淺談 PID 自動控制與編程

黃昭銘 [1*] 黃昱誠 [2]
[1] 宜蘭縣中山國民小學
[2] 國立宜蘭高級中學

壹、前言

12 由於人類知識快速發展，跨學科的整合與應用能力越來越受重視，例如美國推動的 STEM 教育便是一例。新課綱強調學科的整合、創新與應用能力，科技發展與應用更是不可或缺的重要一環，透過資訊科技的學習、應用、整合，藉由科技理解與跨學科學習協助學習者探究未來趨勢與生涯探索規劃。PID 自動控制為工業上常用的控制系統之一，筆者長期參加機器人比賽常利用 PID 自動控制系統讓機器人在比賽中可以穩定運作，但是 PID 的控制原理過於複雜，對於國中、小的學生比較不容易理解其中運作概念，本文藉由多年學習與撰寫 PID 程式的經驗，從介紹 PID 發展歷史、 PID 自動控制概念與實際撰寫三方面出發，透過淺顯易懂的文章介紹，讓更多人能夠對 PID 控制有初步的認識，並且能夠利用 Blocks 程式完成 PID 自動控制程式。

貳、PID 發展簡介

何謂 PID 控制？在自動控制領域中 PID 為常用的自動控制系統一種操作模式。PID 控制也就是將三種不同的控制演算法結合在一起的統稱，他們分別為最基礎的「比例控制」(P 控制) (Proportional control)、「積分控制」(I 控制) (Integral control)、和「微分控制」(D 控制) (Derivative control) (黃冠渝、黃英哲，2018a)。

在 Borase, Maghade, Sondkar & Pawar(2020)回顧 PID 的文獻中指出 PID 控制在自動控制領域應用有很長的歷史，可以追溯到瓦特在 1769 年改良蒸汽機與調節器(Governor)。瓦特所改良的蒸汽機與調節器是首次應用所謂回饋(feedback)概念的裝置。當時這些概念尚未形成正式的 PID 控制的原始架構。PID 的概念一直到 1922 年 Nicholas Minorskygk 為美國海軍設計船隻自動轉向航行設計才正式提出 PID 概念(Bennett, 1996, H¨agglund, & Guzm´an, 2024) 。

*為本文通訊作者

PID 控制的原理其實是由工程師觀察舵手的動作而來，由於舵手在調整船隻的方向時不僅需要根據當前的誤差，也會需要考慮過去以及未來的誤差來做調整，才能使得船隻穩定航行。

在 Borase, Maghade, Sondkar & Pawar(2020)一文中將 PID 的架構整理並繪製成圖 1。從圖 1 中可看到在自動控制過程中，當設定開始進行時(Set point)，針對與原來設定值的差異(Error)，然後透過比例、積分、微分的演算法，計算出與預測量的數值，然後回饋給 Control Output，最後進行後續任務調整。

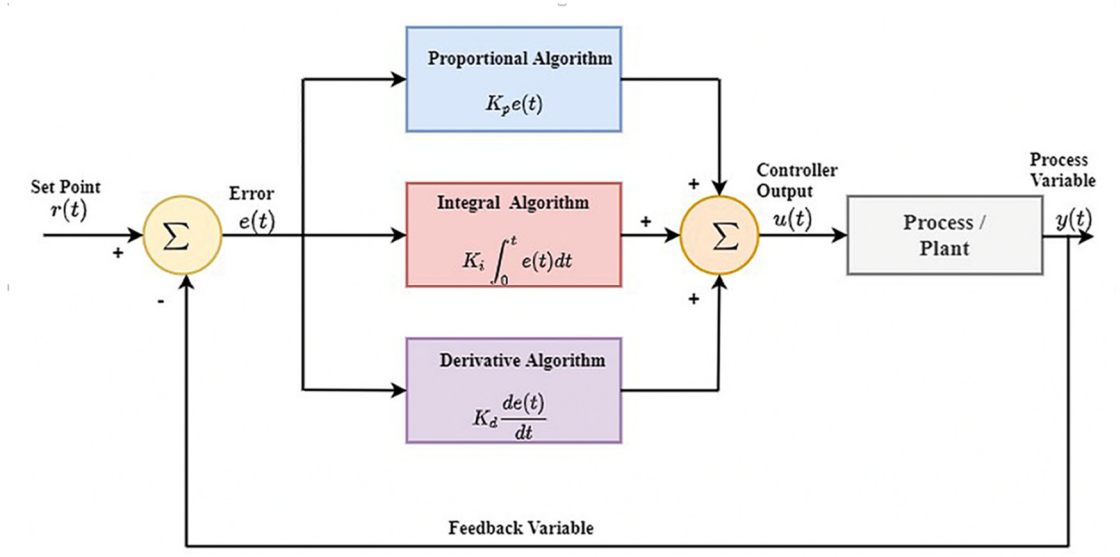

圖 1：PID 自動控制架構圖

資料來源: Borase , RP., Maghade, D. K., Sondkar, & Pawar, S.N. (2020). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control,9,* pp820.

PID 的運作理論牽涉到複雜的演算公式，為了能夠簡單介紹 PID，以下就以船隻航行為例，當船隻今天必須要直線航行，那麼當舵手發現船隻可能產生偏移的時候，就需要調整舵的轉向。如果只有偏移一點，那就可以只做些微的調整；如果偏移很多，那就調整多一點，這也就是所謂的 P 控制（比例控制）。根據誤差（error）乘上一個常數 Kp 來調整控制器輸出，使得系統可以動態進行大致上的調整。

為了清楚呈現 PID control 的不同表現，參考並引用 Zurich Instruments 網站上公布的白皮書[White paper](Zurich Instruments, 2023)中的圖片資料輔助說明 P control、PI control、PID control 這三種的表現。

以船隻轉向為例，如果今天船隻需要右轉九十度，由於一開始的誤差很大，所以 P 控制就可以起到很大的作用，使得船身快速向右旋轉，可是當轉到八十度左右時，這時候誤

差就變小很多，由於水和周遭的空氣可能會產生一些阻力，使得 P 控制所提供的動力不足以使得船身進行轉向，這時「船身當前轉向角度」和「目標轉向角度」間的角度差就是所謂的「穩態誤差」。

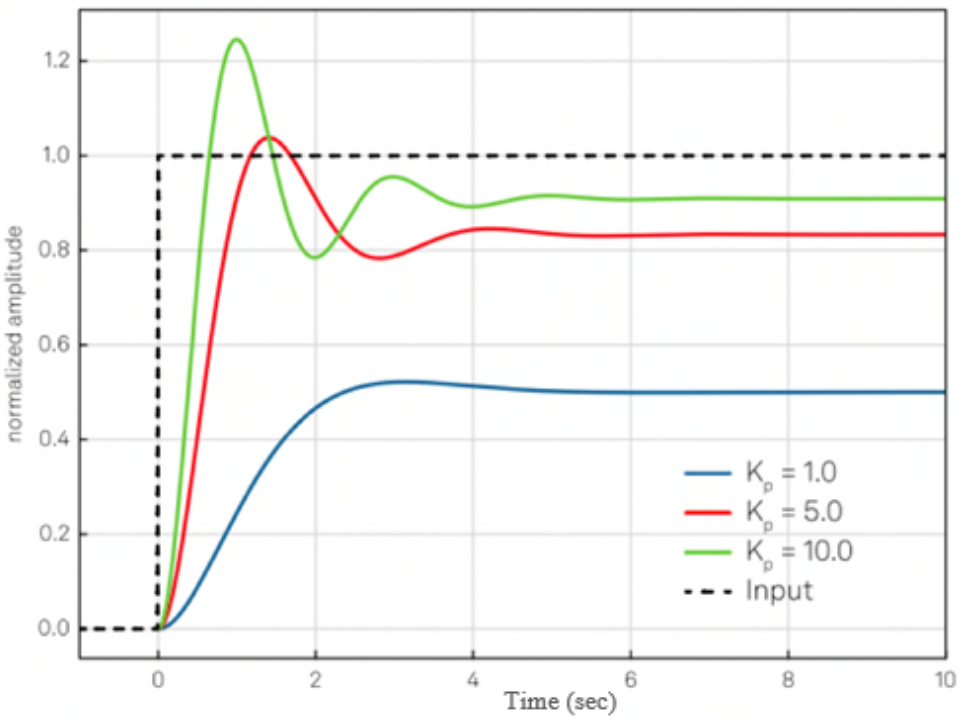

圖 2：P 控制運作介紹圖

資料來源： Zurich Instruments. (2023). Principles of PID Controllers. [White paper]. Zurich Instruments. https://www.zhinst.com/others/en/resources/principles-of-pid-controllers

為了消除穩態誤差，需要持續將舵的角度拉大，使動力提升直到足以讓船身轉動，這種方法就是 I 控制（積分控制），透過將誤差持續累加後乘上一個常數 Ki 來調整輸出，藉此消除穩態誤差(圖 3 所示)。

從圖 3 來看，假設橫坐標為時間單位為秒，當 Kp=1 的時候，如果透過不同 Ki 值的設定，在自動控制整個修正的過程也會有所不同，從圖 3 來看當作用時間 12.5 秒以後，綠色(Ki=1)與紅色(Ki=0.5)的表現已經達到 Set point，顯示適當的 Ki 值可以較快消除先前的穩態誤差。

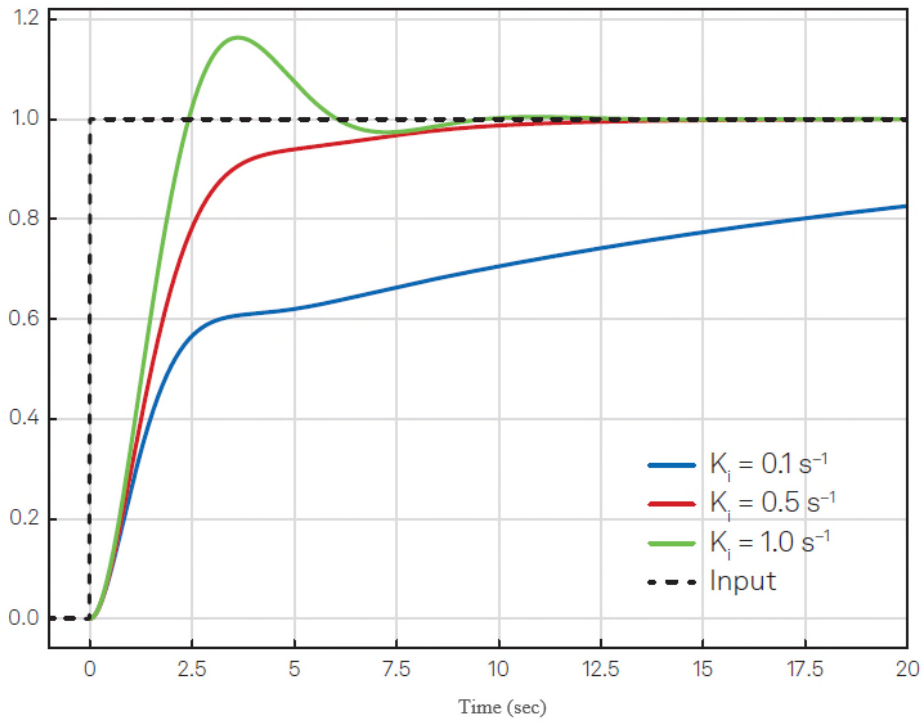

圖 3：PI 控制運作介紹圖

資料來源: Zurich Instruments. (2023). Principles of PID Controllers. [White paper]. Zurich Instruments. https://www.zhinst.com/others/en/resources/principles-of-pid-controllers

而當船身在旋轉時，由於慣性和前兩種控制方式的關係，可能使得船身轉超過目標的角度，雖然在船這個例子來看可能沒什麼太大的問題，但有些地方可不能有超過的情況發生，像是手機貼膜，如果誤差是目前貼膜和手機之間的距離，那麼只要移動超過就是貼膜會直接施力壓在手機上，這種問題可不被允許，所以這時就需要 D 控制（微分控制）來提供一個阻尼避免誤差的發散太大，使得整體可以更快達到目標（收斂）(圖 4 所示)。

從圖 4 來看，假設橫坐標為時間單位為秒，當設定相同的 Kp=4、Ki=1，在 PID 模式下自動控制整個修正的過程也會有所不同，從圖 4 範例來看，沒有 Kd 協助下自動控制的船頭擺動比較多(藍色線條)，往往需要累積一定的誤差才會進行調整，反觀當 Kd=2 的時候船頭的擺動比較小(紅色線條)，主要是 Kd 可以依照先前誤差值數據的累積進行預測並進行作用，防止未來可能過衝或是意料之外的變化。

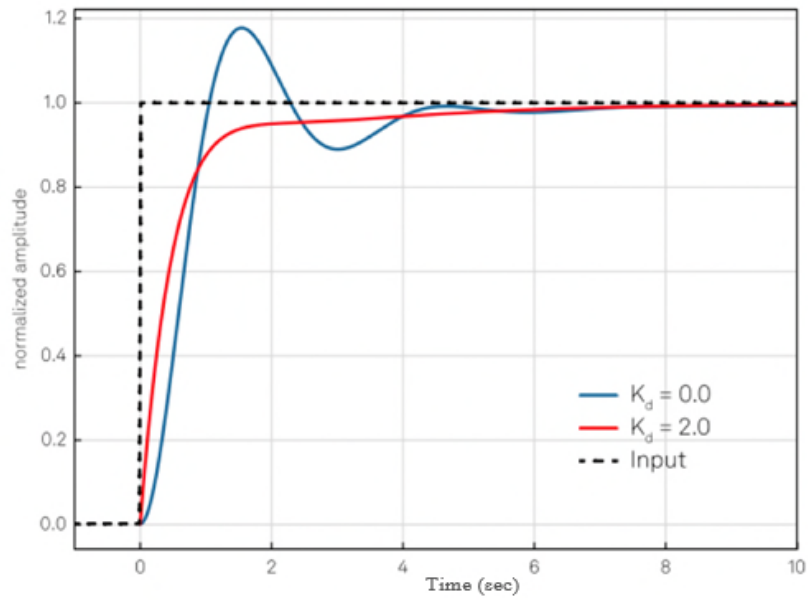

圖 4：PID 控制運作介紹圖

資料來源： Zurich Instruments. (2023). Principles of PID Controllers. [White paper]. Zurich Instruments. https://www.zhinst.com/others/en/resources/principles-of-pid-controllers

為了比較沒有導入 PID 自動控制、導入 P control、導入 PI control 與導入 PID control 的差別，參考網站資料(akYtec GmbH)中的比較說明圖(圖 5 所示)。圖 5 是以恆溫控制為例，Y 軸為溫度，恆溫設定的溫度為 40 度，X 軸則是時間軸。

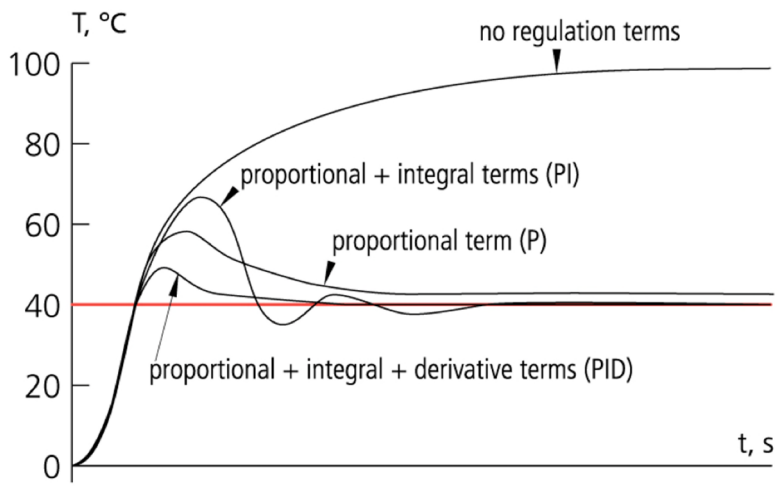

圖 5：PI，及 PID 控制比較說明圖

資料來源： https://d27h1xy6kx2vdd.cloudfront.net/downloads/AddInfoImages/pid/pid_graphEN.jpg

從圖 5 來看，沒有 PID 控制輔助 (no regulation term)，加熱器會一直進行加熱，最後造成溫度會一直上升到 100 度 C。如果透過 P 控制協助，雖然可以控制加熱器持續加溫，往往無法將溫度控制在所設定的 40 度。如果採用 PI 控制的話，一段時間後溫度可以達到設定的溫度，不過前期溫度會有高、低震盪變化現象，如果採用 PID 控制的話，溫度沒有震盪變化現象，而且溫度可以很快達到恆溫狀態。

PID 自動控制系統經過多年的應用與改進，在我們現代生活中的應用範圍非常廣泛，小從冷氣機恆溫控制、熱水器恆溫控制，複雜到汽車自動巡航控制 (Cruise control)、自動駕駛(黃冠渝、黃英哲，2018a)、先進駕駛輔助系統（Advanced Driver-Assistance System, ADAS）(Kukkala, Tunnell, Pasricha, & Bradley, 2018) ，無人機飛行都需要仰賴 PID 控制協助來完成預定的工作目標(Åström, & H¨agglund , 2001; H¨agglund, & Guzm´an, 2024)。

就 PID 控制(Proportional, Integral, Derivative control)設計上，「比例控制」(Proportional control)相對是比較簡單的設計，選擇適當的 Kp 值就可以執行任務；而「積分控制」(Integral control)則牽涉微積分的積分概念；而「微分控制」(Derivative control)是牽涉到微分概念，這兩種控制概念是比較複雜而且困難。

參、PID 撰寫範例

多年來有幸參加 VEX 機器人比賽，尤其自動賽過程中機器人要透過感測器來收集資料，然後進行直線前進、轉彎等任務，常常需要透過 PID 的輔助來完成上述任務，因此累積一些撰寫 PID 程式的經驗。目前國內國中、小學生在求學階段已經學會基礎的 Scratch 程式編程，接下來將利用 VEXcode V5 程式編輯軟體來示範如何利用 block 來完成簡單的 P control、PI control 與 PD control，分享透過 PID control 來修正機器人表現的準確性。

以機器人比賽的底盤為例，由於機器人底盤設計時，會在左右兩邊各有一顆以上的馬達，然後搭配慣性陀螺儀(Inertial Sensor)來偵測機器人偏移角度(angle)。為了讓這些數據能夠正常使用並減少誤差，首先需要將 Inertial sensor 進行歸零。如果使用其中一邊當做判斷是否達成後續條件，可能導致誤差的產生，為此我們需要先計算底盤兩邊馬達的平均移動距離。除此之外，由於移動程式可能是向後，因此旋轉的角度會是負數，因此我們需要加上絕對值(圖 6 所示)。

首先我們來講如何編寫直行程式，由於我們的修正程式會執行許多次，如果每次都從頭再寫一遍程式會太長，因此我們需要先去定義一個模塊，這樣未來使用時才會比較方便(圖 7 上方所示)。首先需要先設定一個迴圈，這個迴圈的目的是檢測機器人是否完成設定的目標，像是直行是否達到設定的距離，而迴圈內需要重複執行的正是我們的修正程式(圖 7 中央部分所示)。

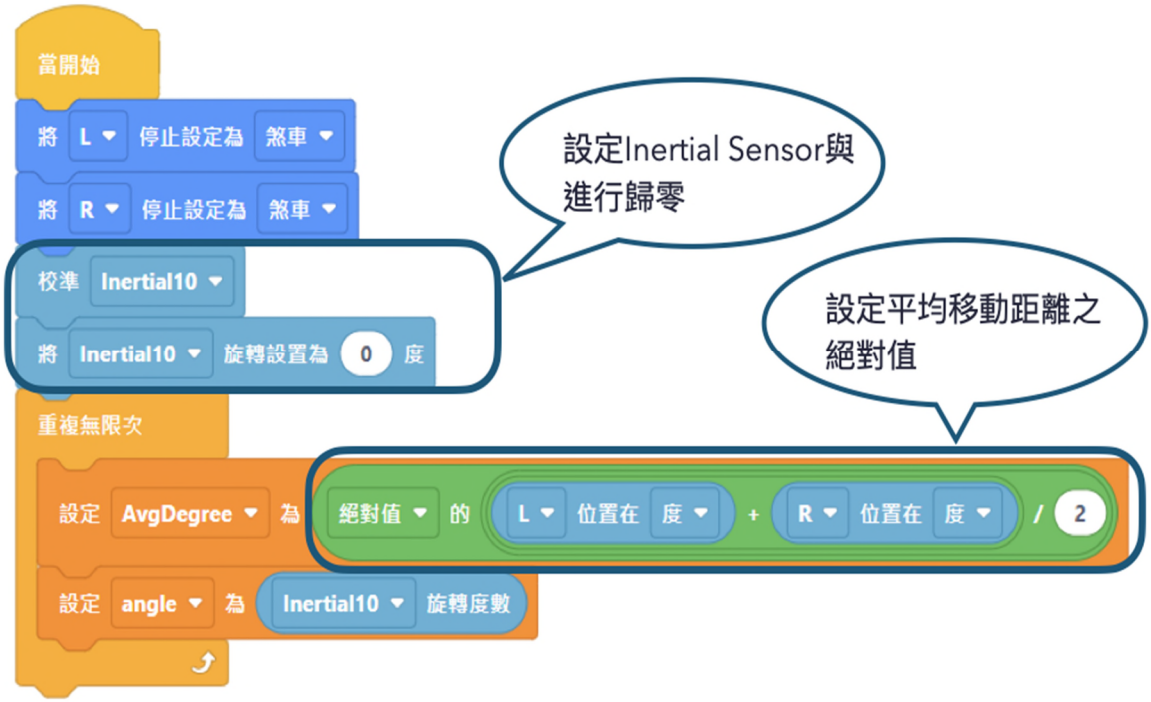

圖 6：Inertial Sensor 與平均距離絕對值設定

先從簡單的 P 控制開始，要讓機器人進行 P 控制，我們需要先設定誤差是多少，而誤差正是當前的角度減去目標的角度。之後我們需要讓誤差乘上一個常數 kp(圖 8 紅色圓圈所示，範例 Kp=0.3)，再將這個結果去調整馬達兩側的輸出。由於我們這次所使用的陀螺儀左邊為負右邊為正，因此當機器人偏向左側時，誤差為負，由於機器人偏左需要將左側加速右側減速，因此左邊的馬達需要「減去」P 控制的調整量，右邊則需要「加上」P 控制的調整量，這樣一來，原本設定的基本速度加上負數就會變成減速，而減去負數就會變成加速。如果換成偏向右側，誤差為正，就會變成左側減速右側加速。在設定完速度之後，就要讓馬達旋轉，到這邊，就是迴圈內所要做的事情，當達成外側的條件後，就會跳出迴圈，讓馬達停下來：(迴圈前要將變數歸零)。

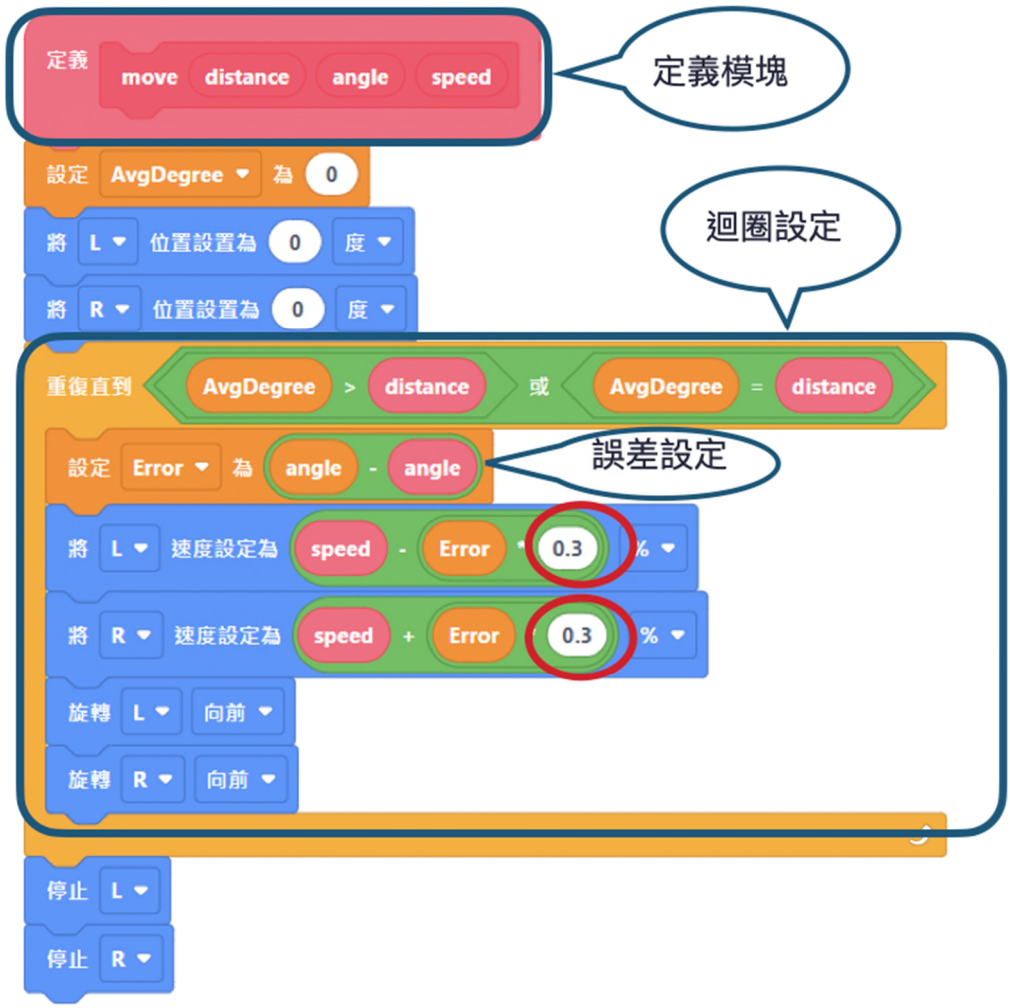

圖 7：定義與 P control 程式範例

完成 P 控制之後，我們接著來講 I 控制，由於 I 控制需要用到「累計誤差」這一變數，因此我們需要在迴圈內持續計算「累計誤差」，而「累計誤差」可以想成「累計誤差+當前誤差」。這樣講可能有人不能理解，「累計誤差」怎麼會等於「累計誤差加上當前誤差」，這樣不合理，但在程式的概念中，設定一個變數的值就是將右側的值存到左側變數。簡單舉個例子，累計誤差假設一開始是 0，當前誤差是 10，那在計算累計誤差的時候就是「0(累計誤差)+10(當前誤差)=10」下次計算時累計誤差就是 10，當前誤差因為可能已經進行了些微的修正，所以當前誤差假設為 8，那麼累計誤差就會是「10(累計誤差)+8(當前誤差)=18」，範例中累計誤差所乘上的數值就是 Ki 值(圖 8 紅色圓圈所示，範例 Ki=0) 。

圖 8：PI control 程式範例

最後是 D 控制，D 控制主要是看誤差的變化量，為了計算誤差的變化量，我們還需要定義一個新變數，來記錄上一次的誤差，這樣新誤差和舊誤差相減便可以得到「變化誤差」(VariedError)(圖 9 所示)，乘上常數 Kd 之後再去調整輸出(圖 9 紅色圓圈所示，範例 Kd=0)，不過這個時候要注意一下正負，如果今天誤差是變小，這個時候不是要加速，而是減速，因為 D 控制的目的之一便是提供一個阻尼的力量，避免修正太多，導致震盪的產生。為了讓變化誤差在修正時可以是負數，因此變化誤差的值會是當前誤差(修正之後 Error)減去過去誤差(修正之前 LastError)。

圖 9：PID control 程式範例

完成移動部分 PID 程式撰寫介紹，接下來要講轉彎的程式，首先一樣要先定義一個新的模塊。之後也是要設定一個迴圈，不過跳出去的條件跟移動的有很大的不同，移動跳出迴圈的條件是移動距離大於等於設定距離，這個條件跟轉彎可以說是沒什麼關係，那我們要怎麼判斷轉彎有達到設定的角度呢？其實很簡單，只要當實際轉向角度等於目標轉向角度就可以了，但是這裡有一個很重要的問題，就是陀螺儀的精度可能可以到小數後好幾位，要實際轉向跟目標轉向完全相同的機率實在很低，因此我們需要設定一個範圍，像可能正負兩度，這樣才可以確保跳得出迴圈，不過當然這個範圍也不能設定的太大，不然轉彎的準度就會太低。

講到這邊，不知道有人有人發現其實這樣會有一個大問題，就是轉彎其實有可能會因為震盪太大導致轉過頭，這個時候轉向角度也會有一段時間轉超過目標角度，轉超過也就代表有轉到目標角度，因此這時就會跳出迴圈，之後又會因為慣性等原因造成更大的誤差，為了避免這樣的情況，我們需要在迴圈內加入一個條件以及一個計數器，當實際轉向和目標轉向差不多時，計數器就會加一，但只要這個條件不成立，就會讓計數器歸零，而外圍的迴圈條件將改為計數器大於一定的數值。之所以改成這樣的條件，就是因為當計數器大於一定的數值時，也就代表機器人的轉向已經趨於穩定。

其他概念都和移動差不多，不過還有一點不同的是，由於轉向的角度可能會有大小的區別，因此變數的地方是可以進行調整的，也就是說，轉三十度，和轉六十度，他們的三個參數(K、Ki、Kd)是可以不一樣的。

在實際比賽中，有時候機器人即使有 I 控制，也會因為離目標角度差太多導致迴圈跳不出去，而在我所比的自動賽中，是有時間限制的，如果因為這樣突發的情況導致整局報銷，風險實在有點大。為了避免直接卡死的問題，我們還可以在外迴圈的地方加上另一個條件，就是限時。當執行轉彎程式的時候，重置系統內建的計時器；當計時器大於一定秒數時，就代表超過時限，這時不用角度到位，一樣也會跳出迴圈(圖 10 所示)。

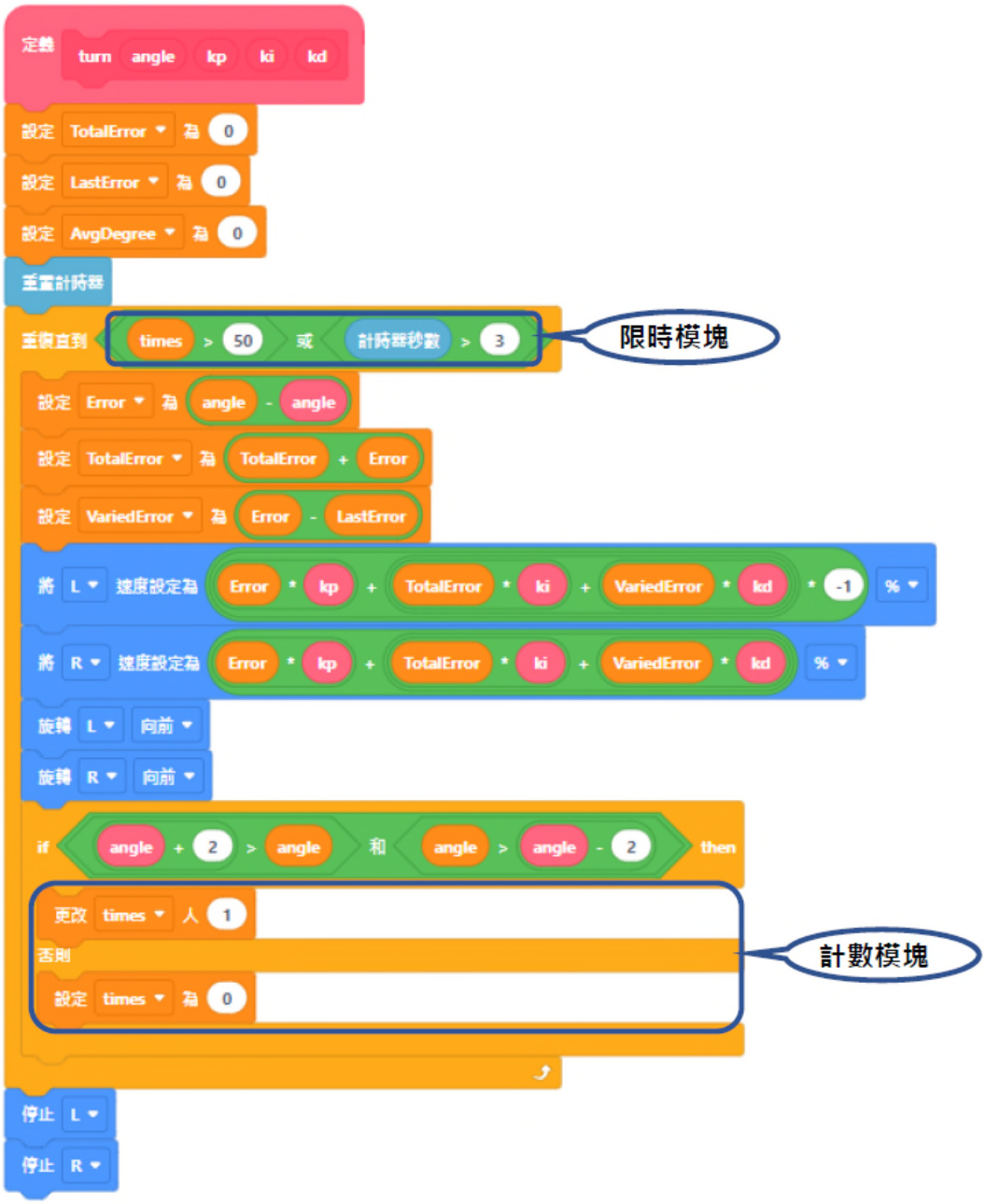

圖 10：PID control 程式範例

肆、結語

撰寫這篇 PID 介紹的原因之一就是因為當初在學 PID 控制的時候，雖然找了很多影片，也有問過一些人，由於專業的理論基礎與複雜的數學公式根本不容易了解 PID 控制，例如複雜的微積分公式、甚至連符號是什麼意思都不清楚。

經過不斷摸索與嘗試之後，漸漸就知道其實只要了解其中的概念，要寫出 PID 真的不是一件難事；更甚至，也有其他的方式可以寫出來。所以重點並不是「公式」，而是其中所蘊含的「概念」，因此想透過這篇簡介，讓之後想要學習 PID 的人可以有更好的資源可以運用，不然上網找真的就好比大海撈針，找了半天還是搞不懂。

所以在寫這 PID 介紹時，希望能夠將原本較為抽象的概念透過解釋成淺顯易懂的文字敘述來介紹，讓更多人閱讀完這篇文章後能夠對 PID 初步的認識與理解。像是「微分控制」我認為就是一個很好的例子，如果今天只是直接照用公式的講法「微分控制」，這樣的講法相信大部分的人都會難以理解，但只要說成「變化誤差」這樣講就比較好理解。

除此之外，像是 I 控制，如何計算累計誤差，如果原本沒有學過 PID 控制的時候，想說這個變數是要記錄從頭到尾所有的誤差再相加，所以要定義一個 list 去紀錄，結果根本不用，在看過一些相關的文獻和有一些基本的程式語言概念後，透過使用「累計誤差＝當前誤差＋累計誤差」方式這樣的方式去紀錄。

未來希望能夠進一步涉獵更多學習更多程式相關的控制方式，透過淺顯易懂的方式來寫相關的文章，使之後也想學習的人可以更快掌握相關的資訊。

參考文獻

黃冠渝、黃英哲（2018a 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/Article/C000003/ detail? ID= 53326f3b-f174-4eb9-bd75-89adc1c585d2

黃冠渝、黃英哲（2018b 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/Article/C000003/ detail?ID =47abfefa-6d4d-4fdd-8220-7156cd5790a1

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。

Åström, K. J. & H ̈agglund, T. (2001). The future of PID control. *Control Engineering Practice, 9,* 1163-1175.

Bennett, S. (1996). A Brief History of Automatic Control. *IEEE Control Systems Magazine, 16,* 17-25.

Borase , R. P., Maghade, D. K., Sondkar, & Pawar, S. N. (2020). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control, 9,* 818-827. https://doi.org/10.1007/s40435-020-00665-4

H ̈agglund, T., & Guzm ́an J. L. (2024). Give us PID controller and we can control the world. *IFAC-PapersOnline, 58,* 103-108.

How to program VEX IQ precise turn with VEXcode blocks by Caution Tape Robotics. (January 16th, 2021). Retrieved January 31, 2020, from https://www.cautiontape.ca/program-vex-iq-precise-turn-with-vexcode-blocks/

How to make VEX IQ robot drive straight with VEXcode blocks by Caution Tape Robotics. (January 16th, 2021). Retrieved January 31, 2020, from https://www.cautiontape.ca/vex-iq-robot-drive-straight-with-vexcode-blocks/

Kukkala, V. K., Tunnell, J., Pasricha, S., & Bradley, T. (2018). Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles. *IEEE Consumer Electronics Magazine, 7*(5), 18-25. https://doi:10.1109/mce.2018.28284

Zurich Instruments. (2023). Principles of PID Controllers. [White paper]. Zurich Instruments. https://www.zhinst.com/others/en/resources/principles-of-pid-controllers