

國立虎尾科技大學
機械設計工程系
專題製作報告

Pyslvs-UI 平面多連桿機構套件 之合成與應用

Synthesis and Application of Pyslvs-UI Planar Multi-link Mechanism Package

指 導 教 授： 李 武 鉦

班 級： 四 設 計 三 甲

學 生： 林昱秀 40723102

林晏瑩 40723103

劉光智 40723145

吳佳穎 40723153

蔡育澧 40723245

中 華 民 國 一 一 零 年 四 月

國立虎尾科技大學 機械設計工程系
學生專題製作合格認可證明

專題製作修習學生：
四設三甲 40723102 林昱秀
四設三甲 40723103 林晏瑩
四設三甲 40723145 劉光智
四設三甲 40723153 吳佳穎
四設三乙 40723245 蔡育澧

專題製作題目： Pyslvs-UI 平面多連桿機構套件之合成與應用

經評量合格，特此證明

評 審 委 員： _____

指 導 老 師： _____
系 主 任： _____

中 華 民 國 一 一 零 年 四 月 二 十 九 日

摘要

生產自動化是現今工業界中最重要的一環，如何以更低的成本與縮短生產製程，來提高在國際上的競爭力，是大家所努力的目標，而身為未來二十一世紀的一員，更需了解其重要性。

關鍵字: 機構模擬、機構分析

目 錄

摘要	i
目錄	i
圖表目錄	iii
第一章 簡介	1
1.1 研究背景與動機	2
1.2 研究目的	2
1.3 研究方法	2
第二章 文獻探討	3
2.1 平面機構分析套件	3
2.1.1 Linkage	3
2.1.2 M.Sketch	3
2.2 平面機構合成套件	5
2.2.1 MeKin2D	5
2.2.2 MechDev	5
2.2.3 WinMecC	6
2.2.4 GIM	6
2.3 自行車避震機構研究	8
2.3.1 Single-pivot suspension(單軸懸架)	8
2.3.2 Linkage-driven Single-pivot Suspension(連桿驅動之單樞軸懸架)	8
2.3.3 High-Pivot Idler Suspension(高樞轉惰輪懸架)	9
2.3.4 Twin-link Suspension(雙連桿懸掛)	9
2.3.5 Horst-link Suspension(霍斯特鏈懸掛)	10
第三章 Pyslvs-UI 套件介紹	12
3.1 Pyslvs-UI 架構與原理	12
3.1.1 平面連桿機構模擬	12
3.1.2 機構合成	12
3.1.3 圖形化使用者介面	14
3.2 Pyslvs-UI 編譯	14
3.2.1 環境與安裝套件	14
3.2.2 編譯步驟	17
3.3 Pyslvs-UI 範例	17
3.3.1 Crank Lifter (曲柄滑塊機構)	17
3.3.2 Crank Rocker(曲柄搖桿機構)	18
3.3.3 Stephenson(史蒂芬生機構)	19
3.3.4 Watt(瓦特氏運動機構)	20
3.3.5 Double Ball Lifter(二段式提球機構)	21
第四章 登山車避震機構	24
4.1 避震機構合成	24
4.2 避震機構評量	25
4.2.1 Anti-squat	25
4.2.2 Anti-rise	25
4.2.3 Leverage Ratio	25

4.3 避震機構分析範例	25
第五章 結論	27
第六章 未來研究建議	28
參考文獻	29
誌謝	30
作者簡介	31

圖 表 目 錄

圖	2.1	Linkage 工作介面	3
圖	2.2	M.Sketch 工作介面	4
圖	2.3	平面連桿之運動學模擬	5
圖	2.4	盤式凸輪機構之合成與分析	5
圖	2.5	WinMecC	6
圖	2.6	GIM 合成介面	7
圖	2.7	單軸懸架	8
圖	2.8	連桿驅動之單樞軸懸架	8
圖	2.9	高樞轉惰輪懸架	9
圖	2.10	雙連桿懸掛	10
圖	2.11	霍斯特鏈懸掛	10
圖	2.12	霍斯特鏈懸掛	11
圖	3.1	實數編碼遺傳算法流程圖	13
圖	3.2	螢火蟲算法流程圖	13
圖	3.3	差分進化演算法流程圖	13
圖	3.4	曲柄滑塊機構運用範圍	18
圖	3.5	曲柄滑塊機構運用範圍	18
圖	3.6	曲柄搖桿機構	19
圖	3.7	曲柄搖桿機構範例	19
圖	3.8	史蒂芬生機構動作動	20
圖	3.9	史蒂芬生機構範例	20
圖	3.10	瓦特氏運動機構	21
圖	3.11	瓦特氏運動機構飛機起落收放	21
圖	3.12	史蒂芬生機構第一型	22
圖	3.13	雙球升降器	22
圖	3.14	搖臂式發球機	23
圖	4.1	Horst-link suspension	26
圖	4.2	Twin-link suspension	26
圖	6.1	後續研究	28

第一章 簡介

生產自動化是現今工業界中最重要的一環，如何以更低成本與縮短生產製程，來提高在國際上的競爭力，是大家所努力的目標，而身為未來二十世紀的數位控制加工機（CNC），其成本昂貴，且體積龐大，目前於市面決定運用在校所學之相關課程，以完成一部具有高精度、體積小故本組成（十萬元以下）的 PC-Based 三軸運動控制實驗台為研究目標，達成在教學上的需要，並增加系內的實驗設備。

本專題主要目的在了解伺服馬達的控制原理、三軸運動控制卡的使用方式、人機介面程式的撰寫與增加機械實務加工的能力。

第 1.1 節 研究背景與動機

隨著科技越來越發達網路上有許多機構分析的套件，但是鮮少有機構合成功能的套件，且大多為收費或無公開原始碼。而本專題所使用的套件-Pyslvs-UI 為本研究室所開發的套件且有公開原始碼，因此此專題利用該套件來合成一登山車避震機構。

第 1.2 節 研究目的

第 1.3 節 研究方法

首先建立一現有登山車避震機構，再針對不影響現有自行車車架的尺寸但對於避震能力有較大影響的接頭利用遺傳演算法求解最佳位置以得到最佳連桿尺寸，接著將上述所得到的接頭位置和連桿長度利用 Inventor 自動產生對應連桿尺寸，後續利用 xml 格式匯入 CoppeliaSim 並利用 CoppeliaSim API 設定馬達讓該機構以一自由度進行運動模擬。上述流程如圖 XXXX 所示。

第二章 文獻探討

第 2.1 節 平面機構分析套件

2.1.1 Linkage

Linkage 為電腦輔助設計的應用程式，用於機械鏈接的原型製作。基本功能為設計帶有樞軸連接器或滑動連接器之機構、驅動機構的輸入可以是旋轉或線性的、可在運動的連桿上安裝齒輪和鏈條機構、將繪圖功能分配給連接點顯示其運動之軌跡。

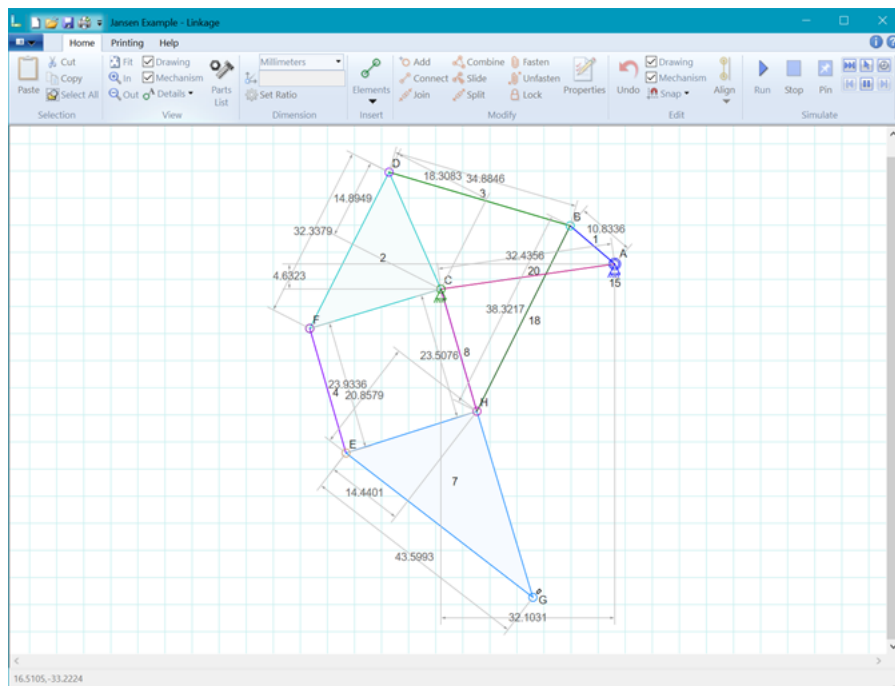


圖 2.1: Linkage 工作介面

2.1.2 M.Sketch

M.Sketch，一種原型設計工具，設計目的在為非專業人士提供簡單直觀的界面，輕鬆學習與使用幾何圖形的模擬，如圖 2.2。其中有五個主要功能：

1. 機構草圖: 軟體介面中，藉由單擊與拖動滑鼠來簡單的繪製連桿接點，也可將接點轉換為滑塊，以此快速設計連桿機構。
2. 運動可視化: 根據馬達的旋轉值可看到動畫上每個鏈接的坐標變化、設定欲關注之端點並給予標記，則可看到該端點預期的運動路徑。
3. 自上而下的設計: 根據繪製所需之運動，生成機構路徑、優化當前配置所適合的路徑。

4. 加工支持: 可導出 PDF 文件中每個鏈接的組合圖、示意圖和零件圖，亦可導出鏈接和關聯零件的 3D 模型，以匯入 3D 列印機。
5. 高可及性: M.Sketch 是作為 Web 的應用程序開發的，用戶可以在 Web 瀏覽器（例如 GoogleChrome）上申請帳號後，即直接可使用，無需在電腦上進行任何安裝。

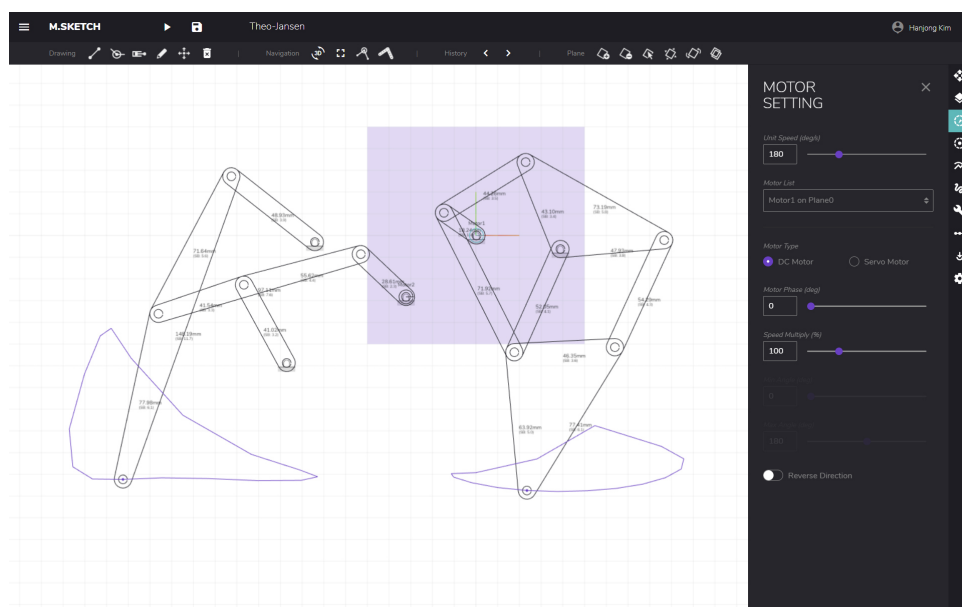


圖 2.2: M.Sketch 工作介面

第 2.2 節 平面機構合成套件

2.2.1 MeKin2D

MeKin2D000 是用 Free Pascal 編寫的子程序包，主要使用模塊化方法對平面連桿進行運動學模擬，用於磁盤凸輪機構的合成和分析，以及漸開線齒輪的產生。其中，MeKin2D 的套件包含 4 個子程式。分別有：

1. LibMec2D: 在模擬中添加了線性和角度測量，作為向量的速度和加速度以及運動點的軌跡。
2. LibMecIn: 定義基本輸入，例如曲柄或滑塊。
3. LibAssur: 對被動模塊進行建模。
4. LibCams: 用於運動學合成以及盤式凸輪的分析。

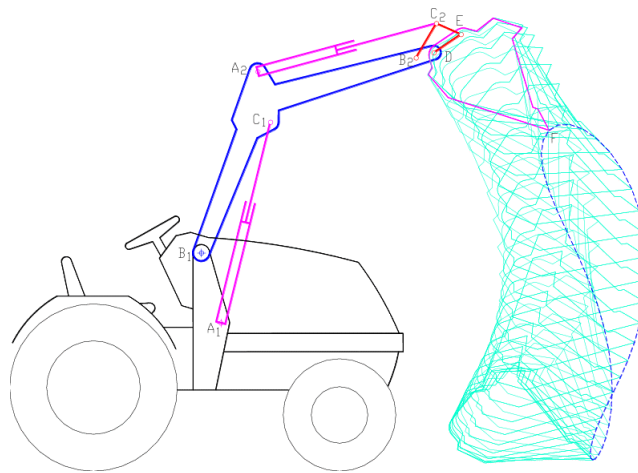


圖 2.3: 平面連桿之運動學模擬

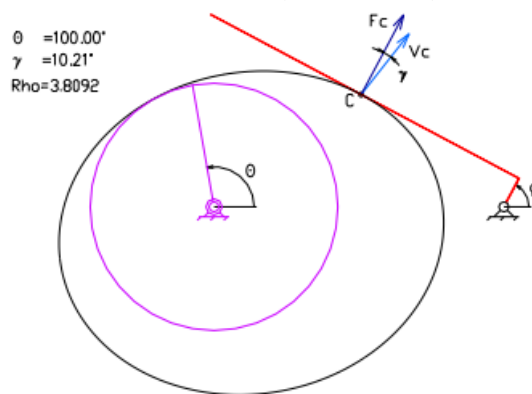


圖 2.4: 盤式凸輪機構之合成與分析

2.2.2 MechDev

MechDev 是一種機構設計軟件，專注於可用性和機構設計功能。基本功能為分析機構運動學、分析運動靜力學、合成與分析帶有滾輪從動件或平面從動件的凸輪機構。

2.2.3 WinMecC

WinMecC 為一種電腦軟件，主要用於一自由度平面機構的分析與合成。基本功能為將機構的運動學和動力學分析之結果數值化或圖視化、可以優化分析創建的機構，依據特定點所期望遵循之路徑進行機構合成，如圖 2.4 所示。

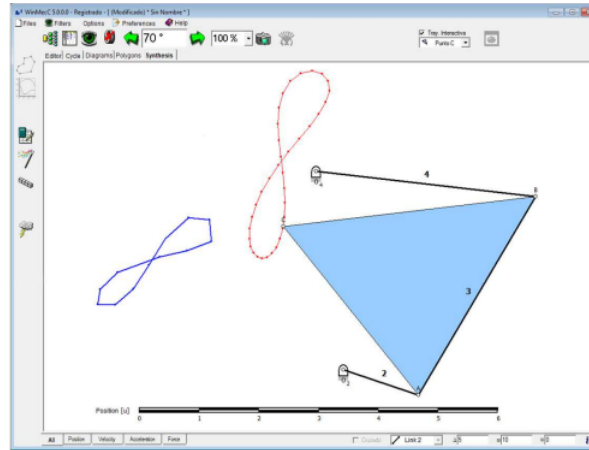


圖 2.5: WinMecC

2.2.4 GIM

GIM 是 COMPMECH Research Group 創建的註冊免費軟件，用於平面機構的運動模擬合成以及機械結構的靜態分析。GIM 軟件具有三個主要模塊：幾何、運動、合成。

1. 幾何模塊: 設計要分析的機構，建立其節點與元件，定義連接點的類型（旋轉接頭或滑塊，並設定固定點。
2. 運動模塊: 透過運動學分析與模擬，在運動點繪製路徑以及路徑的曲率中心，可在畫面上看到運動極點的位置、速度和加速度向量，並繪製與分析數據相關的圖形和表格。
3. 合成模塊: 依據特定點所期望之運動路徑，進行機構軌跡合成。如圖 2.5 所示。

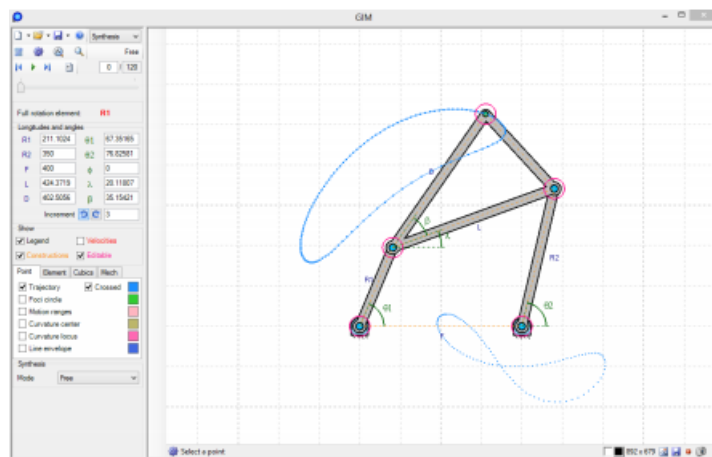


圖 2.6: GIM 合成介面

第 2.3 節 自行車避震機構研究

2.3.1 Single-pivot suspension(單軸懸架)

最簡單的懸架設計為單樞軸，使用搖臂連接後軸、主樞軸、避震器，特點在於後軸直接連接至主樞軸，避震器連接到搖臂上，搖桿環繞主樞軸中心旋轉，槓桿比由避震器的位置決定，固定中心可在行駛過程預測懸架特性，但也意味著不同階段修改懸架特性能力有限，示意圖產品號為 Orange Stage 的 STAGE 6 FACTORY，如圖 (2.1)。



圖 2.7: 單軸懸架

2.3.2 Linkage-driven Single-pivot Suspension(連桿驅動之單樞軸懸架)

如同單樞軸設計藉由後軸通過搖臂連接到主樞軸，增加聯動裝置在避震與搖臂間允許調整槓桿比率曲線，並在搖臂與避震器間建立連桿以改變槓桿比，搖臂圍繞主軸旋轉且將中心固定於整個行程中，示意圖產品車架為 Orange Stage 的 Orange 2020 Alpine 6 Frame 27.5，如圖 (2.2)。



圖 2.8: 連桿驅動之單樞軸懸架

2.3.3 High-Pivot Idler Suspension(高樞轉惰輪懸架)

通常建構在單樞軸或鏈條驅動的單樞軸設計上，其樞軸位置位於鏈條上方，為使控制鏈條增長，鏈輪惰輪將鏈條從鏈輪佈置到主樞軸的頂部或非常靠近主樞軸的位置，主樞軸定位在鏈輪上方更高的位置，使用單樞軸同時中心在整個行程中是固定的。但是，在撞擊過程中，後搖臂在主樞軸下方以曲線向上和向後旋轉，而不是在低樞軸位置向上和向後旋轉，且後軸路徑可以幫助懸架在方形撞擊中保持平穩。鏈條繞到惰輪上，使鏈條與樞軸和後軸成一直線，大大減少了鏈增長的影響。惰輪的位置可以由設計人員用來增加或減少防下蹲的水平，但不會改變通常較高的防起落高度，如圖 (2.3)，產品來源為 Forbidden Druid 的 Druid Frame -2021。



圖 2.9: 高樞轉惰輪懸架

2.3.4 Twin-link Suspension(雙連桿懸掛)

搖臂通過兩個搖桿連桿安裝在框架上，從而將搖臂連接到主機樞軸，通過擺臂或搖臂連桿之一來驅動衝擊。雙連桿懸架將樞軸的數量從一增加到四個，可修改中心位置，進而改變行程不同點的懸架特性，參考產品為 Ibis' DW-link Suspension 的 Mojo HD5 Frame -2020，如圖 (2.4)。



圖 2.10: 雙連桿懸掛

2.3.5 Horst-link Suspension(霍斯特鏈懸掛)

特性在於樞軸底端的鏈條，後軸則安裝在腳撐上增加了樞軸，其修改後的軸距繞即時中心旋轉，改變行進路線的位置，可在行進的不同階段優化防下蹲與上抬。

雙連桿懸掛相比，霍斯特連桿系統的較長連桿通常會提供更平滑的曲線，產品支架為 RAAW Madonna 的 MADONNA V1 - FAST AND PREDICTABLE，如圖 (2.5)，產品為 Canyon Spectral 的 Izzo Pro Race 29，如圖 (2.6)。



圖 2.11: 霍斯特鏈懸掛



圖 2.12: 霍斯特鏈懸掛

第三章 Pyslvs-UI 套件介紹

第 3.1 節 Pyslvs-UI 架構與原理

Pyslvs-UI 是一套利用 Python3 與 PyQt5 建立的平面機構模擬與合成系統。機構模擬與合成的主要核心包括 Python-Solvespace 幾何約束求解程式庫、三角幾何函式程式庫 (tinycadlib)、演算程式庫 (ADesign)、幾何約束求解程式庫 (bgfs)、類型合成程式庫 (topologic)、數目合成程式庫 (number) 等。其中，ADesign 演算程式庫內包含 Real-coded Genetic Algorithm (RGA)、Differential Evolution(DE) 與 Firefly Algorithm (Firefly) 等三種，用於平面機構尺寸合成演算

3.1.1 平面連桿機構模擬

- Python 的 Solvespace：Solvespace 的核心是與 Cython 綁定在一起。
- Pyslvs：使用 Cython 解決 SketchSolve 的核心，其中包括平面機構的創新設計方法。SketchSolve 是一個可用於解決 CAD 軟件中發現的幾何約束問題的項目。Cython 是針對 python 編程語言和擴展的 Cython 編程語言（基於 Pyrex）的優化靜態編譯器。

3.1.2 機構合成

- 尺寸合成 通過隨機變量生成具有路徑限制的機構，生成結構參數來自變數設定，也有其它算法選擇，程式內共包含三種算法：實數編碼遺傳算法、螢火蟲算法與差分進化演算法。實數編碼遺傳算法：在遺傳演算法的演算過程中，將設計變數以實數數值表示，稱為實數編碼遺傳演算法 (RGA)，其演算流程如圖 3.3(1)。
- 數目合成 螢火蟲算法：其演算法是模擬螢火蟲離散的閃爍行為，算法將計算群體中螢火蟲的相對亮度和吸引度，並根據相對亮度決定螢火蟲的移動方向；更新螢火蟲的空間位置，對處在最佳位置的螢火蟲進行隨機移動；根據更新後螢火蟲位置，重新計算螢火蟲的亮度，最終輸出群體極值點和最優個體數值，演算流程如圖 3.3(2)。
- 構造合成 差分進化演算法：是一種求解最佳化問題的進化演算法，其算法原理採用對個體進行方向擾動，以達到對個體的函式值進行下降的目的，同其他進化演算法一樣，差分進化演算法不利用函式的梯度資訊，因此對函式的可導性甚至連續性沒有要求，適用性很強，演算流程如圖 3.3(3)。

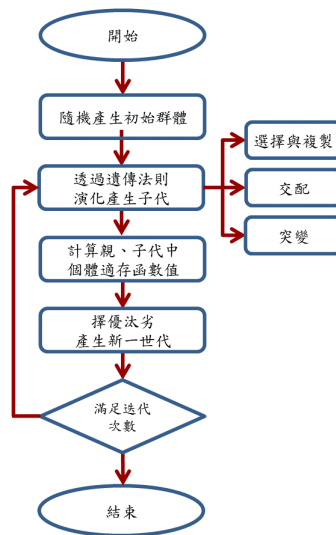


圖 3.1: 實數編碼遺傳算法流程圖

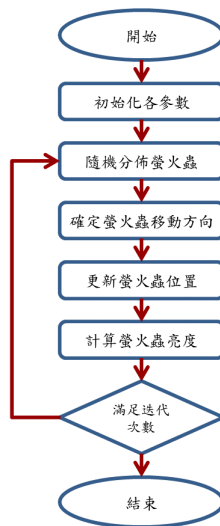


圖 3.2: 螢火蟲算法流程圖

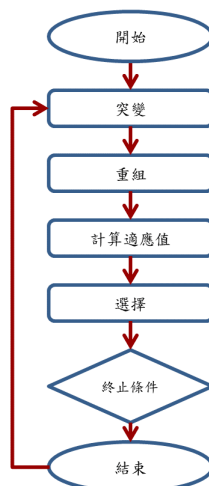


圖 3.3: 差分進化演算法流程圖

3.1.3 圖形化使用者介面

- Qt(c++)

Qt 是一個 C++ 應用程式跨平台開發框架，被廣泛應用於開發圖形化介面程式。主要特色為以 Qt 開發的程式軟體，不須修改原始碼，使用相同的程式碼皆可在支援的平台上執行與編譯，自動依照平台的差異，展示該平台自有的圖形介面風格

第 3.2 節 Pyslvs-UI 編譯

3.2.1 環境與安裝套件

- GIT: 利用以下兩種系統進行遠端檔案版本控制的工具。
Ubuntu:

Listing 3.1: 使用 apt 安裝 Git

```
$sudo apt install git
```

Windows:

Listing 3.2: 下載新版 Git

```
https://git-scm.com/download/win
```

- Cython: 為了增進運算速度將 Python 語言轉換成 C++ 的套件，並使用 pip 安裝 Cython。

Listing 3.3: 安裝 Cython

```
$sudo pip3 install cython  
>pip install cython
```

- Windows 編譯工具: 使用 Windows 中的 GNU 工具包以操作 Make 功能。

Listing 3.4: 下載新版 MinGW(32Bit)

```
https://sourceforge.net/projects/mingw-w64/files/latest/download?source=files
```

另外必須安裝 Microsoft Visual Studio 以編譯 64 位元的 Cython 程式碼，且安裝 Windows tool kit。

Listing 3.5: 安裝各類編譯程式

```
https://www.visualstudio.com/downloads/
```

- SWIG: 將 C 或 C++ 配上自訂接口後，可變成其他高階語言函式庫的套件。
Ubuntu:

Listing 3.6: 使用 apt 安裝

```
$sudo apt install swig
```

Windows:

Listing 3.7: 下載新版 SWIG

```
http://www.swig.org/download.html
```

解壓縮到安裝位置，並將該資料夾加入環境變數。

Listing 3.8: 安裝位置

```
C:\swigwin-3.0.12
```

- PyQt: 於官方網站上下載 Qt 安裝檔，並選擇 PyQt 有所對應的新版本，以免 API 對應不上。

Listing 3.9: 下載 Qt 檔

```
https://www.qt.io/download/
```

Ubuntu:

Listing 3.10: 安裝後將 Qt 函式庫位址加入 /.bashrc 環境變數

```
export QTDIR=/opt/Qt/5.8/gcc_64/  
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${QTDIR}/lib/  
export PATH=${QTDIR}/bin:${PATH}
```

Windows: 安裝後將 Qt 函式庫位址加入環境變數，再另外安裝 PyQt

Listing 3.11: Qt 函式庫位址加入環境變數

```
C:\Qt\5.8\msvc2015_64\bin。
```

- Ubuntu: 由於某部分無法使用 pip，需先確認並刪除 Python 套件安裝位置的 SIP 檔案

Listing 3.12: 刪除 SIP 檔案

```
$sudo rm -f /usr/lib/python3/dist-packages/sip*
```

Listing 3.13: 下載新版 SIP

```
https://riverbankcomputing.com/software/sip/download
```

Listing 3.14: 解壓縮後安裝

```
$python3 configure.py  
$sudo make install
```

Listing 3.15: 核對 Python 內的版本

```
$sip -V  
>>import sip  
>>print(sip.SIP_VERSION_STR)
```

Listing 3.16: 下載新版 PyQt5

```
https://riverbankcomputing.com/software/pyqt/download5
```

Listing 3.17: 解壓縮後安裝

```
$python3 configure.py  
$sudo make install
```

Listing 3.18: 下載新版的 QScintilla 2

```
https://riverbankcomputing.com/software/qscintilla/download
```

Listing 3.19: 本體

```
$cd Qt4Qt5  
$qmake qscintilla.pro  
$sudo make install
```

Listing 3.20: Designer

```
$cd designer-Qt4Qt5  
$qmake designer.pro  
$sudo make install
```

Listing 3.21: Python bundings

```
$cd Python  
$python3 configure.py --pyqt=PyQt5  
$sudo make install
```

- Windows:

Listing 3.22: 下載 Qt 檔

```
https://www.qt.io/download/
```

Ubuntu:

Listing 3.23: 使用 pip 安裝 PyQt5

```
>pip install pyqt5 -U
```

Listing 3.24: 使用 pip 安裝 QScintilla 2

```
>pip install qscintilla
```

- Python 編譯工具: 將 Python 程式碼編譯成可執行檔的工具。
Ubuntu: 使用 PyInstaller 即可。

Listing 3.25: 安裝 PyInstaller

```
$sudo pip3 install PyInstaller
```

Windows: 使用 PyInstaller 編譯主體，再用 cxFreeze 包裝函式庫。

Listing 3.26: 安裝 cxFreeze

```
>pip install cx_Freeze PyInstaller
```

如果使用官方來源有編碼錯誤，可以用下面的倉儲：

Listing 3.27: 下載倉儲

```
>pip install  
https://github.com/pyinstaller/pyinstaller/archive/develop.zip
```

- Eric IDE: 若要進程式協同，需安裝此 Python 編輯器。

Listing 3.28: 下載 Eric IDE

```
http://eric-ide.python-projects.org/eric-download.html
```

Listing 3.29: 執行 install.py

```
$sudo python3 install.py  
>python install.py
```

3.2.2 編譯步驟

- 下載 Pyslvs 倉儲，並進行子倉儲更新

Listing 3.30: 下載倉儲並更新

```
git clone https://github.com/KmolYuan/Pyslvs-PyQt5.git  
git submodule init  
git submodule update
```

Listing 3.31: 安裝 Python 套件

```
pip install -r requirements.txt
```

編譯好的函式庫複製到 core/kernel/對應的資料夾下。

Listing 3.32: 複製函式庫

Ubuntu的Python 3.5 命名為py35。
Windows的Python 3.6 命名為py36w。

並在 core/calculation/calculation.py 中匯入名稱。

Listing 3.33: 匯入名稱

```
if py_nm=='36w':  
    from ..kernel.py36w.slvs import *  
    from ..kernel.pyslvs_generate.py36w import tinycadlib  
    from ..kernel.pyslvs_generate.py36w.planarlinkage import build_planar  
    from ..kernel.pyslvs_generate.py36w.rga import Genetic  
    from ..kernel.pyslvs_generate.py36w.firefly import Firefly  
    from ..kernel.pyslvs_generate.py36w.de import DiffertialEvolution
```

第 3.3 節 Pyslvs-UI 範例

3.3.1 Crank Lifter (曲柄滑塊機構)

主要以 6 個連桿、7 個接頭，4 個雙接頭連桿、1 個三接頭連桿所構成，如圖 (3.1)，由四連桿機構衍伸而來多樣性運動型態、軌跡，可滿足一般運動需求，常用於汽車工業中加速與製動，從踏板的輸入到化油器、製動缸、自行車剎車都廣泛運用在其中，如圖 (3.2)。

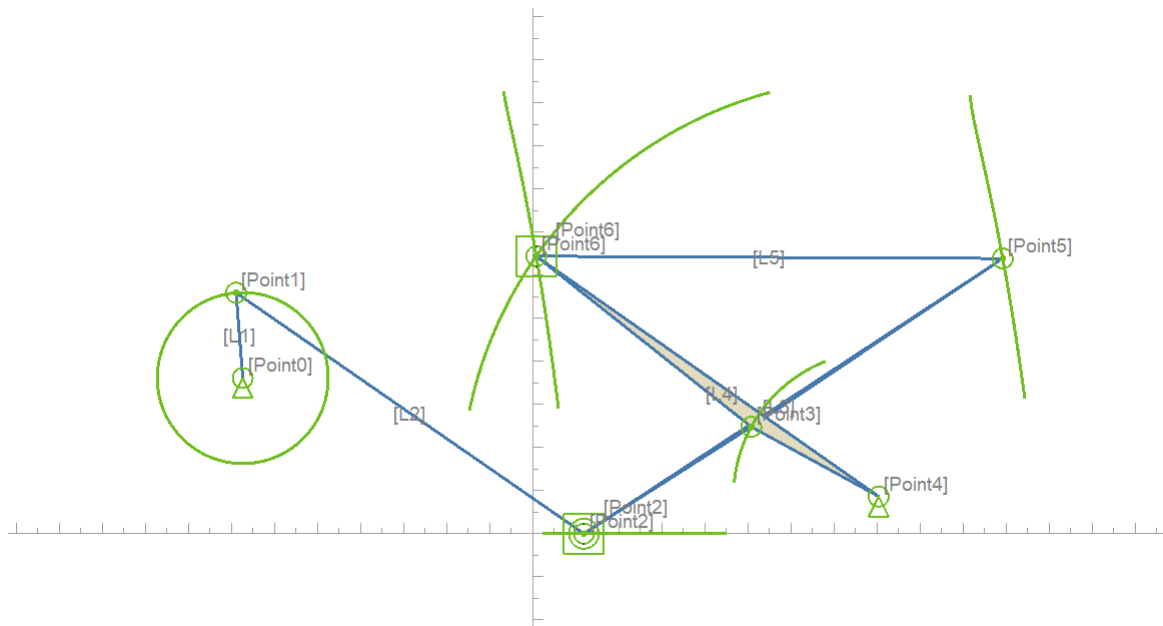


圖 3.4: 曲柄滑塊機構



圖 3.5: 曲柄滑塊機構運用範圍

3.3.2 Crank Rocker(曲柄搖桿機構)

為 4 個連桿、5 個接頭，2 個雙接頭連桿、1 個三接頭連桿所組成，如圖 (5.3)，曲柄旋轉運動透過連接桿轉換至搖桿的搖擺運動，此四連桿機構構造最簡單也最常在生活周遭出現，如：老虎鉗、雷達調整機構、縫紉機腳踏機構、鋼材輸送機，如圖 (5.4)。

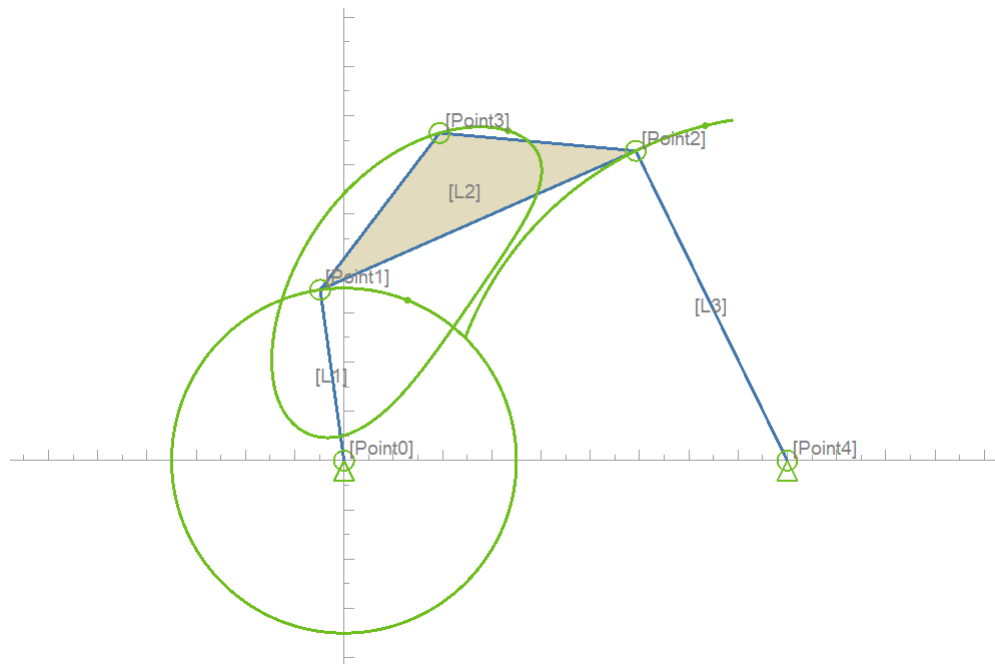


圖 3.6: 曲柄搖桿機構

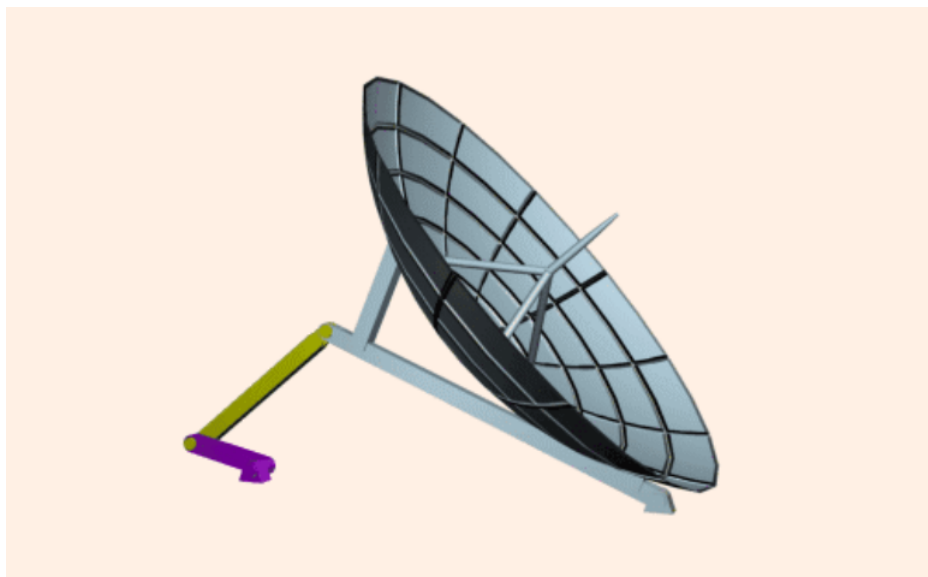


圖 3.7: 曲柄搖桿機構範例

3.3.3 Stephenson(史蒂芬生機構)

為 6 個連桿、7 個接頭，3 個雙接頭連桿、2 個三接頭連桿所組成，使三接頭連桿作為固定桿，其中一雙接頭連桿作曲柄運動，其餘桿件與三接頭連桿一起做搖擺運動，如圖 (3.5)，此機構僅次於四連桿組的重要單自由度連桿機構也稱作複合連桿組機構，常用於自動化夾緊機構、越野摩托車後懸吊系統，如圖 (3.6)。

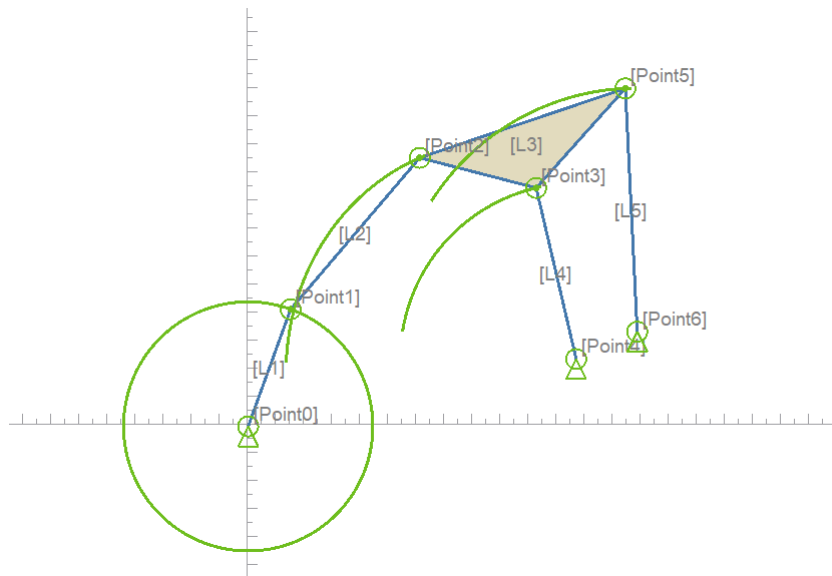


圖 3.8: 史蒂芬生機構作動

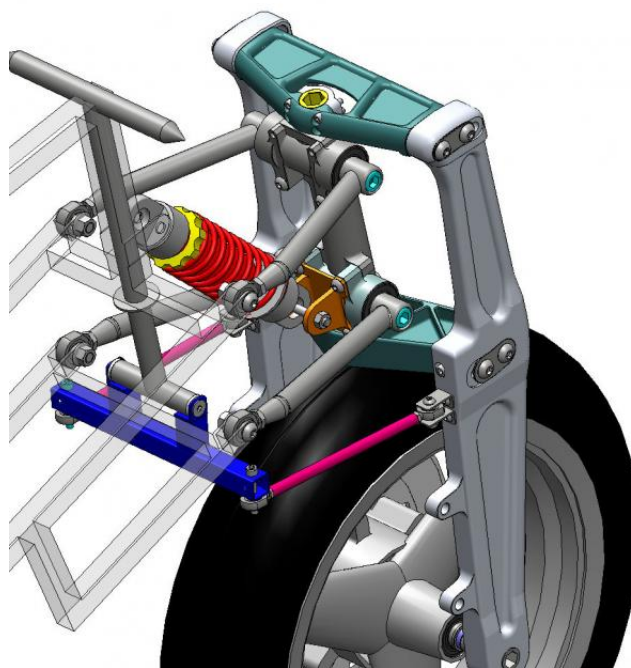


圖 3.9: 史蒂芬生機構範例

3.3.4 Watt(瓦特氏運動機構)

由 6 個連桿、7 個接頭，4 個雙接頭連桿、1 個三接頭連桿構成，本身機械鏈構型為上下左右對稱，使三接頭桿為固定桿另其中一支雙接頭桿為曲柄作旋轉運動，其他桿件則是搖擺運動，如圖 (3.7)，常用於自動化平台舉高移送機構、自動化包裹移送機構、磨耗測試裝置、飛機起落收放機構，如圖 (3.8)。

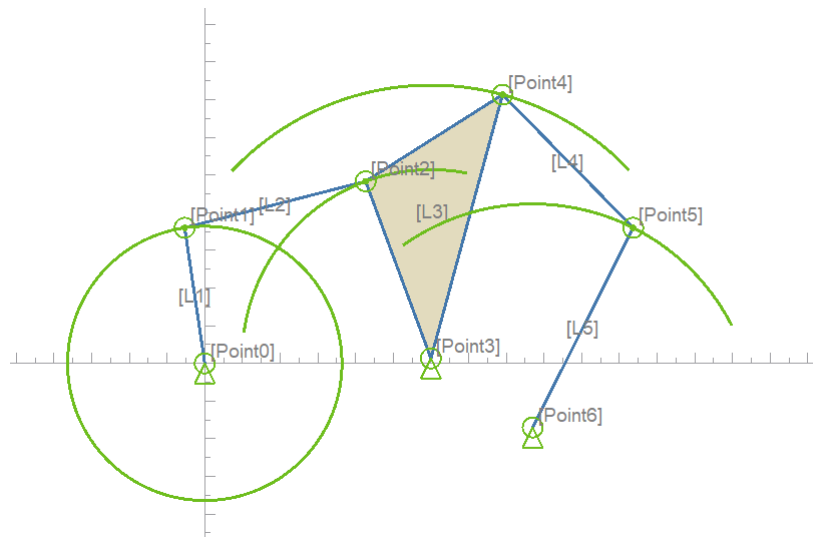


圖 3.10: 瓦特氏運動機構

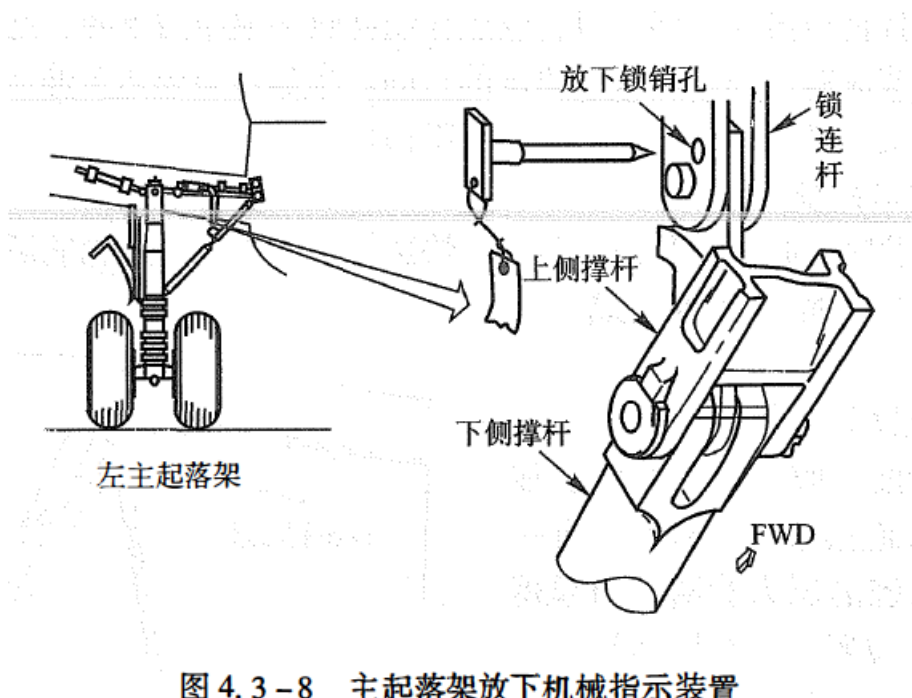


图 4.3 - 8 主起落架放下机械指示装置

圖 3.11: 瓦特氏運動機構飛機起落收放

3.3.5 Double Ball Lifter(二段式提球機構)

由兩個史蒂芬生 (Stephenson) 機構衍伸出來，如圖 (3.9)，將機構融合調整位置後為 10 個連桿、14 個接頭，8 個雙接頭連桿、1 個三接頭連桿所組成，如圖 (3.10)，便於機構執行時接近置球點底部，將球提起並筆直向上提升與下一個提球機構交會完成傳接行程，常運用在搖臂式發球機上，如圖 (3.11)。

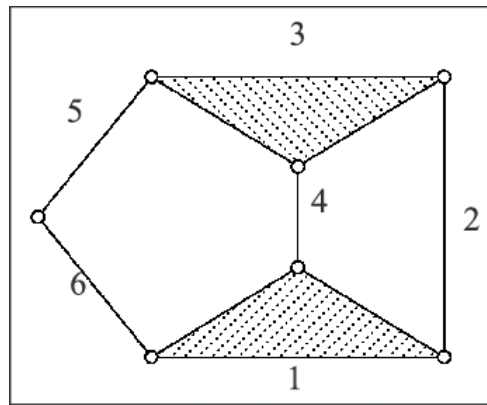


圖 3.12: 史蒂芬生機構第一型

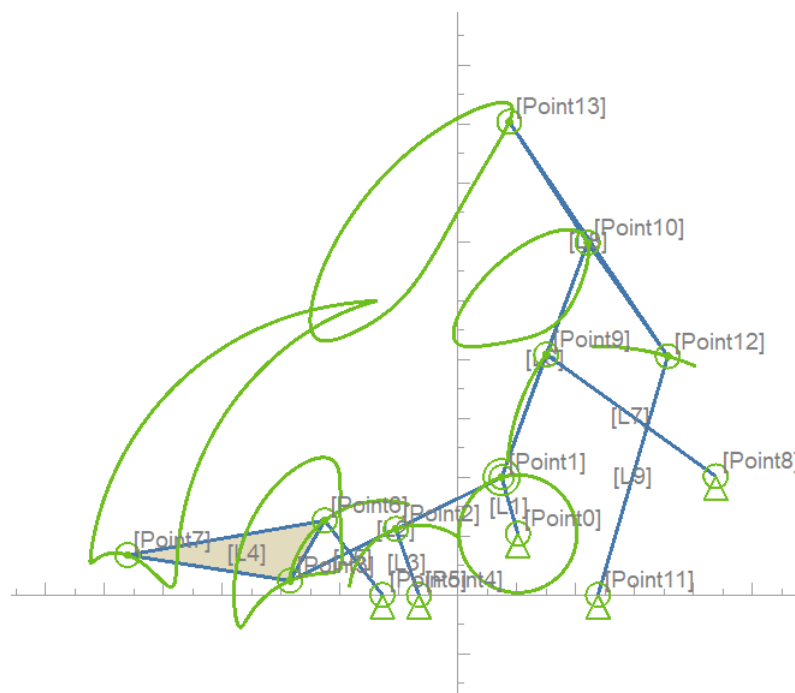


圖 3.13: 雙球升降器



圖 3.14: 搖臂式發球機

第四章 登山車避震機構

第 4.1 節 避震機構合成

機構合成設計需求為騎乘姿勢改變時或遇到顛頗不整的路時仍能保持整體的平衡，即 Anti-squat 及 Anti-rise 的數值能在 100% 附近，而上述合成需求之適應函數可表示為

$$h_2 = d_{of} \left(\frac{r_{ppy} - IFC_y}{r_{ppx} - IFC_x} \right) + \left[r_{ppy} - r_{ppx} \left(\frac{r_{ppy} - IFC_y}{r_{ppx} - IFC_x} \right) \right] \quad (4.1)$$

其中

$$IFC_x = \frac{\left[P_{2y} - \left(\frac{P_{2y} - IC_y}{P_{2x} - IC_x} \right) P_{2x} - otl_b \right]}{\frac{P_{2y} - IC_y}{P_{2x} - IC_x} - otl_a} \quad (4.2)$$

$$IFC_y = otl_a \cdot (IFC_x) + otl_b \quad (4.3)$$

$$IC_x = \frac{P_{3y} - \left[\frac{(P_{3y} - P_{4y})P_{3x}}{P_{3x} - P_{4x}} \right]}{\frac{P_{1y}}{P_{1x}} - \left(\frac{P_{3y} - P_{4y}}{P_{3x} - P_{4x}} \right)} \quad (4.4)$$

$$IC_y = \left(\frac{P_{1y}}{P_{1x}} \right) IC_x \quad (4.5)$$

r_{ppx} : 後輪與地面接觸點到座標原點的 x 座標

r_{ppy} : 後輪與地面接觸點到座標原點的 y 座標

P_{nx} : 點 n 的 x 座標

P_{ny} : 點 n 的 y 座標

第 4.2 節 避震機構評量

4.2.1 Anti-squat

當加速時因不同騎乘姿勢導致的重心位置變化而產生的力量轉移會讓懸架受壓縮力，而 Anti-squat 即為抵抗此後沉的能力。

- $> 100\%$: 表示足以抵抗後沉的力量，而超過的力量將會變成拉伸力
- 100% : 表示完全平衡了負載，即懸吊沒受到壓縮力或拉伸力
- $0 \sim 100\%$: 表示部分後沉力量被抵抗
- $< 0\%$: 表示所有的後沉力量皆無被抵抗

4.2.2 Anti-rise

在剎車的時候因為力量轉移的關係導致後輪產生升起的情況，而 Anti-rise 即為抵抗此後升現象的能力。

- $> 100\%$: 表示足以抵抗後升的力量，而超過的力量將會變成壓縮力
- 100% : 表示完全抵抗了剎車產生的後升現象
- $0 \sim 100\%$: 表示部分後升力量被抵抗
- $< 0\%$: 表示所有的後升力量皆無被抵抗

4.2.3 Leverage Ratio

Leverage ratio 是避震器壓縮量和後輪行程的比值，當較大的槓桿比率會對避震器產生較大的衝擊，而在騎乘時對於地形變化的感受度較不敏感，相反地較小的槓桿比率對於地形的敏感度較高。其計算方式如下所示。

$$\text{Leverage Ratio} = \frac{\text{wheel travel}}{\text{stroke}} \quad (4.6)$$

第 4.3 節 避震機構分析範例

此小節會將章節一所提到的自行車避震機構類型為範例，說明在不同避震機構下經由尺寸合成後得到的評比結果。此評比結果會以自行車後輪的垂直上升行程作為變數來探討 Anti-squat 以及 Anti-rise 的表現。

圖 4.1 為 Horst-link 類型並以 Niner RDO 29 作為該類型自行車範例在後輪行程 0 到 150mm 的 Anti-squat 及 Anti-rise 的表現

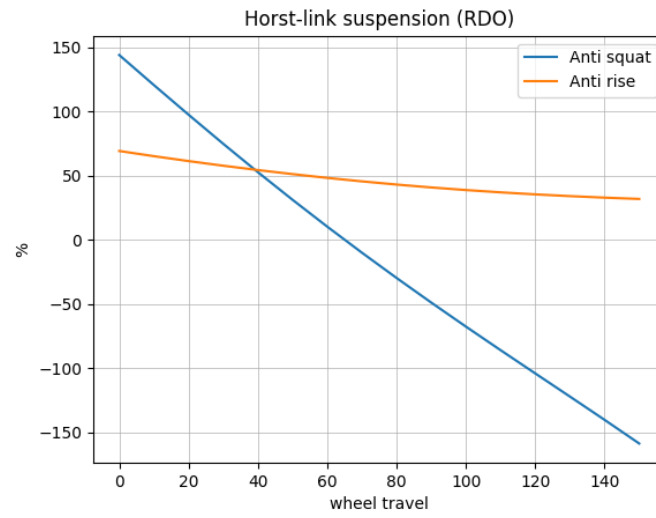


圖 4.1: Horst-link suspension

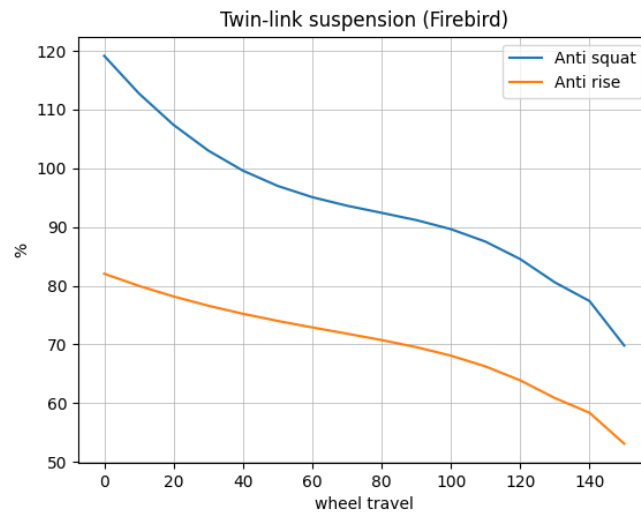


圖 4.2: Twin-link suspension

第五章 結論

test 內文內文內文 123ABC

第六章 未來研究建議

本專題已建立一流程從尺寸合成至動態模擬軟體進行運動模擬，後續可將運動系統改為動力系統並利用有限元素法對該避震機構進行受力情況的分析。

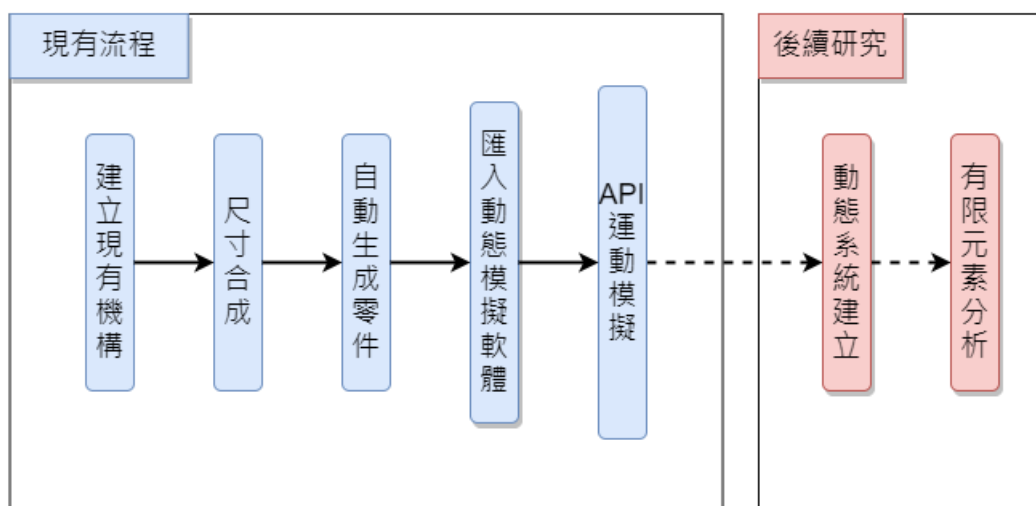


圖 6.1: 後續研究

參考文獻

生產自動化是現今工業界中最重要的一環，如何以更低的成本與縮短生產製程，來提高在國際上的競爭力，是大家所努力的目標，而身為未來二十一世紀的一員，更需了解其重要性。

誌謝

生產自動化是現今工業界中最重要的一環，如何以更低的成本與縮短生產製程，來提高在國際上的競爭力，是大家所努力的目標，而身為未來二十一世紀的一員，更需了解其重要性。

作者簡介

生產自動化是現今工業界中最重要的一環，如何以更低的成本與縮短生產製程，來提高在國際上的競爭力，是大家所努力的目標，而身為未來二十一世紀的一員，更需了解其重要性。