

遠端控制水下無人載具模擬器

指導教授：翁世光 教授

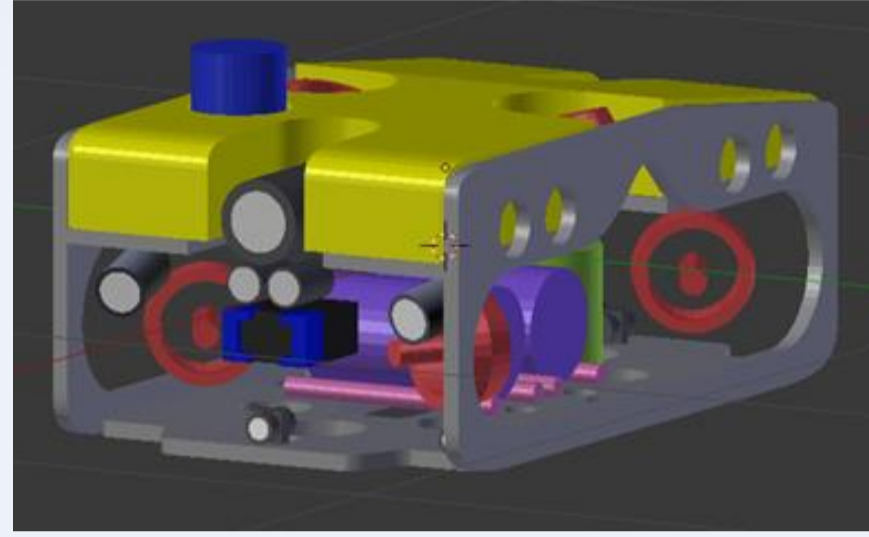
專題生：張智惟、黃筱涓

◆ 簡介

目前市面上水下載具模擬器注重於畫面的呈現，而較少考究於真實物理情況，並且只利用簡單的行為控制ROV，而不是利用多個物理作用合併、交互作用。本專題著重於物理模型的建立，以及行為模式的重建，從最底層的物理模型建構出高層的使用者指令，藉由這些模型來模擬ROV的運動模型，並且使用者可以透過大量參數的調整，達到相對一般化的模擬器。

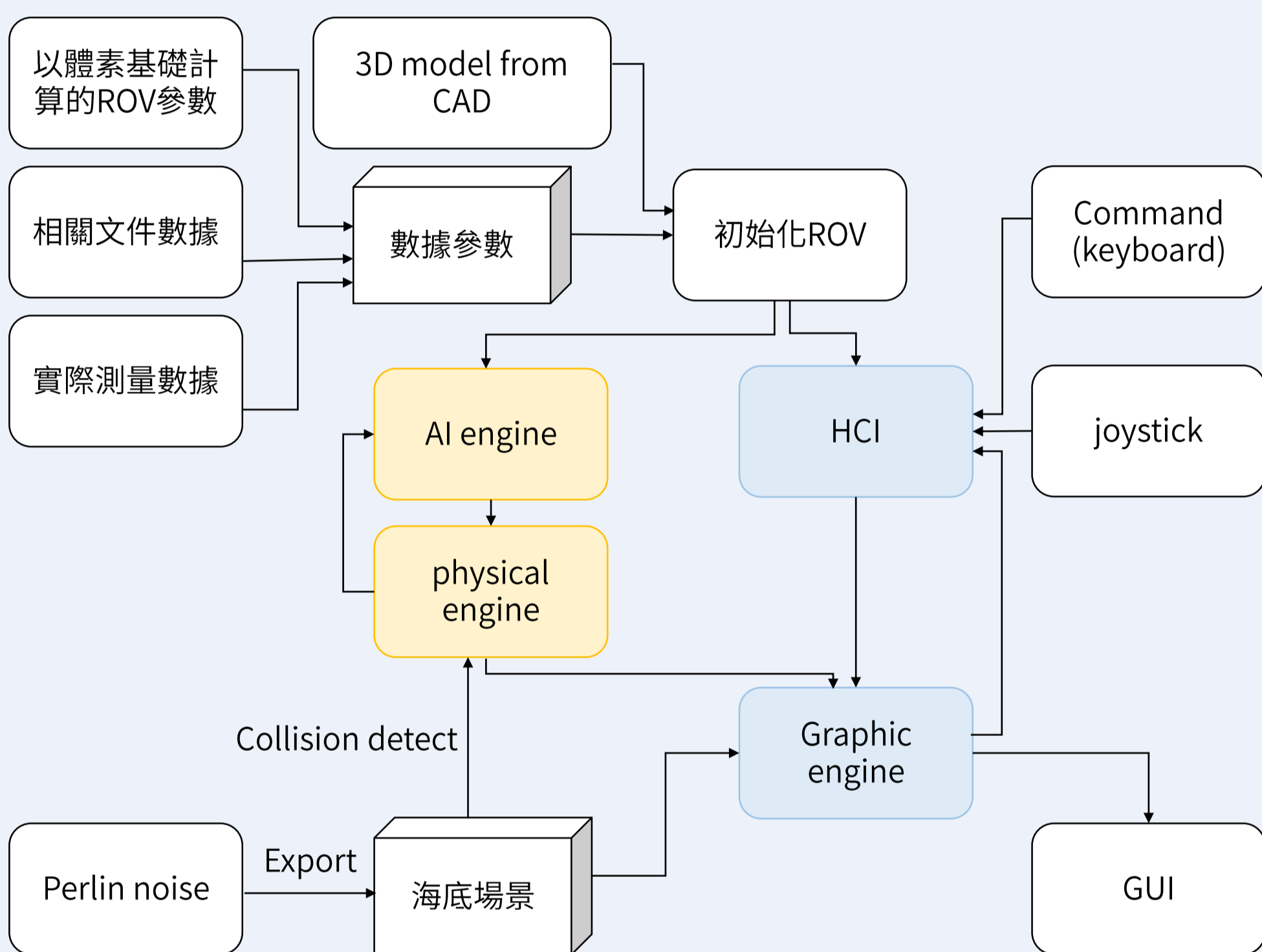


ROV - Investigator 90 實際圖片



ROV - Investigator 90 建模截圖

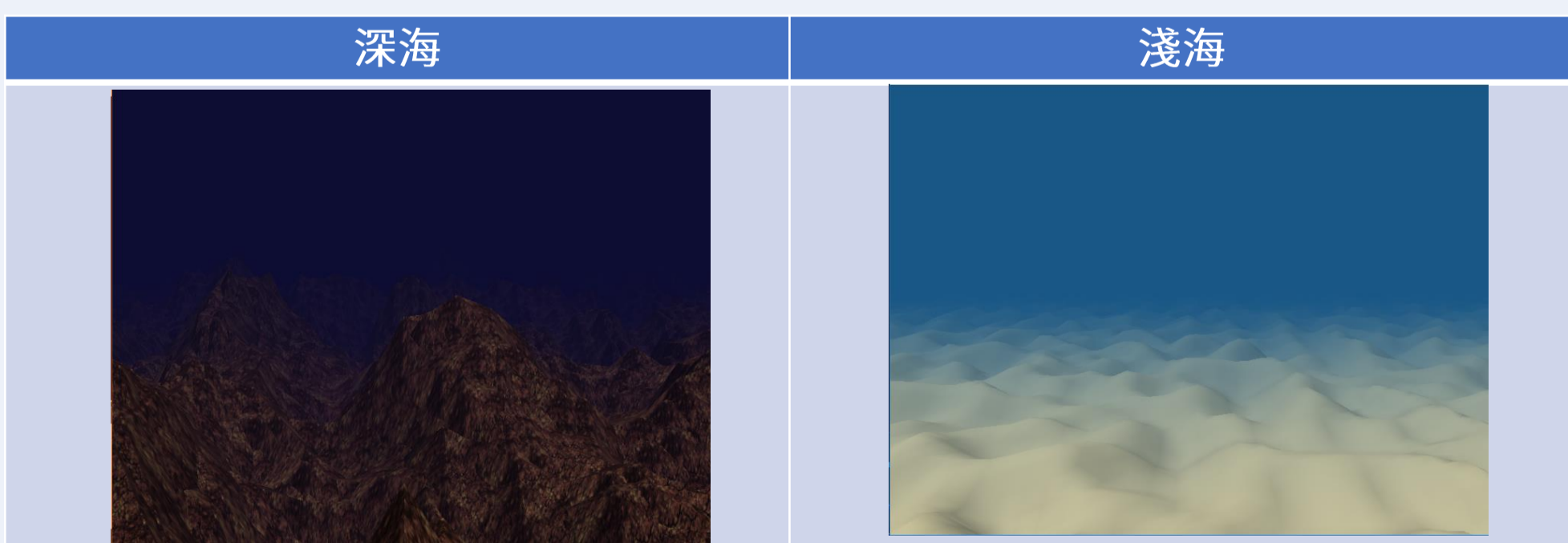
◆ 程式模組架構



藉由預先計算、設定好之參數寫入程式中，其中包含三項，第一，體素基礎所計算的ROV資料，其中包含ROV的質心、體積中心、質量、體積、慣性張量。第二，參考ROV維修手冊設定螺旋槳的推進力以及建模參考。第三、測量數據，藉由實際測量實體ROV，設定螺旋槳角度、各部件配置以及修正。將數據以及預先輸出的3D模型作為參數初始化ROV物件，並計算起始資訊。

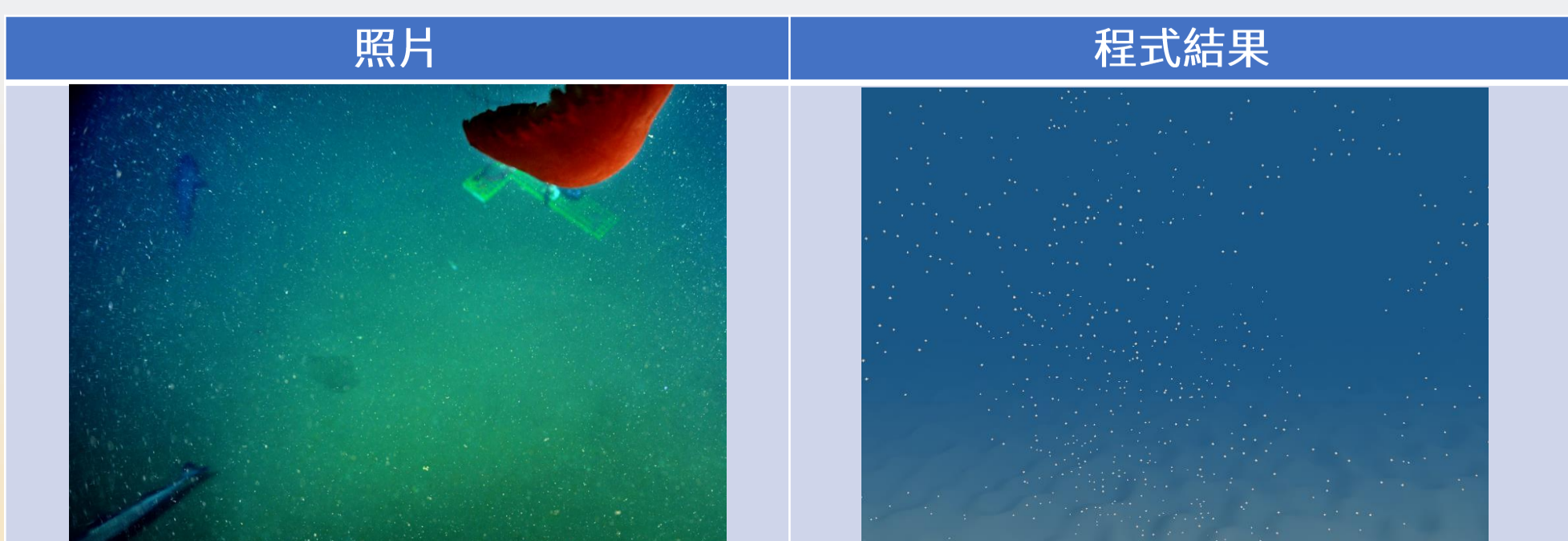
◆ 地形

透過不同頻率的噪聲和不同的材質貼圖，我們可以產生不同的地形。深海的地形可以比較有高低起伏，材質貼圖也是用石頭，而淺海的地形比較平緩，材質貼圖使用沙子。



◆ 海雪

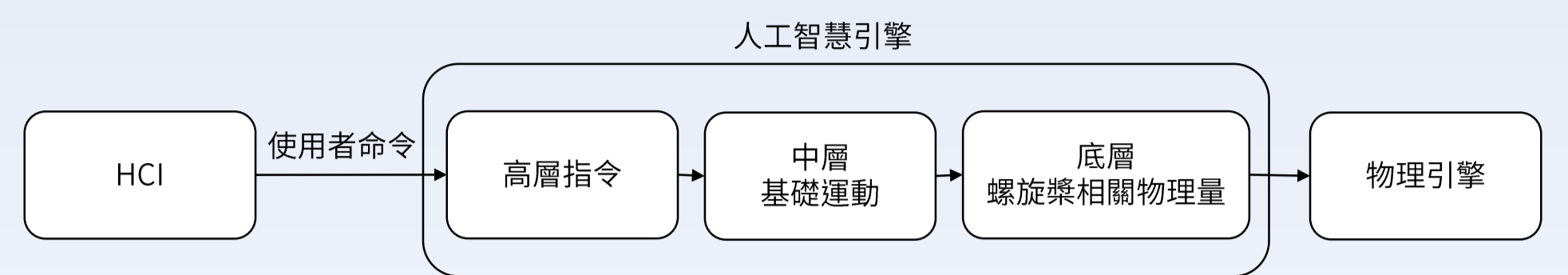
使用柏林噪聲產生海雪貼圖，並且利用GLSL達成去背效果，再產生很多貼上海雪貼圖的billboard，不停的繪製在觀看者周圍，讓觀看者可以看到海雪慢慢落下的景象。



照片出處：<https://ocean.si.edu/ecosystems/deep-sea/marine-snow-staple-deep>

◆ 運動模型實作

• 運動流程圖



• HCI(Human Computer Interaction)

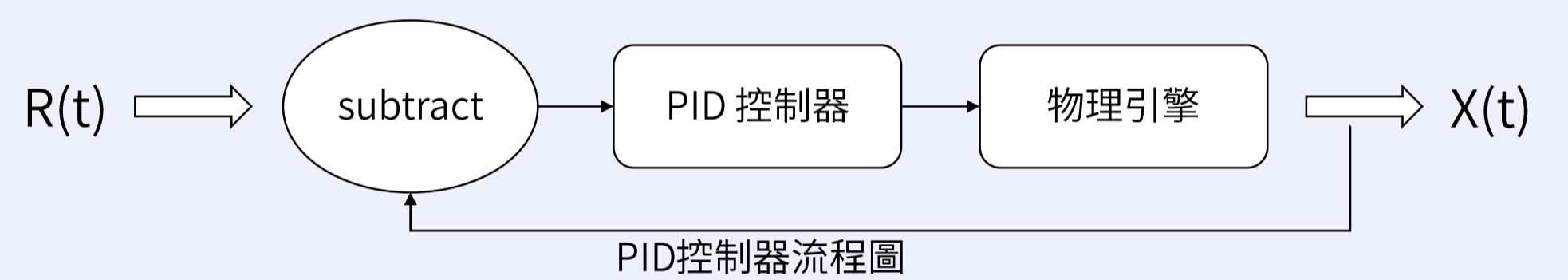
使用者以兩種模式輸入命令，Command模式和Joystick的模式。使用者可以透過Command視窗進行指令輸入，包括下潛、三維移動、旋轉等高階運動，也可以利用手把來直接控制ROV的運動。

• 人工智慧引擎

將使用者藉由HCI給予的高階命令訊號解碼成高階運動，利用多個PID控制器將高階運動轉換至中階的基礎運動，並在將其解碼維自定義的螺旋槳向量，透過個別計算將其轉換至低階的物理量，並送入物理引擎中。

• PID 控制器

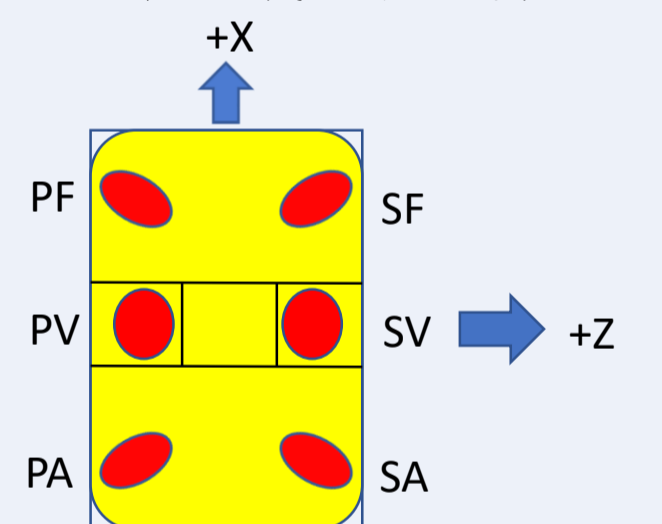
依據本專題的定義，所有高階運動皆由基礎運動組成，因此我們需要將高階運動進一步轉換成基礎運動，因此我們利用PID控制器來決定如何轉換，而PID控制器是將目標值與當前值計算出誤差值，並利用該誤差值來調整下一個單位時間的運動。



• 基礎運動

ROV的基礎運動藉由螺旋槳來達成，本計畫之ROV可以最多做出5種自由度的基礎運動，而這些基本運動則是由所有螺旋槳的不同的出力比例來完成，因此這些基礎運動採用自定義的螺旋槳向量來定義，其代表6個螺旋槳的出力比例。

螺旋槳向量(propeller vector)，為一六元素之向量(六維向量)
 $\langle \text{propeller} \rangle = \langle \text{SV, PV, SF, PF, SA, PA} \rangle$
並且符合 $\{ \text{SV, PV, SF, PF, SA, PA} \in [-1, 1] \}$
其分別表示六個螺旋槳

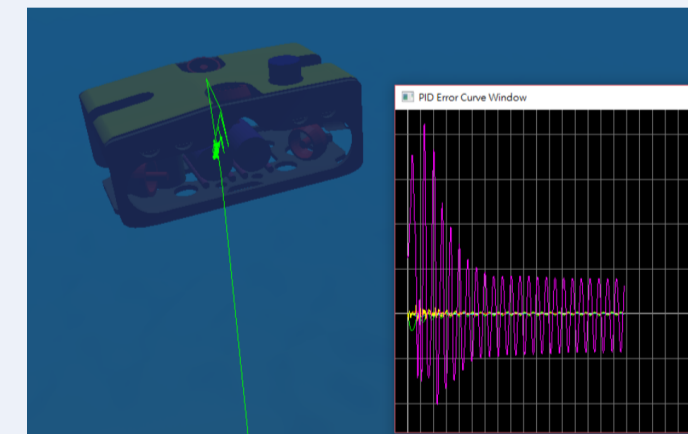


◆ 成果

• 螺旋運動



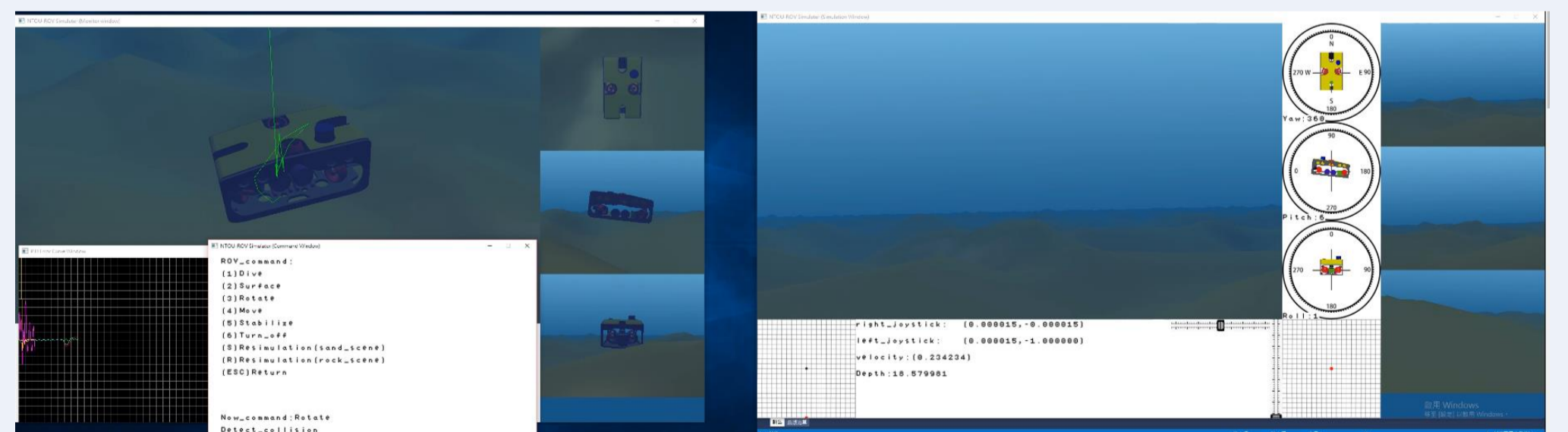
• 自動平衡



• 蜿蜒運動



• 實際畫面



◆ 結論

本計畫藉由考究海中的物理模型，運用剛體運動結合電腦圖學中的基本變換模擬出海中物體的行為，建立了完整的運動模型。除了實作PID控制器，另外也藉由將基礎運動定義為六個元素的向量，再利用向量計算的耦合表示六組螺旋槳的行為，最後由繪圖引擎渲染出ROV的整體運動行為，並記錄移動軌跡，做出複雜的運動行為；另一方面，利用多個著色器將多光源、foggy effect、particle system的效果實現，並使用者介面以及多個資訊提供提供視覺化的監控。