

雙節自主式移動機器人設計研製與應用

黃瑋翔^a、陳逸萱^a、卓冠廷^a、李祖聖^{*a}

國立成功大學 電機工程學系^a

摘要—本論文提出了擁有獨特機構設計的多功能輕型偵察車，其車身為兩節式，分別由兩顆主動輪帶動前行，前後車體的中間有一組馬達搭配上蝸桿蝸輪的機構，使其能夠任意地改變車輛的姿態，去應付各式崎嶇地形，並跨越多重障礙物，使本車應用範圍更為廣泛。車體上裝設景深攝影機以及慣性量測單元，讓車載電腦能夠實時計算車體當下的移動策略，並透過 Wi-Fi 將即時影像回傳給使用者，藉此來達到在險惡地形或是斷垣殘壁中偵察、搜救的功能。¹

關鍵詞：雙節式偵察車、開合機構、避障。

一、前言

在現今能夠跨越障礙物的輪型機器人中，大部分都是針對輪子來做設計與發想，像是主流有用三輪所組成的多輪型機器人[1]，可以自動地上下樓梯，或是擺脫傳統的輪子造型，以弧形輪輻作為主要輪的設計[2]。這些設計有很好的應用性，但在廣泛性方面可能稍有缺陷，因為當遇到較為複雜的障礙物時，這些機器人可能無法順利通過。

針對不平坦路面，常使用輪足型 (wheel-on-leg) 機器人來克服障礙物。例如[3]中的四輪足機器人，透過連桿與彈簧互相拮抗的方式來達到被動順應崎嶇地形的效果，而同樣是四輪足機器人的[4]，在機構被動順應地形的基礎上又再增加了步態最佳化的部分，使其穩健性大幅提升。火星偵察車通常使用六輪車模型，也有許多簡化的版本可參考[5]。但是，這些克服崎嶇地形的的方式會使機器人體積變得較大，若需要進入較狹小的場域，就難以適用。

除了針對輪足的部份來進行改善優化，也有些研究將機器人的結構改為多關節的形式，進一步增加其靈活度。關節式移動機器人的結構中，關節連接處通常需要推進力，同時通過關節自由地改變身體形狀。根據上述特性，有些研究也開發出相關的關節式移動機器人，例如用於檢查狹窄管道的[6, 7]，以及崎嶇地形搜索救援行動[8-10]等應用。樓梯是關節移動機器人難以活動之地形，不同於平面路面，因為樓梯需要在垂直方向上提供動力，同時也需要克服關節卡在樓梯邊緣的現象。先前的研究成功地開發了能夠克服這些挑戰的關節式移動機器人，例如在[10]成功地結合傳感器資訊和地形自適應特性，研製了一款機構及半自主爬樓梯控制方法，成功地克服了樓梯這種複雜地形所帶來的挑戰。

本篇論文結合機器人的優缺點，針對幾個場域進行功

能性設計，並加強擴大這些功能，以實現更多現實生活中的應用。例如，在災區搜救中，可以使用靈活輕巧的車型進入狹小的斷垣殘壁中，盡可能在黃金搜救時間內救援更多受災者；在地形探勘方面，搭配能夠翻山越嶺的功能，克服各種地形障礙，初步描繪當下的地形資訊，並實現實時的資訊共享。

本論文提出了一款雙節式多功能輕型偵察車，透過減少關節數以降低車體重量，並在關節處加上驅動力去控制前後節的夾角變化，搭配上深度感測相機和前後輪軸上的 Inertial Measurement Unit (IMU) 資訊來讓車子可以全自動地跨越障礙物，或是讓使用者根據相機畫面自行手動操作，達到偵察探勘的目的。

二、機構設計

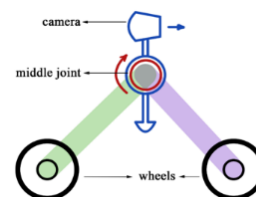
2.1 整體車身設計

為了提升車身對於地形的適應性，我們設計在車體中心增加一個關節自由度，使得車身可從中央關節分成前後兩節，兩節間的夾角可自由調整。前後共配置四輪，並且在中心處加裝攝影鏡頭。側向示意圖如圖一。圖中紫色與綠色桿件分別為前後節，桿件末端各有兩個車輪、紅色為可調整桿件夾角的中心轉子、藍色為固定方向向前的攝影機。下面再詳述各部份如何實現。

2.2 中央關節

中央關節需要撐起前後節，馬達所承受的力矩很大，且本車的中央關節需要旋轉的角度範圍不大，因此決定採用蝸桿蝸輪機構。蝸桿蝸輪可在空間有限的情況下實現高減速比及高轉矩的目的，適合本車中心大扭力且旋轉角度範圍小的情況，其驅動馬達也可移至前節上，達到簡化中心機構的目的。最終採用齒直角模數 1.25、減速比 40:1、條數 1 的蝸桿蝸輪，選擇 KHK 小原齒輪的 BG1.25-40R1J10 蝸輪、SW1.25-R1 蝸桿。

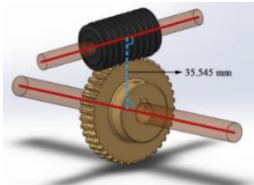
蝸輪之基準圓直徑為 50.09mm、蝸桿之基準圓直徑為 21mm，兩者的轉軸垂直但不相交，兩軸之間的距離為 $(50.09 + 21)/2 = 35.545\text{mm}$ ，如圖二所示。



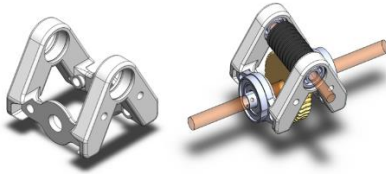
圖一：車體側向簡易示意圖

¹ 本研究由國科會贊助，計畫編號 MOST 109-2221-E-006-196-MY3。

*: Corresponding author.



圖二：轉軸間位置關係



圖三：轉軸支架與蝸桿蝸輪配合圖

為維持圖中轉軸之位置，我們設計了轉軸支架並在支架與轉軸之間配有軸承，此支架使用 3D 列印的方式製造，如圖三。前節將與支架固定；後節將與蝸輪、蝸輪軸固定，此作法即可使蝸桿蝸輪的旋轉帶動前後節開合，蝸輪旋轉角度即為前後節夾角角度之變化量。

2.3 前後節機構與車輪配置

車體前後節為兼顧強度及輕量化的需求，採用了輕巧的1515鋁擠型作為主體架構，相對市面上常見的2系列、3系列鋁擠型重量輕了許多，鋁擠型的擴充性對於後續其他硬體的安裝也十分方便。前後節各以兩支1515鋁擠型縱向形成骨架，橫向再鎖上壓克力板以確保鋁擠型之間平行。

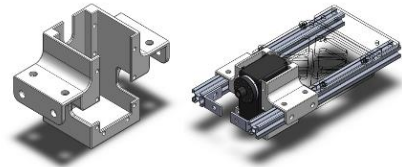
前段提及的中心轉子馬達配置在前節，因此其馬達架需要能與兩隻鋁擠配合，並且使馬達驅動軸與蝸桿轉軸共線，我們因此設計了能將馬達固定在鋁擠型之間的3D 列印馬達架，除了提供支撐馬達的功能外，也能代替壓克力板維持兩鋁擠之間的平行度。鋁擠間剩餘的位置可放置控制板、電池等硬體。

車體前後節為適應崎嶇地形進行開合時，部分姿態可能會造成底盤過低接觸到突出地形的情況，因此需要在車輪與車體之間架高，使得車輪轉軸低於前後節之鋁擠型，進而達到底盤相對抬升的效果。我們採用的架高方式是將車輪馬達架往下設計，使馬達與車輪降低，並且選用直徑 96mm 的大型攀爬車輪，車輪的大直徑與良好抓地力皆降低地形變化對車體前進所產生的影響。

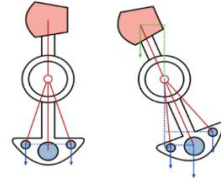
2.4 相機單軸穩定器

由於車體在行進過程中會不斷變形，車體各部位會隨著前後節開合而有方向與角度的變化，因此無法找到一個適合的位置放置鏡頭，為了解決這個問題，我們利用簡單的槓桿原理在車體中心設計一個可以保持相機方向的單軸穩定器，使鏡頭可以持續記錄前方的畫面，設計概念如圖五。

上方紅色部分為相機Intel® RealSense (約為73g)，下方藍色部分為可調節重物，中間圓圈處配有軸承，使整體可以繞著中心軸自由轉動。下方重物部分分成中間大重物與兩側小重物，於直立平衡狀態時(圖五左)，相機與大重物相對中心支點力臂皆為0，兩側小重物相對支點會產生力矩並且互相平衡維持穩定直立；當穩定器傾倒時



圖四：後節骨架示意圖



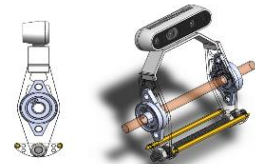
圖五：相機單軸穩定器概念圖

(圖五右)，兩側小重物因傾倒產生的力臂較大，綜合下方重物所有藍色力矩，足夠將上方相機產生的綠色力矩抵銷，並回歸到直立平衡狀態。實際設計架構是由壓克力板和3D列印件組合而成，重物由螺桿、螺母、六角銅柱構成，方便隨時調整重物的重量，前後小重物的妥善配合也可以調整鏡頭的傾角，以達到最穩定的效果，如。

2.5 機構整合與其他硬體配重

綜合上述幾段，將中心部分、相機單軸穩定器、前節、後節全部整合之後，等角視圖如圖七，並且如圖八所示車體前後節可以展開。

車體上配置主控電腦 NVIDIA Jetson Nano、馬達電池、電腦電池、控制板 Arduino Mega 2560、兩個慣性量測單元 x-IMU，需要將這些硬體配置在前節或後節，前節的配重需要大於後節，使整體重心向前，在翻越崎嶇地形收合前後節時才不會容易向後傾倒，提高穩定度與越野能力，因此我們將主控電腦 NVIDIA Jetson Nano、電腦電池放置在前節；馬達電池、Arduino Mega 放置在後節，IMU



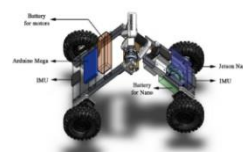
圖六：相機單軸穩定器側視圖、等角視圖



圖七：整體機構等角視圖



圖八：車體展開姿態



圖九：硬體配置圖



圖十：全車實體圖

則放置在前後節末端中心，以便感測兩節傾角，Arduino Mega與IMU的重量輕可忽略不計，前節有Jetson Nano、電腦電池和中心轉子馬達，後節則僅配有馬達電池的重量，達到整體重心向前的目的。所有硬體如圖九，全車共重3.6 kg。實體照如圖十。

三、硬體配置

3.1 馬達選用與配電

伺服馬達選用 Robotics MX-28AR 和 MX-106R，供電皆為 12V，可提供約 2.5 N m 以及 8.4 N m 的扭力，重量分別為 77g 和 165g。由於定位為輕型偵察車，車體需輕量化，考量到驅動輪並非像中央關節需要較大的扭力使車體兩端摺疊，因此採用前者作為動力來源，後者作為中央關節使用的制動器。電池選用 6S LiPo ELL MAX 22.2V 5800 mAh 25C，供應五顆伺服馬達所需電量，從 Robotics e-Manual [11, 12]上得知 MX-28AR 和 MX-106R 在 12V 供電的情況下所需電流分別為 1.4A 跟 5.2A，電流最大值需 10.8A，考量到中央關節並非時刻轉動，因此假設兩者使用時間比為 8:2，電流平均使用量則為 5.52A，5800 mAh 容量的電池，可供馬達轉動約 1 小時，非常充裕。

3.2 通訊裝置和方式

馬達間的通訊裝置選用的是 Robotics U2D2 [13]，利用 RS-485 的通訊協定。兩款馬達可透過 Master/Slave Mode 將其串接在一起，並以同步電纜(Sync Cable)連接省去額外線材的重量。透過 Robotics U2D2，一款小型 USB 通訊轉換器，可做到 Control PC 和 Dynamixel 馬達之間的訊號傳輸。

3.3 車載電腦及供電

車載電腦採用 NVIDIA Jetson Nano，選用此款電腦的原因除了體積小、重量輕以外，經實測後亦能順暢地運行系統及程式。供電上選用 Waveshare 公司推出的 UPS 不斷電供應系統，其上附有小 OLED 螢幕，可即時查看剩餘電量和 CPU 使用率，使用上是相當方便。

3.4 相機及深度感測元件

經考量及比對市面上的深度感測器後，團隊最終選用 Intel® RealSense D435i 作為車載鏡頭，主要是因為此款鏡頭同時兼具了擷取 RGB 與點雲影像的功能，加上官方提供完善的 RealSense SDK [14]，其開發的 RealSense Viewer 可供使用者快速地調整相機參數，並且即時預覽影像。

3.5 慣性量測單元 IMU

為了有效估測車體姿態，決定在車輪前後軸中央裝上 IMU，透過前後節的角度來計算出車子的中間夾角，進而描繪出整台車的姿態。選用的 IMU 為 x-io Technologies 所生產之 x-IMU，其提供了九個維度的資料，分別是三軸陀螺儀、三軸加速度計以及三軸磁力計。由於 Jetson Nano 的 GPIO pin 腳直接與 IMU 做溝通時，程式方面常出現衝突，且收到的資料傳輸斷斷續續不穩定，因此加裝了

表 I
自轉測試結果

	中心偏移量(cm)	
	左自轉	右自轉
測試一	3.1	4.3
測試二	4.5	6.0
測試三	3.7	3.6
平均	3.77	4.63

Arduino Mega 2560 作為中間媒介。利用 Arduino 的 5V 腳位對兩個 IMU 進行供電，再透過 RX1 和 RX2 腳位分別接收前後節 IMU 的資料，雖然多了一個部件，但強化了通訊方面的穩定性以及避免電池供電不穩的情況。

四、馬達控制與車體性能測試

4.1 馬達控制

本次使用的馬達控制方式為速度控制，Dynamixel MX 系列的馬達內部含有控制板，可另做開發撰寫程式，透過 Robotics Wizard 2.0 進行馬達 ID 和 Baud rates 設置，使操作者能根據給定的 ID 去操控馬達轉速，透過 C++ 將官方給的 Dynamixel SDK 進行包裝，再根據使用者需求寫成函式進行使用。由於車輪的轉動幅度超過 360°，位置控制並不適用，加上考量到 Control PC 的運算效能，同時希望能即時修正目標方向上的偏差，因此最終選用速度上的 PD 控制。

4.2 運行方式

運行方式分為手動模式和全自動模式，前者功能為基本的前後左右移動，並根據輸入的速度、方向行進或停止，也可以操控中央關節的開合，使其做到攀爬或穿越狹小地形的操作；後者則能依據車載電腦計算出不同策略而做出相對應的動作。

4.3 性能測試

本小節透過設計去進行自轉中心偏移量測試，了解硬體誤差，再透過軟體做修正。本車自轉的方式是讓前後節車輪軸中點作為旋轉中心，讓輪子分別向前與向後轉動，達到自轉目的。首先將車體中心位置設為自轉中心，針對左右自轉，進行三次測試。實驗結果列於表 I，結果顯示此項誤差偏大，主因是前後節皆為主動輪，理當產生兩個自轉中心，又車體為剛性的且無法共轉，於是前後節就會與地面產生蠻大的相對運動，導致偏移量跟預期結果差異甚大。因此需透過軟體進行修正，以確保車子的轉動角是否符合預期結果。

五、運行程式設計與跨越障礙之決策邏輯

5.1 主控程式

主控程式如圖十一所示，主要包含馬達控制、處理 RealSense 和 IMU 資料以及行進策略。這些感測器所蒐集到的資料，經過程式的運算過後，可以給予車子在不同情況下所應執行的指令動作，藉此達到自主移動的效果。另外也加入手動控制模式，以便於車子在發生不可預期

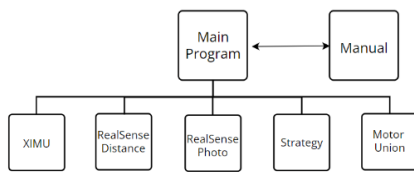
意外時，可以馬上手動介入操作；或是在進行搜救任務時，透過相機回傳的影像，可以讓使用者透過程式遠端操控車子，進行更高難度的行動。

5.2 車體姿態估計

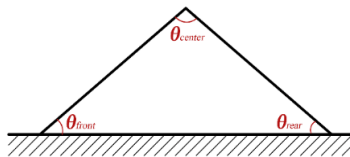
對於直行跨越障礙物的過程，最需觀測的就是車子前後節的俯仰角(pitch axis)。我們定義前節與地面的夾角為 θ_{front} 、後節與地面的夾角為 θ_{rear} ，接著可以將車子前後節與地面的夾角以及車子中心夾角 θ_{center} 簡化成一個三角形，因此可以輕易地由下列式子計算出車子的中心夾角。

$$\theta_{center} = 180^\circ - (\theta_{front} + \theta_{rear}) \quad (1)$$

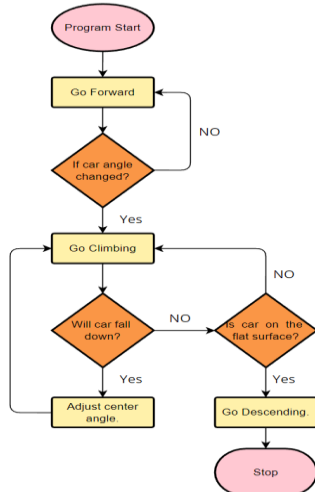
在得到三個角度之後，加上車體前後節的長度，並透過運動模型的建立，我們就能估測出車體目前的姿態。



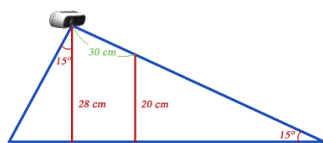
圖十一：程式架構圖



圖十二：車體姿態估計示意圖



圖十三：程式決策邏輯圖



圖十四：RealSense 偵測距離示意圖

5.3 車子行進策略與高度限制

因為我們的目標是實現一款能夠自主移動並跨越障礙的偵察車，所以在車輛行駛過程中，首要任務是確保車體的安全性。在自動模式下，根據車體姿態的不同，我們定義了幾個行駛狀態，包括初始調整(Adjust initial pose)：在行進前會先將車體調整到一個適應性較高的姿態；前進(Go Forward)：在水平姿態下穩定前進；攀爬(Go Climbing)：車體傾斜並開始攀爬；中心角度調整(Adjust Center Angle)：調整車體中心夾角以避免傾倒；攀爬調整(Keep Climbing)：在爬行過程中閉合中心角度以轉移重心；頂部狀態(On Top)：車輛攀爬後回到水平狀態；下降(Go Descending)：車輛進行下降動作。圖十三是車輛在自動模式下的程式決策邏輯流程圖。

根據圖十三我們可以觀察到，當車子行駛過程中，若發現前後節的 IMU 角度變化超過設定的閾值，就能判斷車子的前輪此時已經接觸到障礙物。於是策略程序將當前狀態改為攀爬狀態，隨後車子將持續往前爬行。同時，需要不斷估測車體姿態以及監測中心角度是否超過臨界值。若超過臨界角度，則表示車子存在傾倒風險。此時必須張開車子的中心角度以免發生翻車事故。

在爬行過程中，若遇到坡度較為陡峭或垂直障礙物，車子將閉合角度，以使重心轉移到靠近前節的位置，從而帶動車體向上攀爬並成功克服障礙。當車體已經完全在障礙物之上時，我們設計會讓車子在上面進行姿態調整，如此才能以較為安全的角度緩步下降，至於在下降的過程中，也會時刻地監測車體中心夾角，若超過臨界角，同樣需要讓車子中心角度張開一些。

除此之外，針對高度過高的障礙物，我們設計了避障策略，以免在攀越過程中造成翻覆。實際測試顯示，偵察車可跨越的最大安全高度約為 20 公分。若偵測到的障礙物高度超過此範圍，偵察車將選擇轉向其他路徑行駛。為了測量障礙物高度，我們採用簡易的估測方法，以降低運算資源的需求，估測方法如圖十四所示。車體相機架設的高度為 28cm，我們特意將相機往前傾斜約 15°左右，會這麼做的原因是，若相機與地面垂直的話，會很容易忽略掉一些高度較低的障礙物，經過實際測試，我們發現將相機前傾 15°是一個較佳的選擇，於是安全偵測距離就如(2)式所計算。

$$Dis = \frac{camera_height - safe_height}{\sin 15^\circ} \quad (2)$$

六、實驗結果

本論文主要目標，是希望能夠克服崎嶇地形和跨越障礙物，之前已介紹過機體設計，實現方法和策略。接下來我們將透過兩個實驗，來驗證所提方法之效果。

6.1.1 實驗一：環境設置與說明

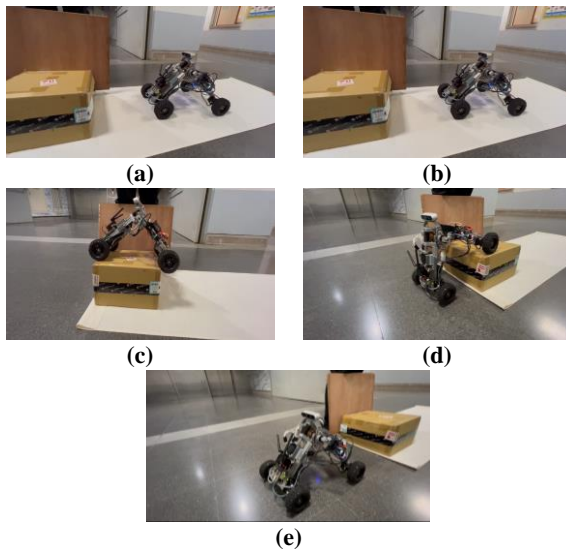
此處的實驗內容是爬紙箱的結果，會以紙箱來做實驗是因為车子在跨越紙箱的時會經歷上攀及下降的過程，如圖十五，因此能同時分析到兩種情形，紙箱的取得也相對容易，使我們能在實驗室中就能進行參數調整。本實驗

所選用的障礙物為一高 18 公分、寬 30 公分、長 32 公分的紙箱，另外為了克服磁磚地面過於光滑的問題，我們在地面鋪上地毯，藉以增加摩擦力。

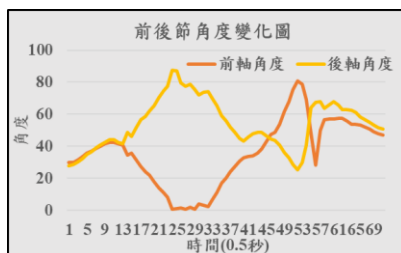
6.1.2 實驗一：結果與討論

最一開始的角度變化是因為感知到車體過於水平，因此策略進入了 Adjust initial pose，此策略是為了讓車體可以用較為穩定且適應性較高的角度來行走，此處我們所設的最佳角度為 40~50 度之間。調整過後我們的策略就進行到 “On floor” 的階段，此時車體緩慢前行。

接下來進入到第三階段 Go Climbing(圖十六第 13 時間點過後)，可以看到前後節角度皆以非常大的斜率往反方向變化，這是因為前輪已經接觸到障礙物並開始向上爬行了，因此前節角度會趨向水平角度來變化，至於後節則會越趨向垂直的 90 度，如圖十五(b)。在第 24 時間點，因為後節角度將到達我們所設置的臨界角 85 度，所以策略進入了 Adjust Center Angle 來打開中心轉子的角度，避免車子往後傾倒。可以注意到在此之後，前節的角度已趨於水平，表示前節已完全平躺在紙箱上，接下來就是要讓後節也向上攀升，策略此時進入到 Keep climbing，讓後節能以更好的角度向上爬，另外也會針對中央關節做調整，使其能夠以夾取的方式向上。所以在這之後，前後節的角度又會漸趨一致，直到車體完全攀上紙箱，如圖十五(c)，此時的策略為 On top，會在紙箱頂部做角度調整，如此才能以較為安全的角度進行下降。



圖十五：車子跨越紙箱過程：(a)為初始狀態 (b)為上攀姿態 (c)為攀爬至頂部 (d)為下降姿態 (e)為下降完成



圖十六：爬箱子過程中車體角度變化圖

再來看第 45 時間點過後，前節的角度快速上升，而後節的角度則下降，表示進入了 Go Descending 狀態，如圖十五(d)，至於在第 56 時間點，前節的角度突然掉到了 28.24 度，則是因為 IMU 的量測誤差所導致，並非車體角度的大幅度改變，在這之後，可觀察到前後節的角度又逐漸趨於一致了。當前後輪都接觸到地面之後，我們會讓車體再進行一次姿態調整，如同最初所進行的方式，方能再以穩定角度行進。

6.2.1 實驗二：環境設置與說明

第二個實驗內容是爬樓梯性能測試，以樓梯來做實驗是因為一般輪型車大多難以攀爬階梯地形，爬樓梯的項目對於輪型車越野能力相當具有指標性。在車體行經過程中會遇到連續重複階梯狀地形，這也更利於我們在數據分析上找到規律性。本實驗自行製作的樓梯單階高為 18cm、寬為 40cm。沒有選用現成的樓梯是因為我們所找到的樓梯側階面都是超過 90° 的，並非與地面垂直，不利於本作品之運行，因此改用自行製作的三階樓梯。

6.2.2 實驗二：結果與討論

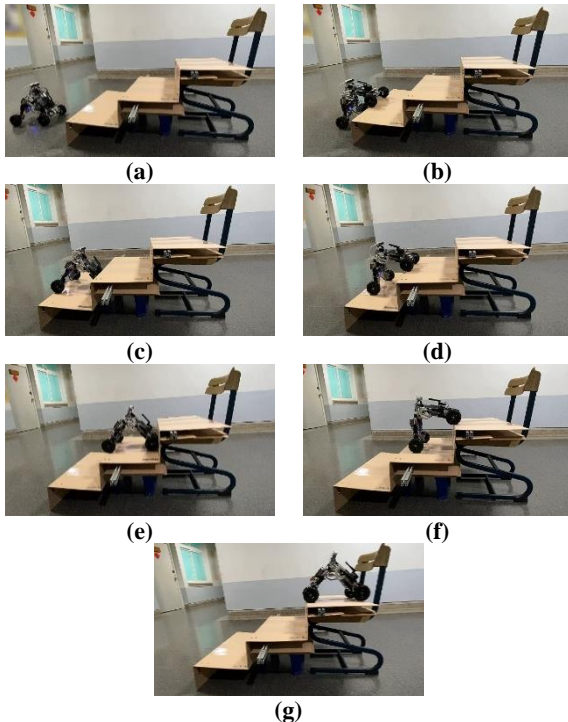
一開始可以看到前後角度皆無變化，不同於實驗一開始的初始姿態調整，本次實驗最初就在我們所設的最佳角度範圍內，因此就跳過了 Adjust initial pose 策略，直接進入 Go Forward 階段前行。從圖十八看到前 15 個時間點的角度皆保持恆定，但兩者卻沒有完全重合，原因應為前後節車身長度的不等所導致的。

在圖十八第 16 至第 20 時間點為 Go Climbing 階段，前輪接觸到樓梯垂直面並爬升當中，在中心轉子沒有運作的狀況下，車體視為剛體，前後角度變化量之絕對值相等，但由於前後節角度正方向定義不同，兩者斜率因而相差一個負號。

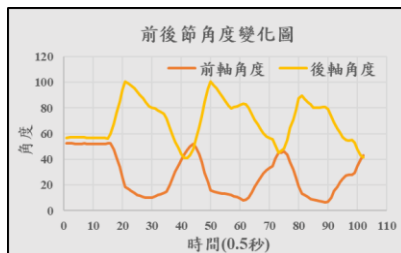
後節角度在第 21 時間點來到第一次峰值，後節角度達到可能會傾倒的臨界角，為避免車體重心超出後輪接觸範圍而傾倒，中心轉子必須張開使 θ_{center} 增加，也就是 Adjust Center Angle 階段。第 29 時間點，中心角度已調整完畢，此時前輪在第一階上、後輪在地面，策略進入 Keep climbing，前後輪皆在水平面前進，因此圖十八第 29 至第 34 時間點間兩者的角度變化皆不大。第 35 時間點後輪接觸到了第一階的垂直面，同樣是在 Keep climbing 階段，第 35 至 40 時間點的斜率相對前段大了一些。

第 40 時間點時，後輪也已經爬上了第一階，但因為角度沒有在我們設定的最佳角度內，因此車體在第一階上收合中心轉子進行了初始姿態調整，可以在圖表上看到前後節在第 40 至第 44 時間點間角度同時上升，完成姿態調整之後，再開始下一階的攀爬。

圖十八整體可以看到三組形狀類似的圖形，即是分別在三階樓梯重複了上述動作，造成些許差異的原因在於自製樓梯架構不夠堅固，木板階面偶爾產生凹陷、變形的狀況，因而導致三組數據的不同。第 40 至第 44 時間點的初始姿態調整也只有在第一階有發生，第二、三階爬上水平面時角度皆剛好落在最佳角度範圍內，圖表上就不會在二、三階看到短暫兩者角度同時上升的階段。



圖十七：車子攀爬樓梯過程：(a)為初始狀態 (b)為上攀第一階姿態 (c)為完成一階 (d)為上攀第二階姿態 (e)為完成二階 (f)為上攀第三階姿態 (g)為完成三階



圖十八：爬樓梯過程中車體角度變化圖

結論

本論文旨在設計一具高度靈活性的多功能雙節輕型偵察車，在一般輪型機器人的基礎上，添加了中心可變形的自由度，此設計使得車體對於多變崎嶇的地形有更高的適應性，中心自由度可依據不同地形，結合感測器數據姿態判斷，歸納出合適的行進策略，大幅的拓展此偵察車在不同工作場域的應用範圍。

相較於其他越野偵察車，本車體積小、重量輕，且具有速度、控制性、機動性和適應性等優勢，輕量、敏捷而不失強度。此外，它還可以進行全自動行駛，亦可讓使用者親自動手操作，透過車上的鏡頭能記錄下實地影像，代替人類到達危險或狹小的地形進行偵察的任務。

未來，本研究預計透過 SLAM 等方法對周邊環境進行建模，以更全面的方式蒐集環境資料，進一步提升偵察車的機動性和安全性，並且能與環境進行更多元複雜的互動任務。同時，加入防水防塵的設計能夠增加車輛的堅固性和可靠性，進一步擴大應用範圍。未來，這輛偵察車甚至可以乘坐太空梭，穿越浩瀚無垠的銀河，抵達遙遠的不毛之地火星，為人類開拓新天地。

參考文獻

- [1] J. Han, S. Lee, E. Hyun, B. Kang, and K. Shin, "The birth story of robot, IROBIQ for children's tolerance," in *Proc. The 18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009.
- [2] Y. Kim, J. Kim, H. S. Kim, and T. Seo, "Curved-Spoke Tri-Wheel Mechanism for Fast Stair-Climbing," *IEEE Access*, vol. 7, pp. 173766-173773, 2019, doi: 10.1109/access.2019.2956163.
- [3] A. Bouton, C. Grand, and F. Benamar, "Design and Control of a Compliant Wheel-on-Leg Rover Which Conforms to Uneven Terrain," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2354-2363, 2020, doi: 10.1109/tmech.2020.2973752.
- [4] M. Bjelonic et al., "Keep Rollin'—Whole-Body Motion Control and Planning for Wheeled Quadrupedal Robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2116-2123, 2019, doi: 10.1109/lra.2019.2899750.
- [5] N. K. #crazyNK, "How to Make a Mars Rover / Rocker bogie Robot - Stair climbing." <https://www.youtube.com/watch?v=3Zx7tGwF5g> (accessed Apr 03, 2023).
- [6] T. Zheng et al., "Design of a Robot for Inspecting the Multishape Pipeline Systems," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 4608-4618, 2022, doi: 10.1109/tmech.2022.3160728.
- [7] A. Kakogawa, K. Murata, and S. Ma, "Automatic T-Branch Travel of an Articulated Wheeled In-Pipe Inspection Robot Using Joint Angle Response to Environmental Changes," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 7, pp. 7041-7050, 2023, doi: 10.1109/tie.2022.3201301.
- [8] W. Zhang, S. Lyu, F. Xue, C. Yao, Z. Zhu, and Z. Jia, "Predict the Rover Mobility Over Soft Terrain Using Articulated Wheeled Bevameter," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12062-12069, 2022, doi: 10.1109/lra.2022.3211153.
- [9] S. Kakkar and M. A. Minor, "Fast and Reliable Motion Model for Articulated Wheeled Mobile Robots on Extremely Rough and Rocky Terrains," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2252-2259, 2019, doi: 10.1109/lra.2019.2900523.
- [10] M. Tanaka, M. Nakajima, Y. Suzuki, and K. Tanaka, "Development and Control of Articulated Mobile Robot for Climbing Steep Stairs," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 531-541, 2018, doi: 10.1109/tmech.2018.2792013.
- [11] ROBOTIS. "ROBOTIS MX-28 e-Manual." https://emanual.robotis.com/docs/en/dxl/mx/mx-28-2/?fbclid=IwAR1_znu-0i9suUmRedQxxRaWRzajrcd7hNZHab9EDGPB4uR6qQfWV_m3r10. (accessed Mar 25, 2023).
- [12] ROBOTIS. "ROBOTIS MX-106 e-Manual." <https://emanual.robotis.com/docs/en/dxl/mx/mx-106-2/?fbclid=IwAR3MfGPrAN61LRKSc8P7boEvkrzpye55muzEXx5ngWZO-NiVaoXh-v31nxA>. (accessed Mar 25, 2023).
- [13] ROBOTIS. "ROBOTIS U2D2 e-Manual." https://emanual.robotis.com/docs/en/parts/interface/u2d2/?fbclid=IwAR0nb-9phfLXGS4emE5SNBFF_h-37al0nFXtKbUrLnHnk44jMZ2Gn9egk7U. (accessed Mar 25, 2023).
- [14] I. Corporation. "IntelRealSense/librealsense." <https://github.com/IntelRealSense/librealsense> (accessed Mar 25, 2023).