

## Finial project

# Random Grouping

要進行隨機分組及隨機點名，必須先了解程式裡導入的模組及在模組中所用的語法、迴圈:request、ast.literal\_eval、random、for 迴圈

(1) request 模組:此模組是為了可以使用 Python 來下載網頁上的資料，是以 request 模組建立適當的 HTTP 請求，透過 HTTP 請求從網頁伺服器下載指定的資料，分為兩種模式 Post、Get。

參考文獻: <https://ithelp.ithome.com.tw/articles/10206215>

(2) ast 模組: 此模組是 python 原始碼到位元組碼的一種中間產物，藉助 ast 模組可以從語法樹的角度分析原始碼結構，其中的語法 ast.literal\_eval 是為了能夠安全地評估表達式節點或包含 Python 文字或容器顯示的字符串。提供的字符串或節點只能由以下 Python 文字結構組成：字符串、字節、數字、元組、列表、字典、集合、布林值、None。

參考文獻: <https://docs.python.org/3/library/ast.html>  
<https://www.itread01.com/content/1544847254.html>

(3) random 模組:此模組用於生成偽隨機數，且有許多種用法，例如:隨機選取 choice、隨機選取 sample (一次選多筆資料)、隨機調換順序 shuffle (就地調換順序)、隨機取得亂數 random (0~1 之間的隨機亂數)、隨機取得亂數 uniform (可以指定範圍)。

參考文獻: <https://ithelp.ithome.com.tw/articles/10207483>

(4) for 迴圈: 在處理資料的時候，時常會需要重複執行某些相同的步驟；而迴圈 (loop) 的作用是讓指定的某段敘述在條件符合的情況下一直重覆執行，是程式設計中很重要的一種控制結構，且分為兩類:for、while

○ for 迴圈基本語法:for 變數 in list、字串、range

○ while 迴圈基本語法:while 布林值

for 迴圈及 while 迴圈比較: for 迴圈比較適用在已知迴圈數的問題，而

while 迴圈則適用在「無法預知迴圈數」的問題上。

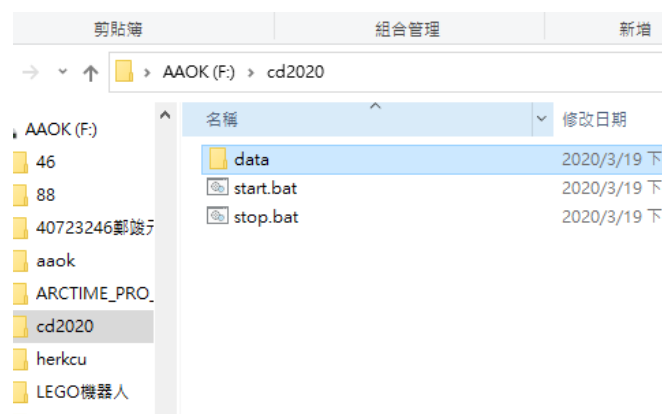
參考文獻: <https://medium.com/ccclub/ccclub-python-for-beginners-tutorial4990a5757aa6>

## Potable System

**text:**

將可攜系統下的 python 3.7.3 系統更新至 python 3.8.2

- 1.先去 GOOGLE SEARCH [PYTHON](#) (點這下載)
- 2.下載 [Windows x86-64 executable installer](#) (點這下載)
- 3.載完點並不要勾選 pip 選項，接著按下 next 選取放置資料夾
- 4.在隨身系統下新增放置資料夾 cd2020，並在 cd2020 下新建資料夾 data，  
data<py382<wscite432<portablegit
- 5.下載 SciTE 系統 [full 64-bit download](#)，下載完後解壓縮至 data 下的 wscite432
- 6.在 cd2020 下建立新的 start.bat 及 stop.bat



- 7.至 SciTE 系統更改 start.bat 及 stop.bat

start.bat:

```
1  @echo off
2  set Disk=y
3  subst %Disk%: "data"
4
5  %Disk%:
6
7  set HomePath=%Disk%:\home
8  set HomeDrive=%Disk%:\home
9  set Home=%Disk%:\home
10 set USERPROFILE=%Disk%:\home
11
12 REM 將系統 Python 程式的 io 設為 utf-8
13 set PYTHONIOENCODING="utf-8"
14
15 #REM for putty
16 #Set GIT_HOME=%Disk%:\portablegit\bin\
17 #Set GIT_SSH=%Disk%:\putty\plink.exe
18
19 set PYTHONPATH=%Disk%\py382\DLLs;%Disk%\py382\Lib;%Disk%\py382\Lib\site-packages;
20 set PYTHONHOME=%Disk%\py382
21
22 set path_python=%Disk%\py382;%Disk%\py382\Scripts;
23 set path_msys2=%Disk%\msys64\mingw64\bin;
24 set path_git=%Disk%\portablegit\bin;
25 set path_tcc=%Disk%\tcc;
26
27 path=%Disk%;;%path_python%;%path_msys2%;%path_git%;%path_tcc%;
28
29 start /MIN cmd.exe
30 start /MIN cmd.exe
31 start /MIN cmd.exe
32 start /MIN cmd.exe
33
34 start /MIN %Disk%\wscite432\wscite\SciTE.exe
35 start /MIN %Disk%\wscite432\wscite\SciTE.exe
36
37 Exit
```

stop.bat:

```
1  @echo off
2  set Disk=y
3  path=%PATH%;
4
5  taskkill /IM python.exe /F
6  taskkill /IM pythonw.exe /F
7  taskkill /IM scite.exe /F
8
9  REM 終止虛擬硬碟與目錄的對應
10 subst %Disk%: /D
11 REM 關閉 cmd 指令視窗
12 taskkill /IM cmd.exe /F
13
14 EXIT
```

8.載入 MSYS2 [msys2-x86\\_64-20190524.exe](#) 至 data

9.載入 PortableGit [64-bit Git for Windows Portable](#) 至 portablegit

10.裝入 [pip](#)，另存新檔到 data

11.測試將 python3.8.2 版本，將少的模組下載回來

```
1 | python get-pip.py
```

```
2 | pip install flask bs4 lxml pelican markdown flask_cors leo
```

# Four Wheel Car

*Four Wheel Car*-Analysis and integration

[BubbleRob](#)  
[tutorial:](#)

40723206 提供

此部分由

在此教程中會讓我們設計簡單的 **BubbleRob** 機器人並進行一些基礎的模擬操作，而大二上我們其實就有實際的以 **v-rep** 操作過這個教程，在這個教程中的說明包括了如何建造模型機器人主體、車輪及後方支撐滑塊，還有如何建造及設置障礙物，以及設定好的距離傳感器後使其去感測並避開我們所設置的障礙物，而最重要的要點是如何去設計指令讓傳動器依照我們所想要的方式去做動，且當距離感測器偵測到障礙物時才能使機器人迅速閃避障礙物，另外還有視覺辨識器，使機器人在移動時，讓我們能夠觀測到機器人行徑方向之路況。

translation Simulation ~ PDF

# 模擬

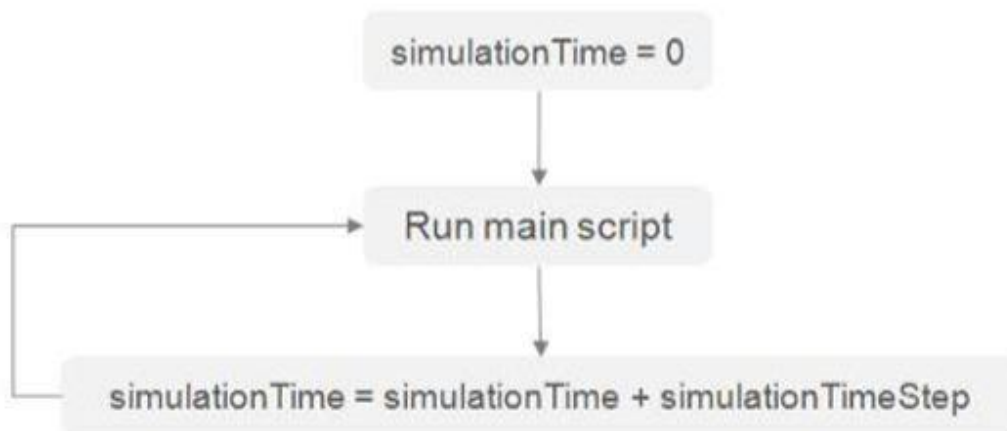
使用[菜單欄->模擬->開始/暫停/停止模擬]



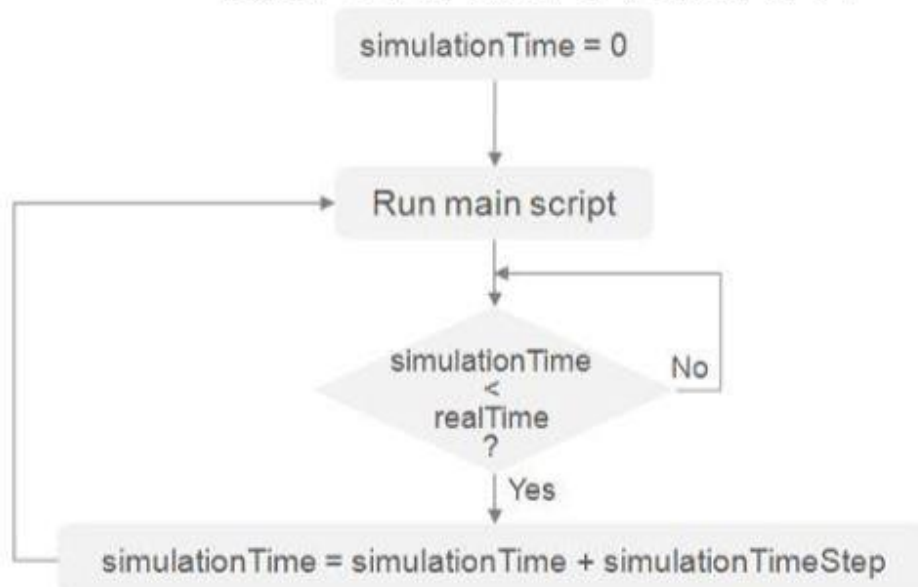
[模擬開始/暫停/停止工具欄按鈕]

## 仿真循環

模擬器通過以恆定的時間步長推進模擬時間來進行操作。  
下圖說明了主要的仿真循環

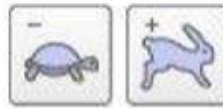


通過嘗試使仿真時間與實時保持同步來支持實時仿真



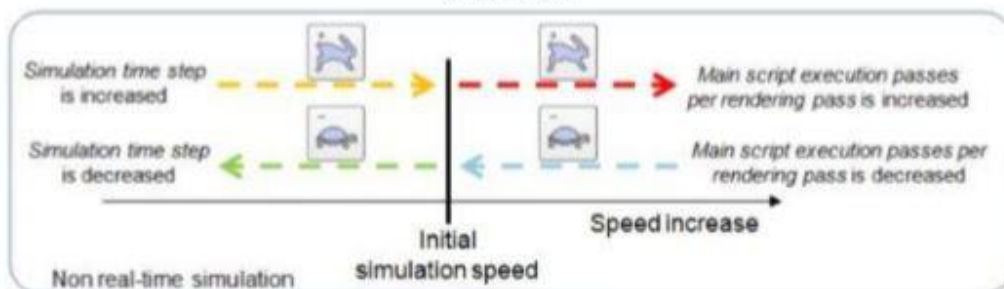
translation Simulation dialog ~ PDF

## 仿真速度

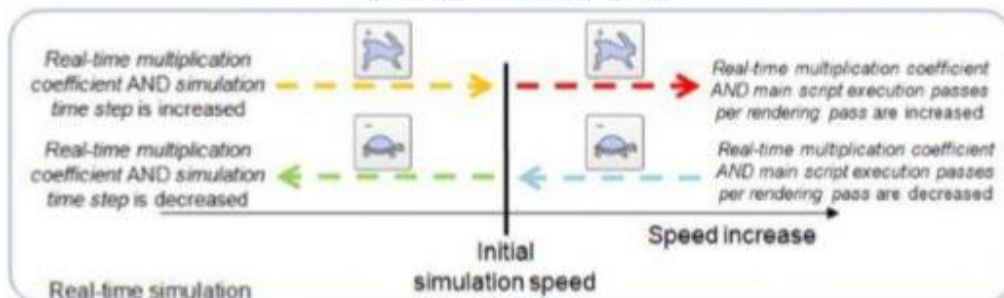


【模擬速度調整工具欄按鈕】

調整模擬速度，以使初始模擬時間步長永遠不會增加  
如下圖示



【非實時模擬的模擬速度調整機制】



【用於實時仿真的仿真速度調整機制】

## 螺紋渲染



【線程渲染工具欄按鈕】

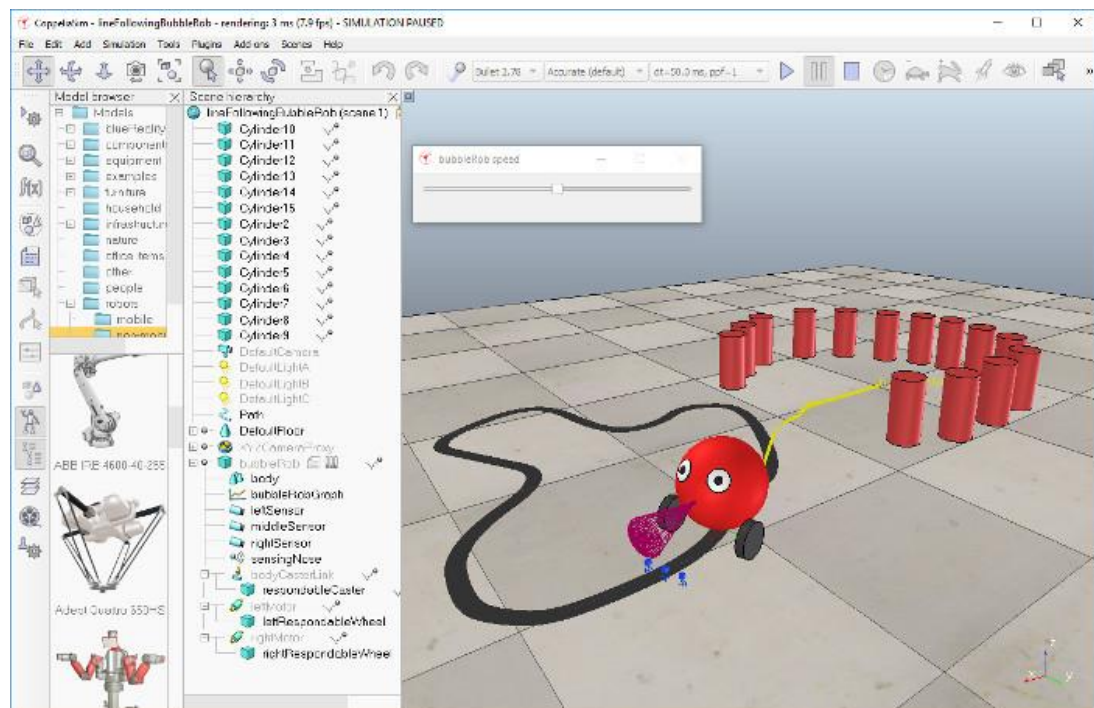
- 渲染將與模擬循環異步進行，並且可能會出現視覺故障
- 該錄像機將不以恆定速度運轉（某些幀可能會跳過）
- 應用程序的穩定性可能會降低
- 某些操作（例如擦除對象等）需要等待渲染線程完成工作才能執行，反之亦然。在那些情況下，循環可能比順序渲染模式花費更多的時間。

Line following BubbleRob tutorial

在本教程中，我們旨在擴展 BubbleRob 的功能，以使他/她遵循地面上的規則。確保您已經閱讀並理解了第一個 BubbleRob 教程。本教程由 Eric Rohmer 提供。

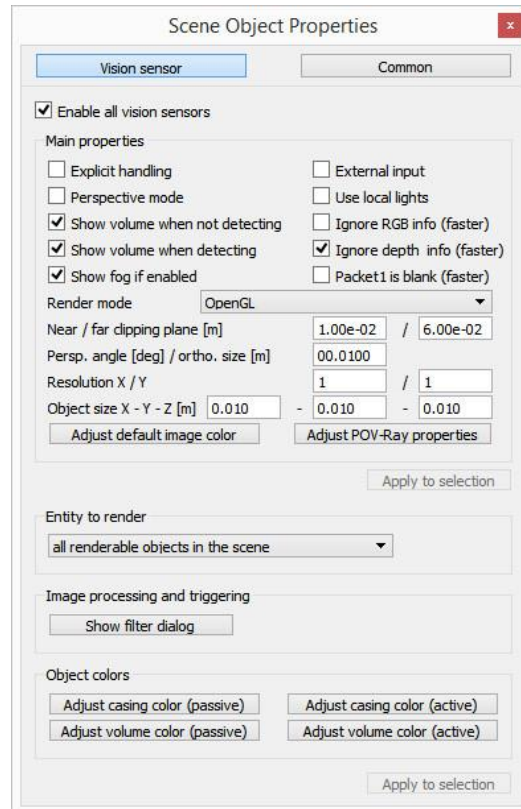
在 CoppeliaSim 的安裝文件夾中的 tutorials / BubbleRob 中加載第一個 BubbleRob 教程的場景。與本教程相關的場景文件位於 tutorials / LineFollowingBubbleRob 中。

下圖說明了我們將設計的仿真場景：



我們首先創建 3 個視覺傳感器中的第一個，並將其附加到 bubbleRob 對象。選擇[菜單欄->添加->視覺傳感器->正交類型]。通過雙擊場景層次中新創建的視覺傳感器圖標來編輯其屬性，並更改參數以反映以下對話框：

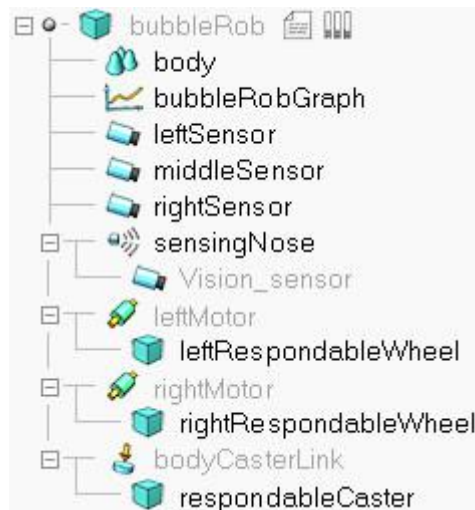




視覺傳感器必需面向地面，因此選擇它，然後在“方向”對話框中的“方向”選項上，將“Alpha-Beta-Gamma”項設置為[180; 0; 0]。

我們有幾種可能性可以讀取視覺傳感器。由於我們的視覺傳感器只有一個像素，並且操作簡單，因此我們只需查詢視覺傳感器讀取的圖像的平均強度值即可。對於更複雜的情況，我們可以設置視覺回調函數。現在，複製並貼上視覺傳感器兩次，並將其名稱調整為 **leftSensor**，**middleSensor** 和 **rightSensor**。將 **bubbleRob** 設置為其關係（即，將其附加到 **bubbleRob** 對象）。現在，您的傳感器在場景層次中應如下所示：





讓我們正確放置傳感器。為此使用位置對話框，在位置選項卡上，並設置以下絕對坐標：左傳感器：[0.2; 0.042; 0.018] 中間傳感器：[0.2; 0; 0.018] 右傳感器：[0.2; -0.042; 0.018]

現在讓我們修改環境。我們可以移去 **BubbleRob** 前面的幾個圓柱體。接下來，我們將構建機器人將嘗試遵循的路徑。從現在開始切換到頂視圖：通過頁面選擇器工具欄按鈕選擇頁面 4。然後點擊[菜單欄->添加->路徑->圓圈類型]。使用鼠標啟用對象移動。您可以通過兩種方式調整路徑的形狀：選擇路徑（並且只有路徑）後，按住 **Ctrl** 並點擊其控制點之一。然後可以將它們拖動到正確的位置。選擇路徑後，進入路徑編輯模式。在那裡，您可以靈活地調整各個路徑控制點。

一旦對路徑的幾何形狀滿意（您隨時可以在以後的階段對其進行修改），請選擇它，然後在路徑屬性中取消選中“顯示點的方向”，“顯示路徑線”和“顯示路徑上的當前位置”。然後點擊顯示路徑整形對話框。這將打開路徑整形對話框。點擊啟用路徑變更，將類型設置為水平線段，並將縮放因子設置為 **4.0**。最後將顏色調整為黑色。我們必須對路徑進行最後一個重要的調整：當前，路徑的 **z** 位置與地板的 **z** 位置重合。結果是有時我們會看到路徑，有時會看到地板（這種效果在 **OpenGL** 行話中被稱為“**z-fighting**”）。這不僅影響我們所看到的，而且還會影響視覺傳感器所看到的。為了避免與 **z** 戰鬥有關的問題，只需將路徑對象的位置向上移動 0.5 毫米即可。

最後一步是調整 **BubbleRob** 的控制器，使其也將遵循黑色路徑。打開附加到 **bubbleRob** 的子腳本，並將其替換為以下代碼：

```
function speedChange_callback(ui,id,newVal)
```

```

        speed=minMaxSpeed[1]+(minMaxSpeed[2]-
minMaxSpeed[1])*newVal/100
    end

function sysCall_init()
    -- This is executed exactly once, the first time this
    script is executed

    bubbleRobBase=sim.getObjectAssociatedWithScript(sim.handle_
self)
    leftMotor=sim.getObjectHandle("leftMotor")
    rightMotor=sim.getObjectHandle("rightMotor")
    noseSensor=sim.getObjectHandle("sensingNose")
    minMaxSpeed={50*math.pi/180,300*math.pi/180}
    backUntilTime=-1 -- Tells whether bubbleRob is in
forward or backward mode
    floorSensorHandles={-1,-1,-1}

    floorSensorHandles[1]=sim.getObjectHandle("leftSensor")

    floorSensorHandles[2]=sim.getObjectHandle("middleSensor")

    floorSensorHandles[3]=sim.getObjectHandle("rightSensor")
    -- Create the custom UI:
    xml = '<ui
title="'.sim.getObjectHandle(bubbleRobBase)..' speed"
closeable="false" resizeable="false"
activate="false">'..[[
        <hslider minimum="0" maximum="100"
onchange="speedChange_callback" id="1"/>
        <label text="" style="* {margin-left: 300px;}"/>
    </ui>
    ]]
    ui=simUI.recreate(xml)
    speed=(minMaxSpeed[1]+minMaxSpeed[2])*0.5
    simUI.setSliderValue(ui,1,100*(speed-
minMaxSpeed[1])/(minMaxSpeed[2]-minMaxSpeed[1]))
end

```

```

function sysCall_actuation()
    result=sim.readProximitySensor(noseSensor)
    if (result>0) then
backUntilTime=sim.getSimulationTime()+4 end

    -- read the line detection sensors:
    sensorReading={false,false,false}
    for i=1,3,1 do

result,data=sim.readVisionSensor(floorSensorHandles[i])
        if (result>=0) then
            sensorReading[i]=(data[11]<0.3) -- data[11] is
the average of intensity of the image
        end
        print(sensorReading[i])
    end

    -- compute left and right velocities to follow the
detected line:
    rightV=speed
    leftV=speed
    if sensorReading[1] then
        leftV=0.03*speed
    end
    if sensorReading[3] then
        rightV=0.03*speed
    end
    if sensorReading[1] and sensorReading[3] then
        backUntilTime=sim.getSimulationTime()+2
    end

    if (backUntilTime<sim.getSimulationTime()) then
        -- When in forward mode, we simply move forward at
the desired speed
        sim.setJointTargetVelocity(leftMotor,leftV)
        sim.setJointTargetVelocity(rightMotor,rightV)
    else

```

```

        -- When in backward mode, we simply backup in a
curve at reduced speed
        sim.setJointTargetVelocity(leftMotor,-speed/2)
        sim.setJointTargetVelocity(rightMotor,-speed/8)
    end
end

function sysCall_cleanup()
    simUI.destroy(ui)
end

```

您可以輕鬆地調試以下視覺傳感器的生產線：選擇一個，然後在場景視圖中選擇[右鍵->添加->浮動視圖]，然後在新添加的浮動視圖中選擇[右鍵->視圖-->將視圖與選定的視覺傳感器關聯]。

## CoppeliaSim

**外部控制器教程** 在 **CoppeliaSim** 中，有幾種方法可以控制機器人或仿真：最方便的方法是編寫一個子腳本來處理給定機器人或模型的行為。這是最方便的方法，因為子腳本直接附加到場景對象，它們將與相關的場景對象一起復制，它們不需要使用外部工具進行任何編譯，它們可以在線程或非線程模式下運行，它們可以通過自定義 **Lua** 函數或 **Lua** 擴展庫進行擴展。使用子腳本的另一個主要優點是：與本節中提到的後 3 種方法（即使用常規 **API**）一樣，沒有通信延遲，並且子腳本是應用程序主線程的一部分（固有的同步操作）。

可以控制機器人或模擬的另一種方法是編寫插件。插件機制允許使用回調機制，自定義 **Lua** 函數註冊，當然還可以訪問外部函數庫。插件通常與子腳本結合使用。使用插件的主要優勢還在於，與本節中提到的後 3 種方法（即使用常規 **API**）一樣，沒有通信延遲，並且插件是應用程序主線程的一部分（固有的同步操作）。插件的缺點是：它們的編程更加複雜，並且也需要使用外部編譯。

控制機器人或模擬的第三種方法是編寫依賴於遠程 **API** 的外部客戶端應用程序。如果您需要從外部應用程序，機器人或另一台計算機運行控制代碼，這是一種非常便捷的方法。遠程 **API** 有兩個版本：基於 **B0** 的遠程 **API** 和舊版遠程 **API**。

控制機器人或仿真的第五種方法是通過 **ROS** 節點。**ROS** 與遠程 **API** 相似，是使多個分佈式進程相互通信的便捷方法。儘管遠程 **API** 非常輕巧且快速，但

它僅允許與 Coppeliasim 通信。另一方面，ROS 允許幾乎將任意數量的進程相互連接，並且提供了大量兼容的庫。

控制機器人或模擬的第六種方法是通過 BlueZero (BØ) 節點。與 ROS 類似，BlueZero 是使多個分佈式進程相互通信的一種便捷方法。

控制機器人或模擬的第七種方法是編寫一個外部應用程序，該應用程序通過各種方式（例如管道，套接字，串行端口等）與 Coppeliasim 插件或 Coppeliasim 腳本進行通信。

# IndustrialAndEngineeringProductDesignCollaboration

翻譯文件擷取:

[IndustrialAndEngineeringProductDesignCollaboration.pdf](#)

協同產品設計過程

本研究旨在闡明工業設計師和工程設計師如何協作

這種聯盟如何反映設計過程。我們與來自六個消費產品製造商的 34 名工業和工程設計師進行了有關實際產品設計項目的深入採訪。

首先從訪問數據中識別出各個設計過程。其次，我們然後使用鑲嵌方法將設計過程進行比較並合併為協作過程。最後，我們簡化了協作流程，以創建具有代表性的流程模型。

結果，我們發現了四種類型的典型協同產品設計流程及其特徵：

E 類型 1：由 ID 主導的概念驅動過程，類型 2：由 ID 主導的內外聯合過程，類型 3：由 ED 主導的內在先流程，類型 4：ID & ED 協同流程。

## 工業和工程設計師之間

它經常被認為工程設計人員使用一個'向外接近 電子商務 發展從功能外觀"，而工業設計師遵循'向內方法 E 從外觀上的功能。

基於這些概念，提出了以下兩種設計策略：一種是"由內而外"的策略，定義為首先設計內部工作部件，從而限制了外部形狀，另一種是"由外而內"的策略。

這些設計策略涉及公司應在特定條件下為特定目的採取的組合設計過程。但是，從經驗證據中很少了解到這兩種策略如何在工業項目中應用。很少有嘗試從工程設計和工業設計的綜合角度來看待設計過程。

## 本文有兩個目標

闡明在實踐中應用的原始協作產品設計過程的形式；其次，確定在不同條件下用於不同目的的不同類型的過程。為此，我們對六家消費產品製造商的工業設計師和工程設計師進行了深入採訪。我們使用“確定了每個公司的產品設計過程鑲嵌法”，其中結合了從訪談數據中提取的各個設計過程以完成一個協作過程。結果，我們確定了四種典型的協作產品設計流程及其特徵。

本文包括三個部分：第一部分詳細介紹了研究方法。第二部分介紹了典型的協作產品設計過程及其特徵。最後，它總結了結果，並討論了對設計文獻的影響和貢獻。

Berends，Reymen，Stultiens 和 Peutz（2011）對五家公司的設計過程進行了調查。Kleinsmann 和 Valkenburg（2003）收集了有關工業中協同設計過程的故事，並確定了協同設計項目的關鍵主題和情節。

## 追溯性賬目的不可靠或不準確（Ackroyd & Hughes，1981 年）

我們採用了三個標準來選擇研究目標：（1）多家公司類似的產品領域；（2）工程和工業設計部門的多個參與者；（3）在整個產品開發週期中已經完成的項目。

## 大多數產品開發活動需要一組輸入

基於階段的模型由旨在執行任務的動作和旨在評估執行任務的結果的決策組成；其中完成構成進展到下一個階段

初步輸出，狀態報告等進行生產（Browning 等人，2006 年）。這一觀點得到了有關設計過程和組織的大量文獻的支持（例如，Burns & Stalker，1961；Clark，1991；Hubka & Eder，2012）。Ulrich 和 Eppinger（2012）將其描述為最終結果所必需的信息，該信息被注入以將當前設計問題轉換為下一個問題以降低不確定性，從而產生輸出，直到形成最終結果為止。這被建模為將功能規範（輸入）轉換為偽影規範（輸出）的過程。指出，從信息處理的角度進行過程建模無法捕獲完整的信息流，因為人們對交互的了解。他們認為，確定組織單位之間的相互作用的重要性，因為它們會引起建立可交付的流程模式的活

動之間的依賴。決定性，過程建模的在此透視基本元件被概括為)'的信息流與IPO（輸入端 E 方法 e 輸出）REP-resentation 和'相互作用組織單位的'（人，團隊，公司等”。

從狀態轉換的角度來看，Reymen（2001）將設計過程描述為通過在每個階段執行任務並評估設計活動將狀態從一種狀態轉移到另一種狀態的過程。這是台門車型，在設計活動發生在舞台的每一個階段，設計輸出的設計審查，從而作出決定之間是否進入下一個階段，重複當前階段或拒絕該項目的進行評估一致，大多數基於階段的模型都提出了這種觀點（Wynn & Clarkson，2005）。基於階段的模型由旨在執行任務的動作和旨在評估執行任務的結果的決策組成；其中完成構成進展到下一個階段（Jansch & Birkhofer，2006; Roozenburg & 交叉，1991）。除此之外，當從一個階段跳到另一個階段或在子過程之間鏈接時，伴隨輸入和輸出的信息流就會發生

“**任務**”來表示項目的初始起點，不同於在每個階段執行的“**任務**”。但是，這兩個術語的用法類似，用以表示執行設計工作，因此，人們對相互連接和處理任務的設計過程達成了普遍共識（Dorst，2008 年）。

從狀態轉換的角度來看，有兩個活動：執行任務和評估結果。通常，評估會在相對較短的時間內進行，以決定“前進，重複或下降”。因此，我們將這些類型的活動（評估，設計審查，檢查，決策）稱為“**事件**”。目前，我們有兩個編碼類別：“**任務**”和“**事件**”。總之，我們可以用四個編碼類別對設計過程進行建模。“**任務**”，“**事件**”，“**信息流**”和“**交互**”，並且可以用階段的連接來表示。此時，設計過程的一個階段可以用來表示'輸入端 E 任務（設計活性）E 事件（deci-錫永製作）在線輸出”。以下描述瞭如何使用以上四種編碼類別對過程元素進行編碼。

**任務**：清除設計人員執行的任務，例如“想法草圖”，“3D 建模”等。“是的，我做的草圖用圓珠筆來表達我的想法。對其進行少量著色或對其進行掃描以在 Photoshop 上進行修飾。'我們將其編碼為帶有“想法草圖”標籤的任務類別。 **事件**：完成特定任務後，將進行一個事件（決策），從而導致下一個動作。一位受訪者說；'我們去樣機公司製造原型，並帶走所有兩個或三個原型。與他們舉行了一次模擬評估會議。工程師們聽老闆的批評。老闆說就是這樣，那就是，如果工程師有不同的想法，他們會提供意見。最後一個被選中。我們在事件類別中使用標籤“模擬評估會議”對此進行了編碼。

**信息流**：當任務或事件發生時會生成信息流。它總是伴隨著輸入和輸出。一位受訪者說；實際上，我們根據工程設計人員提供的規範進行工作。直到有事情出現，我們才開始工作。這說明工業設計師從工程設計師那裡接收到了“規格數

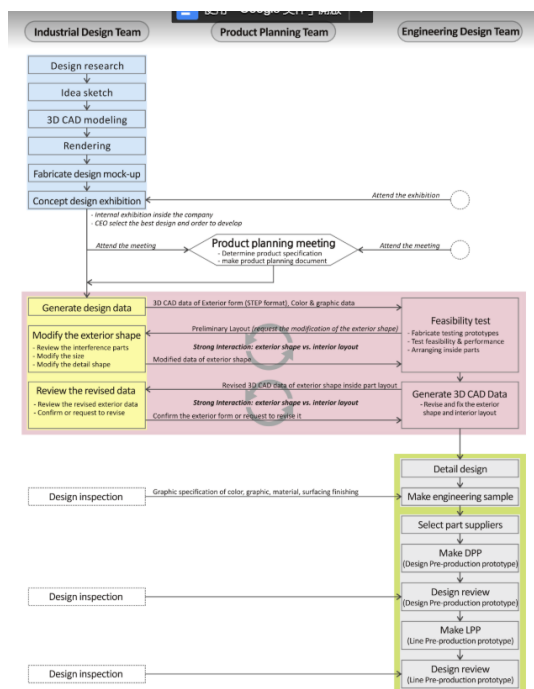


據“作為輸入，他們開始使用他們的工作。我們將此“規範數據從 ED 編碼為 ID”。另一個摘錄是：選擇一個後，我們會將模型的 3D 數據傳遞到工程部門。我們發送“STEP”文件，然後使用“UG”開始設計。這說明他們將 3D CAD 數據作為設計活動的輸出發送給了工程設計人員，這些輸入成為了後續工程設計活動的輸入。我們對“從 ID 到 ED 的 3D CAD 數據”進行了編碼。

交互：發生任務或事件時，兩組經常交互。例如，當工業設計師檢查是否可以修改內部空間時，工程設計師通過提供相關信息為他們提供建議。在這種情況下，會發生激烈的互動，工程設計師將充當“顧問”。情況就是這樣；“他們解釋說”這應該放在這裡和應擺在那裡”。然後，我們修改尺寸和位置，然後再次詢問它們是否可以減小 2 毫米的間隙。如果他們說“是”，我們會將一個部分壓平。我們將其編碼為“ED 到 ID 的顧問角色”。

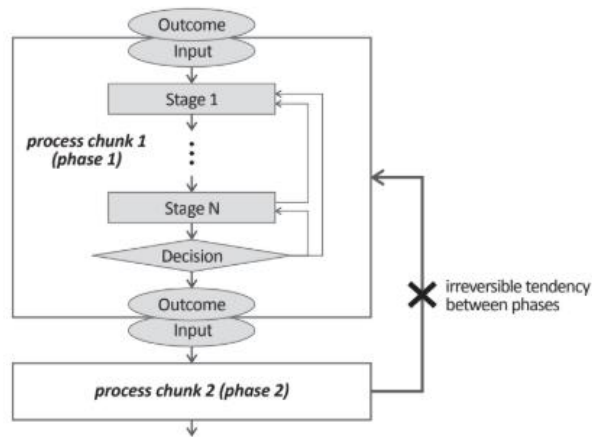
我們從每個轉錄的訪談數據中識別出流程要素。我們盡可能用參與者自己的單詞標記它們。我們將現場說明用作補充數據源。在此過程中，我們按時間順序排列並連接了編碼元素，以構建部分設計過程。該方法被廣泛採用以可視化工業過程模型。

## 輸出研究結果

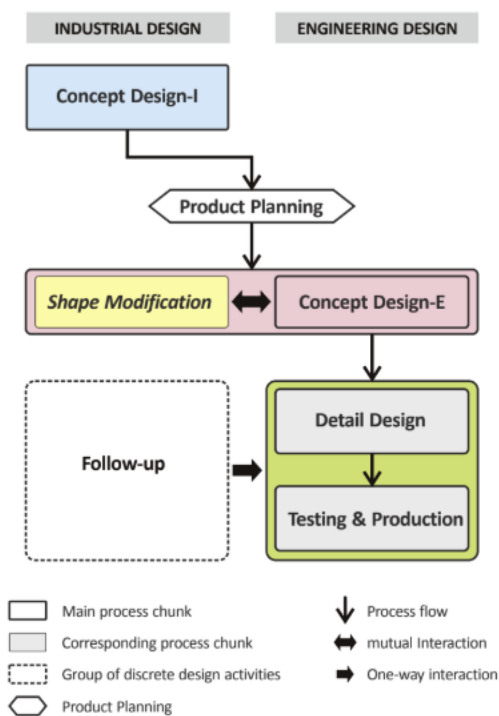


流程塊中的流程從初始輸入開始，並在內部的各個小階段中運行。迭代或反饋可能在塊內的各個小階段之間發生。最後，他們對最終結果做出最終決定。這是一個里程碑，指示該階段將跳到下一階段。例如，在圖 3 的藍色方框中，此

塊的主要工作是開發執行六個小階段的設計概念。當他們處於渲染階段時，如果在渲染評估事件中不滿意渲染結果，則可以返回到構思草圖階段。



在最後階段，設計模型作為結果由高層管理人員確認，然後跳至“產品計劃會議”。在前一個結果進入下一個塊的輸入之後，兩個過程塊之間的過程流幾乎沒有機會被逆轉。跨團隊通常進行最終設計，並由高層經理在每個工作組中批准。回到上一個區塊意味著它無法滿足市場時間表。因此，應該由最高管理層對此事做出決定。



成果輸入

Table 3 Four types of collaborative product design processes

| Company            | Type 1:<br>ID-led<br>Concept-driven<br>Process  | Type 2:<br>ID-led Combined<br>Outside-inside<br>Process | Type 3:<br>ED-led Inside-first<br>Process  | Type 4:<br>ID&ED Synergetic<br>Process                                |
|--------------------|---|---|--|---|
| A                  | ●   |   | ●  | ●   |
| B                  |   | ●   |  |   |
| C                  | ●   |   | ●  |   |
| D                  |   |   | ●  | ●   |
| E                  | ●   | ●   | ●  |   |
| F                  |   |   | ●  |   |
| Goal and Situation | New concept or when market changes fast, many reference products exist and lead-time is short | New concept or re-design                                | Re-design when clear target market exists  | New concept, spontaneously developed process by individual endeavours |
|                    | B2C: Company A & E<br>B2B: Company C  | All B2C   | B2C: Company A, D, E & F<br>B2B: Company C | All B2C   |

工業設計師的設計活動稱為“ Concept Design-I”，而工程設計師的設計活動稱為“ Concept Design-E”。

圖 5 是圖 3 的簡化版本。圖 3 中的彩色框使用了圖 5 中的相同彩色框進行了簡化。圖 3 中的藍色框變為“ Concept Design-I”。圖 3 中的“產品計劃會議”定義為“產品計劃”，並以細長的六邊形可視化。由於它是一個單獨且獨特的設計活動，因此被確定為單個階段。在紅色塊內部，有兩個並行的處理塊；右側的“ Concept Design-E”（紅色的小方框）是“主流程塊”，左側的“形狀修改”（黃色）是“對應的塊”。這兩個塊不視為單獨的階段，因為它們與紅色大塊內的起點到終點相連。這樣，兩個設計師團隊之間的互動就很強。為了區分兩個平行的塊，我們用粗實線表示主塊，並用淺實線表示相應的塊。在綠色塊中，由於所有 12 個流程之間的设计活動都沒有什麼區別，因此我們將詳細設計，測試和生產的兩個順序流程塊合併為一個。因此，在本研究中我們將它們視為一個階段。當綠色方框中的設計過程繼續進行時，工業設計師會根據工程設計師的要求或他們自己的設計檢查目的不時做出反應。他們通常將這些活動稱為“跟進”，與主要任務有所不同。這些活動顯然存在，但不屬於流程塊。它們是如圖 3 所示的離散過程元素，因此用虛線表示。為了幫助視覺理解，實心箭頭，雙向寬箭頭和單個方向寬箭頭分別表示過程流，相互交互和單向交互。

# DesignCollaborationBetterProducts

designcollaborationbetterproducts(資料)

通過協同設計創造更好的產品-心態，工具，過程

今日的數位產品比以往都更加複雜。創建它們需要多個團隊，成員，每個成員都有自己的一套技能和專業知識。以我們的客戶 **Savvy** 為例，與以下人員有緊密合作：產品經理，UX 設計人員，視覺設計師，開發人員，產品顧問和行銷人員。

我們使用協同設計來處理這種複雜的工作。並且正確完成，協同設計可幫助每個專家都有共同的心態，流程和工具，更快共同建立更好的產品，

我們編寫了此資源以幫助其他應用程式創建者採用協同設計文化和流程。學習為什麼協同設計對於創造引人注目的產品以及經驗很重要。我們還將深入探討協同設計的心態和文化、工具的要素，進行有效的協同設計，以及實際工作中的協同設計的例子。

## 什麼是協同"設計"

在進入正題之前，我們需要同時了解什麼是協同設計以及什麼不是。簡單來說，協同設計就是在設計優先的環境中進行協作。但由於設計的本質，特別是我們在 **Savvy** 所做的設計，設計協作趨向於超越人們通常認為的協作。

通常將協作定義為兩個或兩個以上的人共同完成一項任務  
為了實現共同的目標，協同設計甚至包含更多。

協同設計包括更多不同技術，艱鉅挑戰和更大規模的人員，這些都對目標具有深遠的影響。

您可能將協作視為兩個人用心在一個藝術品上。它只需要溝通、團隊合作同心協力。

協同設計是指一組策展人精心設計訪問者的體驗。他們選擇藝術品，選擇在何處以及如何懸掛藝術品，它所居住的房間（以及房間的照明，家具），甚至前面的房間和建築物同整為一個藝術品

## 為什麼協同設計很重要

你可以在沒有協作的情況下設計產品，但是需要一個協同設計過程才能完成好的產品。這就是協同設計如此重要的原因。

我們不僅為客戶構建應用程序。**Savvy** 構建了複雜的多平台體驗和內部系統，甚至在某些情況下還包括整個品牌和業務。為此，我們為了利用專業知識

跨足多個核心項目：策畫、設計、開發和行銷。每個項目都涉及了各成員各自的技能和專長。

協同設計將這些思想融合在一起，將他們各自的專業知識結合在一起創建解決方案，

以解決共同目標的所有方面。而不是以一個人遇到問題角度來看，

協同設計將這個問題擺在所有專家面前，迫使他們考慮新觀點和可能性。

例如，我們在設計過程的早期就讓我們的開發人員參與進來，以確保我們能盡快解決技術問題的機和局限性。

這使我們可以利用並提升我們的技術能力，同時進一步消除在處理程序成本高昂的障礙。

它還限制了一個人可以單獨處理工作的可能，從而使工作保持公開狀態並強調用戶需求和產品目標。

### 培養協同設計心態

在設計過程中建立有效的協同時，正確的思維方式有好的幫助。

它訂定了誰進行協作並如何進行以及達到何種目的的準則。

### 與每個人合作

設計師有時會認為不是設計師的人沒有資格提供良好的產品反饋。

但是出色的設計不只是視覺效果。它涵蓋了有關產品的所有內容，從它的品牌、工程技術及其成長

協同設計包含許多人，每個人都有自己的觀點和優勢。它給大家一個發聲的平台。

這些新鮮的觀點為設計師提供了更多信息。觀點使設計師能夠對其他設計有永久影響去選擇做出正確的開發和營銷選擇。

與客戶的協作與與其他設計師和團隊成員的協同一樣重要。

在 Savvy，我們讓客戶參與其產品的整個創作過程。我們花些時間來解釋選項，並與客戶進行討論以確保我們了解他們的想法。

這使我們能夠及早獲得客戶的認可，並從明確的起點和終點開始工作。

更大的曝光量並不是什麼大驚喜，第一次沒有人知道他們正在做什麼。

客戶合作意味著無需花費高昂成本、後退和不必要的會議就能更快地獲得更完整的產品。

設計很容易陷入角色而忽略全局。通過更早地參與更多的人與隊友建立更緊密的聯繫並共同承擔責任，以及對產品成功的期待。這些能夠創建一個更加開放和透明的流程以及更緊密聯繫和投入的團隊。

合作是持續性的

客戶向我們提出了複雜的挑戰和目標。當然不用說，我們也繼續努力與我們的客戶進行長達數月甚至數年的交流，並積累了過去的大量知識決策，研究和其他有價值的環境。

通過採用上下文優先的方法，你可以確保每個人都處於最高知識水平。這樣，他們就可以在產品上做出最明智的決定。為了做到這一點，在 **Savvy** 我們通過引入所有團隊成員儘早記錄和保存會議內容，建立一些重疊的地方讓下一個團隊成員在過程就像客戶一樣。

在個案協作的基礎上，上下文也同樣重要。請遵循以下準則以保持協作中的上下文優先：

在顯示你的工作之前提供脈絡。如果您正在工作，你的工作是否出現在螢幕上提供信息，你可能因為這樣失去了聽眾的注意力。

你必須描述您要解決的問題或要達到的目標。

介紹與當前問題相關的工作。解釋你的想法以及為什麼要這麼做的某些決定。

具體說明你要反饋的內容。

合作是開放、誠實和無畏的

將你的工作（自己）放在協同設計並不容易。情緒會阻礙提供開放和誠實的反饋，尤其是當你擔心會傷害他人的感受時。

在 Savvy 這並不意味著在協同中沒有情感的位置。感覺如何在設計中很重要。我們為有情緒在決策中有情緒的人創建一個環境。因為離開談話可能會抹消潛在的想法或解決方案。

充其量，僅提供完整圍繞事實和數據進行務實的討論是沒有的。

在最壞的情況下，它可能是誤導或提供虛假的敘述。

我們的團隊還相信，不要害怕提供反饋。這意味著了解需要解決的問題以及客戶的品牌和目標。

這也意味著要問很多提問以發現相關思想或幫助指導決策。

嘗試以探索性和指導性的方式提出反饋意見，以建立並改善工作而不是拆散。

您的反饋意見應具有建設性。與其說你不喜歡什麼，不如說反饋以指出您要幫助解決的問題。

在以下方面提供可行的步驟改善工作，或者至少改善你的思維背後的理由。

而且不要忘記表達你喜歡什麼以及為什麼。

合作不僅僅是新的想法

開放並接受協作反饋是一回事，解析這些想法是另一回事並將它們用作激發自己的催化劑。

這是一項一直不斷發展起來的能力。一種實踐方法是成為更好的聽眾。

通常，在與他人交談時，我們會更多地考慮接下來要說的内容比起別人在說什麼，這會影響反饋過程

尤其是在設計中，因為我們在聽到其他意見或解決方案時通常都知道自己要說什麼。

當您選擇先聆聽然後再做出反應時，它可以讓您完全理解反饋

有人在解說並讓你更深入他們使用的觀點是什麼和他們來自哪裡？

很有可能這是在你創建過程。通過聽取接受並理解反饋的上下文和推理，

正在向更多的視角，思維方式和體驗方式開放你的設計。

然後，您可以針對挑戰目標和用例測試這些新觀點。你可以設計用於它們是否更適合用戶。

當所有協作者都在積極聆聽時，更容易接受反饋。

最終，提供良好反饋的技能來自學習如何獲得反饋。當我們做一個

努力成為更好的聽眾，我們也發現自己變得更加謙虛，並成為更好設計師。

尋找和使用正確的設計協同工具



正確的工具在增強團隊的設計協同思維方面大有幫助。

在這部分，你會了解如何找尋協同工具。我們也會根據你的經驗適合的協同工具。

## 選擇合適的工具

有效的協同工具為協同者消除了所有障礙，使他們在工作上能夠快速輕鬆地連接與互動。

這將重點放在提供反饋上。它們還允許其他人在不破壞原始設計的情況下進行協同。

過去，我們使用依賴於提供基本版本控制的工具，而不是真正的協同功能。

例如，設計人員將保存一個 **Sketch** 文件並將其上傳到 **Dropbox**。

然後，另一個團隊成員將下載它，進行處理，然後重新上傳。文件在他人的手上時沒有簡單的方法進行更改。

我們嘗試了與 **Github** 類似的方法，實踐證明，該工具非常適合管理代碼庫，但在迭代設計工作中卻不那麼重要。

不用說，這些版本控制流程使我們的協同更加耗時、令人困惑，而且不合作性。

但是現在我們根據協同類型從多種更高級的工具中進行選擇並想要實現。

## FIGMA

這是一個協同優先的共享工作區工具。 **Figma** 非常適合在其中有多個人設計文件的相同區域。

你可以同時觀看成員設計或在同一設計上一起工作。

優點:

**Figma** 減少了個人在獨自一人工作的能力

無需添加不必要的修飾或創建靜態的可傳送結果實現協同。所以你無需更改工作流程即可展示設計。

它可以輕鬆地在本機環境中查看並與之互動，並隨意進行調整。

何時使用：文檔編輯，高保真原設計檔，即時協同和隨時改變的客戶想法，通過一系列螢幕的內容來解釋並獲得有關設計方向的反饋。

## MARVEL

雖然 Figma 傾向於感覺更自由和靈活，但 Marvel 允許更標準化的形式合作。這也使我們的客戶可以輕鬆地與我們合作。

優點:

Marvel 是一個更簡潔，更型式和專注的空間，非常適合與非設計合作團隊成員。

它剔除了客戶需帳戶的支付或為了進行工作必需深入了解工具的所有需求。

客戶可以下載視窗並通過 Marvel 在設備環境中查看它們的運行情況。

使用時間：與客戶和開發人員一起呈現更多最終設計工作。（學習比較 Marvel 與其他原型工具。）

其他協同工具

Zeplin 是一個有用的傳遞工具，可讓開發人員深入了解設計工作的細節。。（我們在這裡更深入地討論 Zeplin。）

Quip 是集思廣益和產品/程式文件的絕佳平台。我們用它來記錄並組織團隊成員在一個專案

工作時需要了解的所有背景和知識。對於集思廣益而不是視覺關注的新想法也很有用。

請注意，還有許多其他工具可以為協作增加類似的好處，上面列出的，該列表代表了在我們日常工作中對我們有效的協同工具，並不表示所有選項都可能對您的團隊有效。

Savvy 的程式設計在實踐中協同

現在，讓我們採用上面概述的最佳實踐和工具，並展示它們在現實生活中如何組合在一起。

我們將使用 Press Play 中的實時繪圖體驗來演示設計協同的重要性。這項經驗涉及到

跨學科的團隊成員數量，包括視覺設計師、UX 設計人員、開發人員、產品經理，當然還有客戶。

脈絡與挑戰

**Press Play** 是一個抽獎活動應用程序，可保存每日，每周和每月的圖紙。用戶獲得門票觀看廣告，然後選擇五個表情符號輸入繪圖。然後根據他們的選擇如何與圖形的隨機選擇表情符號匹配。這個特殊的任務有我們為正在等待現場繪畫結果的用戶創造了一個有趣而激動人心的現場體驗。

對於我們來說，為這種體驗創建類似遊戲的動畫非常重要。我們特別想要在用戶去看他們的屏幕時在螢幕上喚起一種嬉戲和期待感所選表情符號與抽獎活動附圖中的表情符號匹配。

就是說，我們需要注意動畫所需的複雜程度及其影響在整個產品的時間表和成本上。

我們目標是將保真度提高到合理的時間，而不會顯著影響項目的預算。

## 協同程式

### 階段一:設計原型與集思廣益

首先，一個 **Savvy** 的 UX 設計師創建了 **Press Play** 的整體用戶體驗和設計原型，確定了需要視窗畫面和每個視窗的時間安排。她還提出了一個粗略的概念，用於實時繪圖動畫，根據客戶的需求和項目的已建立 UX。

她向產品經理和視覺設計師展示了是設計原型和粗糙的動畫。然後，這三個人都與客戶會面，因此每個人都可以直接聽到反饋。

### 第二階段：研究與脈絡

**Savvy** 的視覺設計師負責創建實際的繪圖動畫帶來吸睛的感覺，讓 **Press Play** 產品沒有太多先入為主的知識。為了快速裝訂，他與 UX 設計人員和產品經理進行了深入的交談。他還致力於其他研究有時間了解整體產品目標，挑戰並熟悉工作。至今。如前所述，他是設計原型演示的一部分並為客戶的反饋。

在這種情況下，他進行了一些與手頭任務直接相關的其他研究。通過這樣做，他確保自己了解現場繪畫體驗的要求，目標和挑戰。他查看了具有類似經驗和保真度的其他應用，並參考了動畫，以了解最終動畫到底需要顯示什麼（在這種情況下，表情符號和用戶的表情符

號選擇)。

在過於束縛解決方案之前，他遇到了一個精明的人 iOS 開發人員了解技術限制和注意事項

然後，我們的視覺設計師和 UX 設計師集思廣益，對視覺效果至關重要。他們同意需要為用戶建立緩慢的提示以建立懸念/期望。

### 階段 3：反覆和反饋

當我們的視覺設計師沿多個不同方向工作時，他選擇了 UX 設計師聊聊他在 Figma 中的進度和設計。

通過討論工作，他們激發了更多想法和迭代，

同時確保它們符合客戶期望。還有更多他手下有很多可靠的選擇，

他再次與 iOS 開發人員會面，以確保一切都在從技術角度出發。

### 階段 4：客戶的反饋和發展

當我們找到更多，更最終的體驗版本時，視覺設計師走了通過他們與客戶一起使用 Figma。

Press Play 產品經理和 UX 設計師還提供了反饋和指導。

一旦他們都了解了引起顧客興趣的東西，視覺設計師便開始最大限度地發揮作用。

視覺效果並使它們為開發做好準備。他繼續與 iOS 開發人員合作，在技術層面上充分利用該概念。

### 最終結果

Press Play 的實時繪圖動畫是工作中設計協作的一個示例。一隊跨學科專家一起解決更大的設計和開發挑戰含義。

沒有設計協作，我們將找不到理想的交集用戶體驗，視覺和技術。

當客戶在他的應用程式中看到生動的動畫時，他稱其為“開創性”。

而且，這種 Press Play 體驗的研究，合作和創造對我們有幫助發現產品用戶旅程中的空白。最初，現場繪畫的目的是為了娛樂

向用戶顯示結果的方式。在設計過程中，我們意識到，

如果用戶不觀看實時繪圖，他們就會迷失方向，無法體驗終結繪製的用戶旅程。

基於這一發現，我們決定在其他兩個地方也改進該應用程序。我們添加了一個獲勝者圈子的結果/實時繪圖元素，以及用戶的輸贏歷史記錄部分應用程序的細節部分。最後，設計協作使我們得以實現這一未實現的目標需要用戶體驗。

## 結論說明

設計協同的產品和經驗需要解決建築中伴隨的複雜、關鍵問題。通過利用多個團隊成員的專業知識跨學科的設計協作可確保團隊從各個角度應對挑戰並尋求更好的解決方案。借助正確的思維方式、工具和流程，進行設計協同使團隊能夠通過創造性思維和迭代來更深入。

我們希望本指南為您提供良好的基礎，以便您可以構建自己的有效設計協同過程。

您可以在 [Savvy](#) 博客上了解有關設計和產品策略的更多信息，並隨時在 [Savvy Apps](#) 網站上與我們聯繫以尋求幫助。

# MechanicalDesignProcess thirth segment

## [MechanicalDesignProcess](#)

### 第 3 章 結構注意事項

在前面的章節中，我們定義了成功的設計是什麼，然後繼續確定將要在設計中放置的對象的位置。現在，我們將討論設計的結構性考慮。我覺得我們必須建立在“堅實的基礎”上，以便設計的其餘部分可以以此為基礎。電子外殼（本身）的結構必須堅固到足以在客戶（用戶）使用產品的各種環境中工作。因此，讓我們從討論提供產品的主要考慮因素入手。這個“堅實的基礎”。本章將著重於：

- 利用材料概念的優勢提出結構解決方案
- 定義用於考慮電子外殼結構設計的通用過程
- 查看一些具體說明一般概念的示例我們將在本章中最後一節標題為“獎金部分”。最後一部分旨在給我們在材料強度方面的問題增加一些複雜性，並展示除強度以外的其他考慮因素對於我們的設計選擇如何重要。

#### 3.1 簡介：材料的強度

結構注意事項對於設計的整體成功至關重要。讀者無需獲得機械工程學位或成為材料強度方面的專家即可從本章中受益，設計師對材料的基本強度了解得越多，外殼設計就越好。僅解決問題是不夠的，需要盡可能的以最具成本效益的方式解決問題。設計適用於電子外殼的結構的分為四個基本方法可基本方法：

- 1.查看已經存在的類似產品，並使用已設計的解決方案作為手頭設計的快速起點。這種方法的優點是速度快，但是缺點是您的設計可能會因為缺乏創造力而無法解決特定產品應解決的獨特問題。
- 2.快速的“信封背面”設計。這種方法在簡化的結構元素上使用了一些基本的設計方程式。我們將在本章稍後探討這些設計方法的一些示例，以及一些示例問題。
- 3.更複雜的分析。在 Sect 中對此進行了更多探討。
- 4.當然，過度設計並不是所有設計的正確答案。在競爭激烈的產品市場中，客戶主要根據價格做出購買決定，過度設計可能會導致產品成本增加（或者肯定會增加重量和尺寸）。結構上的過度設計基本上是從一個非常有可能成功通過結構測試的設計開始的，也就是說，要在客戶使用環境中免受衝擊和振動的影響就可以通過。

過度設計原因有以下幾個：

- 設計能夠承受所一些力的未知的高精度。這將在 Sect 中進一步探討。
- 設計中只有一個“安全係數”大於 1.0。安全係數等於 1.0 意味著您的設計僅符合設計標準。在 Sect 中討論了增加安全係數的設計。

### 3.2 結構的設計過程

我想給讀者一個通用的過程，以設計滿足設計要求的電子外殼（或外殼中的單個零件）。通過完成這六個步驟，設計人員應該準備好提出可以使用的材料和橫截面。我將分別細分這六個步驟。

#### 3.2.1 相似設計

業內其他設計如何處理相似情況？其他設計可能來自您自己公司內部的示例（過去的產品），也可能來自您自己公司外部的競爭產品。

#### 3.2.2 分力

確定物體上的力（靜態和動態）—振幅和方向這些力量。通常不考慮零件的自重電子外殼可承受靜態力，但可以考慮使用動態力。在本文中，我指的是“對象”，“零件”和“成員”，但它們都應被視為完全是同一個人。

### 3.2.3 現有的最終條件

確定對象的“最終條件”，即其移動自由度以及如何支撐構件。常見的最終條件是“固定”（不允許移動）或“自由”（允許旋轉）。最終條件會影響確定負載將產生的應力大小。

### 3.2.4 建議材料和橫截面

確定支撐這些材料所需的材料和橫截面組合力，請牢記“力量”屬於材料，力會在這些材料中產生應力。所有材料都有最大應力極限，我們要么開始變形或完全破壞點。

基本上，只有兩種最初選擇來設計更高的承重構件

- 更換材料，從而可以更改應力極限。因此，選擇具有較高應力極限的材料可以在該構件上施加更多的載荷。
- 更改材料的橫截面特性，基本上是構件的第二面積矩以及可以從構件的“中性軸”集中的面積。或質心。增大面積將實質上增加成員承載更多載荷的能力。遠離會員的“中性軸”增加該面積也將有助於會員承擔更多的負載（這就是“工字梁”的原因，“工字梁”的許多橫截面都遠離“中性軸”，是出色的承載構件）。

#### 3.2.4.2 鋁解決方案

設計就是要提出一些合理的選擇，所以讓我們看一下鋁樑。

我們可以選擇， $b = 4$  英寸， $h = 2.5$  英寸。我們可以看到最大應力為 23,100 psi。這高於鋁的最大屈服應力，因此不會造成結構變形。在我們的設計中令人非常滿意。但是要記住光束的高度是我們對慣性矩的計算  $b = 2.5$  英寸和  $h = 4$  英寸？這將是與鋁樑的先前示例。現在，最大壓力將是 14,400 psi，完全在該鋁的最大 22,000 psi 之內。因此，“旋轉-相同的橫截面，其中較厚的方向是在載荷方向上力，使這種材料選擇和橫截面在結構上成功。讓我們看一下鋁樑的重量。169 磅/英尺 3，光束會是 47 磅。相比之下，松木為 40 磅。

總之，我們研究了兩種不同的材料（松木和鋁最小）可用於解決結構問題。我們可以跨領域開發解決結構問題的每種材料的截面積。



在設計中，變形通常與強度同等重要。負載構件可能具有足夠的強度以承受特定的載荷，但是它可能偏轉超出工程材料彈性的不可接受的量。撓度（以及材料的彈性模量  $E$ ）也存在問題正在考慮中的內容將在本章中的一些示例中顯示。上述選擇的經濟性（更換物料或更換物料交叉部分）給 EPE 設計師帶來了一個有趣的問題。的許多組合材料和橫截面積可以使用，但必須根據實際情況進行選擇項目的總體目標。除了運作外，它還必須滿足項目的成本目標，可製造性，風險，重量，上市時間等。這些選擇將是進一步的在“第 4 章-可能有替代解決方案需要進行審查”開始進行調查，測試和製作原型。最大的資產之一設計師可以帶入設計中，以便快速找到要選擇的邏輯在可行的材料/橫截面選擇候選材料中做出解決眼前的問題。

### 3.2.5 合併功能

可以將正在設計的零件與裝配中的另一個零件組合嗎在這部分附近嗎？基本上，兩個單獨的部分可以合併成一個部分？查看被合併部分的“另類思維”方面是實際上希望從一個（設想的）單個零件創建兩個單獨的零件。這可能從而降低了總體成本，降低了組合設計的解決方案。主要選擇之一將是確定如何在生產中製造該解決方案。例如，一些這裡涉及的選擇是：

- 該項目的工具預算是多少？該項目可以“負擔”開支嗎？鑄造，注模，擠壓或其他製造所需的資金可能正在考慮的充氣技術？是否有現有工具可以利用？必須確定找到“投資回收期”解決方案。例如，了解：

- 1.加工費用
- 2.需要多少個零件（在產品“使用壽命”內）
- 3.未加工零件將花費多少
- 4.工具零件將花費多少

### 3.2.6 確定所需的安全

係數此時必須複查“安全係數”的確定。也就是說，必須知道以下問題的答案：

- 如果零件失效，是否有人受傷？生命，金錢和時間上無法預測的失敗的代價是什麼？
- 該特定部分在產品整體功能中的重要性如何？如果這部分失敗，那麼整個產品是否會失敗？

- 對力的了解程度如何？我們是否知道“誤差線”，即力會偏離假定的名義值多少？
- 確定所選設計（材料或幾何形狀）的“關鍵方面”，以及如何在生產中指定，認證和檢查它們？做筆記以確保將執行這些步驟（認證/檢查）。確定設計各個階段所需的測試，以確保最終設計足以將其交付給生產中的客戶。
- 將有一個優化的解決方案，通常可以通過分析設計的主要組成部分並確定設計中“弱鏈接”的位置來找到。這可以通過利用某些測試方法來發現，這些方法會通過超出環境限制的測試（例如高度加速的壽命測試，HALT）而導致失敗。通過首先確定可能發生故障的位置，然後通過測試設計原型，可以生成數據來確定某些段是否接近其設計極限。

如果設計過程中的上述六個步驟中的任何一個都沒有有一定程度的已知答案，則設計人員將面臨：

- 進行進一步查詢以獲取更好的信息。
- 進行設計。設計人員在設計過程的一開始就幾乎不了解零件的所有作用力和相互關係。當然，設計人員可以列出所做的假設以及必不可少的其他信息。當然，可以設計零件，對零件進行原型設計並在需要使用的條件下進行測試。解決“在不知道所有信息的情況下進行設計”這一難題的幾種方法可以被採取讓我們研究一個示例，其中：

設計 1 的重量為目標重量的 110%，但在結構上成功的可能性為 95%。設計 2 是目標重量的 100%，但 75% 在結構上成功的機會。因此，設計 1 超出了目標重量的 10%，但是無法達到通過設計工作的設計目標的風險要低得多結構的觀點。因此，“折衷”是優化設計所需的時間。當然，產品必須從結構的角度進行工作。很難確定在程序開始時就在設計中“預留空間”。繼續前進在不知道所有信息的情況下進行設計的價值在於“基本設計”可以測試。希望可以迅速修改“基本設計”框架，使程序可以在獲得其餘信息時繼續運行。我們可以通過“過度設計”零件來快速前進，或者投入更多時間來“勉強”滿足所有要求。仔細研究了這兩個路徑以下更多解答：

A.“過度設計”零件—這種方法可能保證零件會在測試中的結構功能。這裡的想法是迭代回到測試顯示出節省材料和減輕重量的地方，從而減少了保守的設計適當。這種方法至少使設計會議的機會最大化在項目測試階段的早期就對結構功能進行了要求等等。但是，設計的重量發生了變化，使這些零件更接近“mar-最終成功”將需要時間（和金錢）來重新測試設計以驗證更改。大多數項目在迭代方法上的時間有限獲得“完全”設計的零件。

B.設計零件時使用“僅少量”會議會花費更多時間兼顧重量和強度要求。因此，這種策略與過度設計（以上），因為零件的設計有可能（只是勉強）工作。例如，如果空間和重量減少最大產品要求列表，該設計在“基本”上可以接受結構強度因子，但具有更大的材料和重量節省，可能是需要什麼。這種方法試圖平衡“風險和回報”與應該徵得設計團隊的同意才能前進。有了這個設計可以達到材料和重量的目標。但是，這種設計的風險不大從結構上講，工作率從 5% 上升到 25%。因此“B”設計路徑顯示不滿足產品結構強度要求的較高風險，但將滿足產品的重量要求。

C.上述兩種方法的混合可能是合適的。也就是說，設計會比較保守，而設計的其他部分會更多有風險。這也許使“整體風險承受能力”成為整體風險的一部分。設計。經驗豐富的設計團隊將了解設計中的最佳位置“推翻接受度”。

### 3.3 所需的分析

當然，有許多設計可以保證在電子包裝設計中進行最嚴格的分析。在任何競爭激烈的產品設計領域，將是使用給定技術進行最高生產力的公司，即將大程度地提高其成功機會。如果產品具有：

- 高產量。如果要生產成千上萬個特定單元，則每單位節省一美元可以帶來可觀的總節省。節省大量成本的分析會由於大量生產而帶來大量總體利潤。但是，如果只生產幾個單元，則節省的潛力就大大降低了，一旦設計被認為是可行的，在降低成本上的大量投資就不會帶來可觀的節省。
- 由於要放置產品的環境，因此要求高度安全。例如運輸，公用事業，醫療或教育行業的產品。所有客戶都需要擁有安全操作的產品。
- 對客戶至關重要的“任務”。這通常包括軍事，航天局或政府所需的產品。

請注意，過分設計以致降低公司的盈利能力是沒有任何藉口的。設計師和工程師應該時刻警惕降低成本的可能性。零件的減少，可製造性的設計以及整體設計的優雅，導致了產品的領導地位。只有在設計的第一階段才能最大程度地降低成本。隨著設計甚至進入原型階段，為降低成本而重新設計的成本開始呈指數級增長。這方面的更多內容將在第一章中介紹。

另外，關於安全的注意事項也是適當的。在安全性很重要的任何領域，都不能為產品設計不足的藉口。當然，美國保險商實驗室（UL）和其他安全機構也會對電子設備進行安全考量認證。也就是說，安全機構將獲取產品（規格和工作單元），並對其進行審查和測試。大多數電子產品，當然是在全球範圍內銷售的產品，都必須通過嚴格的代理商批准認證。這方面的更多內容將在第一章中

介紹。首要的設計考慮因素仍然是，並且將永遠是功能。即，零件必須發揮作用 按預期。如果零件在負載下會失效，那麼它的外觀或生產的優雅程度都無關緊要。這是設計人員必須了解載荷的主要原因。

使用有限元分析（FEA）的現代分析軟件解決方案非常實用無處不在。在 Google 上進行的搜索顯示了一些入門資料，例如：

A.有限元分析，麻省理工學院的 David Roylance。描述三個主體步驟為：

- 預處理，將要分析的零件模型（其中的幾何形狀劃分為多個離散的子區域或“元素”）進行連接在離散點稱為“節點”
- 分析，將預處理器準備的數據集用作輸入線性或非線性代數方程組，用於計算應力和位移
- 後處理，以圖形方式顯示結果，以幫助可視化。分析結果

B.線性分析，來自 MIT 開放式課程中的 K. J. Bathe 教授，麻省理工學院。該視頻系列是全面的學習課程，介紹了有效的實體和結構線性分析的有限元程序。

C.有限元分析，H。J. Qi 博士。將 FEA 流程描述為：

- 制定物理模型，即描述（也許簡化）將實際工程問題轉化為 FEA 可以解決的問題
- 通過離散化實體，定義材料屬性來使用 FEA 模型，並應用邊界條件
- 選擇適當的近似函數，制定線性方程，並解決這些方程
- 以數字和視覺格式獲得結果

### 3.4 結構問題：靜載荷

- 問題是可以將其視為“個體”，即將載荷和力施加到 單個構件 要設計的上，也可以認為是“整體結構”，也就是說，它可以是對整個（組裝）結構的分析。
- 許多單個零件（或子裝配體）的供應商會在自己的文獻中提供設計指南，這些指南肯定可供單個設計師使用。以圖形或表格格式顯示的某些信息植根於基本原理，但這可能並不明顯。這種可用的“供應商數據”類型的一些示例是塑料，密封件，EMI 組件和軸承（僅舉幾例）的設計。

電子外殼中靜載荷是由於的構件上的 1.構件自身的重量 2.其他構件施加的載荷 3.由於熱效應，殘餘應力等 引起的載荷。由於施加力而導致以下操作失敗：

- 1.超出材料的屈服強度。
- 2.構件過度偏轉，導致構件超出設計意圖。

由於所有載荷都會產生一定的撓度，因此必須在設計中的某個時刻知道允許構件產生多少撓度。

靜態載荷下的斷裂破壞在韌性材料中不如在脆性材料中常見。在延性構件中，失效通常是由於過度的非彈性作用而導致的，該過度的彈性作用導致在斷裂之前很長的整體變形。

動態負載通常是隨時間變化的負載，而靜態負載在相對較短的時間內不會發生顯著變化。電子外殼設計中常見的一些動態載荷是反復載荷，衝擊載荷和能量載荷。能量負荷是指在撞擊期間傳輸的能量（比作用力）更容易表達的負荷。

重複載荷造成的斷裂通常稱為“疲勞”故障。振動可能會導致疲勞失效。以下三個問題通常涵蓋以下主題：

- 1.什麼是光束（相對於平板）？
- 2.應力公式和最大應力。
- 3.撓度公式和最大撓度。
- 4.截面模量。
- 5.彈性模量。
- 6.結束條件。
- 7.負載條件。
- 8.最壞的情況下加載。
- 9.組合加載。

### **3.4.1 懸臂梁分析**

本節重要性在於強調了橫向剪應力對最大應力的影響。在確定樑的最大應力時，除非用盡所有可能產生最大主應力的彎曲應力和剪切應力組合，否則不要對結果感到滿意。通常，剪切圖和彎矩圖的構造以及彎曲應力和橫向剪切應力的數量級比較將大大簡化問題。

使用梁應力公式，我們將繼續解決電子外殼設計人員可能面臨的問題的“主推力”。那就是確定“蓋板”到外殼底盤的（最大）緊固件距離。這種類型的問題涉及沿外殼提供的環境密封

- 防止灰塵，濕氣和蒸汽
- 足夠的 EMI 屏蔽

我們將在第 4 章中解決問題的屏蔽部分。現在，我們將解決設計基本密封設計幾何形狀方面的“結構性問題”，以保持足夠的強度以提供防潮密封。環境密封問題的結構考慮：

- A.密封材料：“產品環境（密封）”
- B.蓋板和底盤材料：此處涵蓋的彈性模量
- C.此處覆蓋的截面積（需要慣性矩）
- D.此處覆蓋的螺栓間距
- E.此處覆蓋的壓縮擋塊

關於方程式（和答案）的其他一些觀察結果是：

- 1.我們發現，鋼製外殼的螺栓間距大於鋁製外殼的螺栓間距—這意味著較硬的材料將允許較少的撓曲。
- 2.我們看到螺栓間距隨著厚度的三次方而變化—我們期望方程式（用於螺栓間距）可能基於“梁”的慣性矩，從而得出厚度的“立方體函數”。
- 3.我們希望螺栓間距是“ $1/4$  功率”的函數，這是梁在其長度上具有均勻載荷的撓曲的通用方程，該梁的撓度是其長度到四次冪的函數。對於載荷均勻的梁，相同的一般方程式也將根據其材料彈性模量（E）向  $1/4$  屈光力（ $3^{1/4} = 1.3$ ，即  $78.5 / 60$ ）偏轉。

因此，作為設計者，我們首先要估算對於鋁製外殼設計。顯然，我們可以在設計中對此間距進行原型設計，並在盡可能真實的條件下進行測試。值得注意的是，我們還假設了墊圈密封區域的橫截面面積以及墊圈在以下方面的厚度變化：

A.未壓縮狀態（在擰緊緊固件之前）。

B.壓縮狀態（在擰緊緊固件以在設計中設置“止動件”之後，即，緊固件附近的設計特徵專門限制了墊圈的使用範圍）

被過度壓縮。所有墊圈都要求這些“止動”以使緊固件具有指定的壓縮極限。我們還可以查看類似的設計，其中防護等級（空氣或水）與我們設計的水平相匹配。如果我們發現本身設計的尺寸適用於這些設計，則可以我們一些信心，我們當然有成功的機會，還應當指出的是 的因素之一， 整體 設計將有的 最低緊固件量。因此，緊固件之間的 5.0 英寸的距離會比 4.0 更好英寸的距離（從而節省了緊固件並節省了擰緊這些緊固件的工作量）。但是，4.0 英寸的間距將增加墊圈設計在不屬於一部分的附加載荷下密封的可能性。計算（例如衝擊或熱），從而為設計提供一定的安全裕度。

### 3.4.2 撓度公式和最大撓度

我們知道在“無肋”結構上增加肋骨將增加該結構處理更多載荷的能力。通常，可以通過在“常規”壁厚上增加厚度來增加強度，例如：

$\sigma = Mc / I$ ，其中：

$\sigma$ =所考慮構件的應力

$M$  =該構件的最大力矩（通常是力乘以“距離”，即是“的距離”從力到所述構件的部分）

$I / C = Z$ ，這是所考慮的部分的性質，也稱為從中性軸截面模量

$C$  =距離成員到“外部纖維”的

$I$  =成員的慣性矩（大約質心）成員因此，要增加的承載能力，您可以：增加  $I$  和/或減少  $c$ （增加  $Z$ ）。在  $I$  為矩形（長方形是用於製造的構件的共同選擇）其中  $b$  =的長度矩形的底邊， $h$  是矩形的 厚度 。請注意，，增加厚度（ $h$  由於“平方函數”）具有很大的影響。因此，將厚度加倍實際上使光束強度提高四倍。上述說來，加倍厚度將增加構件的重量。對於重量敏感型設計而言，這可能是“災難”。但是，通過添加肋骨，肋骨是厚度的“間斷性”添加，強度顯著提高（而重量僅少量增加）。為什麼是這樣？

儘管肋骨會增加板的總慣性矩，但對於短肋骨，從中性軸到橫截面（ $c$ ）的極端纖維的距離會更快地增加。對於間距較大的肋骨，此效果最為明顯。



### 3.4.3 另一個撓曲問題，這次是卡扣鉤

這裡的想法是，鉤在連接操作過程中短暫偏轉，並卡入底切以完成配合操作。

選擇了以下介紹性問題：

- A.顯示了一種通用的緊固方法（用於塑料）。
- B.顯示了使用材料共同強度的公式，該公式利用了材料的彈性特性，在設計中將撓度用作優點，並優化了橫截面積和均勻應變。
- C.介紹了使用塑料材料進行設計的某些方面。
- D.利用供應商提供的文獻資料（在本例中為 **Mobay Plastics**）。代替使用第一性原理解決一些更複雜（至今）的問題，使用列表選項和列線圖可以大大減少所需的設計時間。

### 3.5 動態載荷

電子機櫃中構件的動態載荷是由於載荷以非穩態方式承受在構件上的。它們包括但不限於：

- A.具有振幅和頻率的振動載荷（包括風力或與地震地面運動相關的慣性力）
- B.離散衝擊載荷在這些振動和衝擊載荷情況下，將探討一些問題。

**章節摘要** 在本章中，介紹了有關外殼結構方面的一些基本考慮。我們可以通過為這些外部船體提供材料來開始這種設計。同樣，我們的設計處置將是選擇船體的橫截面。這些橫截面和材料的最佳選擇是利用易於獲得的材料方程式的強度來進行的。但是，在各種解決方案中都需要做出選擇，並且要確定最佳設計，不僅要單獨考慮結構，還需要更多考慮。

此外，我們還介紹了一種用於設計電子外殼結構的通用過程。首先要看先前的設計，確定結構上的力，然後繼續確定設計中的安全因素。

**參考文獻：**

1. Roark RJ, Young WC (1975) Formulas for stress and strain. McGraw-Hill Book Co., New York

2. Byars EF, Snyder RD (1969) Engineering mechanics of deformable bodies. International

Textbook Co., Scranton

3. Design guide to the selection and application of EMI shielding materials. TECKNIT, EMI

Shielding Products (1991)

4. Injection Molding magazine, May 1998 issue, R. Cramer of Dow Materials Engineering Center

# MechanicalDesignProcess first segment

## 第 1 章成功的設計

### 1.1 設計指南

- 設計指南

助於擴大設計師過去的全部經驗以及他們現在的組織的建立流程。

- 電子產品外殼（**EPE** = 電子產品外殼）

由內部和外部結構組成產品的要素。

- **EPE 設計師**□

理解並符合產品規格的能力。□ 能夠添加並幫助創建產品規格。□ 為產品提出的問題創建創造性的解決方案。

### 1.2 定義整體團隊

團隊可以是：□

- 小型或大型。□
- 位於一個位置或分佈在全球。□

- 資源有限或可以訪問幾乎無限的資源。□
- 是否擁有最新工具。□
- 出於各種原因而實現目標的動機。□
- 經驗豐富。

整個工程包含著多個學科之間的設計融合。例如：軟件和硬體工程、機械工程（包括結構和熱學）、系統工程 ... 。

因此，知道機械工程只是電子產品整體工程設計的一部分，並且許多決策是與其他學科合作而成的。這些都是為了強調機械設計不是“憑空”完成，而是作為團隊共同努力的結果。

### 1.3 產品要求

- 產品將為客戶帶來什麼成就。
- 說明產品如何與客戶互動。
- 產品各組件的描述。
- 展示產品的最終外觀。
- 產品將在運行中和放置的環境，包括：溫度、衝擊、跌落、振動、濕度...。
- 產品需要通過測試達到所有標準。包括安全和法規標準，例如：安全性保險商實驗室（UL）、電動勢電磁干擾（EMI）...。

#### 1.3.1 規範功能

- 最少的輸入。
- 完整規格。
- 工作規範。

#### 1.3.2 準時按計劃進行

#### 1.3.3 交貨的預計成本

- 修改完成項目所需的人力資源。
- 完成的項目所需資源被修改。
- 修訂產品的預期銷售成本。

### 1.4 草繪與細節

EPE 設計人員必須知道何時在以下兩種模式之間切換：

- **草繪或集思廣益：**快速思考產品如何設計，同時獲得他人的反饋更改選擇。規模並不是很重要，速度才是這裡的重點。
- **詳圖：**在“詳細模式”下，可以準確計算出設計中的一些關鍵部分

## 1.5 設計評論

正式的設計評審通常是按照項目時間表上的計劃進行。

非正式的設計評審是偶爾或自發進行的。就像設計師去辦公室並要求同事“看看”、在設計師認為足夠接近設計問題或已有經驗的少數人中召開簡短會議。

## 1.6 溝通

溝通是設計成功的必要條件。這句話本身是正確的，但值得進行更充分的討論。

### 1.6.1 溝通目的

交流的目的是傳達有關設計的信息。

### 1.6.2 溝通的價值

良好的溝通將使項目更加成功。

### 1.6.3 通信鏈中的連接

交流的目標之一是將信息提供給需要的人，團隊包括很多學科。EPE 設計師應在團隊中涉及所有人員做出決定，這些人對於做出決定非常重要。

## 1.7 工程經濟

不論工程經濟學的基本原理，就不可能完成設計。幾乎所有的工作中，“付出什麼代價”都是最重要的考慮因素。在許多產品設計中，它將成為此產品成功或失敗的主要原因之一。

## 章節總結

介紹如何將設計視為“成功”，以及設計如何滿足產品要求。

## 參考文件

1. Thuesen HG, Fabrycky WJ, Thuesen GJ. Engineering Economy, 1971, Prentice-Hall

# MSModelingAndTFApproach

## 第一章 介紹

在過去的幾十年中，我們在電子技術方面所取得的進步和小型化已使工程師能夠採用新產品和新工程學科。在十八世紀初，我們已經看到了結合機械零件和電子零件的新產品的推出。機電一體化應用蓬勃發展的另一個因素是電子零件的價格不斷下降，以及設計超小型系統的挑戰。如今，例如具有高性能的微處理器變得非常便宜，這鼓勵了它們在計算機控制系統中的使用。

### 1-1 機械零件設計

機械部分是機電系統中的基本部分。在該部分的階段設計中，我們將構思和製造組成機電一體化系統的部分。還選擇用於該機電系統的致動器和傳感器。致動器和傳感器的機械部分的設計或選擇都是通過遵循將在本卷的下一章介紹的一些設計規則來完成的。同樣重要的是要記住，機電系統一旦循環使用就成為過時的觀點，這是我們在設計階段必須考慮的重要事項。在設計階段，也應考慮出於維護或其他目的而對系統進行的組裝和拆卸。

### 1-2 電子電路設計

在電子部分，工程師必須設計電路，以確保機電一體化系統的功能。它涵蓋了所需的電子部件的集成，例如電阻器，電容器，集成電路以及所選的微控制器或微控制器。不同組件所需的穩壓電壓也是該步驟的一部分。電子電路的主要部分是微控制器或一組微控制器。在本卷中，我們決定使用一種類型的單片機，它是 Microchip 製造的 dsPIC30F4011。我們沒有給出任何正當的理由，但我們的願望是採用一個微控制器來處理本卷將要介紹的所有示例。因為我們將對所有示例使用相同的結構，所以該選擇還將使讀者容易實現實時實現。

### 1-3 適時實施

在控制部分，工程師必須分析正在研究的系統，並設計適當的控制器以獲得所需的性能。在分析部分，我們應該從建立一個可接受的模型開始，該模型

給出輸入與輸出之間的關係。一旦掌握了動態特性，便會選擇一個採樣週期，並將模型轉換為離散時間形式，然後可以從經典比例積分和微分（PID）控制器，狀態反饋控制器或任何其他可以提供給控制器的控制器中選擇一個合適的控制器。

## 1-4 本書的構造

在傳遞函數方法部分中，將連續時間系統模型轉換為離散時間系統，並建立了不同的分析技術，並綜合了控制器，以確保達到預期的性能。

在狀態空間方法部分，將連續時間系統的模型轉換為離散時間狀態空間表示，並開發了用於控制器分析和合成的不同技術，以確保開發出所需的性能。

## 第二章 機電一體化系統

### 2-1 機電一體化

一體化可以定義為涉及產品設計的跨學科工程，其功能依賴於通過控制方案連接的機械，電氣和電子組件的集成。

如今，機電一體化一詞已享譽全球，許多機電一體化系統都設計用於個人或商業用途。我們日常生活中使用的所有小工具實際上都是機電一體化系統。作為這些小工具的示例，我們提到了使用許多機電一體化系統的筆記本電腦和汽車。

### 2-2 機械零件

機械部分代表了我們正在嘗試設計的機電系統中的主要組件，它可以由現有組件製造或製造。

在本部分的階段設計中，應更加註意以下幾點：

- 價格
- 形狀
- 重量
- 尺寸

注意機電一體化系統的運行環境也很重要。這將幫助我們選擇合適的材料來製造機械零件。機電系統的維護也是一個關鍵點，這就是為什麼我們在設計階段要注意系統所有關鍵部分的可訪問性。在這一階段，考慮到所有機電系統

的回收將無濟於事，因為我們將無法尊重我們為新一代保護的環境，這一點也很重要。

## 2-3 傳感器

傳感器是每個機電系統的關鍵點。從某種意義上說，系統的眼睛可以用來檢測所有類型的變化並採取適當的措施。傳感器可以看作是將物理現象，位置，速度，加速度，溫度，流量等轉換為易於測量或處理的電信號的設備。傳感器由換能器和信號調節單元組成。如今，對於某些現象，存在許多可用於感測它們的傳感器，這使得選擇變得非常困難。即使是對於現場經驗豐富的人員，選擇合適的傳感器始終是一項艱鉅的任務。

## 2-4 執行器

致動器被定義為將某種功率（例如液壓或電力）轉換為線性或旋轉運動的設備。它們代表了手臂軟機電系統。

實際上，使用了不同類型的執行器，其中包括：

- 電動執行器
- 液壓執行器
- 和氣動執行器

一種電動執行器，可以將電能轉換為線性的旋轉運動。它們用於對機電一體化系統的機械部分進行定位或賦予其速度。

除了便宜和簡單之外，直流電動機還易於控制速度，位置和扭矩。儘管它們的同系交流電動機通常在速度控制方面很昂貴，但在低速運行時表現出一些不穩定，並且自身的定位控制也很差。

電動執行器通常是精確而靈活的。它們是精確定位機械零件或在需要時迅速發展力量的理想之選。它們的主要缺點是在其運行過程中需要使用帶電的冷卻系統。對它們進行良好設計和良好保護後，其維護工作將減少至滑動觸點或換向器的變化。如果未安裝保護裝置，大負載可能會燒毀電動執行器的繞組。

步進電機更適合控制不需要反饋的機械零件。這些類型的執行器通常用於開環控制和機械零件的定位。為此，發送一定數量的脈衝這些執行器用於激光打印機，傳真機和大多數計算機設備。

## 2.5 電子電路

電子電路是機電一體化系統的大腦。重新組合被動和集成電路旁的有源元件。它的作用是管理和協調以所需的方式確定組成系統的所有組件的功能。無源元件包括電阻器和電容器，而有源元件可以是執行所需功能的簡單二極管或晶體管或任何集成電路任務。

## 2.6 實時實施

下一步就是簽署我們應該執行的控制算法以確保系統將正確執行其設計任務。這樣的設計算法分為兩個步驟。第一個步驟包括建立數學模型-正確描述輸入與輸入之間的關係的模型系統的輸出。第二步，固定期望的性能，使用適當的技術設計控制器。

## 2.8 結論

機電一體化系統的不同組成部分我們提供了一些機電一體化系統，我們將在這裡使用它們來展示概念。

## 第三章 數學建模

### 3.1 基於物理定律的數學建模

為了說明如何應用此技術，讓我們考慮一定數量的動力系統。我們希望在速度或位置上控制的負載。

#### 3.1.1 傳遞函數的概念

如果我們使用初始條件等於零的拉普拉斯變換，我們將得到：

$$\begin{cases} U(s) = RI(s) + L_m sI(s) + K_w \Omega(s) \\ Js\Omega(s) = K_t I(s) - b\Omega(s) \end{cases} \quad (3.2)$$

結合這些關係和定義之間的傳遞函數速度  $\Omega(s)$  和電壓  $U(s)$ ，我們得到：

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K_t}{(Js + b)(L_m s + R) + K_t K_w} \quad (3.3)$$

如果電樞電感  $L_m$  可以忽略，則傳遞函數變為：



$$G(s) = \frac{K}{\tau s + 1} \quad (3.4)$$

$$K = \frac{K_t}{Rb + K_t K_w}$$

$$\tau = \frac{JR}{Rb + K_t K_w}$$

### 3.1.2 狀態空間描述

現在，如果讓  $x_1(t) = i(t)$ ， $x_2(t) = \omega(t)$  和  $y(t) = x_2(t)$ ，我們得到：

$$\begin{cases} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -\frac{R}{L_m} & -\frac{K_w}{L_m} \\ \frac{K_t}{J} & -\frac{b}{J} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L_m} \\ 0 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \end{cases} \quad (3.6)$$

給出以下標準形式：

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (3.7)$$

where

$$A = \begin{bmatrix} -\frac{R}{L_m} & -\frac{K_w}{L_m} \\ \frac{K_t}{J} & -\frac{b}{J} \end{bmatrix},$$

$$B = \begin{bmatrix} \frac{1}{L_m} \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

系統的這種數學形式在文獻中稱為狀態空間表示。

## 3.2 識別

從前面的示例中可以看出，我們可以用於分析和設計的數學模型並非易事，甚至如果我們可以從物理定律中獲得模型，則不同參數的值該模型可能無法獲得，因此分析模型毫無用處。

系統識別是一個過程，通過該過程數學描述從測試數據中提取動態系統。識別的目的是構造一個算法，該算法將允許根據觀察到的結果建立數學模型數據。

### 3.3 結論

在本章中，介紹了動力學系統的數學建模。並提出了使用物理定律生成模型的技術。並且需要實驗數據來建立所需的模型。

### 3.4 問題

1.在這個問題上，我們要求為直流電動機套件建立數學模型不忽略  $L$ 。我們要求建立：（一）傳遞函數（二）狀態空間描述

2.建立兩輪機器人的數學模型

3.建立懸浮系統的數學模型

4.考慮一個具有傳遞函數的動力學系統。寫 **Matlab** 程序，該程序生成一個數據序列  $(u(k), y(k))$ 。使用這個數據，在 **Matlab** 中編寫程序以識別系統並建立數學模型。比較兩個模型並得出結論。

5.在狀態空間描述中考慮動力系統。編寫 **Matlab** 程序圖生成適當的數據以使用狀態空間標識系統描述方法。使用這些數據編寫一個 **Matlab** 程序來建立一個狀態空間描述，並將其與原始描述進行比較。

## 第四章

### 4.1 導言

現在的單片機功能比較強大，價格也比較實惠。所以大部分的人會選則它。

### 4.4.

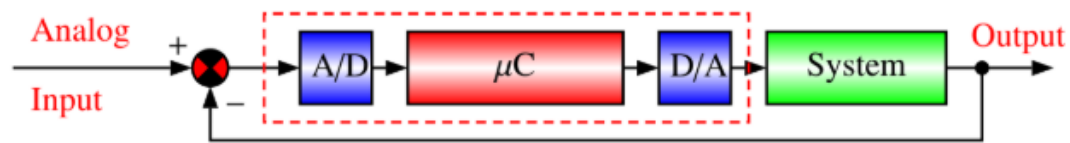
基於轉移函數分析的機電一體化系統。在開/關機的情況下，它用於安全和控制的目的。一般情況下不需要花時間就能計算出脈衝發生器的作用。而對於連續時間的情況，微控制器會重新在每個取樣期接收數據。

在實際使用微控制器控制實際過程時，有以下結構可以採用。

透過輸出通過 **A/D** 轉換為數字值。轉換器的輸出是通過 **A/D** 轉換為數字值。在這種情況下，參考是以數字形式送入。

這種結構用以下方式說明

圖 4.1 和 4.2。



在圖 4.2 的結構中，我們的採樣信號具有以下特徵

有以下優點。

- 1.易於調製
- 2.易於編碼
- 3.易傳輸、易再生並有積極意義。

#### 4.2. 取樣過程

在本章的其餘部分，我們將介紹可用於分析的工具。

這種類型的系統。主要是針對一個給定的實際系統。

- 1.如何確定採樣週期
- 2.如何將連續時間模型轉換為離散時間模型，使用所選的  
抽樣期
- 3.如何確定這類系統的性能，如穩定性、穩定性、穩定性等。  
過衝、沉澱時間等。

4.如何利用根-焦點和 Bode-plot 技術進行離散時間的案例分析。

連續時間和離散時間之間的關係建立。

主要是：兩個域（ $s$  域和  $S$  域）的兩極之間的關係已經建立。

第 3 節介紹了傳遞函數的概念和極點的概念。

在第 4 節中，對給定輸入的時間響應進行了開發，並給出了一個給定輸入的時間響應和

第 5 節介紹了計算它的方法。第 5 節介紹了穩定性問題和系統誤差。根定位和博德圖技術的發展再在第 6 和 7 節中具體介紹了這些技術。這些技術被用於分析和設計。

## 4.2 取樣過程

採樣期的選擇採用香農定理。這香農定理的聲明由以下結果給出。

### 4.2.1

定理 4.2.1 信號  $y(t)$  不含頻率成分大於  $f_h$  是由一組間隔為單一的  $y(t)$  取樣集唯一地重構出來適當的

採樣頻率的選擇應滿足：

採樣器的作用是將連續時間信號轉換為等效的連續時間信號。

脈衝序列，而 ZOH 阻斷了從採樣器接收到的值，從而使讀取這些數據的微控制器通過模擬/數字轉換器。

採樣過程如圖 4.3 所示

反向 Z 變換包括找到與  $f(k)$  相關的表達式。sponds to a given function  $F(z)$ . 是一個非常有用的方法是找到反轉函數的函數  $F(z)$  的擴展為部分分數，

其逆變換為可以在表中找到。這種方法的思路是，首先寫出了 expres-

$z^{-1}$  的函數，然後執行擴展為部分分數

如同通常對連續時間的情況下所做的那樣。這種技術可以從以下幾個方面來說明

以下為例：範例 4.2.7 讓我們考慮以下函數  $F(z)$

並確定  $f(k)$  的表達式。

採樣系統的傳遞函數的概念可以類似地定義為已經為連續時間。

為了澄清這一點，讓我們參考圖 4.10，其中上游採樣器是真實的，而下游採樣器是虛構的。

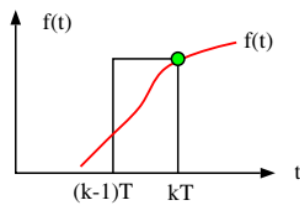
我們假定在同一採樣期同步的理想值。採樣期間，我們假設為理想值，並在同一採樣期同步。

第二個採樣器被引入，目的是定義  $Y(z)$ ，因此，定義

適當的脈衝傳遞函數。根據圖 4.10，我們可以得到:

因為輸出是由虛構的採樣器採樣，所以我們可以有

第二種數值積分方法：積分的近似值  
等於圖 4.8 的面積之一



**Fig. 4.8** Backward integration

按照之前相同的步驟，使用圖 4.8，我們得到：

$$y(kT) = y((k-1)T) + T [-ay(kT) + au(kT)]$$

依次給出  $z$  域:

$$Y(z) = z^{-1}Y(z) - aTY(z) + aTU(z)$$

從中我們有：

$$\frac{Y(z)}{U(z)} = \frac{aT}{1 + aT - z^{-1}} = \frac{a}{\frac{z-1}{zT} + a}$$

像之前一樣再次比較兩個傳遞函數，我們得到以下轉換：

$$s = \frac{z-1}{zT}$$

第三種數值積分方法：在前面的兩個模式中，我們有低估或高估曲線的面積。另一種替代包括計算這兩種方法的平均值。現在參考圖 4.9 所示，我們得到以下積分的近似值：

圖中所示區域的面積。

$$y(kT) = y((k-1)T) + \frac{T}{2} [f(kT) + f((k-1)T)]$$

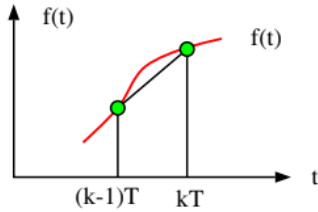


Fig. 4.9 Trapezoidal integration

從這個表達式中我們得到：

$$Y(z) = z^{-1}Y(z) + \frac{T}{2}F(z) + \frac{T}{2}z^{-1}F(z)$$

現在使用 F(z) 的表達式，我們得到：

$$Y(z) = z^{-1}Y(z) + \frac{T}{2}[-aY(z) + aU(z)] + \frac{T}{2}z^{-1}[-aY(z) + aU(z)]$$

最後

$$\frac{Y(z)}{U(z)} = \frac{a}{\frac{2}{T}\left(\frac{z-1}{z+1}\right) + a}$$

像之前一樣進行以下轉換：

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$$

範例 4.2.14 考慮以下傳遞函數：

$$\frac{Y(s)}{U(s)} = \frac{1}{s^2 + 0.4s + 0.4} = \frac{1}{(s + 0.2 + j0.6)(s + 0.2 - j0.6)}$$

我們的目標是看到我們將使用極點的轉換的效果系統的。首先，讓我們確定採樣週期。由於我們有一個二階，我們有：

$$w_n = \sqrt{0.4} = 0.6325 \text{ rad/s}$$

給出  $w_b = w_n$ ，並且採樣週期的適當選擇如下：

$$w_n = \sqrt{0.4} = 0.6325 \text{ rad/s}$$

為此，讓我們使用之前的變換計算極點 該系統：

$$T = \frac{2\pi}{30w_b} = 0.33 \text{ s}$$

使用  $s = z^{-1}/T$ ，相應的傳遞函數為：

$$G(z) = \frac{T^2}{z^2 + (-2 + 0.4T)z + 1 - 0.4T + 0.4T^2}$$

系統在  $z$  平面中的極點為： $z_{1,2} = 0.9338 \pm 0.1987j$

使用  $s = z^{-1}/Tz$ ，相應的傳遞函數為：

$$G(z) = \frac{T^2}{(1 + 0.4T + 0.4T^2)z^2 + (-2 - 0.4T)z + 1}$$

系統在  $z$  平面中的極點為： $z_{1,2} = 0.9064 \pm 0.1689j$

使用  $s = (2/T)(z^{-1}/z + 1)$ ，相應的傳遞函數為：

$$G(z) = \frac{0.25T^2(z+1)^2}{(1 + 0.2T + 0.1T^2)z^2 + (-2 + 0.2T^2)z + 1 - 0.2T + 0.1T^2}$$

系統在  $z$  平面中的極點為： $z_{1,2} = 0.9182 \pm 0.1845j$

使用變換  $s = 1$

$T \ln z$  ( $z = e^{Ts}$ )，極點為  $0.9175 \pm 0.1847j$ 。

從該示例可以看出，梯形近似為

因為它給出幾乎相同的極點，所以它更接近於精確變換。的

其他近似值則得出不同的結果。因此穩定性和精度

在選擇特定方法之前應先進行測試。

作為可用於近似傳遞函數的另一種方法



Z-域在文獻中始終被稱為極點/零點

轉型。它包括執行以下步驟：

$G(s)$  的所有極點都對應於  $z = e^{-sT}$ 。也就是說，如果  $s = -a$ ，則是極點在  $s$  域中，則  $G(z)$  在  $z$  域中的  $z = e^{-aT}$  處有一個極點

對  $G(s)$  的零做同樣的事情

將與  $s = \infty$  相對應的  $G(s)$  的所有極點置於  $z = -1$  處。這表示將  $(z + 1)$ ， $(z + 1)^2$ ，... 加到  $G(z)$  的分子上，使得分子將等於分母之一。

使  $G(s)$  的增益與  $G(z)$  之一相對應。這意味著我們必須為此，請執行以下操作：

$$[G(s)]_{s=0} = [G(z)]_{z=1}$$

範例 4.2.15 在說明這個過程是如何工作的，我們考慮以下內容轉換功能：

$$G(s) = 10/(s + 1)(s + 2).$$

該傳遞函數的極點為  $s_1 = -1$  和  $s_2 = -2$ 。他們對應極點分別是  $z_1 = e^{-T}$  和  $z_2 = e^{-2T}$ 。如果我們將採樣週期固定為  $T = 0.02s$ ，則這些極點變為  $z_1 = 0.9802$  和  $z_2 = 0.9608$ 。

由於分母為度 2，因此分子也應為度

2。為此，我們將分子  $(z + 1)^2$  添加到分子。

然後通過以下公式計算增益：

$$\left[ \frac{10}{(s+1)(s+2)} \right]_{s=0} = \left[ K \frac{(z+1)^2}{(z-0.9802)(z-0.9608)} \right]_{z=1}$$
$$1 = K \frac{4}{(0.285)(0.487)}$$

$$K = 0.0019$$

最後， $z$  域中的傳遞函數由下式給出：

$$G(z) = \frac{0.0019(z+1)^2}{(z-0.9802)(z-0.9608)}$$

作為另一種方法，可以從  $G(s) = N(s) / D(s)$  導出  $G(z)$ ，當  $D(s)$  有不同的根源。可以使用以下公式進行計算：

$$G(z) = \sum_{n=1}^p \frac{N(x_n)}{D'(x_n)} \frac{z}{z - e^{x_n T}}$$

$$\text{with } D'(x_n) = \left. \frac{\partial D}{\partial s} \right|_{s=x_n} \text{ for } n = 1, 2, 3, \dots, p$$

範例 4.2.16 展示如何從帶有分母的  $G(s)$  中獲得  $G(z)$  的想法

具有不同的根源，我們考慮以下傳遞函數：

$$G(s) = \frac{1}{(s+a)(s+b)} = \frac{1}{s^2 + (a+b)s + ab}$$

該傳遞函數的分母和分子由下式給出：

$$D(s) = (s+a)(s+b)$$

$$N(s) = 1$$

關於  $s$  的分母導數由給出：

$$D'(s) = 2s + (a+b)$$

在兩個根上的導數的值是：

$$D'(x_1 = -a) = b-a$$

$$D'(x_2 = -b) = -(b-a)$$

使用這個和前面的公式，我們得到：

$$G(z) = \frac{1}{b-a} \frac{z}{z-e^{-aT}} - \frac{1}{b-a} \frac{z}{z-e^{-bT}} = \left( \frac{1}{b-a} \right) \left[ \frac{z}{z-e^{-aT}} - \frac{z}{z-e^{-bT}} \right]$$

### 4.3 傳遞函數概念

採樣系統的傳遞函數的概念可以類似地定義

已經做過連續時間了。為了澄清這一點，讓我們參考圖 4.10

上游採樣器是真實的，而下游採樣器是虛擬的

我們假設是理想的，並且在相同的採樣週期內保持同步。的

引入第二個採樣器是為了定義  $Y(z)$ ，從而定義

正確的脈衝傳遞功能。根據圖 4.10，我們得到：

$$Y(s) = G(s)U^*(s)$$

由於輸出是由虛擬採樣器採樣的，因此我們可以得到：

$$\begin{aligned} Y^*(s) &= [G(s)U^*(s)]^* \\ &= G^*(s)U^*(s) \end{aligned}$$

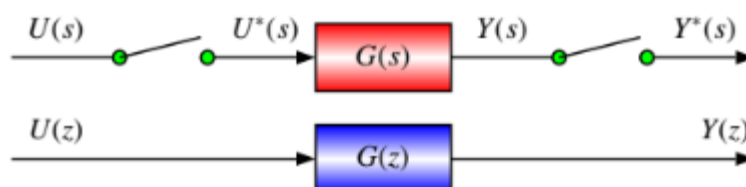


Fig. 4.10 Pulse transfer function definition

如果我們應用  $Z$  變換，則可以獲得：

$$Y(z) = G(z)U(z)$$

從時域開始可以很好地證明這種關係。在事實上，我們有：

現在使用卷積定理，我們得到：

從另一面我們知道你（ $\sigma$ ）可以寫成：

使用此， $y(t)$  的表達式變為：

現在使用採樣信號  $y$  的  $Z$  變換的定義（ $t$ ）我們有：

執行變量  $m = k - l$  的更改，我們得到：

可以重寫如下：

最後，傳遞函數由下式給出：

這是輸出的  $Z$  變換與輸出的  $Z$  變換之間的比率輸入。在操作採樣系統的框圖時，應格外小心。以下關係將有助於此目的。

**例 4.3.1** 在本例中，我們考慮圖 4.11 的系統。以理想的採樣器串行發送兩個系統。兩者的表達傳遞函數是：

我們的目標是為該系統計算等效傳遞函數。

根據此圖，我們得到：

這使：

這又意味著：

使用  $Z$ -transform 表，我們有：

**例 4.3.2** 在本例中，我們考慮重新採樣的情況。在兩個傳遞函數之間串行移動。這種情況說明如下圖 4.12。傳遞函數  $G1(s)$  和  $G2(s)$  由下式給出表達：

其中  $a$  為正標量。

我們的目標是計算等效傳遞函數並將其與在前面的示例中獲得的一個。

在這種情況下，我們有：

依次給出

重要的是要注意，我們為此獲得的等效傳遞函數  
這種情況與我們從上一個示例的系統獲得的情況不同。

使用  $G1(s)$  和  $G2(s)$  的表達式，我們得到：

根據 Z -transform 表，我們有：

例 4.3.3 在本例中，我們考慮具有轉移功能的情況。反饋中的內容，我們將按照  
與在先前的例子。該系統如圖 4.13 所示。傳遞函數由以下表達式給出：

根據此圖，我們可以：

依次給出：

由此我們得到：

這提供了以下脈衝傳遞函數：

從 Z -transform 表中，我們得到：

使用此我們可以獲得：

例 4.3.4 作為前一種情況的第二個例子，讓我們考慮一下系統圖 4.14。問題是  
如何計算脈衝傳遞函數該系統的  $F(z) = Y(z) / G(z)$ 。

由於（請參閱表 Z -transform）

對於閉環脈衝傳遞函數，我們得到以下表達式：

例 4.3.5 在本例中，圖 4.15 所示的系統為零使用訂單保持（ZOH）。

1. 找到開環和閉環脈衝傳遞函數  $Y(z) / U(z)$

2. 如果  $K = 1$  且  $T = 0.1$ ，則找到單位階躍響應

此示例的解決方案可以輕鬆獲得。實際上，我們有：

開環：

從中我們有：

最後我們得到：

閉環：

使用  $z_1 = 1$  和  $z_2 = 1-KT$  的殘差法，以及

$K = 1$ ，我們發現：

$k$  倒  $k = 0、1、2、3、\dots$

如果使用  $T = 0.1s$ ，則得到：

示例 4.3.6 讓我們考慮圖 4.16 的系統併計算轉移功能。使用此圖，我們可以：

依次給出：

使用  $Z$  變換，我們獲得：

例 4.3.7 讓我們考慮圖 4.17 的系統併計算轉移功能。

圖 4.17 反饋中的傳遞函數

使用此圖，我們可以：

依次給出：

現在使用  $Z$  變換，我們得到：

示例 4.3.8 讓我們考慮框圖的動態系統  
如圖 4.18 所示

例 4.3.9 讓我們考慮圖 4.19 的框圖系統  
併計算傳遞函數

例 4.3.10 讓我們考慮圖 4.20 的框圖系統  
併計算傳遞函數

圖 4.20 反饋中的傳遞函數

使用此圖，我們可以：

依次給出：

現在使用  $Z$  變換，我們得到：

根據這些示例，我們始終能夠計算出該系統及其表達方式如下：

其中  $Y(z)$  和  $U(z)$  分別是輸出  $Y(s)$  和  $Z$  的  $Z$  變換輸入  $U(s)$ 。此傳遞函數始終採用以下形式：

其中  $a_i$  和  $b_i$  是實數標量， $n$  是一個整數，表示整數的階數系統。  
多項式  $N(z)$  和  $D(z)$  的根，即以下項的解等式：

分別稱為系統的零點和極點。極點在系統響應中起重要作用。他們的位置非常重要，它與系統性能（例如穩定性，瞬態）有關  
政權等）。

例 4.3.11 讓我們考慮一個具有以下轉移的動力系統  
功能：

計算系統的極點和零點並將它們繪製在  $z$  域中。  
根據傳遞函數的表達式，我們有：

多項式的根對零為  $0.1 \pm 0.1j$ ，對於零為 2 和  $0.2 \pm 0.4j$   
兩極。零都在單位圓內。複雜的兩極也在裡面  
單位圓，而實際圓在該圓之外。  
我們介紹了傳遞函數的概念，並且學習瞭如何  
操作框圖。現在是時候計算傳感器的時間響應了。

給定信號輸入的系統。這是下一部分的主題。

#### 4.4 時間響應及其計算

通常，控制系統必須保證某些性能，例如：

- 穩定時間為給定百分比
- 過衝
- 阻尼比
- 等

對於時間定義，我們要求讀者看一下圖 4.21。有一個想法關於穩定時間，過衝等的概念，讓我們考慮線性時間

輸入  $r(t)$  和輸出  $y(t)$  的不變系統。如果我們在輸入，該系統的輸出將如圖 4.21 所示。從這個數字來看，可以看到建立時間定義為系統響應時間，

$y(t)$  達到誤差帶（由一定百分比，2%，5%等定義）並在其餘時間停留。百分比越低，沉降時間越長時間會。

過衝是給定係統時間響應的另一個特徵。

如果我們參考上圖，則超調定義為最大超出系統輸出的穩態值。通常，我們使用百分比過衝，定義為輸出的最大值減去步長值除以步長值。

錯誤也是輸出行為的另一個特徵。它被定義為輸出所獲得的穩定值與期望值之間的差。為一個具有統一反饋的閉環系統，通過數學定義誤差  $E(z)$  如： $E(z) = R(z) - Y(z)$

其中  $R(z)$  是參考輸入， $Y(z)$  是輸出。

以前，我們開發了可用於計算表達式的工具

給定信號的時間。在這裡，我們將使用它來計算給定的時間響應系統選擇的輸入可能是以下信號之一或組合：

- 狄拉克的衝動
- 步
- 坡道

圖 4.21 步進輸入的時間響應行為

為了計算時間響應，讓我們考慮一個具有脈衝傳輸的系統

給定輸入信號  $U(z)$  的函數  $G(z)$ ，並考慮計算

$y(kT)$  的表達式。該系統如圖 4.22 所示。這個數字可能代表

發送了開環脈衝傳遞函數或其等效的閉環脈衝  
簡化系統框圖後得到的傳遞函數。

從這個數字，我們得到：

$$Y(z) = G(z) U(z)$$

時間響應  $y(kT)$  的計算被引入到

可以使用以下方法之一確定  $Z$  逆變換：

- 擴展為部分分數
- 多項式除法
- 殘留法

為了說明時間響應如何，讓我們考慮以下示例。

例 4.4.1 在本例中，我們考慮直流電動機驅動的速度控制  
通過齒輪給定的機械負載。我們假設系統是使用  
微控制器。系統的傳遞函數由下式給出：

其中  $K = 2$  和  $\tau = 2$  該系統被認為是開環的。在這種情況下，由於我們存在  
ZOH，我們獲得：

使用  $Z$ -transform 表，我們得到：

其中  $T$  是採樣週期。對於我們的系統，由於時間常數等於 2 秒，因此對於  
採樣週期為  $T = 0.2\text{sec}$ 。使用這個，我們得到：

如果現在我們認為信號輸入是單位步長，我們得到

要計算時間響應，我們可以使用表格或繼續進行擴展為部分分數。

使用  $Z$ -transform 表，我們可以：

通過擴展為部分分數，我們得到：

由此我們得到：

現在使用  $Z$ -transform 表，我們得到：

因為  $e^{-0.1} = 0.9048$ 。

例 4.4.2 在本例中，我們考慮直流電動機驅動器的位置控制。  
通過齒輪給定的機械負載。我們假設系統是受控的



使用微控制器。系統的傳遞函數由下式給出：

其中  $K = 2$  和  $\tau = 2$

該系統被認為是開環的。在這種情況下，由於我們存在 ZOH，我們獲得：

使用  $T = 0.2$  秒的 Z 轉換錶，我們得到：

如果現在我們認為信號輸入是單位步長，我們得到

要計算時間響應，我們可以使用 Z-transform 表或繼續用膨脹成部分餾分的方法或用殘渣的方法。

使用 Z-transform 表，我們得到：

通過將方法擴展為部分分數，我們可以：

用殘基的方法，我們得到：

在極點  $z = 1$  和  $z = 0.9048$  處。

這些殘基計算如下：

- 極點  $z$  上的殘差 = 1

- 極點  $z$  上的殘差 = 0.9048

現在使用該表，我們得到：

因為  $e^{-0.1} = 0.9048$ 。

從上一節中計算出的時間響應可以看出對於給定的系統，對於給定的信號，輸出可以取有限值或無限值信號輸入。問題是為什麼會這樣。給出了這個問題的答案通過穩定性分析，這將在下一部分中介紹。

## 4.5 穩定性和穩態誤差

對於連續時間域中的系統，穩定性意味著所有極點必須具有負實部。在變換  $z = e^{Ts}$  的情況下， $T$  是採樣期間，我們看到  $s$  域的左半平面對應於內部單元

4.5。穩定性和穩態誤差 109 圓，因此，如果所有極點都在這個單位圓內為了

分析離散時間系統的穩定性，讓我們考慮離散時間系統的穩定性。圖 4.23。該系統的閉環傳遞函數由下式給出：

其中  $R(z)$  和  $Y(z)$  分別是輸入和輸出。系統的極點是以下特徵方程的解：

$$1 + C(z)G(z) = 0$$

穩定性研究需要計算這些根。對於小訂單系統中，我們總是可以手動求解特徵方程，然後獲得極點和穩定性結論將基於極點所在的事實得出位於。對於高階，不建議使用此方法，而替代方法是需要。已經開發了一些標準來研究穩定性。在這些當中我們引用的標準是陪審團之一和 Raible。

例 4.5.1 讓我們考慮一個具有以下特徵方程的動力系統：

特徵方程的根為： $z = 1/2$  和  $z = 1/4$ 。這些根是位於單位圓內，因此系統穩定。例 4.5.2 讓我們考慮一個具有以下特徵的動力系統：

tic 方程：

系統的根為  $z_{1,2} = 1 \pm j(1/2)$  並且都在單位圓之外表示系統不穩定。研究離散時間系統穩定性的直接方法是將其轉換到等效的連續時間，然後使用 Routh-Hurwitz 的準則。

這個想法是找到一個合適的應用程序來映射單位圓的內部到左側的半平面上。然後，我們可以應用 Routh-Hurwitz 準則。我們正在尋找的轉變是：

在特徵方程中用該表達式代替  $z$  將得到一個新的，在  $w$  中，我們可以應用 Routh-Hurwitz 的準則。

例 4.5.3 為了展示我們如何使用勞斯·赫維茲準則，讓我們考慮具有以下特徵方程的動力學系統：

可以看出極點為  $2$  和  $0.2 \pm 0.4j$ 。因此系統是不穩定。

可以採用以下形式：

Routh-Hurwitz 的條件包括填寫下表：

在第一列的基礎上，我們可以看到符號因此系統不穩定。這證實了我們已經評論過的結果較早。

同樣重要的是要注意  $w$  中特徵方程的根是由：

$$w_1 = 0.3333$$

$$w_{2,3} = -0.5000 \pm 0.5000j$$

這些根也可以使用  $w = z^{-1}/z+1$  從  $z$  域中的根獲得

應用雙線性變換可得出：

依次給出：

應用 Routh-Hurwitz 可以得出：

為了保證穩定性，我們應該確定參數  $K$  的範圍這樣我們就不會在第一欄中更改標誌。對於  $w^0$  行，我們應該擁有 0, i.e.  $K > 1/6.32 = 0.158$ 。對於  $w^2$  行，我們還應該  $2.736 - 6.32K > 0$ ，即  $K < 2.736/6.32 = 0.4329$ 。如果我們看這兩個條件，我們結論是系統對於  $0.158 < K < 0.4349$  是穩定的。

為了檢查這一點，讓我們考慮在間隔內的  $K = 0.2$ 。使用這個值，我們得到以下特徵方程式：

其根為  $z_1 = 0.052 + j0.6044$  和  $z_2 = 0.052 - j0.6044$ 。根是位於單位圓的內部，因此系統穩定。對於  $K = 1$ ，我們獲得：

根是  $z_1 = -0.076$  和  $z_2 = -4.876$ 。然後系統不穩定，因為  $|z_2| > 1$ 。對於離散時間，陪審團已經制定了一個標準，該標準給出了關於任何不求解特徵方程的系統。展示這種方法工作時，讓我們考慮以下具有實係數的特徵多項式：

Jury 的穩定性標準包括建立以下一系列係數：

Jury 的陣列係數計算如下：

$P(z)$  描述的系統穩定的充要條件是：

例子 4.5.5 檢查以下描述的系統的穩定性多項式：

我們形成 Jury 的係數數組：

由於  $n = 3$ ，因此應滿足以下條件：

- $P(1)$  必須為正： $1 + 3.3 + 3 + 0.8 = 8.1 > 0$  為真
- $P(-1)$  必須為負，因為  $n = 3 = \text{奇數}$ ： $-1 + 3.3 - 3 + 0.8 = 0.1 > 0$  這是錯誤的

一個錯誤的條件足以得出系統不穩定的結論。

例 4.5.6 讓我們考慮一個具有以下特徵方程的動力系統：

其中  $K$  是確定參數以使系統穩定的參數。此特徵方程式可重寫如下：

應用 Jury 標準可得出：

因此，如果  $K \in ]0, 2.8[$ ，我們的系統將是穩定的。例如，如果我們將  $K$  固定為 2，

給出以下特徵方程式：

根是  $z_{1,2} = -0.3000 \pm 0.5568j$ ，由於  $|z_{1,2}|$  而在單位圓內  $< 1$ 。

Raible 提出了另一個研究穩定性基礎的標準。這個穩定性標準還包括陪審團標準，即先填充數組，然後再填充關於穩定性的結論。為了展示此標準的工作原理，讓我們考慮以下因素特徵方程：

其中  $a_i$  是真正的標量。

重複這些過程，直到數組獲得  $2n + 1$  行。最後一行僅包含一個數字。Raible 的穩定性標準當  $a_0 > 0$  時，且僅當且僅當多項式的根都在單位圓內  $a(i) > 0, i = 0, 1, \dots, n - 1$

係數  $a(i) > 0, i = 0, 1, \dots, n - 1$  出現在 Raible 的數組中。

備註 4.5.1 假設  $a_0 > 0$  不是限制性的。實際上，當  $a_0 < 0$  時，足以改變多項式  $P(z)$  的所有係數的正負號以獲得  $-P(z)$ ，依次用於 Raible 的標準。

該過程是正確的，因為  $P(z)$  和  $-P(z)$  的根相同。

例 4.5.7 為了說明 Raible 準則是如何工作的，讓我們考慮以下特徵方程式：

係數  $a_0$  必須為正，然後形成多項式

我們提出了一些技術來研究離散時間系統的穩定性 tests。同樣重要的是要注意，我們也可以在頻域。

## 4.6 根軌跡技術

根軌跡技術是一種強大的方法，通常用於連續時間或離散時間系統進行分析或設計。該技術給出了關於增益或更大時閉環動力學的極點行為的想法（ $a$  參數或更多）。直接結論是我們知道，會如何影響系統的穩定性和其他性能參數更改。

如今，有許多工具可以繪製任何動力學系統的根本基因座其中一些可免費使用。在本節的其餘部分，我們將使用 **Matlab** 進行繪製，但我們將製定規則，以獲取如何獲取根軌跡的草圖如果我們手邊沒有電腦。對於連續情況，離散系統的根本軌跡描述為我們以以下形式編寫的特徵方程式：

$$1 + KG(z) = 0$$

其中  $K$  是變化的參數，

其中  $z_1, z_2, \dots, z_n$  是極點， $n_1, n_2, \dots, n_m$  是開環的零點轉換功能。

當參數  $K$  從 0 變為無窮大（ $\infty$ ）時。與我們使用的規則相同用於在  $s$  平面上繪製連續時間系統的根本軌跡到  $z$  平面上離散時間系統之一的繪圖，除了結果的解釋主要在穩定性方面有所不同。

從特徵方程式，我們得到以下條件：

第一個條件稱為幅度條件，而第二個條件稱為幅度條件稱為角度條件。 $Z$  平面上滿足這兩個條件的任何點條件屬於系統的根源。至此對應一個增益  $z_0$  如果這點是  $z_0$ ，那麼我們有：其中  $\theta_0$  是該點的對應角度。如果滿足上述兩個條件，則  $z$  平面的點將屬於根軌跡。位置。通常，除非給定係統的確切根源圖，否則這是一項艱鉅的任務，除非我們有相應的工具。通常，此根軌跡的草圖可以是

使用一些簡單的規則即可輕鬆獲得。其中一些規則是：

1. 分支的數量等於系統的階數，即： $n$ ;
2. 根軌跡相對於實軸對稱。這是由於事實特徵方程的根是實數還是複數。而如果有一個複雜的根，我們自動得到它的共軛。
3. 軌跡起源於開環傳遞函數的兩極，並且終端-在 **this** 傳遞函數的零上展開。解釋為什麼基因座起源從極點，我們可以使  $K$  等於零，而為什麼基因座終止於零可以通過讓  $K$  在等式中變為  $\infty$  來解釋。（4.4）。
4. 漸近線的數目等於兩者之間的差極點  $n$  和開環傳遞函數的零個數  $m$ 。這些漸近線的特徵是：  
參數  $\delta$  給出漸近線與實軸的交點， $\beta_k$  給出使每個漸近線與實軸成的角度。
5. 對於根軌跡的斷點，首先我們確定變化的參數  $K$ ，即：

斷點是以下方程式的解：

從這個方程的根中選擇那些可行的解決方案很重要對於斷點。

6.虛軸在  $z$  平面上的交點可以通過以下方式確定在特性方程式中用  $jv$  替換  $z$  並編寫如下：

依次給出兩個方程式：

解給出相交發生的頻率和相應的收益。

7.從復雜極點出發的角度或到復雜點的到達角度使用角度條件計算零。 如果我們要

計算角度為  $z0$ ，條件角度變為：

例 4.6.1 為了說明根軌跡技術是如何工作的，讓我們考慮圖 4.24 的系統，其中工廠是雙積分器和控制器是一個增益為  $K$  的比例動作，我們假設它在零之間變化和無窮大是由於某些物理原因，例如加熱，老化等。使用  $Z$  變換錶和閉環傳遞的表達式函數，我們得到該系統的以下特徵方程：

根軌跡如圖 4.25 所示。 所有根都在單位圓之外藍色。 系統不穩定。 這意味著比例控制器不能穩定雙積分器。

例 4.6.2 作為根軌跡技術的第二個例子，讓我們考慮圖 4.26 的系統。

根軌跡如圖 4.27 所示。 所有的根都在單位圓內藍色。 因此，系統對於所有增益  $k < 8.65$  都是穩定的。

## 4.7 波特圖技術

頻率響應在天線的分析和設計中起著重要作用。連續時間和離散時間系統。 時間響應，頻率

響應包括通過正弦輸入激勵系統。 在連續-時間系統，事實證明，對於正弦輸入，穩定輸出線性系統是正弦曲線，輸入頻率相同，其幅值和輸出的相位是該頻率的函數。 對於離散時間系統，輸出也是正弦波，頻率與輸入信號和相位相同並且幅度仍然是該頻率的函數。 為了說明這一點，讓我們考慮具有以下傳遞函數的穩定線性系統：

令輸入  $r(t)$  具有以下表達式：

$$r(t) = \sin(wt)$$

其中  $w$  是輸入頻率。此處的幅度等於 1。該信號的 Z 變換由下式給出（請參見 Z 變換錶）：

現在，如果我們認為系統被相應輸出的  $R(z)$  激勵， $Y(z)$  由下式給出：

$$\begin{aligned} Y(z) &= G(z)R(z) \\ &= K \frac{\prod_{i=1}^m (z - n_i)}{\prod_{i=1}^n (z - z_i)} \frac{z \sin(wT)}{(z - e^{-jwT})(z - e^{jwT})} \end{aligned}$$

## 4.8 結論

本章介紹基於傳遞函數概念的分析工具。主要是，我們開發瞭如何計算時間響應和確定係統性能。我們還介紹了根軌跡和波特圖技術。

## 第五章 基於傳遞函數的設計

### 5.1 引言

解決控制設計問題始終是一個挑戰，即使對於更有經驗的人也是如此控制工程師。必須為其設計控制器的系統可以是現有的，但表現不佳，我們想改進，或者我們正在構建的新系統。在這兩種情況下，設計程序均在通過定義所需的性能來獲得系統的數學模型這將使我們能夠確定控制器的結構及其參數。通常，控制系統旨在保證某些性能考慮到系統的閉環動力學。這樣的表演可以歸納為瞬態和穩態的穩定性和行為國家政權。通過尊重給定係統的局限性，它始終是我們試圖通過尋求折衷來改善瞬態狀態的情況在系統可能出現的超調和其速度之間。為了穩定狀態，我們進行搜索以確保誤差小於某個選定的公差。我們將在本章中考慮以響應設計要求的控制器-是經典的，例如比例，積分和微分作用，他們的近似值。本章的其餘部分安排如下。在第 2 節中，控制設計問題已製定。第三部分介紹了設計經典的經驗方法控制器。在第 4 節中，使用根軌跡設計經典控制器方法被開發出來。第 5 節介紹了 Bode 方法。第 6 節介紹了一個案例研究包括為直流電動機套件設計不同的控制器。

### 5.2 控制設計問題的表述

在本章中，我們將考慮性能較差的現有系統，我們想改善。我們的願望是對瞬態同時採取行動通過在閉環中引入控制器來強制整個系統按預期運行。可以在時域或頻域中給出性能。同時領域，穩定性是設計過程中的首要要求。在旁邊

穩定性，我們希望瞬態和穩態機制能夠在理想的方式。在瞬態狀態的時域中，我們應控制過衝，所選時間的上升時間和穩定時間將取決於我們希望為我們的系統保證精度。對於穩態制度，我們希望確保系統的錯誤小於特定的指定值。

在頻域中，除了性能根據閉環動力學的穩定性，增益相位和通常來說，很難在兩者之間建立聯繫時域和頻域的性能。更具體地說，所研究的系統由傳遞函數描述，例如可以使用識別方法獲得。讓我們用  $G(s)$  表示這個傳遞函數。必須在控制的第一階段確定此模型設計。然後，根據控制工程師的性能和專業知識進行設計我們可以選擇可以正確響應設計的控制結構目標。然後，使用適當的方法，我們可以確定控制器的增益。

因此，控制設計問題包括確定：

- 控制器的結構
- 及其參數

使用所需的性能和一些啟發式方法來強制關閉-所選控制器的動態迴路表現出理想的表現。這種方法由於被忽略的不同現象，在實踐中可能需要改進動力學。

我們將在本章中考慮的控制器是經典的控制器，

在文獻中被認為是比例（P），積分（I）和微分（D）動作及其近似也稱為相位滯後超前和相位超前滯後。控制器的傳遞函數將表示為  $C(z)$ 。確定控制器後，相應的差分方程為使用適當的微控制器實時獲取和實現。對於有關此主題的更多詳細信息，我們請讀者參考實現部分詳細。

由於系統性能通常是連續不斷地給出的，因為這樣做更自然。設計步驟可以在連續時間或離散時間。一般來說，設計方法使用以下步驟：

- 表演轉換成桿子
- 選擇所需控制器的結構
- 使用所需的極點確定控制器參數
- 對控制器的參數進行了一些調整以補償

預期行為與實際行為之間可能存在差異設計過程中未考慮的系統零。重要的是要注意，控制器參數的確定可以在連續時間或離散時間完成。在連續時間內在這種情況下，確定控制器參數，然後控制器進行將  $fer$  函數轉換為離散時域，得到差分方程我們應該實時實施。對於離散時間，差分方程為直接獲得併實施。



設計方法可以是以下方法之一：

- 基於經驗方法的設計
- 基於根軌跡法的設計
- 基於波特圖法的設計

在本章的其餘部分，我們將介紹這些方法並給出一些示例展示這些技術如何應用於實際系統。仿真結果將是用來表明其有效性。控制器的設計是連續進行的然後獲得相應的控制器離散時間版本。本章使用在 Boukas 中開發的方法。

### 5.3 基於經驗方法的設計

實證方法基於齊格勒-尼科爾斯的工作。這些方法與其他方法相比，具有優勢，因為它們允許設計除即使在沒有系統數學模型的情況下，也可以找到所需的控制器。Ziegler-Nichols 方法主要基於動力學系統的響應。Ziegler-Nichols 提出了使用時間響應的方法，其他使用時間響應的方法。頻率響應。在本節的其餘部分，我們將介紹這些方法。首先讓我們集中討論時間響應方法。在這些方法中我們可以處理開環穩定和不穩定的系統。第一種方法考慮在原點無極點也不佔優勢的穩定係統的情況複雜的兩極。在這種情況下，階躍響應由圖 5.1 中的給出。從中直接確定參數  $T$ ,  $\tau$  和  $k$  以及  $T_{ab}$ 。其中  $k$  是系統的增益， $\tau$  是上升時間， $T$  是延遲時間。

### 5.4 基於根軌跡的設計

根源技術是控制系統分析和設計的強大工具。在本節中，我們將使用它來設計控制器，以確保理想的表演。該系統的模型應該以

轉換功能。根軌跡技術可用於設計經典控制器。的該方法背後的技術包括選擇使座位穿過表演中給定的極點。

其中  $k$ ， $-z_i$  和  $-p_i$  分別是系統的增益，零和極點。首先讓我們集中討論比例控制器的設計。

其中  $K_P$  是要確定的控制器的增益。基本控制過程中可以知道，比例控制器的作用是同時考慮了瞬態和穩態機制，但其能力有限。它可以減少錯誤，但絕不會使其等於零。計算控制器的增益，我們將使用以下過程

1.計算閉環動力學的特徵方程，即： $1 + K_P G(s)$  令  $K = k K_P$

2.畫出從 0 到無窮大的  $K$  的根軌跡

3.確定基因座和對應於所需的阻尼比  $\xi$  ( $\cos\theta=\xi$ ) 並得到主導的極點對。讓 SD 成為具有虛構積極部分的人。

4.計算得出極點 sd 的增益 K

# MechaFutureAndChallenges

## 第 1 章 機電期貨

### 1.1 挑戰

自 Tetsuro.Mori 引入機電一體化系統概念以來的 40 多年,以表示電子元件的可用性對固有的機械系統的控制和運行的影響越來越大,現在和現在仍然是一個重大和快速的技術變革時期。特別是在 ere 系統內,從硬體到固件和軟體,重點的轉變,導致推出了各種消費類產品,這些消費品的結構圍繞智慧設備的使用而構建

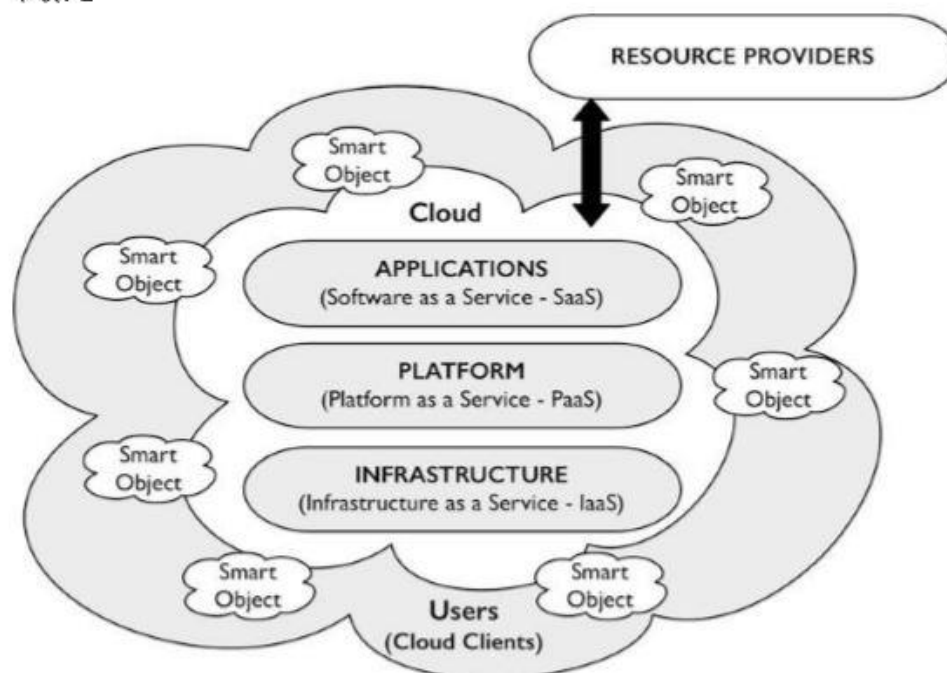


圖 1.1 物聯網基於雲的結構

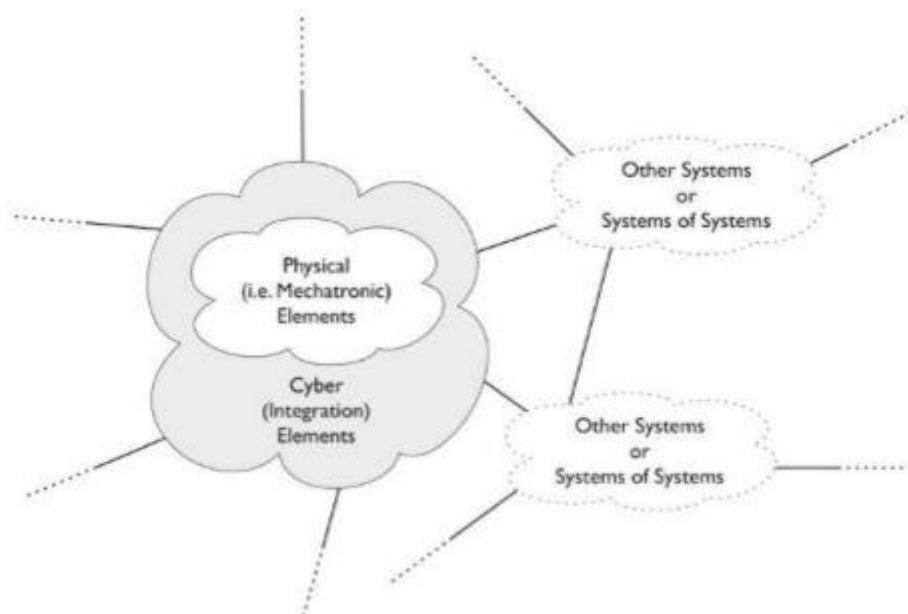


圖 12 網路物理系統

2014 年,與在瑞典卡爾斯塔德舉行的機電一體化論壇會議合作,來自世界各地的一些從業人員被要求用一句話來表達他們對機電一體化在未來幾年面臨的最重大挑戰的看法。收到的回復以圖的形式顯示。

## 12 挑戰

關鍵問題概括為:

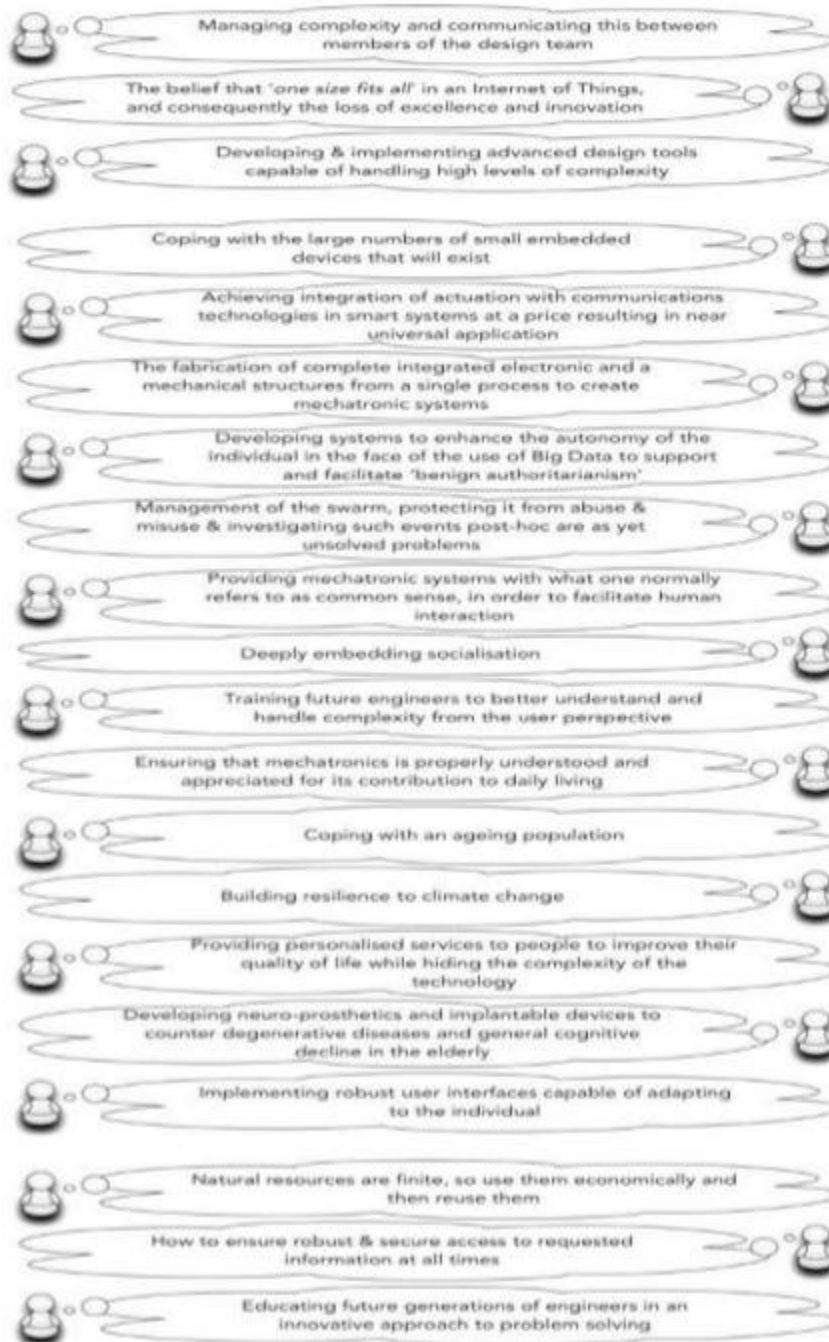
- 設計
- 隱私和安全
- 複雜性與倫理
- 老齡化人口
- 使用者
- 可持續性
- 教育

### 12.1 設計

傳統的設計符號方法通常遵循一條路徑,通過結構化系統定義實現集成,然後進行系統開發過程,並得到適當的測試制度的支援,以確保系統的整體功能。

該方法通過設計理論與設計實踐的協同相互作用,經過多年的發展。然而,設計理論必然落後於實踐,即新設計理論提供的可能性

圖・15 從業者對機電一體化面臨的挑戰的反應



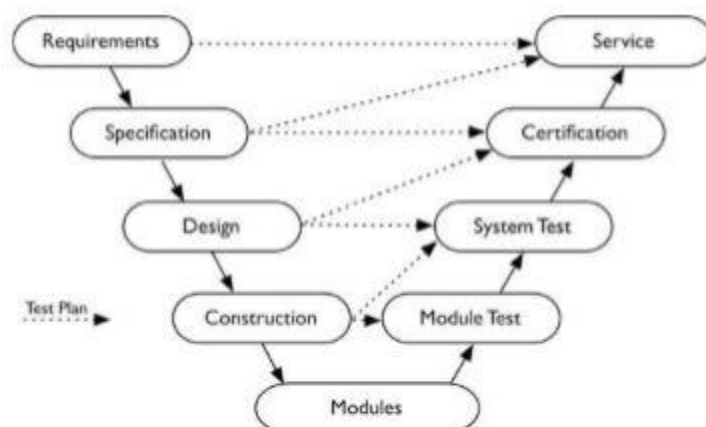


圖 16 簡化的 V 模型

## 12.2 隱私和安全

許多與 IoT 關聯的設備都能夠收集大量個人數據,其中很大一些可能以使用者未知的區域和方法進行。然後,這些資料有可能進行分析,相關誤解對隱私的影響風險然而,這必須與提取有益知識的潛力相平衡。因此,系統設計人員越來越重將隱私置於其設計過程的核心,以每個物聯網、網路物理系統和大數據的上下文中,並且必須反映在設計過程本身以及支援這一點的方法和工具中。

表 12 對系統安全的明顯威脅  
(在 [19]之後)

| 威脅        | 概率 (%) |
|-----------|--------|
| 資料洩露      | 17     |
| 員工錯誤      | 16     |
| 員工擁有的裝置事件 | 13     |
| 雲計算       | 11     |
| 網路攻擊      | 7      |
| 不滿的員工     | 5      |
| 外部顧客      | 5      |
| 以上所有內容    | 19     |
| 以上都不是     | 8      |

## 12.3 複雜性和道德

隨著系統變得越來越複雜,並開始以更大的自主性運行,人們開始質疑所有利益相關者在從醫療保健到自動幕斯車輛等一系列應用和環境中瞭解其性質和

功能的能力問題。例如,個人或個人的福祉或生命責任被委託給該制度。其他問題包括:

- 技術的雙重用途 — 無人機等技術可以與有益的應用相關聯,例如作物管理,但也用於軍事和其他目的。
- 技術對環境的影響 — 將技術引入環境可能會以各種方式破壞和改變環境,即使基本意圖是良性的。
- 技術對全球財富分配的影響——技術的使用可以增加不同社會群體之間的隔閡,即使在同一國家內。
- 分歧與相關的社會技術差距——訪問和使用通過雲提供的服務的能力日益分離。

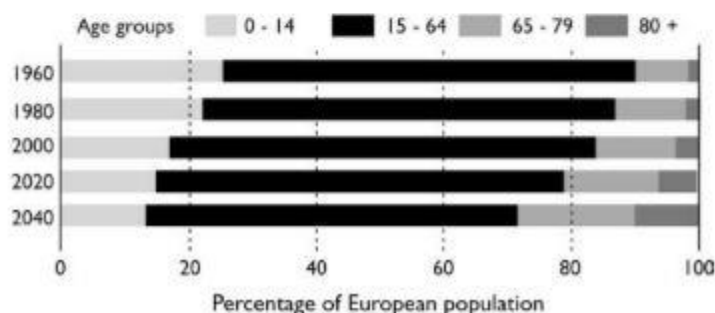


圖 17 An 歐洲人口老齡化(在 +32 後)

- 確保公平訪問技術日誌 — 控制對技術的訪問可以作為對開發的限制。
- 技術成癮——個人沉迷於他們使用的技術。
- 技術鎖定 — 個人可能會鎖定在特定技術中,一個簡單的例子就是在蘋果和 Android 之間做出選擇。
- 人類的非人化和人化——通過剝奪對其活動和福祉的責任。

## 124 老齡化人口

面對人口老齡化,圖 17 展示了歐洲境內年齡組分佈的過去和預測變化,人們提出了如何最好地利用技術支援老年人的問題,並試圖為他們提供老年中日益提高的獨立性。在部分, there 需要確保在物理和資訊領域內提供適當的流動性水準,以防止個人保持獨立性和與社會接觸。



## 1.2.5 使用者

互聯網 capable 設備的提供已經通過社交媒體對社會行為產生了重大影響,但也使 more\_ready 獲取信息的機會比歷史上更加方便。此類設備還支援提高與周圍環境的互動水準,例如智慧家居。此外,還推出了可穿戴設備設備為電子健康、mHealth 等領域的發展提供了機會,以支援個人福祉[35],進而引發 privacy 和個人數據控制問題。

但是,還需要開發新的用戶介面形式,以支援更廣泛的使用者與此類系統交互的能力。特別是,人們越來越要求能夠以不需要複雜形式的通信或有關基礎技術的知識的方式對用戶意圖和上下文進行驗證。

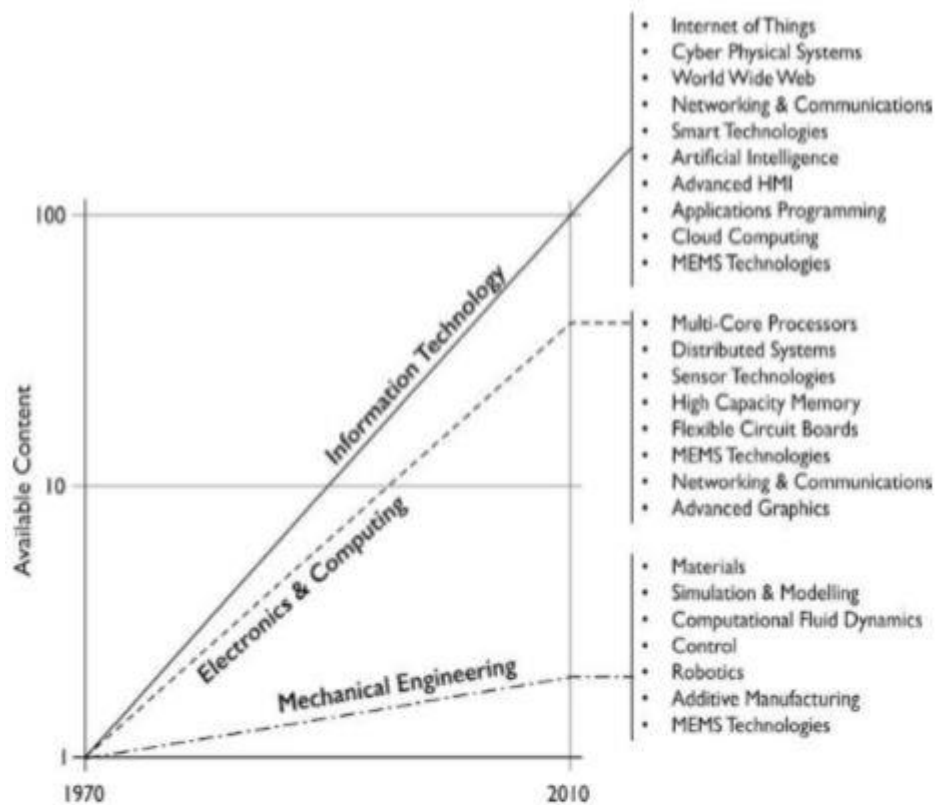
## 1.2.6 可持續性

由於認知到需要以朝著可持續的社會為中心,並圍繞有效管理和利用所有可用資源,建議如圖。

|  |   |   |  |
|--|---|---|--|
| <b>Mobility</b> <ul style="list-style-type: none"><li>+ Personal</li><li>+ Goods</li><li>+ Transport</li></ul>                           | <b>Information</b> <ul style="list-style-type: none"><li>+ Access</li><li>+ Content</li><li>+ Context</li></ul>       | <b>Governance</b> <ul style="list-style-type: none"><li>+ Regulation</li><li>+ Legislation</li><li>+ Taxation</li></ul>       | <b>Energy</b> <ul style="list-style-type: none"><li>+ Sources</li><li>+ Technology</li><li>+ Management</li></ul>              |
| <b>Quality of Life</b> <ul style="list-style-type: none"><li>+ Social impact</li><li>+ Ageing society</li><li>+ Communications</li></ul> | <b>Security</b> <ul style="list-style-type: none"><li>+ Social impact</li><li>+ Reliability</li><li>+ Costs</li></ul> | <b>Benefits</b> <ul style="list-style-type: none"><li>+ Outcomes</li><li>+ Costs</li><li>+ Triple Bottom Line</li></ul>       | <b>Agriculture</b> <ul style="list-style-type: none"><li>+ Food security</li><li>+ Transport</li><li>+ Contamination</li></ul> |
| <b>Health</b> <ul style="list-style-type: none"><li>+ Resources</li><li>+ eHealth/mHealth</li><li>+ Prostheses</li></ul>                 | <b>Interface</b> <ul style="list-style-type: none"><li>+ Format</li><li>+ Personalisation</li><li>+ Users</li></ul>   | <b>Manufacturing</b> <ul style="list-style-type: none"><li>+ Low impact</li><li>+ Minimum waste</li><li>+ Resources</li></ul> | <b>Resources</b> <ul style="list-style-type: none"><li>+ People</li><li>+ Education</li><li>+ Earth</li></ul>                  |

## 1.2.7 教育

面臨平衡技術含量與理解機械工程、電子和資訊技術核心學科整合要求的挑戰



在技術發展相關的課程設計方面的挑戰外，還要考慮其他因素。包括：

### 交貨變更

- 大規模線上開放課程(MOOCs)
- 基於教程和研討會的學習支援。

### 混合學習

- 社交媒體對學習的影響

### 結構問題

- 分散式學習資源。
- 內容的時間值。



## 第二章

### 如何開始

於 20 世紀 70 年代，當時機械系統需要更加精確，迫使行業和學術界探索感測器和電子輔助，這種運動使機械工程師和電子工程師合作和相互理解。特定的模擬工具、共同開發機械裝置，在可製造性、成本和動態方面非常嚴格。在 20 世紀 80 年代的同一時期，在許多行業和學術界，機械工程師開始越來越多的處理動力學和控制問題，機械工程部門也開始出現控制小組，這些都代表著他們擺脫了單一的學科方法。



圖 21 An 飛利浦電子 NV 具有平衡旋轉臂的光學存儲設備

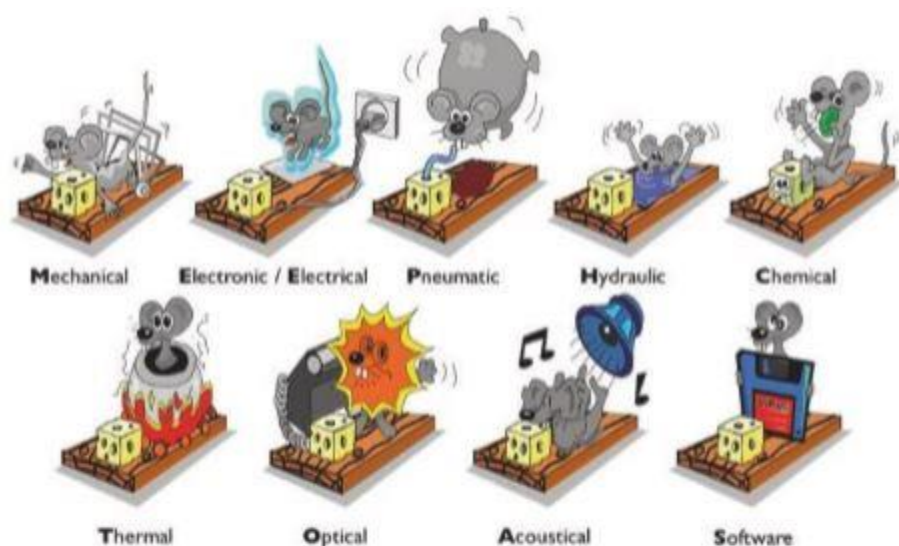


圖 22 Many 給定問題的許多單一學科解決方案 [1]

## 2.2 計算機控制裝置

個人電腦的迅速發展,使模擬和設計工具得到更好的利用,從而在早期階段提高了整體設計過程和設計思想交流的品質。然而,與電子相關,PC 啟用數位化電腦控制的機電一體化系統測試和實現。這需要解決計算機科學工程的作用,並表明需要包括軟體學科,但到一個相當定的程度。這也導致越來越多的系統工程領域,作為一種在工業中處理更複雜的產品和高科技系統的方式。然而,思考"通用"語言,或者至少更好地理解彼此,顯然比硬體域本身更不瑣碎。

## 2.3 應用

機電一體化思維對性能的提高是深刻的,並且得到了廣泛的認可。機電一體化的應用可以在人 y 產品和生產環境中找到。雖然在早期,電機控制是一個經常看到的應用,機電一體化思維也用於液壓系統的設計,壓電驅動執行器,預控設備的建模和控制,科學設備,光機電一體化,汽車機電一體化等。

## 2.4 多物理場

高端機電一體化系統,如圖 23 的晶圓掃描器,用於光學光刻或電子掃描探頭,以及空間複式科學儀器,其誤差預算越來越接近 being 於在各種源上的扁平分佈。

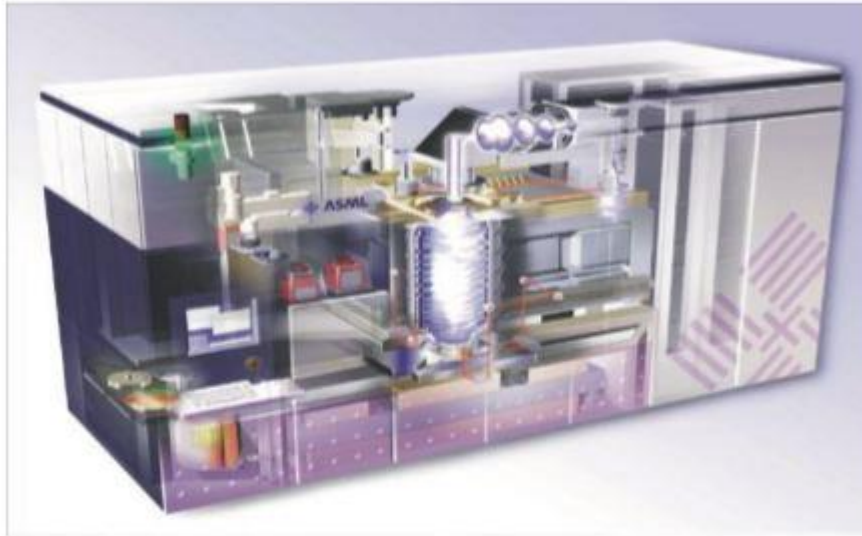
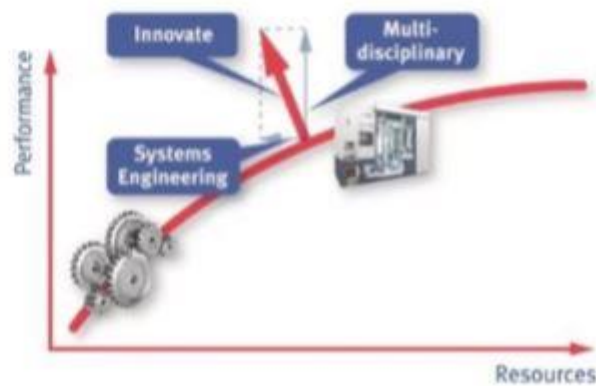


圖 23 晶圓掃描器

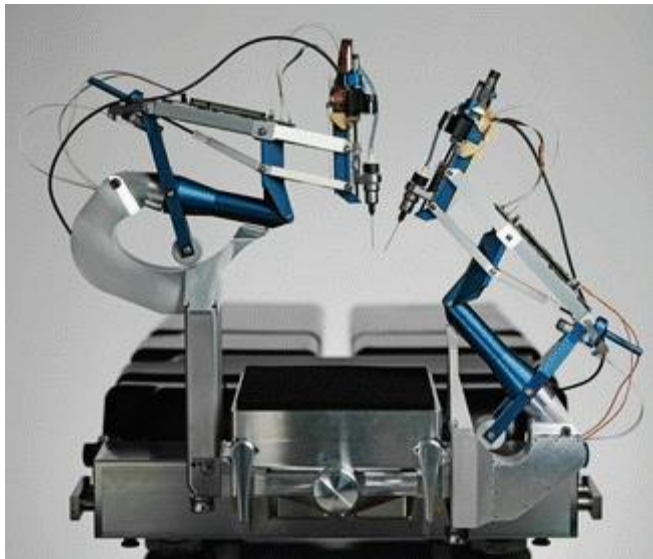
圖 24 The 性能複雜性(資源) 權衡 [4]



## 2.5 機器人

與上述高端系統幾乎相反，機器人技術領域也影響機電一體化領域。在這裡，需要計算機科學領域來應對非結構化和不斷變化的環境。在機器人技術方面，開發主要針對視覺、地圖和本地化，因此不僅要了解環境（“世界建模”），還要了解人工智能（AI）領域，這已經是數十年的承諾，但在未來可能會迅速發展。未來幾年。由於即將推出的自動駕駛汽車，這兩個領域目前都處於加速階段。在動力總成（即電力傳動和變速箱）領域以及計算機科學的應用（例如現代汽車中的傳感器）領域，汽車行業的破壞是巨大的，包括實現的自主功能的迅速發展在乘用車和商用車中。

實際上，這一切都與機電一體化、人工智能、控制有關，機器人技術領域（包括自動駕駛汽車）可以被視為機電一體化領域之外的一個單獨的研究領域，但是例如，工業機器人的速度要求或外科機器人（例如下圖的 **Preceyes** 機器人）的精度要求必須包括在內。機器人動態行為的描述。從剛體建模到柔性系統的轉變，直接使其成為機電一體化的核心。系統工程思想和系統拓撲優化也是如此，在車輛的混合動力系統中也是如此。那麼機電一體化在哪裡結束而機器人技術在哪裡開始呢？



### **The Preceyes eye surgery robot**

## **2.6 網絡物理系統，智能產業和物聯網**

從分散式機電一體化系統向聯網系統的轉變被稱為網絡物理系統領域，指的是控制論領域。研究的問題是如何在數據包（信息）丟失期間或之後保證穩定性和性能，以及如何處理可變延遲。該領域甚至離機電一體化的硬件還很遠，但是發展如此之快，以至於我們應該問一個問題，即如何在遠程狀態監視和服務領域中將網絡控制系統的潛力納入其中。在接下來的十年中，物聯網（IoT）的爆炸式增長進一步要求找到該問題的答案。

機電一體化將滿足物聯網的一個應用是我們製造業的未來。工業 4.0 或智能工業的關注點是關於聯網的現代工業自動化。

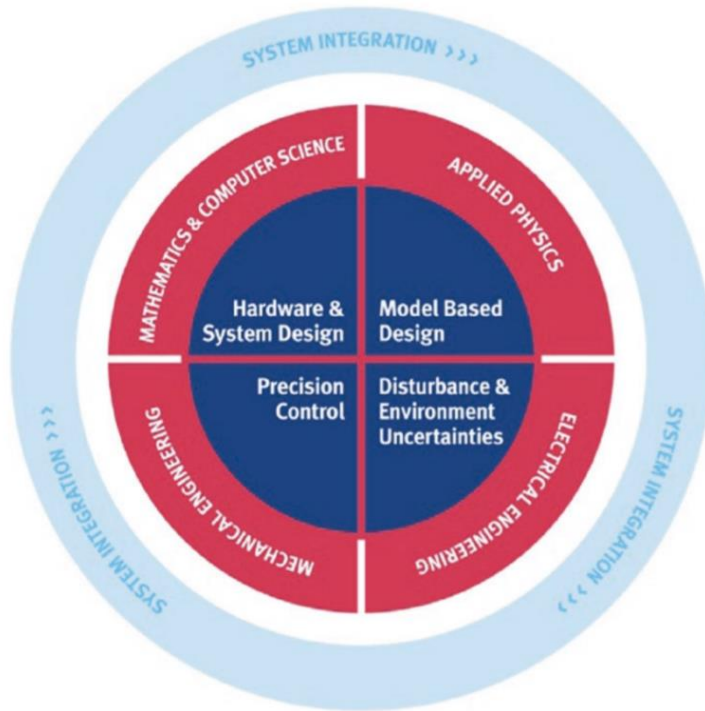
- 如果共享物流知識，優化一個工作站的性能作為整個物流或運營的一部分，如果生產設施中的服務和維修可靠，則對於通過製造工廠的貨物流意味著什麼？工作站是否靈活並且可以適應？
- 這對工業機器人和智能機電一體化生產設備意味著什麼？
- 這將如何影響我們的機電設備和產品的設計要求？

物聯網不僅會改變現代化的工廠。 據估計，到 2020 年，將有 500 億台設備連接到互聯網。這意味著它將進入我們的家庭、家用設備以及汽車。當可穿戴電子設備被進一步推動時，我們被傳感器所包圍，我們只需要朝致動一步就可以閉合迴路並進入機電一體化的世界。

## 2.7 走向系統集成

監督這些發展，我們可能會質疑機電一體化到底是什麼或將是什麼。機電一體化是否受到干擾？它已經蒸發到系統工程中了嗎，它是否是支持學科的一部分，它是否擴大成為網絡物理的中堅力量？而且，如果生物系統也將具有如果實施了技術設備（人類互聯網），那麼機電一體化學科的作用是什麼？我們應該如何在機電一體化思維方面對人們進行教育？在圖 2.6 中，使用系統工程的作用來實現學科和技術貢獻的必要整合。

在本書中，許多提到的發展將得到解決。我們不會對機電一體化的未來及其教育有確切的答案，但是我們也知道，由於我們無法預測未來，因此它應該是強大且適應性強的！我們肯定知道技術發展的步伐正在加快，因此，我們也應該如此！



**Fig. 2.6** 系統工程學科與技術的整合

#### 參考文獻

1. Munnig Schmidt R, Schitter G, van Eijk J (2011) The design of high performance mechatronics. Delft University Press, The Netherlands

ics. Delft University Press, The Netherlands

2. Silvas E, Hofman T, Serebrenik A, Steinbuch M (2015) Functional and cost-based automatic

generator for hybrid vehicles topologies. IEEE/ASME Trans Mechatron 20(4):1561–1572

3. Butler H (2011) Position control in lithographic equipment: an enabler for current-day chip

manufacturing. IEEE Control Syst Mag 31(5):28–47



4. Steinbuch M, Pahnke K, High Tech Systems Centre. Eindhoven University of Technology.

<https://www.tue.nl/en/research/research-areas/high-tech-systems/>

5. Meenink HCM (2011) Vitreo-retinal eye surgery robot: sustainable precision. PhD thesis,

Eindhoven University of Technology

6. Bradley D, Russell D, Ferguson I, Isaacs J, MacLeod A, White R (2015) The internet of

things—the future or the end of mechatronics. Mechatronics 27:57–74

## 第 3 章

### 機電一體化的未來挑戰

#### 3.1 引言

在航空航天工業中，掌握機電一體化系統的設計是一項主要要求。確實，程序成本的很大一部分花在了這些系統的設計上，這也代表了產品性能的很大一部分。在本章中，我們將揭露工業界在機電一體化領域未來幾年將面臨的一些主要挑戰。這些挑戰涉及設計機電一體化系統，其驗證/確認及其操作。

#### 3.2 設計中的挑戰-架構和規模

##### 3.2.1 使用模型確定體系結構的大小

嵌入式技術系統的物理體系結構，例如圖 3.1a 的飛行控制系統的機電執行器或圖 3.2b 的供電網絡的電力電子模塊，是來自不同技術的組件的組合，Van der Auweraer 等人的工作和 Hehenberger 等。強調了這樣的多域系統的設計需要不同的建模層，如圖 3.2 所示：

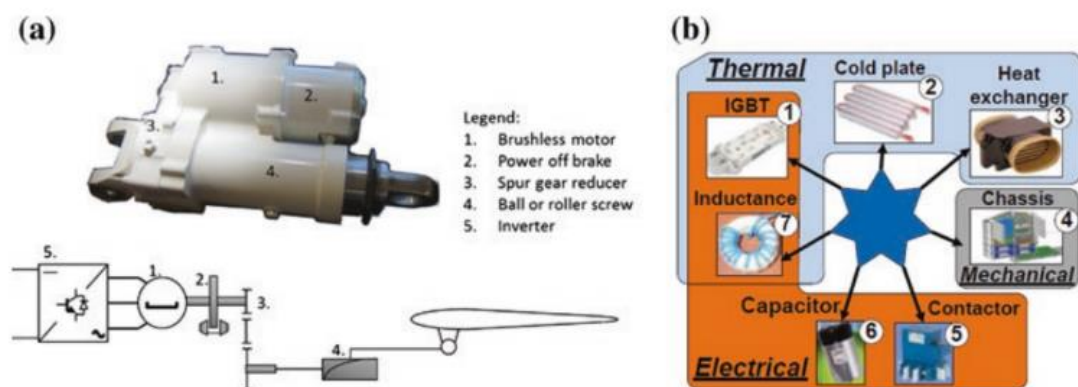


圖 3-1 嵌入式系統的多域架構

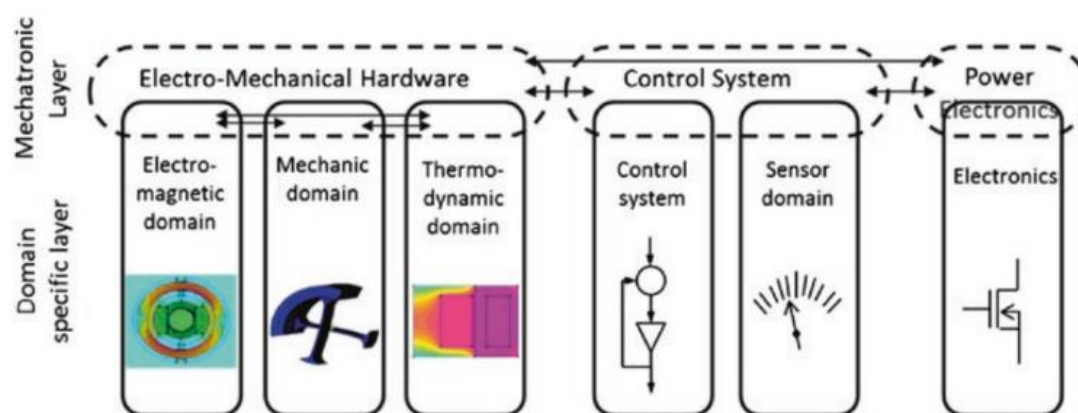


圖 3-2 嵌入式系統的多域架構

- 機電一體化層必須考慮組件之間的功能和物理耦合。通常使用 0D / 1D 模型（也稱為集總參數模型）完成此級別的建模，以代數方程，常微分方程（ODE）或微分代數方程（DAE）表示。
- 基於幾何表示，在上一層中需要一個特定的域層來描述性能極限和參數。特定域現象通常通過偏微分方程（PDE）表示。對於這種水平的建模，可以使用解析模型來簡化幾何，或者對於複雜的 2D 和 3D 幾何，例如通過使用數值逼近法（例如有限元方法（FEM））來實現。

從系統集成商的角度出發，功率元件的設計應允許通過多種相互作用的技術來優化組件的尺寸和規格。這種與組件設計不同的系統級設計，需要在機電一體化層中使用專用模型來表示特定領域層的關鍵信息。Budinger 等人將其稱為



“估計模型”，使設計人員能夠輕鬆地採用多個模型。設計約束。模型圖 3.1 嵌入式系統的多域體系結構圖 3.2 分層設計模型（在 Hehenberger 等人之後）直接且明確地將一些主要特徵（例如組件的總體尺寸）與尺寸確定所需的次要特徵聯繫起來 和優化。 這些估計模型所需的能力如下：

- 提出易於處理並在不同計算工具中實施的表格。
- 使自己易於進行分析操作。
- 可在與最初使用它們的區域稍有不同的區域中重複使用。

為了滿足這些約束，通常使用簡化的分析模型。其中，比例定律已被證明可以有效地代表大範圍變化中的物理現象。但是，這些模型僅在特定條件下有效，其中可以提到幾何形狀，材料相似性和行駛物理現象的唯一性。對於系統設計人員而言，模型應盡可能具有預測性。詳細的有限元模型可以準確預測在這種情況下，物理現象仍然很耗時。儘管最近有工作在減少模型階數上，但在初步設計階段，有限元模型的計算成本仍然很高。因此，為此目的使用元建模技術很有趣。機電設計的挑戰是從集成者的角度開發專門用於選擇系統組件的元建模技術。 Budinger 等人的論文。提出了一種基於縮放定律的元建模方法，該定律從局部數值模擬（FEM）中提取估計模型的簡單全局表達式。

### 3.2.2 機電一體化設計學科系統篇

圖 3.1 中系統的設計受到以下主要方面的驅動，以滿足各種要求：機身與驅動負載之間的集成（質量，幾何包絡），對環境的抵抗力（熱和振動），瞬時功率和節能，動態性能，使用壽命，可靠性，對故障的耐受性或耐受性。表 3.1 總結了這些不同的設計觀點以及基於模型的設計可能的關聯建模級別。

優化此類系統時，這些多種設計觀點產生了真正的挑戰。為了在同一循環中考慮這些標準，來自多學科設計優化（MDO）的工具可能會有用。MDO 是工程領域，它使用統計和優化方法來解決包含 2D / 3D FEM 代表的所有相關學科的設計問題同時進行 0D / 1D 模擬和代數計算。每個特定的計算都被視為一個黑匣子，可以直接調用它，並通過實驗設計（DoE）進行分析。所有計算都可以鏈接在一起，並用於設計探索，靈敏度，優化和穩健性分析； iSight，Optimus 和 ModelCenter 是此類框架的示例。

優化任務需要較少的模型計算時間。替代模型或元模型是另一種模型的簡化或近似描述模型，可以用於表示機電一體化層中的特定領域層模型（FEM）。機電一體化層通常在時域中處理動態系統，並使用瞬態時間仿真，但是在設計優化期間，依靠評估負載曲線的主要諧波來近似時域行為的方法可能很有吸引力。

這些優化和統計支持工具允許以自由選擇的順序鏈接設計過程的步驟。但是，它們在選擇如何構造計算中的連接或將哪些參數作為設計參數，約束或目標考慮在內時沒有提供幫助。

| Requirements          | Corresponding design or sizing viewpoints                     | Algebraic models (0D) | Differential algebraic (1D) equations | CAD (3D) | FEM (3D) |
|-----------------------|---|-----------------------|---------------------------------------|----------|----------|
| Integration           | Mass  | ⊗                     |                                       | ⊗        |          |
|                       | Geometrical envelope  | ⊗                     |                                       | ⊗        |          |
| Mechanical resistance | Transient stress  |                       | ⊗                                     |          | ⊗        |
|                       | Fatigue/Thermal/Wear stress                                   | ⊗                     | ⊗                                     |          | ⊗        |
|                       | Vibration   | ⊗                     | ⊗                                     |          | ⊗        |
| Reliability           | Life time/MTBF/Failure rate                                   | ⊗                     | ⊗                                     |          |          |
|                       | Failure/Critical cases: winding short circuit, jamming, shock |                       | ⊗                                     |          | ⊗        |
| Dynamic/Control       | Natural modes   | ⊗                     | ⊗                                     |          | ⊗        |
|                       | Bandwidth   |                       | ⊗                                     |          |          |
|                       | Precision   | ⊗                     | ⊗                                     |          |          |
| Power/Energy          | Transient input power   | ⊗                     | ⊗                                     |          |          |
|                       | Energy consumption  | ⊗                     | ⊗                                     |          |          |

**表 3.1**

基於知識的工程（KBE）是一門結合了基於知識的方法和計算機輔助設計的學科，對設計支持也很有用。KBE 軟件包專用於集中特定產品設計的所有知識和專長。這些環境中使用的科學概念和方法如下：

- 知識庫和計算機輔助設計：使用非定向的代數方程式（聲明性方法）來充分利用組件的知識（這些方法可以很容易地組合起來以描述不同的體系結構）。
- 約束網絡，圖論和優化：以聲明形式最初定義的一組方程式經過定向，以便獲得可用於優化算法的計算程序。

可以通過符號計算，間隔計算或人工智能來支持對等式的適應或分析。這些 KBE 工具可以是：

- 鏈接到 Genus Designer 中的 CAD 軟件，該軟件捕獲配置規則並為 Solidworks 執行過程自動化。
- 如 Enventive 中所述，專門用於平面機構的概念設計（優化，公差，靈敏度分析）；
- 開發了多個領域，例如 FST 研究所軟件（TUHH 大學），該軟件在相同的基礎上支持飛機升力機構，液壓網絡和 EMA 的初步設計；
- 常規，例如 TKSolver，Ascend，Cades，Design 43 或 PaceLab 套件，並且通常是聲明性語言，可以實現一組根據設計目標，代數方程可與不同的輸入配合使用。

這些工具可以為設計人員提供有趣而重要的幫助，尤其是當系統由於多種技術而變得複雜時。結合 MDO 工具和專用的元建模技術，它們可能代表機電一體化系統設計的未來。

### 3.2.3 綜合最佳架構的能力

在設計機電一體化系統時，可以在體系結構級別設想許多解決方案。選擇最合適的體系結構的任務是一項複雜的任務，目前主要通過手動定義，評估和比較設想的體系結構來完成。在體系結構級別使用設計綜合和優化技術可以使設計人員設想更廣泛的解決方案，其中包括潛在的創新解決方案，並在形式上進行比較（使用定義明確的指標），以選擇合適的解決方案。最合適的一個。

工程設計綜合[23]是一套技術，可基於對人工製品的目的，其預期特性和設計知識（明確形式化或提取的）的知識，對工程人工製品（2D / 3D 形狀，建築等）進行綜合。優化技術會迭代地修改一些初始解決方案（通常是隨機生成的），以優化測試解決方案的特性，但是，他們很少考慮設計知識來生成可行的解決方案。

耦合設計綜合和優化可以生成可行的解決方案，並找到性能最佳的解決方案。由於探索的設計空間可能更大並且該過程不受認知偏見（例如信念）的影響，因此它通常比手動過程更有效。但是，這需要能夠評估任何生成的解決方案以及所有定義的選擇標準（優化目標/約束）。這種類型技術的使用示例是機器人手臂[24]，真空吸塵器[25]或飛機駕駛艙[26]的設計。

### 3.2.4 機電一體化的安全性和可用性研究系統篇

可靠性，可用性，可維護性和安全性（RAMS）標準通常是在設計過程中進行的研究的一部分。為了針對這些標準（可靠性，可用性，可維護性等）量化不同體系結構的性能，必須對每種考慮的替代方案進行大量此類研究。

目前，這些研究由建立機電系統 RAMS 模型並對其進行分析以得出結論的專家進行。RAMS 模型通常使用專門的形式來構建，例如 Petri 網，貝葉斯網絡，可靠性框圖或更高級的語言，例如 Altarica [27]和 Figaro [28]。

為了加快研究速度，將 RAMS 研究與（描述性或行為）體系結構模型聯繫起來將是一個優勢。實際上，RAMS 研究期間可以重用設計模型中嵌入的許多信息。

第一種方法考慮使用 RAMS 數據（故障模式，可靠性比率等）豐富設計模型，以便能夠自動生成 RAMS 模型。例如，SysML [29]模型（具有特定配置文件）可用於自動計算系統級故障率[30]。

另一種方法考慮將設計模型鏈接到 RAMS 模型，以確保 RAMS 模型的一致性。在 MODRIO 項目[31]的框架中，開發了一個原型，以從 Modelica [32]模型和 Figaro 知識庫[33]自動生成 Figaro 模型。使用 Modelica 特定的構造從知識庫和其他必要信息中聲明與 Figaro 塊的對應關係。

### 3.2.5 產品的功能虛擬表示

機電系統的設計和集成是一個多學科的設計過程，需要多個領域進行協作和交換信息。在當今的大型公司和擴展型企業中，掌握這些信息流對於提高設計階段的效率至關重要。

改善團隊之間溝通的一種方法是共享集成了所有觀點的產品的通用虛擬表示。這種表示方式使不同的團隊可以查看其他團隊的約束，並始終可以訪問設計的最新版本。此外，它還可以在仿真過程中考慮來自其他領域的影響。

綜合觀點還可以為決策提供更堅實的基礎，因為它可以了解所有相關學科的所有限制和目標。

這種表示形式的主要技術推動因素是能夠在使用不同工具和數據格式的不同團隊之間交換工程數據，以及能夠集成來自不同學科的模型的能力。

### 3.3 驗證和驗證中的挑戰

#### 3.3.1 虛擬驗證系統的能力

機電一體化系統的驗證對工業界（尤其是在航空航天業中）是一項非常昂貴的任務。確實，這既需要係統可用，又需要開發和製造測試平台，而這些測試平台通常無法在程序之間重複使用。降低測試成本的一種方法是虛擬測試。該實踐旨在開發虛擬手段（系統模型和系統仿真環境）來測試系統並驗證其是否符合要求。例如，實際機翼上的測試可以用有限元模型代替。

虛擬測試的主要好處是成本，因為虛擬測試台的成本通常比實際測試台低得多，並且通常可以從一個程序重用到另一個程序。實際上，每個程序的許多系統都是相似的，並且它們的模型通常可以通過有限的努力就可以適應新的設計。

不太明顯的好處是，虛擬測試台可以在比系統在其生命週期中遇到的實際刺激更近的條件下刺激系統。例如，在機翼彎曲測試中，將載荷局部施加在實際的試驗台上，而載荷則可以均勻地施加在虛擬的試驗台上，從而以更準確的方式表示實際載荷。在系統模型中，某些方面（例如熱效應）也可以更輕鬆地考慮到，而如果需要進行實際測試，則它們需要非常昂貴的測試手段。

要啟用虛擬測試，必須很好地管理驗證過程的多個方面。首先，當然必須對模型及其仿真環境進行驗證。這可以通過將結果與測試台數據或飛行測試數據進行比較來完成。在這方面，需要模型校準和不確定性管理技術。

#### 3.3.2 正式向模型供應商發出模型請求

模型越來越多地用於在系統上執行驗證和確認（V&V）活動。這些模型通常由系統的供應商（內部或外部客戶）設計。但是，在功能，有效性範圍，精度等方面能夠陳述模型期望的內容仍然是一個挑戰。實際上，請求者對總體仿真環境（即其他交互模型，仿真輸入等）具有很少的看法，而很少以正式的方式

傳達給模型開發者。在將預期模型實際提供給模型請求者之前，這通常會導致多次迭代。

最近，模型身份證被提出作為模型要求的標準描述[34]。MIC 允許描述模型的某些所需特性，以指導模型的開發。首先，必須定義模型的接口，以描述模型的端口和交換的變量。第二，模型信息的四個部分必須填寫如下：

對象-型號名稱，粒度級別，參考文檔

對象上下文用法-時間計算，工具

方法-模型尺寸，方法，線性

模型質量-準確性，驗證，確認

這種標準化模型規範的另一用途是對現有模型的重用。實際上，對模型的特徵進行形式化可以在模型數據庫中進行搜索，並可以在不同的上下文中重用模型。

### 3.3.3 交換和無縫集成模型的能力

工業合作夥伴之間關於建立多系統仿真平台以驗證機電一體化系統的目標，主要要素是交換和集成模型的能力。目前，模型集成商傾向於迫使其供應商使用一種工具並遵循特定的建模程序（例如 AP2633 [35]）。在這方面，模型供應商希望使用自己的工具對系統進行建模。因此，能夠交換和集成來自不同工具的模型將是巨大的進步。

功能模型接口（FMI）計劃[36]通過提供獨立於工具的標準來交換動態模型和進行聯合仿真而朝著這個方向發展。它允許生成“中性格式”模型（以 C 代碼和 xml1 文件的形式），這些模型可以無縫集成在兼容平台中。當前，大約有 70 種工具支持該標準。

這些標準模型交換形式的使用提供了靈活性，因為可以使用許多不同的工具來開發用於仿真平台的組成模型。因此，模型提供者可以自由選擇和更改其首選工具，而不會影響整個仿真框架。在模型集成者方面，還可以確保靈活性，因為可以在一組可用工具中選擇和更改集成平台，而不會影響現有模型。

這種類型的標準還可能替為滿足特定需求而內部開發或由工具供應商出售的工具之間的許多點對點接口。對於公司而言，這可以節省大量開發或許可成本。

### 3.3.4 通過以下方式對機電系統進行形式驗證建模

在早期設計階段檢測系統缺陷一直是基於模型的系統工程（MBSE）的核心，以減少全球開發時間，同時提高最終產品的質量。時至今日，仿真是驗證開發中系統行為的最常用方法，但是存在一個固有的主要缺點，即有限數量的測試場景的局限性。

正式的驗證技術可以證明模型確實符合其規範，即使情況是無限的。其中，模型檢查能夠以自動化過程在計算機上執行驗證。

為了確保最佳的可靠性，在航空航天，鐵路，微型計算機等高科技工業領域，以及更普遍的任何關鍵嵌入式系統的開發中，模型檢查的使用已經很普遍。技術和工具已經發展起來，克服了模型檢查的某些原始限制，如今可以處理具有連續和離散零件的物理模型。這被稱為混合模型檢查，這開闢了新的應用前景，尤其是在機電一體化領域。

在當前的現有技術中，混合模型檢查器通常限於證明安全屬性（即係統將永遠不會進入一組特定的狀態），因為它們經常依賴於過度逼近。例如，這使他們成為證明飛機避碰機動性正確性的良好候選人。

圖 3.3 顯示了另一個案例，衛星的任務是根據請求捕獲地球圖像，並在可見時將其下載到地面站。目標是根據特定的地面站配置和每個軌道的最大請求數量，正式驗證衛星的存儲緩衝區不會超出。這些請求離散的，而數據傳輸則是連續建模的。



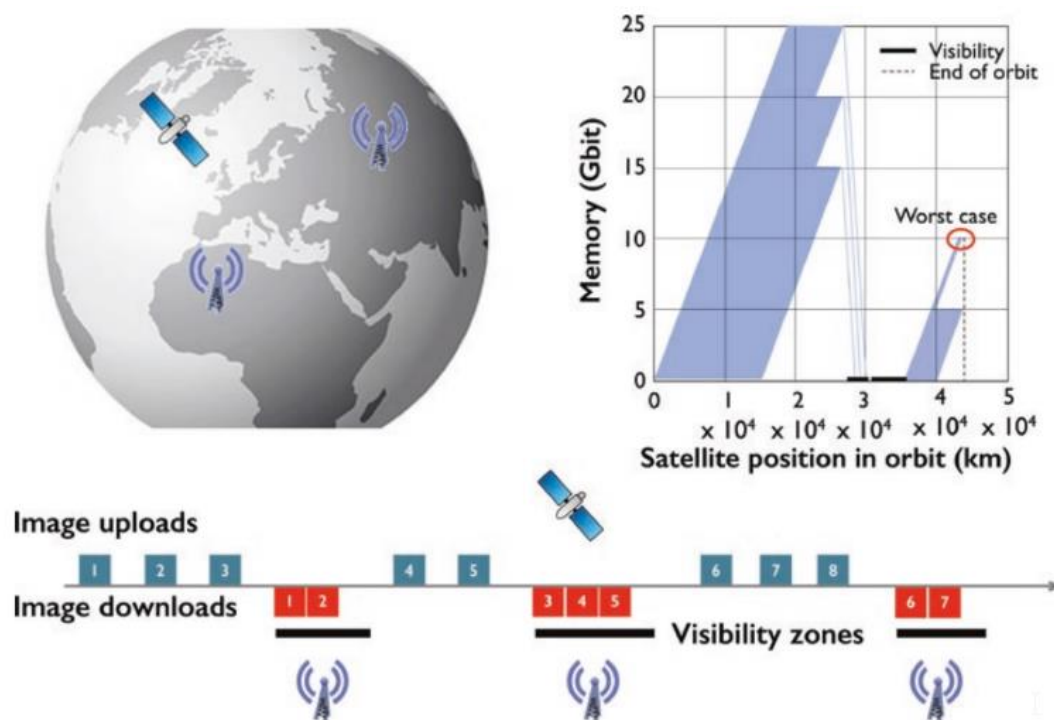


圖 3.3 使用混合模型檢查驗證衛星存儲器

混合模型檢查器的表達方式（例如，它們支持哪種微分方程）和它們提供的過近似方法彼此不同。

當證明安全性還不夠時，混合定理證明如 KeYmaera [37] 可以用作機電一體化背景下的另一種形式驗證選擇。這些嘗試使所需屬性的數學證明自動化，但是通常需要用戶在演示過程中進行一些高級輸入才能得出結論，這對於複雜的系統可能非常棘手。

### 3.3.5 優化測試活動的能力

有時，由於複雜的建模偽像或可伸縮性問題，對機電一體化系統進行形式驗證是不可行的。不論是否可能，一旦構建了產品以檢查其圖 3.3，就仍然需要進行測試活動。圖 3.3 使用混合模型驗證衛星存儲器，以檢查是否符合規格模型並根據要求對其進行驗證。無論如何，對可以執行的有限數量的測試用例的限制

（首先是在仿真中，然後在現實生活中）使得必須確定最相關和最具代表性的測試用例。

趨勢是自動生成最佳測試用例。有兩個主要因素決定如何完成此任務：首先是測試選擇標準，該標準定義了驅動測試用例生成的因素，其次是測試生成技術，即實際產生結果的算法。兩者的典型示例在下面給出[38]。

測試選擇標準：

- 結構模型覆蓋標準-這些利用模型的結構來選擇測試用例。他們根據從控制流到代碼的思想處理整個模型的控制流。
- 數據覆蓋標準—想法是將數據範圍分為等效類，並從每個類中選擇一位代表。通常通過邊界值分析來補充這種劃分，在邊界分析中，還額外選擇了由約束條件確定的數據範圍或邊界的臨界極限。
- 需求覆蓋標準-旨在覆蓋所有受測非正式系統（SuT）需求。 SuT 需求對系統或測試模型/代碼的可追溯性可以支持該標準的實現。幾乎所有測試方法都將其作為目標。
- 測試用例規範-當測試工程師以某種正式符號定義測試用例規範時，這些規範可用於確定將生成哪些測試。明確決定應涵蓋哪組測試目標。
- 隨機和隨機準則—它們大多數適用於環境模型，因為環境決定了 SuT 的使用模式。一種典型的方法是使用馬爾可夫鏈來指定預期的 SuT 使用情況配置文件。另一個示例是除了 SUT 的行為模型之外，還使用統計使用模型。
- 基於故障的標準-這些標準依賴於通常會以故障模型形式設計的典型故障的知識。

測試生成技術：

- 隨機生成-通過採樣系統的輸入空間來完成測試的隨機生成。
- 圖形搜索算法-專用圖形搜索算法包括節點或弧覆蓋算法，例如 Chinese Postman 算法 2，該算法至少覆蓋每個弧一次。

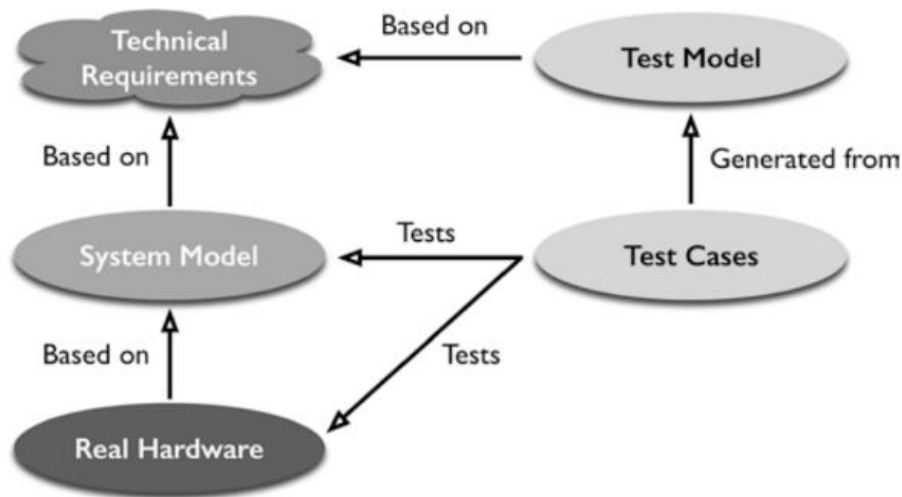


圖 3.4 使用顯式測試模型（頂部）與設計/規格模型的基於模型的測試（底部）

- 模型檢查-模型檢查是一種用於驗證或偽造系統屬性的技術，但可用於基於給定的反例來生成測試用例。
- 符號執行-符號執行的想法是運行一個不具有單個輸入值而是具有一組輸入值的可執行模型。這些表示為約束。通過這種做法，將生成符號跟蹤。通過使用具體值實例化這些跡線，可以得出測試用例。
- 定理證明-通常使用定理證明來檢查直接出現在模型中的公式的可滿足性。在這裡，它使用數學過程來搜索模型的可能執行路徑，從而找到測試用例和反例。
- 在線/離線生成技術-通過在線測試生成，算法可以在測試執行期間對 SuT 的實際輸出做出反應。這個想法也用於實施反應性測試。脫機測試意味著測試用例在運行之前就已生成。一組測試用例僅生成一次，並且可以執行多次。

最合適的測試生成技術通常取決於要處理的源模型的類型。一些方法是根據測試模型創建測試用例的，而另一些方法則將設計模型作為輸入，如圖 3.4 所示。兩者都可以是行為模型，但第一種描述了測試人員對 SuT 進行測試的方式（即，他給出了哪些刺激以及他進行了驗證），而後者則描述了 SuT 的工作方式。因此，觀點（測試者與實施者）以及目標（驗證與解決方案）不同。

### 3.4 運營中的挑戰

#### 3.4.1 使用設計模型改善維護的能力

當前，在航空航天系統的維護中很少重複使用設計模型。但是，有關係統行為的知識可用於改善診斷，預後和維護計劃。

在診斷中，行為模型可用於將對系統的觀察結果與某些故障模式或組件的某些退化相關聯。由於可以檢測到系統故障的根本原因而無需檢查所有部件，因此這允許以更準確和更快的方式估計組件的當前狀態（例如，過濾器堵塞）和維護操作的性能。

同樣，預測活動會嘗試預測系統的未來狀態，以便預測故障併計劃預防性維護任務。例如，通過分析由其傳感器傳輸的數據來估計系統的剩餘使用壽命，可以規劃特定的維護任務（例如，儲油箱加註），以預測所需的資源（備件）並相應地調整操作。

從技術上講，使用模型進行診斷/預後通常需要狀態估計技術（例如線性回歸，卡爾曼濾波器）。這些技術允許通過播放模型的某些參數（故障，降級）來最小化實際系統和模型上的觀測值之間的誤差。以來

模型的幾種配置可能與觀察到的行為相匹配，觀察到的變量的定義是有效診斷的主要驅動力。

實際上，由於所需的計算資源，並非總是可以嵌入模型。但是，可以根據透射遙測（離線 PHM）在地面上完成診斷/預後任務。

#### 3.4.2 使用設計模型改善控制能力

設計模型用於改善操作的另一種用途是將其用於系統控制。實際上，在某些情況下，有必要在系統的控制邏輯中使用行為模型中包含的知識。這稱為模型預測控制（MPC）。如果系統動力學較慢（例如化工廠）或控制系統必須考慮長期使用系統（例如計劃任務資源使用），則 MPC 的使用特別適合。

MPC 的用法示例是能源管理。例如，可以在功率控制算法中使用混合動力推進車輛的簡單模型來優化燃油消耗和給定特定任務的電池使用。與經典控制算法相比，這可以顯著提高性能。但這種智能算法的認證仍然是一個挑戰。

而且，由於嵌入式模型受實時需求的限制，並且由於設計模型通常不是為此類應用程序而設計的，因此必須實現模型的簡化。這種簡化過程會在模型的實時性能和表示性之間進行權衡。

### 3.5 結論

從工業角度看，機電一體化系統設計中的一些主要挑戰已暴露出來。這種發展的主要動力是減少開發成本和時間，以及在成本和性能方面改進設計產品。可以看出，許多挑戰都涉及產品的虛擬化，以改善其設計，驗證或操作。實際上，虛擬化可以以較低的成本在開發的不同階段提供更大的靈活性。在設計中，組件的多樣性和特定領域，機電一體化系統需要在設計過程中將 FEM 和系統級模型無縫集成。為此，借助基於圖形的 MDO 方法，未來的工作可以集中在用於機電一體化大小調整活動和模型易於組裝的專用元模型上。在 V&V 中，未來的工作應集中在機電系統的正式驗證上，因為它將大大降低認證成本。

### References

1. Budinger M, Reysset A, Halabi TE, Vasiliu C, Mare J-C (2013) Optimal preliminary design of electromechanical actuators. Proc Inst Mech Eng Part G J Aersp Eng 228(9):1598–1616
2. Giraud X, Budinger M, Roboam X, Piquet H, Sartor M, Faucher J (2014) Methodologies

for the optimal design of the integrated modular power electronics cabinet. In: European

Conference on More Electric Aircraft

3. van der Auweraer H, Anthonis J, de Bruyne S, Leuridan J (2012) Virtual engineering at work: the challenges for designing mechatronic products. *Eng Comput* 29(3):389–408
4. Hehenberger P, Poltschak F, Zeman K, Amrhein W (2010) Hierarchical design models in

the mechatronic product development process of synchronous machines. *Mechatronics*

20(8):864–875

5. Liscouët J, Budinger M, Maré J-C, Orioux S (2011) Modelling approach for the simulation based preliminary design of power transmissions. *Mech Mach Theory* 46(3):276–289
6. Pahl G (2007) *Engineering design: a systematic approach*. Springer, Berlin
7. Budinger M, Passieux J-C, Gogu C, Fraj A (2014) Scaling-law-based metamodels for the sizing of mechatronic systems. *Mechatronics* 24(7):775–787
8. Sobieszczanski-Sobieski J, Haftka R (1997) Multidisciplinary aerospace design optimization: survey of recent developments. *Struct Optimisation* 14(1):1–23

9. Dassault Systems Isight webpage  
[www.3ds.com/productsservices/simulia/products/isight-simulia-execution-engine/latestrelease/?xtmc=isight&xtcr=1](http://www.3ds.com/productsservices/simulia/products/isight-simulia-execution-engine/latestrelease/?xtmc=isight&xtcr=1) and  
[www.simulia.com/products/portfolio/](http://www.simulia.com/products/portfolio/). Accessed 10 Nov 2015

10. Noesis Optimus website [www.noesisolutions.com/Noesis/](http://www.noesisolutions.com/Noesis/). Accessed 10 Nov 2015

### 3 Future Challenges in Mechatronics 39

11. Phoenix Integration ModelCenter [www.phoenix-int.com](http://www.phoenix-int.com). Accessed 10 Nov 2015

12. Malmquist D, Frede D, Wikander J (2014) Holistic design methodology for mechatronic systems. *Proc Inst Mech Eng Part I J Syst Control Eng* 228(10):741–757

13. Allain L (2003) Capitalisation et traitement des modèles pour la conception en génie électrique, Institut National Polytechnique de Grenoble

14. Genus Designer  
[www.solidworks.com/sw/products/details.htm?productID=2104](http://www.solidworks.com/sw/products/details.htm?productID=2104).

Accessed 10 Nov 2015

15. Enventive [www.enventive.com](http://www.enventive.com). Accessed 10 Nov 2015

16. Pfennig M, Carl UB, Thielecke F (2009) Recent advances towards an integrated and optimized design of high lift actuation systems. SAE Int J Aerosp 3(1):55–64
17. Dunker C, Thielecke F, Homann T (2011) New computer-aided methods for preliminary architecting and sizing of aircraft hydraulic systems, Deutscher Luft- und Raumfahrtkongress
18. Universal Technical Systems TKSolver [www.uts.com](http://www.uts.com). Accessed 10 Nov 2015
19. Ascend [www.ascend4.org](http://www.ascend4.org). Accessed 10 Nov 2015
20. Vesta Solutions Cades [www.vesta-system.fr/en/products/vestacades/](http://www.vesta-system.fr/en/products/vestacades/). Accessed 10 Nov 2015
21. Alber R, Rudolph S (2003) '43'—A Generic Approach for Engineering Design Grammars. In: AAAI Spring Symposium on Computational Synthesis, AAAI Technical Report SS-03-02
22. Pacelab [www.pace.de](http://www.pace.de). Accessed 10 Nov 2015
23. Chakrabarti A (ed) (2013) Engineering design synthesis: understanding, approaches and tools. Springer Science & Business Media, Berlin



24. Chakrabarti A, Bligh TP (2009) An approach to functional synthesis of mechanical design concepts: theory, applications, and emerging research issues. *Artif Intell Eng Des Anal Manuf* 10(4):313
25. Wyatt DF, Wynn DC, Jarrett JP, Clarkson PJ (2012) Supporting product architecture design using computational design synthesis with network structure constraints. *Res Eng Des* 23(1):17–52
26. Albarello N (2012) Model-based trade studies in systems architectures design phases, Ecole Centrale Paris
27. Arnold A, Point G, Griffault A, Rauzy A (1999) The AltaRica formalism for describing concurrent systems. *Fundam Informaticae* 40(2–3):109–124
28. Bouissou M, Villatte N, Bouhadana H, Bannelier M (1991) Knowledge modelling and reliability processing: presentation of the FIGARO language and associated tools, Electricite de France (EDF)
29. OMG SysML [www.omg.sysml.org](http://www.omg.sysml.org). Accessed 10 Nov 2015
30. Helle P (2012) Automatic SysML-based safety analysis. In: *Proceedings 5th International*

Workshop Model Based Architecting and Construction of Embedded Systems (ACES-MB'12), pp 19–24

31. ITEA3 MODRIO [www.itea3.org/project/modrio.html](http://www.itea3.org/project/modrio.html). Accessed 10 Nov 2015

32. Modelica Specification 3.2 Rev. 2  
[www.modelica.org/news\\_items/release\\_of\\_modelica\\_3\\_2\\_](http://www.modelica.org/news_items/release_of_modelica_3_2_revision_2)

revision\_2. Accessed 10 Nov 2015

33. Bouissou M, de Bossoreille X (2015) From Modelica models to dependability analysis.

IFAC-PapersOnLine 48(7):37–43

34. Sirin G, Paredis CJJ, Yannou B, Coatanea E, Landel E (2015) A model identity card to support simulation model development process in a collaborative multidisciplinary design environment. IEEE Syst J, 1–12

35. Benac C (2003) A380 simulation models. Airbus Stand Dev New Media Support Centre [OL]

36. Blochwitz T, Otter M, Åkesson J (2012) Functional mockup interface 2.0: the standard for tool independent exchange of simulation models. In: 9th International Modelica Conference

37. Symbolaris KeYmaera [www.symbolaris.com/info/KeYmaera.html](http://www.symbolaris.com/info/KeYmaera.html). Accessed 10 Nov 2015

38. Zander J, Schieferdecker I, Mosterman PJ (2011) Model-based testing for embedded systems. CRC Press, Boca Raton