

2D MIMO Kalman filter (Project Proposal)

Team Member : 108061272 林孟平 108061271 謝明翰
108060021 許庭崴

Objective

卡爾曼濾波器又稱為最佳線性濾波器，會根據目標變數在不同時間下的觀測值，考慮不同的機率分布，對此變數之未知狀況進行估計。除了時常被應用在飛機、太空梭的導航和控制外，也備廣泛地應用於GPS定位、影像去噪、或是時間序列的計量經濟、訊號處理等領域。在影像相關的應用中，我們會使用卡爾曼濾波器對大量的物體(pixels)進行預測，而其中會包含2D、3D或是更高維度的變數(訊號)，所以計算上往往會耗費大量的CPU資源。因此，我們希望能透過硬體的方式解決濾波器中大量的矩陣運算，並且透過數位電路強大的平行運算能力，提高計算變數數量之Throughput，目標建構出多維度、一次可處理多時間訊號之MIMO卡爾曼濾波器系統，且在最後透過真實存在之時序訊號去驗證濾波器硬體實現之可能性。

Functionality

以預測物體的運動狀態為例，我們主要需要控制位置、速度和noise(觀測產生的不確定性)等三個參數，因此在卡爾曼濾波器中，需要包含以下核心功能:

- (1) 利用前一時刻之運動狀態預測下一時刻之運動狀態，為預測值。
- (2) 利用前一時刻之noise預測下一時刻之noise，為預測值。
- (3) 由noise的觀測值(濾波器輸入)和預測值，計算出卡爾曼係數
- (4) 將卡爾曼係數作用於運動狀態觀測值(濾波器輸入)和預測值，計算出下一時刻修正後的運動狀態(濾波器輸出)
- (5) 將卡爾曼係數作用於noise觀測值和預測值，計算出下一時刻修正後的noise

以上的操作主要包含矩陣乘加法、矩陣轉置以及反矩陣的求取。除此之外，由於牽扯到整數與小數的運算，因此在整個系統中我們預計使用16 bits 2s complement Fixed point的方法完成。

除此之外，為了提高電路的運算能力，我們預計使用MIMO的平行計算架構，希望能同時處理高達4個時間，2個相關維度的濾波器輸入與輸出。

Specification

以能對1080p、60fps之動態影像中每個物體(pixel)預測為基準，定義以下spec:

1. Throughput: $1920 * 1080 * 60$ pixels/s @ 250MHz
2. Area: 450,000 μm^2
3. Power: 120 mW @ 250MHz

Implementation

1. 我們將先用Python實現卡爾曼濾波器的軟體演算法，且此演算法必須考慮到硬體實現的方便性。
2. 完成軟體演算法後，將根據此演算法先行定義好各個module的 I/O ports，並完成RTL硬體實現。
3. Verification將與RTL同步進行，將根據軟體演算法及Module I/O ports，完成testbench，並對完成的RTL進行驗證。
4. 將硬體功能正確性驗證完畢後，將進行RTL synthesis，並根據先前設定之Spec，進行電路performance的優化，最主要的目標在於減少總計算cycle數和縮短critical path，以提高濾波器的效能。
5. Performance 達一定程度後，將進行Placement and Routing，期望能在不更改Spec的狀況下完成後端佈局工作。
6. 若不幸地無法完成P&R，則降低Spec，或是回到RTL設計階段進行修改，直到順利完成所有設計流程。

Verification

在這個project當中，我們將使用離散時間序列的訊號對卡爾曼濾波器進行驗證。在不同的時間對於濾波器輸入不同的觀測值，並比較硬體與軟體計算出來的結果，我們可以檢驗電路的運算方式是否符合軟體上的正確性。

另外，我們也可以根據電路合成以及繞線布局後的結果，觀察電路的計算速度 (throughput)、面積與功耗是否達到先前定義出之spec。

Reference

- [1] Kalman filter Tutorial <https://www.kalmanfilter.net/alphabeta.html>

- [2] Use of a Kalman filter to improve real-time video stream image processing
https://www.cs.cmu.edu/~motionplanning/papers/sbp_papers/kalman/video_kalman.pdf

- [3] Kalman filter – Machine Learning TV
https://www.youtube.com/watch?v=LioOvUZ1MiM&ab_channel=MachineLearningTV