FIR Filter Design Using Transformations

• Objectives
  ➢ Reduces M-D FIR design to 1-D FIR design
  ➢ Efficient realization of designed filters

• Motivation
  ➢ M-D designs based on $L_\infty$ (Chebyshev) norm are difficult
  ➢ 1-D designs very well studied and understood

• Strategy (M = 2; similar for any M-D design)
  ➢ Given ideal 2-D specifications $I(\omega_1, \omega_2)$
    ✓ Transform 2-D specs into appropriate 1-D specs by choosing appropriate 1-D filter $I(\omega)$ → 1-D filter called prototype filter.
    ✓ Design 1-D FIR filter $H(\omega)$ approximating $I(\omega)$ using known 1-D techniques
    ✓ Transform the designed 1-D prototype FIR filter $H(\omega)$ into a 2-D FIR filter $H(\omega_1, \omega_2)$ by using an appropriate 2-D mapping function

1-D FIR prototype $\xrightarrow{F(\omega_1, \omega_2)}$ mapping $\xrightarrow{H(\omega_1, \omega_2)}$ Resulting designed 2-D FIR filter

 ✓ Want to choose 1-D prototype $I(\omega)$ and 2-D mapping function $F(\omega_1, \omega_2)$ such that $H(\omega_1, \omega_2) \sim I(\omega_1, \omega_2)$. 
FIR Filter Design Using Transformations

• Two main issues
  ➢ How can we transform a 1-D FIR filter into a 2-D FIR filter?
  ➢ How do we choose the 1-D filter (prototype filter) and mapping function $F(\omega_1, \omega_2)$?
FIR Filter Design Using Transformations

- How to transform 1-D FIR filter into a 2-D FIR filter?
  - Only symmetric 1-D filters can be transformed by the proposed McClellan Transformation method.
  - Background:
    - McClellan transformation method maps 1-D positive-symmetric FIR filters into 2-D (positive-symmetric) FIR filters.
    - Consider a 1-D zero-phase symmetric filter of length 2N+1
      \[
      H(\omega) = \sum_{n=-N}^{N} h(n) e^{-j\omega n} = h(0) + \sum_{n=-N}^{-1} h(n) e^{-j\omega n} + \sum_{n=1}^{N} h(n) e^{-j\omega n}
      \]
      \[
      = h(0) + \sum_{n=1}^{N} h(n) [e^{j\omega n} + e^{-j\omega n}]
      \]
      \[
      = \sum_{n=0}^{N} a(n) \cos \omega n
      \]
      where
      \[
      a(n) = \begin{cases} 
      h(0), & n = 0 \\
      2h(n), & 1 \leq n \leq N
      \end{cases}
      \]
FIR Filter Design Using Transformations

- How to transform 1-D FIR filter into a 2-D FIR filter? (continued)
  - Background: Chebyshev Polynomials
    - It is well-known that $\cos n\omega$ can be expressed as a polynomial of degree $n$ in the variable $\cos \omega$
    
    $$
    \cos n\omega = T_n(\cos \omega)
    $$
    
    where
    
    $$
    T_n(x) = n^{th} \text{ Chebyshev polynomial}
    $$
    
    $$
    T_0(x) = 1
    $$
    
    $$
    T_1(x) = x
    $$
    
    $$
    T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad -1 \leq x \leq 1
    $$
    
    - Thus:
      
      $$
      H(\omega) = \sum_{n=0}^{N} a(n) \cos \omega n
      $$
      
      $$
      \Rightarrow H(\omega) = \sum_{n=0}^{N} a(n) T_n(\cos \omega)
      $$
FIR Filter Design Using Transformations

- How to transform 1-D FIR filter into a 2-D FIR filter? (continued)
  - Background: Derivation of the Chebyshev Recursion
    \[
    \cos A \cos B = \frac{1}{2} \cos(A + B) + \frac{1}{2} \cos(A - B)
    \]
    ✓ Let \( A = \omega \) and \( B = (n-1)\omega \), then
    \[
    2 \cos \omega \cos(n-1)\omega = \cos n\omega + \cos(n-2)\omega
    \]
    or
    \[
    \cos n\omega = 2 \cos \omega \cos(n-1)\omega + \cos(n-2)\omega
    \]
    \[
    T_n (\cos \omega) = 2 \cos \omega T_{n-1} (\cos \omega) - T_{n-2} (\cos \omega)
    \]
    \[
    T_n (x) = 2x T_{n-1} (x) - T_{n-2} (x)
    \]
FIR Filter Design Using Transformations

- How to transform 1-D FIR filter into a 2-D FIR filter? (continued)
  
  The Transformation:
  
  ✓ McClellan suggested that we can obtain a 2-D zero-phase FIR filter if we make the substitution

  \[
  \cos \omega \quad \text{replace by} \quad F(\omega_1, \omega_2)
  \]

  Then

  \[
  H(\omega) = \sum_{n=0}^{N} a(n) \cos \omega n
  \]

  \[
  H(\omega) = \sum_{n=0}^{N} a(n) T_n (\cos \omega)
  \]

  \[
  \Rightarrow H(\omega_1, \omega_2) = \sum_{n=0}^{N} a(n) T_n [F(\omega_1, \omega_2)]
  \]
FIR Filter Design Using Transformations

• How to transform 1-D FIR filter into a 2-D FIR filter? (continued)
  ➢ The Transformation:

\[
H(\omega) = \sum_{n=0}^{N} a(n)T_n(\cos \omega)
\]

\[
H(\omega_1, \omega_2) = \sum_{n=0}^{N} a(n)T_n[F(\omega_1, \omega_2)]
\]

✓ The mapping function \(F(\omega_1, \omega_2)\) should be real (zero-phase) and

\[|F(\omega_1, \omega_2)| \leq 1 \quad \text{or} \quad -1 \leq F(\omega_1, \omega_2) \leq 1\]

since \(F(\omega_1, \omega_2)\) has to replace \(\cos(\omega)\) and has to take the same values of \(\cos(\omega)\) since argument of Chebyshev polynomial

\(\Rightarrow H(\omega_1, \omega_2)\) takes same exact values of designed 1-D FIR prototype filter \(H(\omega)\).

Consequence:
- values of \(H(\omega_1, \omega_2)\) are determined by values of \(H(\omega)\).
- \(H(\omega_1, \omega_2)\) only takes values of \(H(\omega)\).
FIR Filter Design Using Transformations

• How to transform 1-D FIR filter into a 2-D FIR filter? (continued)

  The Transformation:
  ✓ The mapping function $F(\omega_1, \omega_2)$ should be chosen to be the frequency response of a zero-phase FIR filter
  ✓ If $F(\omega_1, \omega_2)$ is real and $a(n)$ real ⇒ $H(\omega_1, \omega_2)$ is real ⇒ zero-phase
  ✓ If $F(\omega_1, \omega_2)$ is $(2Q+1) \times (2Q+1)$ ⇒ $H(\omega_1, \omega_2)$ is $(2NQ+1) \times (2NQ+1)$ because $H(\omega_1, \omega_2)$ is a polynomial of degree $N$ in $F(\omega_1, \omega_2)$ ⇒ same size as $F^N$

$$f \stackrel{FT}{\leftrightarrow} F$$

$$f \ast f \leftrightarrow F^2$$

$$\vdots$$

$$f \ast f \ldots \ast f \leftrightarrow F^N$$ \hspace{1cm} (2NQ + 1)(2NQ + 1)

$$g \leftrightarrow T_n(F)$$ \hspace{1cm} (2nQ + 1)(2nQ + 1)

$$h \leftrightarrow H(\omega_1, \omega_2)$$ \hspace{1cm} (2NQ + 1)(2NQ + 1)

✓ Size of support of $h(n_1, n_2)$ depends on $Q$ (order of transformation subfilter $F$) and on $N$ (order of 1-D prototype filter)
FIR Filter Design Using Transformations

• How to transform 1-D FIR filter into a 2-D FIR filter? (continued)
  ➢ The Transformation:

\[
H(\omega) = \sum_{n=0}^{N} a(n)T_n(\cos \omega)
\]

\[
H(\omega_1, \omega_2) = \sum_{n=0}^{N} a(n)T_n[F(\omega_1, \omega_2)]
\]

✓ The mapping function \(F(\omega_1, \omega_2)\) determines the symmetries of the designed 2D FIR filter \(H(\omega_1, \omega_2)\). \(\Rightarrow\) \(H(\omega_1, \omega_2)\) have the same symmetries of \(F(\omega_1, \omega_2)\).

✓ Example: For zero-phase, even-symmetric \(H(\omega_1, \omega_2), F(\omega_1, \omega_2)\) should be zero-phase and even-symmetric of the form:

\[
F(\omega_1, \omega_2) = \sum_{q=0}^{Q} \sum_{r=0}^{R} t_{qr} \cos(q\omega_1) \cos(r\omega_2)
\]

\[
+ \sum_{i=0}^{I} \sum_{j=0}^{J} s_{ij} \sin(i\omega_1) \sin(j\omega_2)
\]

Note: Order of \(F(\omega_1, \omega_2) = \max(Q,R,I,J)\).
FIR Filter Design Using Transformations

• How to transform 1-D FIR filter into a 2-D FIR filter? (continued)

  ➢ The Transformation:

  \[
  H(\omega) = \sum_{n=0}^{N} a(n)T_n(\cos \omega) \\
  H(\omega_1,\omega_2) = \sum_{n=0}^{N} a(n)T_n[F(\omega_1,\omega_2)]
  \]

  ✓ The mapping function \(F(\omega_1,\omega_2)\) determines the shape of the contours of the designed 2D FIR filter \(H(\omega_1,\omega_2)\) => Contours of \(H(\omega_1,\omega_2)\) have the same shape as the contours of \(F(\omega_1,\omega_2)\).

  ✓ Let \(F(\omega_1,\omega_2) = C, \quad |C| \leq 1\)

  \[
  \Rightarrow H(\omega_1,\omega_2) = \sum_{n=0}^{N} a(n)T_n[C] = C'
  \]

  ✓ Consequence:
  
  – Shape of contours of \(H(\omega_1,\omega_2)\) determined by \(F(\omega_1,\omega_2)\).
  
  – Value \(C'\) determined by \(\{a(n)\}, \quad n = 1 \text{ to } N\), which depends on the designed 1-D prototype \(H(\omega)\) => prototype \(H(\omega)\) determines the values of \(H(\omega_1,\omega_2)\) along contours.
FIR Filter Design Using Transformations

• Flexible and Modular:
  - Change $F(\omega_1, \omega_2)$ to change contour shape
  - Change $\{a(n)\}$ (prototype) to change values on contours

• Procedure:
  1) Design transformation subfilter $F(\omega_1, \omega_2)$:
      a. to get desired symmetries
      b. to produce or approximate desired contours
      c. OR, if prototype filter already specified, to map values of prototype to desired 2-D locations or contours
  2) Design prototype to get desired values along contours
FIR Filter Design Using Transformations

Example:
Consider
\[ F(\omega_1, \omega_2) = A + B \cos(\omega_1) + C \cos(\omega_2) + D \cos(\omega_1 - \omega_2) + E \cos(\omega_1 + \zeta) \]

Let \( A=-B=-C=-1/2, \) \( D=E=1/4 \) (proposed by McClellan)

\[ F(\omega_1, \omega_2) = -\frac{1}{2} + \frac{1}{2} \cos(\omega_1) + \frac{1}{2} \cos(\omega_2) + \frac{1}{2} \cos \omega_1 \cos \omega_2 \]

\[
\begin{array}{c c c c c c c}
1 & 1 & 1 & & & & \\
8 & 4 & 8 & & & & \\
1 & & -1 & 1 & & & \\
4 & 2 & 4 & & & & \\
1 & 1 & 1 & & & & \\
8 & 4 & 8 & & & & \\
\end{array}
\]

\[ f(n_1, n_2) \]

\[ F(\omega_1, \omega_2) = F(-\omega_1, -\omega_2) = F(-\omega_1, \omega_2) = F(\omega_1, -\omega_2) \text{ (quadrant symmetry)} \]

\[ \Rightarrow \text{designed } H(\omega_1, \omega_2) \text{ has same symmetry} \]
FIR Filter Design Using Transformations

• **Example:** (continued)

\[ F(\omega_1, \omega_2) = -\frac{1}{2} + \frac{1}{2}\cos(\omega_1) + \frac{1}{2}\cos(\omega_2) + \frac{1}{2}\cos \omega_1 \cos \omega_2 \]

Contours of \(F(\omega_1, \omega_2)\):

\[ \text{Tend to be rectangular} \]

\[ \text{Circular around origin} \]

**Note:**

1) \( F(\omega_1, 0) = \cos \omega_1 \Rightarrow H(\omega_1, 0) = \sum_{n=0}^{N} a(n)T_n(\cos \omega_1) = H(\omega_1) \)

=> Prototype exactly mapped along \( \omega_1 = 0 \) axis

2) \( F(0, \omega_2) = \cos \omega_2 \Rightarrow H(0, \omega_2) = \sum_{n=0}^{N} a(n)T_n(\cos \omega_2) = H(\omega_2) \)

=> Prototype exactly mapped along \( \omega_2 = 0 \) axis
FIR Filter Design Using Transformations

- To get lowpass 2-D filter, use

\[ H(\omega) \]
\[ \omega_c \quad \pi \]

- To get bandpass 2-D filter, use

\[ H(\omega) \]
\[ \omega_L \quad \omega_H \quad \pi \]
FIR Filter Design Using Transformations

Mapping $F(\omega_1, \omega_2)$

Prototype Filter $H(\omega)$

Designed 2D Filter $H(\omega_1, \omega_2)$
FIR Filter Design Using Transformations

Mapping $F(\omega_1, \omega_2)$

Prototype Filter $H(\omega)$

Designed 2D Filter $H(\omega_1, \omega_2)$
FIR Filter Design Using Transformations

Desired $I(\omega_1, \omega_2)$

Mapping $F(\omega_1, \omega_2)$

Magnitude of frequency response

Prototype Filter $H(\omega)$

Designed 2D Filter $H(\omega_1, \omega_2)$
FIR Filter Design Using Transformations

- **Design Example:** We want to design a Fan filter using transformation

\[ F(\omega_1, \omega_2) \text{ function} \Rightarrow Q=R=1 \]

2) Use a lowpass 1-D prototype

Requirement: 1) Use a 1st order transformation \( F(\omega_1, \omega_2) \) function => Q=R=1

2) Use a lowpass 1-D prototype
FIR Filter Design Using Transformations

- **Example:** Design of a Fan Filter (continued)

Choice of Transformation Function

- \( I(\omega_1, \omega_2) \) quadrant symmetric => \( F(\omega_1, \omega_2) \) needs to be quadrant symmetric

so, \( F(\omega_1, \omega_2) \) of the form:

\[
F(\omega_1, \omega_2) = \sum_{q=0}^{Q} \sum_{r=0}^{R} t_{qr} \cos(q \omega_1) \cos(r \omega_2)
\]

\[
= t_{00} + t_{10} \cos \omega_1 + t_{01} \cos \omega_2 + t_{11} \cos \omega_1 \cos \omega_2
\]

Note: since quadrant symmetric => need to consider only one quadrant
FIR Filter Design Using Transformations

- **Example:** Design of a Fan Filter (continued)

\[ F(\omega_1, \omega_2) = t_{00} + t_{10} \cos \omega_1 + t_{01} \cos \omega_2 + t_{11} \cos \omega_1 \cos \omega_2 \]

- Determination of parameters:
  - Typically done by:
    1. Exploiting other symmetries
    2. Mapping values of prototype appropriately
FIR Filter Design Using Transformations

• Example: Design of a Fan Filter (continued)

\[
F(\omega_1, \omega_2) = t_{00} + t_{10} \cos \omega_1 + t_{01} \cos \omega_2 + t_{11} \cos \omega_1 \cos \omega_2
\]

1. Exploiting Other Symmetries:

\[
I(\omega_1, \omega_2) \text{ symmetric about line } \omega_2 = -\omega_1 + \pi
\]

=> Choose \( F(\omega_1, \omega_2) \) to be symmetric about same line

\[
F(\omega_1, \omega_2) = F(\pi - \omega_2, \pi - \omega_1)
\]

\[
\Rightarrow t_{00} + t_{10} \cos \omega_1 + t_{01} \cos \omega_2 + t_{11} \cos \omega_1 \cos \omega_2
\]

\[
= t_{00} - t_{10} \cos \omega_2 - t_{01} \cos \omega_1 + t_{11} \cos \omega_2 \cos \omega_1
\]

\[
\Rightarrow \begin{cases}
  t_{01} = -t_{10} \\
  t_{10} = -t_{01}
\end{cases}
\]

\[
\Rightarrow F(\omega_1, \omega_2) = t_{00} + t_{10}(\cos \omega_1 - \cos \omega_2) + t_{11} \cos \omega_1 \cos \omega_2
\]

3 unknowns => need 3 equations to solve for them
FIR Filter Design Using Transformations

- Example: Design of a Fan Filter (continued)

\[ F(\omega_1, \omega_2) = t_{00} + t_{10}(\cos \omega_1 - \cos \omega_2) + t_{11}\cos \omega_1 \cos \omega_2 \]

2. Map values of prototype:

Since used 1-D prototype is lowpass, appropriate mapping is as follows:

<table>
<thead>
<tr>
<th>1-D point ( \omega )</th>
<th>maps into</th>
<th>2-D point ((\omega_1, \omega_2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega = 0 )</td>
<td>( 0, \pi )</td>
<td></td>
</tr>
<tr>
<td>( \omega = \pi )</td>
<td>( \pi, 0 )</td>
<td></td>
</tr>
<tr>
<td>( \omega = \frac{\pi}{2} )</td>
<td>( \frac{\pi}{2}, \frac{\pi}{2} )</td>
<td></td>
</tr>
</tbody>
</table>

Note: To map value at 1-D point \( \omega' \) to a desired 2-D location \((\omega'_1, \omega'_2)\), simply set \( F(\omega'_1, \omega'_2) = \cos(\omega') \).
FIR Filter Design Using Transformations

- **Example:** Design of a Fan Filter (continued)

\[ F(\omega_1, \omega_2) = t_{00} + t_{10} (\cos \omega_1 - \cos \omega_2) + t_{11} \cos \omega_1 \cos \omega_2 \]

**a)** \[ \omega = 0 \rightarrow (\omega_1, \omega_2) = (0, \pi) \]

\[ F(0, \pi) = \cos(0) = 1 \]

\[ \Rightarrow t_{00} + t_{10} (\cos 0 - \cos \pi) + t_{11} \cos 0 \cos \pi = 1 \Rightarrow t_{00} + t_{10} (1 + 1) - t_{11} = 1 \]

\[ \Rightarrow t_{00} + 2t_{10} - t_{11} = 1 \quad (1) \]

**b)** \[ \omega = \pi \rightarrow (\omega_1, \omega_2) = (\pi, 0) \]

\[ F(\pi, 0) = \cos(\pi) = -1 \]

\[ \Rightarrow t_{00} + t_{10} (-1 - 1) + t_{11} \cos \pi \cos 0 = -1 \Rightarrow t_{00} - 2t_{10} - t_{11} = -1 \quad (2) \]

\[ (1) - (2) \Rightarrow 4t_{10} = 2 \Rightarrow t_{10} = 1/2 \]

**c)** \[ \omega = \pi / 2 \rightarrow (\omega_1, \omega_2) = (\pi / 2, \pi / 2) \]

\[ F(\pi / 2, \pi / 2) = \cos \pi / 2 = 0 \]

\[ \Rightarrow t_{00} = 0 \]

\[ \begin{align*}
  t_{00} = 0 \\
  t_{10} = 1/2
\end{align*} \]

\[ \Rightarrow t_{11} = 2t_{10} - 1 = 0 \quad \text{(using (1))} \]

\[ F(\omega_1, \omega_2) = 1/2 \cos \omega_1 - 1/2 \cos \omega_2 \]
FIR Filter Design Using Transformations

**Example:** Design of a Fan Filter (continued)

\[ F(\omega_1, \omega_2) = 1/2 \cos \omega_1 - 1/2 \cos \omega_2 \]

\[(2N + 1) = \text{length of } h(n)\]

\[ F(\omega_1, \omega_2) = 1/2(\cos \omega_1 - \cos \omega_2) \]

\[ H(\omega_1, \omega_2) = h(0) + \sum_{n=1}^{N} 2h(n)T_n(F(\omega_1, \omega_2)) \]

\[ = \sum_{n=0}^{N} a(n)T_n(F(\omega_1, \omega_2)) \]

where \( a(0) = h(0), \ a(n) = 2h(n), \ 1 \leq n \leq N \)
FIR Filter Design Using Transformations

• Review of design procedure

\[ H(\omega) = \sum_{n=0}^{N} a(n)T_n(\cos \omega) \]

\[ H(\omega_1, \omega_2) = \sum_{n=0}^{N} a(n)T_n[F(\omega_1, \omega_2)] \]

\{a(n)\}: from 1-D prototype design

\( F(\omega_1, \omega_2) \): frequency response of low-order FIR zero-phase filter

- If \( h(n) \) is 2N+1 points long and \( f(n_1, n_2) \) is \((2Q+1) \times (2Q+1)\), then \( h(n_1, n_2) \) is \((2NQ+1) \times (2NQ+1)\).
- Contour shape and symmetries depends only on \( F \).
- Amplitude associated with a particular contour depends upon the 1-D prototype
Implementation of Transformed FIR Designs

• Since the filter is FIR, a direct convolution can be used to implement the filter
• A DFT can also be used
• There also exists a special implementation that exploits the design
  ➢ Based on the recursion of Chebyshev polynomials
    \[ T_n[F(\omega_1, \omega_2)] = 2F(\omega_1, \omega_2)T_{n-1}[F(\omega_1, \omega_2)] - T_{n-2}[F(\omega_1, \omega_2)] \]
    \[ T_0[F(\omega_1, \omega_2)] = 1 \]
    \[ T_1[F(\omega_1, \omega_2)] = F(\omega_1, \omega_2) \]
  ➢ Thus, if we have \( T_{n-1}[F] \) and \( T_{n-2}[F] \), we can get \( T_n[F] \), because \( F(\omega_1, \omega_2) \) is an FIR filter.

\[
\begin{array}{c}
T_{n-2}[F] \quad - \\
T_{n-1}[F] \quad 2F(\omega_1, \omega_2) \quad + \\
\hline
\hline
T_n[F]
\end{array}
\]
Implementation of Transformed FIR Designs

- From the Chebyshev polynomials’ recursion, we obtain:

  \[ T_{n-2}[F] \]
  \[ T_{n-1}[F] \rightarrow 2F(\omega_1, \omega_2) \rightarrow T_n[F] \]

- In addition, we can get \( T_{n-1}[F] \) and \( T_{n-2}[F] \) from the polynomial that come before and so on until we get back to \( T_1[F] \) and \( T_0[F] \).
Implementation of Transformed FIR Designs

- Finally, the frequency response $H(\omega_1, \omega_2)$ of the 2D designed FIR filter is just a linear combination of the $T_i[F]$

$$H(\omega_1, \omega_2) = \sum_{n=0}^{N} a(n) T_n[F(\omega_1, \omega_2)]$$

where $a(0) = h(0)$ and $a(n) = 2h(n), 1 \leq n \leq N$.

- Note: This structure will work for filters of any dimensionality.

To realize an M-dimensional filter $H(\omega_1, \omega_2, \ldots, \omega_M)$, we simply substitute an M-dimensional mapping function $F(\omega_1, \omega_2, \ldots, \omega_M)$.
Implementation of Transformed FIR Designs

- Consider the first-order transformation:

\[ F(\omega_1, \omega_2) = A + B \cos \omega_1 + C \cos \omega_2 + D \cos(\omega_1 - \omega_2) + E \cos(\omega_1 + \omega_2) \]

\[ f(n_1, n_2) = \begin{cases} 
A, & n_1 = n_1 = 0 \\
B/2, & n_1 = \pm 1, n_2 = 0 \\
C/2, & n_1 = 0, n_2 = \pm 1 \\
D/2, & n_1 = \pm 1, n_2 = -n_1 \\
E/2, & n_1 = \pm 1, n_2 = n_1 
\end{cases} \]

- Direct implementation of F requires 5 multiplications per output and 8 adds per output

  ➢ **Note:** Multiplications reduced by combining terms
Implementation of Transformed FIR Designs

- Computational Complexity
  - To implement each filter F of size \((2Q+1)x(2Q+1)\) requires in general:
    \[ (2Q + 1)^2 \text{ Multiplications/Output} \]
    \[ (2Q + 1)^2 - 1 \text{ Additions/Output} \]
    \[ (2Q + 1) \text{ Rows of storage} \]
  
  - For the whole filter \(H(\omega_1, \omega_2)\), we have \(N\) filters \(F\):
    \[ N(2Q + 1)^2 + (N + 1) \text{ Mults/Output} \]
    \[ N(2Q + 1)^2 + (N - 1) \text{ Adds/Output} \]