第二十三屆旺宏科學獎

成果報告書

參賽編號：SA23-206

隊伍名稱：沒有禿頭的工程師

作品名稱：電腦視覺系統導入機器人實現延長續航時間之可行性

參賽類別：電子電機

關鍵字：機器人、電腦視覺、摩擦力

目錄

圖目錄

表目錄

摘要

本研究的主要目的是運用雙目影像辨識技術結合 3D 建模來協助仿生機器人維持平衡與適應環境。研究的目標是透過即時分析周遭環境的深度資訊，建立 3D 模型，再將此數據回傳給仿生機器人系統，使機器人能依據其重心位置及腳部與地面的摩擦力，計算出其最大可傾斜角度，進而調整姿態以維持平衡。此研究針對四足仿生機器人，設計出一套優化的系統模式與視覺演算法。透過 3D 建模獲取的環境資訊，機器人能精準評估其姿態穩定度，並做出適當的動作調整。這樣的設計不僅使機器人能在複雜環境中保持平衡，更可藉由減少不必要的動作來節省能源，延長運作時間。這項研究結合了電腦視覺、3D 建模與機器人動力學等跨域的知識，主要在提升仿生機器人的環境適應能力與能源效率。透過視覺感知與演算法的應用，機器人能更靈活地應對不同環境挑戰，為未來機器人在戶外及災難救援等領域的實際運用有相當大的助益。

關鍵詞：機器人、影像辨識、摩擦力

電腦視覺系統導入機器人實現延長續航時間之可行性

壹、前言

一、研究動機

隨著科技的進步，四足機器人作為各科技公司投入研發的項目，其原理為透過模仿四足動物的運動方式去克服輪式或履帶式機器人所不能處理的複雜地形。這種設計使得機器人在理論上擁有越野穿越能力和環境適應性。然而，在實際應用中，我們發現市場上的四足機器人，其平衡和地形適應以及續航能力仍有待提升。經過討論研究後我們發現大多數生物都是透過視覺去感知環境和自身姿態，但是這些四足機器人大多限於將畫面回傳給操控者而已，所以我們提出一個研究方向：運用雙目影像識別技術對前方環境3D 建模，結合靜摩擦力和臨界角的物理計算，對四足機器人的平衡控制進行優化。透過這種方法，我們可以更準確地計算機器人與地面的接觸條件，預測可能的滑動或側翻風險，並即時調整機器人的步態和姿態，以達到更高的穩定性和行動效率，最終達到省電的效果。

二、研究目的

我們的目標是透過本次研究提供一種新的視覺輔助平衡模式以支援仿生機器人的發展。為實現這一目標，我們將研究分為幾個階段。透過不同階段的探索，我們將獲得豐富的知識，不僅僅是在實現預期目標方面，更是在整個領域的深入理解上。我們主要想研究的目標如下：

(一)了解四足機器人運動方式

(二)了解摩擦力與傾斜角的影響

(三)了解 OpenCV 使用方法

(四)了解如何使用 Tinkercad

(五)了解視覺平衡的原理

三、文獻探討

（一）MiDaS 深度預測模型

MiDaS 模型使用 CNN (Convolutional Neural Network，卷積神經網路) 以及單目視覺進行深度估計，它有著大量針對不同需求以及硬體效能的模型，且輸出僅一張深度圖，方便讀取以及資料處理。它使用 RGB 影像對輸入的每一個相素與鏡頭進行相對距離的估計，將資料整合後輸出一張深度圖。

圖 1 彩色輸入與輸出對比圖(圖片來源：midas 論文)

（二）雙目立體視覺

雙目立體視覺以模仿人類視覺系統的方式，它透過不同視角的圖像來重建場景的三維結構。這種技術基於視差，即同一物體在兩個圖像中的位置差異，從而計算出物體的深度資訊。雙目立體視覺廣泛應用於機器人尋路、自動駕駛汽車、虛擬實境（VR）和擴增實境（AR）等領域。

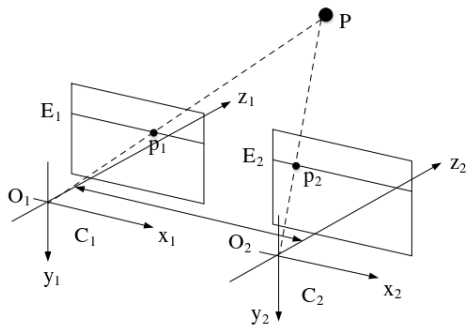


圖 2 雙目視覺系統概念圖(圖片來源：

https://www.researchgate.net/figure/Schematic-diagram-of-binocular-vision_fig1_354996938)

（三）利用振盪模式的四腳機器人自適應行走控制

CPG 網路由雙主單元和基於陀螺儀訊號（包括偏航角和俯仰角）的從單元子集組成。主單元的反應提供了控制四足機器人第一關節的能力，而從單元可以產生對稱訊號來控制第二和第三關節。 CPG 網路透過使用超音波感測器實現運動步態的無縫切換以停止和啟動(Yong Zhang，2023)。

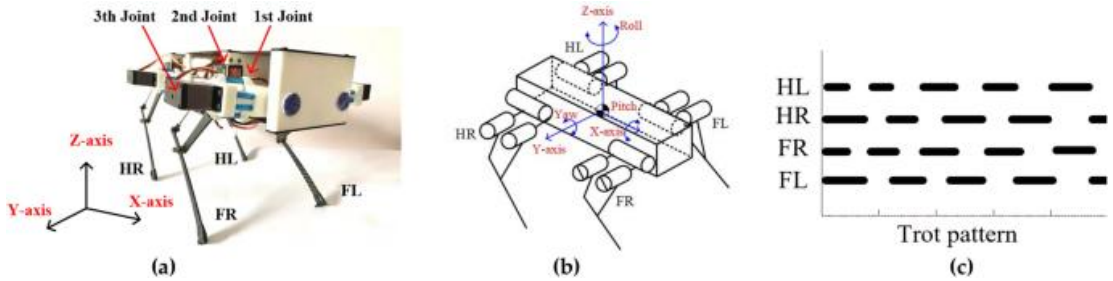


圖 3(a)四足機器人的機械結構； (b)機器人結構示意圖； (c)小跑步態。

(圖片來源：https://www.nature.com/articles/s41598-023-47022-x)

（四）深度強化學習控制 Solo12 四足機器人

針對四足動物提出了許多基於運動規劃和軌跡最佳化的控制方法。提出了一種全身控制公式，可輸出必要的低階控制，以便在較短的時間範圍內追蹤基本軌跡，實現了類似的基於 MPC 的方法，同時簡化了全身控制計算的解決方案。雖然所有這些方法都能產生穩健的動態控制器(Yong Zhang，2023)。

圖 4 Solo12 四足動物在真實環境中和由透過深度強化學習學習的反應

(圖片來源：https://www.nature.com/articles/s41598-023-38259-7)

（五）帶臂的四足機器人的魯棒運動操縱

機器人從所有四隻腳在預先指定的位置開始，然後將其右後腳抬離地面一小段時間。我們還限制了夾持器在整個運動過程中保持在某個位置不動。在整個運作過程中，我們在操作規劃階段預先指定了機器人每隻腳的位置。當命令機器人轉動手輪時，我們透過前向運動計算機器人在那一刻的腳位置，確保機器人的腳在轉動輪子時保持在這些位置(Henrique Ferrolho，2023)。

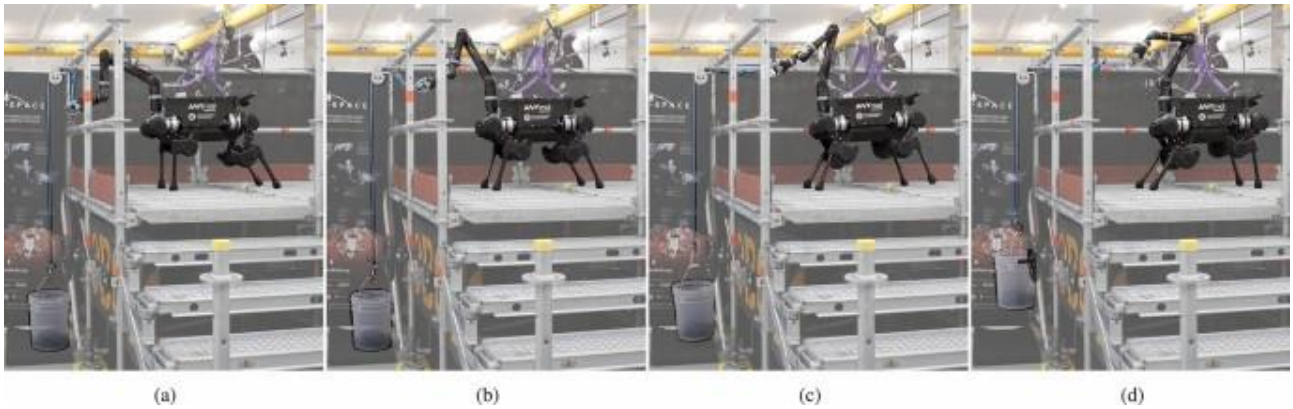
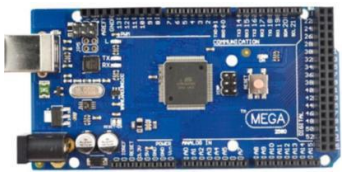
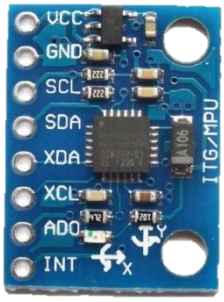


圖 5 帶臂的四足機器人的快照

(圖片來源：https://link.springer.com/article/10.1007/s10514-023-10146-0)

貳、研究設備及器材

硬體	
筆記型電腦	 圖 6 筆記型電腦(圖片來源：Google)
Arduino MEGA	 圖 7 Arduino MEGA(圖片來源：作者自攝)

WEMOS LOLIN D32 ESP32	 圖 8 ESP32 (圖片來源：作者自攝)
MPU6050	 圖 9 MPU6050 (圖片來源：作者自攝)
鏡頭	 圖 10 鏡頭(圖片來源：作者自攝)
雙目鏡頭	 圖 11 雙目鏡頭(圖片來源：作者自攝)
3D 列印機	 圖 12 3D 列印機(圖片來源：作者自攝)
電源供應器	 圖 13 電源供應器(圖片來源：作者自攝)

軟體	
OpenCV(電腦視覺)	 圖 14 Open CV(圖片來源：OpenCV 官網)
Arduino	 圖 15 Arduino (圖片來源：維基百科)
PlatformIO	 圖 16 PlatformIO (圖片來源：PlatformIO 官網)
Tinkercad	 圖 17 Tinkercad (圖片來源：維基百科)
Python	 圖 18 Python(圖片來源：Python 官網)
Yolov4	 圖 19 YOLO(圖片來源：YoLo 官網)

參、研究過程或方法

開始研究前，我們經過討論以決定仿生四足機器人的功能，包含影像識別、動態平衡等，以及其外觀樣式，為掌握整個研究進度，因此繪製流程圖，便於我們開發過程中檢視研究完成度。

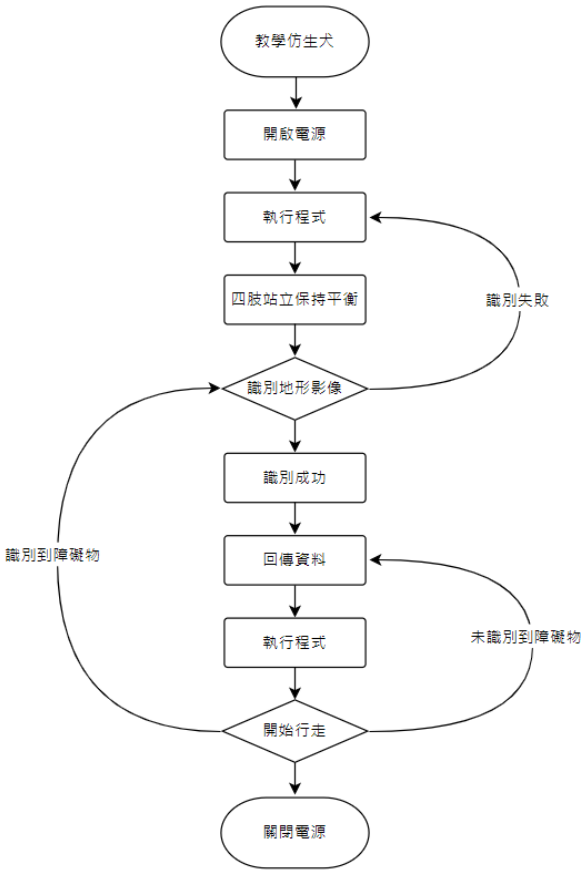


圖 20 研究流程圖

一、四足機器人硬體開發

我們參考了網路上有關四足機器人的資料，並且自行設計各關節結構，再利用 3D 列印機印出進行加工處理組裝，這部分花了我們近 2 個月時間，雖然過程中，我們有參考國外的共享開發資料，但自製難度還是很高。

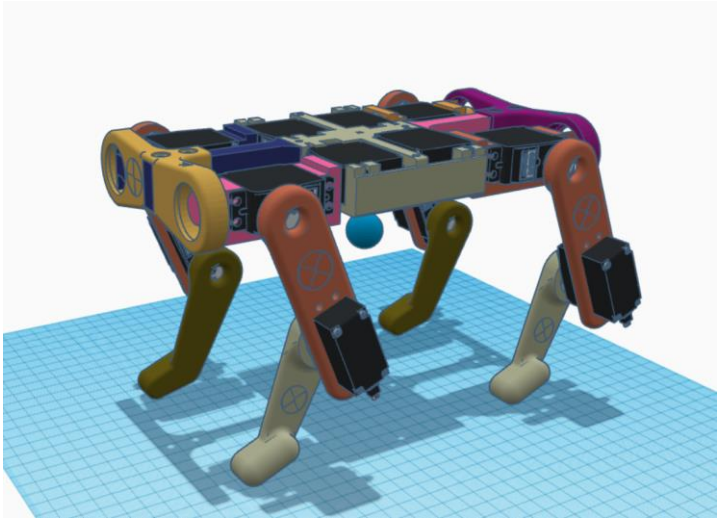


圖 21 四足機器人 3D 設計圖預覽

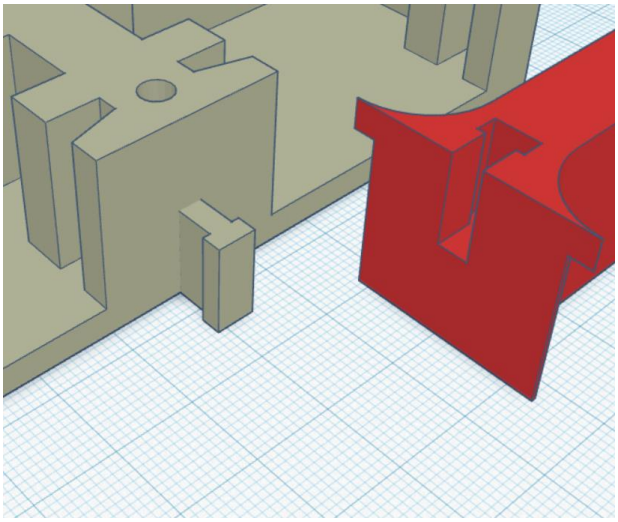


圖 22 自行以 tinkercad 繪製的連接結構

在電路部分我們以 LOLIN D32 ESP32 作為四足機器人的大腦、PCA9685 16 路 PWM 擴展板對舵機進行控制以及 MPU6050 六軸慣性感測器進行姿態解析。而這些元件都由 I²C 進行連接。

LOLIN D32 有鋰電池充電 IC TP4054 可以在沒有額外電源的情況下保持運作、PCA9685 可以使用 I²C 對 16 路 PWM 進行控制、MPU6050 則可以提供 XYZ 三個方向上的加速度和角速度。

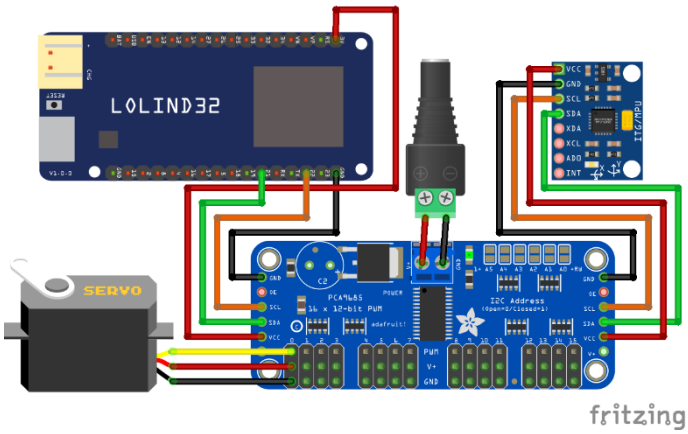


圖 23 機器人電路設計

二、姿態&運動演算法

在姿態和運動的過程中，需要對所有關節進行統一控制。而在移動狀態下每條腿的末端位置都不一樣，所以我們需要透過一個期望的末端位置對每條腿進行單獨控制。

（一）二維運動原理

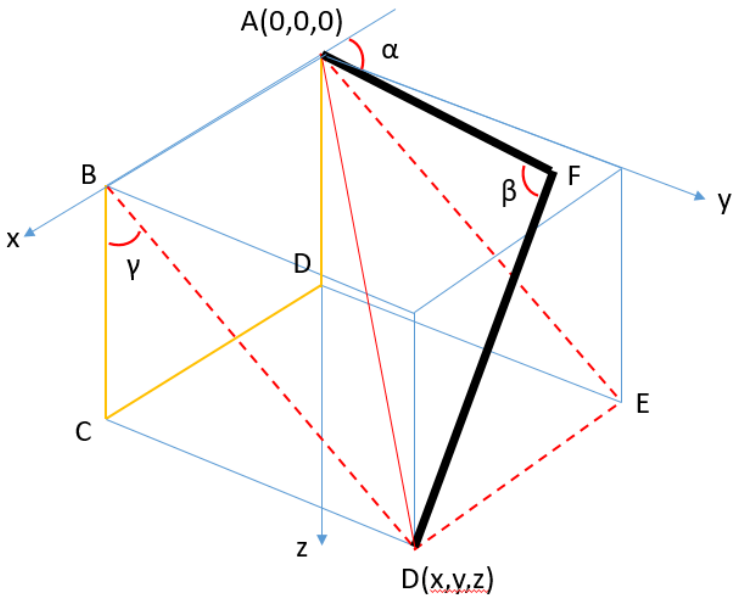


圖 24 運動模型(圖片來源：https://imuncle.github.io/content.html?id=61)

腿部在平面內有兩個自由角度分別為 α 和 β，想要根據預定坐標解出 α 和 β 需要解三角函數。圖 24 中的 a 和 b 為機器人的兩條腿可以直接測量。既然如此便可計算出直線 c 的長度，然後在以餘弦定理就可求解出 α 和 β。

$$\alpha = 180 - \tan^{-1}\frac{y}{x} - \cos^{-1}\left(\frac{a^2+x^2+y^2-b^2}{2a\sqrt{x^2+y^2}}\right)$$

$$\beta = \cos^{-1}\left(\frac{a^2+b^2+x^2-y^2}{2ab}\right)$$

資料來源：https://imuncle.github.io/content.html?id=61

（二）三維運動模型

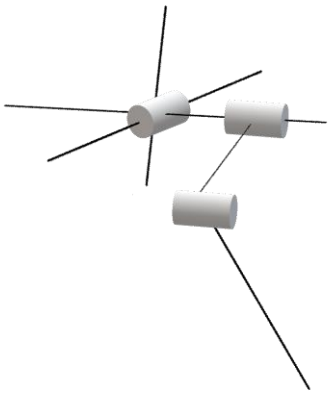


圖 25 三維運動模型(圖片來源：使用小畫家 3D 製作)

依照圖 25，我們可以在添加一個活動角度使得腿部變化為三維運動增加機動性。

圖 24 中黑色粗線代表機器人的大腿與小腿部分，α、β 和 γ 則代表的是各舵機的角度位置。而 ABCD 四點則是機器人站立時腿部所在平面，即此面垂直於地面。

既然如此，可透過三角函數推導出以下公式：

$$\alpha = 180 - tan^{-1}\left(\frac{\sqrt{y^2+z^2}}{x}\right) - cos^{-1}\left(\frac{a^2+x^2+y^2+z^2-b^2}{2a\sqrt{x^2+y^2+z^2}}\right)$$

$$\beta = cos^{-1}\left(\frac{a^2+b^2-x^2-y^2-z^2}{2ab}\right)$$

$$\gamma = tan^{-1}\left(\frac{y}{z}\right)$$

資料來源：https://imuncle.github.io/content.html?id=61

（三）姿態控制

我們要探討的是四足在傾斜地面行走時傾斜機身與不傾斜時的能源消耗表現，而 pitch 軸的傾斜會造成前後腿的抬腿高度差異，因此先探討 roll 軸傾斜對四足造成的影響。

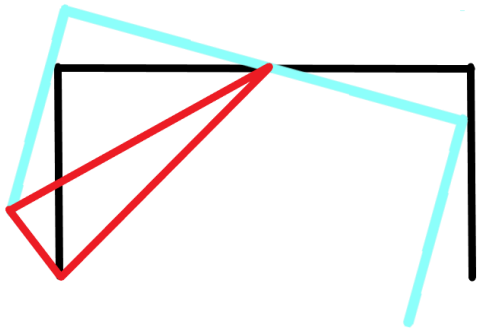


圖 26 平衡角度分析圖

Roll 軸傾斜時不能以接觸地面為轉軸，應以四足重心點作為轉軸。我們設計的四足其中有 8 顆舵機都是在機身上，並且機身上還搭載了電源模塊，所以簡化模型以機身重心作為轉軸進行姿態解析。

可以看到在經過 roll 軸旋轉後，四肢末端與未旋轉前的點以及轉軸形成圖中的等腰三角形。其頂角 α 是 roll 軸傾斜角度，腰長透過結構可以計算出。然後使用餘弦函數就可以解底邊長，最後就可以化簡公式如下。

$$\iota = \sqrt{2L^2(1-\cos\alpha)}$$

$$\beta = \theta - \frac{\alpha}{2}$$

$$\delta x = -\iota\cos\beta$$

$$\delta z = \iota\sin\beta$$

資料來源：https://imuncle.github.io/content.html?id=73

三、仿生四足行走分析

在行走姿態我們參考犬科動物的運動模式。

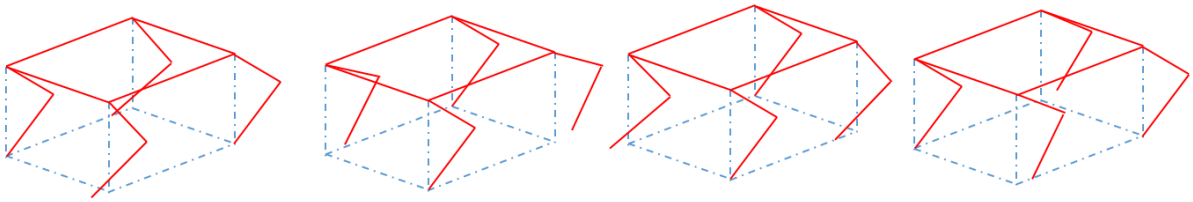


圖 27 行走動作分析圖(來源：https://imuncle.github.io/content.html?id=63)

基本上先邁出左前和右後腿（第一幅圖），然後機身往前移動，直到左前和右後腿又回到初始位置（第二幅圖），此時邁出右前和左後腿（第三幅圖），機器人身體繼續向前移動， 直到右前和左後腿又回到初始位置（第四幅圖），然後開始往復循環。

程式實現：

分析後發現，四條腿是依次落地和抬起的，動作順序可以是：右前→左後→左前→右後。假設動作時間為 T，那麼單一條腿抬腿向前的動作需要在腿向前的動作需要在 1T 內完成，觸地狀態為 3T 內完成，所以週期時間為 4T 進行迴圈。

```
176 > void ClearLegData(struct Leg_t * leg)···
182
183 > void ServoControl(void)···
190
191 > void snycWrite(uint8_t ID[], uint8_t IDN, uint8_t MemAddr, int position[])···
200
201   void ListLeg(void)
202   {
203       static int count = 0;
204       count ++;
205       uint8_t cycle = 36;
206       FR_Leg.last_state = FR_Leg.state;
207       BL_Leg.last_state = BL_Leg.state;
208       FL_Leg.last_state = FL_Leg.state;
209       BR_Leg.last_state = BR_Leg.state;
210       if(count<cycle)
211       {
212           FR_Leg.state = Leg_t::Up;
213           BL_Leg.state = Leg_t::Down;
214           FL_Leg.state = Leg_t::Down;
215           BR_Leg.state = Leg_t::Down;
216           if(count < cycle/2)
217           {
218               FR_Leg.d_z = -count*40/cycle;
219           }
220           else
221           {
222               FR_Leg.d_z = -(cycle-count)*40/cycle;
223           }
224       }
```

圖 28 抬腿程式

為了方便後期維護程式以及增加易讀性我們仿生犬抬腿程式只負責控制腿部上下移動，對於另外的行走則是單獨控制腿部在平面二維的移動。

```
void Walk(struct Leg_t * leg)
{
    uint8_t cycle = 36;
    int StepLength = 2*body.vx;
    if(Abs(body.vx) < 2)
    {
        leg->d_x = 0;
        leg->count = 0;
        return;
    }
    if(leg->state != leg->last_state) leg->count = 0;
    leg->count++;
    if(leg->state == Leg_t::Up)
    {
        leg->d_x = -StepLength+leg->count*2*StepLength/cycle;
    }
    else if(leg->state == Leg_t::Down)
    {
        leg->d_x = StepLength-leg->count*2*StepLength/(2.5*cycle);
    }
}
```

圖 29 腿部行走程式

從上面程式可以看出，在腿抬起時從-StepLength 移動到 +StepLength 實現向前邁進，在放下時又從+StepLength 移動到 -StepLength 實現將身體向前移動，腿往後移動。

四、摩擦係數

摩擦力是兩物體接觸面分子的交互作用力，表面粗糙的程度不同，摩擦力就不同。物體的拉力即為摩擦力的大小；當物體的拉力大於分子之間的交互作用力時，物體就會開始運動，這時的拉力大小就是靜摩擦力最大值，稱為最大靜摩擦力，最大靜摩擦力正比於物體接觸面的正向力。 N 即 ：

$$F_{s,max} = \mu_s N$$

當物體開始運動後，摩擦力仍會存在，稱之為動摩擦力f_k。動摩擦力比最大靜摩擦力在小一點，且可視為定值，其亦正比於物體接觸面的正向力 N，即：

$$F_k = \mu_k N$$

（一） 動摩擦力

動摩擦力是一種阻礙兩個物體相對運動的力量。當兩個物體接觸並試圖相對移動時，它們之間的表面不會完全光滑，而是有微小的凸起和凹陷。這些微觀結構會相互摩擦，產生一股阻礙物體相對運動的力量，我們稱之為動摩擦力。

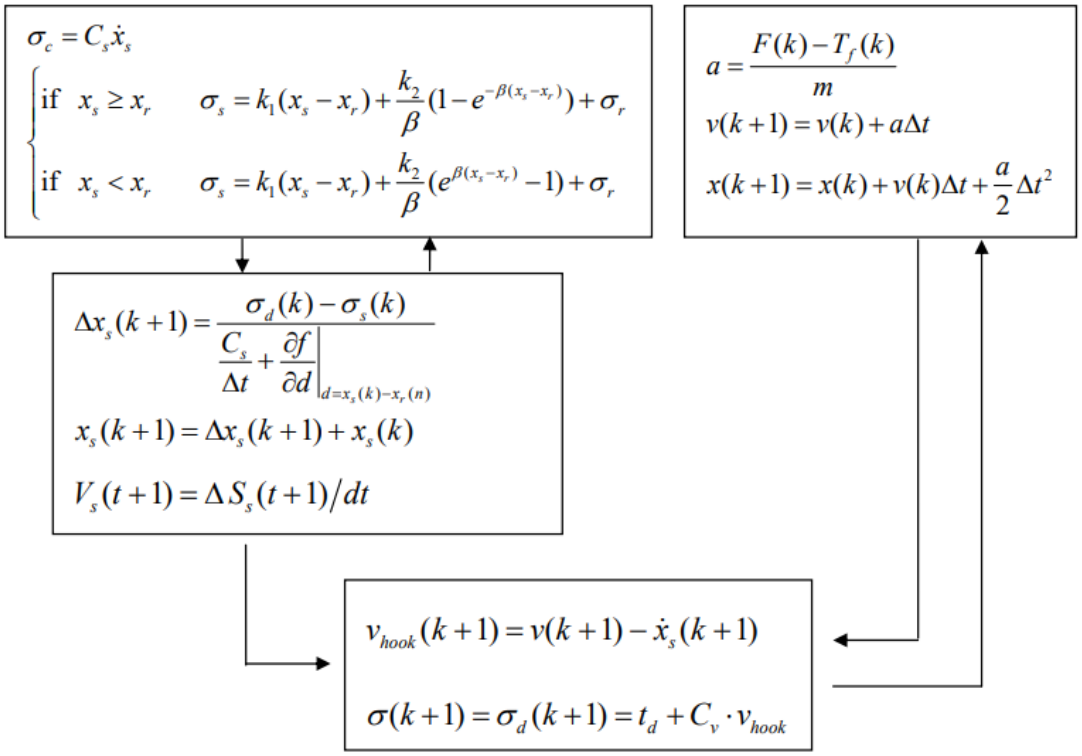


圖 30 動摩擦力流程圖(圖片來源：http://klab.me.ncku.edu.tw/2021 課程/系統動態分析與模擬課程資料/Week%201/期末論文與報告/Dynamics/09%20 陳彥丰%202006.pdf)

(二) 最大靜摩擦力

靜摩擦力是兩個物體相對靜止時所存在的摩擦力。當兩個物體接觸並試圖相對移動時，如果施加的外力還不足以克服兩物體間的摩擦力，則這種摩擦力被稱為靜摩擦力。

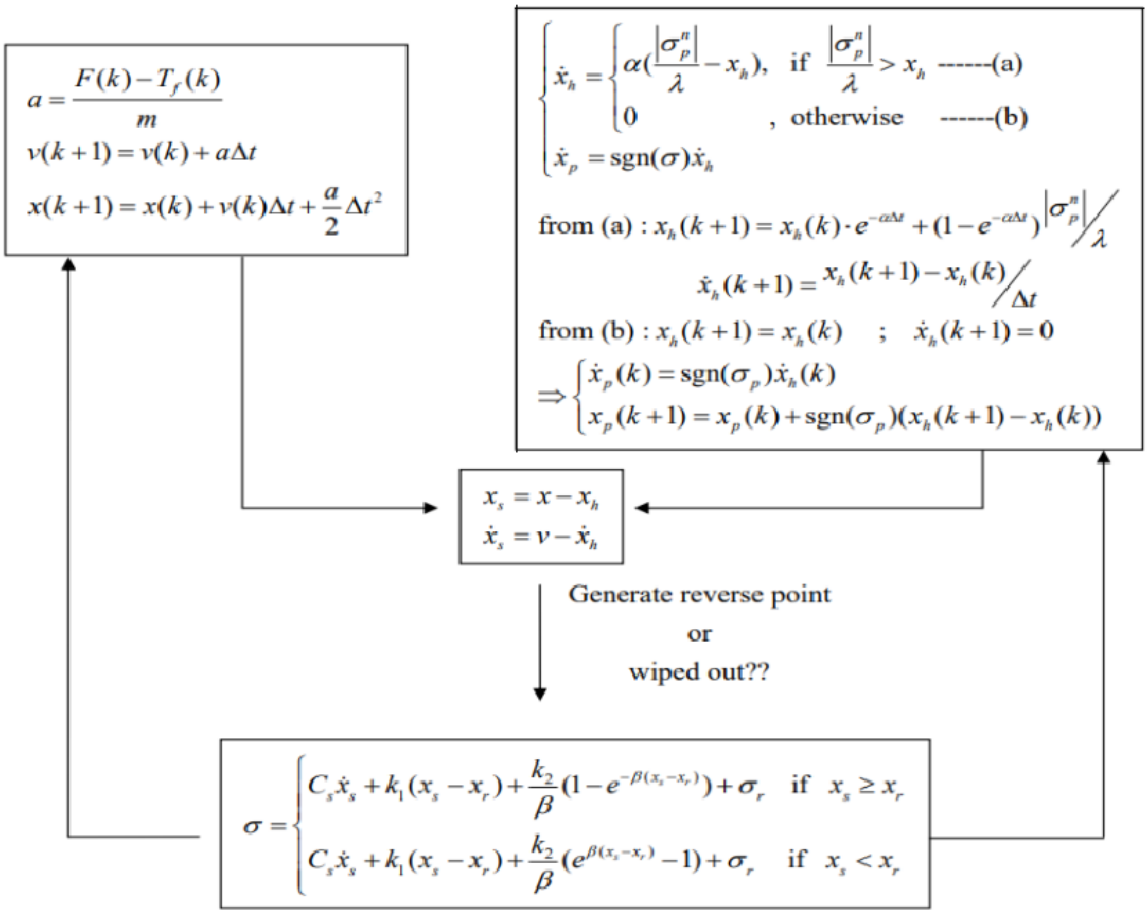


圖 31 最大靜摩擦力流程圖(圖片來源:http://klab.me.ncku.edu.tw/2021 課程/系統動態分析與模擬課程資料/Week%201/期末論文與報告/Dynamics/09%20 陳彥丰%202006.pdf)

五、應用平衡視覺之後，能源節省之間的關係

當考慮能源節省時，靜摩擦力在設計和操作設備時扮演了重要角色。以下是靜摩擦力如何影響能源節省的一些方面：

（一）啟動能源消耗：在開始移動之前，物體必須克服靜摩擦力。這代表在啟動設備時，必須施加足夠的力量才能克服靜摩擦力，並且這會消耗額外的能源。

（二）維持運動：一旦物體開始移動，靜摩擦力通常會減少為動摩擦力。但在靜止和運動之間轉換時，需要克服靜摩擦力所需的能量可能會比保持運動時消耗的能量更多。

（三）運動效率：靜摩擦力的大小直接影響了設備的運動效率。減少靜摩擦力可以使設備更容易啟動並保持運動，從而減少能源消耗。

（四）設計優化：考慮到靜摩擦力可以有助於設計更有效率。透過選擇表面材料、優化接觸面積等方法，可以減小靜摩擦力，從而節省能源。

（五）控制策略：在操作設備時，可以透過適當的控制來減少靜摩擦力的影響。例如，可以透過減少啟動時的加速度或改變運動方向等方式來降低靜摩擦力所需的能量。

考慮靜摩擦力在設計和操作中的影響對於能源節省至關重要。透過減少靜摩擦力或者有效地管理它，可以降低能源消耗，提高系統的運行效率，從而達到能源節省的目的。

六、減少摩擦力與能源節省的關係

（一）降低機械能源消耗：機械中運作中的摩擦會導致能源消耗增加。透過減少摩擦，可以減少機械運轉時所需要的能量，從而降低能源成本。

（二）熱能損失：摩擦會產生熱能，在一些情況時是可盡量避免的。例如，在機械零件之間的摩擦會導致它們加熱，進而導致能源的浪費。透過減少摩擦，可以減少這種熱能損失，從而節省能源。

（三）效率提升：減少摩擦可以提高機器和設備的效率。當零件之間的摩擦減少時，系統需要的能量就會降低，從而提高整體效能。可以使用更少的能源來實現相同的工作量。

（四）延長設備壽命：高摩擦力不僅會增加能源消耗，還會加速設備的磨損和老化。降低摩擦可以減少設備的磨損，延長其壽命，進而減少能源消耗與資源浪費。

（五）促進可持續發展：能源是有限的資源，減少能源消耗有助於減緩對自然資源的耗竭，推動社會走向更加可持續的發展方向。

（六）降低環境影響：減少能源消耗有助於減少環境污染和溫室氣體排放。因此，減少摩擦可以間接降低對環境的負面影響。

七、臨界傾斜角計算

要計算仿生四足機器人在什麼情況下會開始滑動和側翻，我們需要考慮以下因素：

1.重心位置

2.摩擦係數

3.支撐面積以及形狀

由以上因素可以獲得兩個思考方向，分別是在傾斜情況下重力所帶來的側向力超過腳底的最大摩擦力的臨界傾斜角，以及重心超越支撐形狀側面的臨界傾斜角。

當地面開始傾斜,仿生四足的垂直線會隨著傾斜逐漸超出其支撐面,然後失去平衡。設支撐面短邊長度為 W，重心到地面的距離為 H，地面傾斜角度為 θ，其重心與支撐面的相對位移可化簡為以下公式：

$$x_{\Delta} = H\,tan(\theta)$$

當位移超過支撐面短邊長度一半就會開始失去平衡，即傾斜角需求：

$$2H\,tan(\theta) \leq W$$

而臨界角度同時也發生在沒有足夠摩擦力的情況,即將發生滑動的瞬間。在這一點，靜摩擦力達到最大值，等於靜摩擦係數 μ 乘以重力的垂直分量。

此時，傾斜角度 θ 滿足以下關係：

$$tan(\theta) = \mu$$

由於靜摩擦力最大值需要與傾斜面上重力的水準分量平衡，我們需要選擇最小的作為仿生四足的姿態傾斜最大角度。

八、傾斜分析

(一)視覺分析

想要分析傾斜只需要透過物件的四個頂點分析長邊斜率即可，然後使用正反切換計算出傾斜角 θ。我們使用 OpenCV 來處理影像進行分析。

1.獲取鏡頭影像，然後進行影像格式轉換後使用高斯模糊對噪點進行處理增加識別精度。

2.為避免有大塊噪點影響辨識，二值化的圖片進行物件框選在考慮到影像雜訊、光照或是有與物件顏色類似的物品在旁邊，我們需要分離出二值圖最大的圖塊，也就是距

離攝影機最近的物件。我們先使用 findContours 找出所有圖塊輪廓，然後以回傳值的資料數去判斷是否有輪廓被標記出來。最後使用 max 函式選出最大的圖塊，最後利用 minAreaRect 對輪廓進行面積計算就可以剔除畫面中只有雜訊的問題。

3.傾斜角可以利用物件長邊兩個點的二維坐標位置去計算兩點所形成的線之斜率 slope，最後使用三角函數可知θ的正切函數值為直線的斜率，即$m = tan(\theta)$而我們需要透過斜率求傾斜角θ是用反正切函數$\theta = \tan^{-1}(m)$。

4.我們使用的是已知的物件，其長寬高都是我們知道的。而物件在畫面中的大小與其距離成比例關係，所以我們可以大致判定物件距離。

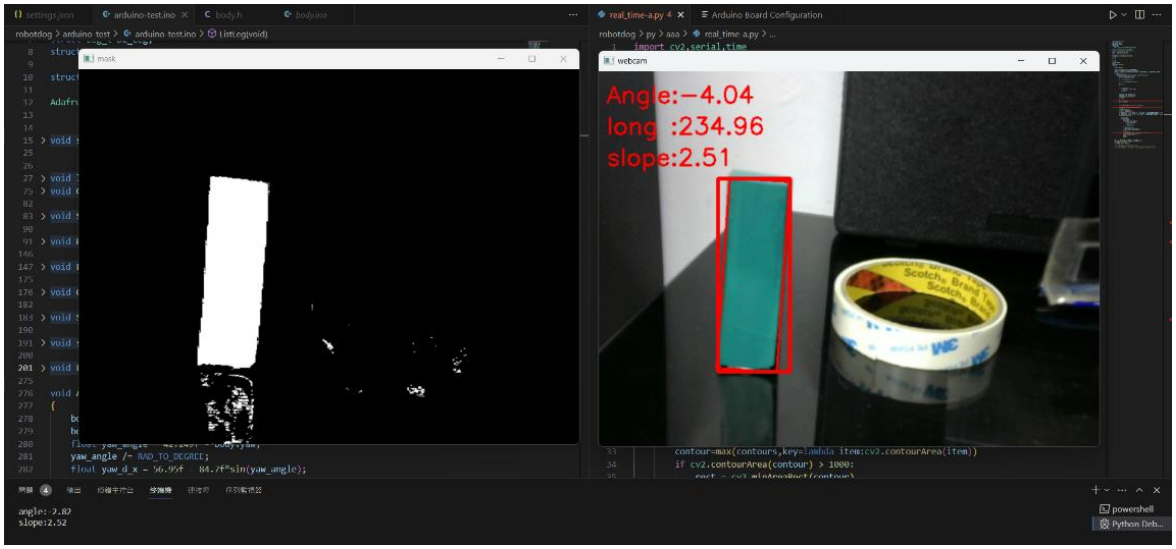


圖 32 物件角度識別程式

(二)陀螺儀分析

使用視覺去分析傾斜角需要有物件在機器學習的模型內，並且對於環境光照有一定需求。所以我們加入了 MPU6050 對傾斜分析進行輔助，透過三軸加速度就可以計算出目前的姿態。相比需要物件和環境光照條件，無處不在的重力顯得更加穩定。以其 x 軸和 z 軸的比值經由三角函數進行計算就可以獲得傾斜角。

$$tan^{-1}\frac{z}{x}$$

陀螺儀在實際使用時會有雜訊產生，如果某個時間點的雜訊太大會嚴重影響系統對於傾斜角的判斷，並且產生非預期內的結果。在查詢資料後我們發現卡爾曼濾波作為一個自我遞歸的濾波器就非常適合處理陀螺儀這種不完全、包含雜訊的測量值。它與其他

濾波器最大的不同在於它只需要上一次的估計值和現在的量測值就可以計算出當前的估計值。

卡爾曼濾波概念如下：

$$\hat{x}_t = (1 - K_t)\hat{x}_{t-1} + K_t z_t$$

$\hat{x}_t$是時間 t的估計值。

$\hat{x}_{t-1}$是 $t-1$ 的估計值。

z_t是時間 t的實際觀測值。

K_t是卡爾曼增益。

卡爾曼濾波是依靠卡爾曼增益作為權重去判斷是估計值重要還是實際觀測值重要。在我們將陀螺儀的數值帶入卡爾曼增益後我們使用序列埠繪圖儀來觀測經過濾波後的波形與未經過濾波波形的差異。我們觀察到當將陀螺儀放在平面時，在失真的情況下經過濾波的波形幾乎沒有變動。大幅度搖晃的情況下可以看到波形其實有延遲，但是經過兩次數據更新後就會跟上，程式單次回圈平均為 10ms 所以實際角度延遲為 20ms，所以延遲時間還在接受範圍內

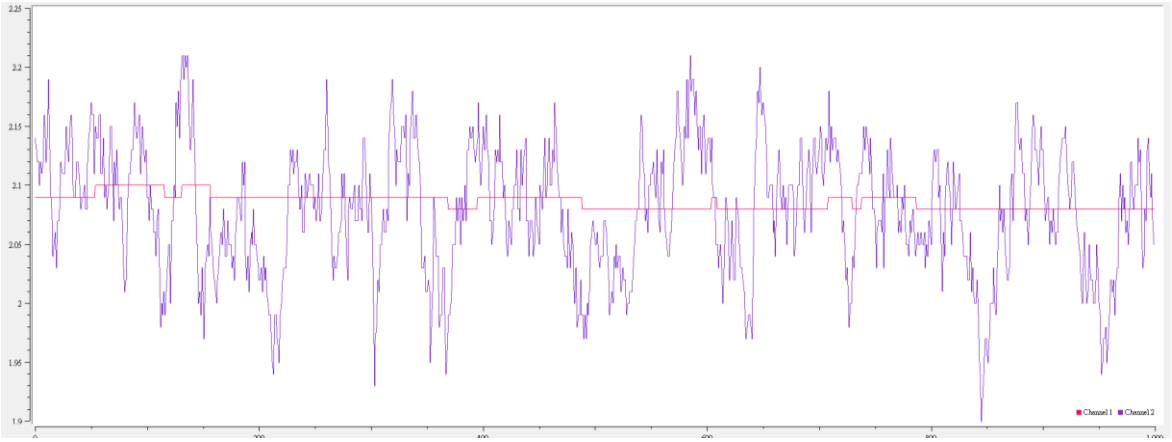


圖 33 平面狀態下濾波波形比較

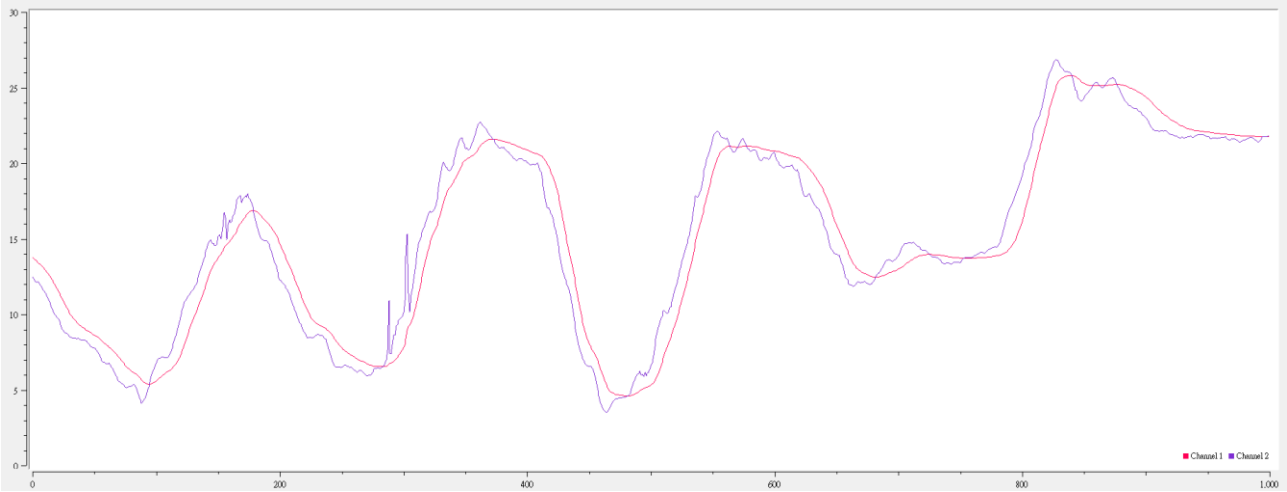


圖 34 傾斜狀態下濾波波形比較

九、雙目視覺

雙目視覺以左右鏡頭影像的視差，直接對前方物件或影像像素進行距離測量，計算出物件對雙目模組的相對距離。我們先訓練了 YOLOv4 模型用以偵測物件在圖片中的位置然後計算左右影像中物件的水平位置變化。其公式如下：

$$Z = \frac{f \times B}{d_\Delta}$$

B基線距離：兩鏡頭中心之間的水平距離。

f焦距：相機鏡頭到成像平面（CMOS 感光元件）的距離。

d_Δ視差：物件在左右影像上投影點的水平距離差與物體距離遠近成反比。

Z物體實際距離：物體距離相機的距離。

在 Python 的 openCV 裡面內置了 StereoBM 和 StereoSGBM 兩種立體匹配算法，前者對每個圖像區塊進行匹配來估算視差而後者是半全局匹配的算法，在計算時，在局部匹配，還考慮了全局信息。在經過實際測試後我們發現StereoBM生成的視差圖比較破碎，StereoSGBM 生成的則比較完整。為了更方便的調整參數，使用拍攝好的圖片以及 openCV 的滑桿設計了調參數用的測試程式。

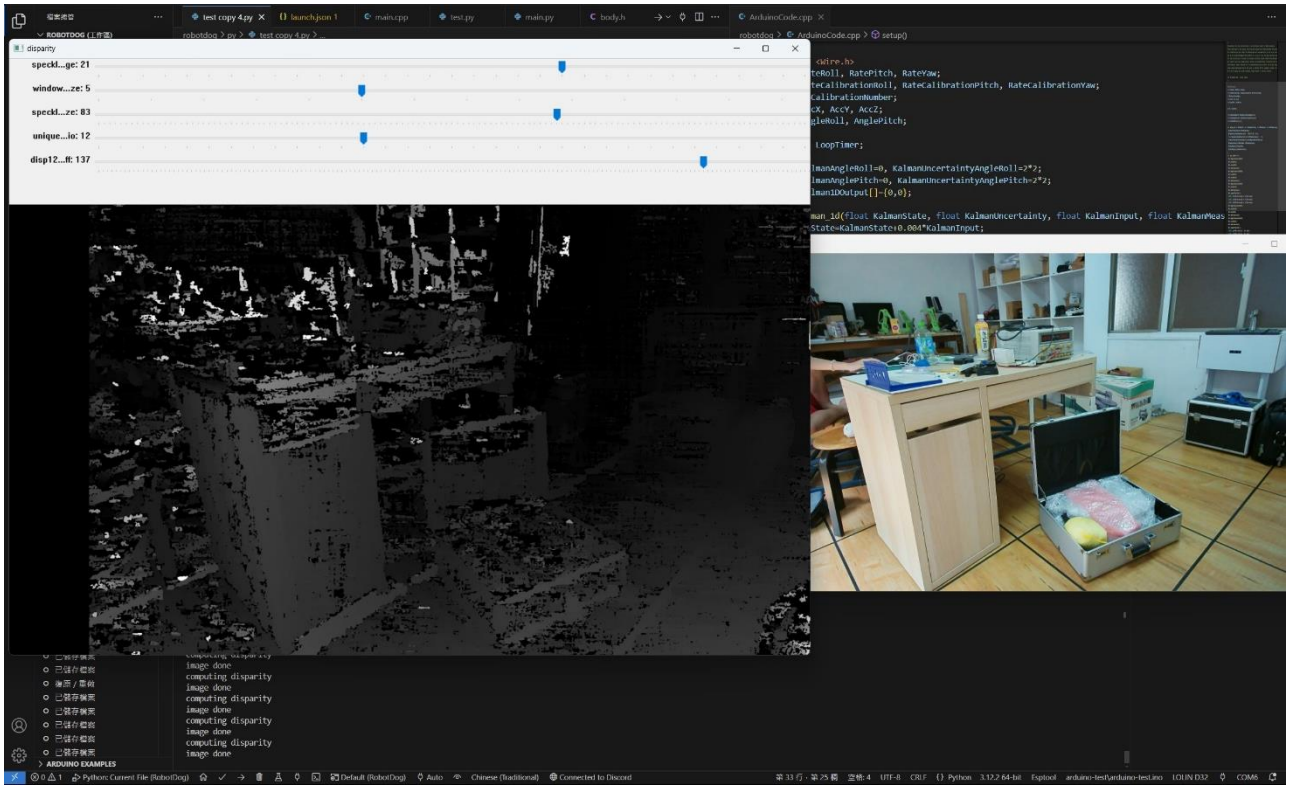


圖 35 雙目視覺調試程式

在獲得調試後的參數後就可以帶入雙目視覺的鏡頭程式獲取即時的視差圖，而視差圖顏色越亮的部分就是距離鏡頭越近。

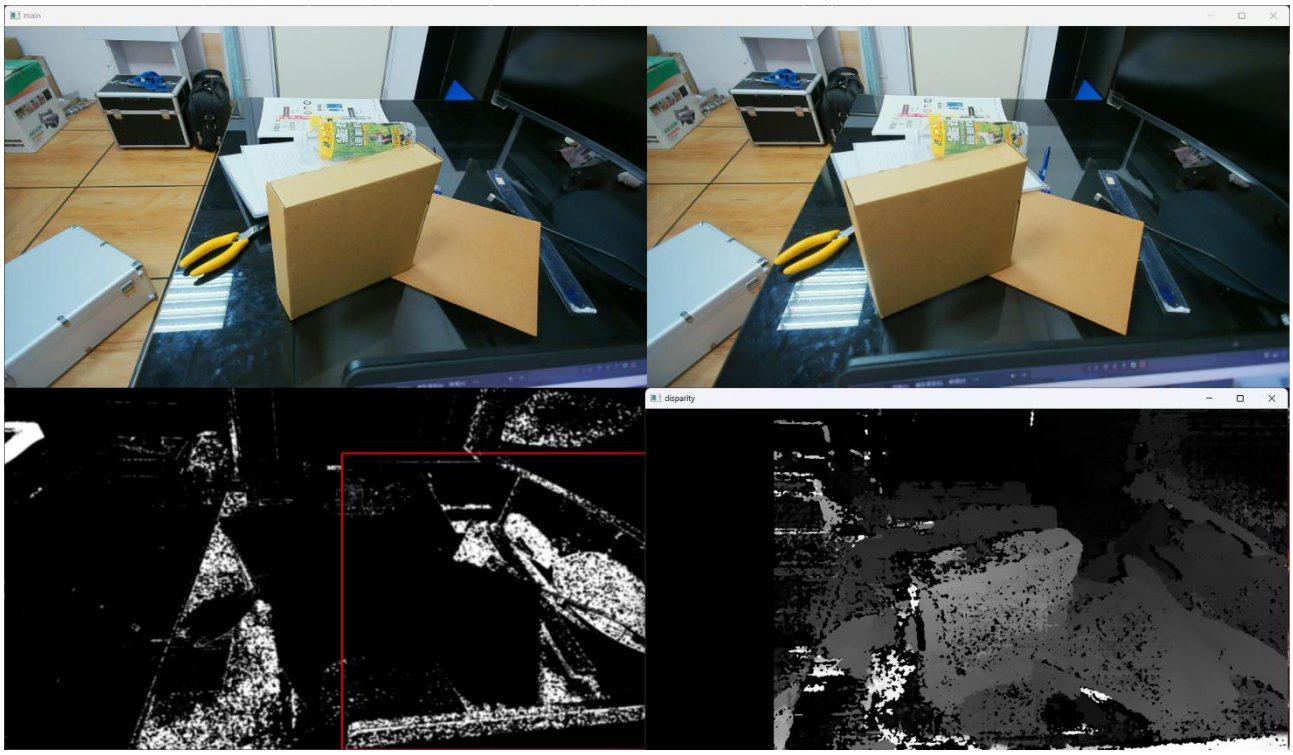


圖 36 雙目視覺計算視差圖

肆、研究結果

研究現況說明：

（一）研究產出的姿態演算法可依據視覺算法對地面物件識別後進行分析得出地面與仿生四足的角度差，並且計算自身姿態是否超過臨界角，若超過才開始自動進行姿態平衡，以節省非必要動作的方式節省電力需求。

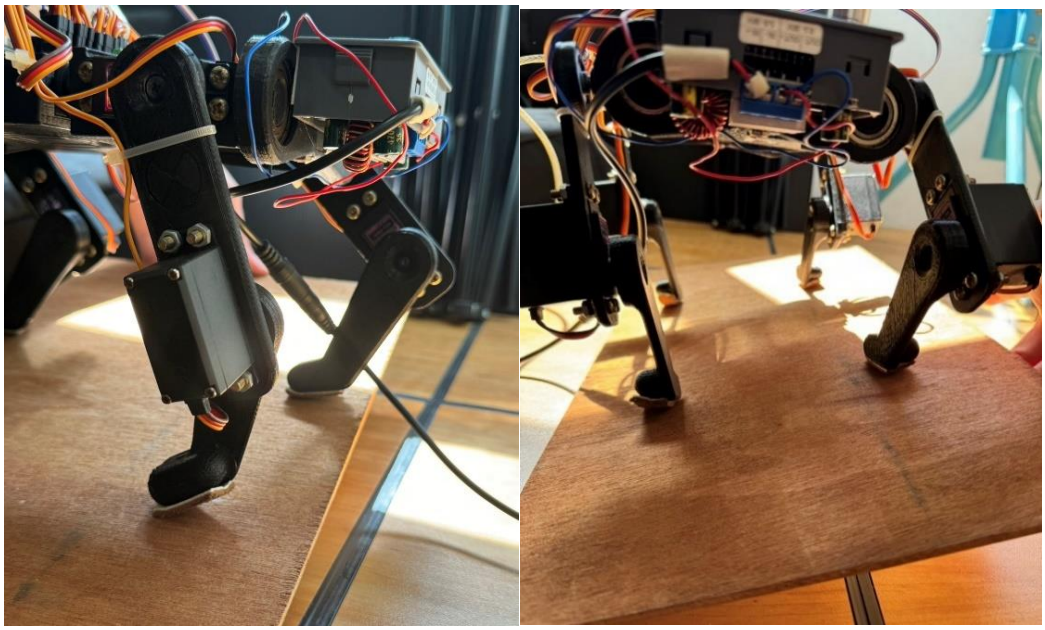


圖 37 腿部彎曲角度實測

（二）使用研究的姿態平衡模式在測試環境中透過直流變壓器的統計平均每小時可以節省約 100mAh 的電量，節省 20%電量並延長 25%的使用時間，根據電池容量計算在平衡與行走狀態延長總續航時間約 15 分鐘、延長站立時間約 20 分鐘，延長行走時間約 7 分鐘。

實驗數據統計：

表 1 電量及時間統計表

日期	測試時間	舊平衡模式消耗電量	新平衡模式消耗電量	省電比例	節省電量	每小時節省電量
04/05	10min	93 mAh	75 mAh	20%	15mAh	90mAh
04/12	30min	280mAh	232mAh	18%	48mAh	96 mAh
04/19	60min	546mAh	442mAh	20%	104mAh	104 mAh
04/26	40min	388mAh	318mAh	20%	70mAh	105 mAh

（三）我們將 PID 融入了舵機控制使其不容易因舵機快速停止時的慣性所產生非必要的抖動影響整體運動穩定性，也讓行走與平衡可以更加穩定。

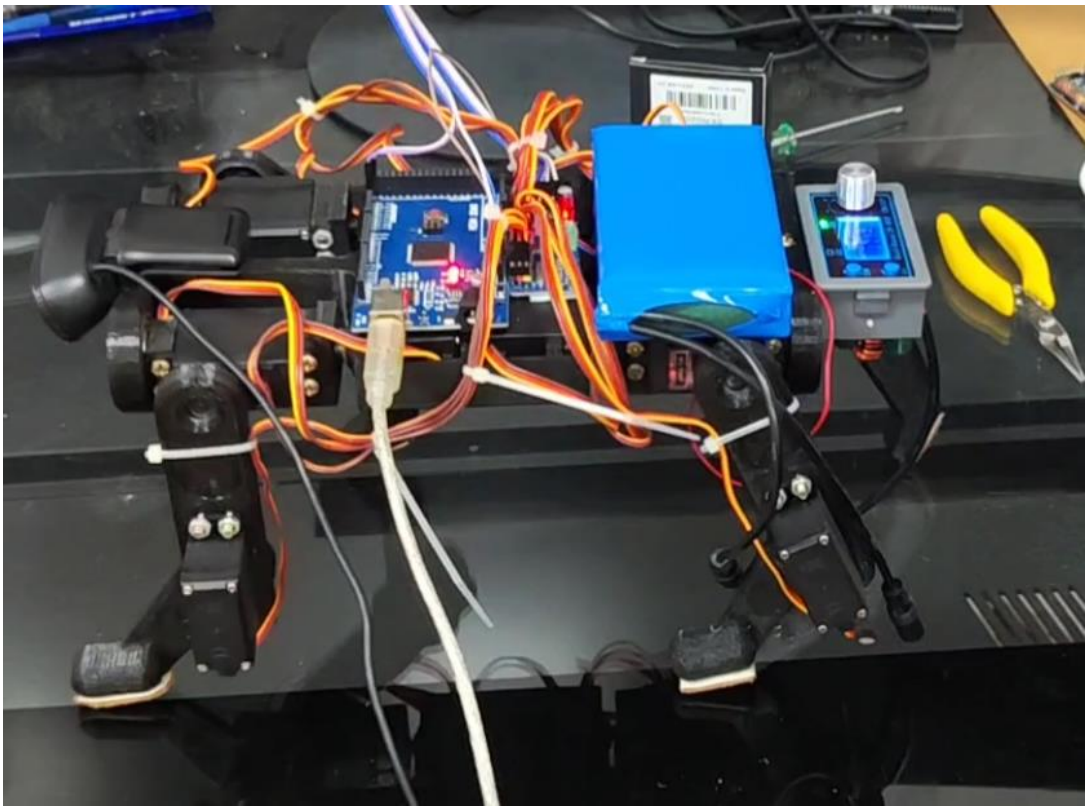


圖 38 四足機器人移動中姿態

（四）在不同的環境中進行有效的圖像處理和數據分析，經過獲取鏡頭影像，然後進行影像格式轉換後使用高斯模糊對噪點進行處理增加識別精度只需要透過物件的四個頂點分析長邊斜率即可，然後使用正反切換計算出傾斜角 θ，為避免大塊噪點影響辨識，二值化的圖片進行物件框選在考慮到影像雜訊、光照或是有與物件顏色類似的物品在旁邊，分離出二值圖最大的圖塊。

伍、討論

一、採用無刷電機以及 FOC 控制是否較佳？

目前市面上較多機器人都是使用無刷電機作為關節驅動，在實驗過程中，我們發現，舵機雖然具有減速機構使其扭矩較大，但其速度較慢，不足以執行高機動性的

動作例如跑和跳。無刷電機驅動所需電流較大以及 FOC 算法網路上的資料量較少，因此我們目前在持續不斷的實驗並且修正。

二、我們遇到的困難是什麼？

以雙目鏡頭進行物件辨識後，使用左右鏡頭視差計算物件距離的實驗已經完成，但由於顏色差別僅限於明亮度，所以對於單色地形的辨識精度仍不佳，所以目前仍在研究是否可以用點陣投影的方式加強精度。

在一開始我們是以單目視覺搭配 Midas 模型進行深度辨識，但單目視覺不能對非標準物件進行測量，而且它的距離是估算的，並非真正意義上的距離量測，準確度比較不穩定。

陸、結論

無人化生產和熄燈工廠一直是許多製造業企業積極追求的目標藉此降低人力成本並提高生產效率。在本研究中，我們設計一個採用視覺感測的四足機器人，其能夠對周遭環境變化進行視覺感知，並將影像資料回傳給筆電進行分析，以控制機器人的平衡和避障，其中最重要的特色就是，我們讓機器人平衡，不是整個達到水平模式，而是我們計算最大的傾斜角，只要機器人的動作回復到最大的傾斜角以內，即可進行下一個動作，因為機器人不至於發生傾倒的事件，這個創新的研究，可以讓我們節省機器人回復原動作所需的電力，如此將能增加其續航力，這對機器人來說相當重要，因為機器人 30 分鐘充一次電跟 45 分鐘充一次電，關係到其工作效能。而與使用傳統感測器(陀螺儀)的機器人不同，視覺感測模仿生物的視覺系統，可以捕捉更多環境資訊並做出反應，因此讓機器人具備更高的自主性和適應能力，本研究初步證明，這種以視覺取代傳統感測器的設計，在機器人控制方面具有良好的可行性。雖然目前的視覺感測誤差仍待降低，但本研究已為機器人技術導入新的可能性。未來我們將持續改進演算法，並結合更強大的運算能力，讓視覺感測在機器人上的應用更加精準。我們相信視覺感測機器人將是實現未來智能化生產與服務的重要一環，本研究為這類機器人的發展奠定初步基礎，雖然仍需不斷改進，但實現無人化生產的目標已漸漸接近。

柒、參考資料及其他

一、文淵閣工作室(2021)。Python 零基礎入門班。碁峰資訊股份有限公司。

二、李春雄(2014)。動畫圖解資料結構：使用 C#。全華圖書。

三、楊高科(2018)。圖像處理、分析與機器視覺（基於 LabVIEW）。清華大學出版社。

四、陳彥夆(2006)。摩擦行為的模擬與分析。國立成功大學機械工程學系。

五、Rene Ranftl* ,Katrin Lasinger* ,David Hafner ,Konrad Schindler and Vladlen Koltun (2021) ,Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. IEEE.

六、soumith chintala. (2019, August 20). Deep Learning with PyTorch: A 60 Minute Blitz. PyTorch. https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html.

七、OpenCV Bootcamp. (n.d.). OpenCV University. https://opencv.org/university/free-opencv-course/.

八、四足机器人制作(一) 腿部运动算法. (2019, June 8). Gitblog. https://imuncle.github.io/content.html?id=61.

九、Zhang, Y., Qian, Y., Ding, Y. et al. Adaptive walking control for quadruped robot by using oscillation patterns. Sci Rep 13, 19756 (2023). https://doi.org/10.1038/s41598-023-47022-x.

十、Aractingi, M., Léziart, PA., Flayols, T. et al. Controlling the Solo12 quadruped robot with deep reinforcement learning. Sci Rep 13, 11945 (2023). https://doi.org/10.1038/s41598-023-38259-7.

十一、Ferrolho, H., Ivan, V., Merkt, W. et al. RoLoMa: robust loco-manipulation for quadruped robots with arms. Auton Robot 47, 1463–1481 (2023). https://doi.org/10.1007/s10514-023-10146-0.