second Law)來更新 ROV 的速度與角速度並計算其單位時間的位移量(Displacement)與角位移量(Angular Displacement)。

在物理引擎的最後利用計算出的位移量與角位移量，計算並更新轉換矩陣，使 ROV 運動。

第5-2節 計算力與力矩

第5-2-1節 重力與浮力

利用第3-2-1節計算的ROV基本物理量，計算ROV所受到的浮力與重力。

$$\vec{f}_g = M * g * \vec{n} \quad (5-1)$$

$$\vec{f}_b = V * g * \vec{n} \quad (5-2)$$

M 與 V 分別為第3-2-1節計算出的質量與體積

$$g = 9.8(m/s^2)$$

\vec{n} 為力在世界座標系內的方向，重力的方向恆朝向 $-Y_w$ 軸，而浮力的方向恆朝向 $+Y_w$ 軸

利用所得浮力計算浮力力矩，而重力作用於質量中心，因此其力矩為0。

$$\vec{\tau}_b = (\bar{B} - \bar{G}) \times \vec{f}_b \quad (5-3)$$

\bar{B} 與 \bar{G} 為第3-2-1節計算的浮心與質心

第5-2-2節 推進器推力

在計算推進器推力的部分，利用人工智慧引擎計算出的推進器向量 \overrightarrow{PV} 來統合推進器的總施力與總力矩。

推進器的總施力：

$$\vec{f}_p = \sum_{n=0}^5 c_i * \vec{f}_i \quad (5-4)$$

推進器的總力矩：

$$\vec{\tau}_p = \sum_{n=0}^5 \bar{r}_i \times (c_i * \vec{f}_i) \quad (5-5)$$

c_i 為推進器向量 \overrightarrow{PV} 的分量

\vec{f}_i 為表3-3推進器的施力方向

第5-2-3節 阻力

阻力是物體於流體中相對運動所產生與運動方向相反的力，在[4]這本書的 Skin Friction Drag 章節內有敘述，阻力一般可以分為形狀阻力(form drag)與摩擦阻力(skin friction)，形狀阻力的大小取決於物體線性運動方向的切面面積，而摩擦阻力則是因為流體延物體表面行徑時所產生的摩擦力，因此與物體平行線性運動的表面積大小有關，將切面面積除以表面面積得到一個比例稱其為縱橫比(aspect ratio)，由圖 5-2 可以得知，隨著縱橫比越大形狀阻力減少而摩擦阻力則會增大。

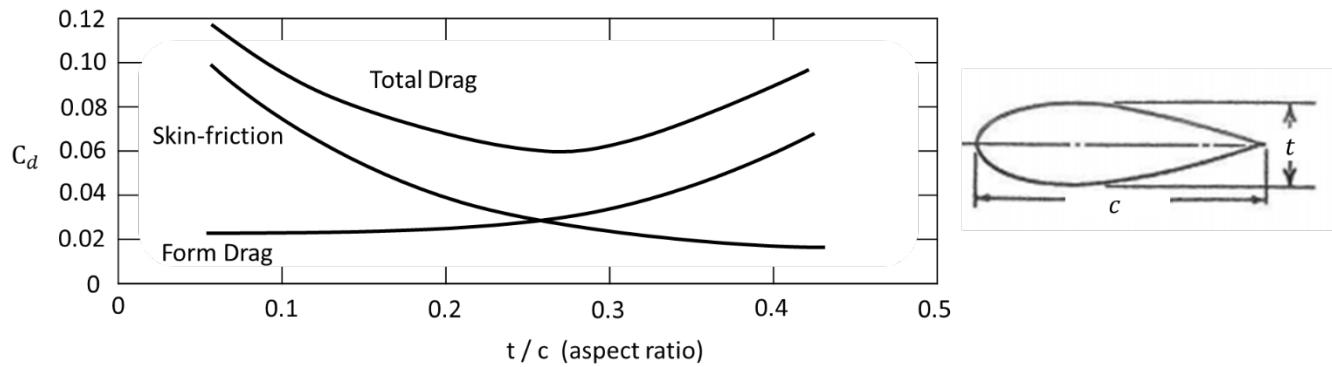


圖 5-2 形狀阻力與摩擦阻力

根據物體的縱橫比，可以利用一個無因次的阻力係數來統合描述其兩種摩擦力，如表 5-1 所列出一些基本形狀其阻力係數，更詳細的也可以參考[4]內的 Introduction of Drag 章節。配合阻力係數，利用阻力方程式來計算物體所受阻力，其大小約和物體的相對速度平方成正比，而此篇論文將阻力又分為線性運動的阻力(Drag force)與旋轉運動的阻力(Resistance force)。

表 5-1 阻力係數

名稱	Sphere	Half-sphere	Cone	Cube	Angled Cube
形狀					
C_d	0.47	0.42	0.50	1.05	0.80
名稱	Long Cylinder	Short Cylinder	Streamlined Body	Streamlined Half-body	
形狀					
C_d	0.82	1.15	0.04	0.09	

一、線性阻力：

線性阻力只針對 ROV 的線性運動，因此其作用點為 ROV 的質心，施力方向則與線性速度反向，並不會因 ROV 的旋轉而改變其大小與方向，也不會產生力矩使 ROV 旋轉。

線性阻力利用以下公式：

$$\vec{f}_a = -0.5 * C_d * \rho * A * \vec{v}^2 \quad (5-6)$$

$C_d=1.05$ ，參考表 5-1 阻力係數。

$$A = \langle A_{front}, A_{top}, A_{side} \rangle * \|v\|$$

A 為垂直運動方向的切面面積，因此透過計算 ROV 三個平面的面積，再以速度的方向來做為權重。

$\rho = 1.025$ ，為海水密度

\vec{v} 為物體的線性運動速度

二、旋轉阻力：

旋轉阻力是計算 ROV 旋轉時所產生的抵抗力矩，因次必須點出旋轉阻力在 ROV 上的作用點與作用力方向，透過將 ROV 其角速度轉換至六軸的切線速度來計算阻力施力方向與施力點，即可以利用類似於線性阻力的方式計算阻力大小。

帶入線性阻力的計算方式，唯有將 \vec{v} 改為以 ROV 六軸的切線速度 \vec{v}_\perp 。如圖 5-3。

計算公式為：

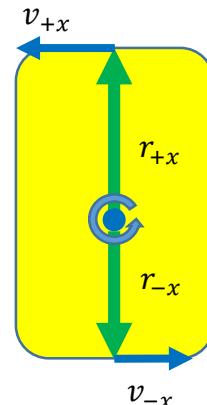


圖 5-3 切線速度

$$\vec{f}_r = -0.5 * C_f * \rho * A * \vec{v}_\perp^2 \quad (5-7)$$

$$\vec{v}_\perp = \vec{\omega} \times \vec{r} \quad (5-8)$$

$C_f = 1.05$

A 則為垂直 \vec{v}_\perp 方向的切面面積，而因切線速度非延單一軸速度，因此也採取將速度做為權重的方式來進行計算。

旋轉阻力產生的力矩：

$$\vec{\tau}_r = \vec{r} \times \vec{f}_r \quad (5-9)$$

\vec{r} 為以 ROV 的質心向六軸 $\pm x, \pm y, \pm z$ 延伸之向量

第5-3節 運動學計算(Kinematics)

將計算獲得的水下靜力、水下動力做一個統合，並依此為總受力、總力矩，來更新ROV下一時間的速度、位置、角度與姿態。

首先，必須先定義兩種運動其座標系統。線性運動因其速度為世界座標系統的速度，水下靜力的重力與浮力也是垂直於世界座標系的 Y_w 軸，因此本研究統一將線性運動於世界座標系做計算。而旋轉運動因其慣性張量是根據其運動座標系所計算得出，而慣性張量為一個三成三的矩陣，在做運算時非常耗時，若轉換至世界座標系統需要每一個時間點都做轉換，因此旋轉運動以運動座標系為其座標系統。

第5-3-1節 線性運動

計算總受力，包含浮力 \vec{f}_b 、重力 \vec{f}_g 、推進器推力 \vec{f}_p 、因為線性速度產生的阻力 \vec{f}_d 與旋轉速度產生的阻力 \vec{f}_r ：

$$\vec{f} = \vec{f}_b + \vec{f}_g + \vec{f}_p + \vec{f}_d + \vec{f}_r \quad (5-10)$$

利用牛頓第二運動定律求得瞬時加速度：

$$\vec{a} = \vec{f}/M \quad (5-11)$$

用瞬時加速度更新下一時間的速度：

$$\vec{v}(t+1) = \vec{v}(t) + \vec{a} * \Delta t \quad (5-12)$$

計算時間 t 內的位移量：

$$\Delta\vec{x} = \vec{v}(t+1) * \Delta t \quad (5-13)$$

利用位移量 $\Delta\vec{x}$ 去更新第 5-4 節轉換矩陣中的平移矩陣(Translate Matrix)

第5-3-2節 旋轉運動

計算總力矩，包含浮力力矩 $\vec{\tau}_b$ 、推進器施力力矩 $\vec{\tau}_p$ 與旋轉阻力力矩 $\vec{\tau}_r$ ：

$$\vec{\tau} = \vec{\tau}_b + \vec{\tau}_p + \vec{\tau}_r \quad (5-14)$$

利用歐拉第二定律計算瞬時角加速度：

$$\vec{\alpha} = \bar{I}^{-1} * \{\vec{\tau} - \vec{\omega}(t) \times [\bar{I} * \vec{\omega}(t)]\} \quad (5-15)$$

\bar{I} 即為第3-2-2節所計算的慣性張量。

用瞬時角加速度更新下一時間的角速度：

$$\vec{\omega}(t+1) = \vec{\omega}(t) + \vec{\alpha} * \Delta t \quad (5-16)$$

計算時間t內的角位移量：

$$\Delta\vec{r} = \vec{\omega}(t+1) * \Delta t \quad (5-17)$$

利用角位移 $\Delta\vec{r}$ 去更新第5-4節轉換矩陣中的旋轉矩陣(Rotation Matrix)

第5-4節 轉換矩陣(Transformation Matrix)的建立

在物理引擎的最後一步為建構 ROV 的轉換矩陣，轉換矩陣同時也為繪圖引擎內的模型矩陣(Model Matrix)，其目的是將物體由該物體的身體座標系，轉換至其位於世界座標器的位置與姿態；隨著轉換矩陣的改動，ROV 即可表現出動畫。因此，利用第 5-3 節所計算出的線性位移與角位移來建立平移矩陣 T_b 與旋轉矩陣 R_b ，最後再將其合併為轉換矩陣 M_b 。

如第 5-3 節中所提，ROV 的線性運動是在世界座標系所計算，其計算出的線性位移代表著質心於世界座標系的位移量，而此處所需計算的平移矩陣代表的是身體座標系的原點於世界座標系的位移量，因此必須經過轉換才能建立平移矩陣。另外旋轉運動的計算是在運動座標系內計算，因此也必須經過轉換來獲得身體座標系的旋轉矩陣。

第5-4-1節 線性運動的影響

在線性系統下所計算出來的位移量 $\Delta\vec{x}(d_x, d_y, d_z)$ ，當不考慮旋轉時，其不僅是質心的位移量，也是身體座標系原點的位移量，如圖 5-4 中綠色線段部分，因此可以直接將 $\Delta\vec{x}$ 建立成平移矩陣 T_l 。

$$T_l = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5-18)$$

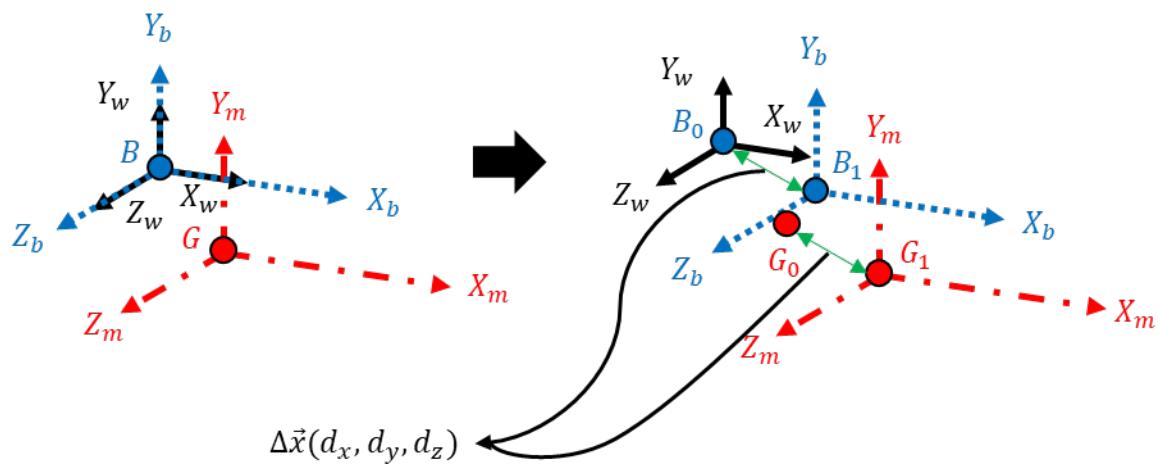


圖 5-4 身體原點的線性位移

第5-4-2節 旋轉運動的影響

透過在旋轉運動下所計算出來的 $\Delta\vec{r}(\psi, \theta, \phi)$ ，在假設其角度很小的前提下，旋轉運動具交換律，因此可以利用通解產生一新的旋轉矩陣 R_n 。

$$R_n = R_z(\phi)R_y(\theta)R_x(\psi) = \begin{bmatrix} 1 & -\phi & \theta & 0 \\ \phi & 1 & -\psi & 0 \\ \theta & \psi & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5-19)$$

R_n 為在第 t+1 時間點於運動座標系的旋轉量，因旋轉矩陣代表的是該空間三軸所指的方向，透過圖 5-5 可以知道，身體座標系與運動座標系的方向是一致的，因此 R_n 也理解為第 t+1 時間點身體座標系因旋轉量 $\Delta\vec{r}$ 所產生的矩陣。

在旋轉運動時，物體依質心為旋轉中心做旋轉，因此身體座標系的原點 B_0 也會因旋轉而產生位移 $\Delta\vec{x}_r(d_{rx}, d_{ry}, d_{rz})$ ，如圖 5-5 中綠色線段部分，而利用其位移建立的矩陣稱為 T_r 。

$$\Delta\vec{x}_r = B_1 - B_0 = R_n * B_0 - B_0 \quad (5-20)$$

$$T_r = \begin{bmatrix} 1 & 0 & 0 & d_{rx} \\ 0 & 1 & 0 & d_{ry} \\ 0 & 0 & 1 & d_{rz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5-21)$$

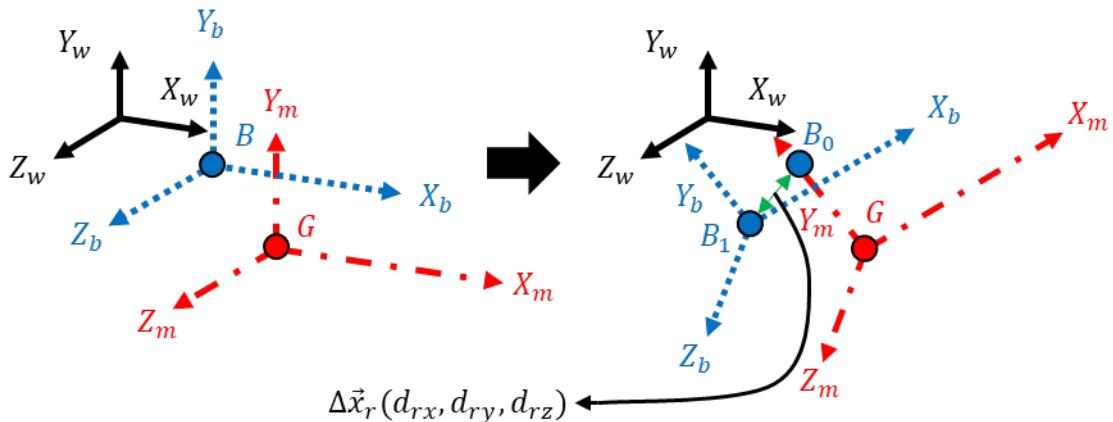


圖 5-5 身體原點的旋轉位移

第5-4-3節 平移矩陣 T_b

T_b 指的是任意時間點由身體座標系轉換至世界座標系的平移矩陣，會受到兩種運動影響，線性運動所產生的位移 T_l 與旋轉運動使身體座標系原點 B_0 旋轉而產生的位移 T_r 。因此，在第 $t+1$ 時間點的位移量所形成的矩陣稱其為 T_n ，而其值為 T_l 與 T_r 相乘。在時間點 $t+1$ 的平移矩陣 T_{t+1} ，則為時間點 t 的平移矩陣 $T(t)$ 乘以第 $t+1$ 時間點的位移量矩陣 T_n 。

$$T_n = T_l * T_r \quad (5-22)$$

$$T_b = T(t + 1) = T(t) * T_n \quad (5-23)$$

第5-4-4節 旋轉矩陣 R_b

R_b 指的是任意時間點由身體座標系轉換至世界座標系的旋轉矩陣，也代表該時間點，身體座標系三軸的方向。

經由第 $t+1$ 時間點的旋轉量形成的 R_n 旋轉矩陣，將其與時間點 t 的旋轉矩陣 R_t 相乘，就可以得到時間點 $t+1$ 的旋轉矩陣 R_b 。

$$R_b = R(t + 1) = R(t) * R_n \quad (5-24)$$

第5-4-5節 轉換矩陣 M_b

在計算出身體座標系原點於第 $t+1$ 時間點的總位移，與身體座標系的總旋轉後，將其結合即可以獲得第 $t+1$ 時間點的轉換矩陣 M_b 。

$$M_b = T_b * R_b \quad (5-25)$$

第6章 繪圖引擎與人機介面

在本章為描述將物理引擎計算過後的資訊，透過繪圖引擎來使 ROV 模型運動，以及將內部的物理量與運動參數顯示於螢幕上。如圖 6-1。



圖 6-1 模擬器主畫面

本章會講解幾個主要的實作模組，包含：虛擬實境畫面、羅盤 ROV 歐拉角的計算與操作介面。操作介面可分為指令模式(Command Mode)與搖桿模式(Joystick Mode)；其中指令模式透過位於主畫面上的資訊(Information)與功能(Function)欄位來進行操作。

第6-1節 虛擬實境(Virtual Reality)

本模擬器提供使用者虛擬實境的功能，使用者可以利用提供的各種鏡頭，觀看 ROV 的運動模擬，也可以利用這些鏡頭，進行教學的使用。

本模擬器提供的鏡頭種類：

1. ROV 內建鏡頭

本研究參考的 ROV 擁有 4 個鏡頭，包含位於 ROV 正上方的主鏡頭與同樣朝向 ROV 正面，分別置於 ROV 左右兩側的鏡頭；此三鏡頭功能目的除了顯示 ROV 的模擬環境之外，也是為了捕捉 ROV 機械手臂的位置，讓使用者能夠知道其操作的機械手臂運作狀況。最後 ROV 還有一顆鏡頭朝向正後方，並配備一顆 LED 探照燈。如圖 6-2。

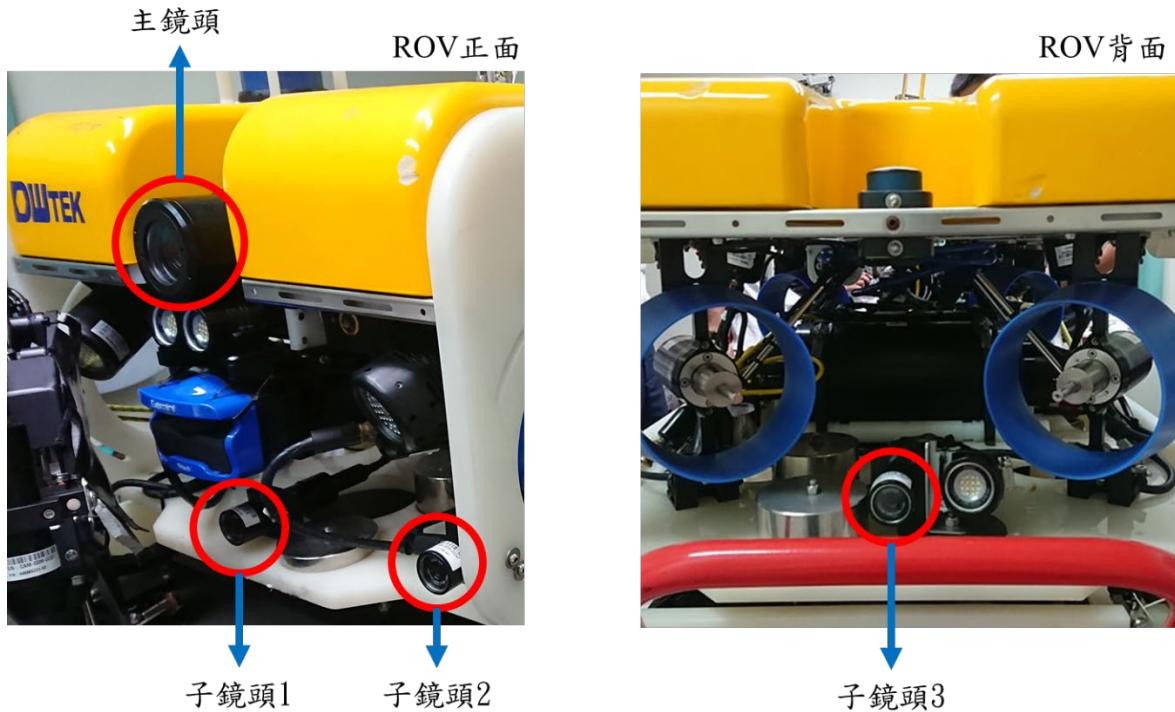


圖 6-2 ROV 攝影機配置

2. 六軸平行投影鏡頭

提供給使用者用於測量，配合背景為黑白方格的測量環境，讓使用者更清楚 ROV 運動的模擬狀況。如圖 6-3。在第 7 章的實作測量，就有使用到此功能。



圖 6-3 六軸平行投影鏡頭

3. 自由鏡頭

提供使用者透過操作滑鼠與鍵盤直接操控攝影機，在模擬環境中自由穿梭，以第三人稱模式觀看 ROV 的運動模擬。其情景如圖 6-4。



圖 6-4 第三人稱自由鏡頭

第6-2節 ROV 歐拉角的視覺化與計算

在圖 6-1 中位於主畫面右側的羅盤，為 ROV 的歐拉角資訊，如圖 6-5。由左至右分別為橫搖角(Yaw Angle)、側傾角(Roll Angle)與縱搖角(Pitch Angle)。

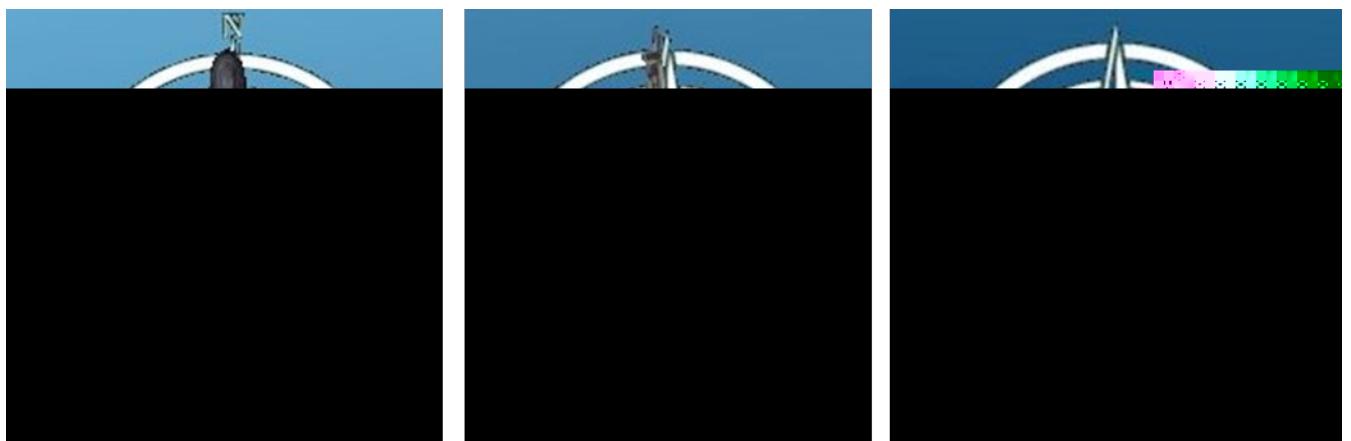


圖 6-5 ROV 歐拉角羅盤

ROV 的歐拉角參數可以由旋轉矩陣 R_b 中取得，此處的旋轉矩陣不同於公式((5-19)的旋轉矩陣 R_n ， R_n 的旋轉矩陣因三個歐拉角為小角度所以可以進行化簡，此處的旋轉矩陣雖同樣由 (ψ, θ, ϕ) 三個歐拉角組成，但三個歐拉角為累積起來的大角度，因此不可化簡。

R_b 表示為下述：

$$R_n = R_z(\phi)R_y(\theta)R_x(\psi) \\ = \begin{bmatrix} \cos \theta \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi & 0 \\ \cos \theta \sin \phi & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & 0 \\ -\sin \theta & \sin \psi \cos \theta & \cos \psi \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6-1)$$

透過公式直接由旋轉矩陣 R_b 取得歐拉角，譬如透過公式(6-1)(5-25 中

$$R_{31} = -\sin \theta$$

經由計算獲得其中一角

$$\theta = -\sin^{-1} R_{31}$$

但要注意的是，因為 $\sin \theta = \sin(\pi - \theta)$ ，所以無法從 R_{31} 獲得一個有方向性的角度。

因此 θ 擁有兩種可能：

$$1. \theta_1 = -\sin^{-1} R_{31}$$

$$2. \theta_2 = \pi - \theta_1 = \pi + \sin^{-1} R_{31}$$

同時，因為 θ 的關係， ψ 也會受到影響：

$$\psi = \tan^{-1} \left(\frac{\frac{R_{32}}{\cos \theta}}{\frac{R_{33}}{\cos \theta}} \right)$$

更重要的是，若當 $\cos \theta = 0$ ，會有很多種狀況需要考慮，因此本研究不使用此種方法來計算歐拉角。

在此研究，本研究透過單一軸只會受到兩種選轉影響的特性，來求得歐拉角，由第 5-4 節可以知道 R_b 所代表的即為當前物體的三軸方向，而每一軸會被兩種旋轉所影響，因此只需要將其中幾軸去除掉不需要的角度，就能從中解析出歐拉角的一角。

範例：橫搖角(Yaw)

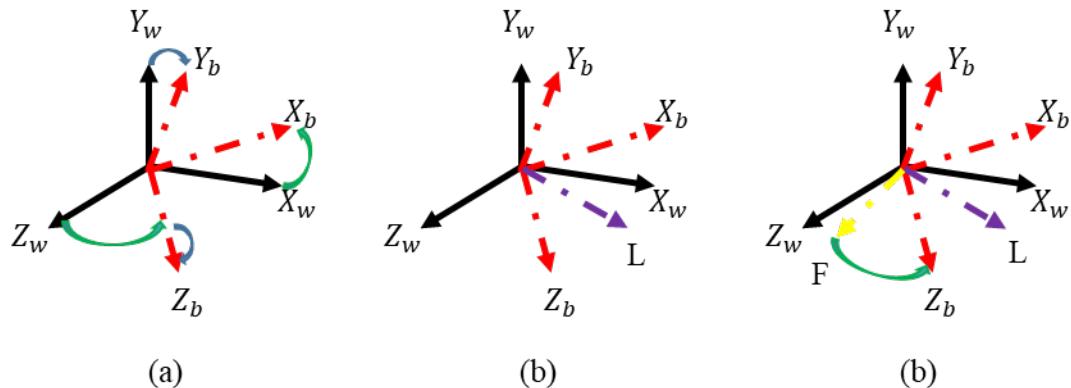


圖 6-6 歐拉角的計算-橫搖角

圖 6-6 中

- (a) 黑色三軸為世界座標系，紅色三軸為身體座標系，身體座標系是從世界座標系經由橫搖一圖中綠色線段與縱擺一圖中藍色線段達成。
- (b) 利用 Y_b 軸外積 Z_w 軸獲得圖中紫色線段 L 軸。
- (c) 將 L 軸外積 Y_b 軸獲得圖中黃色線段 F 軸，而 F 軸與 Z_b 軸之間的角度即為橫搖角。

第6-3節 操作介面

第6-3-1節 指令模式

指令模式利用模擬器主畫面上的資訊欄(Information)與功能欄(Function)進行操作；位於模擬器主畫面下方的資訊欄，會以數值與曲線的方式顯示 ROV 目前的相關參數，包含 ROV 的位置、姿態、線性速度與角速度，同時將一時段內的資訊利用曲線顯示給使用者。



圖 6-7 資訊(Information)欄

自動操控的輸入端也位於資訊欄，可以輸入目標值(Target Value)包含 ROV 的目的地(Position)、目的姿態(Orientation)。圖 6-7 中白色線段為基準值 0，而紅色線段即為設定的目標值，綠色曲線為參數收斂的過程。

位於模擬器主畫面右側的功能欄，如圖 6-8，可以透過此處以直接操作 ROV 的六軸運動並操作 ROV 功能的開關與切換，包含指令模式與搖桿模式之間的切換，機械手臂的第二控制搖桿開啟與關閉，或是 ROV 的包圍體(Bounding Volume)與行駛路徑(Path)的顯示與隱藏。

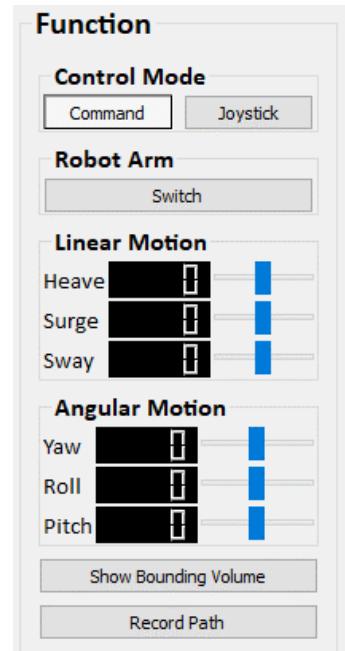


圖 6-8 功能(Function)欄

第6-3-2節 搖桿模式

本模擬器使用兩隻搖桿，均使用 Trustmaster T.16000M FCS HOTAS 控制器，第一支搖桿主要為控制 ROV 的六軸運動與自動控制系統，第二支搖桿為負責操作機械手臂。第一搖桿將縱移(Surge)、橫移(Sway)、起伏(Heave)與橫搖(Yaw)分別部屬於搖桿的三個自由度與一個滑塊(Slider)，並且將自動操作功能如自動定深(Auto Depth)、自動定向(Auto Heading)設置於搖桿按鈕。如圖 6-9 所示。

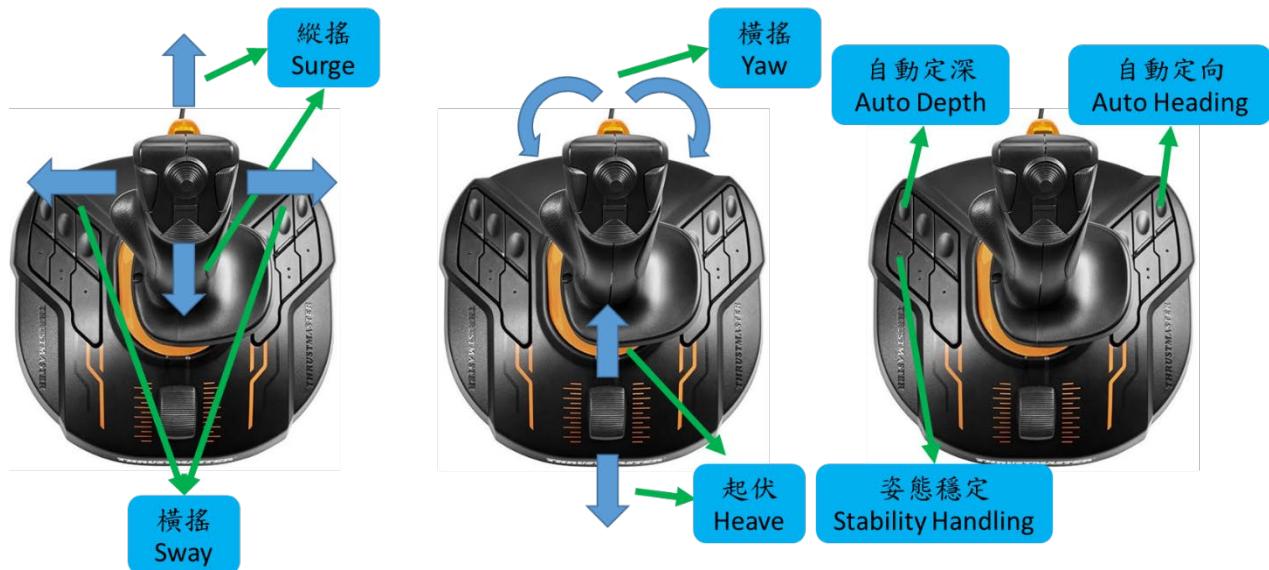


圖 6-9 第一搖桿-ROV 控制搖桿

第二搖桿負責操作機械手臂的運動，本模擬器將 ROV 的機械手臂分為 6 節，每一機械手臂均有一個自由度可以操作，如圖 6-10 與圖 6-11。

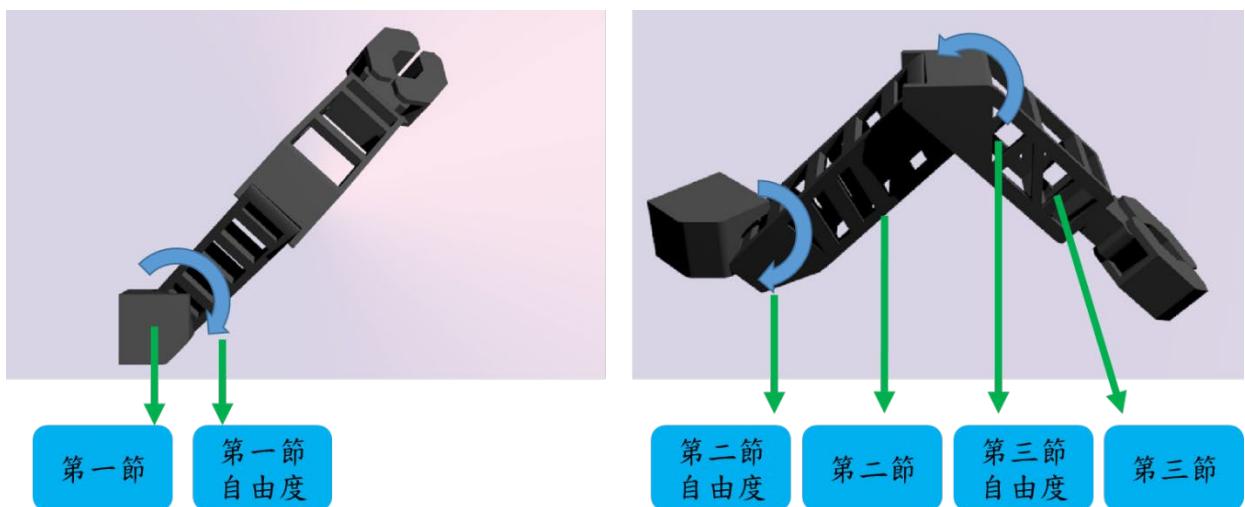


圖 6-10 一~三節機械手臂自由度

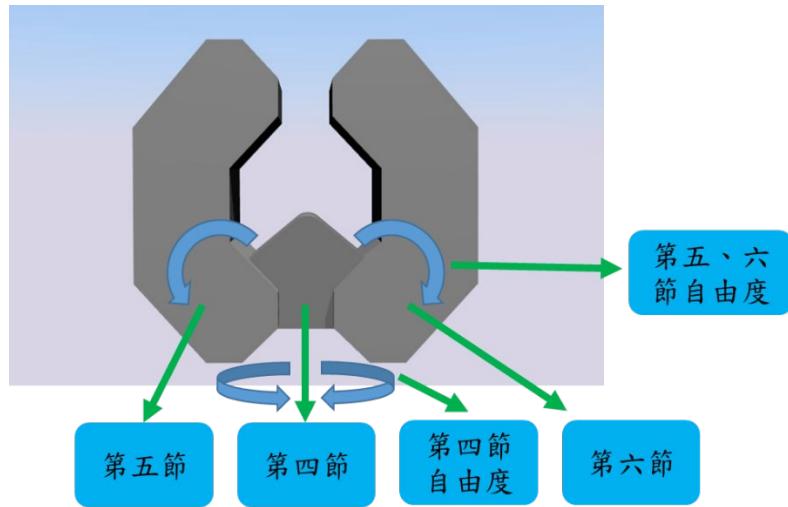


圖 6-11 四~六節機械手臂自由度

如 ROV 控制搖桿，第二搖桿將六節機械手臂各自的自由度控制分別部屬至搖桿的三個自由度與兩組按鍵上，如圖 6-12。



圖 6-12 第二搖桿-機械手臂控制搖桿

第7章 實作結果

第7-1節 基本物理量

以下是經由第 3-2-1 節的(3-1、(3-2、(3-3、(3-4 公式計算後所得出的結果與其位置示意圖：

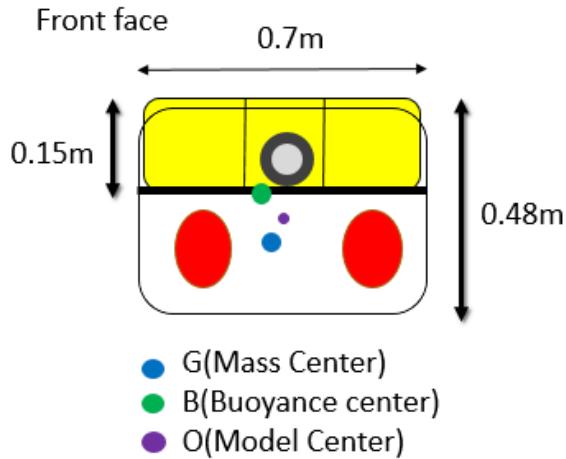


表 7-1 ROV 基本物理量

質量	114.46kg
體積	$0.133m^3$
質量中心	$<-0.0351, -0.0891, 0.0027>$
體積中心	$<-0.0226, 0.0202, 0.0069>$

單位：公尺

圖 7-1 質心與浮心的位置

由圖 7-1 可知計算出來的質心位置與浮心位置有符合第 2-5 節提及之水下載具質心與浮心相對位置的需求，在鉛直方向質心位於浮心下方約 0.11 公尺處，而在水平方向質心位於浮心 XZ 平面(-0.0577,-0.0042)的位置，距離大約為 0.0578 公尺。

經由第 3-2-2 節公式(3-5 計算後所獲得的主 ROV 慣性張量(Tensor of Inertia)：

$$\bar{I} = \begin{bmatrix} 6.46 & -3.57 & -4.85 \\ -3.57 & 13.48 & -2.61 \\ -4.85 & -2.61 & 12.04 \end{bmatrix}$$

第7-2節 穩定測試

在這一節展示的為 ROV 水下靜力的變化，將機械手臂的重力、浮力、與其慣性張量的變化都考慮後的運動差異。

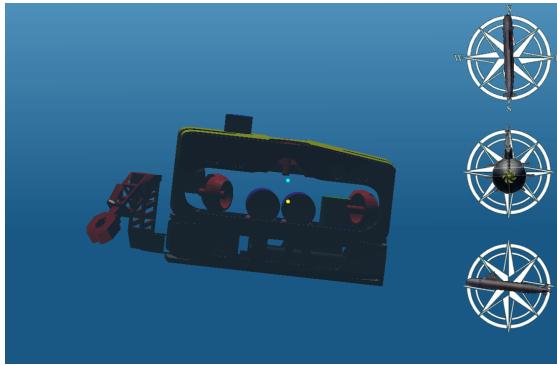


圖 7-3 無考慮機械手臂水下靜力平衡

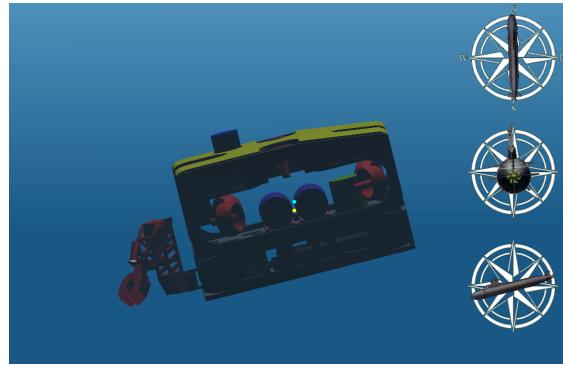


圖 7-2 考慮機械手臂水下靜力平衡

上圖中淺藍色點為浮力作用中心，而黃色點為重力作用中心，兩張圖表示在水下靜力的力矩達到平衡狀態時，考慮機械手臂與不考慮機械手臂的兩種狀態，無論有無考慮機械手臂，重力與浮力因為於不同鉛錘線而產生穩定力矩，使 ROV 朝向淨力矩為零的方向旋轉，又因為受到水中的旋轉阻力，使每次受到的穩定力矩逐漸減小，最終達到淨力矩平衡。

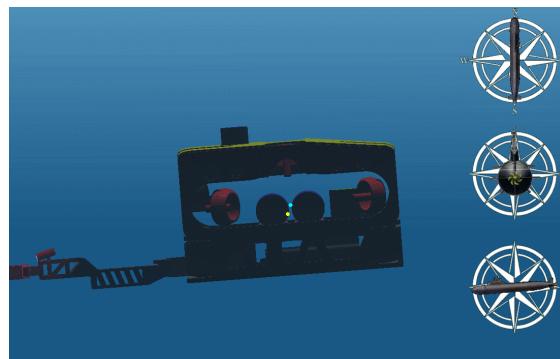


圖 7-4 自動穩定平衡姿態

以 PID 控制推進器調整因機械手臂運動而傾斜的 ROV 至平穩狀態，有助於操作者執行其他運動。

第7-3節 自動控制測試

這一節會測試經由 PID 控制器來使 ROV 達成某種特定角度或前往特定目的地，並觀察其收斂狀況。

測試一：位置維持(Positioning)，控制 ROV 保持於(0,-5,0)的位置

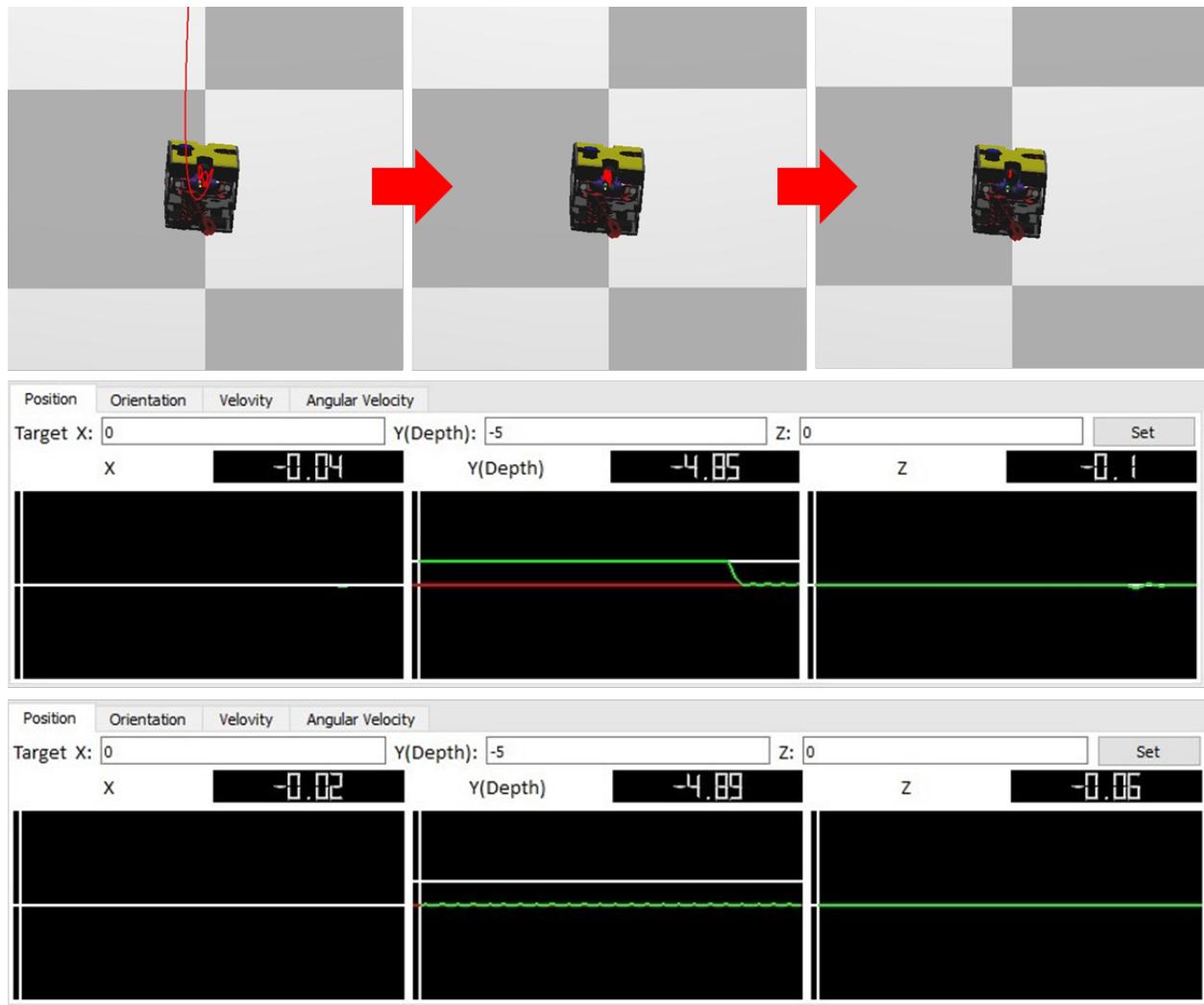
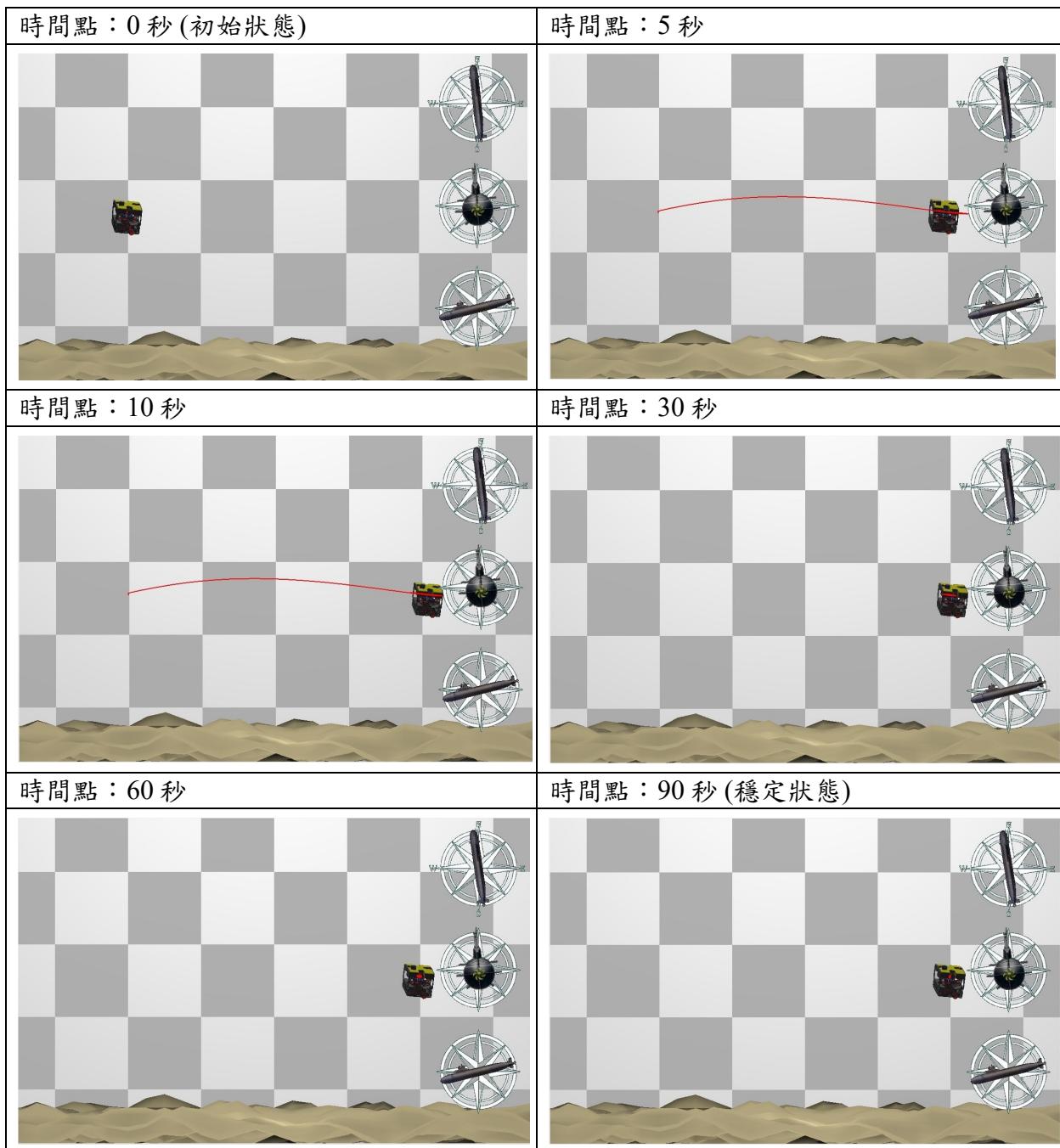


圖 7-5 穩定位置測試

當 ROV 收斂於目的地後，在水平方向的穩態誤差大約為 0.02~0.06 公尺，而鉛錘方向因為受到浮力與重力的影響，其穩態誤差約為 0.11 公尺。

測試二：移動至固定位置，將位置由(-4,-5,0)移動至(4,-5,0)

表 7-2 自動控制測試-位置維持(Positioning)

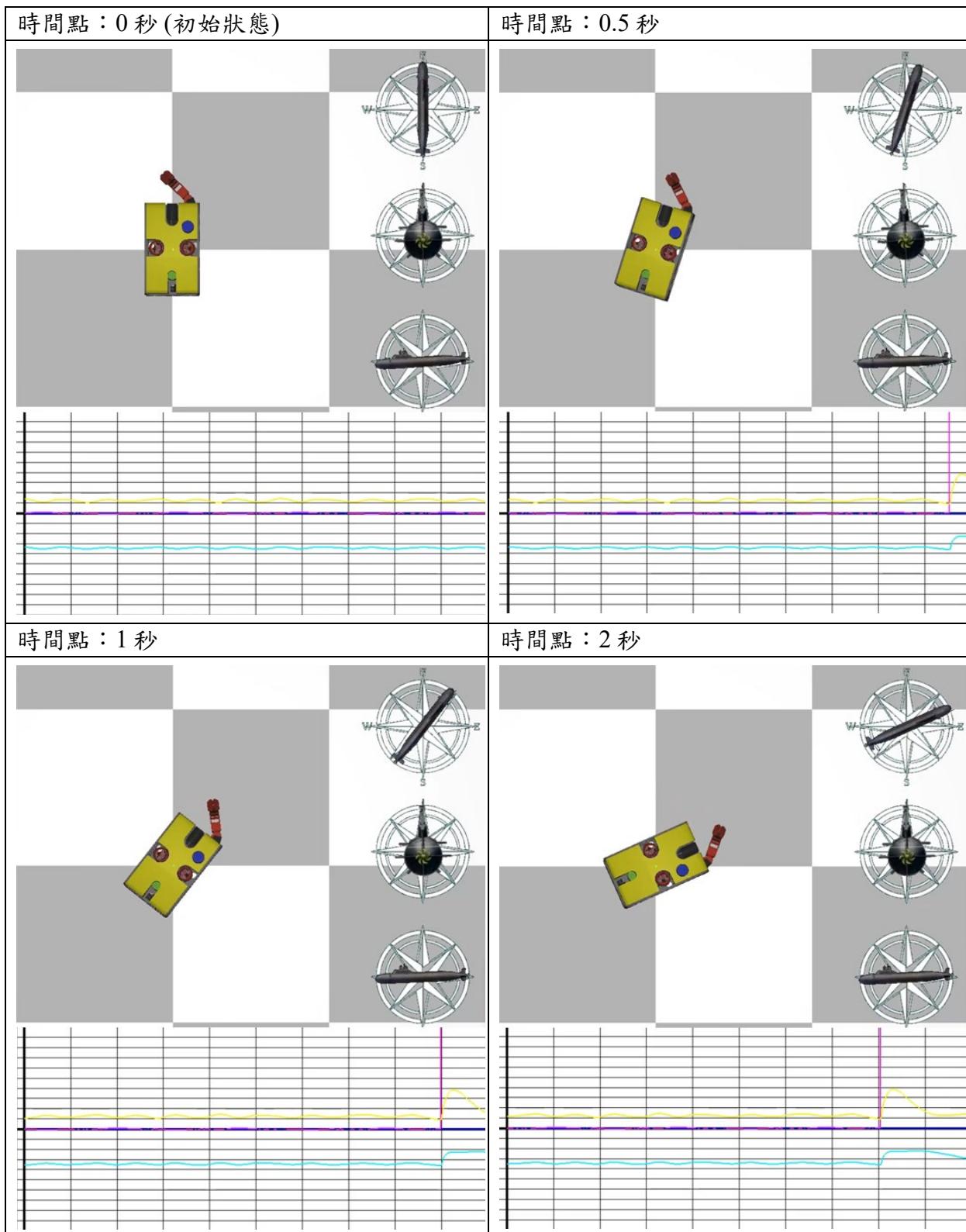


註：

1. 背景方格邊常為 2 公尺。
2. 每 0.1 秒紀錄 ROV 的位置，陣列大小 200，以紅色線段畫出，紅色線段顯示 20 秒後消失。

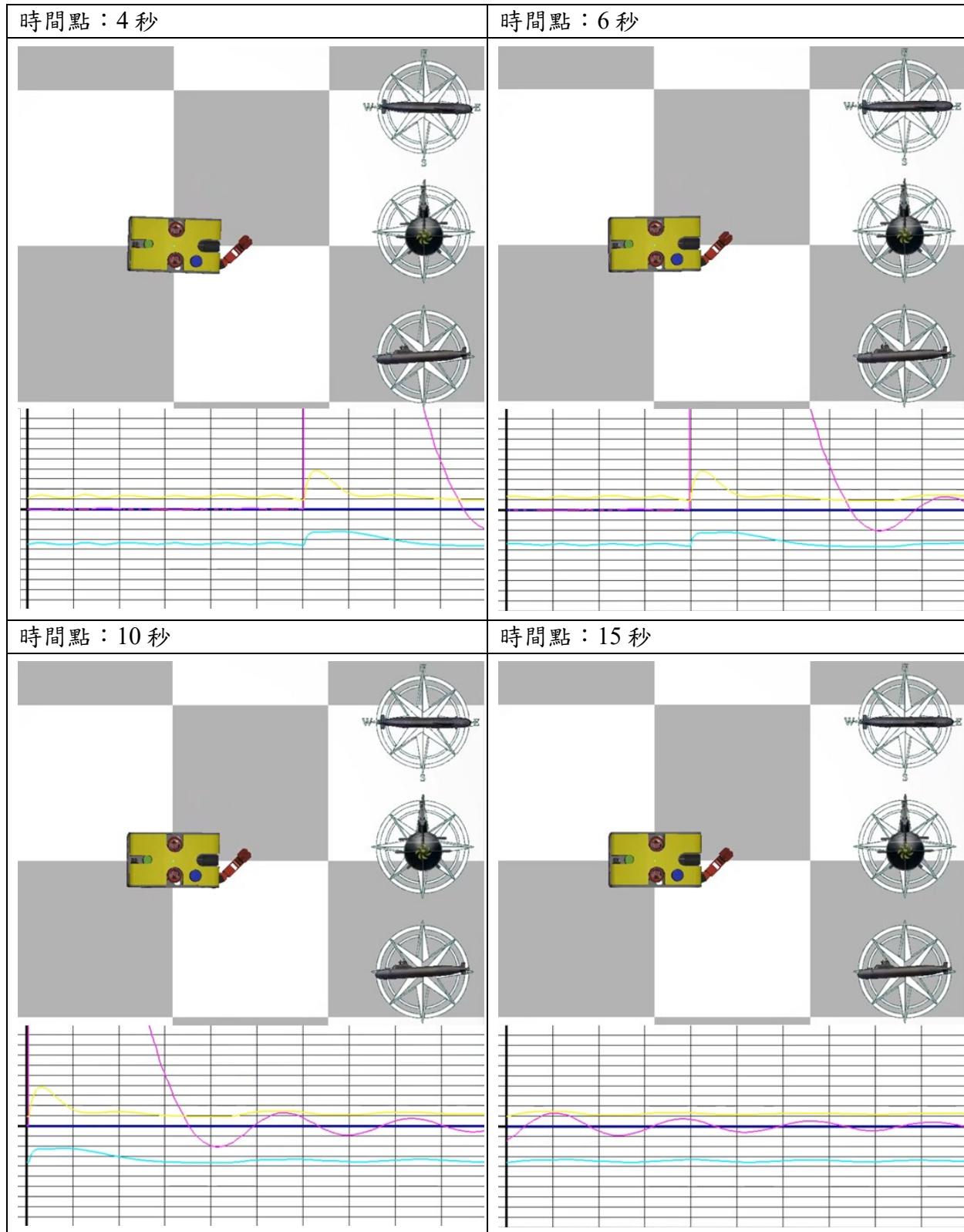
測試三：旋轉至固定角度，橫搖角由 0 度轉至 90 度，縱搖與側傾保持 0 度。

表 7-3 自動控制測試-姿態維持(橫擺 Yaw-航向)



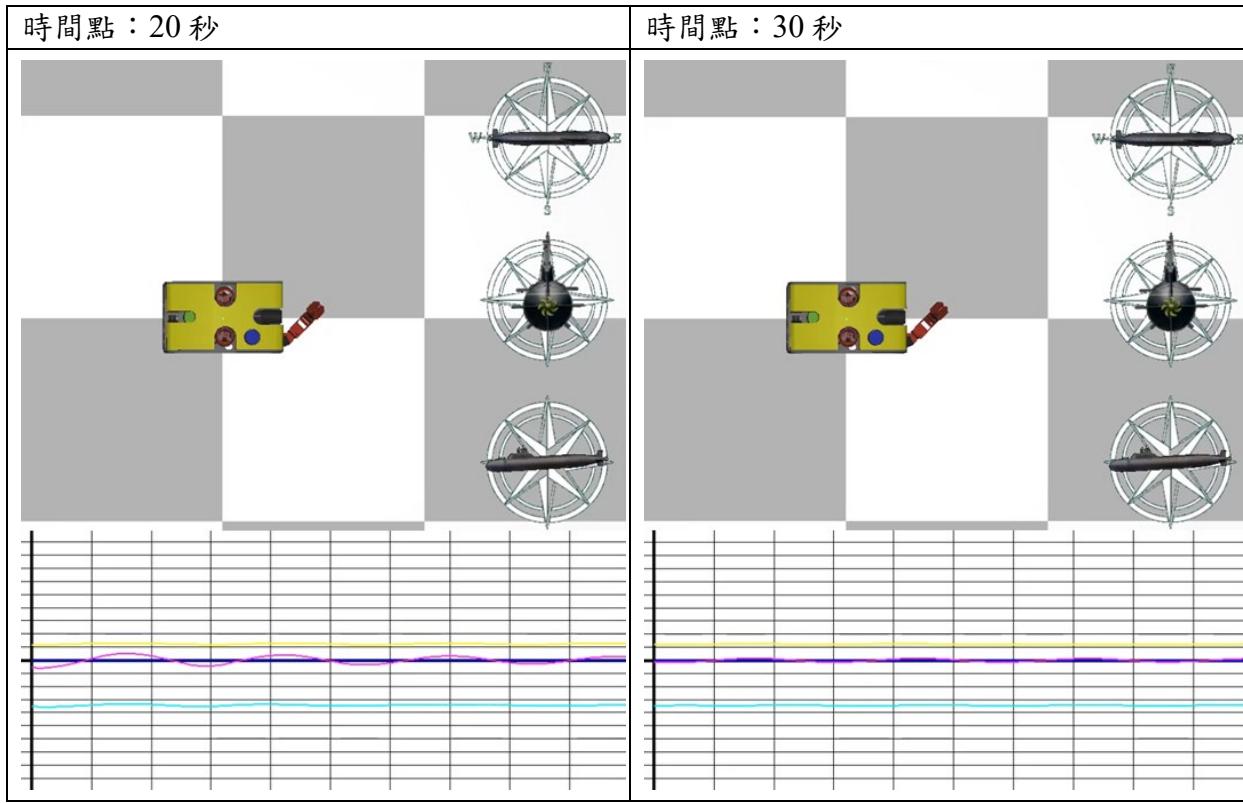
註：橫軸每單位為 1 秒，縱軸每單位為 1 度。

震盪期：



註：橫軸每單位為 1 秒，縱軸每單位為 1 度。

穩定期：



註：橫軸每單位為 1 秒，縱軸每單位為 1 度。

由表 7-3 內所示，ROV 的縱搖角(淺藍色線段)因為受到機械手臂的影響，雖然利用 PID 控制推進器將其平衡至 0 度，但是仍然會有接近於 3~4 度的穩態誤差，而測試的橫搖角(紫色線段)經過一定時間後，其穩態誤差大約為 0.2 度。

第7-4節 手動操作測試

在[12]這篇論文中，其著重於三個運動上：垂直運動、旋轉運動與 Z 型運動，在其論文中，其中就有包含 Turning Circle 與 Helix 動作，在本研究使用搖桿手動操作進行操作並測試。



圖 7-6 基本運動-Turning Circle

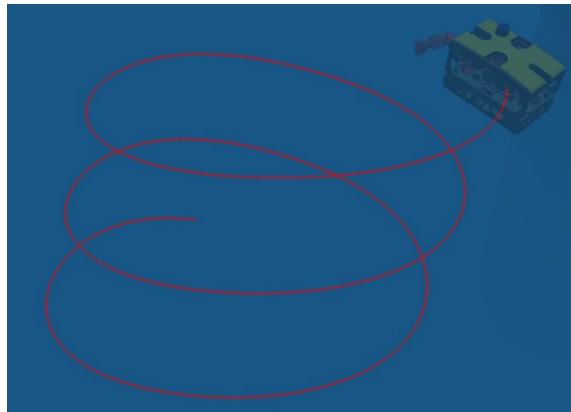


圖 7-7 基本運動-Helix

第8章 結論與未來發展

這份研究透過以遊戲程式設計的方式實做了一個用於訓練與計畫演練的即時水下無人載具運動模擬器，而非傳統的透過公式的推導模擬來建立數學模型；本研究利用體素(Voxel based)的方法計算 ROV 的水下靜力如質量與浮力，並根據平行軸定理(Principle of Parallel Axis Theorem)加以考量機械手臂運動時 ROV 水下靜力的變化；在使用者與操作介面部分本模擬器提供操作員直接下達命令的操作方式(Command Mode)與使用搖桿(Joystick Mode)的操作方式，並利用人工智慧引擎的 PID 控制器將其轉換為推進器向量(Propeller Vector)的系統以統合各個推進器之間的作用，以達成六軸運動的控制訊號，同時 PID 控制器也能執行維持平衡、維持深度等功能，唯目前穩態誤差在鉛直方向仍然有著 10 公分的誤差。

一般而言，模擬器的驗證較難以達成，大部分的論文其驗證方式均為操控 ROV 以執行特定的動作，將預期路線或位置、角度與實際模擬資訊進行比對，也有些論文直接操作真實的 ROV 於水槽內進行運動模擬；而此篇論文除了達成第一類的基本動作測試與 PID 自動控制測試外，更希望透過設計更多場景與任務，提供操作員使用與訓練，參考其使用行為與反饋以改良此模擬器。

本研究目前只針對單一一款 ROV 設計，但是此架構可以套用於任何多推進器的無人水下載具，未來希望能基於此基礎，引入更多水下無人載具的參數與模型，利用同一系統流程來達成不同水下無人載具的模擬，也可以透過此方式來驗證此模擬器的方法之缺失與不足。

最後，此論文詳述了機械手臂對 ROV 運動的影響，而未來，此模擬器的研究目標會朝向幾個大目標：1.模擬纜線對 ROV 的影響，計算纜線的張力、浮力與重力對於 ROV 運動時的作用，對於纜線的模擬，已有非常多的文獻可以參考。2.ROV 於海面的模擬，考慮 ROV 在施放與回收時，會因海浪的作用而不易進行操作，此研究可參考海面船隻的相關研究。3.計算 ROV 於海流內的穩定，利用本研究的人工智慧引擎，以推進器來穩定因受到海流影響的 ROV，保持穩定的位置與姿態。

參考文獻

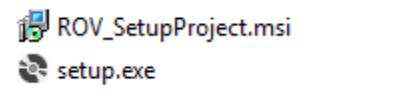
- [1] A. Abdulghany, "Generalization of parallel axis theorem for rotational inertia," *American Journal of Physics*, vol. 85, no. 10, pp. 791-795, 2017.
- [2] D. Baraff, "Physically based modeling: Rigid body simulation," *SIGGRAPH Course Notes, ACM SIGGRAPH*, vol. 2, no. 1, pp. 2-1, 2001.
- [3] J. Bender, K. Erleben, and J. Trinkle, "Interactive simulation of rigid body dynamics in computer graphics," in proceedings of *Computer Graphics Forum*, 2014, vol. 33, no. 1: Wiley Online Library, pp. 246-270.
- [4] C. E. Brennen, An Internet Book on Fluid Dynamics. 2004.
- [5] H. Bruyninckx, "Simulation, Modeling and Programming for Autonomous Robots: The Open Source Perspective," presented at the *Proceedings of the 1st International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, Venice, Italy, 2008.
- [6] C. Chin, W. Lin, and J. Lin, "Experimental validation of open-frame ROV model for virtual reality simulation and control," *Journal of Marine Science and Technology*, vol. 23, no. 2, pp. 267-287, 2018.
- [7] M. L. Corradini, A. Monteriu, and G. Orlando, "An actuator failure tolerant control scheme for an underwater remotely operated vehicle," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1036-1046, 2010.
- [8] M. L. Corradini and G. Orlando, "A discrete adaptive variable-structure controller for MIMO systems, and its application to an underwater ROV," *IEEE Transactions on Control Systems Technology*, vol. 5, no. 3, pp. 349-359, 1997.
- [9] J. Da Cunha, R. R. Costa, and L. Hsu, "Design of a high performance variable structure position control of ROVs," *IEEE Journal of Oceanic Engineering*, vol. 20, no. 1, pp. 42-55, 1995.
- [10] E. C. De Souza and N. Maruyama, "Intelligent UUVs: Some issues on ROV dynamic positioning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 1, pp. 214-226, 2007.
- [11] B. Denardo, "Demonstration of the parallel-axis theorem," *The Physics Teacher*, vol. 36, no. 1, pp. 56-57, 1998.
- [12] W. Deng and D. Han, "Study on simulation of remotely operated underwater vehicle spatial motion," *Journal of Marine Science and Application*, vol. 12, no. 4, pp. 445-451, 2013.
- [13] DWTEK. "ROV Investigator 90 Information." <http://www.dwtekmarine.com/products9-level-2-menu11.html> (accessed July 10., 2019).
- [14] J. Evans and M. Nahon, "Dynamics modeling and performance evaluation of an autonomous underwater vehicle," *Ocean Engineering*, vol. 31, no. 14-15, pp. 1835-1858, 2004.
- [15] Z. Fabekovic, Z. Eskinja, and Z. Vukic, "Micro rov simulator," in proceedings of *ELMAR 2007*, 2007: IEEE, pp. 97-101.
- [16] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *The International Journal of Robotics Research*, vol. 2, no. 1, pp. 13-30, 1983.
- [17] R. Featherstone, Robot dynamics algorithms. 1984.
- [18] B. Fletcher and S. Harris, "Development of a virtual environment based training system for ROV pilots," in proceedings of *OCEANS 96 MTS/IEEE Conference Proceedings. The Coastal Ocean-Prospects for the 21st Century*, 1996, vol. 1: IEEE, pp. 65-71.

- [19] J. D. Geder, J. Palmisano, R. Ramamurti, *et al.*, "Fuzzy logic PID based control design and performance for a pectoral fin propelled unmanned underwater vehicle," in proceedings of *2008 International Conference on Control, Automation and Systems*, 2008: IEEE, pp. 40-46.
- [20] M. H. Khodayari and S. Balochian, "Modeling and control of autonomous underwater vehicle (AUV) in heading and depth attitude via self-adaptive fuzzy PID controller," *Journal of Marine Science and Technology*, vol. 20, no. 3, pp. 559-578, 2015.
- [21] marinesimulation. "Multi-ROV training simulator." <https://marinesimulation.com/rovsim-gen3/> (accessed June 24, 2019).
- [22] T. F. Ogilvie and E. O. Tuck, "A rational strip theory of ship motions: part I," University of Michigan, 1969.
- [23] T. Prestero, "Development of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle," in proceedings of *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No. 01CH37295)*, 2001, vol. 1: IEEE, pp. 450-455.
- [24] M. Ryan and D. Lamothe, "Pentagon: Chinese naval ship seized an unmanned U.S. underwater vehicle in South China Sea," 2016.
- [25] N. H. Tehrani, M. Heidari, Y. Zakeri, *et al.*, "Development, depth control and stability analysis of an underwater Remotely Operated Vehicle (ROV)," in proceedings of *IEEE ICCA 2010*, 2010: IEEE, pp. 814-819.
- [26] UnicomCommunication. "ROV Interactive Simulator." <http://unicomcommunications.co.uk/simulation.html> (accessed June 24, 2019).
- [27] U. F. von Lukas, "Virtual and augmented reality for the maritime sector—applications and requirements," *IFAC Proceedings Volumes*, vol. 43, no. 20, pp. 196-200, 2010.
- [28] S. M. Zanoli and G. Conte, "Remotely operated vehicle depth control," *Control engineering practice*, vol. 11, no. 4, pp. 453-459, 2003.

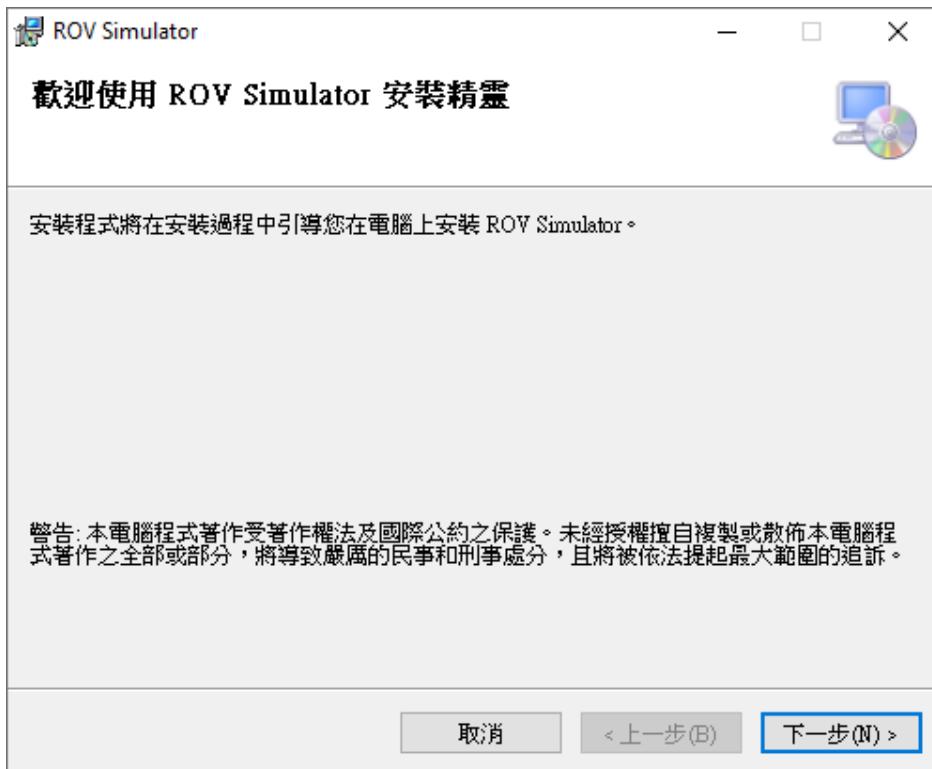
附錄

1. 安裝指南

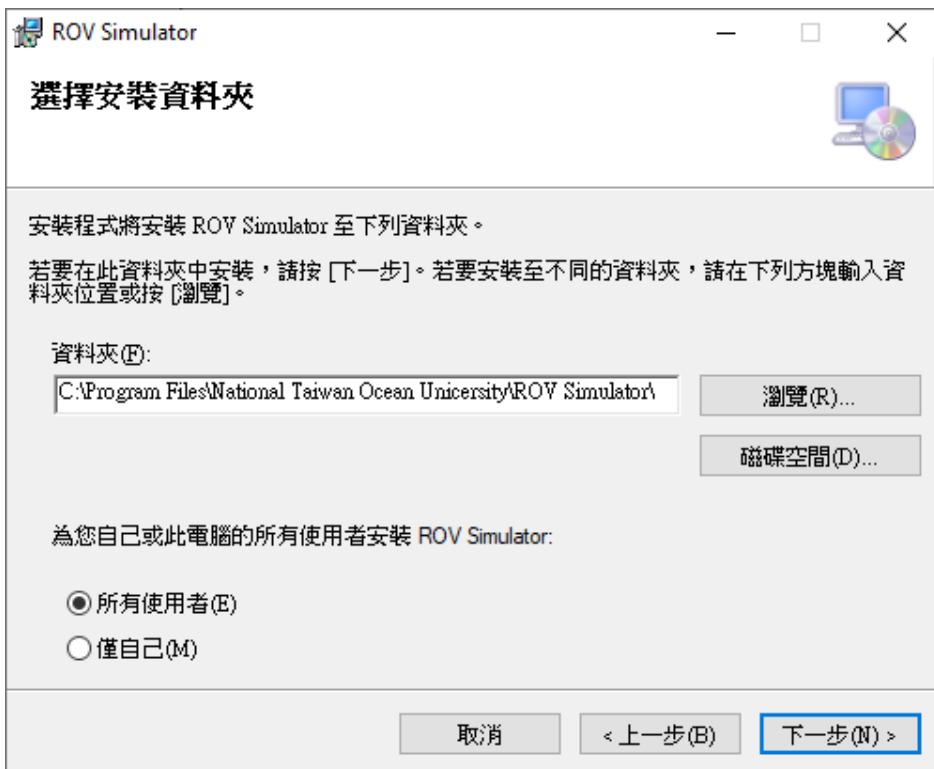
- 使用於本書最後面的安裝光碟
- 執行「setup.exe」



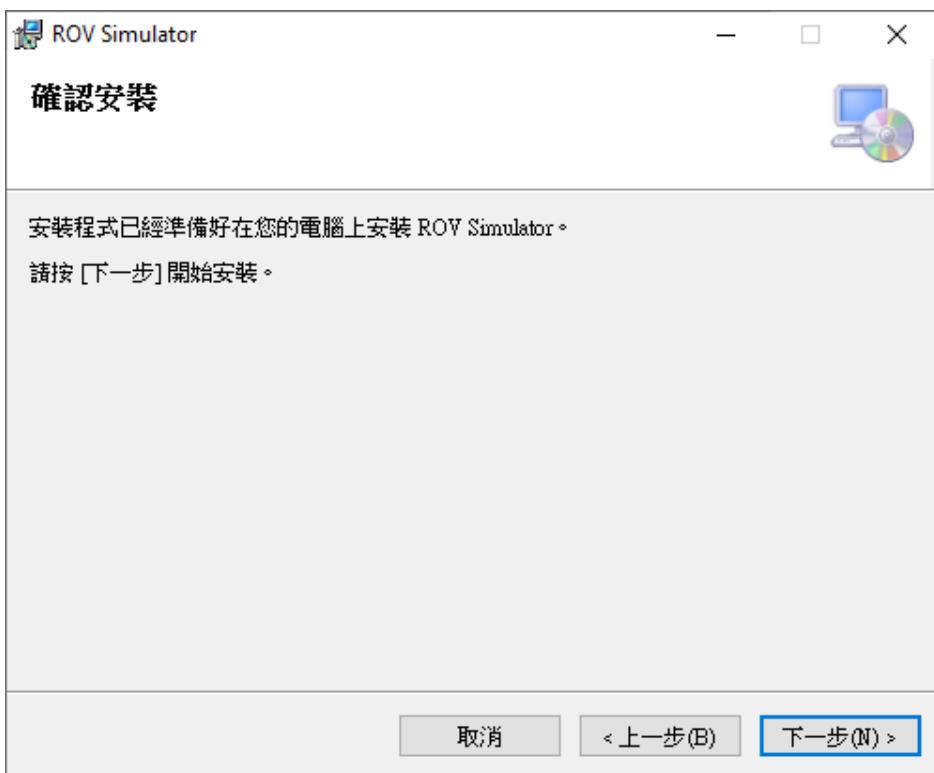
- 點選「下一步」



- 選擇路徑與使用者，點選「下一步」



- 點選「下一步」，確認安裝



2. ROV 模擬器操作指南 (ROV Simulator Operation Guide)

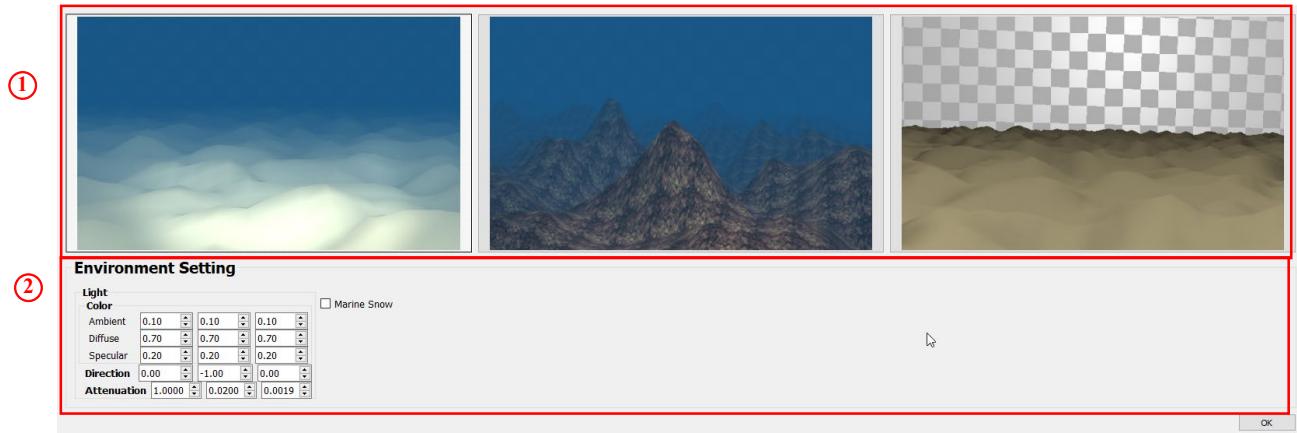


圖 1 場景選擇與環境參數視窗

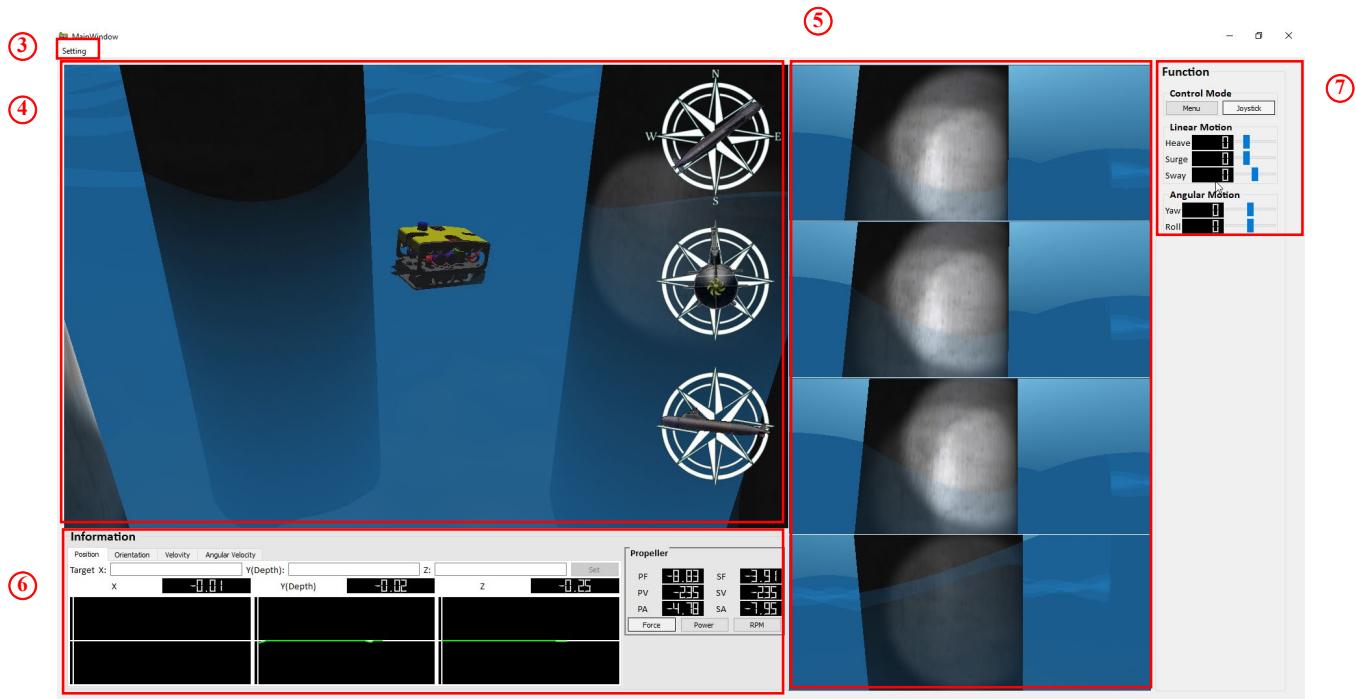


圖 2 主視窗

①場景選擇



圖 3 場景選擇視窗

程式開始執行後首先出現此視窗，提供三種場景：淺海沙地、深海岩地與用於測試的平行投影方格場景，選定場景後，需要變更也能在主視窗重新開啟此視窗進行重新選擇。

②環境參數

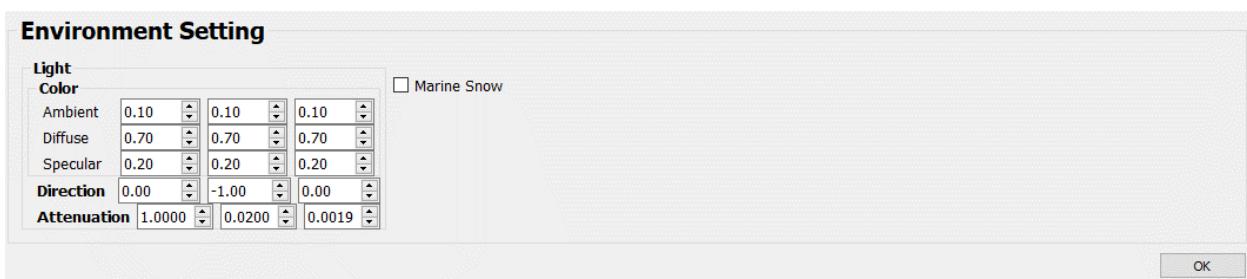


圖 4 環境參數視窗

提供設定一些環境參數，例如環境光線、海雪開關等等，建議使用預設值，決定好後按下確定即可執行主程式。

③主視窗設定

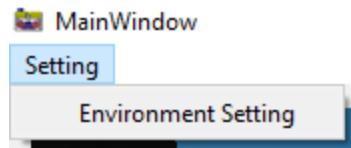


圖 5. 主視窗設定

在進入主視窗後，若想要重新更換場景或環境參數，點選左上角 Setting 並選擇 Environment Setting 即可重新叫出場景選擇與環境參數視窗。

④、⑤ROV 模擬顯示主、子畫面

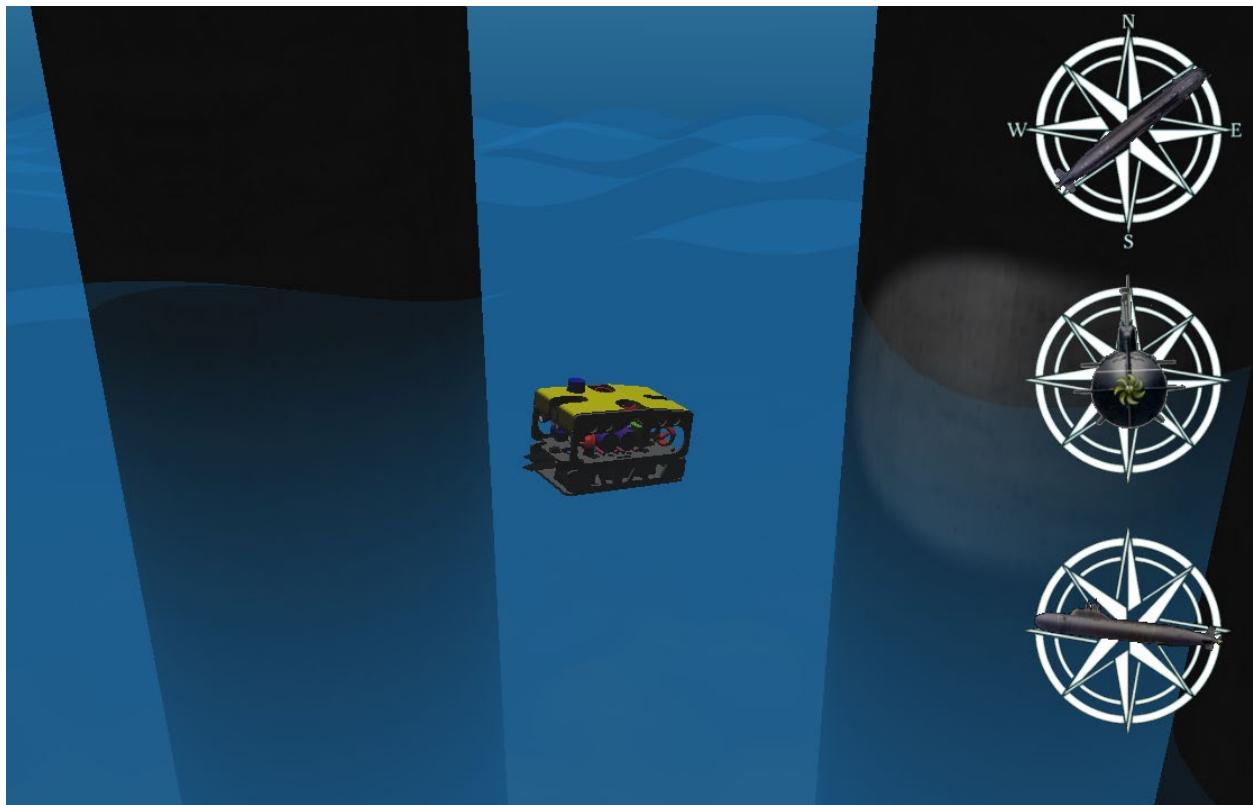


圖 6 ROV 模擬顯示主畫面

ROV 模擬的結果主要產生在這個地方，而右側羅盤分別顯示：1.航行方向-橫搖角 (Yaw Angle) 2.側傾角(Roll Angle) 3.俯仰角 (Pitch Angle)。

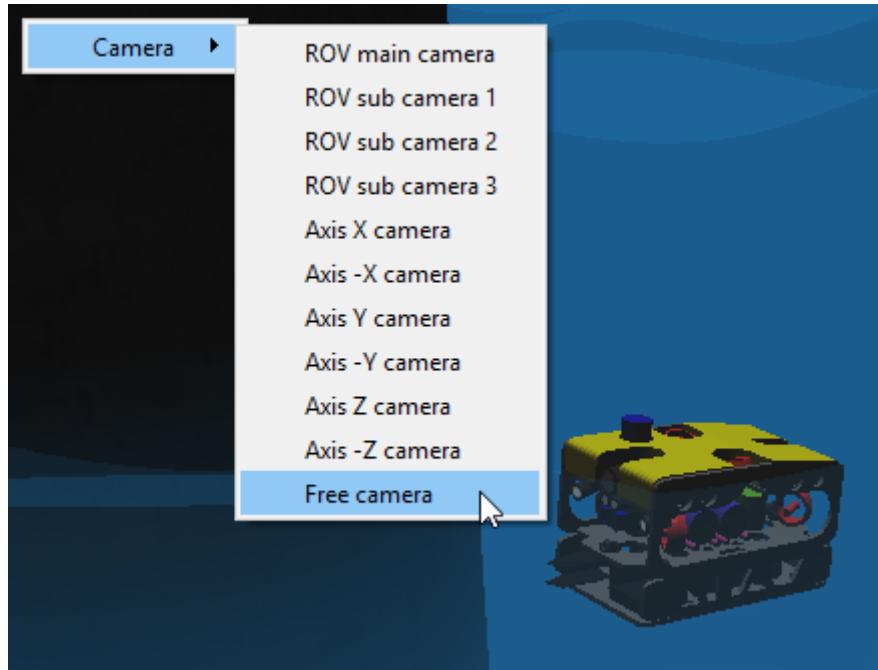


圖 7 主畫面鏡頭選擇

以滑鼠右鍵點擊 ROV 模擬視窗(主畫面及子畫面)即可跳出鏡頭選單，提供：

1. 以 ROV 的自身鏡頭，包含主鏡頭加三個子鏡頭，兩個朝前方，一個朝後方。
2. 以 6 軸(前後、上下、左右)平行投影觀看 ROV 鏡頭。
3. 自由鏡頭。(只有主畫面可使用)

選擇自由鏡頭後滑鼠轉化為鏡頭角度控制，並以鍵盤 WASD 控制鏡頭移動，持續按壓左 ctrl 鍵即可重新顯示滑鼠。

⑥ ROV 資訊視窗

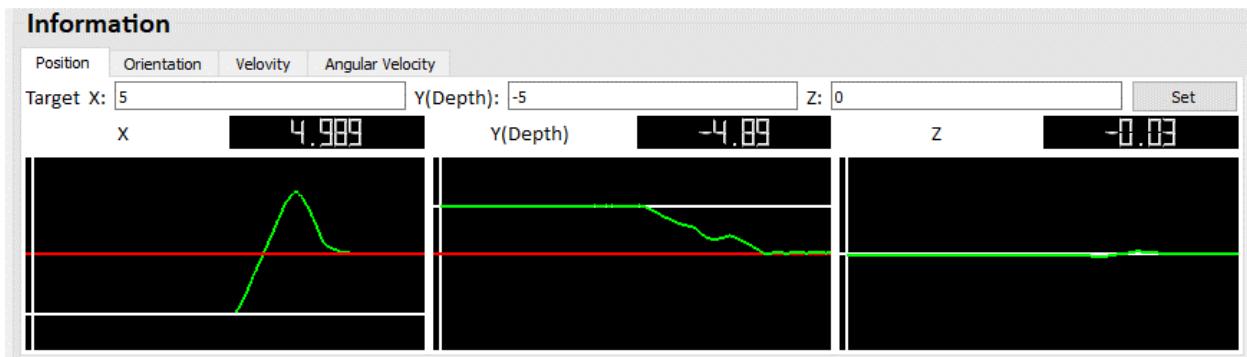


圖 8 ROV 資訊顯示(Position)

資訊視窗可以顯示 ROV 目前的：位置(Position)、姿態(Orientation)、速度(Velocity)、角速度(Angular Velocity)。

其中位置與姿態在 Command Mode(功能⑦)下可以透過設置目標(Target)並按下設置(Set)來讓 ROV PID 運作，透過 PID Controller ROV 會自行前往並保持在目標位置、姿態，若想取消 PID 運作，將 Target 清空並再次按設置即可。

白色橫軸 – 0

綠色曲線 – 近六秒內數值

紅色橫軸 – 目標值

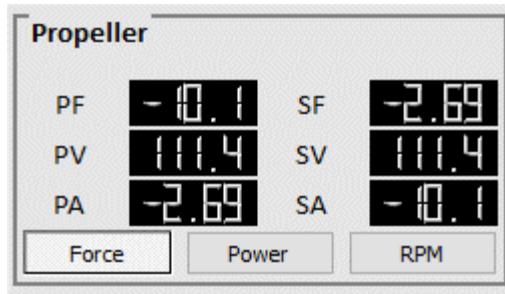


圖 9 Propeller 狀態視窗

此視窗顯示此 ROV 六個螺旋槳(Propeller)狀態，可以選擇官愾騎出力大小、功率與轉速。

⑦ROV 操控模式切換與 6DoF 操作

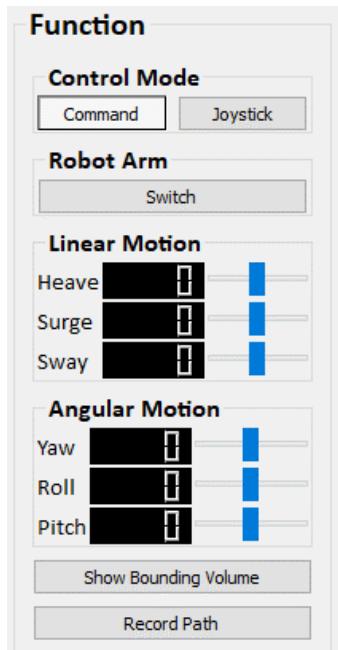


圖 10 ROV 操控模式切換與 6DoF 操作

此處可以切換 ROV 的操控模式(Control Mode)。

Command Mode :

功能一：

透過⑥資訊欄設定目標

功能二：

透過功能欄操作 6 DoF Motion Control

Linear Motion(Heave、Surge、Sway)

Angular Motion(Yaw、Roll、Pitch)

Joystick Mode :

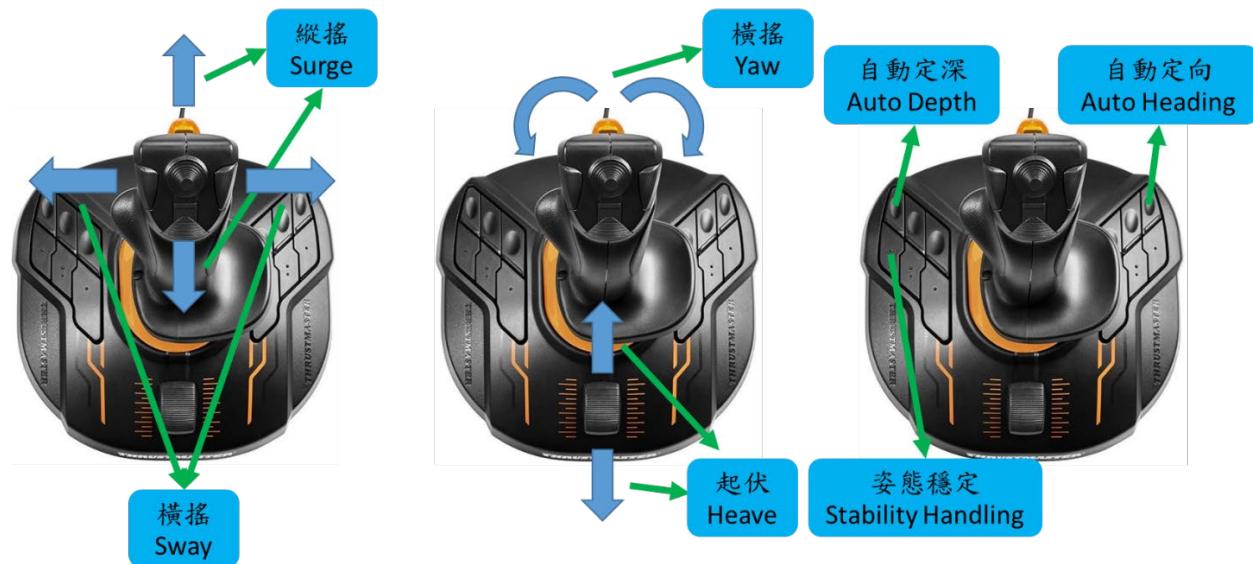
於功能欄(Function)內將 Control Mode 切換至 Joystick Mode

後即可使用

使用 Thrustmaster T.16000M FCS 搖桿



搖桿一：



搖桿二：



全文完