

國立虎尾科技大學

機械設計工程系

電腦輔助設計實習 第四小組期末報告

40523119 林建安

40523120 侯安調

40523121 施恩澤

40523128 許凱翔

40523129 郭育良

40523130 陳士正

指導教授：嚴家銘

目錄

摘要.....	5
壹、前言	6
一、研究背景與動機	6
二、報告架構	6
貳、研究設備及器材	7
參、研究過程或方法.....	12
肆、問題以及討論	17
伍、結論	17

摘要

撰寫本篇報告主要目的是將本學期後半段的積電資整合專案，做一個系統化的整理與檢討，除了將寫實的過程言簡意賅的表達出來之外，也是將過程所得到的寶貴的經驗記錄下來。報告重點會以概論、實作過程、遇到的困難為主要內文。

關鍵字:機電資整合、動態模擬、機械設計

壹、前言

一、研究背景與動機

為因應國內高端機械設計相關人士的需求，該報告之課程將引導學員們，學習如何進行機械設計、分析、模擬，最中間進行資料版次管理、協同設計、解決問題，最後再以本報告呈現最後的表達。

二、報告目標與內容架構

本報告目標將說明身為一位機械設計工程師，在設計流程應有的步驟與流程，實際操作的各種情況以及問題，以完成一組機電資整合玩具為最後目標。組成架構為:工具選用、實作過程、問題與討論。

參、工具選用

一、軟體

- Fossil SCM

Fossil 是一個簡單且高效能的軟體管理系統。可能很多人會有問，為什麼我們不用 Github 就好呢？就以目前來說，我們所使用到的功能，都跟 Github 重疊 (e.g. [blog/GitHub page](#), [wiki](#), [timeline](#))，那我們作為一個機械設計工程師，選用 Fossil 的優勢究竟何在？

當我今天要設計一台複雜的機器，組成的零件可能有上百個甚至是上千個的時候，該如何進行一個有效率的管理與溝通。用隨身檔案櫃來摳去嗎？還是利用 Github 的倉儲呢？！我不得不說，其實 Github 比較適用於「軟體」工程師使用，對於機械工程師來說，用起來不太友善。

這時候，我們需要一個專案級的版本控制系統，來進行組態管理，不但管理文件還能管理人員。Fossil 除了提供 wiki 功能之外，還有 Bug tracking、tickets 以及 technotes 的功能。Bug tracking 確保當產品出現問題，可以快速得知處理的進度跟結果。Technote 能將有用得技術文件留下來供彼此交流學。至於 Tickets 呢？它可以解釋成任務，擁有權限的人，足以派發 Tickets 給其他人，就是老闆叫你工作的意思，

完成則提交，退回來還是得繼續做，一切的行為紀錄都留在 Fossil 裡，

所以呢？還需要打卡嗎？

除了版本控制之外，還能夠分別給予使用者不同的權限，確保任何檔案的安全性以及分工合作。

再來擁有瀏覽器介面的 Fossil，也是它的一大優點。各式文件與功能都在網頁上操作，簡單明瞭又即時。

最後它的網路架構非常親民，簡單 http 協定即可搞定，你不要一個固定 IP 又或是一台伺服器，只要鍵入 fossil ui，登登登，屬於自己本地端的版本控制系統就完成了。

Fossil 這個部分最後想補充一下，版本控制系統究竟是多重要。以後到外面工作，不要說三流的公司好了，起碼二流的公司行號，都會使用這類的內容版本控制系統。圖中可以看到，很多人一起工作，沒有使用 SCM 的結果，零散的數位檔案、殘破不堪的紙本會議記錄簿、有一點小事就要開會的主管，還是說利用 Fossil 這類的軟件來進行有效的溝通呢？

這其實就是團隊合作，與以往我們所體驗的合作關係不太一樣，覺得見到面才叫有參與，拿張 A4 白紙畫心智圖才叫做討論，現在更應該學習有效率的雲端協同合作！

- Solvespace

Solvespace 是一個可參數化的 3D 建模軟體。說老實話，起初剛使用它的時候，真的覺得，怎麼這麼難用，到底誰會用這種東西。

可能以往使用好軟體習慣了，一時間難以適應這麼不人性化的免費開源軟體。

不過經過一些練習之後，還是能夠發現它的一些特點。雖然操作上有點不太方便，但熟悉介面與功能的時候，還是能夠用的挺上手的。且在學生時期，我們總是一昧的尋求盜版軟體，要我花錢買嗎？甲卡邁欸啦！想要畫個 3D 還得下載 7GB 的 SW，還得擔心破解程式被安裝後門軟體，用的膽戰心驚的，相信市面上的一些小公司一定也是用盜版軟體在做商業盈利。

Solvespace 就會是我們能夠好好利用的工具，只要一個 6MB 的檔案，放在隨身碟裡，到哪裡我都能夠作設計，而且除了支援 Windows 之外，Linux 跟 MacOS 都能使用，對於 MacOS 來說是一大幫助，以前都得利用虛擬機或是 Bootcamp 跑 Windows 的 SW，簡單的零組件我就可以用 solvespace 來進行繪製。

雖然它檔案很小，不過基本該有的功能都有，對於輕量級的專案設計來說，是個非常棒的好工具，除此之外，還提供點追蹤路徑分析功能，對於在進行複雜結構的幾何運算，相當有幫助。

最後，除了可儲存成特有的 slvs 檔之外，還提供匯出成其他可交換的檔案格

式，讓其他軟體進行讀取，真的是麻雀雖小，五臟俱全。

- Onshape

這次課程使用的設計工具，除了 Solvespace 之外，另一個則是 Onshape，它是一個第一個且完全雲端的 3D 電腦輔助設計服務，它可以讓你在任何環境的瀏覽器上，甚至是手機、平板電腦上來做使用。

哇，這實在太潮了！Solvespace 根本不夠看了，6MB 不用！只要讓我上網就可以開始設計了，而且能夠加入協同者一起進行設計，對於機械設計來說，是一大新體驗，以往畫好圖只能以檔案傳送的方式供其他人做資訊交換，傳來傳去的有夠麻煩，能夠直接在網路上做協調設計，效率提高了很多呢！不過，如果你想擁有私人專案，得額外付費，擁有這樣的功能與服務，我想花錢付費不是問題的。

- V-REP

[V-REP](#) 是機器人模擬軟體中的瑞士刀,您再也找不到支援更多功能,更多特色或有更詳盡的應用程式介面的機器人模擬軟體:

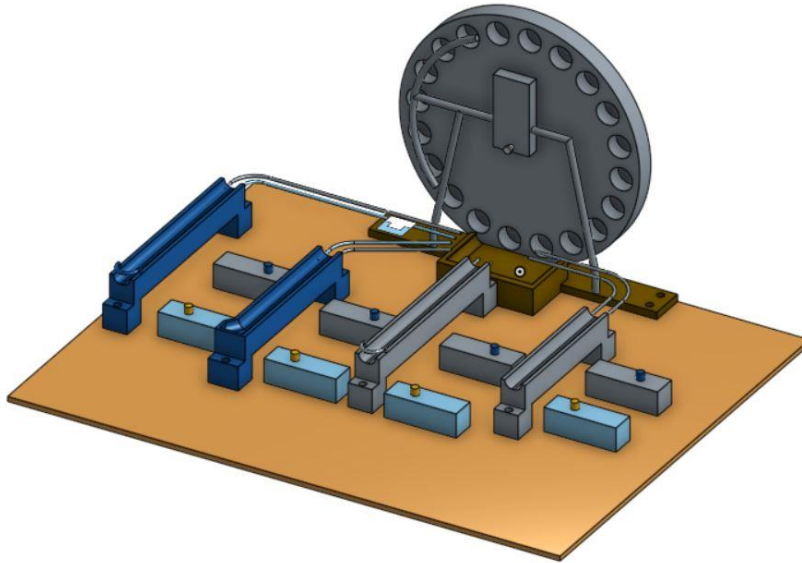
- 跨平台支援(Windows, MacOSX, Linux)
- [支援六種撰寫程式的方式 \(嵌入式腳本、插件、附加組件、ROS 節點、遠端客戶端應用程式介面、或自訂解決方案\)](#)
- 支援七種程式語言 ([C/C++](#)、[Python](#)、[Java](#)、[Lua](#)、[Matlab](#)、[Octave](#)、和 [Urbi](#))
- [400 種以上的應用程式介面函數](#)
- [支援 100 種 ROS 服務、30 種 ROS 發佈方式、25 種 ROS 訂戶類型](#)、具備可擴充性
- [支援 4 種物理引擎 \(ODE, Bullet, Vortex, Newton\)](#)
- 整合 ray-tracer ([POV-Ray](#))
- [具備完整的運動學計算器](#) (支援任何機構的正運動學與逆運動學計算)
- [網格-網格干涉偵測 \(meshes, octrees and point clouds\)](#)
- [網格-網格最短距離計算 \(meshes, octrees and point clouds\)](#)
- [支援路線及運動完整規劃](#) (支援二~六維的完整約束,車輛類型載具的非完整約束,以及對於連鎖運動的規劃)
- [內建影像處理的視覺感測器](#) (完全可擴充)

- [真實的接近感測器](#) (偵測範圍內的最短距離計算)
- [內建含編輯器的客製化使用者介面](#)
- [完全整合第四類 Reflexxes Motion Library](#) 以及 [RRS-1 介面規格](#)
- [模擬表面切削](#)
- [支援資料的儲存與視覺化](#) (time graphs, X/Y graph or 3D curves)
- [整合形狀編輯模式](#)
- 支援噴水與氣體噴射的動態顆粒模擬
- [支援拖放功能的模型瀏覽器](#) (模擬計算中仍可使用)
- 支援多重的取消/重做、[影像錄製](#)、油漆噴塗、[建立詳盡的文檔](#)等功能

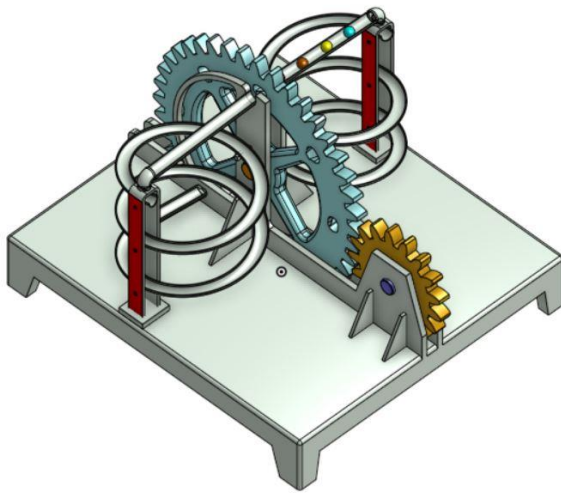
機器人, 機器人學, 模擬器, 模擬計算, 運動學, 動力學, 路徑規劃, 最短距離計算, 碰撞偵測, 視覺感測器, 影像處理, 接近感測器, 油漆噴塗模擬.

肆、研究過程或方法

一、繪製主體



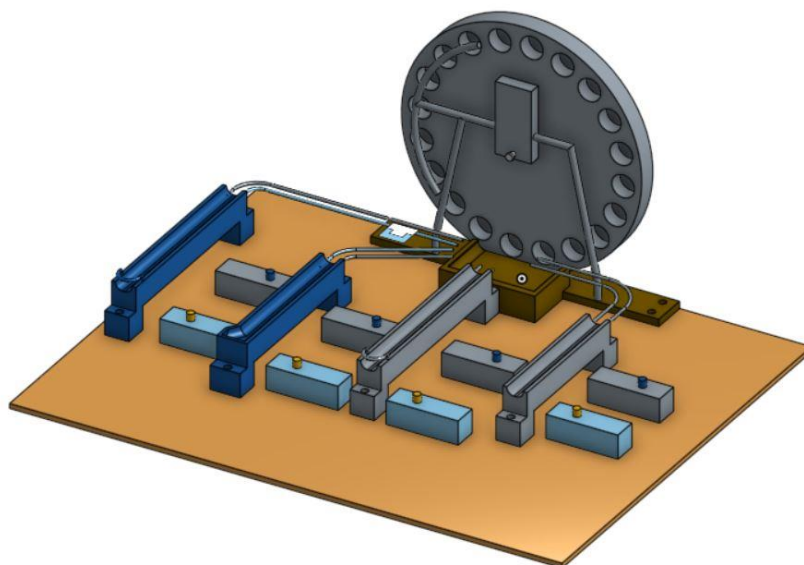
原先設計是利用簡單的圓盤凹巢將球帶至高再將球送出，下面分別可以裝上四種不同的模組，互相不會影響到，但是最後因為上面預想機構過於複雜，很難去繪製出理想的運動情形的組立，最後組員們重新繪製了新的主體。



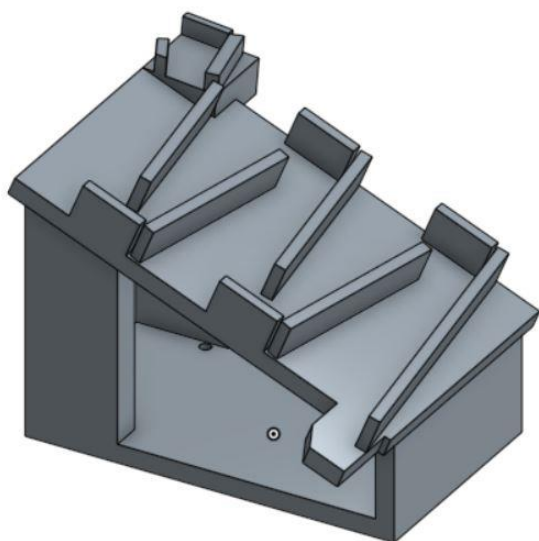
新的主體依然用圓盤將球帶往上方，傳動方面選用齒輪作傳輸，接球軌道改得比較簡單，但是最後在模擬的時候，因為對軟體不太熟悉，沒有辦法好好的將真實狀態模擬出來。

二、 軌道設計

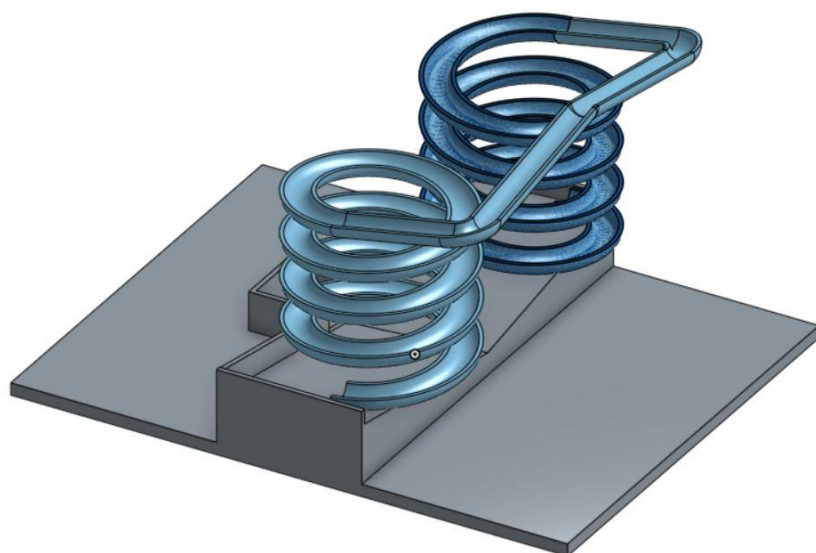
40523119 林建安



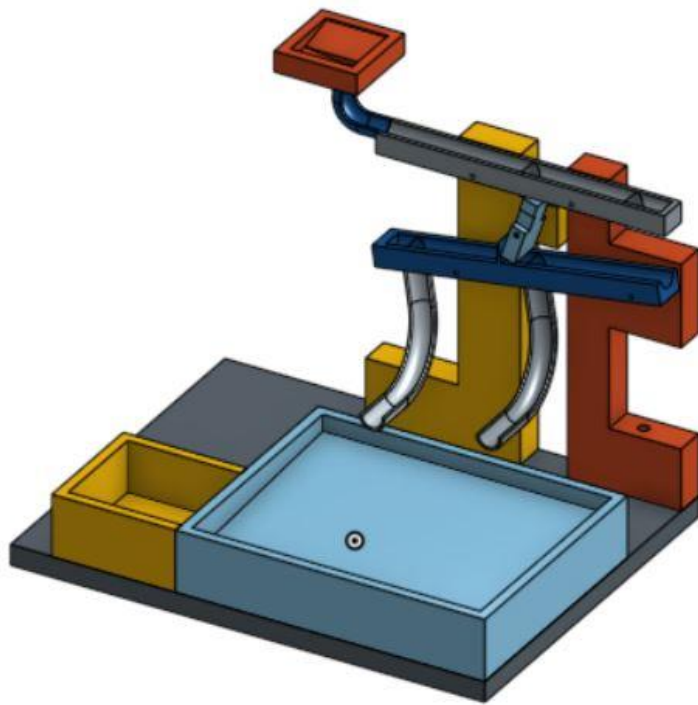
40523128 許凱翔



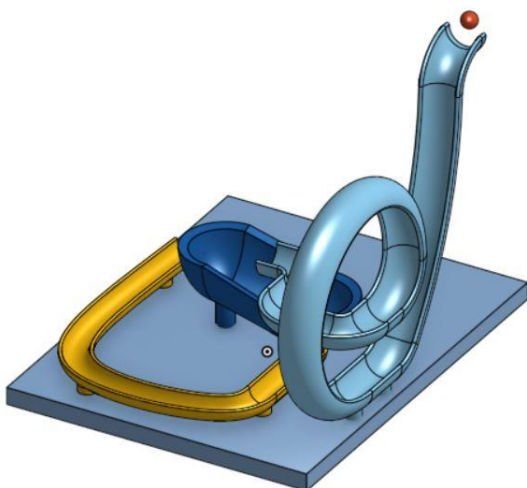
40523129 郭育良



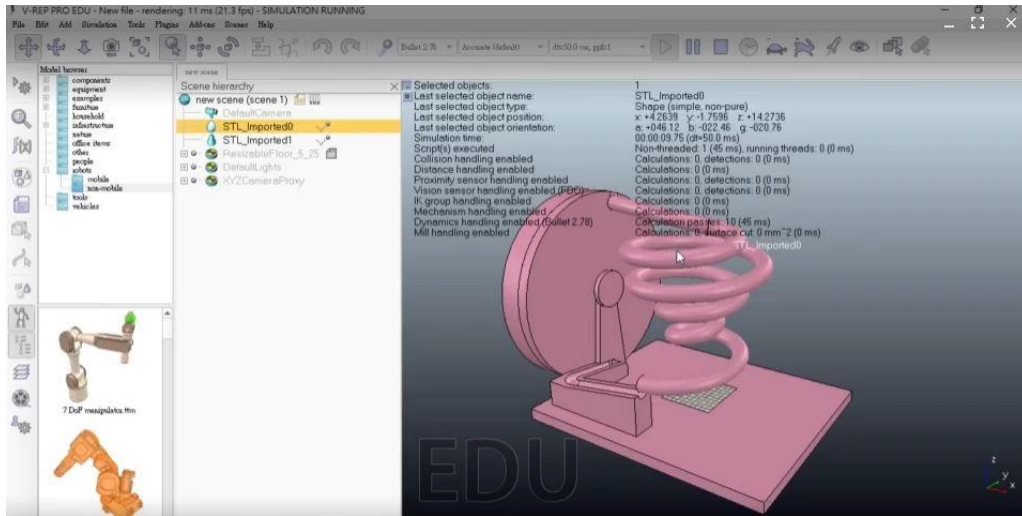
40523121 施恩澤



40523120 侯安調



40523130 陳士正



伍、問題以及討論

1.原主體問題

這次我們原先是參照網路上的類似風車原理來當作我們主體，載運球方面都沒問題，但是在出口的時候發現畫不出來，經過我們小組討論一同決定要換個方向想一個另一種類型的主體，並放棄原來的主體。

2.導致問題(1)的原因

畫不出來最主要的原因就是我們對 onshape 這個還不是非常的熟悉，在很多方面都還不是很明白它的運作方式，所以需要我們更加努力去鑽研它來解決這問題。

3.新主體模擬情況

新主體前期模擬是非常順利地但是到最後的轉彎點球卻會滑出軌道，主要是因為我們在畫製的時候沒有設計好，這也是我們經驗不足所需要加強的地方。

陸、結論

學號:40523119

個人簡報影片:<https://youtu.be/UbCH1xv2CeU>

第 4 組 - https://github.com/s40523119/ag4_cadp_finalproject/blob/gh-pages/report/report.pdf

個人課程心得:

這次的課程，相較之前學習負擔較大一些，期末的專案題目，我認為對我們來說是相對困難的，沒有機構學的基礎，很難憑空想出有創意或是可行性高的機構，亦或是很容易發生干涉衝突。就連起初的"設計"對我們來說就是一大挑戰，網路神人影片範例百百種，各組都是挑選喜歡或是有趣的範例做為參考，但是沒有實體的情況下，只能隔空抓藥，憑藉著影片的樣子，自行揣摩尺寸比例等等，每個人再做設計的時候，都發生諸多問題。得靜下來好好想一下，範例那樣的設計，究竟與那些"關鍵"尺寸或是特徵有關，我們就必須去做數值運算，從單純的繪圖跨出到"設計"的第一步。

學期初所教的 solvespace，後來仔細想想，工具的可攜性與靈活度深深影響著工程師的工作效率，在什麼的條件下，選擇有利於我們開發的工具才是首要，而不是追求高端的繪圖應用。

後半段所練習的 V-REP，我想這應該是大多數同學的痛處，我也認為他頗有技術難度。跨平台的支援，方便的串接 API，支援多種程式語言，它的支援度與強大，伴隨著較高的學習門檻。一直可以感覺得出來，課堂上老師不斷地想要激發同學們"自學"的動機，希望可以渴望追求知識。但是就好比一年級的程式課程，大多數的人碰壁了之後，就停留原地甚至逃避，往往丟失了學習機會。

V-REP 官方文件自己說可以模擬全自動化工廠，各種物理狀態情形，程式的強大不在話下。想要專精它的話，得有個自學的契機，找個自己有"興趣"的實作，想盡辦法的實現它。V-REP 在繁體中文的網站，幾乎看不太到，只有一個菁英高端的學校的有發表數篇文章與各種影片，前途無量。

最後，透過這門課程，讓我們"簡單"的了解，設計應有的概念與流程，並得知原來模擬對於重大專案的重要性。想必老師並非要考倒大家，而是學習到藏在魔鬼中的那些細節，包括我自己，每個人都得在更努力才行。