

# 有「感」而發-探討外部感應器對機器人競賽表現效能評估

## Exploring the efficiency of Robot with external sensor

黃昱誠<sup>1</sup> 陳宥潔<sup>1</sup> 陳怡翔<sup>2</sup> 黃昭銘<sup>3</sup>  
HUANG, YU-CHENG<sup>1</sup> CHEN, YU-CHIEN<sup>1</sup> CHEN, YI-HSIANG<sup>2</sup> HUANG, CHAO-MING<sup>3</sup>

<sup>1</sup>宜蘭縣立復興國民中學

<sup>1</sup> Yi-Lan Fu-Shing Junior High School

<sup>2</sup>宜蘭縣立復興國民中學生活與科技教師

<sup>2</sup> Yi-Lan Fu-Shing Junior High School Teacher

E-mail : [sag0802@mail.ilc.edu.tw](mailto:sag0802@mail.ilc.edu.tw)

<sup>3</sup>宜蘭縣宜蘭市中山國民小學教師

<sup>3</sup> Jhong-Shan Elementary School Teacher

E-mail : [stanely503@gmail.com](mailto:stanely503@gmail.com)

### 摘要

從 2018 年開始參加 VEX 機器人比賽，雖然成績每年都持續進步，有一個項目一直讓我們很頭痛，那就是「自動控制賽(Program)」。自動控制賽只能用程式(Program)來控制。機器人透過程式來完成比賽任務本來就非常具有挑戰性，往往只是誤差幾度，就會造成預設目標與機器人位置產生嚴重偏差，所謂「差之毫釐，失之千里」就是這個道理。本次研究主要透過比較三種不同的編程方式：1. 「程式」、2. 「程式+陀螺儀」與 3. 「程式+陀螺儀+比例控制」，進行機器人完成任程成效評估。研究結果顯示直接使用「程式」效果最差，「程式+陀螺儀」次之，「程式+陀螺儀+比例控制」表現最佳。

**關鍵字：**機器人、外部感應器

### Abstract

We participated the robot competition since 2018. The most difficult part of robot competition is Program, a kind of autonomous mode. It is very hard to have robot go straight and turning precisely by coding program only. This study tried to compare three programming methods, including 1. programming, 2. programming and gyro sensor, 3. programming and gyro sensor as well as proportional control. The method of programming and gyro sensor as well as proportional control shows the best performance including heading straight and turning precisely. Additionally, by mean of proportional control code it provides a mechanism of modification during robot moving.

**Keywords :** robot, external sensor

## • 前言

連續參加機器人比賽已經超過三年時間，我們每年比賽的時候，有一個項目一直讓我們很頭痛，那就是：技能挑戰賽(Skill Challenge)。技能挑戰賽有分兩種，一種是「遙控器控制賽(Driver)」，一種是「自動控制賽(Program)」。遙控器控制賽從名稱就可以知道這個項目是要用遙控器來控制機器人，而自動控制賽就是不能用遙控器，只能用程式(Program)來控制，但因為常受到外在變數的影響，常常導致每次的結果都不一樣，所以它是我們每年的惡夢，重點是，不管誤差是幾度，只要你再前進或後退，機器人最後到的位置一定會超乎你的想像，正所謂「差之毫釐，失之千里」。這個差之千里的狀況往往讓我們在比賽中花費許多時間進行試誤學習，耽誤我們其他練習的機會。

因此，我們希望透過這次的研究比較不同的編程組合讓機器人執行任務，然後針對資料進行三個向度的比較與評估：1.準確性、2.成功率與 3.穩定性，希望能夠找出最佳方式應用在未來新的賽季機器人比賽，藉此來提高自動控制賽的成績表現，一方面讓我們有更充裕的時間與合作賽隊進行團隊挑戰賽練習。

我們研究的目的是為了增加機器人在自動控制賽賽場的移動方位與角度的準確性，與每一次表現的穩定性與成功率，進而減少編程修改與調整參數時所需的時間。為什麼要減少編程的時間呢？這是因為我們以往在編程的時候，機器人的電量、馬達的狀況都不太一樣，導致每一次測試出的結果都不一樣，但我們又不知道該怎麼辦，只能一直嘗試調整參數與數值，所以我們每年都要花上大把的時間在編程上。以下將針對本次研究所要探討的準確性、穩定性與成功率進行說明。

### 一、準確性:

在自動控制賽中各賽隊需要針對比賽任務規劃機器人執行任務流程，整個比賽限時 1 分鐘，換言之各賽隊需要先規劃機器人移動路徑、轉彎角度、依執行任務而撰寫程式。因此能夠精準移動每一個設定的位置執行任務就是非常重要的工作。如何保持機器人能夠保持直線前進方位、轉彎角度與移動距離正確就是成功的關鍵。所以本次研究便是要評估三種編程方式是否可以提高機器人在移動與轉彎後方位與角度是否能夠與預設值保持一

致。

## 二、成功率:

而為什麼要增加自動控制賽的成功率呢？這是因為我們辛辛苦苦花了好多時間所寫的程式，成功率大概也才 20%-30%，如果再加上一些場地因素，可能就只剩下 10% 而已。本次研究除了測量機器人移動後的角度，我們也模擬比賽安排推 riser<sup>1</sup> 的任務，看看不同編程方式所產生的角度偏差對於機器人執行任務的成功率的影響。

## 三、穩定性:

在自動控制賽中每一賽隊僅有「三次」挑戰的機會，而且技能挑戰賽的時間是有限制的，一旦團隊挑戰賽賽事結束後，技能挑戰賽也就跟著結束，更可怕的是如果只有嘗試一次自動控制賽，當技能挑戰賽時間結束後，你所剩餘的挑戰次數也不能再去挑戰。一般來說自動控制賽的場地是不對外開放練習，無法事先反覆練習。如何讓機器人能夠克服外在不確定的因素，例如場地與機器人狀況，也就是當機器人若稍有角度偏差現象發生時，仍然能夠成功完成任務，跟穩定性有絕對關係。

綜合上述，我們本次研究的首要目標就是讓探討機器人的準確性機器人能轉到正確的角度，和保持直線前進；接著透過是否能夠將 riser 推入得分框完成任務來探究成功率，最後評估每次執行任務時產生角度誤差的大小與成功率的關係，藉此評估機器人的穩定性。我們希望在這次研究中評估並找出最佳的組合方式，運用在明年賽季的自動控制賽中，讓我們競賽成績能夠更上層樓。

## ● 文獻探討

本次研究將採用在自動控制常運用到的 PID 控制，PID 控制(Proportional, Integral, Derivative control)，PID 控制廣泛用在自動控制設計上，「比例控制」(Proportional control)是比較簡單的設計，而「積分控制」(Integral control)與「微分控制」(Derivative control)是比較複雜困難的，本次研究主要是採用比例控制方式來進程式編程。以下將簡單針對 PID 控制進行簡單說明。

---

<sup>1</sup> 此為比賽物件，請參閱 p5 說明與圖 3 圖示。

本次 PID 說明我們以倒水為例，以「比例控制」(Proportional control)來說：假設我們需要在燒杯中倒入 500cc 的水，我們的容器只能一次只能倒入 300cc 的水(設定的比例值)，所以燒杯內不足 500cc 時，需要加水，超過 500cc 的時候，又需要將多餘的水倒出，透過重複倒入、倒出的方式最後讓燒杯內的水達到 500cc，這個過程是比較緩慢的。

再以「積分控制」(Integral control)來看上述倒水例子，這時候容器倒入的水量不再受比例控制，倒入的水量以燒杯內所需要的剩餘水量為依據，當燒杯內還需要 500cc 的水量，這時候倒入的水量會增加，例如一次倒 300cc，第二次導入水量時則考慮目前進需要 200cc 水量，這時倒入的水量會降低為 150cc，第三次倒水時則會參考目前燒杯僅剩下 50cc 的容量，然後調整第三次倒水的量，透過這個方式完成 500cc 的任務。

「微分控制」(Derivative control)主要是參考變化量進行調整，例如倒水過程中突然減少或增加預定的水量，這時候透過「微分控制」可以察覺突然的改變量，然後進行調整，讓水面可以穩定達到 500cc 的水量。

## ● 研究實施與設計

### ● 研究工具

本次研究所使用的設備與器材總共分成 3 大類，分別為 1.比賽場地、2.機器人器材、3.編寫程式軟、硬體設備。以下將針對這 3 大類研究設備與器材進行介紹。

#### 一、比賽場地：

VEX IQ 機器人比賽有規定的場地，必須使用 VEX IQ 所認證的場地，本次使用的場地規格為 2020-21 賽制所使用的正式場地規格，場地長與寬尺寸為 8 英尺 x6 英尺的長方形場地(圖 1 所示)。

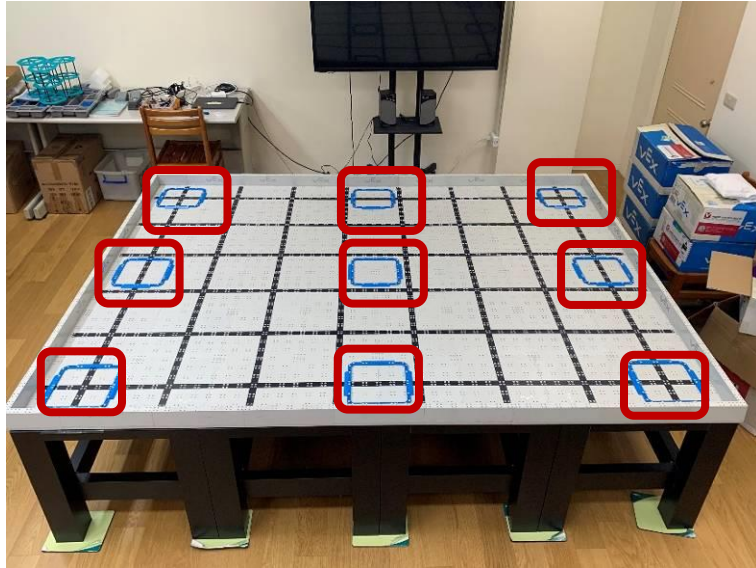


圖 1.比賽場地(8 英尺 x6 英尺)

本次研究將採用 2020-21 VEX IQ 台灣公開賽的比賽主題得分套件，在場地上面分別有九個藍色的正方形得分框(圖 4 中的紅色框框)，得分框為面積 8 英尺 x8 英尺表面積為 64 平方英尺的正方形(圖 2 所示)。

比賽時需要將 riser 放入得分框內方可成功獲得分數，如果 riser 壓在藍色得分框中則不予計分。

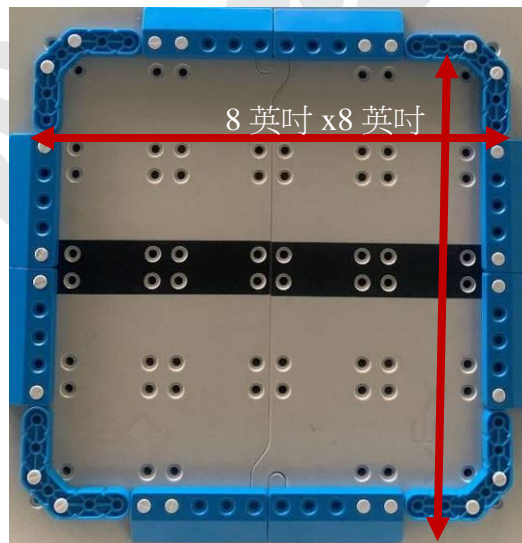


圖 2.得分框

本次比賽需要將 riser 放入或推入得分框即可獲得分數，riser 為高度 8.7 英尺高，長度與寬度各為 7 英尺(底面積經過估算約為 49 平方英尺)八角形柱

狀體(圖 3 所示)。



圖 3.riser 外觀

## 二、. 機器人器材：

VEX IQ 機器人比賽有規定的材料，機器人所使用的任何零件必須使用 VEX IQ 所認證的零件，本次研究機器人所使用的零件包含 1.基本結構材料、2.控制與感應器設備。基本結構包含積木、輪軸、連接器、插銷、馬達與車輪(圖 4 所示)。

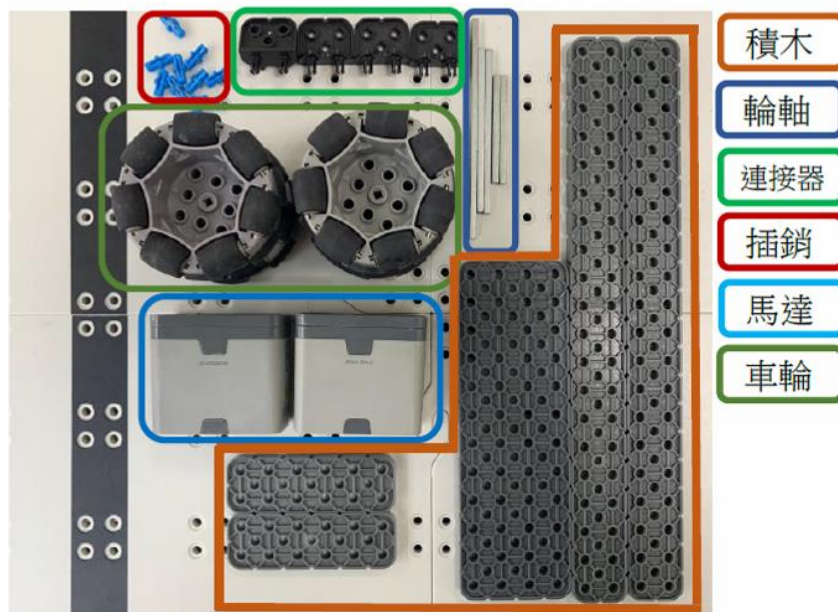


圖 4.基本結構材料

而控制與感應器材料部份則包含遙控器、LED 燈、陀螺儀、主機、電池與傳輸線(圖 5 所示)，透過這些零件完成機器人組裝與相關訊號傳輸與讀取，完成編程式指定的任務。

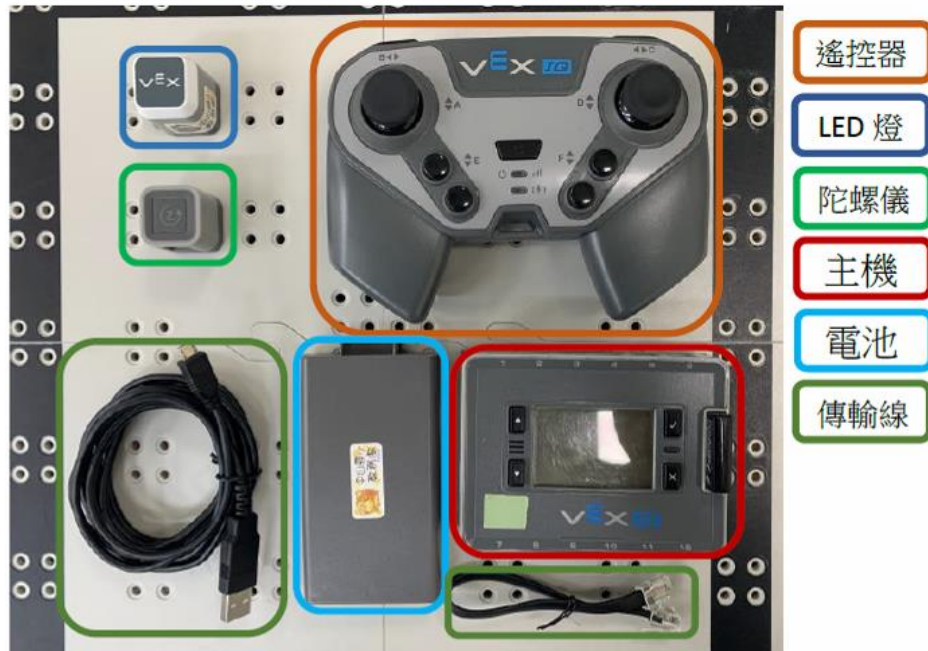


圖 5. 控制與感應器設備

### 三、編寫程式軟、硬體設備:

本次研究主要探究「程式編程」與「使用外部感應器」對於「機器人表現」成效評估，所使用的編程軟體為 VEX 官方釋出的 VEXcode IQ V2.0.5 版本<sup>2</sup>進行本次研究程式編程的軟體(如圖 6 所示)。

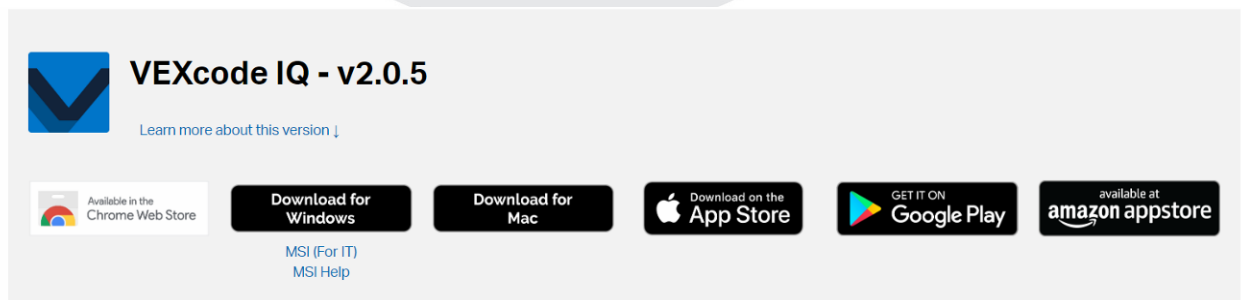


圖 6. VEXcode IQ 官網截圖

<sup>2</sup> 官方 VEXcode IQ 下載網址: <https://www.vexrobotics.com/vexcode-download>

本次所使用的資訊設備為 ASUS 公司生產型號 C214M 的 Chromebook<sup>3</sup>擔任程式編程資訊設備，所以下載的 VEXcode IQ 版本為與 Chrome 相容作業系統版本。實際 VEXcode IQ 操作與 C214M 請參考圖 7 所示。

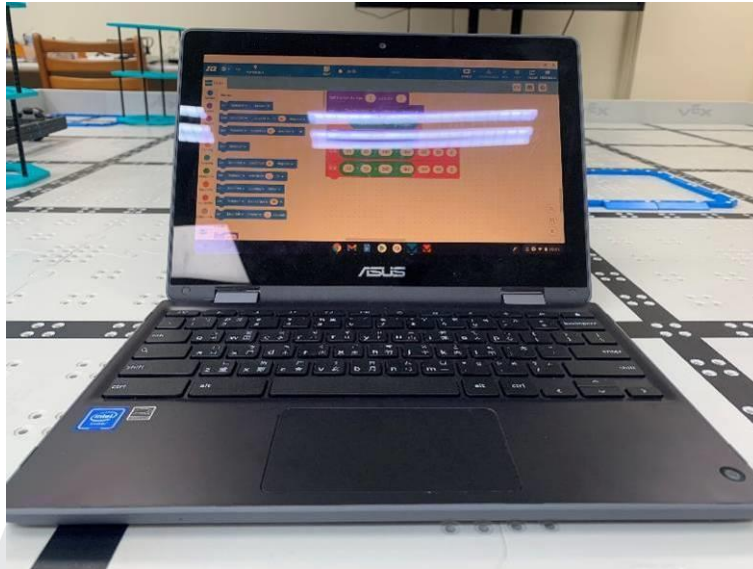


圖 7. C214M 與 VEXcode IQ 操作示意圖

## 肆、研究過程與方法

### ● 研究設計

本次研究主要透過比較三種不同的編程方式:1.單純使用程式、2.配合外部感應器(陀螺儀, Gyro sensor)的協助與 3.配合外部感應器(陀螺儀, Gyro sensor)加上比例控制(Proportional control)方式,進行機器人完成任務成效評估。本章節將針對 1.實驗設計 2.程式設計(VEXcode IQ)兩部分說明。

#### 一、實驗設計

由於正式比賽中技能挑戰賽只有自動控制賽(Program)需要撰寫程式,所以研究的過程與方法主要著重在自動控制賽的比賽方式。為了降低比賽的變因,所以我們針對今年比賽主題進行簡化,並且將機器人行進的路線固定以減少誤差。

研究的過程主要紀錄機器人實際位置與預設的方向之間的誤差角度,並且紀錄是否有將 riser 推入得分框完成任務。為了能夠讓研究更精準,每一種編程方式都需要完成 20 次資料收集,為了讓研究變因降到最低,我們採用每一輪都需要完成這三

<sup>3</sup> 相關硬體規格請參閱 ASUS 官網: <https://smb.asus.com/product/intro/26>

種程式，並且每輪資料收集之前都採用抽籤的方式決定三種方式的先後順序，然後依照每一輪的順序執行任務與收集資料，最後完成 20 輪的資料收集，也就是每種方式收集 20 次。

在資料收集過程中機器人主要轉彎方式採用直角(90 度)方式，機器人移動方式採用直線行進方式，我們在過程中設立 4 個資料收集點方便我們進行資料收集(圖 8 所示)。在第 2、3、4 測量點進行角度資料收集後，機器人需要再完成將 riser 推入得分框的任務，是否能夠將 riser 推入得分框也將一併紀錄，最後計算成功率。

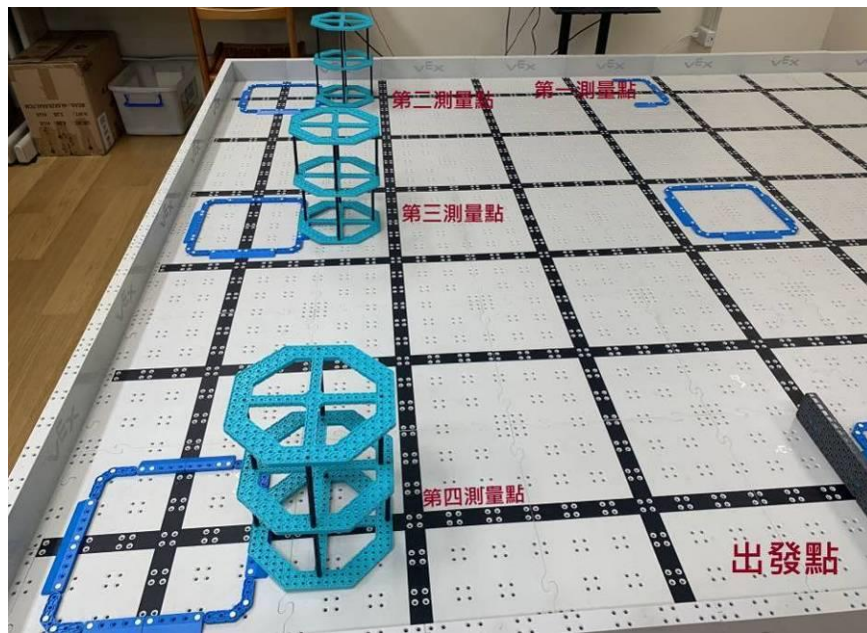


圖 8. 資料收集測量點位置圖

綜合上述的研究流程我們繪製實驗設計圖(圖 9 所示)，本次研究變因組合有三種方式，分別為單純程式(不使用 Gyro 與 PID 控制)，第二種為單純程式加使用 Gyro，第三種則是單純程式加使用 Gyro 加 PID 控制<sup>4</sup>。

<sup>4</sup> 為統一文章內各種方法名稱將第一種方式命名為「程式」，第二種名稱為「程式+陀螺儀」，第三種名稱為「程式+陀螺儀+比例控制」。

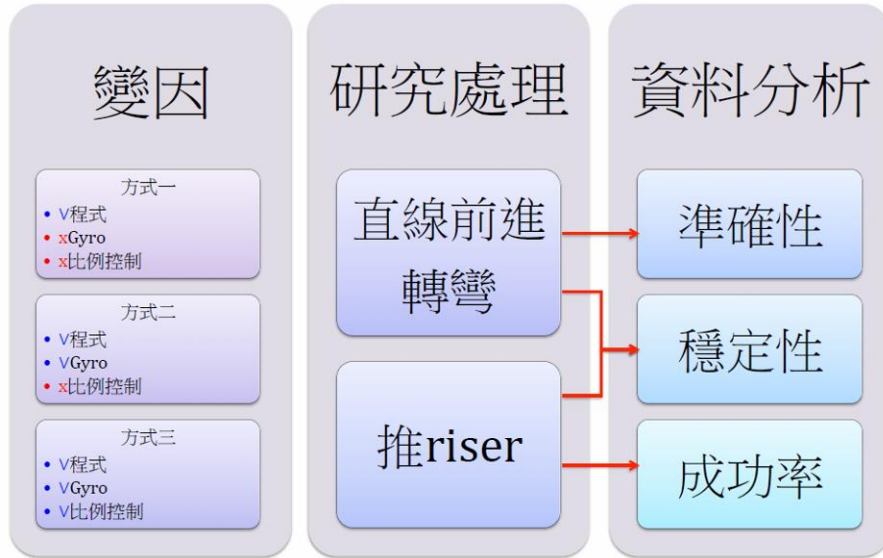


圖 9. 實驗設計圖

研究處理則是設計機器人行走路線與任務，本次研究主要探討機器人偏移角度，與是否能夠順利執行任務(將 riser 推進得分框)。資料收集則是提供評估機器人表現，包含穩定性、準確性與任務完成的成功率。經過事先計畫我們將機器人的行走路徑與資料收集位置進行測量並規劃行徑路線與任務工作，繪製成圖 10 機器人工作任務流程圖，透過統一的流程降低研究的誤差。

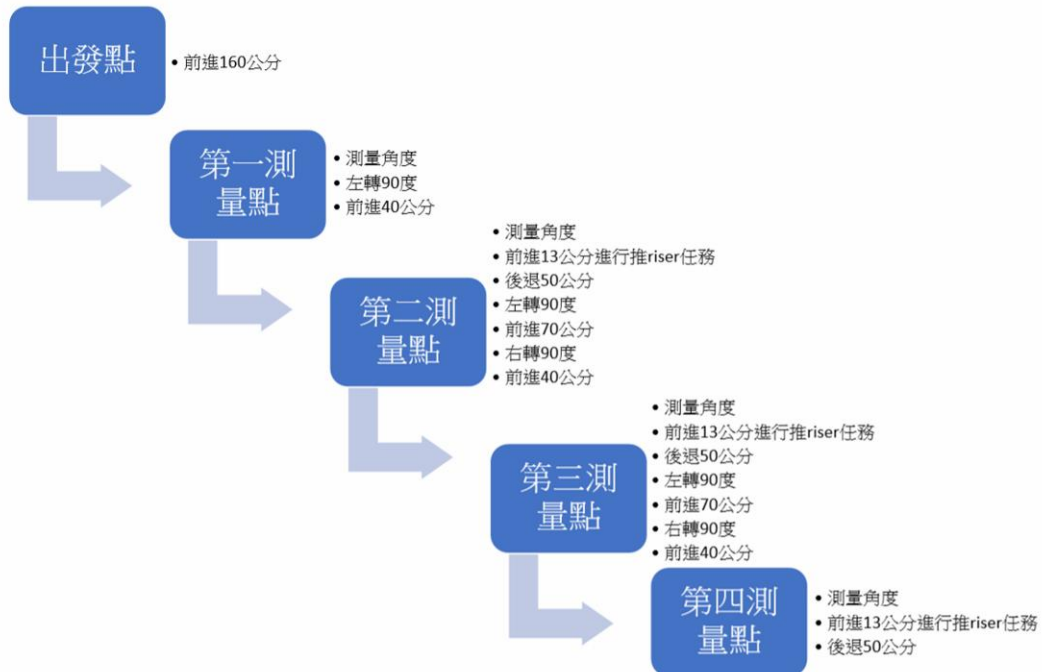


圖 10.機器人工作任務流程圖

針對偏移角度資料收集則是記錄機器人在四個測量點與原訂目標的角度差異進行紀錄，針對任務完成的成功率則是採計機器人是否能將 riser 推進得分框，成功與否則是機器人將 riser 推進得分框後 riser 不可以與得分框有任何碰觸，我們將成功與失敗的案例繪製成圖 11 進行圖解。

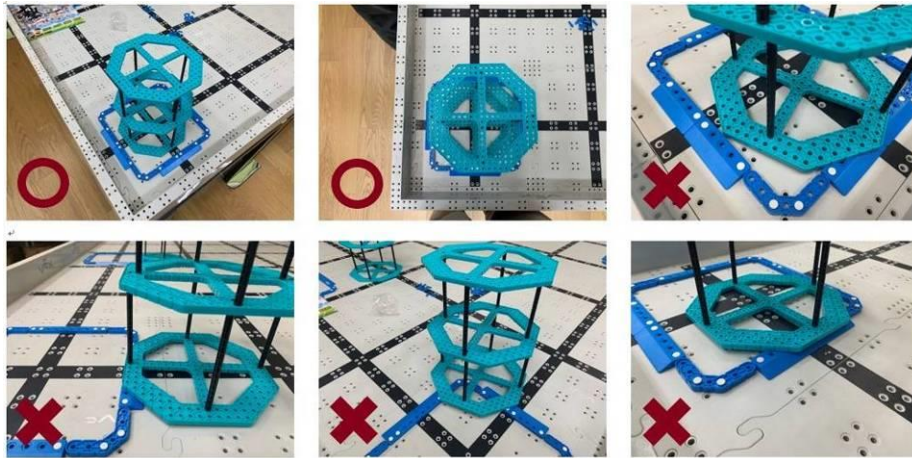


圖 11. 任務成功與失敗示意圖

本次研究中我們使用兩種外部感應器，一種是「陀螺儀」(Gyro sensor)(圖 12 所示)，另一種是「觸控 LED」燈(touch LED)(圖 13 所示)，「陀螺儀」主要是協助測量方位角度，由於在每一個測量點的如果要進行機器人角度測量需要讓機器人停止，如果將每一個測量點都獨立寫成一個程式，這樣下來就需要寫四組程式，由於機器人的主機只可以容納四組程式，這樣一來就無法達成隨機選取測量方式，此外，在切換四組程式的時候需要按到主機按鈕，容易造成機器人移動影響資料收集。

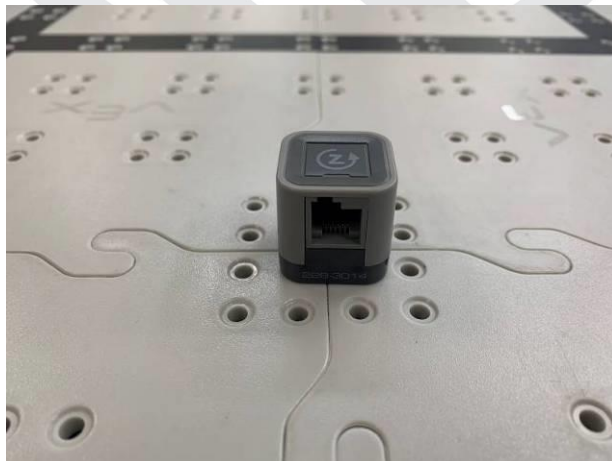


圖 12. 陀螺儀(Gyro sensor)



圖 13. 觸控 LED 燈(touch LED)

如果配合觸控 LED 燈的使用就可以當成一個動作來控制指令(開始)，而且利用燈號顏色可以知道程式是否已經完成或是正在進行。有關觸控 LED 燈的使用我們利用圖 14 的程式塊進行說明。

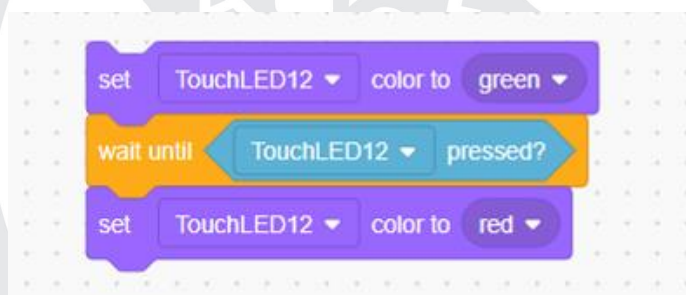


圖 14. 觸控 LED 燈程式塊

從圖 14 第一個程式塊可以看到我們讓 LED 燈亮「綠燈」，主要的目的就是「確認先前的程式已經執行完畢」，第二個程式塊就是「等待 LED 燈被碰觸」，主要的目的就是讓我們可以在測量點完成角度測量，當測量完成後就可以透過碰觸 LED 燈「直接啟動機器人」進行下一個程式。第三個程式塊我們設定 LED 燈亮「紅燈」，這個目的就是讓我們確認接下來的程式已經開始執行。

一開始測量角度時是採用量角器來測量，不過容易碰觸到機器人造成誤差，後來我們決定透過程式的協助(圖 15 所示)，直接撰寫程式將量測的角度數值直接呈現在主機螢幕上(圖 16 所示)，一方面方便我們讀取，另一方面可以降低在測量時碰觸到機器人所產生的研究誤差。

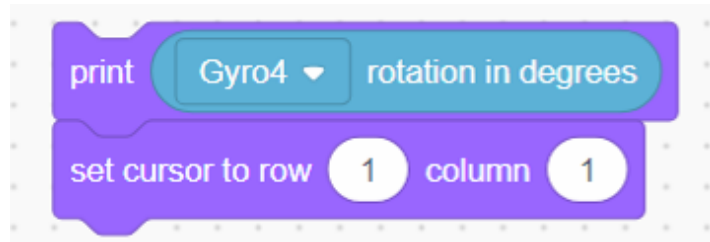


圖 15. 顯示機器人角度程式塊

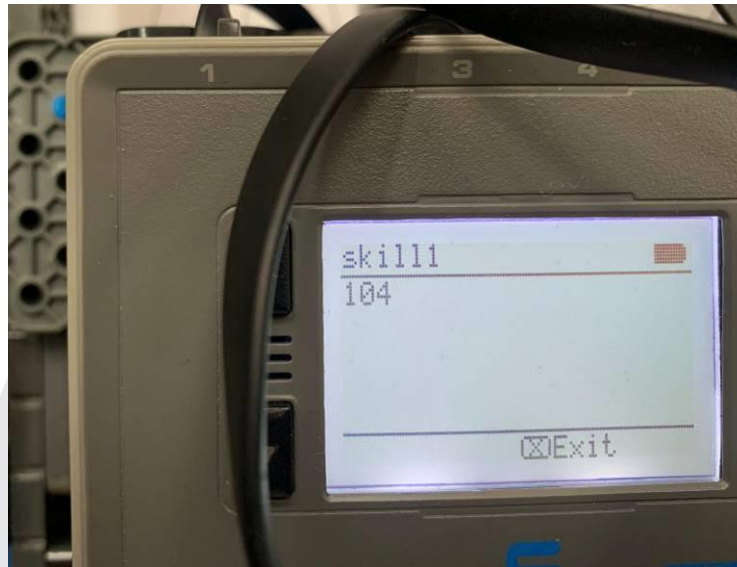


圖 16. 主機顯示機器人角度示意圖

機器人最後的組裝我們採用最簡單的結構方式來組裝，主要就是完成底盤設計，我們底盤設計採用最簡單的結構，完成後的研究底盤如圖 17 所示，底盤主要包含主機、萬向輪(減少摩擦力)、陀螺儀、觸控 LED 燈、馬達裝置，底盤完成後接著就是程式設計。

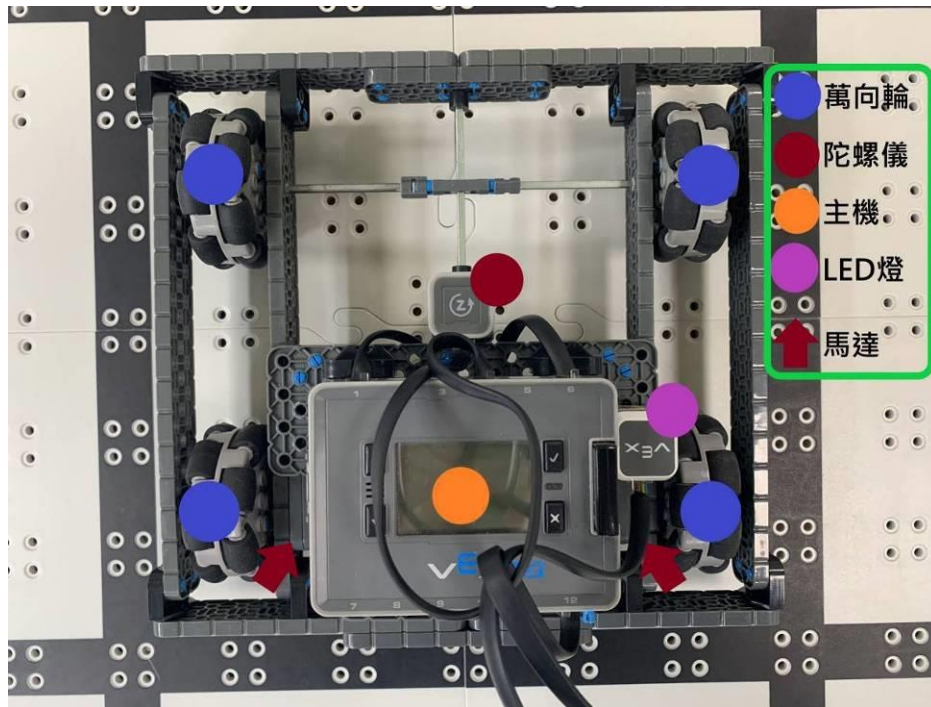


圖 17. 機器人底盤結構圖

## 二、程式設計(VEXcode IQ)

本次研究所使用的軟體為VEXcode IQ，分別設計三種程式進行資料收集，分別為1. 「程式」、2. 「程式+陀螺儀」(配合外部感應器「陀螺儀」，Gyro sensor的協助)與3. 「程式+陀螺儀+比例控制」(配合外部感應器「陀螺儀」，Gyro sensor 加上比例控制Proportional control)。以下將針對第一、第二種程式進行說明。這兩種程式主要差別在有沒有「陀螺儀」，在主要程式設計是一樣的。

第一種與第二種程式主要差別在於「陀螺儀」的使用，第一種方式我們將「陀螺儀」連結到主機第四連結孔，不過沒有編寫設定「陀螺儀」功能程式，第二種方式則是在驅動模式下(Drivetrain)設定「陀螺儀」，這樣「陀螺儀」就利用內建的程式執行。第三種方式則是採用陀螺儀與比例控制方式來完成執行。

總結上述，本次研究主要探究三種不同程式與感應器組合，透過四個測量點記錄機器人偏移角度，與是否成功將 riser 推入得分區，評估本次研究的結果並找出最佳組合，並將結果應用在我們未來 VEX 機器人競賽。

### ● 結果與討論

在資料收集上面如果測量時機器人偏右我們以負值加以記錄，如果偏左的話則是

以正值紀錄，如果與原來方向一致(90)度角沒有偏差則紀錄值為 0。本次研究是模擬正式自動控制賽，一旦開始進行資料收集，研究者是不能針對機器人進行修正，換言之在執行任務過程中這些偏差的角度是會累加。研究結果將針對本次研究所探討的 1.準確性、2.成功率與 3.穩定性進行說明。

### 一、準確性

經過資料收集後，我們分別將三種程式組合在四個測量點所測得的角度針對偏左、無偏移、偏右發生次數進行統計並繪製成表 1 進行說明。

表 1.不同方式各測量點偏差角度比較表

	第一測量點			第二測量點			第三測量點			第四測量點		
	偏左	無偏移	偏右	偏左	無偏移	偏右	偏左	無偏移	偏右	偏左	無偏移	偏右
單純程式	8	4	8	19	0	1	20	0	0	20	0	0
程式+陀螺儀	6	8	6	16	3	1	3	3	14	9	8	3
程式+陀螺儀+比例控制	0	20	0	0	20	0	0	20	0	0	20	0

從表一可以看到第三種方式「程式+陀螺儀+比例控制」組合在每一個測量點都能完成 20 次精準無偏移達到預設的方位角度，無誤差狀況總共 80 次。

第二種方式「程式+陀螺儀」在第一測量點無偏移 8 次，第二測量點無偏移 3 次，第三測量點無偏移 3 次，第四測量點 8 無偏移 8 次，無偏移狀況共計 20 次。

第一種方式「程式」方式無偏移僅有發生在第一測量點 4 次，其餘測量點皆產生偏移現象。為了清楚呈現每個方式在各個測量點的偏移角度值，我們將測量的角度依照以下方式記錄：偏左的話則是以正值紀錄，如果與原來方向一致(90)度角沒有偏移則紀錄值為 0，偏右的話則是以負值紀錄，並製作成圖 28。

為了清楚將三種方式進行比較，在圖 28 之中我們以英文字 A 代表「程式」方式，以 B 代表「程式+陀螺儀」，以 C 代表「程式+陀螺儀+比例控制」方式，英文字後面的數字代表測量點順序，透過這個方式將三種結果呈現在圖 18。

從圖 18 中可以看到 A 方式機器人偏移的角度介於-2~24 度之間，仔細觀察每一個測量點偏差角度，在第一測量點時偏移角度介於-2~3 度之間。第二測量點偏移

角度介於-4~12 度之間，到了第三測量點偏移的角度介於 8~16 度之間，到了第四測量點則偏移角度介於 10~24 度之間。

從圖 18 來看透過單純程式來完成任務與我們先前所遇到的狀況一樣，只要一有偏離則無法挽回，而且越到後面偏離的角度越大。這個現象顯示機器人如果受到場地或機器人本身狀況影響持續產生角度偏差，這些偏差是可能累計，最後造成如此大角度的偏差。

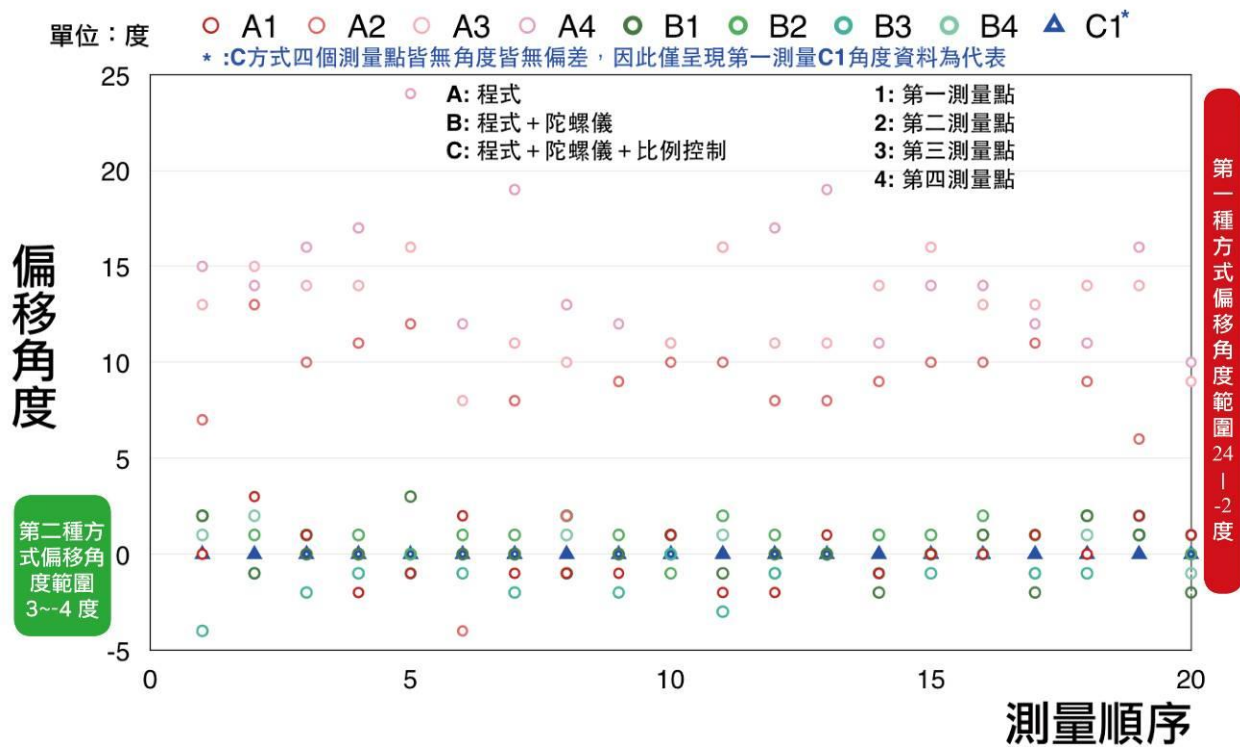


圖 28. 單純程式方式在不同測量點偏移角度折線分析圖

第二種方式(B)則是「程式+陀螺儀」，從圖 28 來看有陀螺儀協助的機器人的偏移角度介於-4~3 度之間，偏離情況比第一種方式小。在第一個測量點表現上偏離角度介於-2~3 度之間，在第二個測量點則偏移的角度介於-1~2 度之間，第三個測量點則偏移的角度介於-4~2 度之間，第四個測量點的偏移角度介於-1~2 度之間。第二種方式藉由陀螺儀的協助機器人可以自行調整偏離的角度，不過還是有角度偏離的現象發生。

第三種方式(C)則是「程式+陀螺儀+比例控制」進行資料收集，從圖 30 來看在不同測量點所測量到的偏離角度值都為 0，這個結果顯示透過上述組合方式能夠有

效準確控制機器人前進的方向。

從上述準確性比較來看，第三種方式的準確性最好，第二種方式次之，如果只單純使用程式撰寫的方式準確性最差。第一種方式機器人容易受到場地的影響，並且偏差角度受到累加的影響，造成最後偏離角度最大的現象。

## 二、成功率

本次研究除了探討機器人偏移角度之外，還要探討完成任務的成功率，經過上述的數據分析，我們也記錄是否能夠將 riser 成功推入三個得分框，經過分析每一組研究組合每一次需要需要完成 3 次推 riser 任務，在 20 次測試後每一組需要完成 60 次的任務。我們將三種研究方式成功將 riser 推入得分框的次數進行分析會製成表 2。表 2.不同方式推入 riser 成功率比較表

	第二測量點	第三測量點	第四測量點
單純程式	0	0	0
程式+陀螺儀	95%	85%	90%
程式+陀螺儀+比例控制	100%	100%	100%

從表 2 來看，第一種只用「程式」的方式成功率為 0，如果搭配「陀螺儀」協助的話成功率可以達到 85%-95%，如果再加上「比例控制」的協助則成功率可以達 100%。從成功率來看，機器人產角度偏移後對於接下來的需要完成的任務就會產生影響，如果機器人無法自行修正與調整偏移角度的話，對於執行任務的成功率就會大打折扣。

## 三、穩定性

上述成功率分析中雖然第二種方式「程式+陀螺儀」的成功率介於 85%-95%之間，但是我們想進一步探討是否能夠在某種偏移角度範圍仍然可以成功完成任務，因此我們將每個測量點偏移角度進行平均值計算，並製作成圖 19。從圖 19 可以看到第一種方式「程式」第一測量點為 0.05 度，第二測量點為 8 度，第三測量點為 12.6 度，第四測量點為 14.65 度。

第二種「程式+陀螺儀」在第一測量點為 0.05 度，第二測量點為 1 度，第三測量點為-0.85 度，第四測量點為 0.35 度。

第三種「程式+陀螺儀+比例控制」的表現在每個測量點的偏移角度平均值為 0。透過圖 19 與表 2 結果來看第二種「程式+陀螺儀」穩定性，在第二測量點時當平均偏移角度越大(1 度)時成功率高(95%)，在第三測量點平均偏移角度-0.85 時成功率最低 (85%)，在第四測量點時平均偏移角度最小(0.35 度)成功率為 90%。如果從第三種方法「程式+陀螺儀+比例控制」來看當偏移角度越低成功率越高，不過從第二種方法「程式+陀螺儀」表現來看似乎與這個看法有所出入，因為在第四個測量點平均偏移角度為 0.35 度明顯優於第一測量點(1 度)，可是成功率卻比第二測量點低。

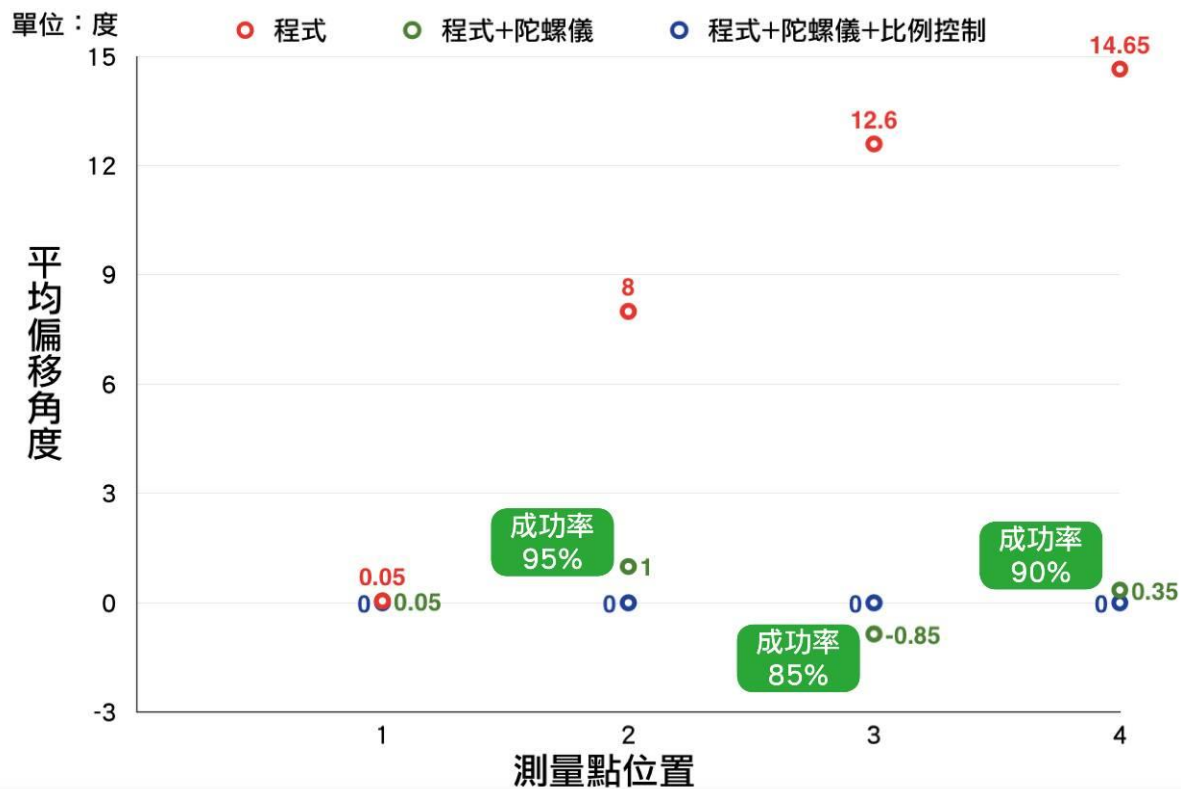


圖 19. 不同方式在不同測量點偏移角度平均值折線分析圖

為了進一步分析第二種方式的穩定性，我們將探討第二、三、四測量點的偏移角度與成功與否的關係，由於第二種方法偏移角度介於偏左 2 度~偏右 4 度，所以我們僅針對這個區間進行製作成表 3。從表 3 來看：當機器人偏移角度為 0 度的時候成功率達 100%，當偏左 1 度時發生一次無法將 riser 成功推入得分框，偏右 1 度時也發生 1 次失敗，當偏右 2 度時成功率為 100%，當偏左 2 度時發生兩次失敗，當偏右 3 與 4 度時則成功率為 0。從表 3 來看，當偏右角度大於 3 度以上時，成功率為 0，

但是在偏左 2 度~偏右 2 度之間，扣除移角度為 0 情況之外，其成功率雖然高但是還是存在不可預測的風險，穩定性還是比第三種「程式+陀螺儀+比例控制」方式的穩定性低。

表 3.第二種方式「程式+陀螺儀」偏移角度與失敗次數分析表

		左 2	左 1	0	右 1	右 2	右 3	右 4
第 2 測量點	發生次數	5	11	3	1	0	0	0
	失敗次數(%)	1(20%)						
第 3 測量點	發生次數	2	1	3	9	3	1	1
	失敗次數(%)				1(11%)		1(100%)	
第 4 測量點	發生次數	1	8	8	3	0	0	0
	失敗次數(%)	1(100%)	1(12.5%)					

綜合所述，透過第二、三、四測量點平均偏移角度與成功率進行穩定性分析，第一種方式「程式」穩定性最不好，第三種方式「程式+陀螺儀+比例控制」穩定性最好。第二種方式「程式+陀螺儀」方式的穩定性介於中間。由於機器人在比賽中僅有的3次嘗試機會，我們希望機器人在每次嘗試機會都能成功，這也是我們要求機器人穩定性的原因。

本次研究結果顯示在第一種只有採用 VEXcode IQ「程式」撰寫的方式執行路徑與推 riser 任務，機器人偏移的角度介於-2~24 度之間，在第一測量點時偏移角度介於-2~3 度之間。第二測量點偏移角度介於-4~12 度之間，到了第三測量點偏移的角度介於 8~16 度之間，到了第四測量點則偏移角度介於 10~24 度之間。從數據來看機器人一直往左邊偏斜，這可能與機器人無法隨時修正或是保持移動角度有關，這個現象顯示如果單純使用這個方式，一旦機器人受到 riser 撞擊影響導致角度偏移、或是場地摩擦力、輪胎等變因的影響，偏移的現象無法修正，多次偏移將持續惡化嚴重造成自動控制賽比較容易失敗導致低成功率與低穩定性。

第二種方式則是「程式+陀螺儀」，偏離情況比第一種方式小。在第一個測量點表現上偏離角度介於-2~3 度之間，在第二個測量點則偏移的角度介於-1~2 度之間，第三個測量點則偏移的角度介於-4~2 度之間，第四個測量點的偏移角度介於-1~2 度之間。單純從偏移角度來看明顯優於第一種方式，我們發現「陀螺儀」的使用主要當機器人在轉彎的時候會有修正的機制，雖然在直線前進時無法立即調整，還是導致在各測量點還是有些許的偏移現象。不論是撞擊 riser 產生的偏移、場地摩擦力

等因素影響，透過「陀螺儀」的協助大大改善偏移的現象，這個修正也提高將 riser 推入得分框的成功率，不過還是無法達成百分之百的成功率。在穩定性表現上第二種方式的平均偏移角度與成功率的關係無法穩定預測，所以穩定性次之。

第三種方式則是「程式+陀螺儀+比例控制」進行資料收集，這個方式在不同測量點所測量到的偏離角度值都為 0，透過事先將比例控制程式融入前進、後推與轉彎定義模塊的撰寫，提供機器人不論前進、後退、轉彎隨時修正的能力，讓機器人有效控制機器人前進的方向，進而提高百分之百的任務成功率，而且能夠穩定完成任務。

依照本次研究所探討機器人的「準確性」、「成功率」與「穩定性」結果繪製成表 4，從表 4 來看在「準確性」表現結果也是第三種「程式+陀螺儀+比例控制」優於第二種「程式+陀螺儀」，第三名為單純「程式」方法。

表 4. 不同方式機器人總體表現比較表

	第一名		第二名		第三名
準確性	程式+陀螺儀+比例控制	>	程式+陀螺儀	>	單純程式
成功率	程式+陀螺儀+比例控制	>	程式+陀螺儀	>	單純程式
穩定性	程式+陀螺儀+比例控制	>	程式+陀螺儀	>	單純程式

成功率最高為「程式+陀螺儀+比例控制」，其次為「程式+陀螺儀」，成功率最低則是單純「程式」方法。在穩定表現上第三種「程式+陀螺儀+比例控制」優於第二種「程式+陀螺儀」，第三名為單純使用「程式」方法。

## ● 未來展望

本次研究結果顯示如果單純使用我們先前的方式不利用其他「感應器」協助，我們在 VEX 機器人比賽中，尤其是自動控制賽項目中我們可說毫無勝算。如果搭配「陀螺儀」的協助，雖然可以提高「準確性」、「成功率」與「穩定性」，不過在「成功率」與「穩定性」還是有改進的空間，尤其機器人比賽僅有 3 次自動控制賽嘗試的機會，機器人需要穩定表現，不可以起伏不定。

此外比賽中不光是移動機器人而已，每年的比賽都有不同的任務，例如今年比賽就不光是需要將 riser 推入得分框，還要設計抬升手臂將 riser 堆疊在一起，此外機

器人自動控制賽需要在一分鐘完成任務，提高機器人移動速度等種種的因素讓我們對於機器人的「穩定性」要求需要更高。

目前來看第三種方式在現有的「感應器」協助下加入「比例控制」協助，能大大提高機器人「準確性」、「成功率」與「穩定性」，這個研究組合讓我們信心大增，透過這個方式不但解決我們機器人角度偏移的問題，也讓我們的機器人在轉彎、後退時都可以透過「比例控制」協助進行修正並保持我們所設定的目標前進。

本次研究中我們閱讀與瀏覽相關的參考資料，發現 PID 控制除了廣泛用於工業自動控制產品與設計，例如恆溫控制、水量控制等。隨著科技發展也應用在無人機或空拍機盤旋(hovering)功能之中。在未來相信結合外部感應器與 PID 控制的科技應用將蓬勃發展，例如 google 正在研發無人自駕車，正是展望未來世界提前部屬。

在 2021-22 的 VEX 機器人比賽我們會將「陀螺儀」與「比例控制」使用在「自動控制賽」的「程式」撰寫，當然「比例控制」還有改進的空間，例如需要較長的修正時間與突發事件的修正機制，我們會持續探究「積分控制」(Integral control)與「微分控制」(Derivative control)，透過這些方式維持我們機器人的性能與表現，最後能夠獲得 VEX 機器人比賽中 Excellent Award 最高榮耀。

## 參考文獻

### • 中文部分

黃冠渝、黃英哲（2018 年 10 月 18 日）。自駕車置中行駛的科技：PID 控制——讓自駕車各行其道(一)。科技大觀園。取自 <https://scitechvista.nat.gov.tw/c/sTCZ.htm>。

黃冠渝、黃英哲（2018 年 10 月 18 日）。自駕車置中行駛的科技：PID 控制——讓自駕車各行其道(二)。科技大觀園。取自 <https://scitechvista.nat.gov.tw/c/sTCB.htm>。

什麼是 PID 控制？兩分鐘快速理解 PID 的本質！（2020年3月22日）。圖靈雞科技俱樂部。取自 <https://www.youtube.com/watch?v=VRUSG7G58yY>。