

透過線上估測方法進行即時直流馬達故障偵測

By Karthik Srinivasan, MathWorks

控制演算法通常是根據一個名義上的受控體模型來開發，當受控體動態行為出現輕微改變時，增益及相位邊限(phase margins)等的設計準則可以確保其表現可在可被接受的範圍內。然而，如果受控體動態行為因為一個元件或感應器故障而出現巨幅改變，則有可能出現次佳或者甚至是具有毀滅性的結果。

為了確保這些失誤不會造成嚴重的後果，偵測故障的發生是相當重要的。線上的模型可以幫助您從模型中比對受控體的量測值與預測值，並且在兩者之間的差異超過一定門檻時偵測到故障的發生。

本文將說明如何開發一個直流馬達(DC motor)的線上受控體模型，以及如何使用這個模型來即時偵測馬達的動態變化。

問題設計

我們使用一個業餘直流馬達(圖 1)，連接到一個 Arduino[®] Uno 板子作為設計的範例。

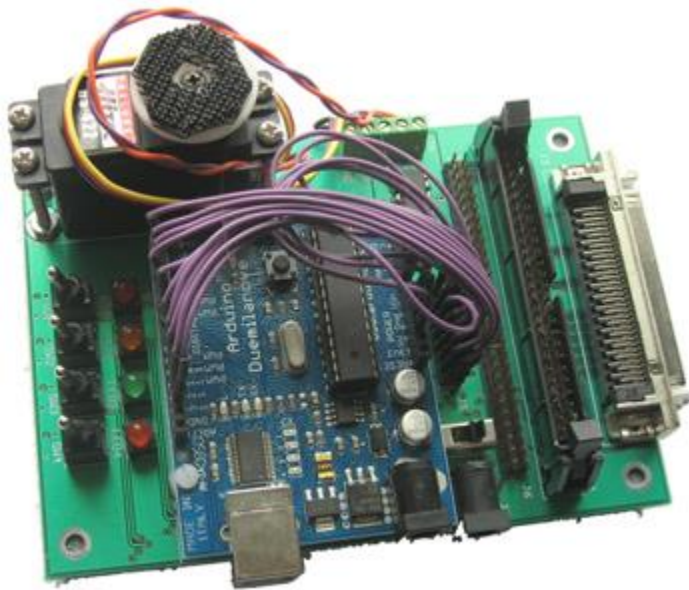


圖 1. 直流馬達連接到一塊 Arduino Uno 板。

我們對這個直流馬達下了追蹤一個方波(square wave)的參考角度的指令，並在 20 秒之後加入一個干擾來讓馬達停止追蹤這個參考角度(圖 2)。

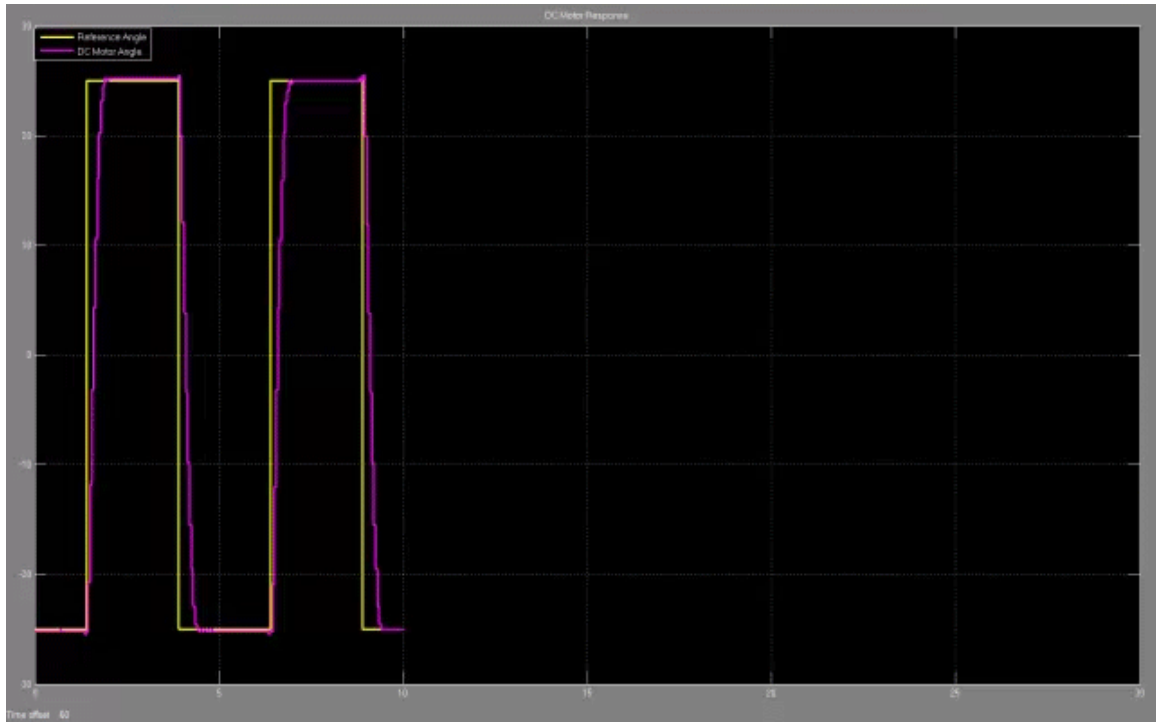


圖 2. 參考角度追蹤故障。

我們想要能夠偵測直流馬達運行時的馬達行為變化。為此，我們將設計一個 Arduino Uno 板程式來執行線上估測。

線上故障偵測

為了設計這個 Arduino Uno 的程式，我們在 Simulink[®] 利用系統辨識工具箱(System Identification Toolbox™)內的模塊組建立一個簡單的兩段模型(圖 3)。

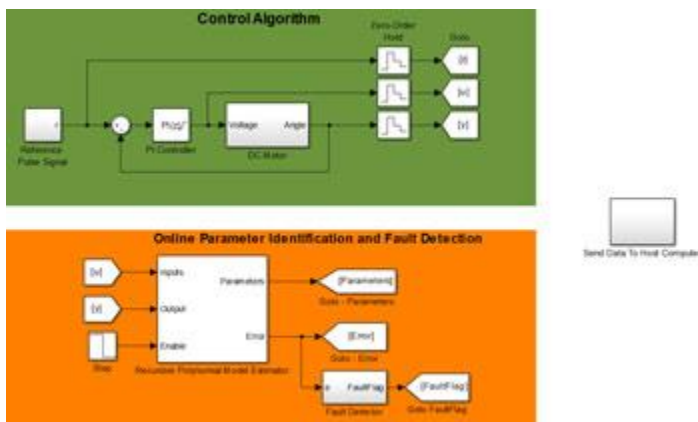


圖 3. 直流馬達控制以及線上參數辨識的 Simulink 模型。

上：控制演算法。下：線上參數辨識及錯誤偵測演算法。

模型的控制演算法使用馬達角度測量以及一個簡單的 PID 控制器來將估算電壓對馬達下達要求，進而用來追蹤角位基準。

下半部的模型是執行線上參數辨識及錯誤偵測。這個模型是以系統辨識工具箱(System Identification Toolbox)中的遞迴多項式估測器(Recursive Polynomial Estimator)模塊(函式庫)來建構。

我們提供 DC 馬達的電壓指令及馬達角度量測值作為模塊的輸入值。當我們利用一個 ARMAX 模型結構獨立地建立雜訊及動態模型後，則開始配置這個模塊用以估測這個 DC 馬達的 ARMAX 多項式模型。

該如何選擇正確的參數值，部份來自於科學、其餘則來自於多次的試驗(trial)與錯誤。因為 DC 馬達模型通常是以二階系統建立，我們選擇 2 作為極數(A(q) terms)，並分別選擇 2 及 1 作為 B(q)及 C(q)。我們藉由觀察 DC 馬達一個步階(step)響應輸入所需要花費的時間，並將花費時間除以估測器的取樣時間來得到輸入延遲的值。結果顯示輸入延遲(nk)值為 2(圖 4)。

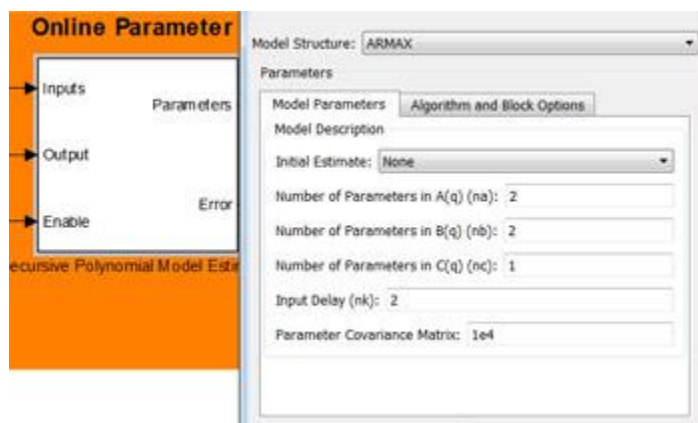


圖 4. 估測模型的參數。

遞迴多項式估測器模塊提供了啟用或停用等兩個參數估測選項(圖 5)。

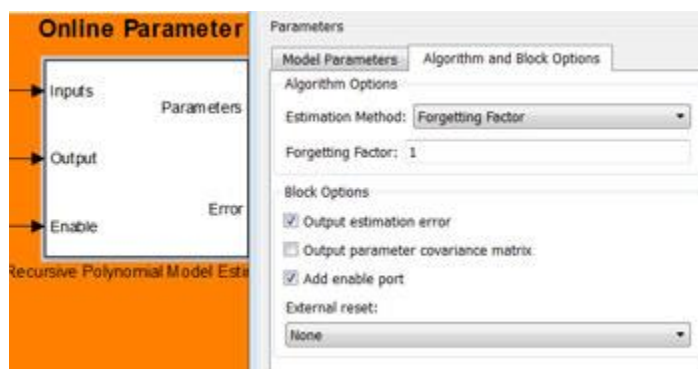


圖 5. 參數估測的遞迴多項式估測器模塊之啟用。

我們將使用啟用埠來執行這項首次 10 秒鐘實驗的線上參數估測-時間足以讓估測的參數涵蓋其穩態值。10 秒之後，我們停止估測。當估測被停用，模塊並不會更新 DC 馬達的參數，而是會輸出其先前估測的數值。我們可以在馬達正常操作條件下的指令電壓已知的情況下，利用這些估測的參數值來預測馬達角度，並將預測值與量測的馬達角度做比較。

為了要比較預測及實際量測的馬達角度，我們啟用錯誤埠，這可以輸出上一步的預測錯誤值(模塊中馬達角度量測值與估測值的差異)。

我們把稍早一步預測錯誤的低通濾波版本傳送到故障偵測演算法，以一個二態 Stateflow[®](事件導向系統模擬軟體)圖表執行(圖 6)。當濾波器錯誤大於這個臨界值時，Stateflow[®]圖表會把故障信號設定為 1；經過 10 秒之後如果錯誤值小於 1，則重設故障信號為 0。

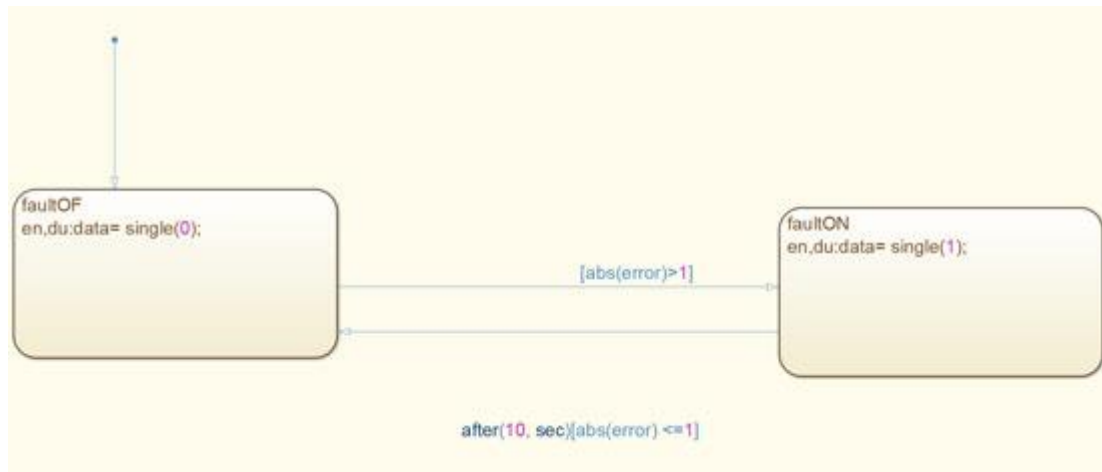


圖 6. 在 Stateflow 執行故障偵測演算法。

接著，我們使用 Simulink 裡面的 Run on Target Hardware 功能，把這個 Simulink 模型轉檔佈署到 Arduino Uno 板。

結果

現在，我們已經完成偵測 DC 馬達運作時動態行為變化的設置了(圖 7)。

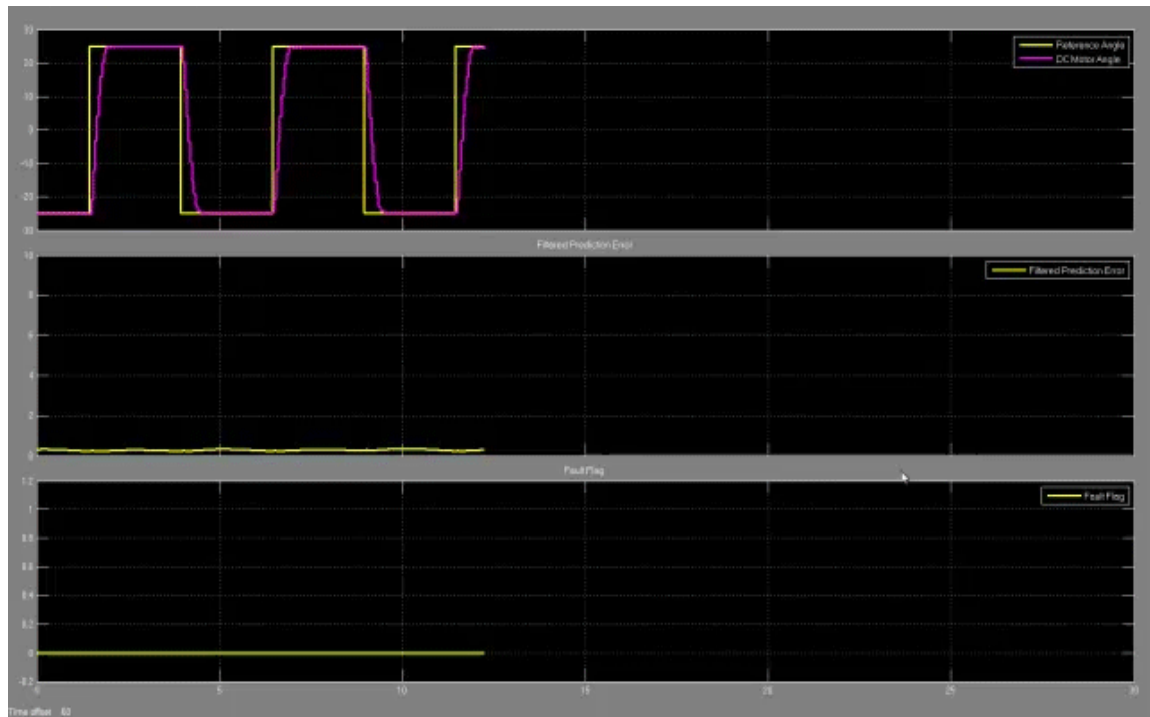


圖 7. 即時故障偵測。

我們讓線上參數估測演算法在前 10 秒去估測這個指定 DC 馬達的動態。經過 20 秒左右的模擬，我們在 DC 馬達放入一個干擾。當干擾一發生，這個預測錯誤迅速顯現，而演算法將故障信號設定為 1。當干擾被移除，演算法即重設故障信號為 0。

方法的延伸

這個範例中，我們使用線上估測使得物理系統的在故障發生時立即可被偵測到。線上估測常見的應用也包含利用估測受控體模型，依據受控體模型的變化來修改控制器的適應性控制；或者可用於軟感應，也就是當你為了估測受控體模型的回饋控制故障偵測或而產生“量測值”的應用。

藉由模擬線上估測進行有效性檢測後，接下來您就可以為模型產生程式碼，轉檔佈署到目標的硬體之上。