

國立虎尾科技大學

機械設計工程系

協同產品設計實習 ag4 期末報告

手足球對打模擬

指導教授：嚴家銘

學生：

設計二甲 40623102 吳柔燕

設計二甲 40623110 翁俊揚

設計二甲 40623111 王映捷

設計二甲 40623113 吳承哲

設計二甲 40623121 蔡朝旭

設計二甲 40623124 葉修宏

設計二甲 40623132 趙家傑

設計二甲 40623144 林昭權

設計二甲 40623152 潘季宏

設計二甲 40623157 李昀霖

2019-06-03

前言

相信大家小時候都玩過或看過類似的手足球機台，但是現在在我們周遭這種東西已經少之又少。為了重回小時候對手足球的熱情，我們試著把它用電腦模擬出來，並製作回球機構，使我們的手足球機台可以比以前的更加方便。

目錄

前言.....	2
壹、摘要.....	6
貳、設計與繪圖.....	7
2.1 零組件尺寸分析.....	7
2.2 參數設計與繪圖	14
2.3 細部設計與 BOM	17
參、V-rep 動態模擬	18
3.1 送球機構設計與模擬	18
3.2 簡化模擬.....	19
3.3 設計與模擬影片.....	23
3.4 系統功能展示.....	24
3.4.1 手足球雙人鍵盤對打	24
3.4.2 手足球手控與電腦對打	25
3.4.3 手足球電腦與電與對打.....	26
3.4.4 模擬問題與討論.....	27
3.5 問題和討論	29

圖目錄

圖 1 檯桌.....	7
圖 2 橫桿插入孔尺寸.....	7
圖 3 舊球門尺寸.....	8
圖 4 球門尺寸最終版.....	8
圖 5 橫桿尺寸.....	8
圖 6 球員尺寸.....	9
圖 7 球尺寸.....	9
圖 8 送球軌道.....	10
圖 9 送球軌道-2.....	10
圖 10 送球軌道-3.....	11
圖 11 送球軌道-4.....	11
圖 12 送球軌道-5.....	11
圖 13 送球軌道-6.....	12
圖 14 送球軌道-7.....	12
圖 15 送球軌道-8.....	12
圖 16 送球機構.....	13
圖 17 送球機構-2.....	13
圖 18 送球機構-3.....	13
圖 19 球桌.....	14
圖 20 手桿.....	14
圖 21 球員.....	15
圖 22 運球轉輪.....	15
圖 23 運球機構.....	16
圖 24 零件 BOM.....	17
圖 25 3D 零組件爆炸圖.....	17
圖 26 匯出 onshape.....	19
圖 27 匯入.STL 檔後的對話框.....	20
圖 28 分離模型步驟點選.....	20
圖 29 模型爆開.....	21
圖 30 頁面選擇器工具欄按鈕.....	21
圖 31 形狀編輯模式工具欄按鈕.....	21
圖 32 簡化桌檯對話框 -	22
圖 33 桌檯簡化完成後.....	22
圖 34 模擬問題討論-1.....	27
圖 35 模擬問題討論-2.....	27

圖 36 模擬問題討論-3.....	28
圖 37 模擬問題討論-4.....	28
圖 38 常見問題和討論-1	29
圖 39 常見問題和討論-2	29
圖 40 常見問題和討論-3	30

壹、摘要

手足球桌檯設計與繪圖

手足球 vrep 模擬

迴球、送球機構設計

迴球、送球機構模擬

Vrep 程式簡化模擬

桌檯細部設計與 BOM

貳、設計與繪圖

2.1 零組件尺寸分析

足球系統的零組件尺寸分析（可行性分析）

尺寸單位：inch(吋)

1. 檯桌尺寸（長:56 寬:30 高:5）

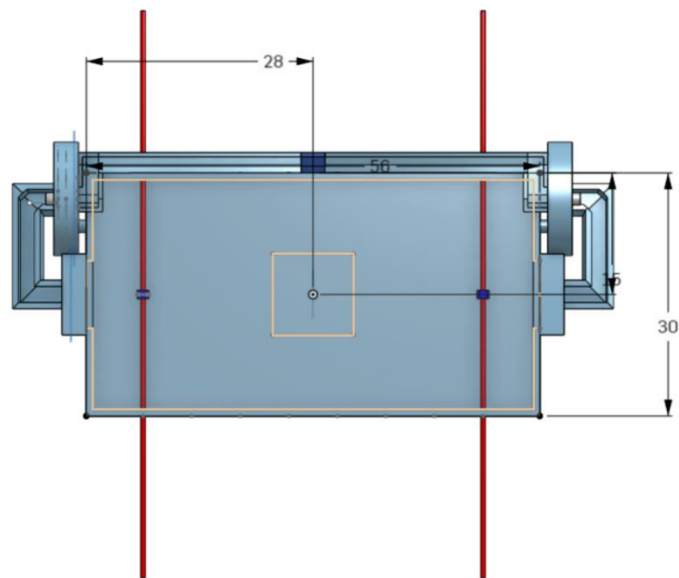


圖 1 檯桌

2. 橫桿插入孔尺寸(直徑:0.563 距底部:2.75 孔間距:6)

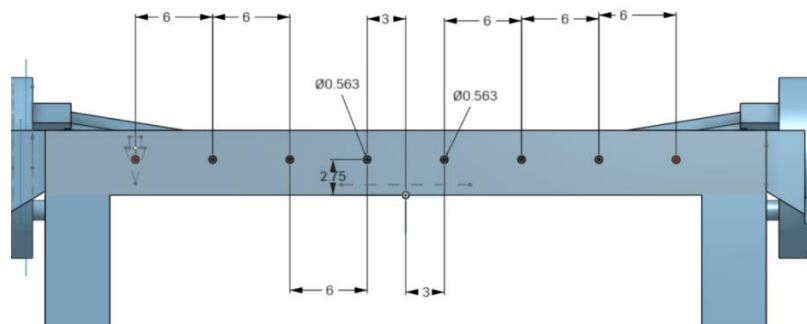


圖 2 橫桿插入孔尺寸

3. 球門尺寸(寬:4.5 長:3 距底部:0.8) (初版)

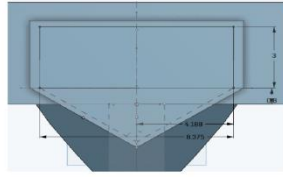


圖 3 舊球門尺寸

球門尺寸**最終版** (為了連接送球軌道與送球機構) (寬: 10)

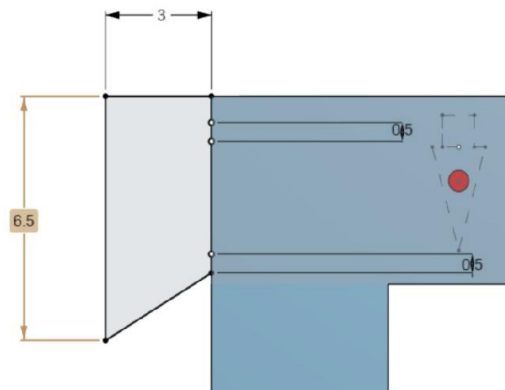


圖 4 球門尺寸最終版

4. 橫桿尺寸(長:70 直徑:0.563)

*球桿直徑與桌檯孔洞呈鬆配合，考慮到是否會讓球員在移動時無法完整移動至左右，以及使用者好操作等因素，決定將原本的 40in 改成 70in。

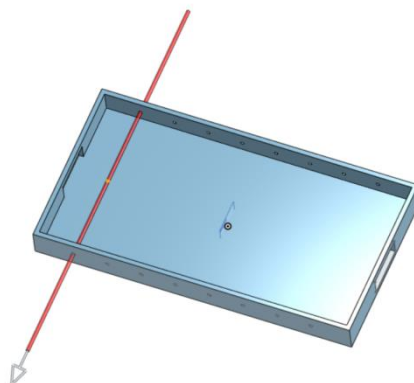


圖 5 橫桿尺寸

5. 球員尺寸(長:1.5 寬:1 高:3 圓孔:0.563 距底部:0.9)

*球員孔洞需與衡感直徑呈緊配合，必須使球員能隨橫桿轉動。本來設定球員尺寸時，為了讓球員能順利擊球而避免卡到球的問題，所以球員與底部的距離必須小於球的直徑。

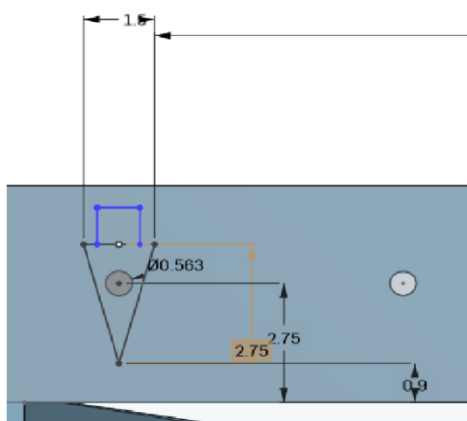


圖 6 球員尺寸

6. 球尺寸(直徑:0.15)

*球的直徑要大於球員距底板的距離，才不會發生卡球等問題。

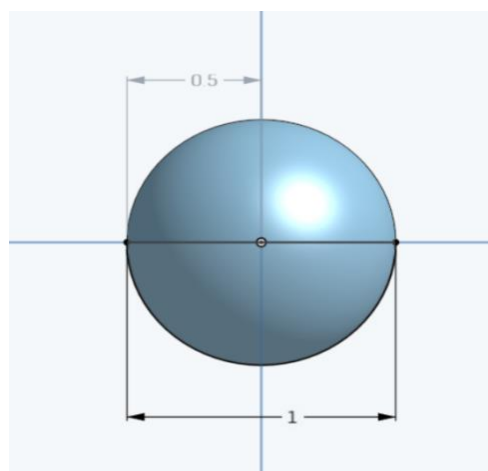


圖 7 球尺寸

7. 送球軌道

*由於尺寸需要配合送球機構以及球體尺寸的關係，軌道內側寬度設為2。

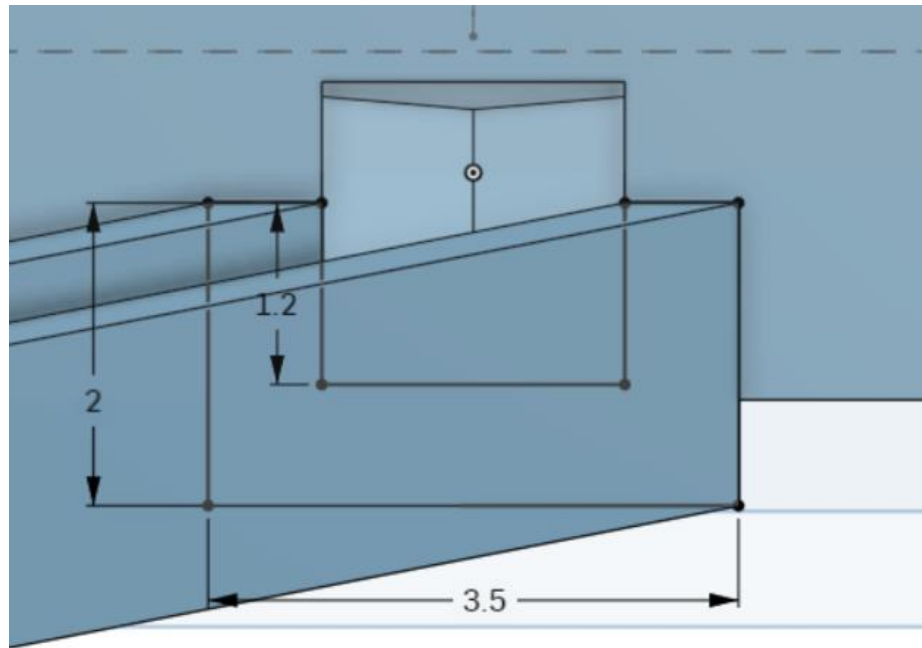


圖 8 送球軌道

*為了讓球能順利進入送球機構，球門至最靠近送球機構軌道的彎道改為倒角。

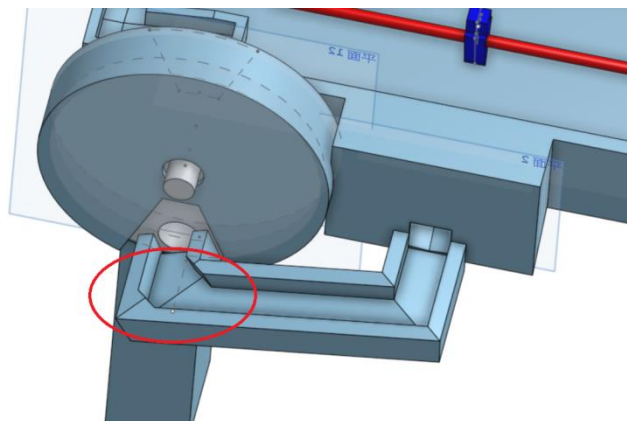


圖 9 送球軌道-2

*為了使球能不因為速度過快而彈至軌道外，而將軌道坡度減緩至10度左右。

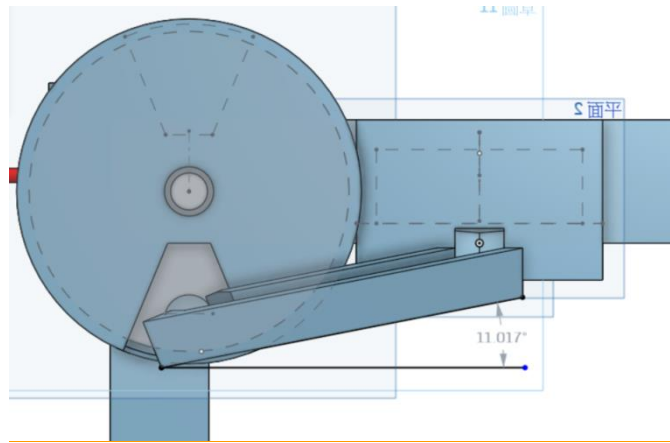


圖 10 送球軌道-3

*為了讓球能從最後的V字軌道順利進入場地，而把入口改為方形入口。

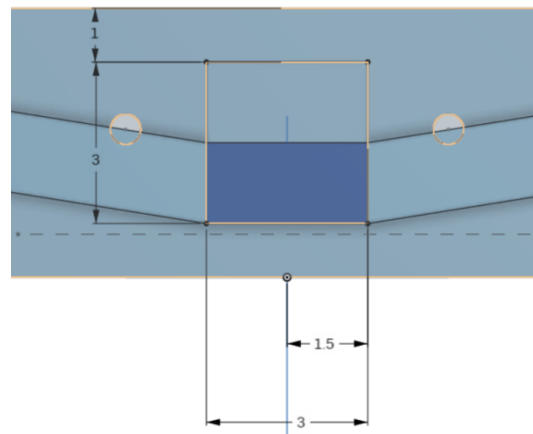


圖 11 送球軌道-4

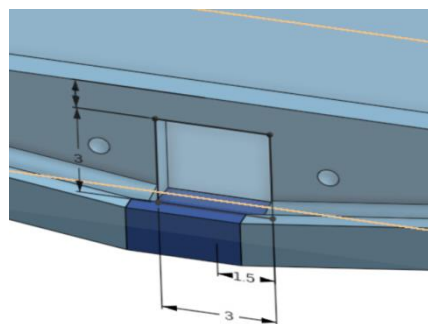


圖 12 送球軌道-5

*為了讓球能順利的進入場中，方形入口前的軌道改為 L 型斜坡。

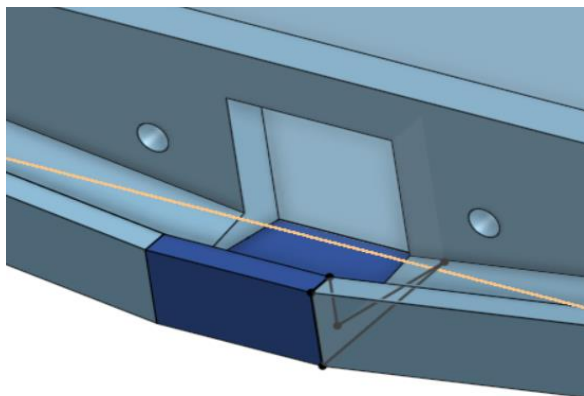


圖 13 送球軌道-6

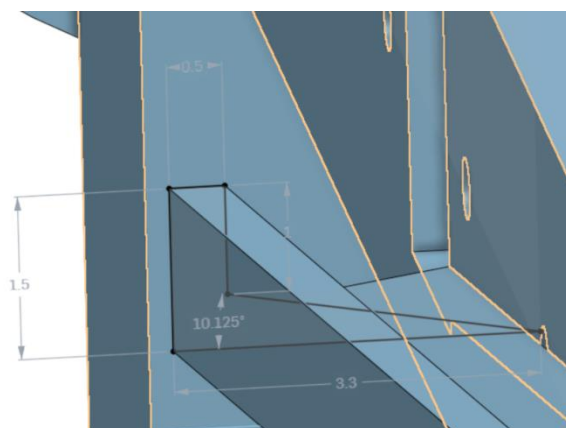


圖 14 送球軌道-7

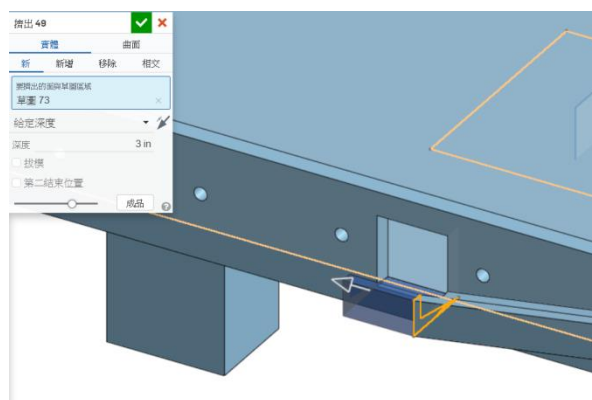


圖 15 送球軌道-8

8. 送球機構

*送球機構由圓盤和主旋轉軸組成，經由圓盤上的圓孔收入圓球，再由主傳動軸旋轉帶入至上方軌道。

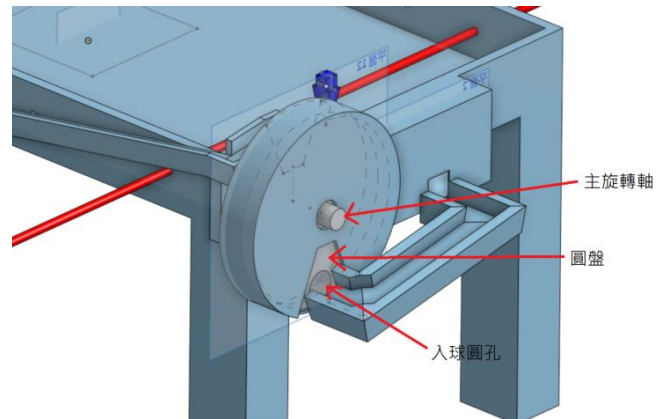


圖 16 送球機構

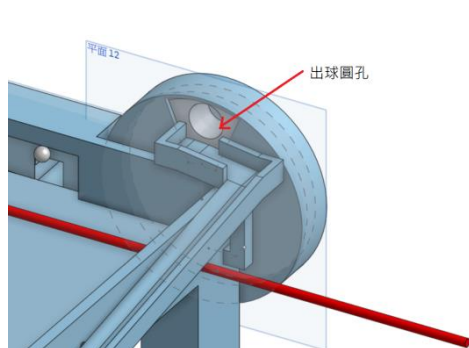


圖 17 送球機構-2

*為了讓球能順利進入軌道，入球孔大小與出球孔大小有差距，進而形成小斜坡。

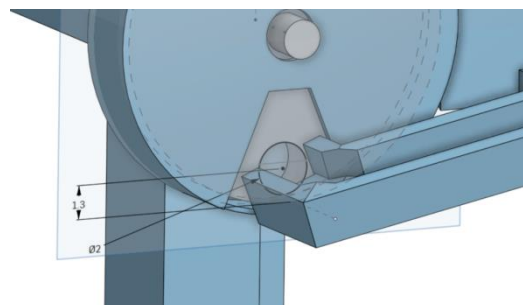


圖 18 送球機構-3

2.2 參數設計與繪圖

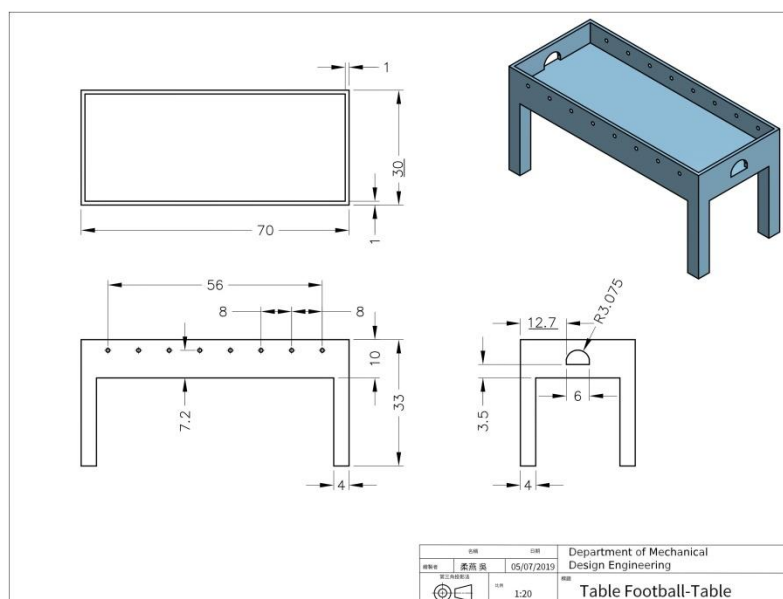


圖 19 球桌

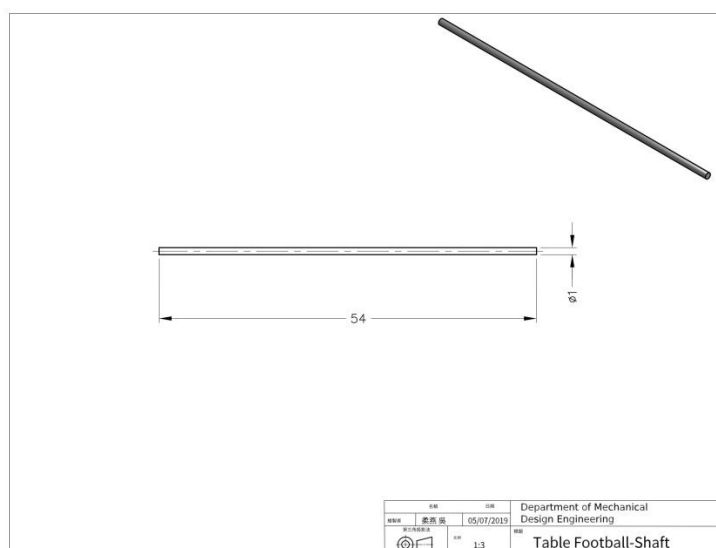


圖 20 手桿

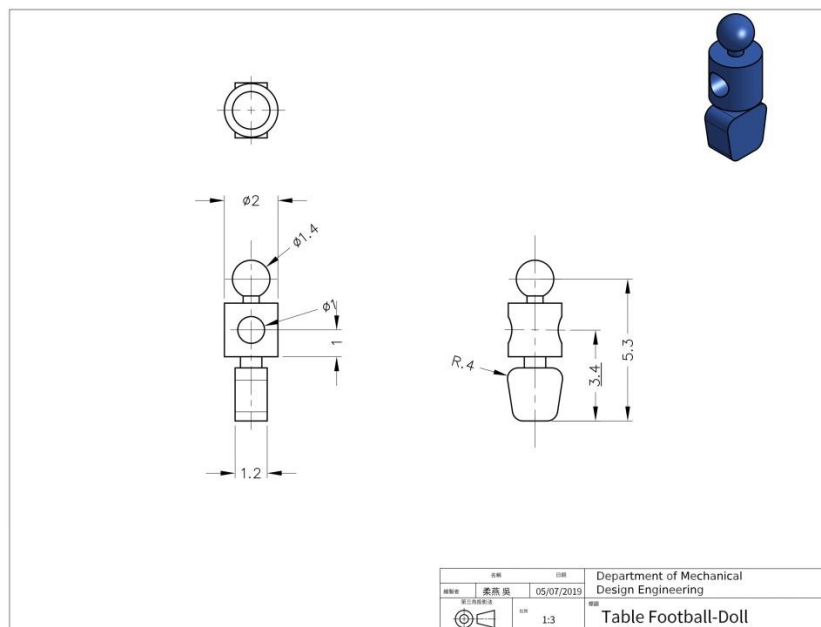


圖 21 球員

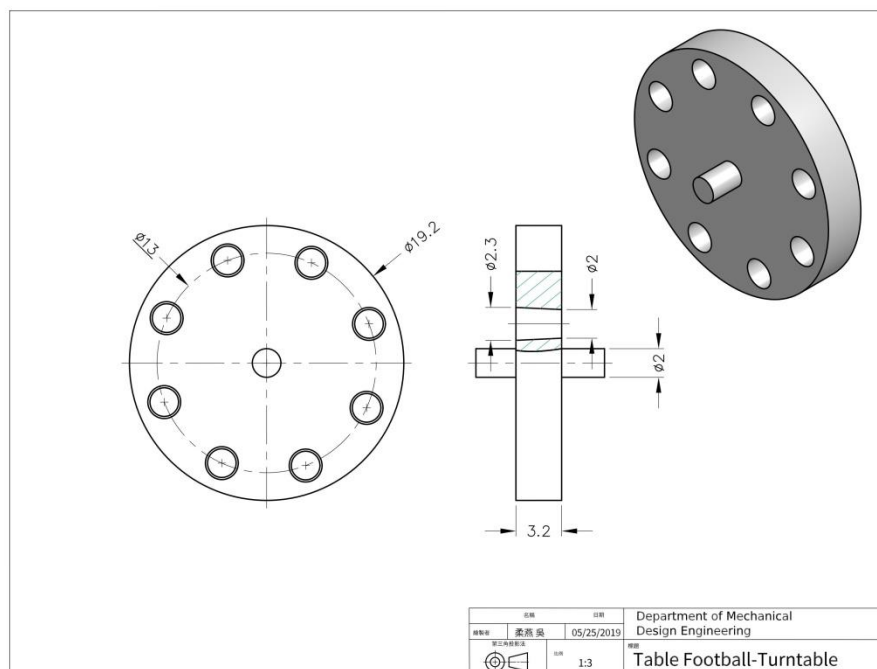


圖 22 運球轉輪

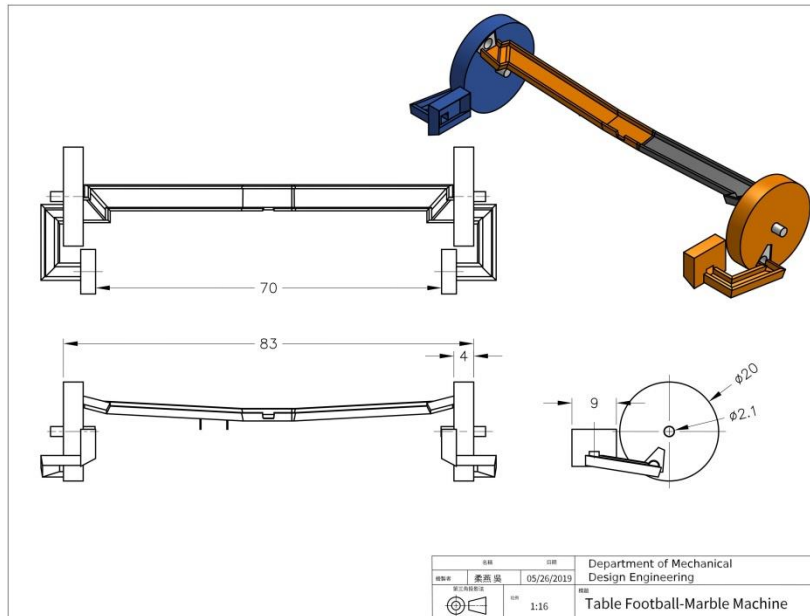


圖 23 運球機構

2.3 細部設計與 BOM

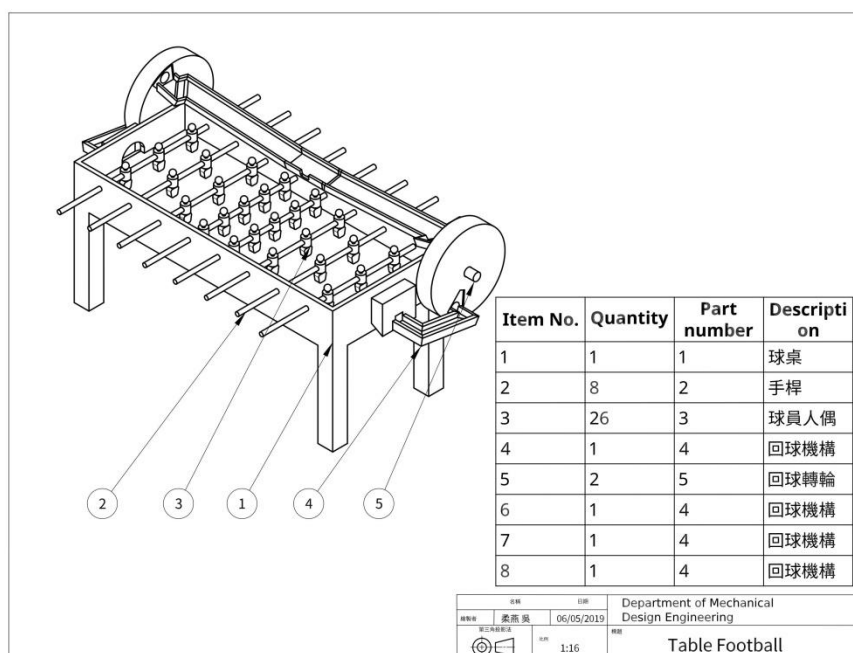


圖 24 零件 BOM

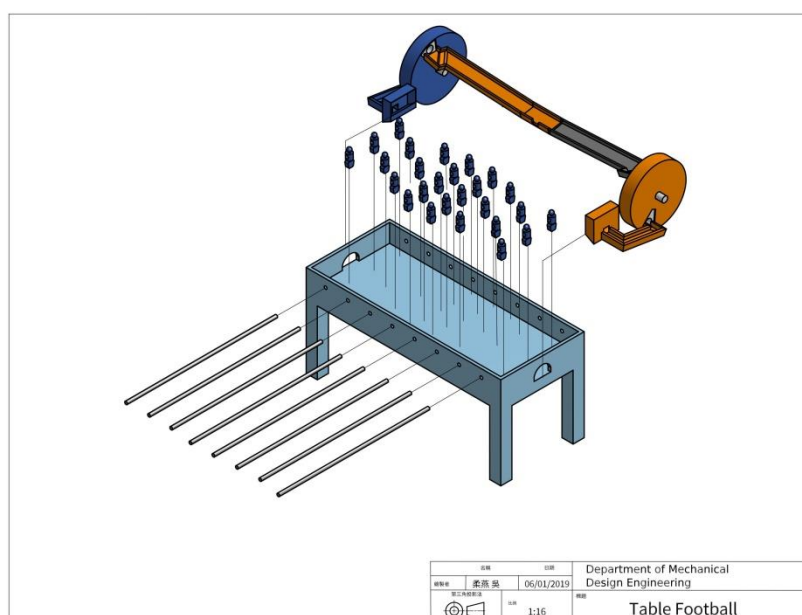


圖 25 3D 零組件爆炸圖

參、V-rep 動態模擬

3.1 送球機構設計與模擬

手足球發球與進球後自動送球機構設計與 V-rep 動態模擬（機構與傳動系統設計與模擬）

可採用螺旋機構或多連桿機構設計，透過進球感測器感應後，將儲存於螺桿上的球送回球檯。

參考網址：<https://youtu.be/IjoSLafsA2w>

<https://youtu.be/ZAGrFV1jCdk>

<https://youtu.be/UQPJ713QPd0>

3.2 簡化模擬

首先在 onshape 將想要使用的零件圖或組合圖按照圖 26 出成 .stl 檔，再 v-rep 中開啟 .stl 檔（使用 [File -> Import -> Mash...]），會出現如圖 27，依據個人所需去做點選，在按 OK 即可在視窗中導入模型。可以從圖 28 中看出，導入的模型是未分離的模型（如若是零件圖則不須此步驟），所以我們使用 [Edit -> Grouping/Merging -> Divideselectedshapes] 來將模型中的物件全都爆開，如圖 29。

接下來說明桌檯簡化步驟，先來進桌檯的簡化，（使 [Edit>CopsyselectedObjects]）到一個新建的場景（使用 [File -> Newscene]），再將人物貼上（使用 [Edit -> Paste buffer]）。再點選頁面選擇器工具欄按鈕，如圖 30，使得在簡化的過程中更容易點選，接著選取桌檯再點選形狀編輯模式工具欄按鈕來進行簡化如圖 31 在此我框選桌檯，再點選簡化的對話框 Operations on selected triangles 中的 Extract cuboid 如圖 32，之後會出現 Primitive cuboid 的對話框並按下 OK 即會產生一個立方體如圖 33，桌檯的簡化就完成了。

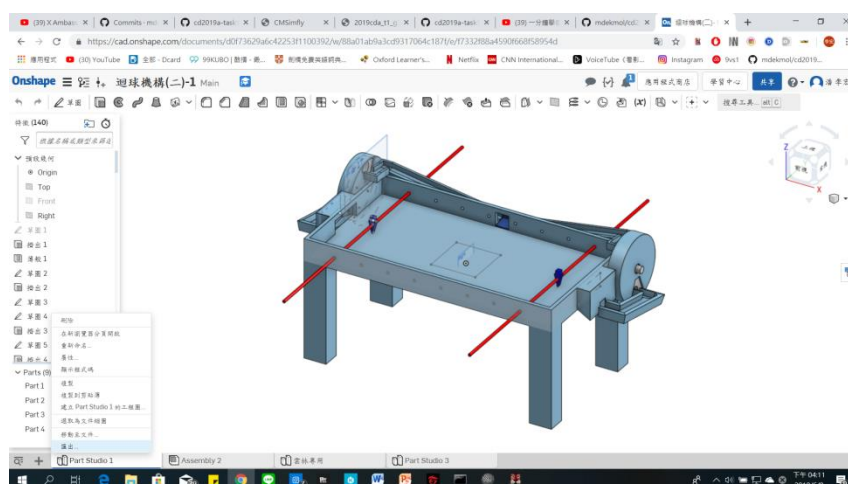


圖 26 匯出 onshape



圖 27 匯入.STL 檔後的對話框

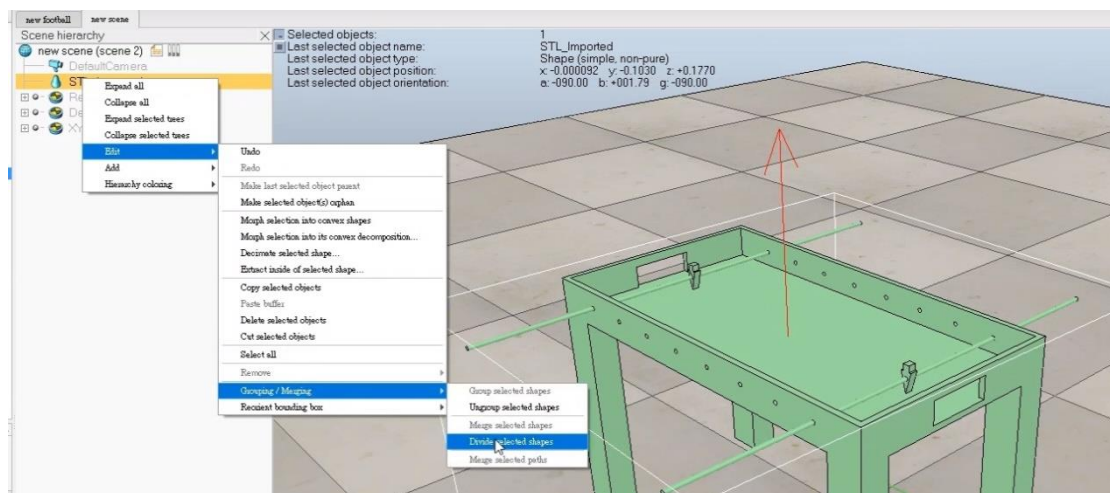


圖 28 分離模型步驟點選

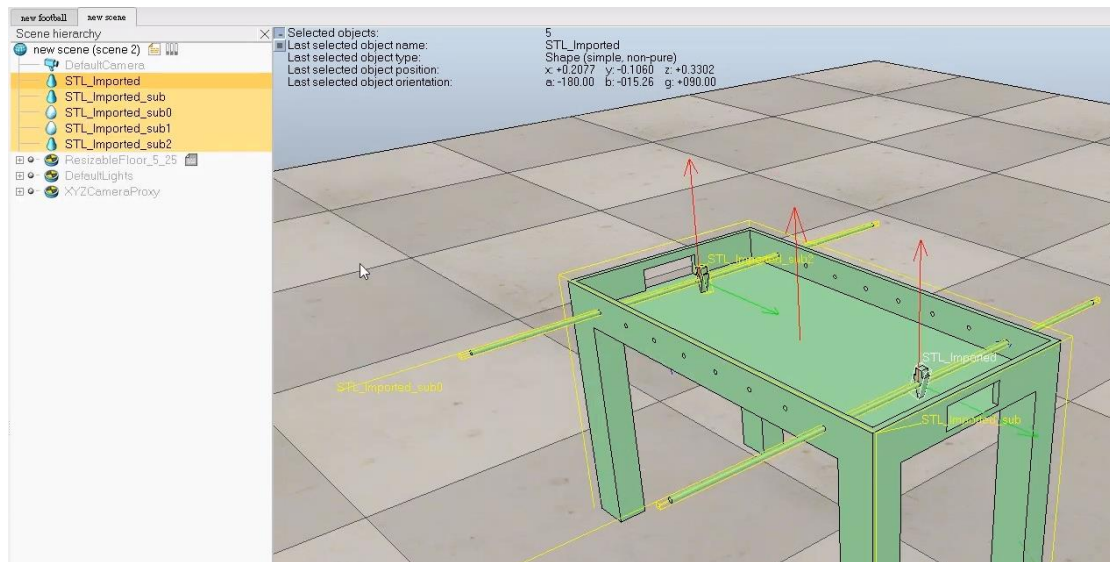


圖 29 模型爆開



圖 30 頁面選擇器工具欄按鈕



圖 31 形狀編輯模式工具欄按鈕

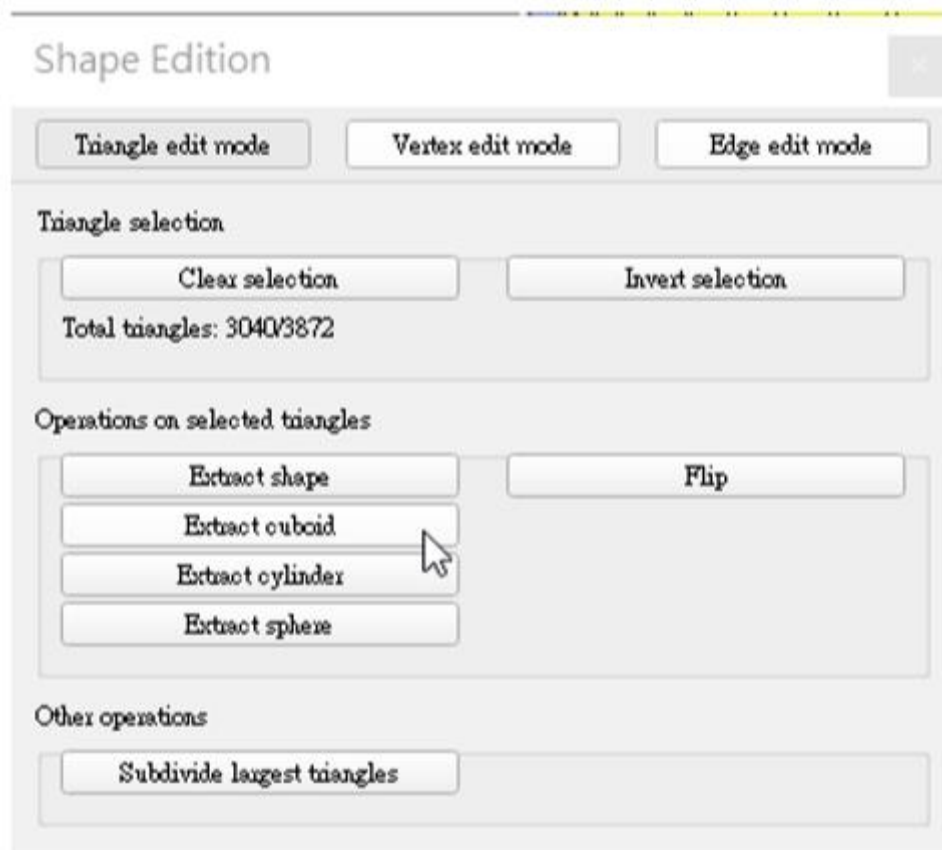


圖 32 簡化桌檯對話框 -

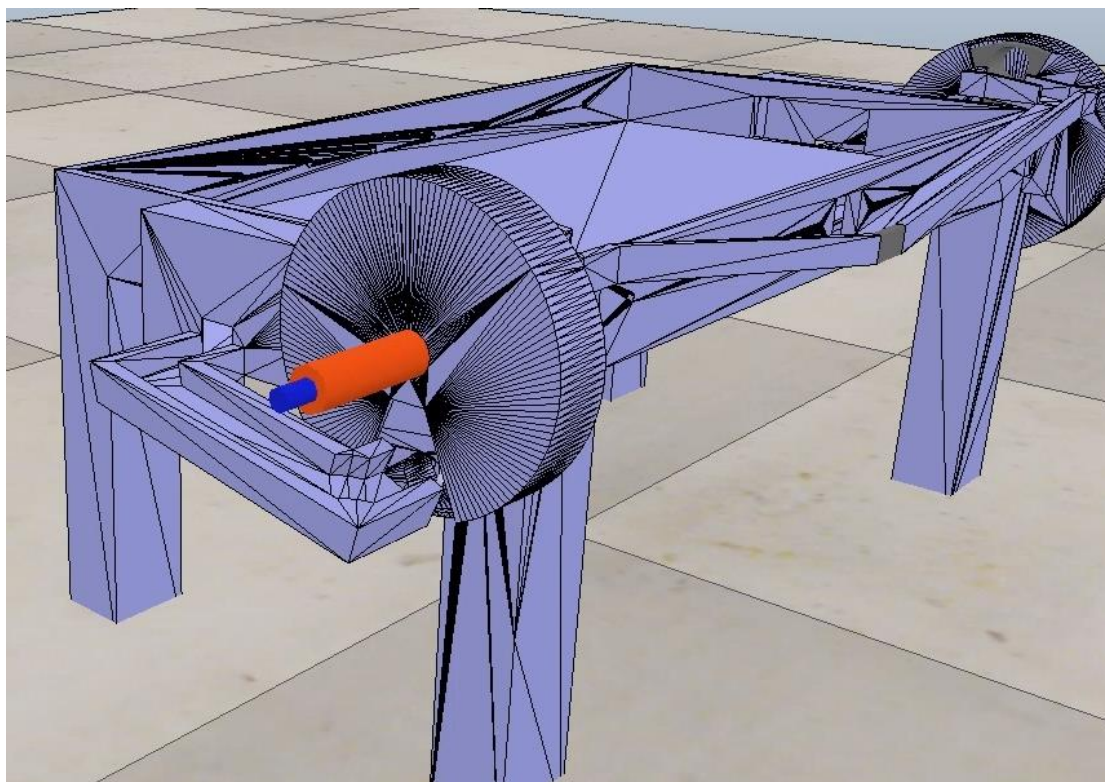


圖 33 桌檯簡化完成後

3.3 設計與模擬影片

這是當初我們討論的第一個桌檯設計圖，理解非常的簡單，就像是電梯跟自動麻將桌相似，球往桌檯中央下方進入桌檯，利用一片小平板將球送入場內。[桌檯初始設計構想影片](#)

但是因為小平板不能自動上下移動，所以我們改變了想法軌道也重新製作，這是[新的桌檯設計構想影片](#)。

構想完之後，將 onshape 匯出、爆炸、簡化，並將手把增加軸

[Vrep 基本設定](#)

再來是將[手足球迴球機構](#)、[手足球送球機構](#)加入 vrep 程式進行模擬

3.4 系統功能展示

3.4.1 手足球雙人鍵盤對打

[手足球雙人鍵盤對打 ttt 檔](#)

[對打影片網址](#)

```
import vrep
import keyboard
from time import sleep
import sys, math
vrep.simxFinish(-1)
clientID = vrep.simxStart("127.0.0.1", 19997, True, True, 5000, 5)
KickBoV = 90
n=1
L_KickBoV = (math.pi/180)*KickBoV
R_KickBoV = -(math.pi/180)*KickBoV
if clientID!= -1:
    print('connect successfully')
else:
    print('connect failed')
    vrep.simxFinish(clientID)
print('program ended')

errorCode,Sphere_handle=vrep.simxGetObjectHandle(clientID,'Sphere',vrep.simx_opmode_oneshot_wait)
errorCode,Pole_handle=vrep.simxGetObjectHandle(clientID,'Pole',vrep.simx_opmode_oneshot_wait)
errorCode,Revolute_handle=vrep.simxGetObjectHandle(clientID,'Revolute',vrep.simx_opmode_oneshot_wait)
errorCode,Prismatic_handle=vrep.simxGetObjectHandle(clientID,'Prismatic',vrep.simx_opmode_oneshot_wait)
errorCode,Revolute2_handle=vrep.simxGetObjectHandle(clientID,'Revolute2',vrep.simx_opmode_oneshot_wait)
errorCode,Prismatic2_handle=vrep.simxGetObjectHandle(clientID,'Prismatic2',vrep.simx_opmode_oneshot_wait)
errorCode,Pole2_handle=vrep.simxGetObjectHandle(clientID,'Pole2',vrep.simx_opmode_oneshot_wait)

if errorCode == -1:
    print('Can not find left or right motor')
    sys.exit()

def start():
    errorCode = vrep.simxStartSimulation(clientID,vrep.simx_opmode_oneshot_wait)
    while True:
        try:
            if keyboard.is_pressed('r'):
                vrep.simxSetJointTargetVelocity(clientID,Revolute_handle,R_KickBoV,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('t'):
                vrep.simxSetJointTargetVelocity(clientID,Revolute_handle,L_KickBoV,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('q'):
                vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,0.05,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('w'):
                vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,0,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('e'):
                vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,-0.05,vrep.simx_opmode_oneshot_wait)

            elif keyboard.is_pressed('.'):
                vrep.simxSetJointTargetVelocity(clientID,Revolute2_handle,L_KickBoV,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('/'):
                vrep.simxSetJointTargetVelocity(clientID,Revolute2_handle,R_KickBoV,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('b'):
                vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,0.05,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('n'):
                vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,0,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('m'):
                vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,-0.05,vrep.simx_opmode_oneshot_wait)
            else:
                pass
        except:
            break
    start()
```


3.4.2 手足球手控與電腦對打

[手足球手控與電腦對打 ttt 檔](#)

[對打影片網址](#)

```
import vrep
import keyboard
import time
import sys, math
vrep.simxFinish(-1)
clientID = vrep.simxStart("127.0.0.1", 19997, True, True, 5000, 5)
KickBoV = 360
L_KickBoV = (math.pi/180)*KickBoV
R_KickBoV = -(math.pi/180)*KickBoV
if clientID!= -1:
    print('connect successfully')
else:
    print('connect failed')
    vrep.simxFinish(clientID)
print('program ended')

err,Sphere_handle=vrep.simxGetObjectHandle(clientID,'Sphere',vrep.simx_opmode_oneshot_wait)
err,Pole_handle=vrep.simxGetObjectHandle(clientID,'Pole',vrep.simx_opmode_oneshot_wait)
err,Revolute_handle=vrep.simxGetObjectHandle(clientID,'Revolute',vrep.simx_opmode_oneshot_wait)
err,Prismatic_handle=vrep.simxGetObjectHandle(clientID,'Prismatic',vrep.simx_opmode_oneshot_wait)
err,Revolute2_handle=vrep.simxGetObjectHandle(clientID,'Revolute2',vrep.simx_opmode_oneshot_wait)
err,Prismatic2_handle=vrep.simxGetObjectHandle(clientID,'Prismatic2',vrep.simx_opmode_oneshot_wait)
err,Pole2_handle=vrep.simxGetObjectHandle(clientID,'Pole2',vrep.simx_opmode_oneshot_wait)

if err == -1:
    print('Can not find left or right motor')
    sys.exit()

def speed(handle,speed):
    err = vrep.simxSetJointTargetVelocity(clientID,handle,speed,vrep.simx_opmode_oneshot_wait)
def start():
    err=vrep.simxStartSimulation(clientID,vrep.simx_opmode_oneshot_wait)
    while True:
        err,position_LR=vrep.simxGetObjectPosition(clientID,Revolute2_handle,-1,vrep.simx_opmode_oneshot)
        err,position_S=vrep.simxGetObjectPosition(clientID,Sphere_handle,-1,vrep.simx_opmode_oneshot)
        err,position_RR=vrep.simxGetObjectPosition(clientID,Revolute_handle,-1,vrep.simx_opmode_oneshot)
        LP =position_S[1] -position_LR[1]
        LPP =position_S[0] - position_LR[0]
        print(position_S)
        if LPP > -0.015:
            speed(Revolute2_handle,L_KickBoV)
        elif LPP <= -0.015:
            speed(Revolute2_handle,R_KickBoV)
        else:
            pass
        try:
            if keyboard.is_pressed('r'):
                vrep.simxSetJointTargetVelocity(clientID,Revolute_handle,R_KickBoV,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('t'):
                vrep.simxSetJointTargetVelocity(clientID,Revolute_handle,L_KickBoV,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('q'):
                vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,0.1,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('w'):
                vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,0,vrep.simx_opmode_oneshot_wait)
            elif keyboard.is_pressed('e'):
                vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,-0.1,vrep.simx_opmode_oneshot_wait)
            else:
                pass
        except:
            break
        Pv = LP*2.5
        vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,Pv,vrep.simx_opmode_oneshot_wait)

vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,0,vrep.simx_opmode_oneshot_wait)
vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,0,vrep.simx_opmode_oneshot_wait)

start()
```

3.4.3 手足球電腦與電與對打

[手足球電腦與電與對打 ttt 檔](#)

[對打影片網址](#)

```
import vrep
import keyboard
import time
import sys, math
vrep.simxFinish(-1)
clientID = vrep.simxStart("127.0.0.1", 19997, True, True, 5000, 5)
KickBoV = 360
L_KickBoV = (math.pi/180)*KickBoV
R_KickBoV = -(math.pi/180)*KickBoV
if clientID!= -1:
    print('connect successfully')
else:
    print('connect failed')
    vrep.simxFinish(clientID)
print('program ended')

err,Sphere_handle=vrep.simxGetObjectHandle(clientID,'Sphere',vrep.simx_opmode_oneshot_wait)
err,Pole_handle=vrep.simxGetObjectHandle(clientID,'Pole',vrep.simx_opmode_oneshot_wait)
err,Revolute_handle=vrep.simxGetObjectHandle(clientID,'Revolute',vrep.simx_opmode_oneshot_wait)
err,Prismatic_handle=vrep.simxGetObjectHandle(clientID,'Prismatic',vrep.simx_opmode_oneshot_wait)
err,Revolute2_handle=vrep.simxGetObjectHandle(clientID,'Revolute2',vrep.simx_opmode_oneshot_wait)
err,Prismatic2_handle=vrep.simxGetObjectHandle(clientID,'Prismatic2',vrep.simx_opmode_oneshot_wait)
err,Pole2_handle=vrep.simxGetObjectHandle(clientID,'Pole2',vrep.simx_opmode_oneshot_wait)

if err == -1:
    print('Can not find left or right motor')
    sys.exit()

def speed(handle,speed):
    err = vrep.simxSetJointTargetVelocity(clientID,handle,speed,vrep.simx_opmode_oneshot_wait)
def start():
    err=vrep.simxStartSimulation(clientID,vrep.simx_opmode_oneshot_wait)
    while True:
        err,position_LR=vrep.simxGetObjectPosition(clientID,Revolute2_handle,-1,vrep.simx_opmode_oneshot)
        err,position_S=vrep.simxGetObjectPosition(clientID,Sphere_handle,-1,vrep.simx_opmode_oneshot)
        err,position_RR=vrep.simxGetObjectPosition(clientID,Revolute_handle,-1,vrep.simx_opmode_oneshot)
        LP =position_S[1] - position_LR[1]
        LPP =position_S[0] - position_LR[0]
        RP =position_S[1] - position_RR[1]
        RPP =position_S[0] - position_RR[0]
        print(position_S)
        if LPP < -0.015:
            speed(Revolute2_handle,R_KickBoV)
        elif LPP >= -0.015:
            speed(Revolute2_handle,L_KickBoV)
        else:
            pass
        if RPP > 0.015:
            speed(Revolute_handle, L_KickBoV)
        elif RPP <= 0.015:
            speed(Revolute_handle,R_KickBoV)
        else:
            pass
        Pv = LP*3
        PPV=RP*3
        vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,Pv,vrep.simx_opmode_oneshot_wait)
        vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,Pv,vrep.simx_opmode_oneshot_wait)
        vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,0,vrep.simx_opmode_oneshot_wait)
        vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,0,vrep.simx_opmode_oneshot_wait)

start()
```

3.4.4 模擬問題與討論

在模擬上遇到的幾個問題

1. 開始模擬時桿子會自行移動

```
vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,0,vrep.simx_opmode_oneshot_wait)  
vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,0,vrep.simx_opmode_oneshot_wait)
```

圖 34 模擬問題討論-1

給他初始速度為 0 即可靜止

2. 程式 GO 的時候 def SPEED 發生錯誤

```
- def speed(handle,speed):  
    err = vrep.simxSetJointTargetVelocity(clientID,handle,speed,vrep.simx_opmode_oneshot_wait)
```

圖 35 模擬問題討論-2

給他定義值即可解錯

3. 開始模擬時桿子追蹤球的 Y 軸位置與球的 Y 軸路徑相反

```
- def speed(handle,speed):
    err = vrep.simxSetJointTargetVelocity(clientID,handle,speed,vrep.simx_opmode_oneshot_wait)
- def start():
    err=vrep.simxStartSimulation(clientID,vrep.simx_opmode_oneshot_wait)
    while True:
        err,position_LR=vrep.simxGetObjectPosition(clientID,Revolute2_handle,-1,vrep.simx_opmode_oneshot)
        err,position_S=vrep.simxGetObjectPosition(clientID,Sphere_handle,-1,vrep.simx_opmode_oneshot)
        err,position_RR=vrep.simxGetObjectPosition(clientID,Revolute_handle,-1,vrep.simx_opmode_oneshot)
        LP=position_S[1]-position_LR[1]
        LPP=position_S[0]-position_LR[0]
        RP=position_S[1]-position_RR[1]
        RPP=position_S[0]-position_RR[0]
        print(position_S)
        if LPP < -0.015:
            speed(Revolute2_handle,R_KickBoV)
        elif LPP >= -0.015:
            speed(Revolute2_handle,L_KickBoV)
        else:
            pass
        if RPP > 0.015:
            speed(Revolute_handle, L_KickBoV)
        elif RPP <= 0.015:
            speed(Revolute_handle,R_KickBoV)
        else:
            pass
        Pv = LP*3
        PPV=RP*3
        vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,Pv,vrep.simx_opmode_oneshot_wait)
        vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,Pv,vrep.simx_opmode_oneshot_wait)
        vrep.simxSetJointTargetVelocity(clientID,Prismatic_handle,0,vrep.simx_opmode_oneshot_wait)
        vrep.simxSetJointTargetVelocity(clientID,Prismatic2_handle,0,vrep.simx_opmode_oneshot_wait)
```

圖 36 模擬問題討論-3

注意 下放給定移動值的數值是否有負號，有的畫刪除即可，或是更改上方的正負關係即可

4. 無法擊球

```
print(position_S)
if LPP < -0.015:
    speed(Revolute2_handle,R_KickBoV)
elif LPP >= -0.015:
    speed(Revolute2_handle,L_KickBoV)
else:
    pass
if RPP > 0.015:
    speed(Revolute_handle, L_KickBoV)
elif RPP <= 0.015:
    speed(Revolute_handle,R_KickBoV)
else:
    pass
Pv = LP*3
PPV=RP*3
```

圖 37 模擬問題討論-4

若無法擊球可嘗試在距離潛在負號或是更改大小於關係即可

5. 注意程式裡與 V-REP 的軸名稱要相同

3.5 問題和討論

Q：遇到球無法滾出球門？

A：有可能是因為你簡化的過程，簡化的模型太高，而導致球會卡在球門，所以遇到類似情形把簡化的模型刪掉即可

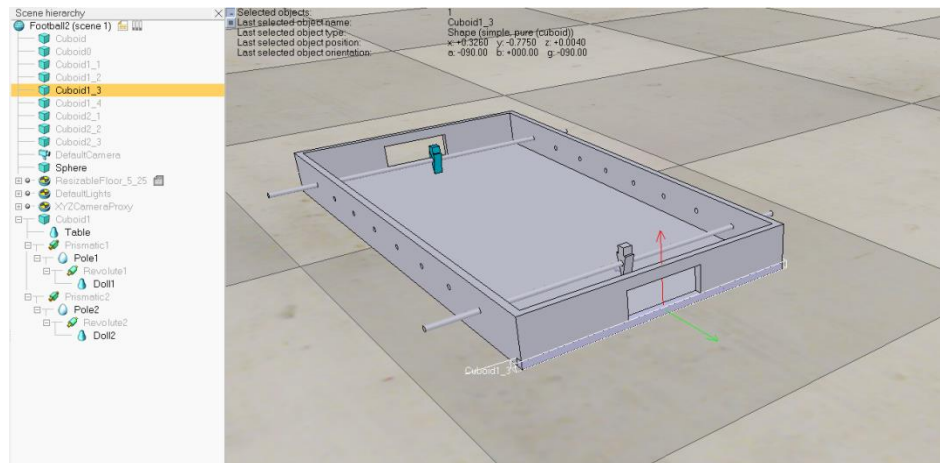


圖 38 常見問題和討論-1

Q：球進門後卡在球門下？

A：繪製草圖時因不小心讓那裏成為平台，而導致球無法向下滾動，所以我們把那個地方改成斜面即可！



圖 39 常見問題和討論-2

Q：球順著軌道滾動後卡在回球機構的門口？

A：把送球機構的孔洞放大，讓球得以順利通過返回球桌！

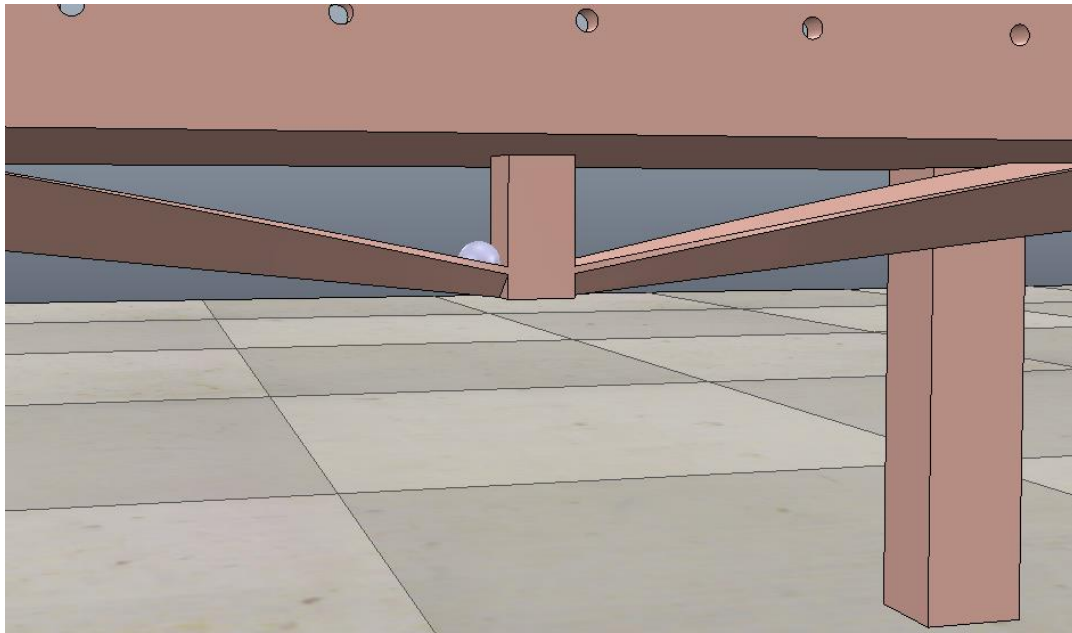


圖 40 常見問題和討論-3

Q：如何讓桿子偵測到球體的位置並進行移動和反擊？

A：先利用指令得知球體的 Y 方向座標，再進行相減得到值即可確認球體在桿子的右邊還是左邊，並使桿子確實反擊

Q：要如何得知球體目前的位置？

A：使用指令 `number returnCode,number
parentObjectHandle=vrep.simxGetObjectParent(number clientID,number
objectHandle,number operationMode)`即可得知目前球體的座標位置