第十八屆旺宏科學獎

成果報告書

參賽編號：SA18-367

作品名稱：「跡」裡尋「規」——

四輪軌跡關係式之研究

姓名：吳少愚

關鍵字：轎車軌跡、軌跡關係式、自走車

摘要

本研究旨在探討轎車四輪軌跡關係式。利用車體結構的限制，例如:「後輪恆與車身平行，因此在不打滑、甩尾的情況下，後輪軌跡沿著切線延長軸距，便會是當時刻的前輪位置」，藉此發展出多種方向的推論方式。研究最後更將結論應用於諸多方面，例如在刑事鑑定中以實際辨析之方法，作圖求得輪距、軸距與行車方向等車輛資訊。

壹、研究動機

「咻!」一聲過彎，在迷人的跑跑卡丁車（賽車遊戲）世界裡，這似乎是最爽快的一件事了。但仔細一瞧，車子只留下了兩條後輪胎痕，難道是有什麼理由讓後輪自動跟著前輪的軌跡嗎？但現實中車輛經過彎便會留下前後兩組胎痕，不禁令我更加好奇——車子四輪胎痕具有什麼關係式存在呢？又可否將其關係式應用於其他方面上呢？

貳、研究目的及問題

一、找出四輪軌跡關係式。

(一)給定軸距、後輪軌跡，推導出以後輪表示的前輪函數。

(二)給定軸距、前輪軌跡，推導出以前輪表示的後輪函數。

(三)給定輪距、後左輪軌跡，推導出以後左輪表示的後右輪函數。

(四)給定輪距、後右輪軌跡，推導出以後右輪表示的後左輪函數。

(五)給定輪距、前左輪軌跡，推導出以前左輪表示的前右輪函數。

(六)給定輪距、前右輪軌跡，推導出以前右輪表示的前左輪函數。

二、從一組胎痕中找出車輛資訊。

(一)分辨四輪胎痕。

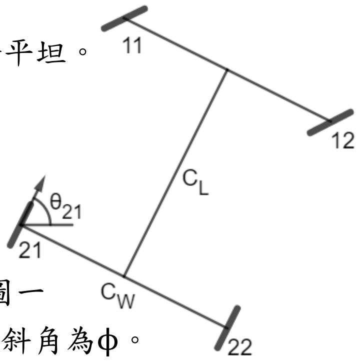
(二)軸距、軸寬。

(三)行車方向及速度。

參、研究設備及器材

自走車：含 Arduino UNO 控制板、馬達。 軟體：Arduino IDE、GeoGeBra、MATLAB 等。

肆、名詞釋義及假設



圖一

一、令此說明書中之車輛為鋼體及車輪無寬度之系統；路面為完全平坦。

二、軸距C_L：前輪軸心和後輪軸心之間的水平距離；

輪距C_W：左輪軸心和右輪軸心之間的水平距離。

三、前左輪軌跡$(X_{11}(t), Y_{11}(t))$，前右輪軌跡$(X_{12}(t), Y_{12}(t))$；

後左輪軌跡$(X_{21}(t), Y_{21}(t))$，後右輪軌跡$(X_{22}(t), Y_{22}(t))$。

四、前左輪斜率$\tan\theta_{11}$，前右輪斜率$\tan\theta_{12}$；

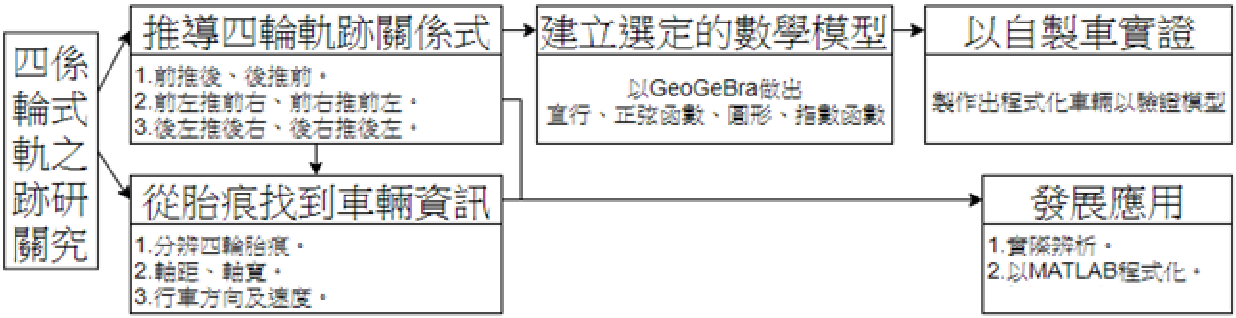
後左輪斜率$\tan\theta_{21}$，後右輪斜率$\tan\theta_{22}$，前輪相對於車身的偏斜角為ϕ。

五、前輪軌跡$(X_F(t), Y_F(t)) \equiv (X_{1*}(t), Y_{1*}(t))$；後輪軌跡$(X_R(t), Y_R(t)) \equiv (X_{2*}(t), Y_{2*}(t))$。

六、令此說明書中之車輛恆不打滑及甩尾。

七、此報告書說明圖中黑色為前輪、右側輪胎痕；紅色為後輪、左側輪胎痕，以方便辨識。

伍、研究方法及流程

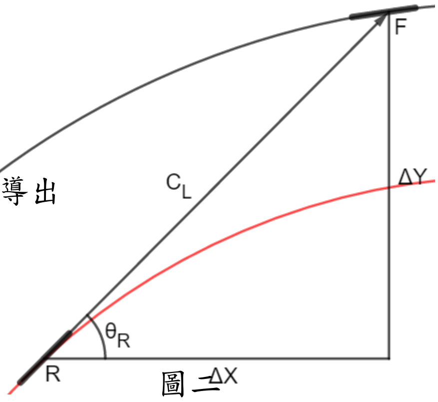


陸、研究過程

一、找出四輪軌跡關係式。

（一）給定C_L、$(X_R(t),Y_R(t))$，推導出$(X_F(t),Y_F(t))$。

一輛車中後左推前左及後右推前右，因結構對稱性可知推導出的公式將會相同，因此推導的皆為「後推前之公式」。如圖二，我們發現「後輪沿切線延長軸距，便會是當時的前輪位置」，因此，我們可以利用此車體限制來求出前輪的軌跡方程式。



圖二

定理 1.1

已知C_L、$(X_R(t),Y_R(t))$，若車輛向右行駛，即$X_F(t)-X_R(t)>0$，則前輪參數式$(X_F(t),Y_F(t))$可表示為：

$$X_F(t)=X_R(t)+\frac{C_L\cdot X_R{}'(t)}{\sqrt{X_R{}'(t)^2+Y_R{}'(t)^2}}$$

$$Y_F(t)=Y_R(t)+\frac{C_L\cdot Y_R{}'(t)}{\sqrt{X_R{}'(t)^2+Y_R{}'(t)^2}}$$

若車輛向左行駛$X_F(t)-X_R(t)<0$，則前輪參數式$(X_F(t),Y_F(t))$可表示為：

$$X_F(t)=X_R(t)-\frac{C_L\cdot X_R{}'(t)}{\sqrt{X_R{}'(t)^2+Y_R{}'(t)^2}}$$

$$Y_F(t)=Y_R(t)-\frac{C_L\cdot Y_R{}'(t)}{\sqrt{X_R{}'(t)^2+Y_R{}'(t)^2}}$$

【證明】

如圖二可推知：

若$X_F(t)-X_R(t)>0$（向右行駛）；

$$X_F(t)=X_R(t)+\Delta X=X_R(t)+C_L\cdot\cos\theta_R=X_R(t)+C_L\cdot\frac{1}{\sqrt{1+\tan^2\theta_R}}$$

$$= X_R(t) + C_L \cdot \frac{1}{\sqrt{1+\left(\frac{Y_R{}'(t)}{X_R{}'(t)}\right)^2}} = X_R(t) + \frac{C_L \cdot X_R{}'(t)}{\sqrt{X_R{}'(t)^2+Y_R{}'(t)^2}}$$

$$Y_F(t) = Y_R(t) + \Delta Y = Y_R(t) + C_L \cdot \sin\theta_R = Y_R(t) + C_L \cdot \frac{\tan\theta_R}{\sqrt{1+\tan^2\theta_R}}$$

$$= Y_R(t) + C_L \cdot \frac{\frac{Y_R{}'(t)}{X_R{}'(t)}}{\sqrt{1+\left(\frac{Y_R{}'(t)}{X_R{}'(t)}\right)^2}} = Y_R(t) + \frac{C_L \cdot Y_R{}'(t)}{\sqrt{X_R{}'(t)^2+Y_R{}'(t)^2}}$$ （式 1.1.1）

若$X_F(t) - X_R(t) < 0$(向左行駛)；

同理可得證。(式 1.1.2)

定理 1.2

當後輪軌跡為一圓，已知C_L、$(X_R(t), Y_R(t)) = (r \cdot \cos t, r \cdot \sin t)$，則若車輛逆時針行駛，則前輪參數式$(X_F(t), Y_F(t))$可表示為：

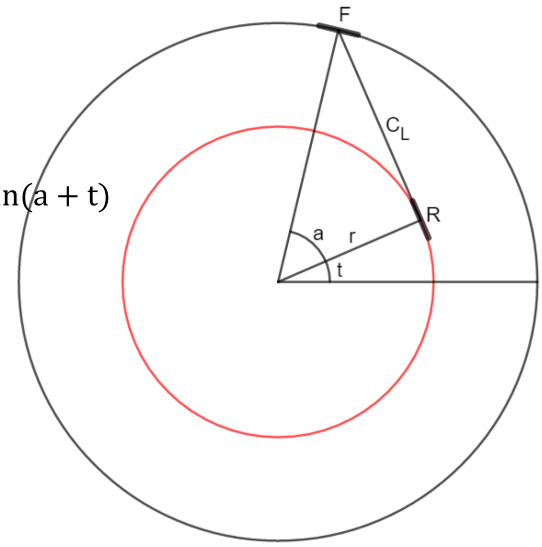
$$X_F(t) = r \cdot \cos t - C_L \cdot \sin t$$
$$Y_F(t) = r \cdot \sin t + C_L \cdot \cos t$$

若車輛順時針行駛，則前輪參數式$(X_F(t), Y_F(t))$可表示為：

$$X_F(t) = r \cdot \cos t + C_L \cdot \sin t$$
$$Y_F(t) = r \cdot \sin t - C_L \cdot \cos t$$

【證明】

如圖三可推知：



圖三

若車輛逆時針行駛：

$$X_F(t) = \sqrt{r^2 + C_L{}^2} \cdot \cos(a+t)\,;\, Y_F(t) = \sqrt{r^2 + C_L{}^2} \cdot \sin(a+t)$$

以合角公式展開：

$$X_F(t) = \sqrt{r^2 + C_L{}^2} \cdot (\cos a \cdot \cos t - \sin a \cdot \sin t)$$

$$= \sqrt{r^2 + C_L{}^2} \cdot \left(\frac{r}{\sqrt{r^2+C_L{}^2}} \cdot \cos t - \frac{C_L}{\sqrt{r^2+C_L{}^2}} \cdot \sin t\right)$$

$$= r \cdot \cos t - C_L \cdot \sin t$$

$$X_F(t) = \sqrt{r^2 + C_L{}^2} \cdot (\sin a \cdot \cos t + \cos a \cdot \sin t)$$

$$= \sqrt{r^2 + C_L{}^2} \cdot \left(\frac{C_L}{\sqrt{r^2+C_L{}^2}} \cdot \cos t + \frac{r}{\sqrt{r^2+C_L{}^2}} \cdot \sin t\right) = r \cdot \sin t + C_L \cdot \cos t$$ （式 1.2.1）

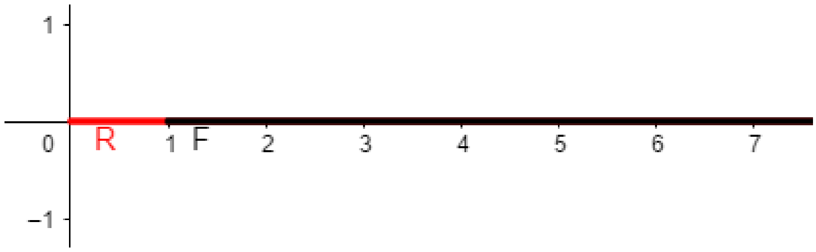
若車輛順時針行駛：

同理可得證。(式 1.2.2)

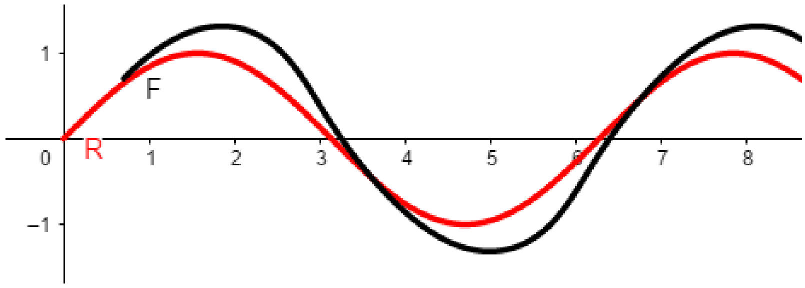
2.建立選定的數學模型：

利用 GeoGeBra 語法：curve[f(t), g(t), t, a, b]指令會產生參數式(f(t), g(t))的圖形，t 的範圍為 a≤t≤b，試做出幾種數學模型。

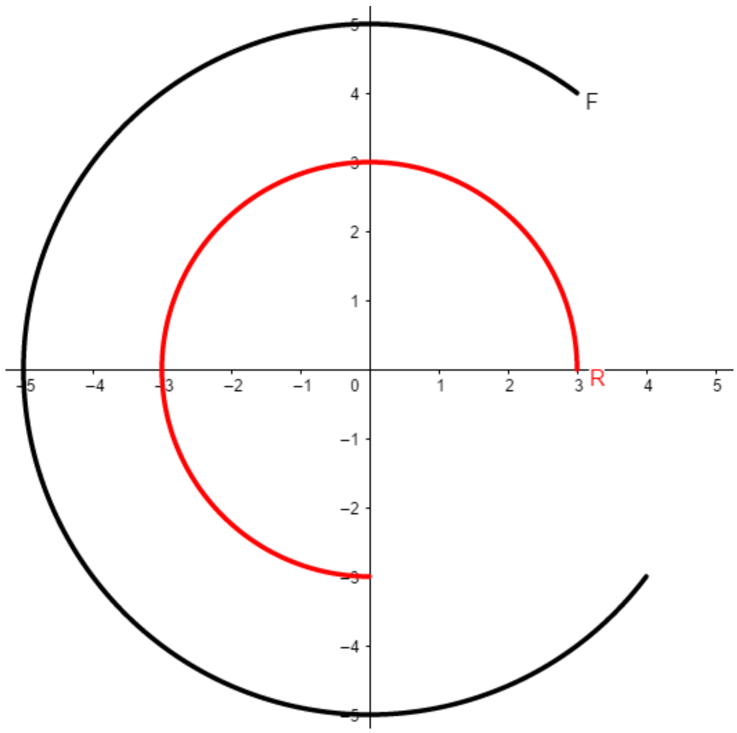
(1)直行：$C_L = 1$、$\left(X_R(t), Y_R(t)\right) = (t, 0)$、$X_F(t) - X_R(t) > 0$，化簡後得：



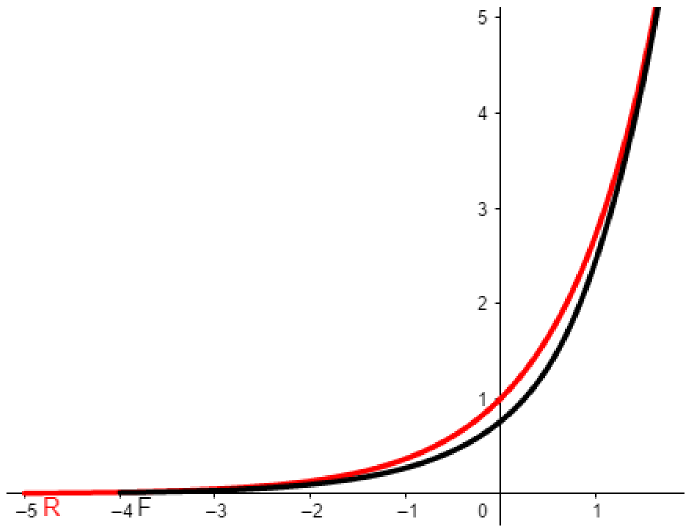
(2)正弦：$C_L = 1$、$\left(X_R(t), Y_R(t)\right) = (t, \sin t)$、$X_F(t) - X_R(t) > 0$，化簡後得：



(3)圓形：$C_L = 4$、$\left(X_R(t), Y_R(t)\right) = (3\cos t, 3\sin t)$、車輛逆時針行駛，可得：

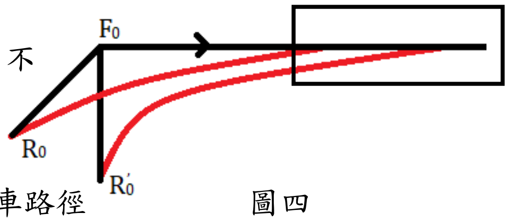


(4)指數：$C_L = 1$、$\left(X_R(t), Y_R(t)\right) = (t, \exp(t))$、$X_F(t) - X_R(t) > 0$，化簡後得下頁圖：



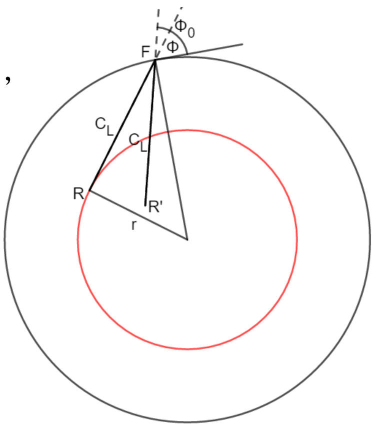
（二）給定C_L、$(X_F(t), Y_F(t))$，推導出$(X_R(t), Y_R(t))$。

從圖四可以知道，在相同的C_L、$(X_F(t), Y_F(t))$下，不同的ϕ_0仍會造成不一樣的後輪軌跡，因此在不知道ϕ_0為何的情況下，我們只能討論後半段(圖三黑框處)的軌跡趨向，如此一來便不會受到ϕ_0的影響；而只要行車路徑夠長、轉彎不太過劇烈(曲率變化率不大)，便會趨於某種的運行模式，為此我先提出了一個猜想：



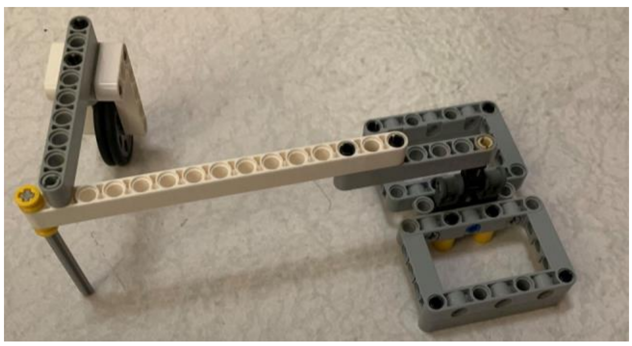
圖四

從後推前的結果中我們知道當後輪軌跡為一半徑為 r 的圓時，前輪軌跡將會形成一個半徑為$\sqrt{r^2 + {C_L}^2}$的圓(如圖三)，但若此時將後輪擺放於當時半徑為 r 的圓以外區域上(圖五非紅色區域)，再以前輪繞半徑為$\sqrt{r^2 + {C_L}^2}$的同心圓，後輪軌跡是否會逐漸回到半徑為 r 的圓上呢?



圖五

因此，我做了一個簡單的定性實驗：



右圖裝置能將前輪軌跡限制於一半徑為 17 公分的圓內，此時$C_L = 8$；若在下面墊一張畫有半徑 17、15 公分的圓的紙上，會發現隨著前輪的行走，後輪果真逐漸回到半徑為 15 公分的圓上！我稱之這個現象為後輪的「追隨性」。如此一來，若行車路徑夠長、轉彎不太過劇烈(曲率變化率不大)，便會滿足追隨性後的結果，也就可以推導出後半段的行車趨向。

定理 2.1

已知C_L、$(X_F(t),Y_F(t))$，若車輛向右行駛，即$X_F(t)-X_R(t)>0$，則後輪軌跡趨向$(X_R(t),Y_R(t))$可表示為：

$$X_R(t)=X_F(t)-C_L\cdot\frac{1}{\sqrt{1+M^2}}$$

$$Y_R(t)=Y_F(t)-C_L\cdot\frac{M}{\sqrt{1+M^2}}$$

$$M\equiv\frac{\dfrac{C_L\cdot\left(\frac{Y_F'(t)}{X_F'(t)}\right)'}{\sqrt{\left(1+\left(\frac{Y_F'(t)}{X_F'(t)}\right)^2\right)^3-{C_L}^2\cdot\left(\frac{Y_F'(t)}{X_F'(t)}\right)'^2}}+\dfrac{Y_F'(t)}{X_F'(t)}}{1-\dfrac{C_L\cdot\left(\frac{Y_F'(t)}{X_F'(t)}\right)'}{\sqrt{\left(1+\left(\frac{Y_F'(t)}{X_F'(t)}\right)^2\right)^3-{C_L}^2\cdot\left(\frac{Y_F'(t)}{X_F'(t)}\right)'^2}}\cdot\dfrac{Y_F'(t)}{X_F'(t)}}$$

若車輛向左行駛$X_F(t)-X_R(t)<0$，則後輪軌跡趨向$(X_F(t),Y_F(t))$可表示為：

$$X_R(t)=X_F(t)+C_L\cdot\frac{1}{\sqrt{1+M^2}}$$

$$Y_R(t)=Y_F(t)+C_L\cdot\frac{M}{\sqrt{1+M^2}}$$

【證明】

若$X_F(t)-X_R(t)>0$(向右行駛)；將後推前的公式移項：

$$X_R(t)=X_F(t)-C_L\cdot\frac{1}{\sqrt{1+\tan^2\theta_R}} \quad \text{———}(1)$$

$$Y_R(t)=Y_F(t)-C_L\cdot\frac{\tan\theta_R}{\sqrt{1+\tan^2\theta_R}} \quad \text{———}(2)$$

由下頁圖六可知：

$\tan\phi=\tan(\theta_R-\theta_F)$；

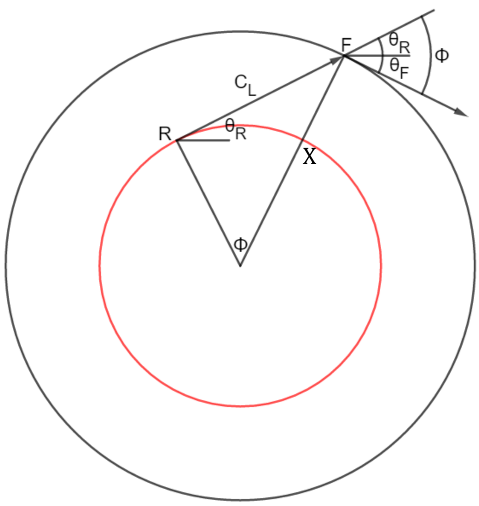
$$\tan\theta_R=\tan(\phi+\theta_F)=\frac{\tan\phi+\tan\theta_F}{1-\tan\phi\cdot\tan\theta_F} \quad \text{———}(3)$$

求$\tan\phi$：

$X\equiv$前輪某一時刻所對應的曲率半徑；

$$X=\frac{\left(1+\left(\frac{Y_F'(t)}{X_F'(t)}\right)^2\right)^{\frac{3}{2}}}{\left(\frac{Y_F'(t)}{X_F'(t)}\right)'}$$

$$\sin\phi=\frac{C_L}{X}=\frac{C_L\cdot\left(\frac{Y_F'(t)}{X_F'(t)}\right)'}{\left(1+\left(\frac{Y_F'(t)}{X_F'(t)}\right)^2\right)^{\frac{3}{2}}}$$



根據物理限制：$-90° < \phi < 90°$，因此

$$\tan\phi = \frac{\sin\phi}{\sqrt{1-\sin^2\phi}} = \frac{\frac{C_L\cdot(\frac{Y_F'(t)}{X_F'(t)})'}{(1+(\frac{Y_F'(t)}{X_F'(t)})^2)^{\frac{3}{2}}}}{\sqrt{1-(\frac{C_L\cdot(\frac{Y_F'(t)}{X_F'(t)})'}{(1+(\frac{Y_F'(t)}{X_F'(t)})^2)^{\frac{3}{2}}})^2}} = \frac{C_L\cdot(\frac{Y_F'(t)}{X_F'(t)})'}{\sqrt{(1+(\frac{Y_F'(t)}{X_F'(t)})^2)^3-C_L{}^2\cdot(\frac{Y_F'(t)}{X_F'(t)})'^2}} \quad \text{———}(4)$$

將(4)、$\tan\theta_F = \frac{Y_F'(t)}{X_F'(t)}$代入(3)得：

$$\tan\theta = \frac{\frac{C_L\cdot(\frac{Y_F'(t)}{X_F'(t)})'}{\sqrt{(1+(\frac{Y_F'(t)}{X_F'(t)})^2)^3-C_L{}^2\cdot(\frac{Y_F'(t)}{X_F'(t)})'^2}}+\frac{Y_F'(t)}{X_F'(t)}}{1-\frac{C_L\cdot(\frac{Y_F'(t)}{X_F'(t)})'}{\sqrt{(1+(\frac{Y_F'(t)}{X_F'(t)})^2)^3-C_L{}^2\cdot(\frac{Y_F'(t)}{X_F'(t)})'^2}}\cdot\frac{Y_F'(t)}{X_F'(t)}} \quad \text{———}(5)$$

令$M = \tan\theta$，代入(1)、(2)、(5)，得證(式 2.1.1)。
若$X_F(t) - X_R(t) < 0$(向左行駛)；
同理可得證。(式 2.1.2)

定理 2.2

當後輪軌跡為一圓，已知C_L、$(X_F(t), Y_F(t)) = (r\cdot\cos t, r\cdot\sin t)$，則若車輛逆時針行駛，則後輪軌跡趨向$(X_R(t), Y_R(t))$可表示為：

$$X_R(t) = \frac{r^2 - C_L{}^2}{r}\cdot\cos t + \frac{C_L\sqrt{r^2 - C_L{}^2}}{r}\cdot\sin t; Y_R(t) = \frac{r^2 - C_L{}^2}{r}\cdot\sin t - \frac{C_L\sqrt{r^2 - C_L{}^2}}{r}\cdot\cos t$$

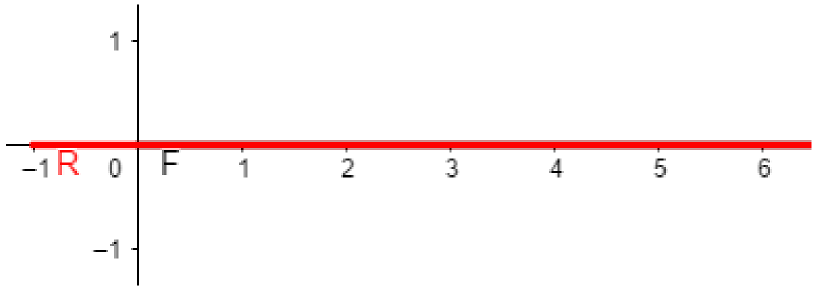
若車輛順時針行駛，則後輪軌跡趨向$(X_R(t), Y_R(t))$可表示為：

$$X_R(t) = \frac{r^2 - C_L{}^2}{r}\cdot\cos t - \frac{C_L\sqrt{r^2 - C_L{}^2}}{r}\cdot\sin t; Y_R(t) = \frac{r^2 - C_L{}^2}{r}\cdot\sin t + \frac{C_L\sqrt{r^2 - C_L{}^2}}{r}\cdot\cos t$$

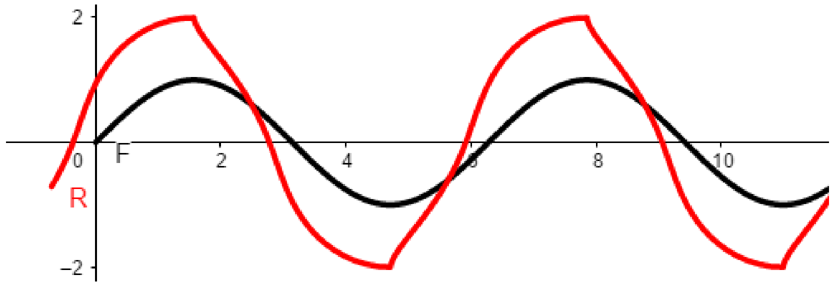
【證明】同**定理 1.2** 之證法，將合角改成差角即可證(式 2.2.1)(式 2.2.2)。

2. 建立選定的數學模型：

(1)直行：$C_L = 1$、$(X_F(t), Y_F(t)) = (t, 0)$、$X_F(t) - X_R(t) > 0$，化簡後得：

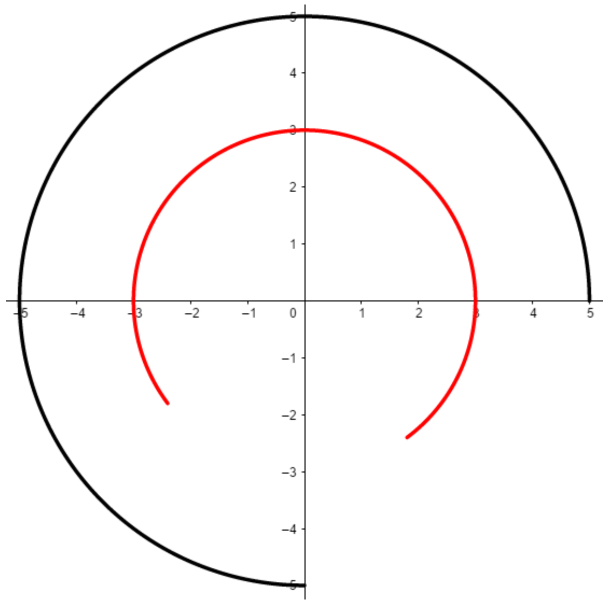


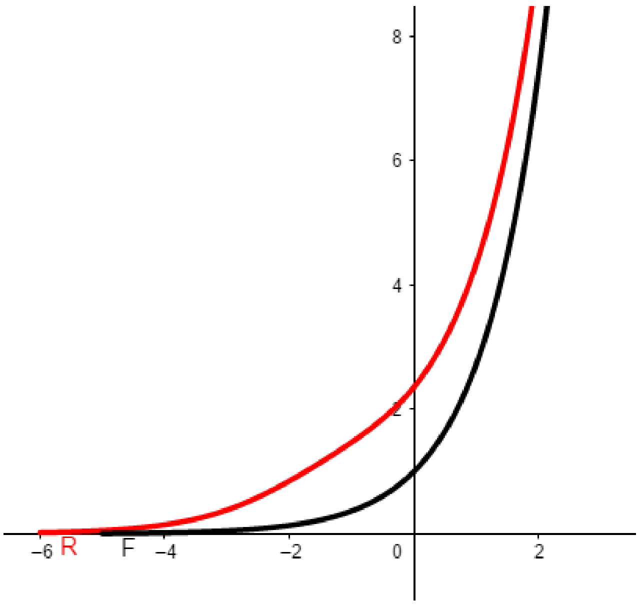
(2)正弦：$C_L = 1$、$\left(X_F(t), Y_F(t)\right) = (t, \sin t)$、$X_F(t) - X_R(t) > 0$，化簡後得：



(3)圓形：$C_L = 4$、$\left(X_F(t), Y_F(t)\right) = (5\cos t, 5\sin t)$、車輛逆時針行駛，可得下左圖：

(4)指數：$C_L = 1$、$\left(X_F(t), Y_F(t)\right) = (t, \exp(t))$、$X_F(t) - X_R(t) > 0$，化簡後得下右圖：





從建模的圖中可以發現，當轉彎太過劇烈(曲率變化率大)，便不能在每一個時刻都滿足追隨性後的結果，如正弦、指數函數中有部分圖形顯得不合理。

(三)給定C_W、$(X_{21}(t), Y_{21}(t))$，推導出$(X_{22}(t), Y_{22}(t))$。

定理 3.1

已知C_W、$(X_{21}(t), Y_{21}(t))$，若車輛向右行駛，即$X_F(t) - X_R(t) > 0$，則後右輪參數式$(X_{22}(t), Y_{22}(t))$可表示為：

$$X_{22}(t) = X_{21}(t) + C_W\left(\frac{Y_{21}'(t)}{\sqrt{X_{21}'(t)^2 + Y_{21}'(t)^2}}\right)$$

$$Y_{22}(t) = Y_{21}(t) - C_W\left(\frac{X_{21}'(t)}{\sqrt{X_{21}'(t)^2 + Y_{21}'(t)^2}}\right)$$

若車輛向左行駛$X_F(t) - X_R(t) < 0$，則後右輪參數式$(X_{22}(t), Y_{22}(t))$可表示為：

$$X_{22}(t) = X_{21}(t) + C_W\left(\frac{Y_{21}'(t)}{\sqrt{X_{21}'(t)^2 + Y_{21}'(t)^2}}\right)$$

$$Y_{22}(t) = Y_{21}(t) - C_W\left(\frac{X_{21}'(t)}{\sqrt{X_{21}'(t)^2 + Y_{21}'(t)^2}}\right)$$

一般車輛的架構垂直構成，因此後推前中「後輪沿切線延長軸距，便會是當時的前輪位置」的想法，在後左推後右中可改寫為「後左輪沿法線延長輪距，便會是當時的後右輪位置」，如此我們便可以利用此車體限制來求出後右輪的軌跡方程式。

【證明】

同**定理** 1.1 之證法，將切線$(\frac{Y_{21}'(t)}{X_{21}'(t)})$改成法線$-(\frac{X_{21}'(t)}{Y_{21}'(t)})$即可得證(式 3.1.1)(式 3.1.2)。

定理 3.2

當後左輪軌跡為一圓，已知C_W、$(X_{21}(t), Y_{21}(t)) = (r \cdot \cos t, r \cdot \sin t)$，則若車輛逆時針行駛，則後右輪參數式$(X_{22}(t), Y_{22}(t))$可表示為：

$$X_{22}(t) = (r + C_W) \cdot \cos t$$
$$Y_{22}(t) = (r + C_W) \cdot \sin t$$

若車輛順時針行駛，則後右輪參數式$(X_{22}(t), Y_{22}(t))$可表示為：

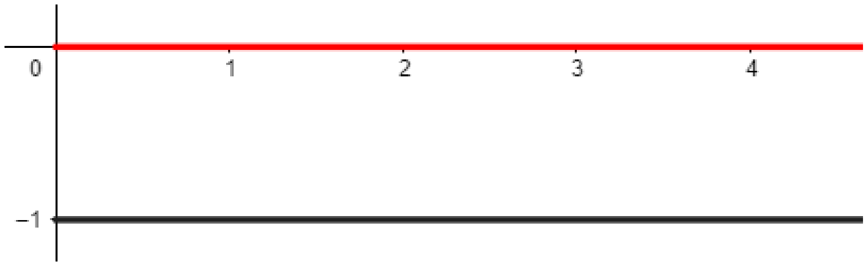
$$X_{22}(t) = (r - C_W) \cdot \cos t$$
$$Y_{22}(t) = (r - C_W) \cdot \sin t$$

【證明】

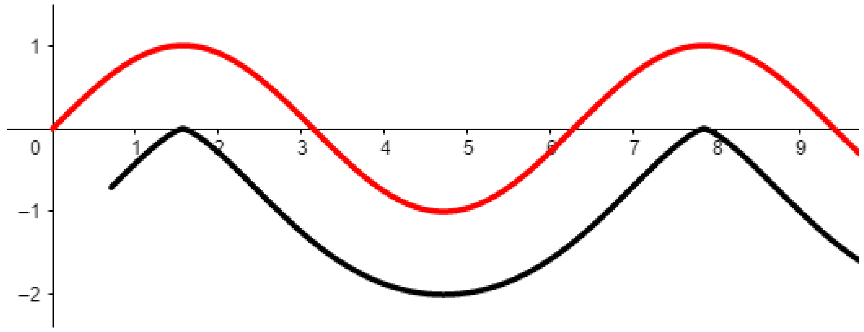
車輛行駛一圓，則法線必過圓心，沿著法線延長便會得到同心圓；因此後右輪軌跡與後左輪軌跡為同心圓，且同一個時刻位於同一個有向角度，在逆時針行車時後右輪繞圈之半徑大了C_W；而在順時針行車時後右輪繞圈之半徑則會小C_W，得證。

2. 建立選定的數學模型：

(1)直行：$C_W = 1$、$\big(X_{21}(t), Y_{21}(t)\big) = (t, 0)$、$X_F(t) - X_R(t) > 0$，化簡後得下頁圖：

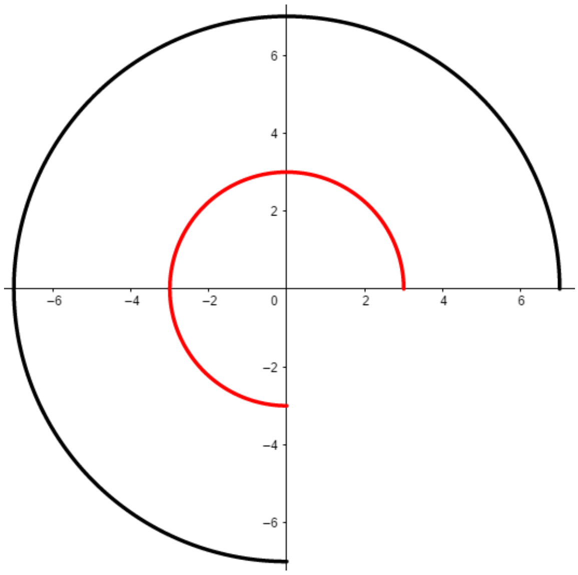


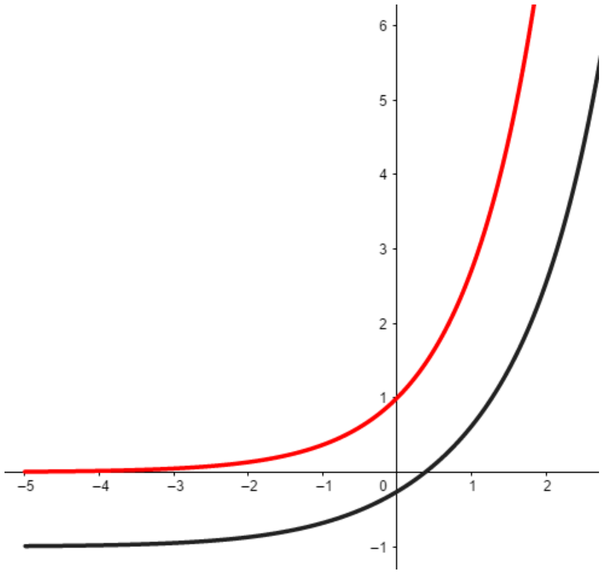
(2)正弦：$C_W = 1$、$\big(X_{21}(t), Y_{21}(t)\big) = (t, \sin t)$、$X_F(t) - X_R(t) > 0$，化簡後得：



(3)圓形：$C_W = 4$、$\big(X_{21}(t), Y_{21}(t)\big) = (3\cos t, 3\sin t)$、車輛逆時針行駛，可得下頁左圖：

(4)指數：$C_W = 1$、$\big(X_{21}(t), Y_{21}(t)\big) = (t, \exp(t))$、$X_F(t) - X_R(t) > 0$，化簡得下頁右圖：





（四）給定C_W、$(X_{22}(t), Y_{22}(t))$，推導出$(X_{21}(t), Y_{21}(t))$。

與後左推後右的方式相同，只有其延長之向量與後左推後右的向量互為相反向量，因此**定理 4.1**、**定理 4.2** 與**定理 3.1**、**定理 3.2** 證明方法相同，下頁便不在贅述證明。

定理 4.1

已知C_W、$(X_{22}(t), Y_{22}(t))$，若車輛向右行駛，即$X_F(t) - X_R(t) > 0$，則後左輪參數式$(X_{21}(t), Y_{21}(t))$可表示為：

$$X_{21}(t) = X_{22}(t) - C_W\left(\frac{Y_{22}'(t)}{\sqrt{X_{22}'(t)^2 + Y_{22}'(t)^2}}\right)$$

$$Y_{21}(t) = Y_{22}(t) + C_W\left(\frac{X_{22}'(t)}{\sqrt{X_{22}'(t)^2 + Y_{22}'(t)^2}}\right)$$

若車輛向左行駛$X_F(t) - X_R(t) < 0$，則後左輪參數式$(X_{21}(t), Y_{21}(t))$可表示為：

$$X_{21}(t) = X_{22}(t) - C_W\left(\frac{Y_{22}'(t)}{\sqrt{X_{22}'(t)^2 + Y_{22}'(t)^2}}\right)$$

$$Y_{21}(t) = Y_{22}(t) + C_W\left(\frac{X_{22}'(t)}{\sqrt{X_{22}'(t)^2 + Y_{22}'(t)^2}}\right)$$

定理 4.2

當後右輪軌跡為一圓，已知C_W、$(X_{22}(t), Y_{22}(t)) = (r \cdot \cos t, r \cdot \sin t)$，則若車輛逆時針行駛，則後左輪參數式$(X_{21}(t), Y_{21}(t))$可表示為：

$$X_{21}(t) = (r - C_W) \cdot \cos t$$

$$Y_{21}(t) = (r - C_W) \cdot \sin t$$

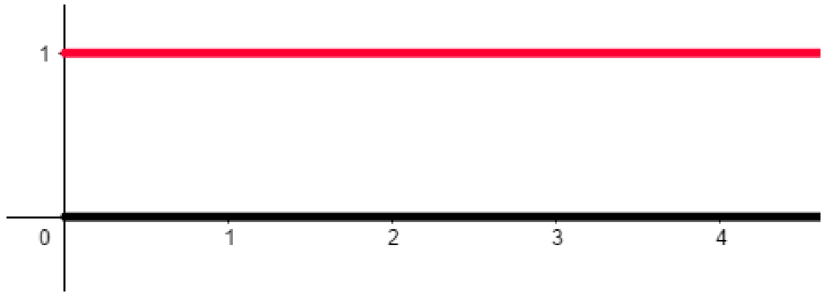
若車輛順時針行駛，則後左輪參數式$(X_{21}(t), Y_{21}(t))$可表示為：

$$X_{21}(t) = (r + C_W) \cdot \cos t$$

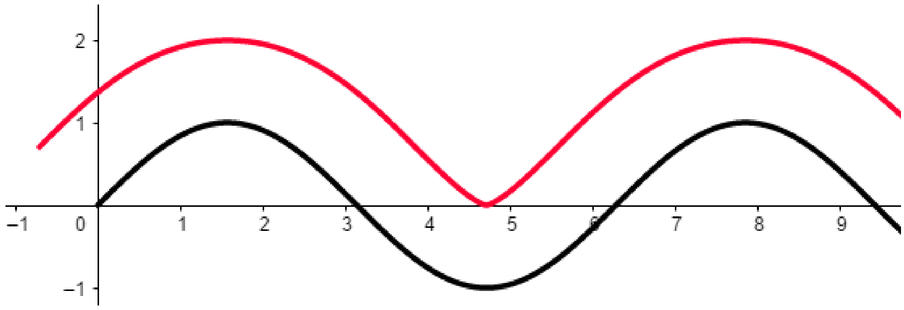
$$Y_{21}(t) = (r + C_W) \cdot \sin t$$

2.建立選定的數學模型：

(1)直行：$C_W=1$、$(X_{22}(t),Y_{22}(t))=(t,0)$、$X_F(t)-X_R(t)>0$，化簡後得：

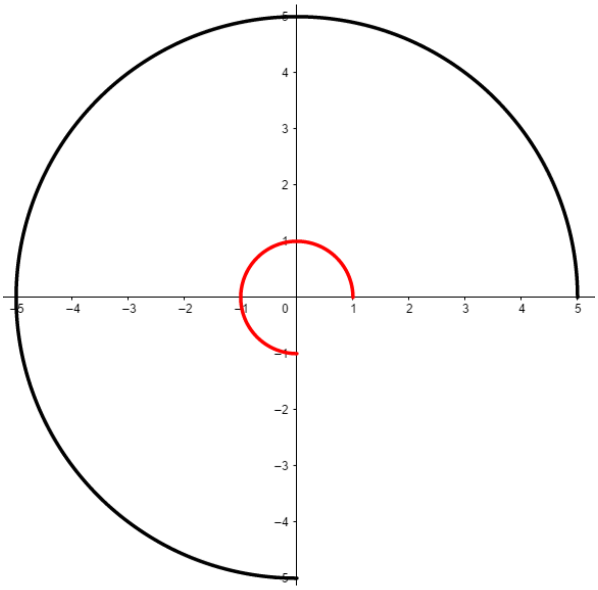


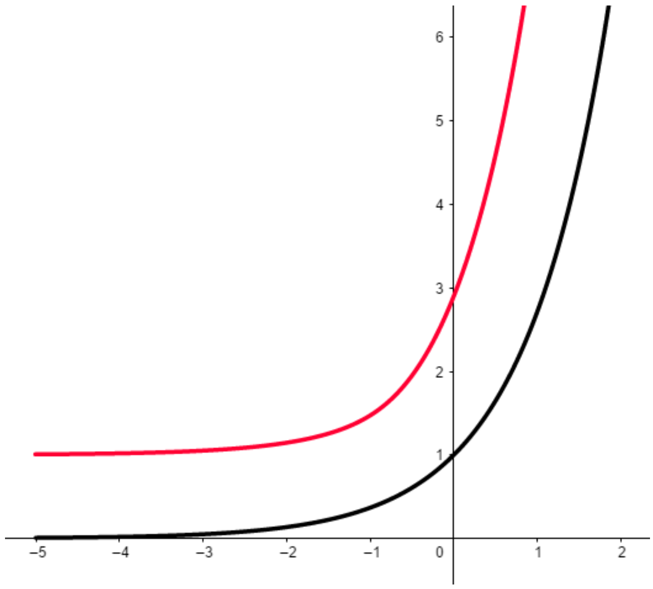
(2)正弦：$C_W=1$、$(X_{22}(t),Y_{22}(t))=(t,\sin t)$、$X_F(t)-X_R(t)>0$，化簡後得：



(3)圓形：$C_W=4$、$(X_{22}(t),Y_{22}(t))=(5\cos t,5\sin t)$、車輛逆時針行駛，可得下左圖：

(4)指數：$C_W=1$、$(X_{22}(t),Y_{22}(t))=(t,\exp(t))$、$X_F(t)-X_R(t)>0$，化簡後得下右圖：





※(三)、(四)之討論：

從平行曲線的定義來看，會發現後輪的兩側必會形成平行；因此在極端數據中，如：正弦函數的圖中有一部分會出現「急轉彎」的現象，並非不可能產生、也不是不合理的。

(五)給定C_W、$(X_{11}(t),Y_{11}(t))$，推導出$(X_{12}(t),Y_{12}(t))$：

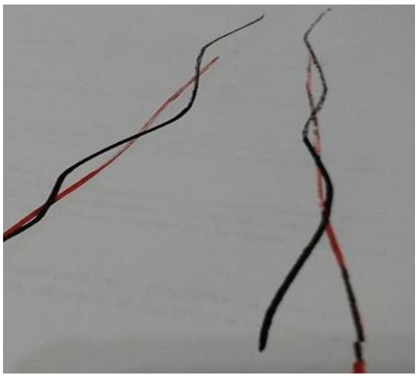
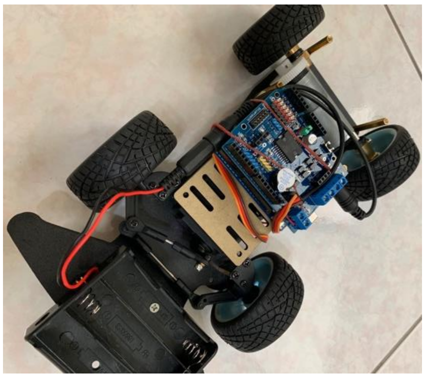
根據圖一可以知道後輪兩側互相平行，然而連接前輪兩側的橫桿斜率卻是由θ_R所控制，因此與前推後一樣必須給定ϕ_0、C_L才能解數值解，以原條件是無限多組解的；因此我將在未來展望，用 MATLAB 來解出數值解。

(六)給定C_W、$(X_{12}(t), Y_{12}(t))$，推導出$(X_{11}(t), Y_{11}(t))$：

理由同(五)。

二、以自製車輛實證： 尚未完全完成

我曾用一般的自製模型驗證，發現實驗結果與定理所述是相同的，但無法得出完整的結論及數據分析；因此我利用 Arduino UNO 控制板、馬達、18650 充電電池及一些零組件來做出一輛自走車，配合程式碼來驗證定理。

※目前進度：

以後推前為例，我們目的是做出可程式化的車輛讓後輪跑出特定軌跡，以驗證前輪軌跡是否如同定理 1.1 所述，因此我們找來了使用 Arduino IDE 的遙控車程式自行修改；然而自製車的輸出是$v(t)$、$\phi(t)$：$v(t)$將位置函數微分一次即可求，而$\phi(t)$則須利用定理 2.1 之證明中，$\tan\phi = \tan(\theta_R - \theta_F)$來求得，利用 MATLAB 便可得到$v(t)$及 $\phi(t)$，Arduino IDE 及 MATLAB 的程式碼寫於附錄二。

三、從一組胎痕中找出車輛資訊。

根據一、的想法：「後輪沿切線延長軸距，便會交於前輪軌跡」；因此我們只要依照下列步驟即可得到行經車輛的資訊。

步驟 1：找出目標車輛胎痕，利用後側輪左右互相平行的特性，找到其一組前後輪的系統。

步驟 2：在兩條軌跡上各做 n(n≧2)條切線，共 2n 條，產生 4n 個方向的射線。

步驟 3：其中必有 n 條射線接到另一胎痕的長度相同，此長度為軸距；射線方向必為同向(順向或反向)，便是當時行進方向；此 n 條射線起始點的共同胎痕即為後輪胎痕。

步驟 4：根據前後輪關係，即可判斷左右兩側，在一側作法線，交另一側的距離即為輪距。

藉此，即可分辨(一)四輪胎痕(二)軸距、輪距(三)行車方向。

然而，車輛胎痕的形成方式與速度無關！只要在不打滑或甩尾的情況下，胎痕的形成便不受速度所影響；因此無法從一組胎痕中，看出當時的行車速度。

柒、研究成果及未來展望

一、公式推導：

(一)前推後：

已知C_L、$(X_R(t), Y_R(t))$，若車輛向右行駛，即$X_F(t) - X_R(t) > 0$，則前輪參數式$(X_F(t), Y_F(t))$可表示為：

$$X_F(t) = X_R(t) + \frac{C_L \cdot X_R{}'(t)}{\sqrt{X_R{}'(t)^2 + Y_R{}'(t)^2}}$$

$$Y_F(t) = Y_R(t) + \frac{C_L \cdot Y_R{}'(t)}{\sqrt{X_R{}'(t)^2 + Y_R{}'(t)^2}}$$

若車輛向左行駛$X_F(t) - X_R(t) < 0$，則前輪參數式$(X_F(t), Y_F(t))$可表示為：

$$X_F(t) = X_R(t) - \frac{C_L \cdot X_R{}'(t)}{\sqrt{X_R{}'(t)^2 + Y_R{}'(t)^2}}$$

$$Y_F(t) = Y_R(t) - \frac{C_L \cdot Y_R{}'(t)}{\sqrt{X_R{}'(t)^2 + Y_R{}'(t)^2}}$$

當後輪軌跡為一圓，已知C_L、$\left(X_R(t), Y_R(t)\right) = (r \cdot \cos t, r \cdot \sin t)$，則若車輛逆時針行駛，則前輪參數式$(X_F(t), Y_F(t))$可表示為：

$$X_F(t) = r \cdot \cos t - C_L \cdot \sin t$$

$$Y_F(t) = r \cdot \sin t + C_L \cdot \cos t$$

若車輛順時針行駛，則前輪參數式$(X_F(t), Y_F(t))$可表示為：

$$X_F(t) = r \cdot \cos t + C_L \cdot \sin t$$

$$Y_F(t) = r \cdot \sin t - C_L \cdot \cos t$$

(二)前推後：

已知C_L、$(X_F(t), Y_F(t))$，若車輛向右行駛，即$X_F(t) - X_R(t) > 0$，則後輪軌跡趨向$(X_R(t), Y_R(t))$可表示為：

$$X_R(t) = X_F(t) - C_L \cdot \frac{1}{\sqrt{1 + M^2}}$$

$$Y_R(t) = Y_F(t) - C_L \cdot \frac{M}{\sqrt{1 + M^2}}$$

$$M \equiv \frac{\dfrac{C_L \cdot \left(\frac{Y_F{}'(t)}{X_F{}'(t)}\right)'}{\sqrt{\left(1 + \left(\frac{Y_F{}'(t)}{X_F{}'(t)}\right)^2\right)^3 - C_L{}^2 \cdot \left(\frac{Y_F{}'(t)}{X_F{}'(t)}\right)'^2}} + \dfrac{Y_F{}'(t)}{X_F{}'(t)}}{1 - \dfrac{C_L \cdot \left(\frac{Y_F{}'(t)}{X_F{}'(t)}\right)'}{\sqrt{\left(1 + \left(\frac{Y_F{}'(t)}{X_F{}'(t)}\right)^2\right)^3 - C_L{}^2 \cdot \left(\frac{Y_F{}'(t)}{X_F{}'(t)}\right)'^2}} \cdot \dfrac{Y_F{}'(t)}{X_F{}'(t)}}$$

若車輛向左行駛$X_F(t) - X_R(t) < 0$，則後輪軌跡趨向$(X_F(t), Y_F(t))$可表示為：

$$X_R(t) = X_F(t) + C_L \cdot \frac{1}{\sqrt{1 + M^2}}$$

$$Y_R(t) = Y_F(t) + C_L \cdot \frac{M}{\sqrt{1 + M^2}}$$

當後輪軌跡為一圓，已知C_L、$\left(X_F(t), Y_F(t)\right) = (r \cdot \cos t, r \cdot \sin t)$，則若車輛逆時針行駛，則後輪軌跡趨向$(X_R(t), Y_R(t))$可表示為：

$$X_R(t)=\frac{r^2-{C_L}^2}{r}\cdot\cos t+\frac{C_L\sqrt{r^2-{C_L}^2}}{r}\cdot\sin t;Y_R(t)=\frac{r^2-{C_L}^2}{r}\cdot\sin t-\frac{C_L\sqrt{r^2-{C_L}^2}}{r}\cdot\cos t$$

若車輛順時針行駛，則後輪軌跡趨向$(X_R(t),Y_R(t))$可表示為：

$$X_R(t)=\frac{r^2-{C_L}^2}{r}\cdot\cos t-\frac{C_L\sqrt{r^2-{C_L}^2}}{r}\cdot\sin t;Y_R(t)=\frac{r^2-{C_L}^2}{r}\cdot\sin t+\frac{C_L\sqrt{r^2-{C_L}^2}}{r}\cdot\cos t$$

（三)後左推後右：

已知C_W、$(X_{21}(t),Y_{21}(t))$，若車輛向右行駛，即$X_F(t)-X_R(t)>0$，則後右輪參數式$(X_{22}(t),Y_{22}(t))$可表示為：

$$X_{22}(t)=X_{21}(t)+C_W(\frac{{Y_{21}}'(t)}{\sqrt{{X_{21}}'(t)^2+{Y_{21}}'(t)^2}})$$

$$Y_{22}(t)=Y_{21}(t)-C_W(\frac{{X_{21}}'(t)}{\sqrt{{X_{21}}'(t)^2+{Y_{21}}'(t)^2}})$$

若車輛向左行駛$X_F(t)-X_R(t)<0$，則後右輪參數式$(X_{22}(t),Y_{22}(t))$可表示為：

$$X_{22}(t)=X_{21}(t)+C_W(\frac{{Y_{21}}'(t)}{\sqrt{{X_{21}}'(t)^2+{Y_{21}}'(t)^2}})$$

$$Y_{22}(t)=Y_{21}(t)-C_W(\frac{{X_{21}}'(t)}{\sqrt{{X_{21}}'(t)^2+{Y_{21}}'(t)^2}})$$

當後左輪軌跡為一圓，已知C_W、$(X_{21}(t),Y_{21}(t))=(r\cdot\cos t,r\cdot\sin t)$，則若車輛逆時針行駛，則後右輪參數式$(X_{22}(t),Y_{22}(t))$可表示為：

$$X_{22}(t)=(r+C_W)\cdot\cos t$$

$$Y_{22}(t)=(r+C_W)\cdot\sin t$$

若車輛順時針行駛，則後右輪參數式$(X_{22}(t),Y_{22}(t))$可表示為：

$$X_{22}(t)=(r-C_W)\cdot\cos t$$

$$Y_{22}(t)=(r-C_W)\cdot\sin t$$

（四)後右推後左：

已知C_W、$(X_{22}(t),Y_{22}(t))$，若車輛向右行駛，即$X_F(t)-X_R(t)>0$，則後左輪參數式$(X_{21}(t),Y_{21}(t))$可表示為：

$$X_{21}(t)=X_{22}(t)-C_W(\frac{{Y_{22}}'(t)}{\sqrt{{X_{22}}'(t)^2+{Y_{22}}'(t)^2}})$$

$$Y_{21}(t)=Y_{22}(t)+C_W(\frac{{X_{22}}'(t)}{\sqrt{{X_{22}}'(t)^2+{Y_{22}}'(t)^2}})$$

若車輛向左行駛$X_F(t)-X_R(t)<0$，則後左輪參數式$(X_{21}(t),Y_{21}(t))$可表示為：

$$X_{21}(t)=X_{22}(t)-C_W(\frac{{Y_{22}}'(t)}{\sqrt{{X_{22}}'(t)^2+{Y_{22}}'(t)^2}})$$

$$Y_{21}(t)=Y_{22}(t)+C_W(\frac{{X_{22}}'(t)}{\sqrt{{X_{22}}'(t)^2+{Y_{22}}'(t)^2}})$$

當後右輪軌跡為一圓，已知C_W、$(X_{22}(t), Y_{22}(t)) = (r \cdot \cos t, r \cdot \sin t)$，則若車輛逆時針行駛，則後左輪參數式$(X_{21}(t), Y_{21}(t))$可表示為：

$$X_{21}(t) = (r - C_W) \cdot \cos t$$
$$Y_{21}(t) = (r - C_W) \cdot \sin t$$

若車輛順時針行駛，則後左輪參數式$(X_{21}(t), Y_{21}(t))$可表示為：

$$X_{21}(t) = (r + C_W) \cdot \cos t$$
$$Y_{21}(t) = (r + C_W) \cdot \sin t$$

二、應用：

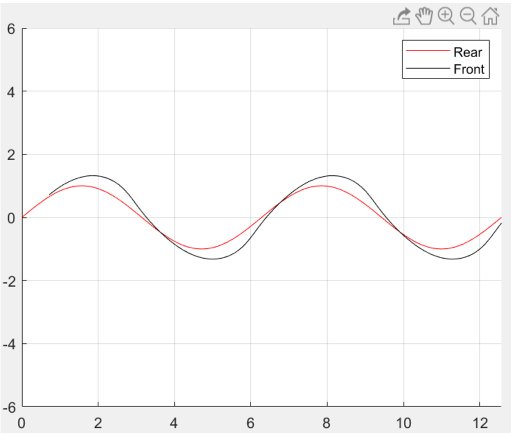
(一)實際辨析：

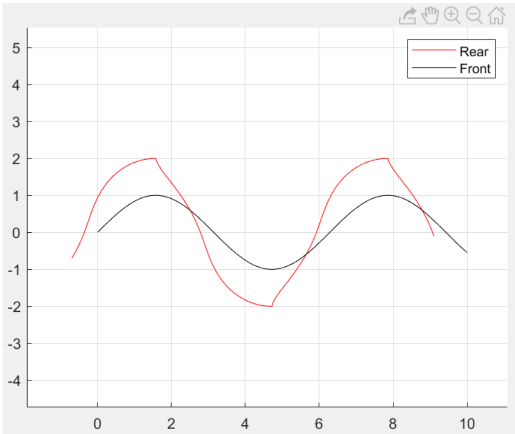
從一組胎痕中能分辨四輪胎痕、軸距、輪距及行車方向，在刑事鑑定上便可發揮作用，如：以軸距、輪距去推斷逃逸車型，以行進方向推知逃逸方向等，皆是實際辨析的應用。

(二)以 MATLAB 程式化：

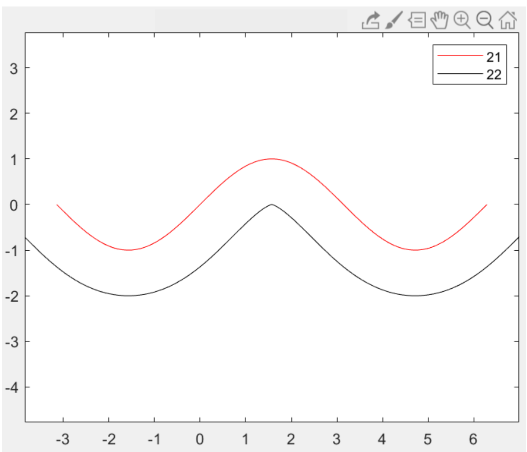
在建模時，GeoGeBra 只是繪出函數圖形，並不是真正的電腦模擬；然而若能將公式程式化，只要輸入任意函數或代入多個座標，便能以數值微分的方式繪出前輪軌跡之程式；另外，一般情況中，車輪留下的軌跡常不為一特定函數，因此我選用數值近似而不是解析式，可以用選取多點的方式，將複雜的後輪函數改成數值微分，在實際運用時將較為實際，下頁圖皆以 sin 函數為例，可發現與 GeoGeBra 所繪一致，程式碼寫於附錄一。

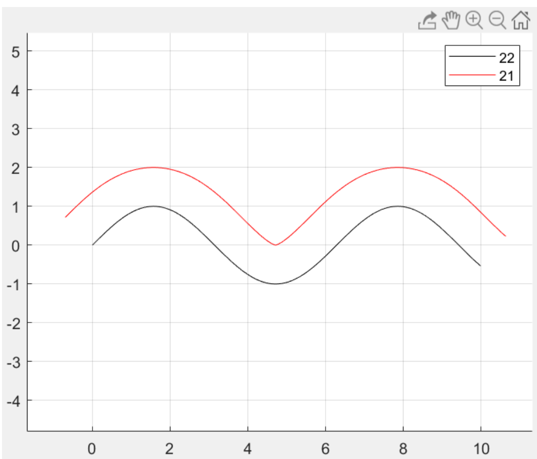
後推前、前推後：





後左推後右、後右推後左：





三、未來計畫(目前進度)：

(一)利用 Arduino 做出程式化的車輛以驗證定理：

以後推前為例，我們目的是做出可程式化的車輛讓後輪跑出特定軌跡，以驗證前輪軌跡是否如同定理 1.1 所述，因此我們找來了使用 Arduino IDE 的遙控車程式自行修改；然而自製車的輸出是$v(t)$、$\phi(t)$：$v(t)$將位置函數微分一次即可求，而$\phi(t)$則須利用定理 2.1 之證明中，$\tan\phi = \tan(\theta_R - \theta_F)$來求得，利用 MATLAB 便可得到$v(t)$及 $\phi(t)$，Arduino IDE 及 MATLAB 的程式碼寫於附錄二。

(二)前推後給定ϕ_0之討論：

若給定ϕ_0，便等同與解一個完整的微分方程，但此微分方程過於複雜，解不出解析式，因此我曾在 MATLAB 用前尤拉法解方程組，希望在給定ϕ_0的情況下解出無限逼近的數值解，但由於其運算速度慢及誤差過大導致沒有成功解出，原程式碼寫於附錄三。

(三)前左推前右及前右推前左給定ϕ_0、C_L之討論：

理由同(二)。

(四)推導出內輪差公式：

若成功推導出完整的四輪軌跡關係式，便能用積分求面積的方式推導出內輪差公式，如此一來便能找到讓內輪差為最小的過彎方式，在未來無人駕駛盛行的時代，或許能程式化後發展出其應用性。

捌、參考資料及其他

一、洪維恩 著‧2013 年 9 月 20 號‧Matlab 程式設計 第 2 版‧旗標出版社。

二、取自 http://z0fu6.blogspot.com/2011/07/blog-post.html

三、原線上 Arduino IDE 程式：

```
#include <Servo.h>

const int led_head_1 =A0;//頭燈腳位

const int led_head_2 =A1;//頭燈腳位

const int led_back_1 =A2;//倒車燈腳位

const int led_back_2 =A3;//倒車燈腳位

const int led_stop_1 =A4;//煞車燈腳位

const int led_stop_2 =A5;//煞車燈腳位

const int moto_control=13;//控制前進後退腳位

const int moto_pwm=11;//控制馬達調速腳位

const int servo_pin=9;//伺服馬達腳位
```

```
const int servo_setup=94;//開機馬達角度
const int servo_min=55;//伺服馬達最小角度
const int servo_max=125;//伺服馬達最大角度
const char forward='f';//前進指令
const char backward='b';//後退指令
const char led_open='o';//開燈指令
const char led_close='c';//關燈指令
const char data_end='$';//結束字元
char    c;//接收字元
String    data;//接收資料
String    angle;//角度
String    speed;//速度
int    speed_data;//速度資料
int    speed_old;//舊速度
Servo servo_9;


void setup()
{
   Serial.begin(57600);
   servo_9.attach(servo_pin);
   pinMode(led_head_1, OUTPUT);//設定腳位
   pinMode(led_head_2, OUTPUT);
   pinMode(led_back_1, OUTPUT);
   pinMode(led_back_2, OUTPUT);
   pinMode(led_stop_1, OUTPUT);
   pinMode(led_stop_2, OUTPUT);
```

```
    pinMode(moto_control, OUTPUT);

    pinMode(moto_pwm, OUTPUT);

    servo_9.write(servo_setup);//伺服馬達轉向

}



void loop()

{

        if (Serial.available() > 0) {//藍芽是否有值傳送

            c = Serial.read();//將傳送的值給 c

            if (c == forward) {//前進

                digitalWrite(13,HIGH);//馬達正轉

                digitalWrite(led_back_1,LOW);//關後退燈

                digitalWrite(led_back_2,LOW);

                data = "";//清空資料

            } else if (c == backward) {//後退

                digitalWrite(13,LOW);//馬達反轉

                digitalWrite(led_back_1,HIGH);//開後退燈

                digitalWrite(led_back_2,HIGH);

                data = "";//清空資料

            } else if (c == led_open) {

                digitalWrite(led_head_1,HIGH);//開頭燈

                digitalWrite(led_head_2,HIGH);

                data = "";//清空資料

            } else if (c == led_close) {

                digitalWrite(led_head_1,LOW);//關頭燈
```

```
            digitalWrite(led_head_2,LOW);
            data = "";//清空資料
          } else if (c == ',') {
            speed = data;
            data = "";//清空資料
            speed_data = (map(speed.toInt(),0,100,0,255));//將 0-100 對應到 0-255
            analogWrite(11,speed_data);
            if (speed_data < speed_old) {//速度減速
              digitalWrite(led_stop_1,HIGH);//開煞車燈
              digitalWrite(led_stop_2,HIGH);
            } else {
              digitalWrite(led_stop_1,LOW);//關煞車燈
              digitalWrite(led_stop_2,LOW);
            }
            speed_old = speed_data;
            data = "";//清空資料
          } else if (c == data_end) {//結束封包
            angle = data;
            data = "";//清空資料
            servo_9.write((map(angle.toInt(),0,180,servo_min,servo_max)));//伺服馬達轉向
          } else {
            data += String(c);//將 c 的資料一直累積
          }
        }
    }
```

附錄

一、MATLAB 電腦模擬。

(一)後推前：

1. 一般軌跡公式：

```
CL = 1;          %input
N = 1000;        %input
t0 = 0;          %input
t1 = 4*pi;       %input
dt = (t1- t0)/N;
t = linspace(t0,t1,N+1);

% initialization of position of rear wheel
XR = t;          %input
YR = sin(t);     %input
XR1 = 0*t;       %let XR1=XR'
XR1(1) = 0;
for i=2:N+1
    XR1(i) = (XR(i)-XR(i-1))/dt;
end

YR1 = 0*t;       %let YR1=YR'
YR1(1) = 0;
for i=2:N+1
    YR1(i) = (YR(i)-YR(i-1))/dt;
end

XF = XR + CL.*XR1./sqrt(XR1.^2+YR1.^2);
YF = YR + CL.*YR1./sqrt(XR1.^2+YR1.^2);

curveRear = animatedline('Color', 'r');
curveFront = animatedline('Color', 'k');

grid on;
axis([t0 t1 -6 6])

for i=1:N+1
    addpoints(curveRear, XR(i), YR(i));
```

```
    drawnow limitrate
    addpoints(curveFront, XF(i), YF(i));
    drawnow limitrate
    legend('Rear', 'Front');
end
```

2. 圓形軌跡公式：

```
CL = 4;
r = 3;
N = 1000;
t0 = 0.5;
t1 = 3*pi/2;
dt = (t1-t0)/N;
t = linspace(t0,t1,(t1-t0)/dt+1);

XR = r*cos(t);
YR = r*sin(t); % input

% generating of first derivatives on X by Euler backward method
XR1 = 0*t;   %let XR1=XR'
XR1(1) = 0;

for i=2:N+1
    XR1(i) = (XR(i) - XR(i-1))/dt;
end
% generating of second derivatives on X by second-order backward method
XR2 = 0*t;
XR2(1) = 0;
XR2(2) = 0;

for i=3:(t1-t0)/dt+1
    %XR2(i) = (XR(i) - 2*XR(i-1) + XR(i-2))/(dt^2);
    XR2(i) = (XR1(i) - XR1(i-1))/dt;
end


YR1 = 0*t;    %let YR1=YR'
YR1(1) = 0;
for i=2:(t1-t0)/dt+1
```

```
    YR1(i) = (YR(i)-YR(i-1))/dt;
end

YR2 = 0*t;
YR2(1) = 0;
YR2(2) = 0;
for i=3:(t1-t0)/dt+1
    %YR2(i) = (YR(i) - 2*YR(i-1) + YR(i-2))/(dt^2);
    YR2(i) = (YR1(i) - YR1(i-1))/dt;
end

% generating of velocities and accelerations
v3D = zeros(3,N+1);
a3D = zeros(3,N+1);
b3D = zeros(3,N+1); % binomial in Frenet frame
Dir = strings(1,N+1);
k = [0,0,1];

for i=1:N+1
    v3D(:,i) = [XR1(i), YR1(i), 0]';
    a3D(:,i) = [XR2(i), YR2(i), 0]';
    b3D(:,i) = cross(100*v3D(:,i)', 100*a3D(:,i)');
    if (dot(b3D(:,i), k)>0)
        Dir(i) = "left";
        XF(i) = r.*cos(t(i)) - CL.*sin(t(i));
        YF(i) = r.*sin(t(i)) + CL.*cos(t(i));
    else
        Dir(i) = "right";
        XF(i) = r.*cos(-t(i)) + CL.*sin(-t(i));
        YF(i) = r.*sin(-t(i)) - CL.*cos(-t(i));
    end
end

curveRear = animatedline('Color', 'k');
curveFront = animatedline('Color', 'r');

grid on;
axis([-10 10 -10 10]);
```

```
axis equal;
for i=3:N+1
    addpoints(curveFront, XF(i), YF(i));
    drawnow limitrate

    %hold on
    addpoints(curveRear, XR(i), YR(i));
    drawnow limitrate
    %hold on
    legend('Front', 'Rear');
    %pause(0.1)
end
```

(二)前推後：

1. 一般軌跡公式：

```
CL = 1;        %input
t0 = 0;        %input
t1 = 10;       %input
N = 1000;
dt = (t1 - t0)/N;
t = linspace(t0,t1,(t1-t0)/dt+1);

XF = t;        %input
YF = sin(t);   %input

YF1 = 0*t;
YF1(1) = 0;

YF2 = 0*t;
YF2(1) = 0;
YF2(2) = 0;

for i = 2:N+1
    YF1(i) = ( YF(i) - YF(i-1) )/dt;
end
for i = 3:N+1
    YF2(i) = ( YF1(i) - YF1(i-1) )/dt;
end
```

```
M = (YF1+((CL.*YF2)./sqrt((1+YF1.^2).^3-CL^2.*YF2.^2)))./(1-
((CL.*YF2)./sqrt((1+YF1.^2).^3-CL^2*YF2.^2)).*YF1);
XR = XF - CL./sqrt(1+M.^2);
YR = YF - (CL.*M)./sqrt(1+M.^2);

curveRear = animatedline('Color', 'r');
curveFront = animatedline('Color', 'k');

grid on;
axis([-10 10 -10 10]);
axis equal;
legend('Rear', 'Front');
for i=2:N+1
    addpoints(curveFront, XF(i), YF(i));
    drawnow limitrate
    addpoints(curveRear, XR(i), YR(i));
    drawnow limitrate
end
```

2. 圓形軌跡公式：

```
CL = 3;
r = 5;
N = 1000;
t0 = 0;
t1 = 4*pi/3;
dt = (t1-t0)/N;
t = linspace(t0,t1,(t1-t0)/dt+1);

XF = r*cos(-t);
YF = r*sin(-t);

% generating of first derivatives on X by Euler backward method
XF1 = 0*t;   %let XR1=XR'
XF1(1) = 0;
for i=2:(t1-t0)/dt+1
    XF1(i) = (XF(i)-XF(i-1))/dt;
end
% generating of second derivatives on X by second-order backward method
XF2 = 0*t;
```

```
XF2(1) = 0;
XF2(2) = 0;

for i=3:(t1-t0)/dt+1
    %XF2(i) = (XF(i) - 2*XF(i-1) + XF(i-2))/(dt^2);
    XF2(i) = (XF1(i)-XF1(i-1))/dt;
end

YF1 = 0*t;    %let YR1=YR'
YF1(1) = 0;
for i=2:(t1-t0)/dt+1
    YF1(i) = (YF(i)-YF(i-1))/dt;
end

YF2 = 0*t;
YF2(1) = 0;
YF2(2) = 0;
for i=3:(t1-t0)/dt+1
    %YF2(i) = (YF(i) - 2*YF(i-1) + YF(i-2))/(dt^2);
    YF2(i) = (YF1(i)-YF1(i-1))/dt;
end

% generating of velocities and accelerations
v3D = zeros(3,N+1);
a3D = zeros(3,N+1);
b3D = zeros(3,N+1); % binomial in Frenet frame
Dir = strings(1,N+1);
k = [0,0,1];

% making of XR and YR
for i=1:N+1
    v3D(:,i) = [XF1(i), YF1(i), 0]';
    a3D(:,i) = [XF2(i), YF2(i), 0]';
    b3D(:,i) = cross(100*v3D(:,i)', 100*a3D(:,i)');
    if (dot(b3D(:,i), k)>0)
        Dir(i) = "left";
        XR(i) = ((r.^2 - CL.^2)./r).*cos(t(i)) + (CL.*sqrt(r.^2 -
CL.^2)./r).*sin(t(i));
```

```
        YR(i) = ((r.^2 - CL.^2)./r).*sin(t(i)) - (CL.*sqrt(r.^2 -
CL.^2)./r).*cos(t(i));
    else
        Dir(i) = "right";
        XR(i) = ((r.^2 - CL.^2)./r).*cos(-t(i)) - (CL.*sqrt(r.^2 -
CL.^2)./r).*sin(-t(i));
        YR(i) = ((r.^2 - CL.^2)./r).*sin(-t(i)) + (CL.*sqrt(r.^2 -
CL.^2)./r).*cos(-t(i));
    end
end

curveRear = animatedline('Color', 'r');
curveFront = animatedline('Color', 'k');

grid on;
axis([-10 10 -10 10]);
axis equal;
for i=3:(t1-t0)/dt
    addpoints(curveFront, XF(i), YF(i));
    drawnow limitrate

    %hold on
    addpoints(curveRear, XR(i), YR(i));
    drawnow limitrate
    %hold on
    legend('Front', 'Rear');
    %pause(0.1)
end
```

(三)後左推後右：

1. 一般軌跡公式：

```
CW = 1;
r = 5;
N = 1000;
t0 = -pi;
t1 = 2*pi;
dt = (t1-t0)/N;
t = linspace(t0,t1,N+1);
```

```
X21 = t; %input
Y21 = sin(t); %input

X21_1 = 0*t;  %let X21_1=X21'
X21_1(1) = 0;

for i=2:N+1
    X21_1(i) = (X21(i) - X21(i-1))/dt;
end

Y21_1 = 0*t;  %let Y21_1=Y21'
Y21_1(1) = 0;
for i=2:N+1
    Y21_1(i) = (Y21(i) - Y21(i-1))/dt;
end

X22 = X21 + CW.*Y21_1./sqrt(X21_1.^2+Y21_1.^2);
Y22 = Y21 - CW.*X21_1./sqrt(X21_1.^2+Y21_1.^2);

curve21 = animatedline('Color', 'k');
curve22 = animatedline('Color', 'r');
grid on;
axis([-10 10 -10 10]);
axis equal;
legend('21', '22');

for i=3:N+1
    addpoints(curve21, X21(i), Y21(i));
    drawnow limitrate

    %hold on
    addpoints(curve22, X22(i), Y22(i));
    drawnow limitrate
    %hold on
    %legend('Front', 'Rear');
    %pause(0.1)
end
```

```
plot(X21,Y21,'r','DisplayName','21');
hold on
plot(X22,Y22,'k','DisplayName','22');
title('from 21 to 22');
xlabel('x');
ylabel('y');
axis equal;
legend('show');
```

2. 圓形軌跡公式：

```
CW = 3;
r = 5;
N = 1000;
t0 = 0;
t1 = 3*pi/5;
dt = (t1-t0)/N;
t = linspace(t0,t1,(t1-t0)/dt+1);

Y21 = r*cos(t); %input
X21 = r*sin(t); %input

X21_1 = 0*t;  %let X21_1=X21'
X21_1(1) = 0;

for i=2:(t1-t0)/dt
X21_1(i) = (X21(i)-X21(i-1))/dt;
end

Y21_1 = 0*t;  %let Y21_1=Y21'
Y21_1(1) = 0;
for i=2:(t1-t0)/dt
Y21_1(i) = (Y21(i)-Y21(i-1))/dt;
end

X22 = X21 + CW.*Y21_1./sqrt(X21_1.^2+Y21_1.^2);
Y22 = Y21 - CW.*X21_1./sqrt(X21_1.^2+Y21_1.^2);

curveRear = animatedline('Color', 'r');
curveFront = animatedline('Color', 'k');
```

```
grid on;
axis([-10 10 -10 10]);
axis equal;
legend('21', '22');

for i=1:N+1
    addpoints(curveFront, X21(i), Y21(i));
    drawnow limitrate
    addpoints(curveRear, X22(i), Y22(i));
    drawnow limitrate
end
legend('show');
```

(四)後右推後左：

1. 一般軌跡公式：

```
CW=1;       %input
N = 1000;   %input
t0=0;       %input
t1=10;      %input
dt = (t1-t0)/N;
t=linspace(t0,t1,N+1);

X22 = t;
Y22 = sin(t); %input
X22_1 = 0*t;  %let X22_1=X22'
X22_1(1) = 0;
for i=2:N+1
   X22_1(i) = (X22(i) - X22(i-1))/dt;
end
Y22_1 = 0*t;  %let Y22_1=Y22'
Y22_1(1) = 0;
for i=2:N+1
   Y22_1(i) = (Y22(i) - Y22(i-1))/dt;
end
X21 = X22 - CW.*Y22_1./ sqrt(X22_1.^2 + Y22_1.^2);
Y21 = Y22 + CW.*X22_1./ sqrt(X22_1.^2 + Y22_1.^2);
```

```
curve22 = animatedline('Color', 'k');
curve21 = animatedline('Color', 'r');
grid on;
axis([-10 10 -10 10]);
axis equal;
legend('22', '21');

for i=1:(t1-t0)/dt
    addpoints(curve22, X22(i), Y22(i));
    drawnow

    addpoints(curve21, X21(i), Y21(i));
    drawnow

end
```

2. 圓形軌跡公式：

```
CW = 1;        %input
t0 = 0;        %input
t1 = 10;       %input
r = 3;
N = 1000;
dt = (t1 -t0)/N;
t = linspace(t0,t1,N+1);

Y22 = r*cos(t);      %input
X22 = r*sin(t);      %input
X22_1 = 0*t;  %let X22_1=X22'
X22_1(1) = 0;
for i=2:(t1-t0)/dt
    X22_1(i) = (X22(i) - X22(i-1))/dt;
end
Y22_1 = 0*t;  %let Y22_1=Y22'
Y22_1(1) = 0;
for i=2:(t1-t0)/dt
   Y22_1(i) = (Y22(i) - Y22(i-1))/dt;
end
X21 = X22 - CW.*Y22_1./ sqrt(X22_1.^2 + Y22_1.^2);
Y21 = Y22 + CW.*X22_1./ sqrt(X22_1.^2 + Y22_1.^2);
```

```
curve22 = animatedline('Color', 'k');
curve21 = animatedline('Color', 'r');
grid on;
axis([-10 10 -10 10]);
axis equal;
legend('Right', 'Left');
title('From Right to Left');

for i=1:N+1
    addpoints(curve21, X21(i), Y21(i));
    drawnow

    addpoints(curve22, X22(i), Y22(i));
    drawnow
end
```

二、自走車程式。

(一)修改後 Arduino IDE：

```
#include <Servo.h>
const int led_head_1 =A0;//頭燈腳位
const int led_head_2 =A1;//頭燈腳位
const int led_back_1 =A2;//倒車燈腳位
const int led_back_2 =A3;//倒車燈腳位
const int led_stop_1 =A4;//煞車燈腳位
const int led_stop_2 =A5;//煞車燈腳位
const int moto_control=13;//控制前進後退腳位
const int moto_pwm=11;//控制馬達調速腳位
const int servo_pin=9;//伺服馬達腳位
const int servo_setup=100;//開機馬達角度
const int servo_min=55;//伺服馬達最小角度
const int servo_max=125;//伺服馬達最大角度
const char forward='f';//前進指令
const char backward='b';//後退指令
const char led_open='o';//開燈指令
const char led_close='c';//關燈指令
const char data_end='$';//結束字元
const int myspeed=0;//設定速度量值
char    c;//接收字元
```

```
String   data;//接收資料
String   angle;//角度
String   speed;//速度
int   speed_data;//速度資料
int   speed_old;//舊速度
Servo servo_9;

void setup()
{
  Serial.begin(57600);
  servo_9.attach(servo_pin);
  pinMode(led_head_1, OUTPUT);//設定腳位
  pinMode(led_head_2, OUTPUT);
  pinMode(led_back_1, OUTPUT);
  pinMode(led_back_2, OUTPUT);
  pinMode(led_stop_1, OUTPUT);
  pinMode(led_stop_2, OUTPUT);
  pinMode(moto_control, OUTPUT);
  pinMode(moto_pwm, OUTPUT);
  servo_9.write(servo_setup);//伺服馬達轉向
}
void loop()
{
        digitalWrite(13,HIGH);//馬達正轉
        speed_data = (map(myspeed,0,100,0,255));//將 0-100 對應到 0-255
        analogWrite(11,speed_data);
        servo_9.write((map(servo_setup,0,180,servo_min,servo_max)));//伺服馬達轉向

    }
```

（二）求ϕ_0之 MATLAB 程式碼。

```
XF1=0*t;    %let XF1=XF'
XF1(1)=0;
for i=2:(t1-t0)/dt
XF1(i)=(XF(i)-XF(i-1))/dt;
end
YF1=0*t;    %let YF1=YF'
YF1(1)=0;
for i=2:(t1-t0)/dt
```

```
YF1(i)=(YF(i)-YF(i-1))/dt;
end
tanR=XR1./YR1;
tanF=XF1./YF1;
phi=(tanR-tanF)./(1.+tanR.*tanF);
phid=atand(phi);
plot(phid)
```

三、前尤拉法解之 MATLAB 程式碼。

```
function [XR, YR] = NewR0(CL, N, i)

%CL = 1;
phi0 = pi/2;  %in radians
%N = 1000;
t0 = 0;
t1 = 4*pi;
dt = (t1-t0)/N;
t=linspace(t0,t1,N+1);

XF=t;
YF=sin(t);
XF1=(XF(i+1)-XF(i))/dt;
YF1=(YF(i+1)-YF(i))/dt;

I=[XF(1)+CL*XF1/sqrt(XF1^2+YF1^2);YF(1)+CL*YF1/sqrt(XF1^2+YF1^2)];
rot=[cos(phi0),-sin(phi0);sin(phi0),cos(phi0)];
R0=rot*I;

syms x y
sol = solve((x-XF(i))^2 + (y-YF(i))^2-CL^2 , (y-R0(2))/(YF(i)-R0(2)-(x-
R0(1))/(XF(i)-R0(1))));

for k=1:2
    if (sol.x(k,1) > R0(1) & sol.x(k,1) < R0(2))
        XR = double(sol.x(k,1));
        YR = double(sol.y(k,1));
    end
end
fprintf('XR(%d) = %f, YR(%d) = %f\n', i, XR, i, YR);
```