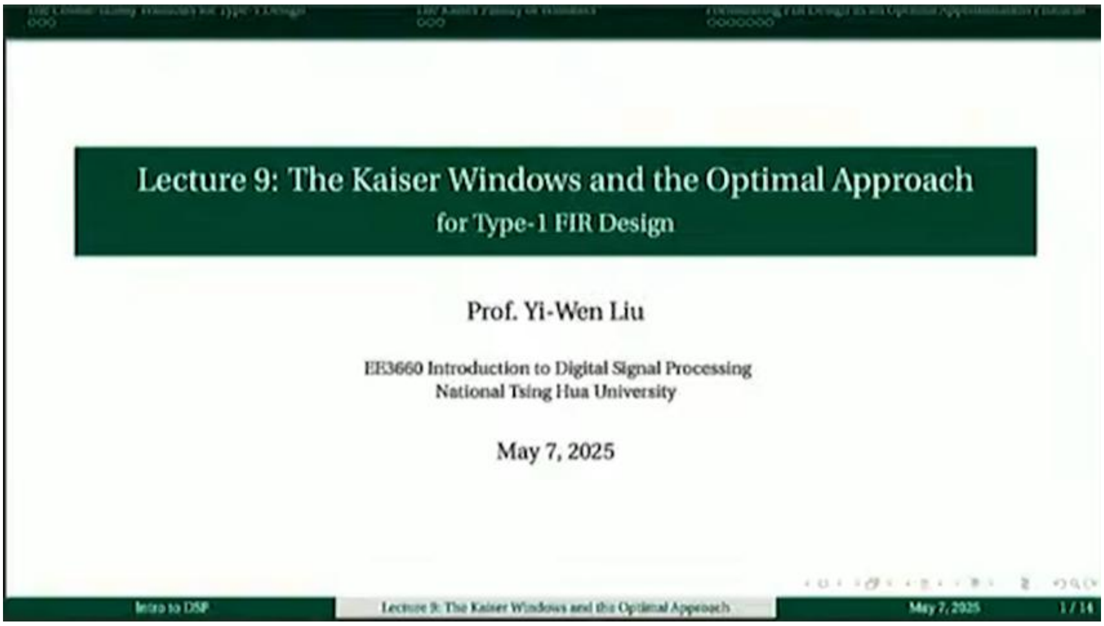
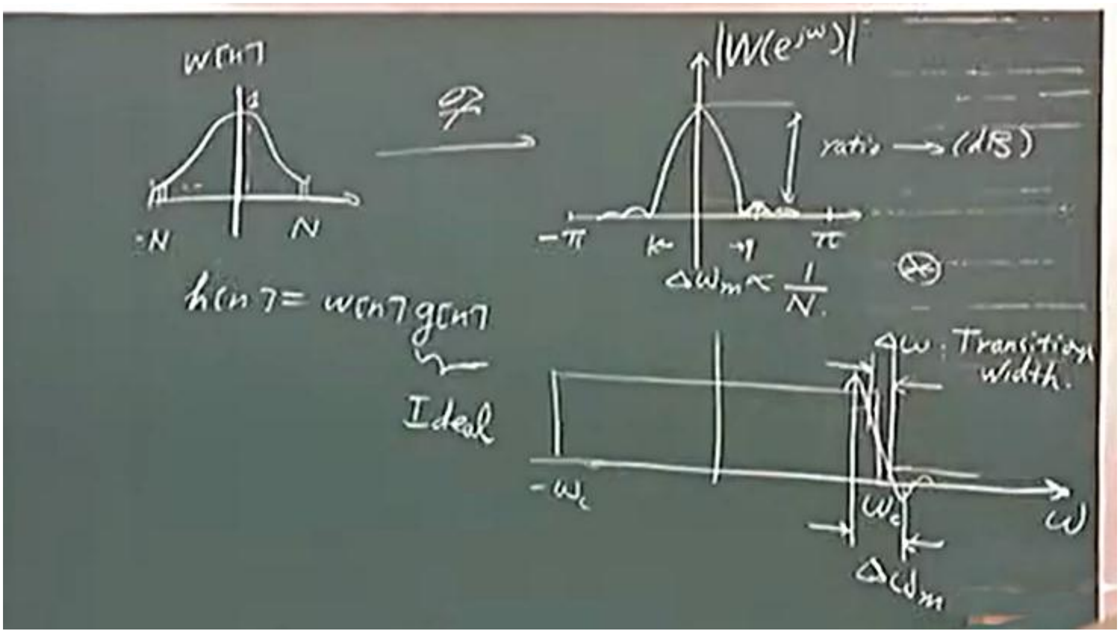
Lecture9 The Kaiser Windows and the Optimal Approach for Type-1 FIR Design





The type I low-pass design

Consider the type I[1] low-pass filter with a cutoff frequency of ω_c. We first shift the ideal response to the right,

$$g[n] = \frac{\sin\omega_c(n-N)}{\pi(n-N)}.$$

Then, we design a filter by $h[n] = w[n]g[n]$, and the window function $w[n]$ is defined in $n = 0:2N$ as follows,

- Rectangular: $w[n] = 1$
- Hann: $w[n] = 0.5 - 0.5\cos(\pi n/N)$
- Hamming: $w[n] = 0.54 - 0.46\cos(\pi n/N)$, and
- Blackmann: $w[n] = 0.42 - 0.5\cos(\pi n/N) + 0.08\cos(2\pi n/N)$.

Also, $w[n] = 0$ elsewhere. These windows are called the cosine family of windows.

[1] Definition of Type I: $h[M-n] = h[n]$ with M being an even number. See Sec. 5.7.3 of O&S.

Rivalry between the cosine and the Kaiser family

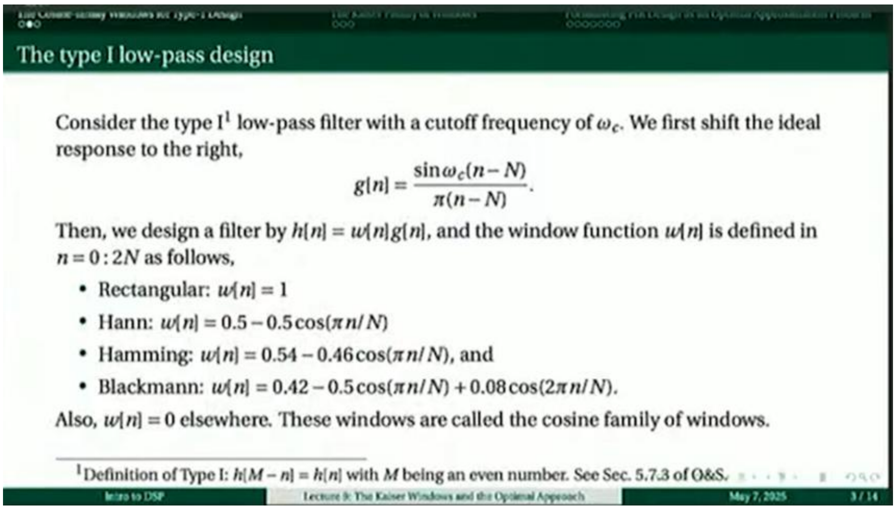
TABLE 7.2 COMPARISON OF COMMONLY USED WINDOWS

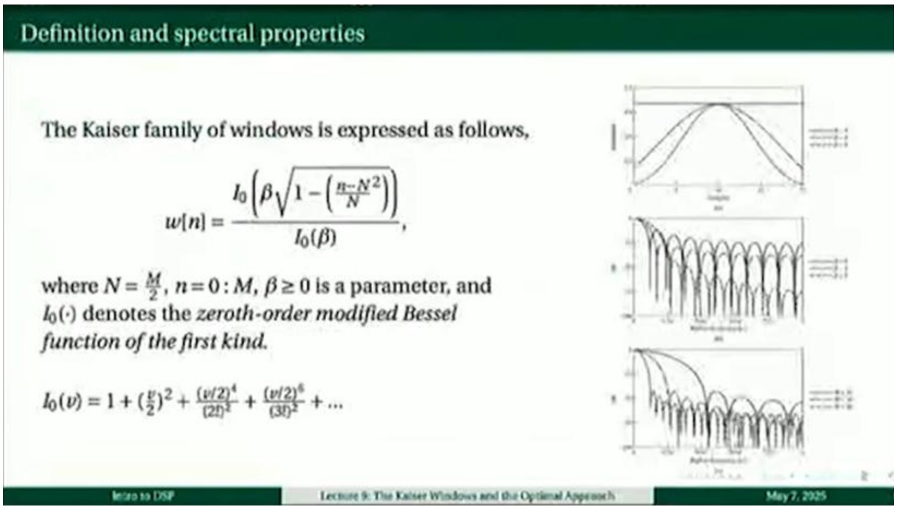
Type of Window	Peak Side-Lobe Amplitude (Relative)	Approximate Width of Main Lobe	Peak Approximation Error, $20\log_{10}\delta$ (dB)	Equivalent Kaiser Window, β	Transition Width of Equivalent Kaiser Window
Rectangular	−13	$4\pi/(M+1)$	−21	0	$1.81\pi/M$
Bartlett	−25	$8\pi/M$	−25	1.33	$2.37\pi/M$
Hann	−31	$8\pi/M$	−44	3.86	$5.01\pi/M$
Hamming	−41	$8\pi/M$	−53	4.86	$6.27\pi/M$
Blackman	−57	$12\pi/M$	−74	7.04	$9.19\pi/M$

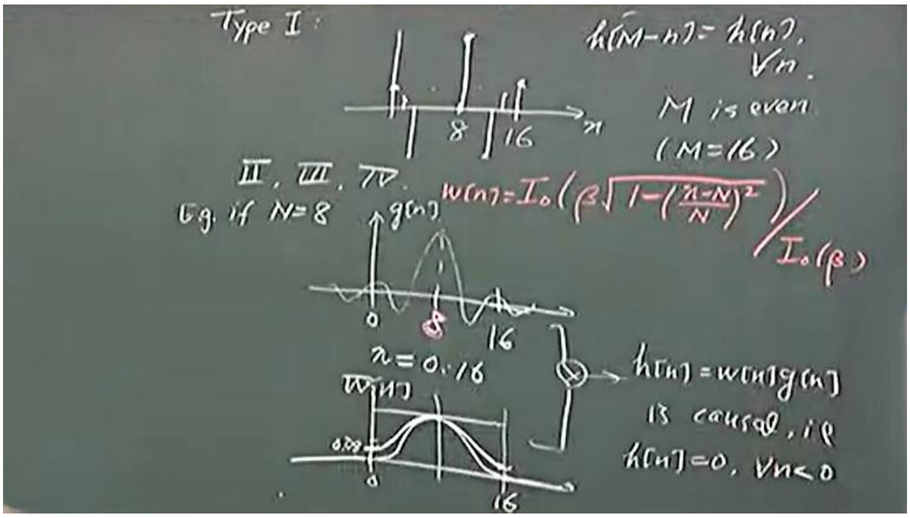
Remarks: Another family of windows can be constructed to outperform the cosine family for FIR filter design. The family of windows, called Kaiser windows, can be continuously tuned by a parameter $\beta \geq 0$.

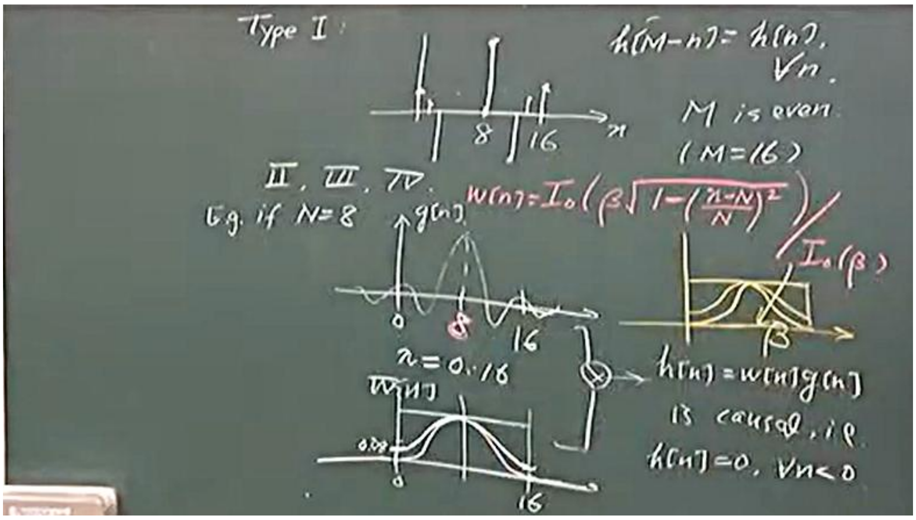
Nevertheless, the cosine family is better suited for *blockwise processing* of streaming audio and speech. We will cover it near the end of this semester.

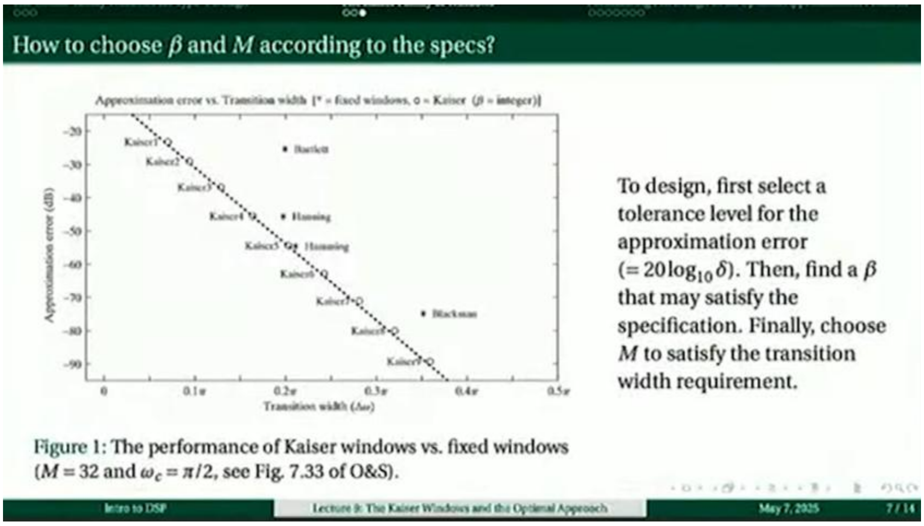
1. The Cosine-family Windows for Type-1 Design
2. The Kaiser Family of Windows
3. Formulating FIR Design as an Optimal Approximation Problem

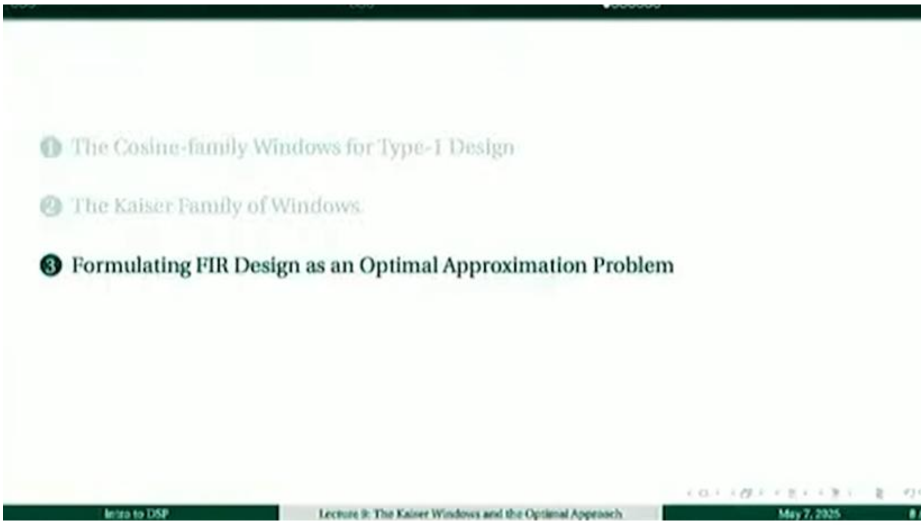

The type I low-pass design
Consider the type I[1] low-pass filter with a cutoff frequency of ω_c. We first shift the ideal response to the right,
$$g[n] = \frac{\sin \omega_c (n-N)}{\pi (n-N)}.$$
Then, we design a filter by $h[n] = w[n]g[n]$, and the window function $w[n]$ is defined in $n = 0:2N$ as follows,
- Rectangular: $w[n] = 1$
- Hann: $w[n] = 0.5 - 0.5\cos(\pi n/N)$
- Hamming: $w[n] = 0.54 - 0.46\cos(\pi n/N)$, and
- Blackmann: $w[n] = 0.42 - 0.5\cos(\pi n/N) + 0.08\cos(2\pi n/N)$.
Also, $w[n] = 0$ elsewhere. These windows are called the cosine family of windows.
[1] Definition of Type I: $h[M-n] = h[n]$ with M being an even number. See Sec. 5.7.3 of O&S.
Intro to DSP
Lecture 9: The Kaiser Windows and the Optimal Approach
May 7, 2025
3 / 14


Definition and spectral properties
The Kaiser family of windows is expressed as follows,
$$w[n] = \frac{I_0\left(\beta\sqrt{1-\left(\frac{n-N}{N}\right)^2}\right)}{I_0(\beta)},$$
where $N = \frac{M}{2}$, $n = 0:M$, $\beta \geq 0$ is a parameter, and $I_0(\cdot)$ denotes the *zeroth-order modified Bessel function of the first kind.*
$I_0(v) = 1 + (\frac{v}{2})^2 + \frac{(v/2)^4}{(2!)^2} + \frac{(v/2)^6}{(3!)^2} + \ldots$
Intro to DSP
Lecture 9: The Kaiser Windows and the Optimal Approach
May 7, 2025


Type I :
$h[M-n] = h[n], \forall n.$
M is even
(M=16)
8
16
n
II, III, IV.
E.g. if N=8
g[n]
$w[n] = I_0\left(\beta\sqrt{1-\left(\frac{n-N}{N}\right)^2}\right) / I_0(\beta)$
0
8
16
n = 0:16
w[n]
0.08
0
16
$h[n] = w[n]g[n]$
is causal, i.e.
$h[n] = 0, \forall n < 0$


Type I
h[M−n] = h[n], ∀n.
M is even
(M=16)
8
16
II, III, IV.
E.g. if N=8
g[n]
w[n] = I₀(β√(1−((n−N)/N)²)) / I₀(β)
0
8
16
β
n = 0..16
w[n]
h[n] = w[n]g[n]
is causal, i.e.
h[n] = 0, ∀n<0
0
16


How to choose β and M according to the specs?
Approximation error vs. Transition width [* = fixed windows, o = Kaiser (β = integer)]
Approximation error (dB)
Transition width (Δω)
Kaiser1
Kaiser2
Kaiser3
Kaiser4
Kaiser5
Kaiser6
Kaiser7
Kaiser8
Kaiser9
Bartlett
Hanning
Hamming
Blackman
To design, first select a tolerance level for the approximation error (= 20log₁₀δ). Then, find a β that may satisfy the specification. Finally, choose M to satisfy the transition width requirement.
Figure 1: The performance of Kaiser windows vs. fixed windows (M = 32 and ωc = π/2, see Fig. 7.33 of O&S).
Intro to DSP
Lecture 8: The Kaiser Windows and the Optimal Approach
May 7, 2025
7 / 14


1 The Cosine-family Windows for Type-1 Design
2 The Kaiser Family of Windows
3 Formulating FIR Design as an Optimal Approximation Problem
Intro to DSP
Lecture 8: The Kaiser Windows and the Optimal Approach
May 7, 2025

Preparation

Constraints: Consider any arbitrary zero-phase type I filter $h[n]$ defined on $n = -N : N$ with $h[-n] = h[n]$.
Frequency response: $A(\omega) = H(e^{j\omega}) = h[0] + 2\sum_{n=1}^{N} h[n]\cos(\omega n)$.
Goal: To minimize the following objective function:

$$\max_{\omega \in F} |E(\omega)|$$

where F denotes a union of passband(s) and stopband(s), and

$$E(\omega) = W(\omega)[H_d(e^{j\omega}) - A(\omega)]$$

contains the desired response $H_d(e^{j\omega}) \in \mathbb{R}$ and a flexible weighting function $W(\omega) > 0$.

Intro to DSP | Lecture 9: The Kaiser Windows and the Optimal Approach | May 7, 2025 | 9 / 14

Converting to a polynomial approximation problem

Define $x = \cos\omega$ and note that $A(\omega)$ can always be written as a polynomial of x in $[-1, 1]$. Denote the results conversion as

$$A(\omega) = h[0] + 2\sum_{n=1}^{N} h[n]\cos(\omega n) = P(x) = \sum_{k=0}^{N} a_k x^k.$$

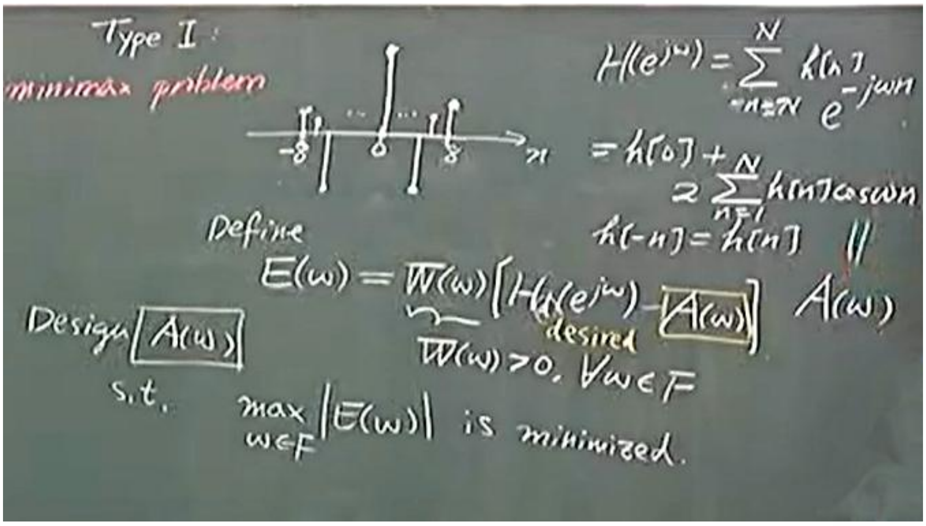
Remarks: $\cos(n\omega) := T_n(\cos\omega)$ and $T_n(x)$ is called the n^{th} order Chebyshev polynomial. E.g., $T_2(x) = 2x^2 - 1$.

Then, the problem formulation becomes to find the optimal N^{th}-order polynomial $\hat{P}(x)$:

$$\hat{P}(x) = \arg\min_{P_N(\mathbb{R})} \left(\max_{x \in \cos(F)} W_P(x) |D_P(x) - P(x)| \right)$$

where $W_P(x)$ is the weighting after converting to the x domain and $D_P(x)$ denotes the desired response in the x domain.

Intro to DSP | Lecture 9: The Kaiser Windows and the Optimal Approach | May 7, 2025 | 10 / 1



$$A(\omega) = h[0] + 2h[1]\cos\omega + 2h[2]\cos(2\omega) + \cdots + 2h[8]\cos(8\omega)$$

$$\cos(2\omega) = 2\cos^2\omega - 1 = 2x^2 - 1 = T_2(\cos\omega)$$

$$\cos(8\omega) \rightarrow T_8(x) \quad \text{\# Chebyshev Polynomial}$$

$\Uparrow x = \cos\omega$

$$\cdots = P(x) = a_0 + a_1 x + \cdots + a_8 x^8$$

$$A(\omega) = h[0] + 2h[1]\cos\omega + 2h[2]\cos(2\omega) + \cdots + 2h[8]\cos(8\omega)$$

$$\cos(2\omega) = 2\cos^2\omega - 1 = 2x^2 - 1 = T_2(\cos\omega)$$

$$\cos(8\omega) \rightarrow T_8(x) \quad \text{\# Chebyshev Polynomial}$$

$\Uparrow x = \cos\omega$

$$\cdots = P(x) = a_0 + a_1 x + \cdots + a_8 x^8$$

$$\hat{P}(x) = \underset{\boxed{P(x)} \in P_N(\mathbb{R})}{\arg\min} \left(\max_{x \in \cos(F)} W_p(x) \left| D_p(x) - P(x) \right| \right)$$

Applying the alternation theorem to assess optimality (Fig. 7.43 in O&S)

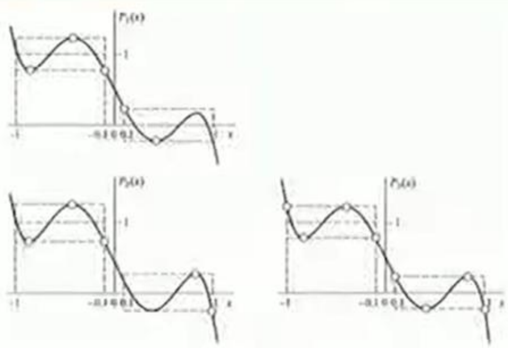


Figure 2: Suppose that $F_p = [-1, -0.1] \cup [0.1, 1]$, $D_P(x) = 1$ in $[-1, -0.1]$ (passband), and $D_P(x) = 0$ in $[0.1, 1]$ (stopband). Which one is optimal?

Intro to DSP | Lecture 9: The Kaiser Windows and the Optimal Approach | May 7, 2025

The alternation theorem

Let F_P denote a union of disjoint closed intervals of the real axis x. Then, the necessary and sufficient condition for a *unique*

$$\hat{P}(x) = \underset{\mathrm{P_N(R)}}{\arg\min}\left(\max_{x\in F_P} W_P(x)|D_P(x) - P(x)|\right)$$

is that there exist at least $N+2$ distinct values x_i in F_P, denoted as $x_1 < x_2 < ... < x_{N+2}$ such that

$$E_P(x_i) = -E_P(x_{i+1}) = \pm\max_{x\in F_P}|E_p(x)|$$

for $i = 1, 2, ..., N+1$. Here, $E_P(x) = W_P(x)[D_P(x) - P(x)]$.

Before we study how to find such a polynomial, let us visualize the theorem in the next few slides.

Intro to DSP | Lecture 9: The Kaiser Windows and the Optimal Approach | May 7, 2025 | 1

Converting to a polynomial approximation problem

Define $x = \cos\omega$ and note that $A(\omega)$ can always be written as a polynomial of x in $[-1, 1]$. Denote the results conversion as

$$A(\omega) = h[0] + 2\sum_{n=1}^{N} h[n]\cos(\omega n) = P(x) = \sum_{k=0}^{N} a_k x^k.$$

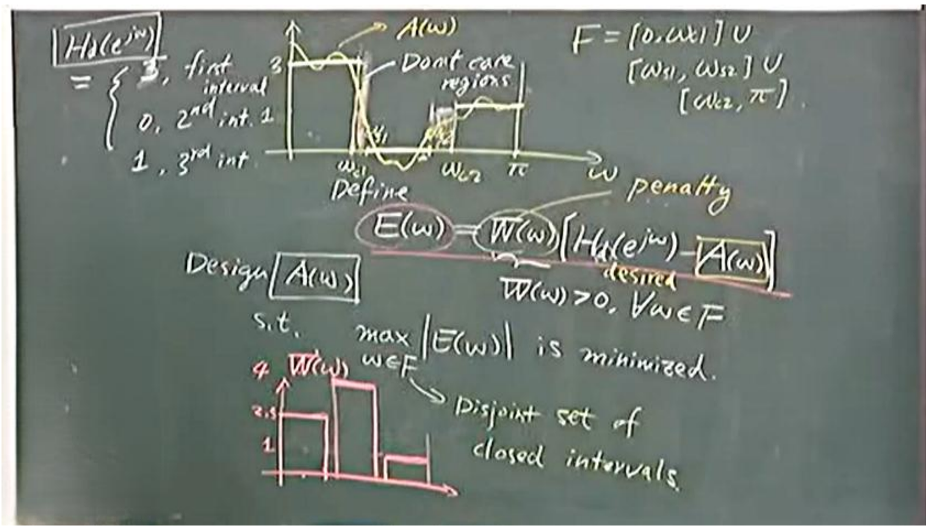
Remarks: $\cos(n\omega) := T_n(\cos\omega)$ and $T_n(x)$ is called the n^{th} order Chebyshev polynomial. E.g., $T_2(x) = 2x^2 - 1$.

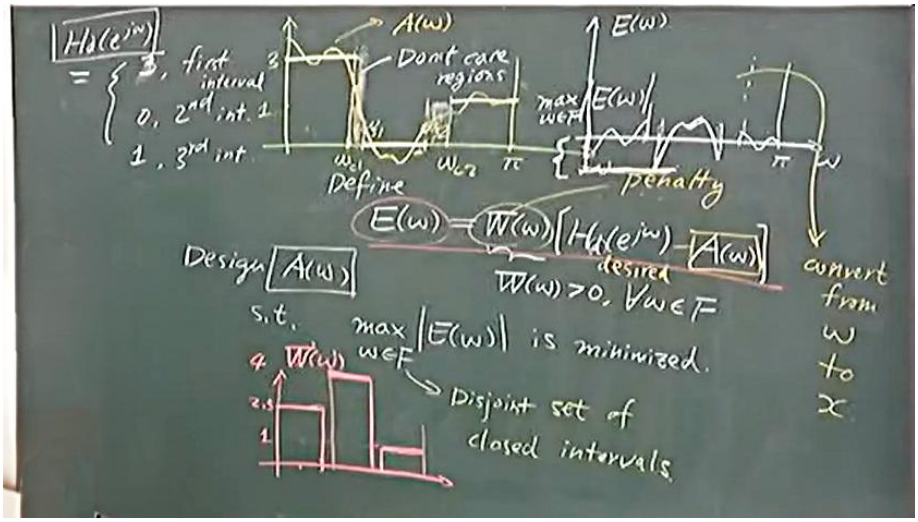
Then, the problem formulation becomes to find the optimal N^{th}-order polynomial $\hat{P}(x)$:

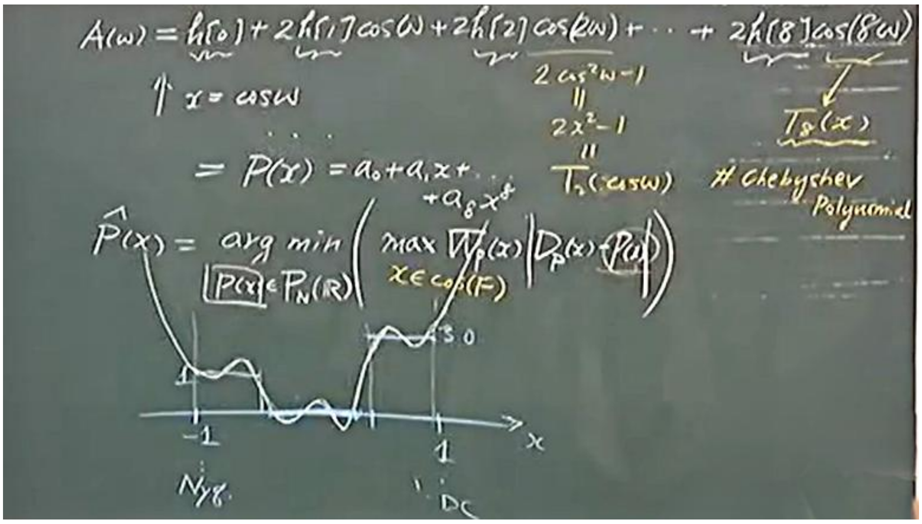
$$\hat{P}(x) = \underset{\mathrm{P_N(R)}}{\arg\min}\left(\max_{x\in\cos(F)} W_P(x)|D_P(x) - P(x)|\right)$$

where $W_P(x)$ is the weighting after converting to the x domain and $D_P(x)$ denotes the desired response in the x domain.

Intro to DSP | Lecture 9: The Kaiser Windows and the Optimal Approach | May 7, 2025 | 10 / 14







The alternation theorem

Let F_P denote a union of disjoint closed intervals of the real axis x. Then, the necessary and sufficient condition for a *unique*

$$\hat{P}(x) = \underset{\mathrm{P_N(\mathbb{R})}}{\operatorname{argmin}} \left(\max_{x \in F_p} W_P(x) |D_P(x) - P(x)| \right)$$

is that there exist at least $N+2$ distinct values x_i in F_P, denoted as $x_1 < x_2 < \ldots < x_{N+2}$ such that

$$E_P(x_i) = -E_P(x_{i+1}) = \pm \max_{x \in F_p} |E_p(x)|$$

for $i = 1, 2, \ldots, N+1$. Here, $E_P(x) = W_P(x)[D_P(x) - P(x)]$.

Before we study how to find such a polynomial, let us visualize the theorem in the next few slides.

Intro to DSP | Lecture 9: The Kaiser Windows and the Optimal Approach | May 7, 2025

Applying the alternation theorem to assess optimality (Fig. 7.43 in O&S)

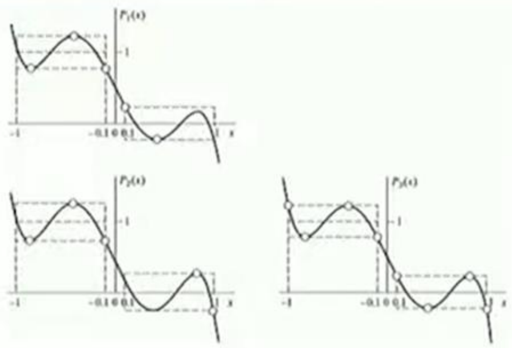


Figure 2: Suppose that $F_p = [-1, -0.1] \cup [0.1, 1]$, $D_P(x) = 1$ in $[-1, -0.1]$ (passband), and $D_P(x) = 0$ in $[0.1, 1]$ (stopband). Which one is optimal?

Preparation

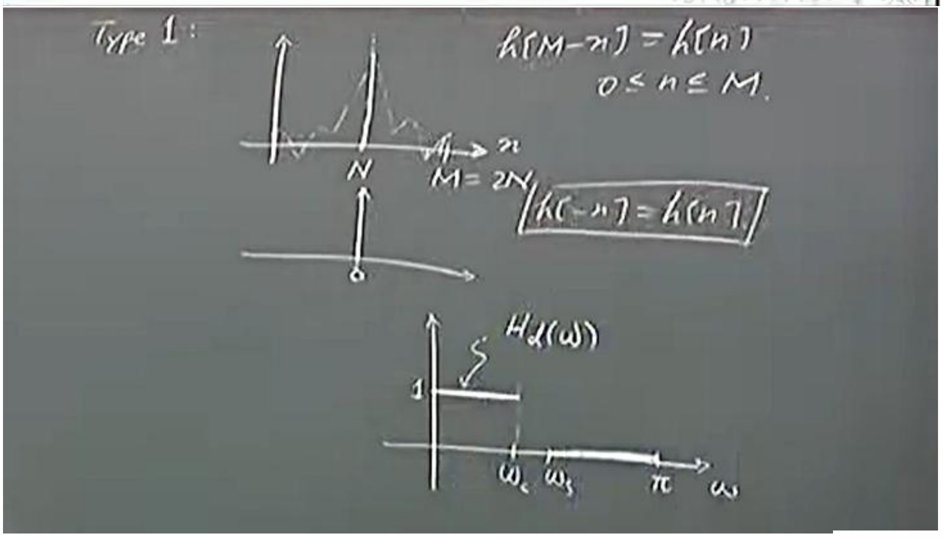
Constraints: Consider any arbitrary zero-phase type I filter $h[n]$ defined on $n = -N : N$ with $h[-n] = h[n]$.
Frequency response: $A(\omega) = H(e^{j\omega}) = h[0] + 2\sum_{n=1}^{N} h[n]\cos(\omega n)$.
Goal: To minimize the following objective function:

$$\max_{\omega \in F} |E(\omega)|$$

where F denotes a union of passband(s) and stopband(s), and

$$E(\omega) = W(\omega)[H_d(e^{j\omega}) - A(\omega)] \qquad (1)$$

contains the desired response $H_d(e^{j\omega}) \in \mathbb{R}$ and a flexible weighting function $W(\omega) > 0$.



Converting to a polynomial approximation problem

Define $x = \cos\omega$ and note that $A(\omega)$ can always be written as a polynomial of x in $[-1, 1]$. Denote the results conversion as

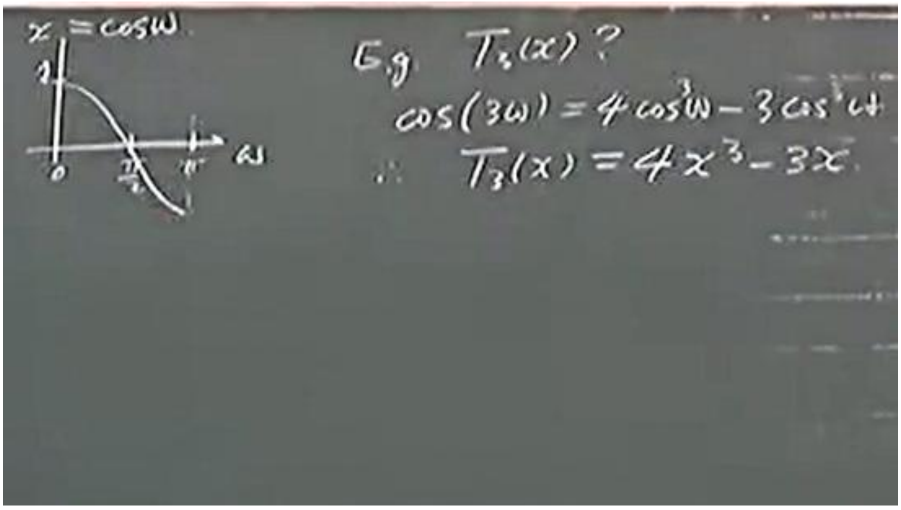
$$A(\omega) = h[0] + 2\sum_{n=1}^{N} h[n]\cos(\omega n) = P(x) = \sum_{k=0}^{N} a_k x^k.$$

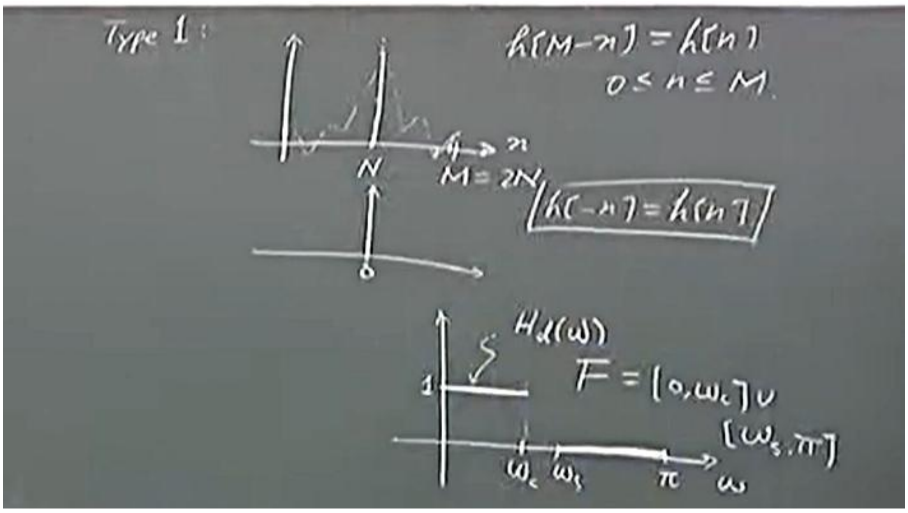
Remarks: $\cos(n\omega) := T_n(\cos\omega)$ and $T_n(x)$ is called the n^{th} order Chebyshev polynomial. E.g., $T_2(x) = 2x^2 - 1$.

Then, the problem formulation becomes to find the optimal N^{th}-order polynomial $\hat{P}(x)$:

$$\hat{P}(x) = \underset{P(x)\in \mathsf{P}_N(\mathbb{R})}{\arg\min} \left(\max_{x\in\cos(F)} W_P(x)|D_P(x) - P(x)| \right)$$

where $W_P(x)$ is the weighting after converting to the x domain and $D_P(x)$ denotes the desired response in the x domain.





Define $x = \cos\omega$ and note that $A(\omega)$ can always be written as a polynomial of x in $[-1, 1]$. Denote the results conversion as

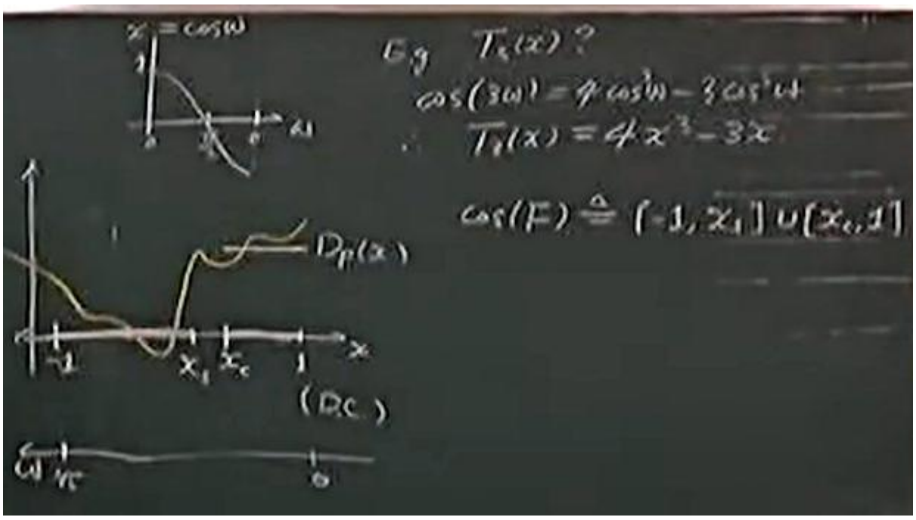
$$A(\omega) = h[0] + 2\sum_{n=1}^{N} h[n]\cos(\omega n) = P(x) = \sum_{k=0}^{N} a_k x^k.$$

Remarks: $\cos(n\omega) := T_n(\cos\omega)$ and $T_n(x)$ is called the n^{th} order Chebyshev polynomial. E.g., $T_2(x) = 2x^2 - 1$.

Then, the problem formulation becomes to find the optimal N^{th}-order polynomial $\hat{P}(x)$:

$$\hat{P}(x) = \underset{P(x)\in \mathsf{P}_N(\mathbb{R})}{\operatorname{argmin}} \left(\max_{x\in\cos(F)} W_P(x)|D_P(x) - P(x)| \right)$$

where $W_P(x)$ is the weighting after converting to the x domain and $D_P(x)$ denotes the desired response in the x domain.



The alternation theorem

Let F_P denote a union of disjoint closed intervals of the real axis x. Then, the necessary and sufficient condition for a *unique*

$$\hat{P}(x) = \underset{P(x)\in \mathsf{P}_N(\mathbb{R})}{\operatorname{argmin}} \left(\max_{x\in F_p} W_P(x)|D_P(x) - P(x)| \right)$$

is that there exist at least $N+2$ distinct values x_i in F_P, denoted as $x_1 < x_2 < \ldots < x_{N+2}$ such that

$$E_P(x_i) = -E_P(x_{i+1}) = \pm\max_{x\in F_p}|E_p(x)|$$

for $i = 1, 2, \ldots, N+1$. Here, $E_P(x) = W_P(x)[D_P(x) - P(x)]$.

Before we study how to find such a polynomial, let us visualize the theorem in the next few slides.

Applying the alternation theorem to assess optimality (Fig. 7.43 in O&S)

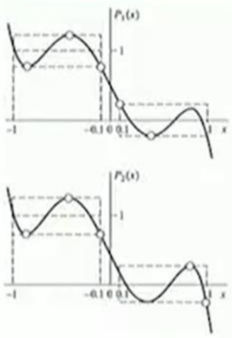
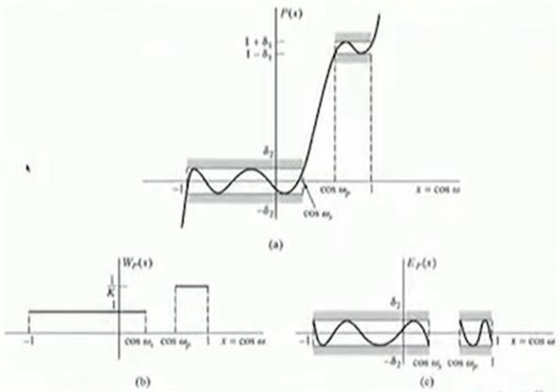


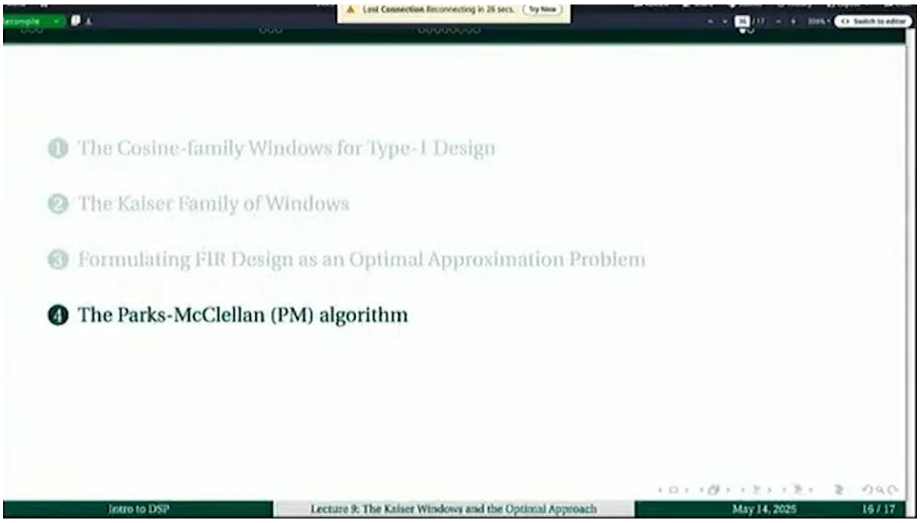
Figure 2: Suppose that $F_p = [-1, -0.1] \cup [0.1, 1]$, $D_P(x) = 1$ in $[-1, -0.1]$ (passband), and $D_P(x) = 0$ in $[0.1, 1]$ (stopband). Which one is optimal?

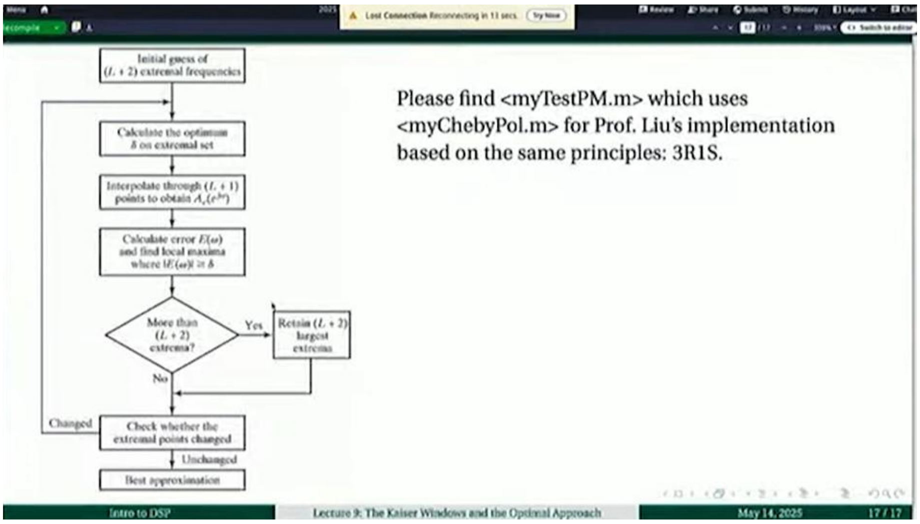
Converting back from x to ω domain (Fig. 7.45 in O&S)

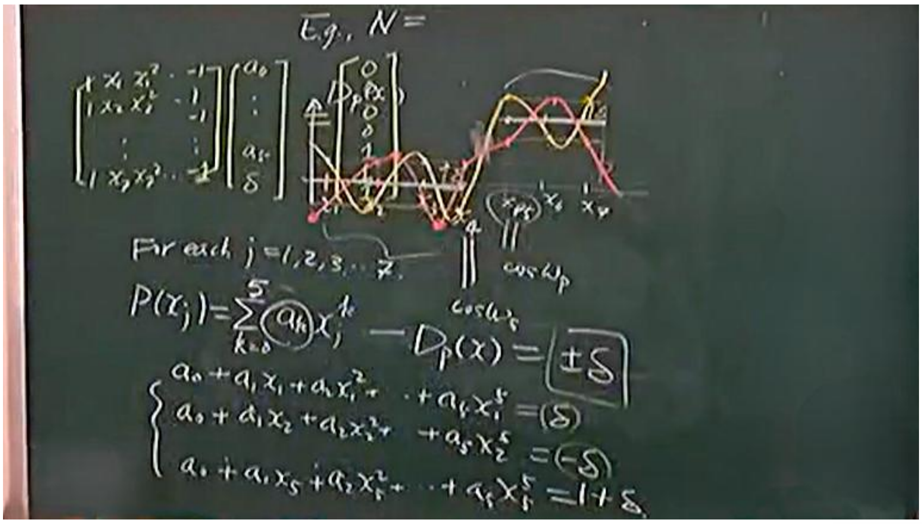


Summary

- The optimal approximation approach provides a unified framework for designing low-pass, high-pass, band-pass, band-reject filters, and so on.
- It also gives us the flexibility to vary the tolerance level between different bands.
- The search for the optimal polynomial involves 3 'R' steps and 1 'S' step: **R**andom guess, **R**efinement, **R**epeat, and a **S**topping criterion. We will explain it next.


1 The Cosine-family Windows for Type-1 Design
2 The Kaiser Family of Windows
3 Formulating FIR Design as an Optimal Approximation Problem
4 The Parks-McClellan (PM) algorithm
Intro to DSP
Lecture 9: The Kaiser Windows and the Optimal Approach
May 14, 2025
16 / 17

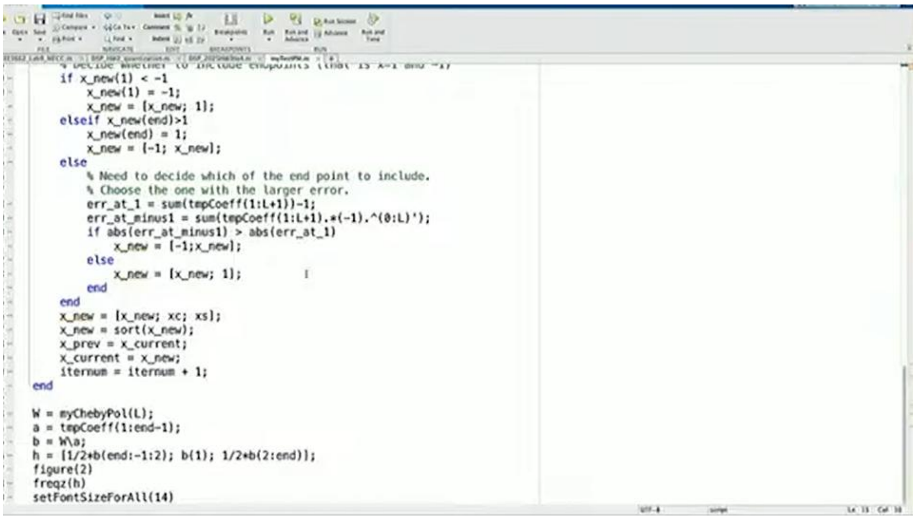

Initial guess of (L + 2) extremal frequencies
Calculate the optimum δ on extremal set
Interpolate through (L + 1) points to obtain $A_e(e^{j\omega})$
Calculate error $E(\omega)$ and find local maxima where $|E(\omega)| \geq \delta$
More than (L + 2) extrema?
Yes
Retain (L + 2) largest extrema
No
Changed
Check whether the extremal points changed
Unchanged
Best approximation
Please find <myTestPM.m> which uses <myChebyPol.m> for Prof. Liu's implementation based on the same principles: 3R1S.
Intro to DSP
Lecture 9: The Kaiser Windows and the Optimal Approach
May 14, 2025
17 / 17

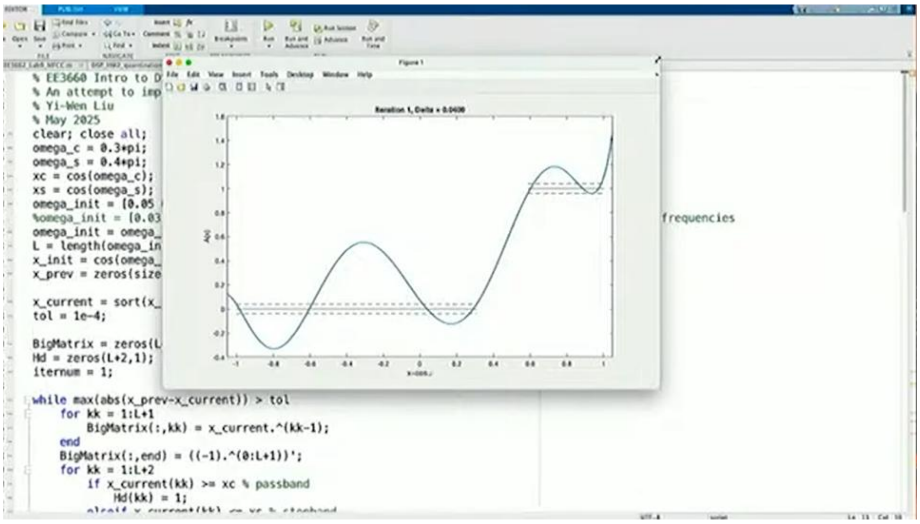

E.g., N =
For each j = 1, 2, 3, ... 7.
$P(x_j) = \sum_{k=0}^{5} a_k x_j^k - D_p(x) = \boxed{\pm\delta}$
$\cos\omega_p$
$\cos\omega_s$
$a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_5 x_1^5 = \delta$
$a_0 + a_1 x_2 + a_2 x_2^2 + \cdots + a_5 x_2^5 = -\delta$
$a_0 + a_1 x_5 + a_2 x_5^2 + \cdots + a_5 x_5^5 = 1 + \delta$

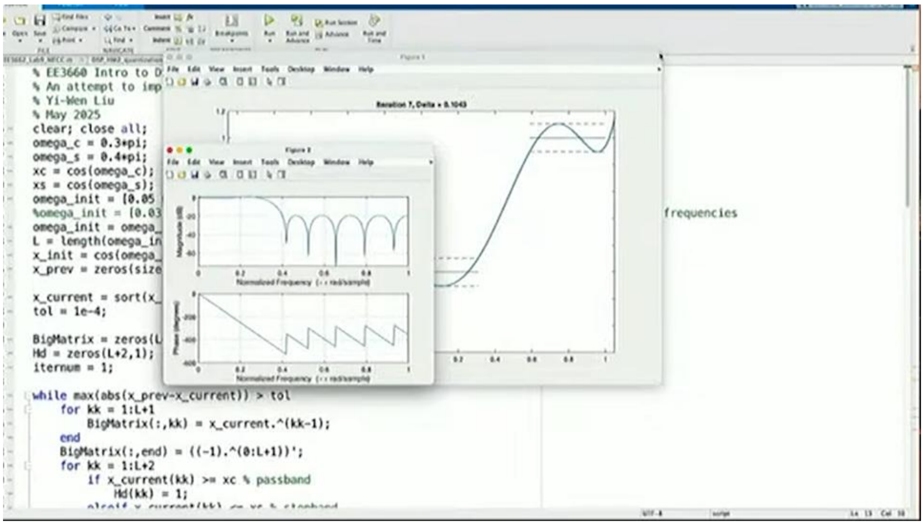
```matlab
% EE3660 Intro to DSP
% An attempt to implement the Parks-McClellan algorithm from scratch
% Yi-Wen Liu
% May 2025
clear; close all;
omega_c = 0.3*pi;
omega_s = 0.4*pi;
xc = cos(omega_c);
xs = cos(omega_s);
omega_init = [0.05 0.12 0.17 omega_c/pi omega_s/pi 0.48 0.7 0.92 1]*pi;
%omega_init = [0.03 0.05 0.12 0.17 0.26 omega_c/pi omega_s/pi 0.5 0.65 0.7 0.92 1]*pi; % L+2 frequencies
omega_init = omega_init(:);
L = length(omega_init)-2;
x_init = cos(omega_init);
x_prev = zeros(size(x_init));

x_current = sort(x_init);
tol = 1e-4;

BigMatrix = zeros(L+2,L+2);
Hd = zeros(L+2,1);
iternum = 1;

while max(abs(x_prev-x_current)) > tol
    for kk = 1:L+1
        BigMatrix(:,kk) = x_current.^(kk-1);
    end
    BigMatrix(:,end) = ((-1).^(0:L+1))';
    for kk = 1:L+2
        if x_current(kk) >= xc % passband
            Hd(kk) = 1;
```

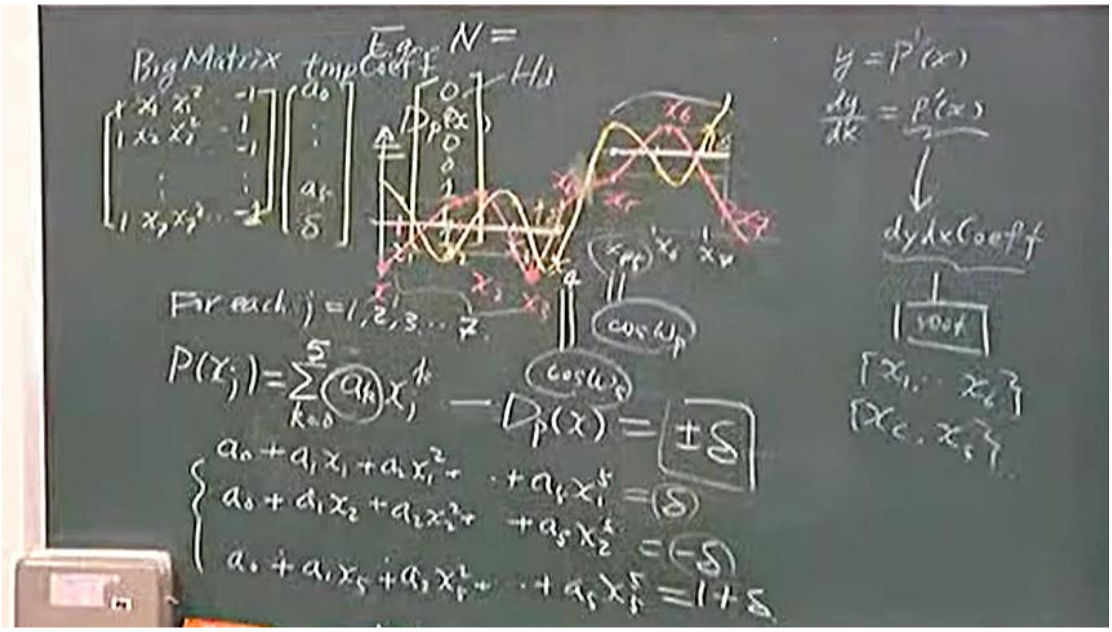
```matlab
Hd = zeros(L+2,1);
iternum = 1;

while max(abs(x_prev-x_current)) > tol
    for kk = 1:L+1
        BigMatrix(:,kk) = x_current.^(kk-1);
    end
    BigMatrix(:,end) = ((-1).^(0:L+1))';
    for kk = 1:L+2
        if x_current(kk) >= xc % passband
            Hd(kk) = 1;
        elseif x_current(kk) <= xs % stopband
            Hd(kk) = 0;
        end
    end
    tmpCoeff = BigMatrix\Hd;
    delta = abs(tmpCoeff(end));

    figure(1);
    xx = -1.05:0.01:1.05;
    xthispower = ones(size(xx));
    yy = tmpCoeff(1)*ones(size(xx)); % constant term
    for kk = 1:L
        xthispower = xthispower.*xx;
        yy = yy + tmpCoeff(kk+1)*xthispower;
    end
    plot(xx,yy,'LineWidth',2); hold on;
    line([xc 1],[1 1],'color','k');
    line([-1 xs],[0 0],'color','k');
    line([xc 1],[1+delta 1+delta],'linestyle','--','color','k');
    line([xc 1],[1-delta 1-delta],'linestyle','--','color','k');
```

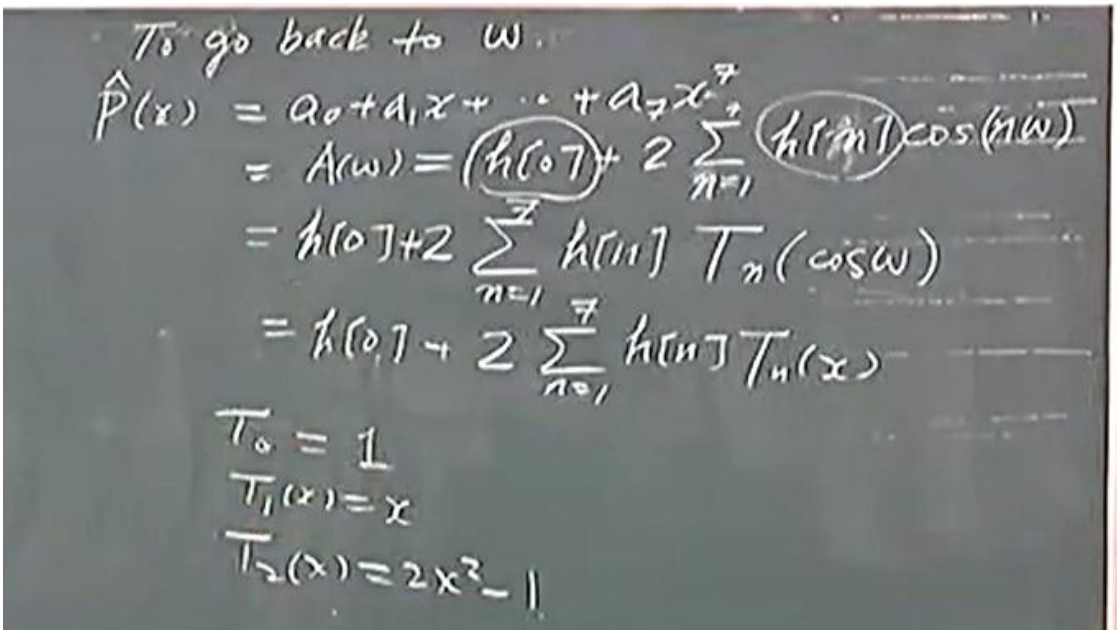
```matlab
    ylabel('A(x)');
    xlim([-1.05 1.05])
    setFontSizeForAll(14);
    pause;
    hold off;
    % Find L extrema of y(x) by identifying g(x) = dy/dx first.
    power = 1:L; power = power(:);
    dydxCoeff = tmpCoeff(power+1).*power; % a1 + (2*a2)x + (3*a3)x^2+...+(L*a_L)x^(L-1)
    % Find locations where 1st derivative vanishes
    x_new = roots(dydxCoeff(end:-1:1));
    x_new = sort(x_new);

    % Decide whether to include endpoints (that is x=1 and -1)
    if x_new(1) < -1
        x_new(1) = -1;
        x_new = [x_new; 1];
    elseif x_new(end)>1
        x_new(end) = 1;
        x_new = [-1; x_new];
    else
        % Need to decide which of the end point to include.
        % Choose the one with the larger error.
        err_at_1 = sum(tmpCoeff(1:L+1))-1;
        err_at_minus1 = sum(tmpCoeff(1:L+1).*(-1).^(0:L)');
        if abs(err_at_minus1) > abs(err_at_1)
            x_new = [-1;x_new];
        else
            x_new = [x_new; 1];
        end
    end
    x_new = [x_new; xc; xs];
```


if x_new(1) < -1
x_new(1) = -1;
x_new = [x_new; 1];
elseif x_new(end)>1
x_new(end) = 1;
x_new = [-1; x_new];
else
% Need to decide which of the end point to include.
% Choose the one with the larger error.
err_at_1 = sum(tmpCoeff(1:L+1))-1;
err_at_minus1 = sum(tmpCoeff(1:L+1).*(-1).^(0:L)');
if abs(err_at_minus1) > abs(err_at_1)
x_new = [-1;x_new];
else
x_new = [x_new; 1];
end
end
x_new = [x_new; xc; xs];
x_new = sort(x_new);
x_prev = x_current;
x_current = x_new;
iternum = iternum + 1;
end
W = myChebyPol(L);
a = tmpCoeff(1:end-1);
b = W\a;
h = [1/2*b(end:-1:2); b(1); 1/2*b(2:end)];
figure(2)
freqz(h)
setFontSizeForAll(14)


% EE3660 Intro to D
% An attempt to imp
% Yi-Wen Liu
% May 2025
clear; close all;
omega_c = 0.3*pi;
omega_s = 0.4*pi;
xc = cos(omega_c);
xs = cos(omega_s);
x_current = sort(x_
tol = 1e-4;
iternum = 1;
while max(abs(x_prev-x_current)) > tol
for kk = 1:L+1
BigMatrix(:,kk) = x_current.^(kk-1);
end
BigMatrix(:,end) = ((-1).^(0:L+1))';
for kk = 1:L+2
if x_current(kk) >= xc % passband
Hd(kk) = 1;
Figure 1
Iteration 1, Delta = 0.0430
frequencies


% EE3660 Intro to D
% An attempt to imp
% Yi-Wen Liu
% May 2025
clear; close all;
omega_c = 0.3*pi;
omega_s = 0.4*pi;
xc = cos(omega_c);
xs = cos(omega_s);
x_current = sort(x_
tol = 1e-4;
iternum = 1;
while max(abs(x_prev-x_current)) > tol
for kk = 1:L+1
BigMatrix(:,kk) = x_current.^(kk-1);
end
BigMatrix(:,end) = ((-1).^(0:L+1))';
for kk = 1:L+2
if x_current(kk) >= xc % passband
Hd(kk) = 1;
Iteration 7, Delta = 0.1043
Figure 2
Normalized Frequency (×π rad/sample)
frequencies


BigMatrix
tmpCoeff
E.g. N =
y = P'(x)
dy/dk = P'(x)
dydxCoeff
roots
For each j = 1, 2, 3, ... 7
P(x_j) = Σ_{k=0}^{5} a_k x_j^k
D_p(x) = ±δ
a0 + a1 x1 + a2 x1^2 + ... + a5 x1^5 = δ
a0 + a1 x2 + a2 x2^2 + ... + a5 x2^5 = −δ
a0 + a1 x5 + a2 x5^2 + ... + a5 x5^5 = 1 + δ


To go back to ω.
P̂(x) = a0 + a1 x + ... + a7 x^7
= A(ω) = h[0] + 2 Σ_{n=1} h[n] cos(nω)
= h[0] + 2 Σ_{n=1}^{7} h[n] T_n(cos ω)
= h[0] + 2 Σ_{n=1}^{7} h[n] T_n(x)
T_0 = 1
T_1(x) = x
T_2(x) = 2x^2 − 1