

Random grouping

- 1.如何從網站中抓取所需之資料(如名單)?
- 2.順利抓取名單後如何運用於程式中並使其能夠正確的讀取?
- 3.如何在上述皆達成後順利達到分組的效果? 我們必須要了解 `import` 的 `request`、`ast`、`random` 是甚麼模組。

(1) `Request` 為能夠下載網頁的一項 `python` 中的模組，我們能夠使用 `request.get`(加入網址)的方式來達到擷取網頁資料的目的。

參考連結:<https://blog.gtwang.org/programming/python-requestsmodule-tutorial/>

(2) `ast` 中的 `literal_eval` 可以將剛剛擷取下來的網頁資料(僅允許為字符串，字節對象(`bytes`)，數值，元組，列表，字典，集合，布爾值和 `None`)才可以進行轉換。

參考連結: <https://docs.python.org/zh-cn/3/library/ast.html>

(3) `random` 中有許多不同的用法，在每周抽點中我們使用的是 `shuffle` 的用法，此用法會將給定的元素隨機打亂，以達到亂數的目的。

參考連結: <https://ithelp.ithome.com.tw/articles/10207483>

最後我們可以經由 `for` 迴圈對剛剛打亂的元素進行我們所想要的分法已達到亂數分組中的分組的目的。

Portable system

問題:如何對現有的可攜進行更新 python 至 3.8.2 的動作,可攜系統的完整安裝流程,及 python3.8.2 的運行需要做哪一些更動?

(1)對舊有可攜進行更新將可攜系統下的 python 3.7.3 系統更新 python 3.8.2

1. 進入 <https://www.python.org/downloads/windows/>
2. 往下滑到最底,下載 Windows x86-64 executable installer(上下兩者分別為 Windows x86-64 embeddable zip file 和 Windows x86-64 web-based installer 這兩者是嵌入式 64 位元 python 的可執行壓縮檔案和 64 位元網際端的可執行檔案)在本次可攜的更新中不會使用到
3. 接著開始下載切記不要勾選 pip 選項,接著按下 next
4. 先將下載下來的 python3.8.2 放置桌面,並且將 y 槽下的 data 新增加一個 py382 的資料夾方便之後可以做測試是否 3.8.2 之 python 可以正常的運行
5. 將剛剛下載之 python3.8.2 放置於剛剛創建的 py382 中切記不要將原先的 py373 刪除,若之後 py382 的執行產生錯誤才可以使用 py373 做返回原版本的動作
6. 更改執行檔 start_mdcourse.bat 將底下 py373 處更改為自己下在 python3.8.2 所儲存的資料夾名稱
7. 重新啟動剛剛更新的執行檔
8. 執行 `python get-pip.py`
9. 執行 `python -m pip install flask flask-cors markdown lxml bs4 pelican leo`

four wheel car

本資料由 [40723222](#) 提供

BubbleRob tutorial:在此教程中可以透過我們自行設置的 BubbleRob 機器人進行初步的模擬，在上學期中我們有實際的以 v-rep 操作過這個教程，在這個教程中有含括了很多包括如何建造模型(機器人本體、輪子、後方支撐滑塊、障礙物)，並且經由設定好的距離傳感器去感知並且避開我們所設立的障礙物，其中很重要的要點是如何去設計指令使輪胎兩側的傳動器依照我們所想要的方式去做運動，且當距離感測器接近我們所設立的障礙物時，他必須要做出相對應的反應部會使其碰到障礙物，另外還必須要考慮到視覺辨識系統，使機器人在移動時我們能夠觀測到機器人前方之狀況，此教程能夠使我們更加有效率的使用 v-rep 這套模擬軟體。

本資料由 [40723243](#)

Line following BubbleRob tutorial:在此模擬教程中我們銜接上面的

BubbleRob 教程，我們想要使其能夠按照我們所規劃的路徑移動，在四輪車的

方面這樣的功能能夠使我們有效的規劃四輪車的行走路徑，以達到更好去控制

四輪的目的，在無外部控制器或者是想要直接在模擬器的環境中直接模擬可以

採用該方法達到比較好的效率。

本資料由 40723240 提供

External controller tutorial:遠端控制器教程中，當我們想要從遠端電腦的方式控制我們在模擬中的物件例如:四輪車或機器人時我們就會需要一項遠程控制的控制器，我們可以參考 [ROS tutorial](#) 及 [ROS 2 tutorial](#) 來達到我們想做控制

的想法，ROS 主要比較像是對於機器人的中間控制系統，他能夠藉由傳感器來指引機器人該往哪個方向移動，如下圖所示，就是 ROS 較好的解釋。

ROS:比較像是在負責為機器人的各個元件進行溝通與操作的一個框架，以機器人的行走為例，要讓機器人走路需要控制其底下的馬達、還有一些感測器以進行避開障礙物的功能等等，ROS 內就有這種類型的函式可以讓控制馬達的程式得以與接收感測器的程式溝通。



ast 模組:<https://docs.python.org/zh-cn/3/library/ast.html>

Rendom 模組:<https://ithelp.ithome.com.tw/articles/10207483>

Request:<https://blog.gtwang.org/programming/python-requestsmodule-tutorial/>

mde.tw 中分組之程式:<http://mde.tw/cd2020/content/Grouping.html>

v-rep 使用手冊: <http://www.coppeliarobotics.com/helpFiles/index.html>

Assignment 2

Topic 0 :

三種分工方式

1. 主導概念化流程
2. 主導產品外觀及內部組件
3. 協同流程
 - 1.設計團隊搜尋設計目標>制定草稿>3D 建模>初代樣品
 - 2.工程團隊設計細節>製作模型>挑選零件>設計原型>審視流程
>建立生產線
 - 3.產品團隊負責整合流程協同雙方作業

結論：

公司生產應設有獨立的設計部門與工程部門，因應市場上的需求，同時也需要負責整合的部門，讓設計與工程部門能夠更加密切的合作，同時也能獲得雙方部門所欠缺人才的資訊，對於面試求職者時能夠有更好的審查方式，已挑選所需要的人才加入生產行列。

DigitalProductCollaboration

設計部門協同合作時，GITHUB 不是該文章作者認為最好的協同方式，因而舉例以下協同工具，用於實際應用面上。

FIGMA : <https://www.figma.com/>

適合用於設計文件的同一區域多工進行，可以時時觀看所有設計者的改動或者對於同一份工作一起協同進行。

MARVEL : <https://marvelapp.com/>

相對於 FIGMA 的靈活運用，MARVEL 有更標準化的協同流程，使得設計團隊與客戶可以更輕鬆的合作。

ZEPLIN : <https://zeplin.io/>

這是一個有用的傳遞工具，讓設計團隊能更加深入了解工作細節。

總結：

本文強調了協同的重要性，同時舉例初除了 GITHUB 以外的協同工具，利用這些工具可以讓工作能順利的運行，同時提到協同者的培養及其他公司是如何去執行這一塊的過程，最後也提到協同最重要的還是以版本化的資料才能夠使設計團隊達到溝通合作的目的。

MechanicalDesignProcess

CH1：

本章節闡述何謂成功的設計，主導者需要清楚的了解團隊的所有事項、作業系統的流程、協同工具的應用等，同時也須了解市場走向及競爭對手的弱點，並利用此弱點來擊敗對手。

CH2：

本章節講述設計產品時的過程階段，及該階段所需解決的問題。

1.草圖：

盡可能大膽地去設想與設計，以免後續花費更多時間解決問題。

2.圖形化&數字化：

將理想轉換為虛擬圖形，利用製圖工具來完整呈現想法，同時也能確認尺寸及精度的設想是否太刁鑽，進而修改。

3.原形製造&分析：

以上述條件製造出模型，同時分析產品數據，並將所以分析結果記錄下白，且須讓設計團的所有人清楚了解。

4.修正、更改 & 完成設計：所有設計者對於該產品的問題加以討論，讓產品更合乎市場的需求，再重複上面的流程，最終完成該產品的設計。

CH3:

該章節說明兩種設計的基本方式。

- 1.尋找現有的產品，參考其性質並加以製造，該作法快速且問題少，可是創新程度低，容易過於平常，導致市場下降。
- 2.利用簡單的元素進行設計，組合出簡單幾何造型產品，該作法需有基礎的設計團隊及流程，也需參考市場走向及產品完整性，避免過於敷衍的創作，以免得不代市場的青睞。

CH4：

該章節提到材料的挑選及應用。

- 1.零件的材料
- 2.零件的表面處理&尺寸經度
- 3.生產過程是否符合零件需求
- 4.零件&加工成本是否合乎需求
- 5.是否需要二次加工或其他處理流程以上都是在挑選材料及應用會遇到的問題，需要一開始就思考好材料的種類，避免後期因加工或成本導致無法繼續，若材料在某些需求上達不到要求或者合乎要求的材料過於昂貴，可以考慮放寬限制，減少某些要求，才能使產品具有彈性，避免多餘的耗費。

四輪車馬達設定及鍵盤控制加速及轉彎

本資料由 40723221 提供

先從所使用的 3d cad 軟體中將零件繪製好，將各零件轉為 stl 檔，使用 v-rep 中的 import 導入剛剛轉檔好的 stl 零件，轉好後 將零件所對應的相對運動位置依照圖示中的方式排序，馬達也必須設定好(可以從 add > joint 中加入所需要的馬達電機)，其排序的方式會影響到之後設定馬達的運動模擬方式，並且在主體零件上添加指令輸入用的子程式(可以從 add > assoicated 中選擇要加入的子程式選項)

在電機馬達設定好之後可以點左上圖所示的馬達圖標，然後點擊 子視窗下的 show dynamic properties dialog，可以藉由勾選 motor properties 以啟動馬達，並可以藉由調整轉動速度和力矩達到所 想要馬達轉動之效果，底下的 lock motor when target velocity is zero 則可以使馬達速度為零時鎖定住馬達，如果想要馬達處於被動狀態(被動關節)則可以將馬達啟動之選項關閉。

設定好馬達後，設定主體零件和其他附屬零件的動力 學模擬，同樣點選想要設定零件之圖示，並點擊 show dynamic properties dialog，點擊後若想要將該物件設定為可碰撞或者是可以被其他物件響應的狀態則可勾 選 body is respondable，而底下的 body is dynamic 則 可以使該物件在模擬時為動態執行，使在模擬時物件 受到動力學的影響而產生移動，或轉動等變化。

從上述的 add > assoicated 可以得到一個可編輯的子程式，在該子程式中可以利用指令去 使我們設定好的物件透過電機組轉速或者式 角度的控制以使得我們的機構能夠依照我們 想要的方式去做運動，左圖中的程式碼可以 使我們得 4 輪車被鍵盤中的上下左右鍵控制，使其達到加速減速左轉右轉的運動模擬型態(因為空間所以只擷取部分程式碼)。

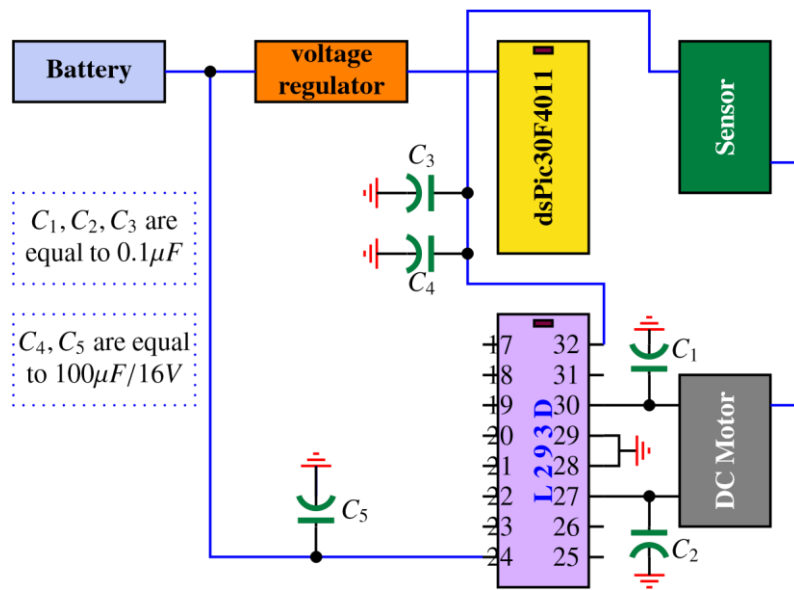
結論:對於動力學中的設定也就是 show dynamic properties dialog 點進去後設定的 body is respondable 和 body is dynamic 都必須要設定完整才可以使四輪車依照所 想要的方式提供動態及物理碰撞，若要使四輪車可以依照所想要的方式運行，在下一部份使用鍵盤控制也不需從設定子程式開始，也就是從(add > associated > non threaded)開始，設定好子程式編輯器之後才可以點開子程式編輯器開始我們對於鍵盤前後左右的程式編輯，我們可以從程式中去選擇我們想要控制的馬達，以這個前後左右的四輪車為例，我們可以透過已經更名為 left_m 和 right_m 的兩顆電機馬達對其做轉速的控制，也就是我們在上面程式的圖式中看到的那一段程式碼我們可以透過 syscall_init 來對我們的馬達做控制。

Assignment 3

通過學術示例來說明其應用,向讀者展示結果如何應用。更具體地說,我們已經看到了如何設計機電一體化系統,我們已經提出了我們必須遵循的二步步驟,以成功設計所需的機電一體化系統。

對於控制 algorithms,我們介紹的大多數例子都是具有完美模型的學術。不幸的是,對於一個實用的系統,我們將有一個實現,可以描述系統在某些特定條件,由於某些原因,這個模型不會完美地工作,如預期的那樣,實時實現演演算法。這可能是由忽略的動態引起的,這些動態可能會改變某些頻率的行為。

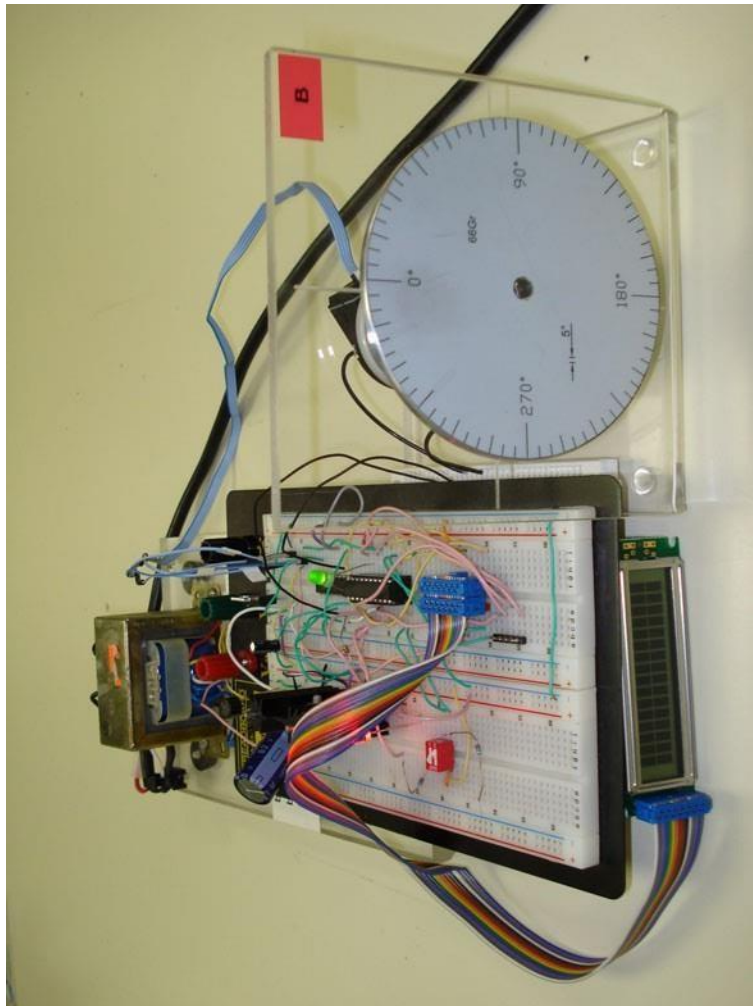
作為第一個例子,讓我們考慮直流電機驅動機械部件的速度控制。該電機的數據表給出了所有重要的參數,因此可以輕鬆獲得該執行器的傳輸功能。我們在此示例中考慮的負載是一個小磁碟,帶有脫離,我們希望在速度上和以後的位置進行控制。



為了識別我們的系統,我們可以使用即時實現設置和適當的 C 程序進行即時操作。由於微控制器擁有有限的記憶體,因此標識可以分為兩個步驟。首先,在第一個實驗中確定增益 K ,然後使用此增益計算可用於計算常量時間 τ 的穩定狀態值。

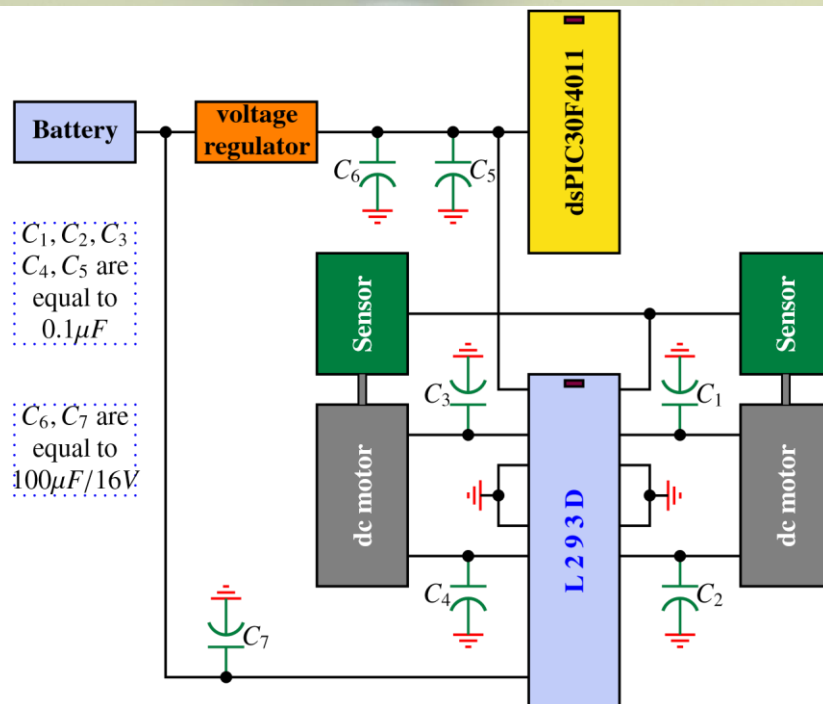
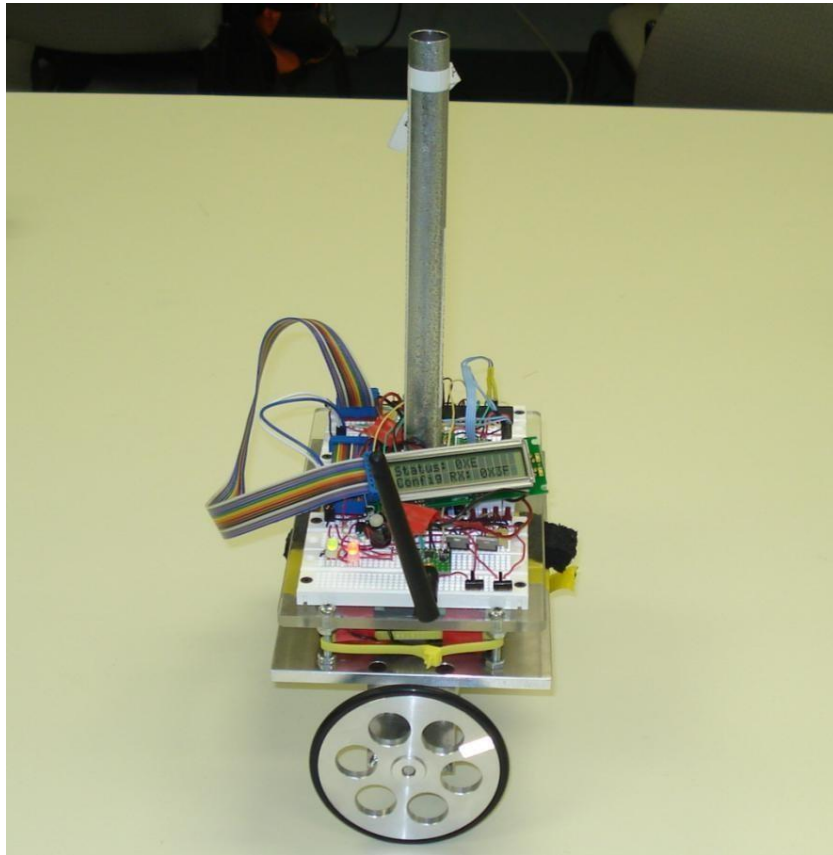
為了設計控制器,我們首先應該指定我們希望我們的系統具有的性能。作為第一個工具我們需要我們的系統是穩定的。還需要系統速度在瞬態機制上有良好的行為,在穩定狀態制度中零錯誤,以便進行步驟參考。對於瞬態,我們希望負載具有小於或等於 5% 的分載時間

$\frac{3\pi}{5}$,而過衝小於或等於 5%。為了完成適當的控制器的設計,我們可以在連續時間進行設計,然後獲得應在軟體部分程式設計的演演演算法,或者直接在離散時間使用所有設計。在本示例的其餘部分中,我們將選擇第二種方法。



從系統傳輸函數的運算式和所需的性能,它重新蘇爾 t ,我們至少需要一個比例和整合器 (PI) 控制器。從控制的角度來看,平衡機器人是一個具有挑戰性的系統,因為它是一個不穩定的開放循環系統。Thi 的系統吸引了很多研究人員,為此提出了許多設計建議。在這裏,我們將介紹在美卡電子實驗室開發和測試的設計,在蒙特爾理工學院。

給出了機器人的概念。它開發的目的是研究,使機電一體化的學生實現他們的控制演演算法,並熟悉複雜的系統。機器人有兩個獨立的車輪,每個車輪由直流電機通過齒輪驅動,比率為 1:6。每個電機都有一個編碼器來測量軸的速度。兩個電機連接到機器人的車身上。其他感測器(如加速度計和陀螺儀)用於測量傾斜角度。引入了適當的濾除器,以消除測量的雜訊,從而獲得有用的控制信號。



(平衡機器人電子電路)

我們可以像對直流電機套件和兩個車輪機器人一樣試驗所有其他控制器,但我們更願意讓這部分作為練習,讓讀者練習這些工具。請注意, 我們邀請他/她在連續時間和離散時間的情況下進行設計,並比較結果。該系統的維度允許這一點。

本章介紹了在蒙特爾理工學院機電一體化實驗室開發的一些案例研究。我們介紹了機電一體化系統設計的所有步驟,其中有一系列細節。重點介紹了每個系統控制演演演算法的 design。