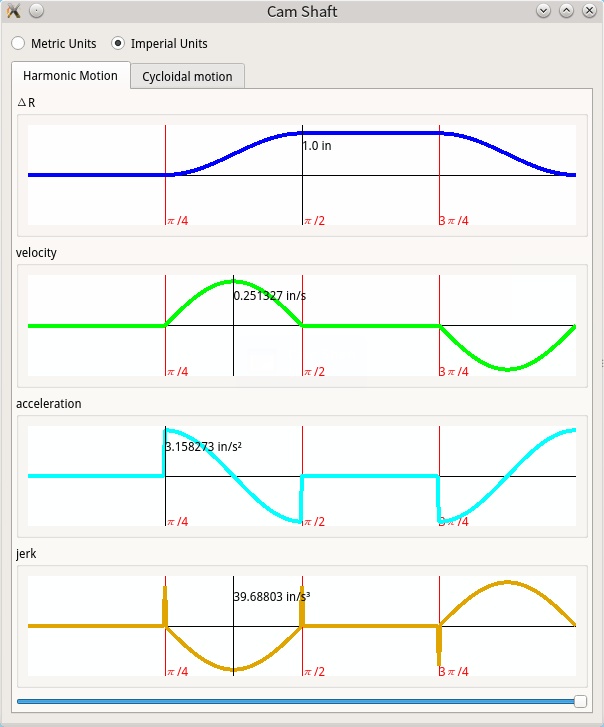
機構學報告

班級：四設計三乙

學號：40323230

姓名：張元



程式倉儲：<https://github.com/40323230/camCal>

繪製工具：Python 3.4 PyQt 5.7 QPainter

\*由於編譯環境在Linux作業系統，無法在Windows平台運行。

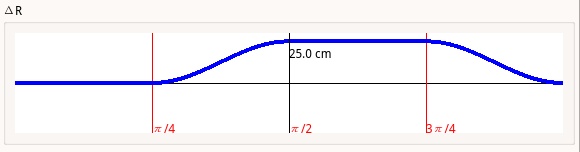
\*若要在Windows平台編譯，必須安裝相關工具（Python3、PyQt、PyInstaller），下載原始碼後製作，硬碟暫無空間所以不方便做出Windows平台版本。

使用單位：英制公制可切換

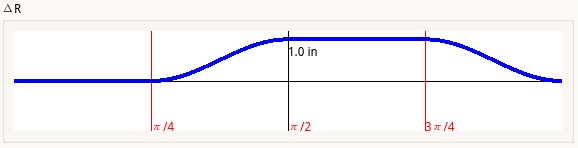
簡諧運動

ΔR公式：

單位：cm

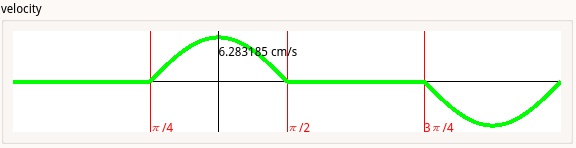


單位：in

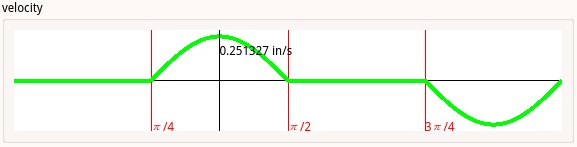


velocity公式：

單位：cm/s

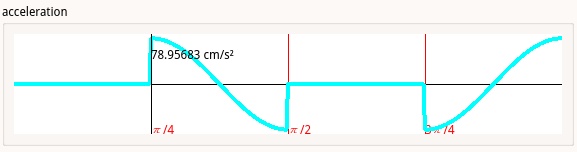


單位：in/s

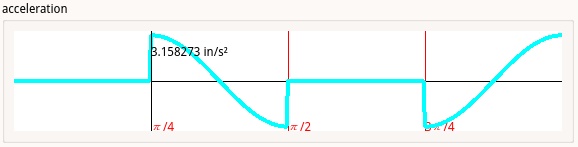


acceleration公式：

單位：cm/s2

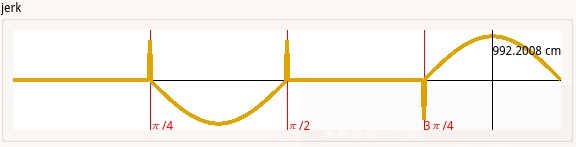


單位：in/s2

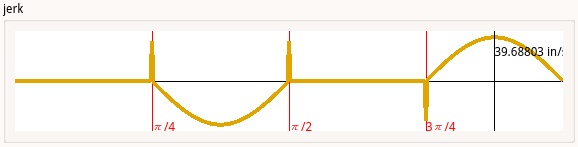


jerk公式：(寫反)

單位：cm/s3



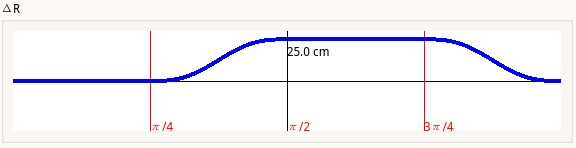
單位：in/s3



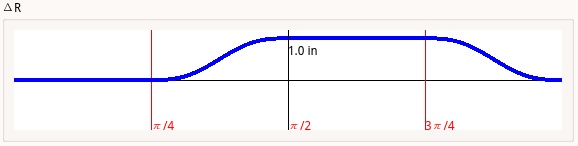
擺線運動

ΔR公式：

單位：cm

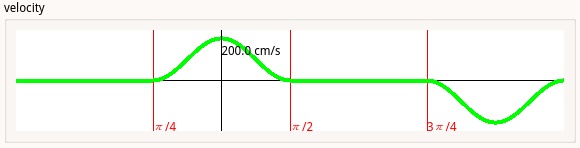


單位：in

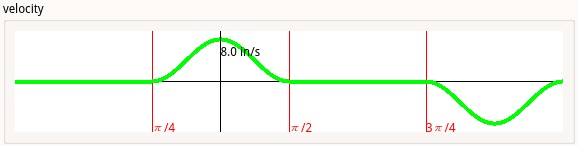


velocity公式：

單位：cm/s



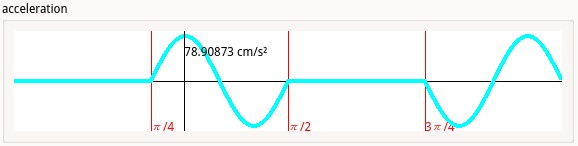
單位：in/s



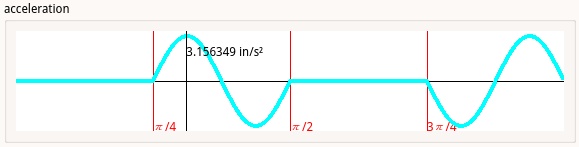
acceleration公式：

8\*pi\*\*2\*sin(4\*i/180\*pi)

單位：cm/s2



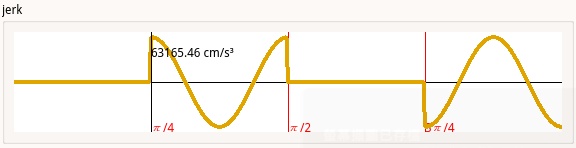
單位：in/s2



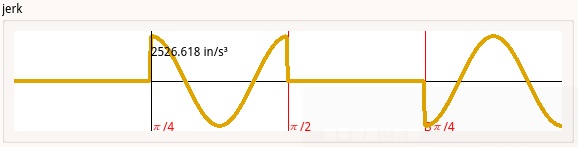
jerk公式：

256\*pi\*\*2\*cos(4\*i/180\*pi)

單位：cm/s3



單位：in/s3



程式碼簡述

簡單介紹程式碼中，公式運作的部份。

由於一部分的程式是用來產生圖形界面的，所以詳細內容不便繁述。

launch\_cam.py

這個腳本專門呼叫整個程式運作，使用Python直譯器可以馬上啟動core資料夾的程序。

main.py

主程式，操控界面所有選項。繼承了8個圖形並插入界面中。

tabs.py

設立兩大class（簡諧運動和擺線運動），各包含4個圖形畫布。這兩個class有附屬一個function用來切換公英制單位。

canvas.py

自創一個名為chart的class，繼承自QWidget class，相當一個空白區域，當為QWidget定義paintEvent（繪製事件）後，就能使用QPainter在區域中著色。這個事件的作為是**動態**的，因此用內部或外部function更改變數時，paintEvent可以畫出不同的圖形。

所有的圖表都是使用這個畫布畫出的，藉由更改算出的結果來改變每個畫布的內容。

formula.py

公式存放區，return結果回tab.py的兩大class，並把結果填入canvas.py的畫布中。

總共有8個函式，一次傳回：一筆360個的XY點座標集、y軸縮放倍率、一筆尋找最大值的數據集。