

電池組建模、模擬以及在多核即時目標上之實現

Battery Pack Modeling, Simulation, and Deployment on a Multicore Real Time Target

Javier Gazzarri, Nishant Shrivastava, Robyn Jackey, and Craig Borghesani
MathWorks 公司

摘要

電池管理系統 (Battery Management System, BMS) 的設計是一項複雜的任務，需要摹仿電池單元(battery cell)在各種操作條件下的電化學行為之複雜模型。等效電路(Equivalent circuits)非常適合這個任務，因為它們提供的擬真度(fidelity)和模擬速度之間的平衡，其參數可以直接反映實驗的觀察，且可調整擴展的。可擴展性(Scalability)對於即時(Real-Time)模擬的階段特別重要，因為電池組(battery pack)模型會在一個物理上連接到負責監測和控制的外接硬體之即時(Real Time)模擬器上執行。隨著現代電池系統多由數百個電池單元組成，因此若能採用一種建模和模擬方法，能夠擴充到同時處理多個基本電池單元範例中，且還能保持即時的效能是非常重要的。

在之前的文章中，我們曾提出建立單一電池單元(battery cell)模型的技術，該模型包括電池單元的電化學指紋，是以等效電路模型擬合到實驗數據為基礎。在本文中，我們將之前的模型擴展為一個電池組(battery pack)，使用自動化程式建立電池單元、配置及連接，然後再利用這個設計，擴展運用到任意尺寸和電子拓撲(topology)等。此外，本文中我們也對在多核目標硬體上執行即時任務的電池分區機制(partitioning schemes)進行評估，以確保能有效地利用硬體資源、平衡計算負載，以及對分散式系統之求解精度所可能造成的計算延遲的潛在影響進行研究。在執行即時任務並轉成 C 程式碼之前，利用模型探查並評估分區模型確實有助於確認多核心的配置，其結果是：最低的平均周轉時間，以及最短的任務起點/終點間的時間間隔。

而且透過本方法所獲得的模型，用來後續建立多種操作情境及充電設計、平衡設計以及安全等相關程序也相當有用。

簡介

許多電池應用，特別是高功率等級的電池，其所需要的串聯或幾個並聯的電池單元連接是由電特性負載的拓撲來決定。當電池組的尺寸增大，所需要的模擬系統之求解器計算量需求也隨之增加，如果模型需要即時模擬進行硬體迴圈 (hardware-in-the-loop, HIL) 測試，這個問題將變得更加嚴重。現代電腦硬體所提供的功能，例如平行處理(parallel)可以增加模擬的速度，但一般來說仍難以充份利用這些功能優勢，主要還是來自於模型執行過程中多執行緒(multithreading)所帶來的挑戰。

在多核目標上執行即時任務應該是適合電池組的模擬，因為電池組的重複性質代表著我們可用一個相對簡單的方式來均勻地分割模型。儘管有著如此明顯的

簡單性，就求解精度之數據傳輸延遲來說，模型分割/分區的影響性仍是一個比較難以解決的問題。

在程式碼轉檔執行到 HIL 平台之前，很自然的，第一步是透過電腦進行桌機模擬，這項工作可讓設計人員在程式碼轉檔並執行到目標硬體之前，透過模擬技術確認電池模型的最佳分區模型。這一步驟能使設計人員在橫跨多個執行緒 HIL 應用中找到最均衡的計算負載分配。

本文第二項工作的目標是，提供一種能夠支援任何數量配置成串聯或並聯的電池單元組件的可擴充的方法，我們利用 MATLAB 程式來建立、配置和連接每個模型組件，包括電池單元模塊和電力零件的電池負荷及模型分區。

模型分區(PARTITIONING)

隨著系統複雜性的增加，模擬硬體上的運算需求也隨之增加。想要進行電池組的建模及模擬，會需要一個模型能涵蓋整個電池組進行熱管理和充電平衡，並且可以進行即時模擬。

執行硬體迴圈 HIL 測試，即時(Real-Time)運行是基本的要求。對於大的模型，即時性能可能難以實現，特別是如果系統包含的零組件出現數字僵硬(numerically stiff)，以及子系統具有不同的時間常數等狀況。解決大型系統即時模擬問題的方式之一是，將多核的運行透過模型分區來分散運算負載；透過這種方法，一些任務可被同時執行，這可降低每個模擬步驟之求解器的執行時間。

在 Simulink 軟體中，用戶可在最頂層指派個別任務到模型的不同區塊，以對目標硬體配置模型。每個任務都需要一個相關的採樣時間，這必須是頂層模型的整數倍。這種模型劃分可在不考慮目標硬體規格、不需使用者干預，透過一個調度器(scheduler)即可以最佳的方式指派任務到 CPU 的多核心中。從硬體映射而來的這種模型的分拆方法，可以將設計擴展而且可輕易的使用到未來日益強大的目標硬體中。在圖 1 中，每個模型模塊(block)都配有指派的任務，之後 Simulink 調度器將據此產生用於執行多執行緒的 C 程式碼產生，以進行後續目標硬體的實現。

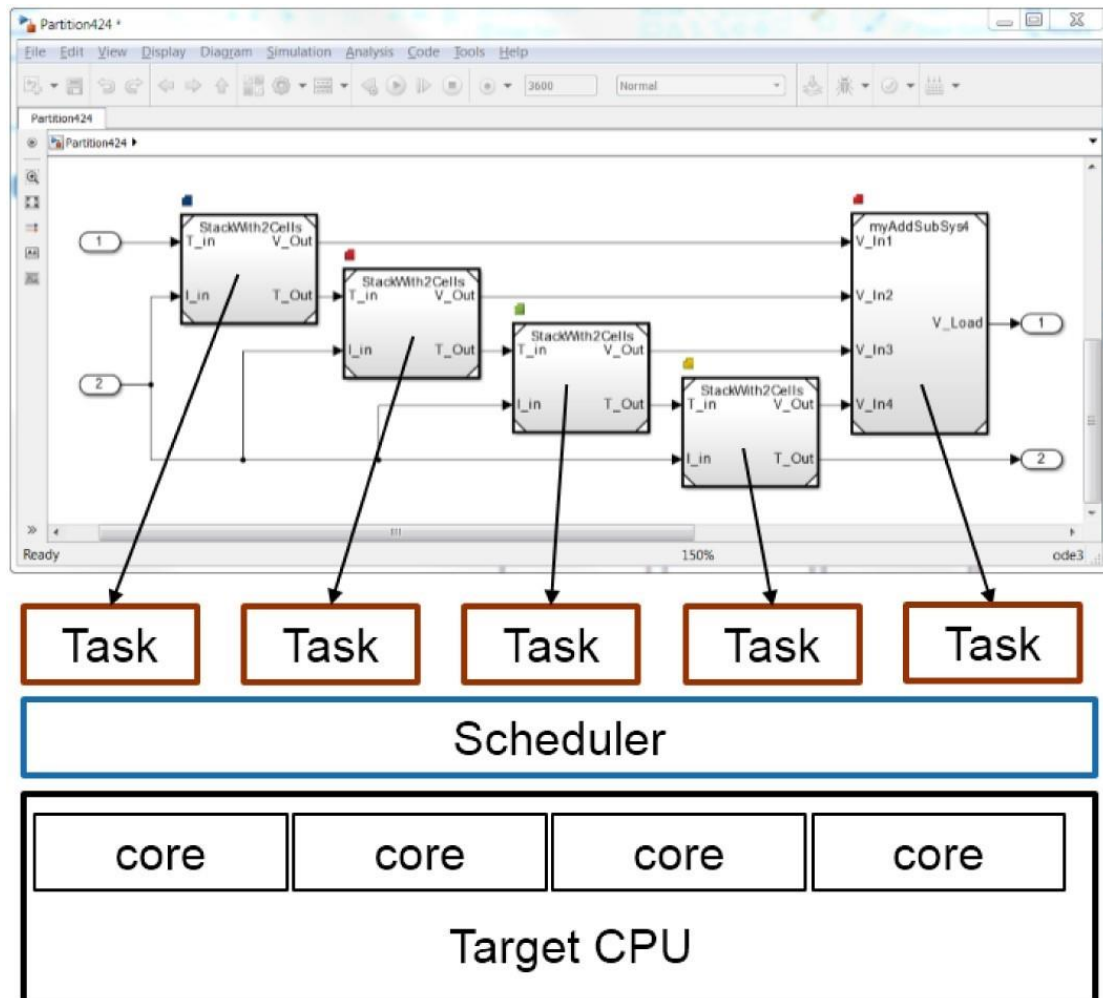


圖 1: Simulink 中執行分區的示意圖。使用者可分配不同的任務到模型的不同部分中，而 Simulink 調度器(scheduler)則可找到計算負載[4]的最佳硬體分佈。

雖然並行(concurrency)只可能在目標硬體的執行階段執行，Simulink 中仍提供了分析工具(profiling tool)，以評估在桌機上進行模型分區的性能。此工具可針對運算負載平衡去模擬模型分區的效能，同時也能顯示每一任務平均周轉的最多次數等量化資料，我們利用這個工具來評估一個電池組的電池分區策略。

同質(homogeneous)結構的電池組，使得它適合並行(concurrency)因為該模型可被劃分成類似的單元在運算硬體中平衡負載，這很重要；但是，為了確保分區所可能導致的延遲和數據傳輸的延宕的區分模型驗證，並沒有驗證求解器的性能。因此在下章節“結果”部分的第一部分，我們將詳細描述這個驗證步驟；之後其他章節則描述八個鋰離子電池串聯的電池組結構其最佳分區方案的過程。

結果

基準模型

用於驗證分區策略的基準模型由八個電池單元串聯，各電池單元包含一個含充電狀態之等效電路以及溫度的相關參數，如在[1,2,3]的描述，圖 2 表示基準模

型與分區模型的比較。電池單元、電壓感測器、電流源(current source)，和熱傳(heat transfer)模塊都是 Simscape 軟體函式庫的一部分，它們相互連接形成一個物理（非因果）網絡。在本模擬中，25A 放電脈衝序列在一小時內從一個充飽電的電池組中部分慢慢放電。

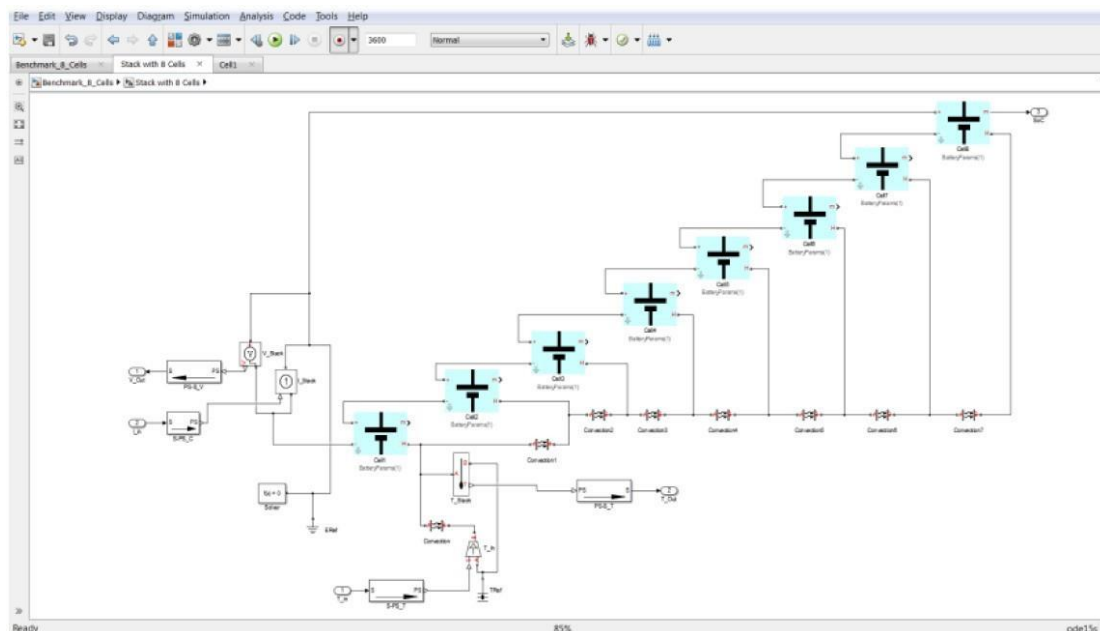


圖 2: 八個鋰離子電池單元串聯連接的基準模型圖。熱流連接模塊模型在相鄰電池單元之間進行熱導傳遞。基準模型採用變步隱式求解器（variable step implicit solver, ode15s）所推薦的數值剛性(numerically stiff)系統。

輸入到模型的輸入值是放電電流和環境溫度，輸出值是電池組電壓和溫度。基準模型使用變步長求解器(variable-step solver)做為最大精度。隨後的模型版本都配置了固定步長求解器(fixed-step solver)，目的是為了進行即時模擬。比較變步和固定步長的結果以確定任何可能損失的精度是很重要的，特別是電壓和電流的快速變化狀況下。

圖 3 顯示電壓 (V)、溫度 (°C)、電流 (A)，和電荷（或 SOC）追蹤電與熱刺激的結果。

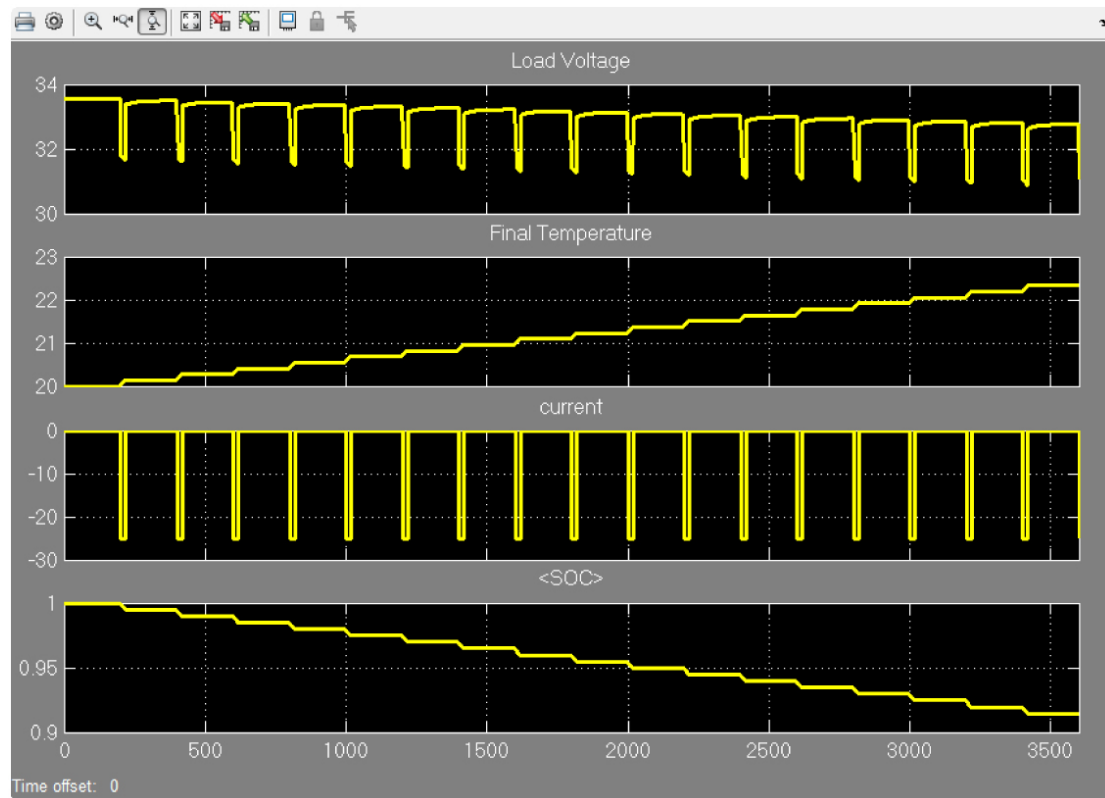


圖 3: 電池組之電壓 (V)、溫度 (°C)、放電電流 (A)、和 SoC (-) 做為比較的基準模型。

圖 4: 為基準模擬和三個分區機制的八組電池組電壓響應圖：1) 無分區的變步長求解器，2) 無分區的固定步長求解器，3) 每四個電池單元為一組，分成兩個任務的分區，和 4) 每兩個電池單元為一組，分區為四個任務的分區。

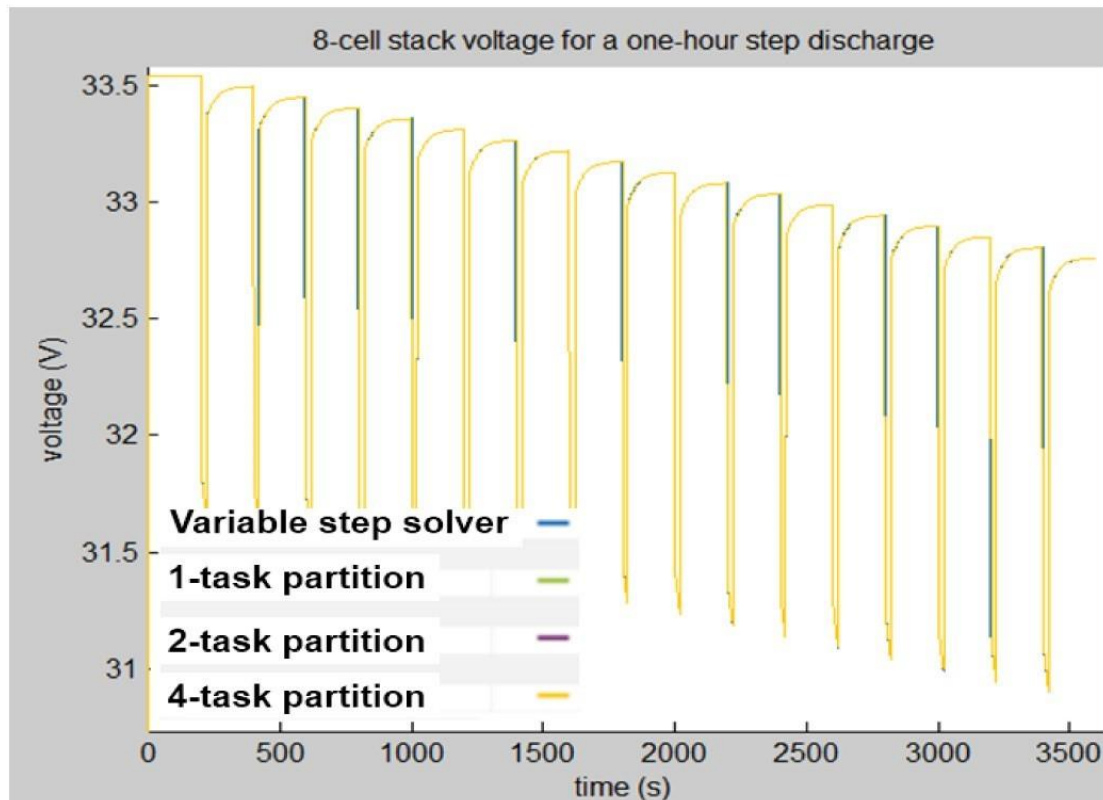


圖 4 顯示出所有分區方案疊加在基準模型的電壓響應結果，這結果的相似性驗證出了該模型的三個分區版本：

- 1) 一個任務分區：一個任務，8 個電池單元的電池組，採用固定步長求解器。
- 2) 二個任務分區：劃分為兩任務，每四個電池單元中為一組，採用固定步長求解器。
- 3) 四個任務劃分：劃分為四個任務，每兩個電池單元為一組，採用固定步長求解器。

分區程序

針對給定的電池尺寸，有許多種電池的分區方案可以用來進行即時的並行實現執行，例如一個具有四核心的即時電腦會建議以每兩個電池單元為一組，分為四區的分區較佳。進行區分時，考慮到影響模型邊界的性能和精度是很重要的，因為這些安排也會影響到任何兩個相鄰分區之間的潛在數據傳輸延遲。因此在執行時間改善和延遲管理進行妥協，最後將會找出一個最優化的方案。

將模型建立的工作流程自動化，將是簡化探索找出最佳分區方案任務的方法。這種自動化工作流程可以讓使用者評估多種配置，並評估多核心計算的好處。使用者還可以權衡各種調度器主導的效能，以及求解器對各種任務間數據傳輸延遲的影響。Simulink 提供了一個分析工具可執行這種權衡之評估。

為了支援自動探索不同電池分區結構的方便方法，我們用 **MATLAB** 程式建立了一個電池模型；這種方法提供輕易的調用一個函式，就能建立一個電池組模型及其分區的任務，程式是根據所提供的參數將數個模塊單元組合在一起，我們將基本函數命名為 `CE_CreateModel`，且它有七個參數。

```
CE_CreateModel(ModelName, n_Stacks,  
n_Cells, n_Tasks, base_rate, f, choice)
```

由參數到函數會需要：模型名稱、電池組將被分割成電池堆(**stacks**)的數量、每一電池堆中電池單元的串連數量、電池模型被分區後的任務數量、高層求解器取樣率、模型任一部份需要以較低頻率來求解的相乘採樣率之整數因子數，以及選擇桌機模擬或進行即時模擬等。

`CE_CreateModel` 這函數會需要以下的模塊被事先定義好：

- 1) 電池單元(**Cell**): 模塊中先定義好以等效電路為基礎的電池類型，包括熱力效應的。
- 2) 負載(**Load**): 界定好電池充/放電循環。
- 3) 輸入(**Input**):指定之環境溫度。
- 4) 輸出(**Output**): 此模塊包含輸出模塊，如繪圖範圍。

透過使用這些 **Simulink** 模塊，**MATLAB** 程式腳本會自動地放置和連接任何數量的電池單元模塊之電和熱結構，然後組成一個電池組模型，將電池組連接到其負載，並且指定配置的首要選項和分區。函數 `add_block` 和 `add_line` 被用來做為程式層級以操作 **Simulink** 模塊。下面我們將說明如何來建立一個電池堆，並進行一系列的相鄰電池單元的電連接串聯。

```
% Placement of n_Cells battery cell blocks and  
creation of a handle to its port designation  
for i = 1:n_Cells  
    myCells(i) = add_block('LiBatteryBlocks/Lithium  
Cell', [StackName '/Cell', num2str(i)]);  
    myCellPorts(i) = get_param(myCells(i),  
'PortHandles');  
end  
% Series connection of battery cell blocks  
for i = 1:n_Cells-1  
    add_line(StackName, myCellPorts(i).LConn(1),  
myCellPorts(i+1).LConn(2));  
end
```

這段 **MATLAB** 的程式表示負責任務的彙整，即分派模型模塊到不同的任務中，以便讓調度器能夠針對分配的執行進行多執行緒的工作。下面的範例程式碼表示任務是如何分派給電池單元模塊的，向量 `myBlock` 是用來控制每個電池堆，以讓使用者可以指定一個單獨的任務。


```

k = 1;
l = 1;
for i = 1:n_Tasks
    for j = k : l*blocks_per_task;
        taskConfig.map(myBlock(j),myTask(i));
    end
    k = k + blocks_per_task;
    l = l+1;
end

```

圖 5 顯示如何利用本技術建立電池組的範例。在每個電池堆 **Stack1**, **Stack2**, **Stack3**, 和 **Stack4** 模塊中都有兩個串聯的電池單元在裡面，每個模型上方的小方塊則參照模塊的指派任務到其所分配(mapped)的地方。

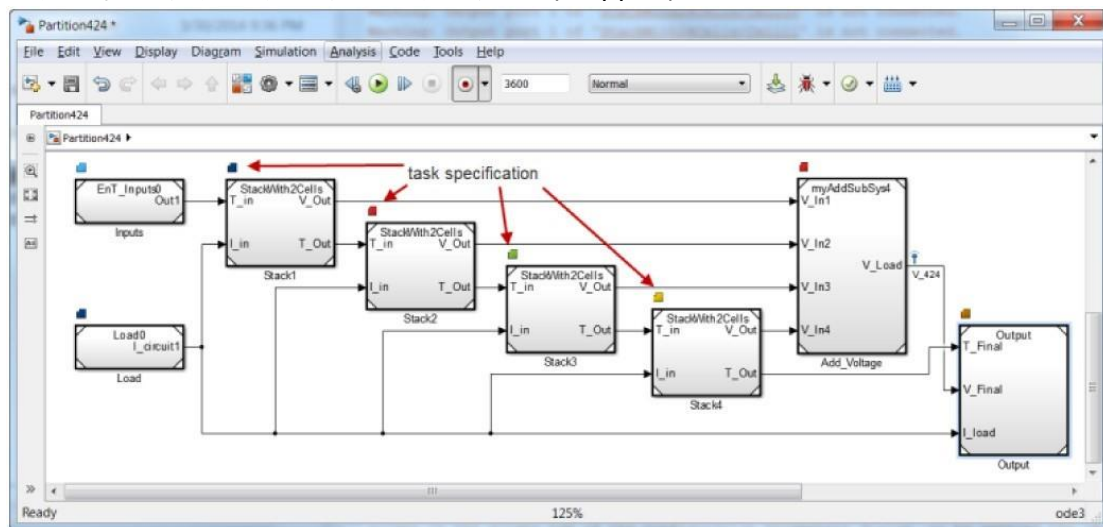


圖 5: 8 個電池單元組模型的頂層分區，分為 2 個電池堆和分配成四個任務。

雖然所有在頂層的模塊被需要有任務分配，在本範例中，電池組外部的相關四個模塊之計算負載是可以忽略不計，因為這些模塊是由非常簡單的元件所組成。圖 6 顯示出每個電池堆模塊內簡單的雙電池電路。

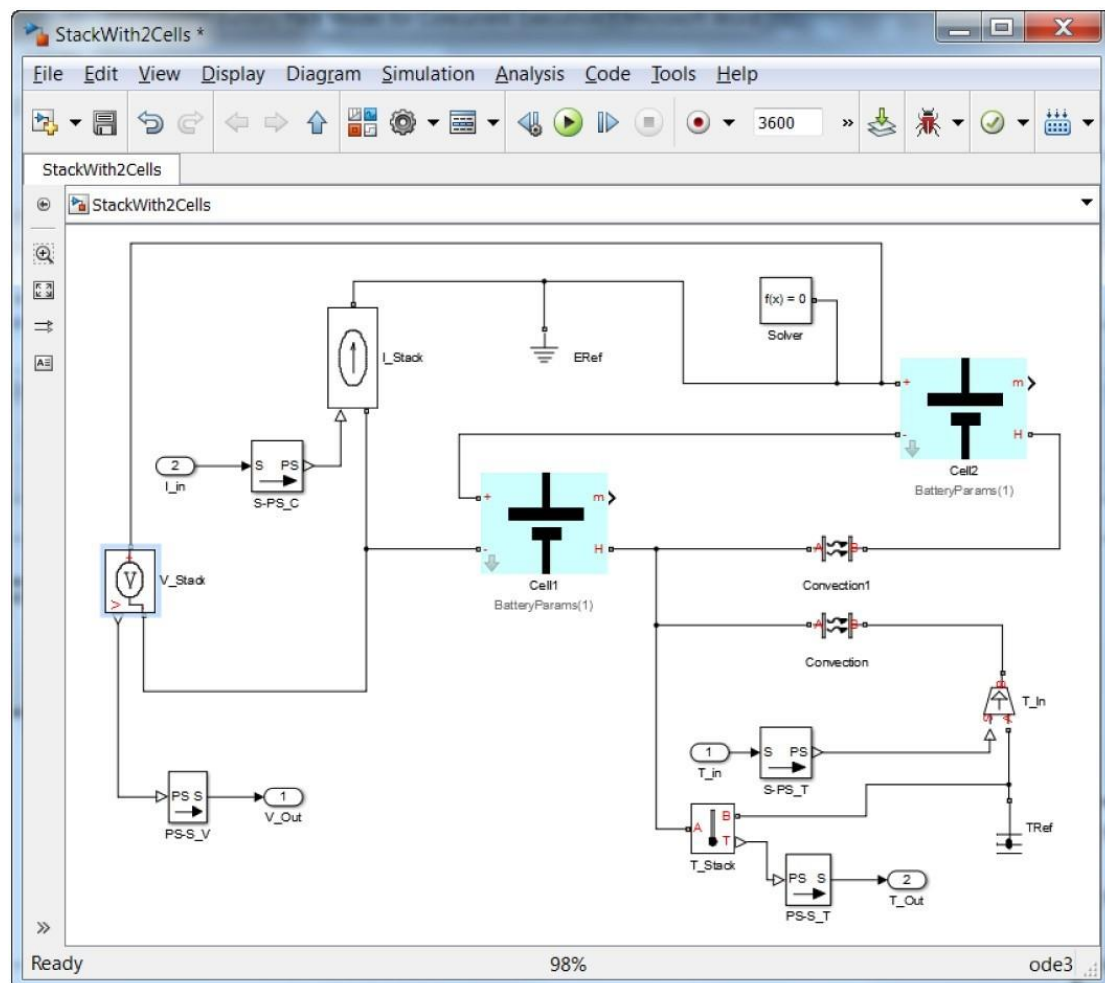


圖 6: 圖 5 電池堆內的電池模塊圖，兩個電池單元以電和熱串連在一起。

在桌機進行分析

Simulink 中提供了一個分析工具，可以讓模型在下載到即時電腦評估計算負載平衡之前，在桌機進行模擬。此分析器的結果包括一個圖表，顯示了負載分佈、每任務分派到模型中的平均和最大周轉時間、以及顯示該任務到核心分派的圖表。下面的章節將說明如何在本工具運用在本電池模型中，我們用的是聯想 (Lenovo) 的筆記型電腦與英特爾®Core™ i7-2620M 雙核 CPU，在 Windows 7 6.1 上執行 2.7GHz 超執行緒(hyperthreading)任務。

案例 1 – 沒有分區（或一個任務分區），固定步長求解器@ 10 毫秒（具有八電池單元的電池堆和一項任務）

這種情況下和基準模型相同，除了它使用固定步長求解器之外。本案例沒有執行並行模擬任務，因為八個電池單元都被分派到整合到單一任務中。此電池堆的平均周轉時間是 209 微秒(209 μ s)（圖 7）。負載平衡是，當然是不適用在這種情況下，因為沒有分區。重要的是，由於這報告是在非即時的桌機電腦上各種條件的效能結果，其結果不應該被視為是真正執行即時模擬的效能定量預測，因為即時模擬將會在不同的硬體上執行。

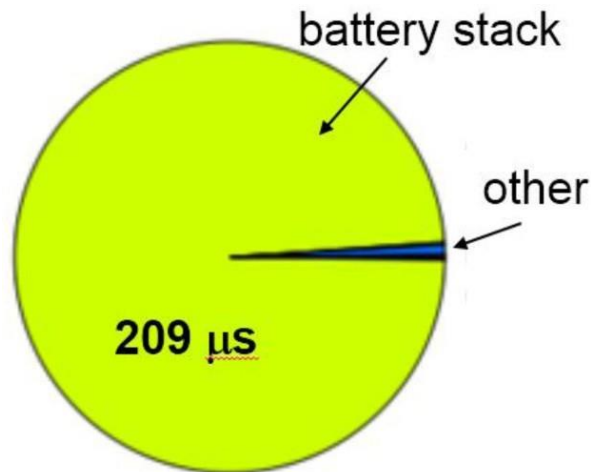


圖 7: 沒有執行並行的分析報告，固定步長求解器@ 10 毫秒(10 ms)。負責解決電池堆任務的平均周轉時間為 209 微秒(209 μ s)。藍色部分所對應到的是非電池堆的組件，如來源和 Simulink 模塊。

案例 2 - 雙任務分區（兩個電池堆，每四個電池單元為一組，分成兩個任務）
 第二種情況有兩個任務劃分，分析報告顯示平均執行速度快了兩倍，最大 108 微秒（平均 103 微秒）時間。在這種情況下，分散運算負載到兩個任務中執行有其好處，它並不會讓調度器的效能抵銷。

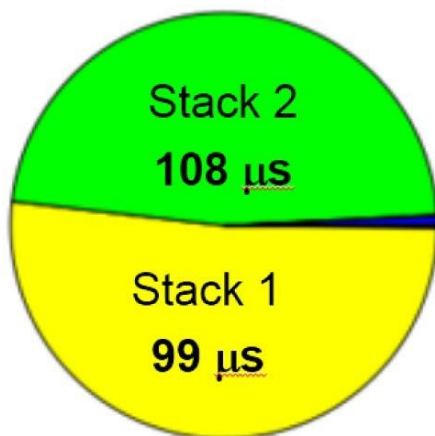


圖 8: 兩個任務劃分分析報告，採用固定步長求解器@ 10 毫秒。平均執行時間在沒有並行狀況下大約是一半，這表示，執行速度幾乎有 100% 的改善效果。

案例 3 - 四項任務劃分（4 個電池堆，每兩個電池單元為一組，分為四項任務）
 由於我們使用的即時電腦配有四核心處理器，將電池堆分為四個主要任務似乎是很自然的選擇，圖 9 顯示了這種狀況的分析結果。

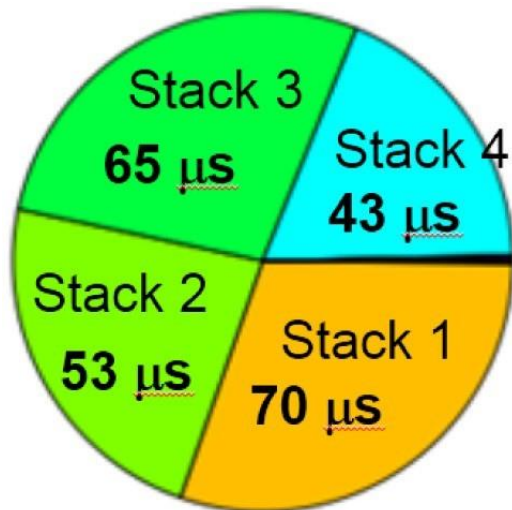


圖 9: 四任務劃分的分析報告，固定步長求解器@ 10 毫秒。

四個任務劃分的平均周轉時間為 43 微秒(43 μ s)到 70 微秒(70 μ s)之間不等，平均為 57 微秒，平均時間為原配置的 25%多一些，顯示了多執行緒執行的關係，調度器和數據傳輸被影響的結果。此外，報告也指出，分區配置的結果稍比平衡計算負載少一些，也表明這個分區配置導致運算負載有些許的不平均，參見餅圖。

案例 4 - 大模型（4 個電池堆，每 24 個電池單元為一組電池堆，四項任務）

自動化建立模型以及分區的好處在處理流程被擴大時更加明顯，本章節描述了一個之前已呈現過的流程，但現在是針對更顯著的大模型。在本案例中，我們模擬了 96 個串聯的電池單元，共劃分為四個電池組，每個電池堆含 24 個電池單元。分析工具指出，每組平均周轉時間介於 703 微秒(703 μ s)和 1165 微秒(1165 μ s)之間，比之前的範例時間多 16 倍（平均），相對於模型的 12 倍大來說，其運算時間相對於模型大小略多於線性增加的比率，因此當模型規模擴大並執行即時模擬時，這將是一個重要的考慮面向。

圖 10 表示出 Simulink 分析工具的進階功能：任務分配示意圖，它標示處理器核心（以零為基礎，從 0 到 3 的四核機器）在每個任務的分派並作為求解器步驟的一個函數。此圖在探索負載均衡作為模擬時間的函數時是很有用的。

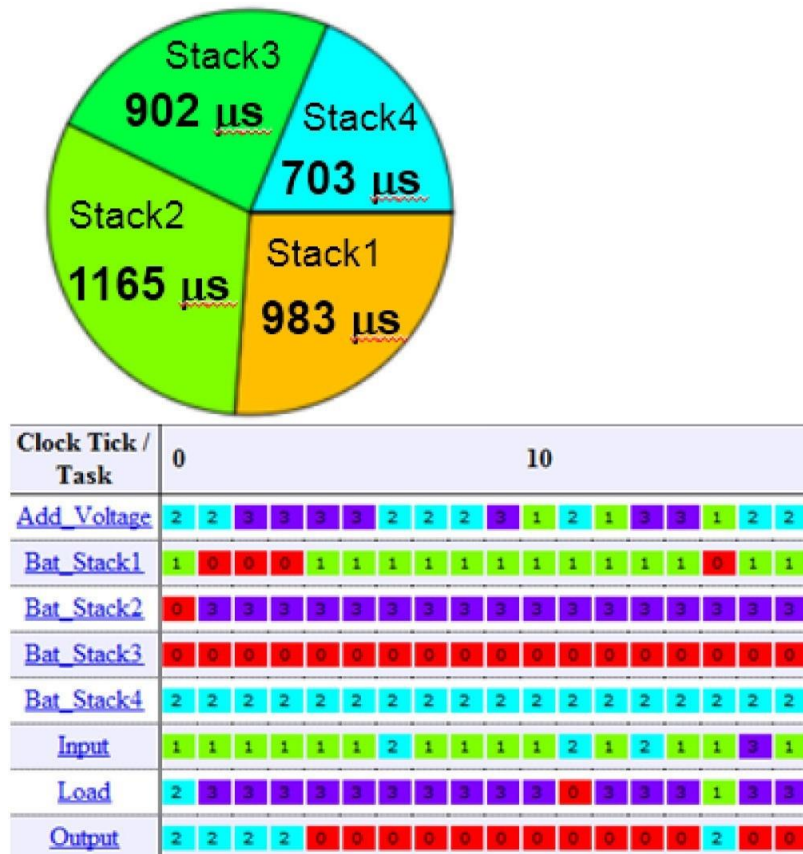


圖 10: 96 個串聯電池模型的分析結果。上圖：執行時間是比前一例子多 16 倍，而電池堆則為 12 倍。下圖：前 20 個求解器步驟的處理器任務分派圖。數字代表從零開始的核處理器之標示。

即時模擬結果

上述過程的最終目標是為了準備一個電池組模型的即時模擬。即時模擬的一個典型應用是進行硬體迴圈 HIL 測試。一般來說，用來進行分析的桌機硬體與真正執行即時運算的電腦不同，從前者所得到的結果是不能定量轉移到後者的，桌機電腦操作系統的非確定性性質強化了這一說法。然而，如果桌機電腦的配置能愈接近即時模擬電腦的架構是最好的。

圖 11, 12 和 13 顯示在一 PC(HP xw4600 with Intel Core2 Quad processor @2.5 GHz, 6.144MB RAM)上，運行案例 1 和案例 2 的 Simulin 的即時核心結果。這些圖表示任務的執行時間和它的硬體核心分配（在每長條圖左下角的數目）在給定的採樣時間的結果截圖。在垂直方向上的長條圖重疊表示並行的任務。仔細看案例 1 和案例 2 的長條圖長度(執行時間)，比案例 2 和案例 3 稍長，這跟之前章節我們透過桌機分析的結果相一致。

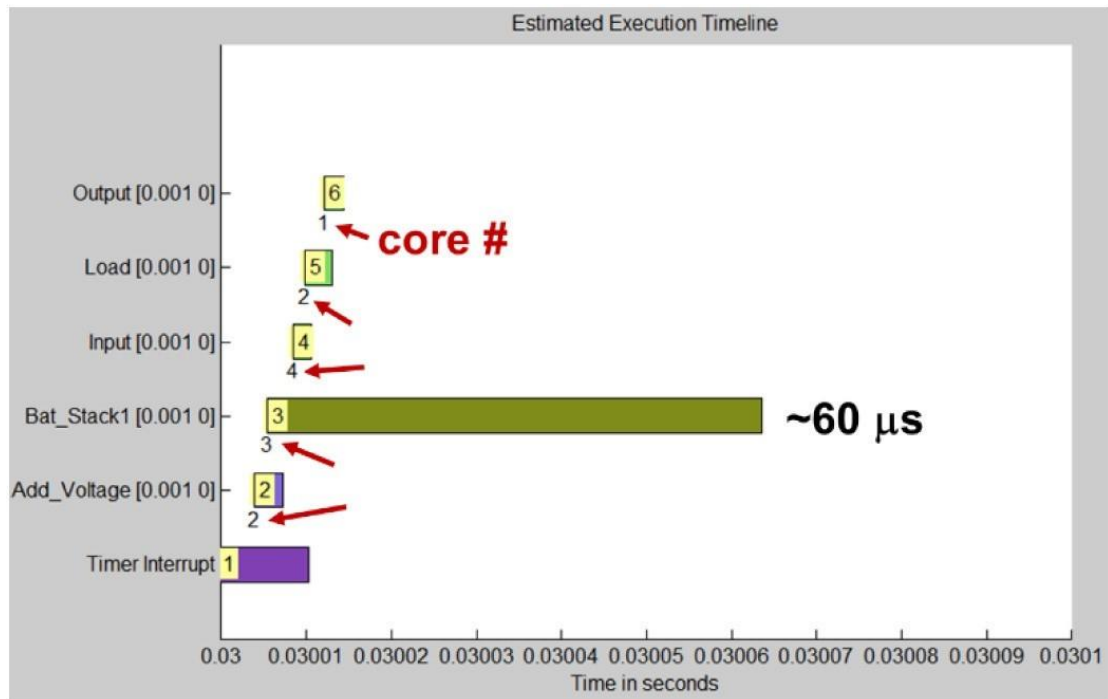


圖 11: 計算負載為 0.03 秒的分佈截圖。案例 1 – 即時，無電池組分區，固定步長求解器@ 10 毫秒(10ms)（一具八個電池單元的電池堆和一個任務 - Bat_Stack1）。箭頭表示處理器的核心數量。

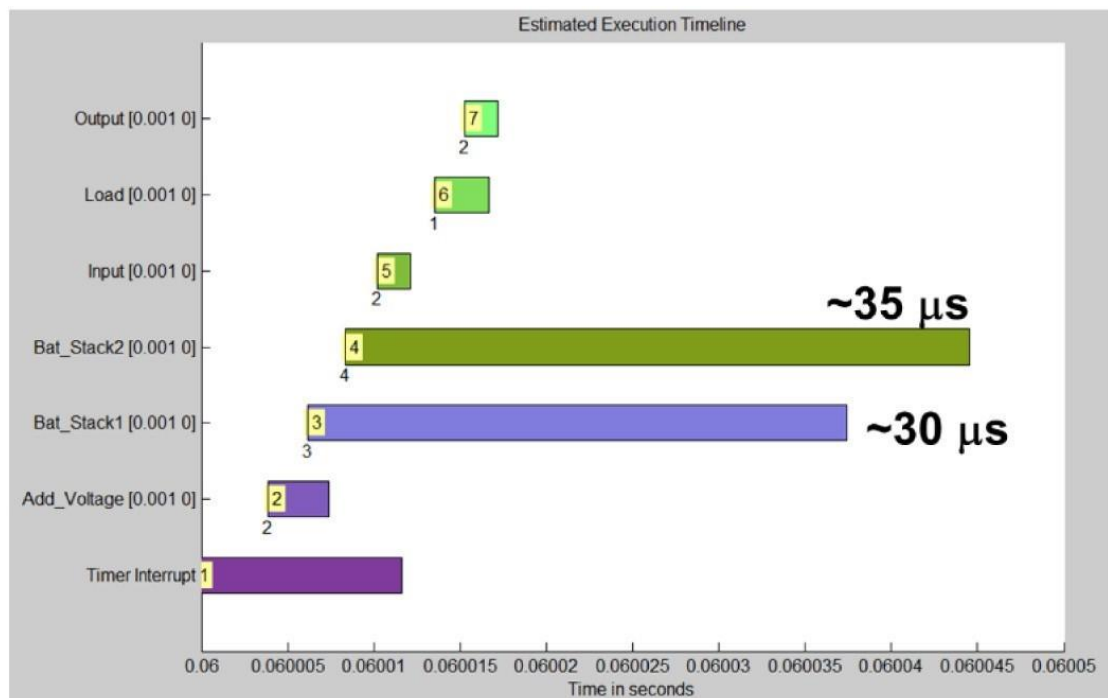


圖 12: 計算負載為 0.06 秒的分佈截圖。案例 2 – 即時。雙任務分區（兩個電池堆，每個各具有四個電池單元，兩個任務）。執行時間(Bat_Stack1 和 Bat_Stack2)幾乎是案例 1 的一半。

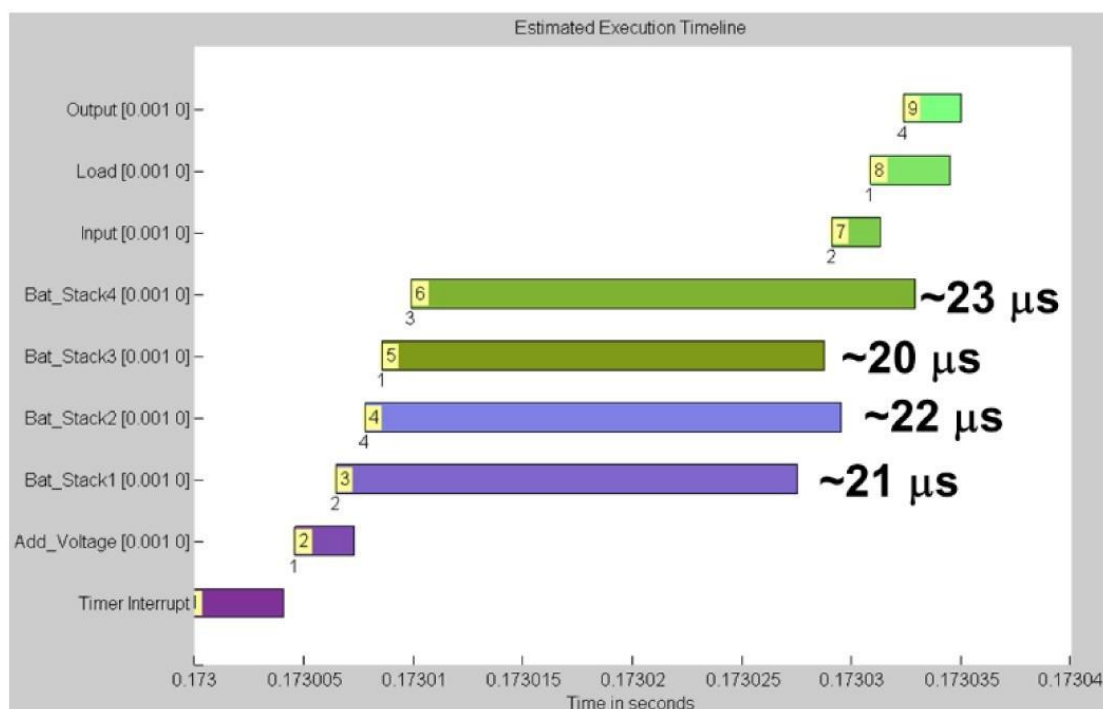


圖 13: 案例 3 - 四任務劃分（4 個電池堆，每個電池堆具有兩個電池單元，四個任務）。相對於案例 2 的兩倍分區，其結果是執行時間比兩倍分區每區的時間還多。

最後，圖 14 表示，96 個電池組模型分成四個任務分區的即時模擬結果。該模型在運行即時沒有超時執行的狀況，初始延遲約為 500 毫秒(ms)。

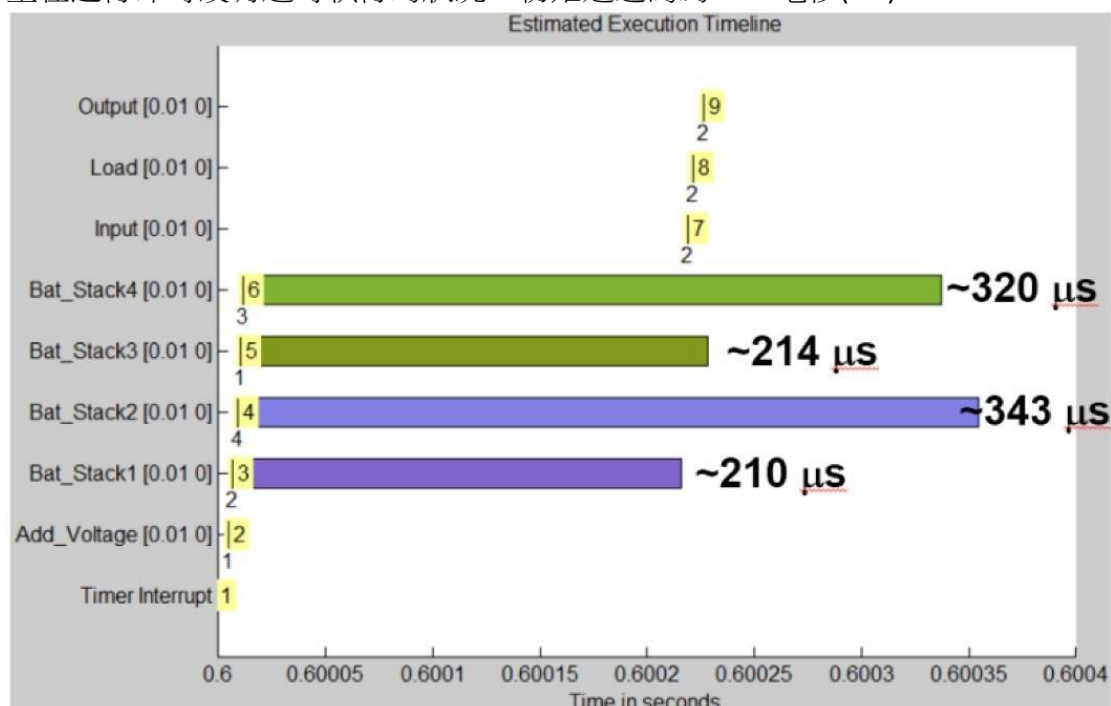


圖 14. 計算負載為 0.6 秒的分佈截圖。案例 4 - 四任務劃分，大型模型（4 個電池堆，每電池堆由 24 個電池單元組成，四項任務）

結論與展望

本文提出了一種靈活的方法，可用來建立適合於即時目標分區和多核執行的電池組模型，本方法的特點是使用 **MATLAB** 程式來自動建立電池模塊、配置、連接和進行電池分區。利用程式腳本建立模型可以更容易地模擬許多分區架構，並確定最少執行時間或平衡計算負載分配的最佳結構。而 **Simulink** 分析工具的好處在於可評估於桌機上權衡不同的分區方案，並顯現出當應用到不同尺寸的模型時，執行時間會呈非線性增加。

所有模型和分區都在一個 10 毫秒採樣速率的即時環境中進行測試。八個電池單元的模型，在不同分區情況下，無論運行在即時環境下或在桌機進行分析預測都具有一致性的效能表現。最後應用在 96 電池單元/四個任務分區的模型中，同樣執行即時模擬，隨著分區數量的增加，相對於初始時間，執行時間亦隨之增加。

在這篇文章中介紹的方法，也適用於本篇案例之外的電池系統設計，只要以自動化的程式腳本來取代或重新命名模塊即可，該程式腳本所建立的模型及設定步驟也可成為一個最佳機制的目標函數，之後透過系統化及自動化的方式來找到最優化的配置。

參考文獻

1. Huria, T., Ceraolo, M., Gazzarri, J., Jackey, R., "High fidelity electrical model with thermal dependence for characterization and simulation of high power lithium battery cells", Electric Vehicle Conference (IEVC), 2012 IEEE International, March 2012
2. Jackey, R., "A Simple, Effective Lead-Acid Battery Modeling Process for Electrical System Component Selection," SAE Technical Paper 2007-01-0778, 2007, doi:10.4271/2007-01-0778.
3. Jackey, R., Saginaw, M., Sanghvi, P., Gazzarri, J. et al., "Battery Model Parameter Estimation Using a Layered Technique: An Example Using a Lithium Iron Phosphate Cell," SAE Technical Paper 2013-01-1547, 2013, doi:10.4271/2013-01-1547