# FIR Filter Design

Lecture and notes by Prof. Brian L. Evans (UT Austin)
Scribe: Magesh Valliappan (UT Austin)
Based on notes from Prof. Russell Mersereau (Georgia Tech)

October 5, 2000

## 1 Review

An FIR or non-recursive filter has an impulse response with finite support.

$$h(n_1, n_2) = 0, \quad \text{unless} \quad \begin{array}{l} 0 \leq n_1 \leq N_1 - 1 \\ 0 \leq n_2 \leq N_2 - 1 \end{array}$$

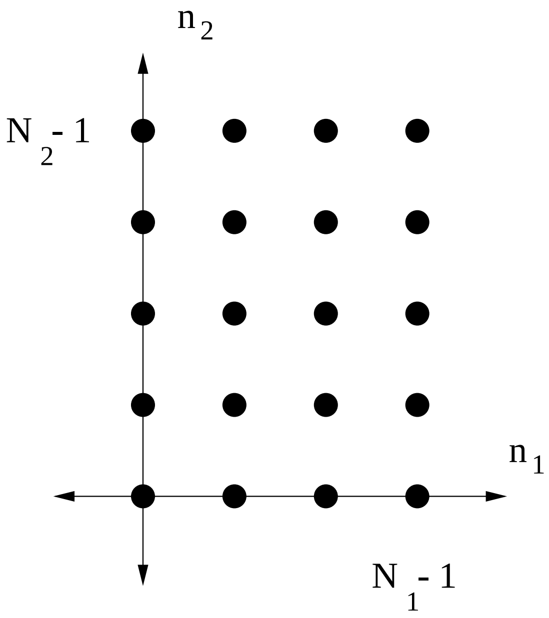
$$y(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} h(k_1, k_2) x(n_1 - k_1, n_2 - k_2)$$

## 2 Frequency Response

$$H(\omega_1, \omega_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} h(k_1, k_2) e^{-j\omega_1 k_1 - j\omega_2 k_2}$$

The frequency response is a 2-D polynomial in $e^{-jw_1}$ and $e^{-jw_2}$.

These filters can be implemented directly using the convolution sum or they can be implemented using the DFT.




# 3 Zero-Phase Filters

A filter has a zero-phase response if its frequency response is real-valued

$$H(\omega_1, \omega_2) = H^*(\omega_1, \omega_2)$$

A zero-phase filter must have an odd-number of samples in its support with the origin at the center.

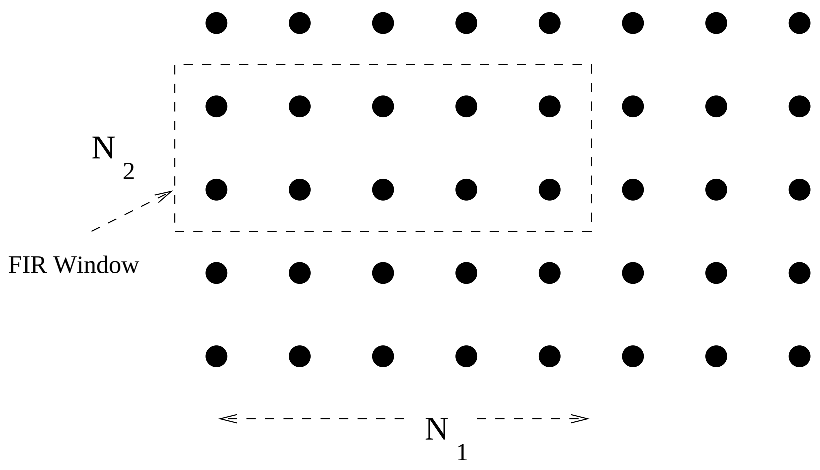
If the filter has real coefficients, then the impulse response must be symmetric about the origin, i.e. $h(n) = h(-n)$. If the filter has imaginary coefficients, then the impulse response must be anti-symmetric about the origin, i.e. $h(n) = -h(-n)$.

# 4 Direct Implementation of FIR Filters

An $N_1 \times N_2$ point filter requires

- $N_1 N_2$ multiplies per output sample
- $N_1 N_2 - 1$ additions per output sample

- $N_2$ Rows of storage for a row-by-row implementation




We can compute the outputs in any order we desire— even in parallel.

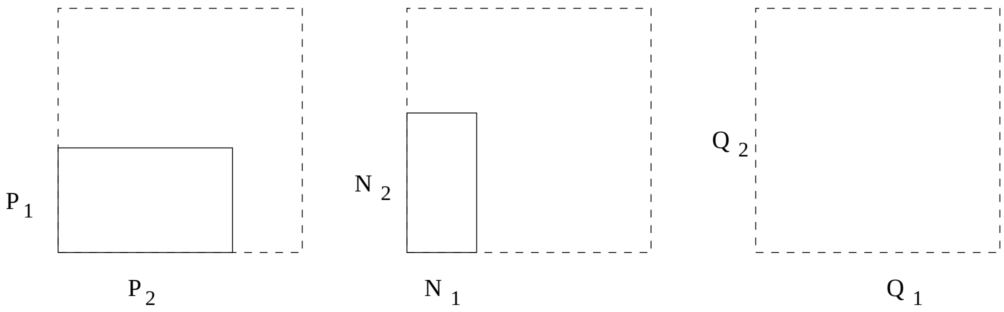
Normal choices are row-by-row and column-by-column.

MMX Technology: 64-bit data registers can be broken into 4 segments of 16 bits or 8 segments of 8 bits. Multiplication can be applied to four 16-bit segments at the same time, whereas addition can be applied to eight 8-bit segments at the same time. MMX supports saturating arithmetic (results "rail out" instead of wrap around as would be the normal case for two's complement arithmetic). 4:1 parallelism for multiplication and 8:1 parallelism for addition.

# 5 DFT Implementation of FIR filters

- $y(n_1, n_2) = x(n_1, n_2) * *h(n_1, n_2)$
- $\hat{y}(n_1, n_2) = DFT^{-1}\left\{X(k_1, k_2)H(k_1, k_2)\right\}$
- $\hat{y}(n_1, n_2)$ is the circular convolution of $x$ and $h$. It is a spatially aliased version of $y(n_1, n_2)$.
- If the DFT size is large enough to contain $y(n_1, n_2)$, then

$$y(n_1, n_2) = \hat{y}(n_1, n_2)$$




$$Q_1 \geq P_1 + N_1 - 1$$
$$Q_2 \geq P_2 + N_2 - 1$$

- Computation:

$$\frac{3\frac{Q_1Q_2}{2}\log_2 Q_1Q_2 + Q_1Q_2}{Q_1Q_2} \quad \text{complex mults/output sample}$$

- The FFT implementation can require considerably less computation, but more storage and I/O than a Direct Implementation.
- Block Convolution implementations are also possible.

# 6 2-D Filter Design using Windows

## 6.1 1-D Procedure (Review)

- Set $h(n) = i(n)\,w(n)$
- Then $H(\omega) = \dfrac{1}{2\pi}\displaystyle\int_{-\pi}^{\pi} I(\alpha)W(\omega-\alpha)d\alpha$
- The Window $w(n)$ should be chosen
  - To be $N$ points long
  - $W(\omega) \approx \delta(\omega)$

- $w(n)$ should be symmetric if $h(n)$ is to be linear phase.

$$w(n) = w(N-1-n)$$

## 6.2 The 2-D Procedure

- Set $h(n_1, n_2) = i(n_1, n_2)w(n_1, n_2)$
-

$$H(\omega) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} I(\alpha_1 \alpha_2) W(\omega_1 - \alpha_1, \omega_2 - \alpha_2) d\alpha_1 d\alpha_2$$

- The window $w(n_1, n_2)$ should be chosen
  - To be $N_1 \times N_2$ points in extent
  - $W(\omega_1, \omega_2) \approx \delta(\omega_1, \omega_2)$
  - $w(n_1, n_2)$ should be symmetric if the filter is to be linear phase.

$$w(n_1, n_2) = w(N_1 - 1 - n_1, N_2 - 1 - n_2)$$

Choosing the Window

- $w(n_1, n_2) = w(n_1)\, w(n_2)$
  - outer product window
  - $W(\omega_1, \omega_2) = W_1(\omega_1)W_2(\omega_2)$, where $W_1$ and $W_2$ are "good" 1-D Windows.
  - Rectangular region of support
  - Main lobe shape and side lobe heights can be calculated using 1-D results.
- $w(n_1, n_2) = w_a(\sqrt{n_1^2 + n_2^2})$, where $w_a(\cdot)$ is a "good" 1-D continuous window function.

Recall from Example 6 in Chapter 1 that the discrete-time impulse response of an ideal circularly bandlimited signal of radius $R$ is

$$i(n_1, n_2) = \frac{R}{2\pi} \frac{J_1(R\sqrt{n_1^2 + n_2^2})}{\sqrt{n_1^2 + n_2^2}}$$

This ideal impulse response has two-sided infinite extent in each dimension. A window-based FIR filter design would be to apply a window to this ideal response.

For a comprehensive listing of windowing functions, see

- F. J. Harris, "On the Use of Windows for Harmonic Analysis with the DFT," *Proc. of the IEEE*, pp. 51–83, Jan. 1978,

## 6.3 Performance of windows as filters

1-D FIR Filter Design

- Parks McClellan (Remez exchange) gives the shortest linear phase FIR filter for a given piece-wise constant magnitude response specification.
- Windows have closed-form solutions but are much longer ($\approx \times 2$ for Kaiser windows)