

Course 18.327 and 1.130

Wavelets and Filter Banks

**Sampling rate change operations:
upsampling and downsampling;
fractional sampling; interpolation**

Downsampling

Definition:

(↓2)

$$\begin{bmatrix} M \\ x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ M \end{bmatrix}$$

=

$$\begin{bmatrix} M \\ x[0] \\ x[2] \\ x[4] \\ M \\ \& \end{bmatrix}$$

As a matrix operation:

$$\begin{bmatrix} & & M & & & & \\ L & 1 & 0 & 0 & 0 & 0 & L \\ L & 0 & 0 & 1 & 0 & 0 & L \\ L & 0 & 0 & 0 & 0 & 1 & L \\ & & M & & & & \end{bmatrix}$$

$$\begin{bmatrix} M \\ x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ M \end{bmatrix}$$

=

$$\begin{bmatrix} : \\ x[0] \\ x[2] \\ x[4] \\ : \\ : \end{bmatrix}$$

Upsampling

Definition:

(↑2)

$$\begin{bmatrix} M \\ x[0] \\ x[1] \\ x[2] \\ M \end{bmatrix}$$

=

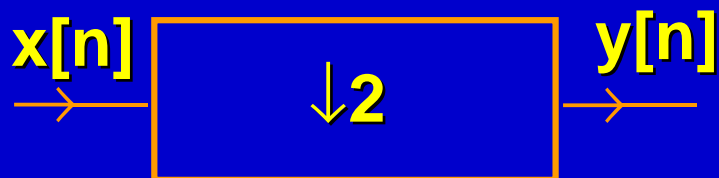
$$\begin{bmatrix} M \\ x[0] \\ 0 \\ x[1] \\ 0 \\ x[2] \\ 0 \\ M \end{bmatrix}$$

As a matrix operation:

$$\begin{bmatrix} & & M & & \\ L & 1 & 0 & 0 & L \\ L & 0 & 0 & 0 & L \\ L & 0 & 1 & 0 & L \\ L & 0 & 0 & 0 & L \\ L & 0 & 0 & 1 & L \\ L & 0 & 0 & 0 & L \\ & & M & & \end{bmatrix} \begin{bmatrix} M \\ x[0] \\ x[1] \\ x[2] \\ : \\ : \\ : \end{bmatrix} = \begin{bmatrix} M \\ x[0] \\ 0 \\ x[1] \\ 0 \\ x[2] \\ 0 \\ M \end{bmatrix}$$

Downsampling

Downsampling by 2



$$y[n] = x[2n]$$

$$Y(\omega) = \sum_n x[2n]e^{-i\omega n}$$

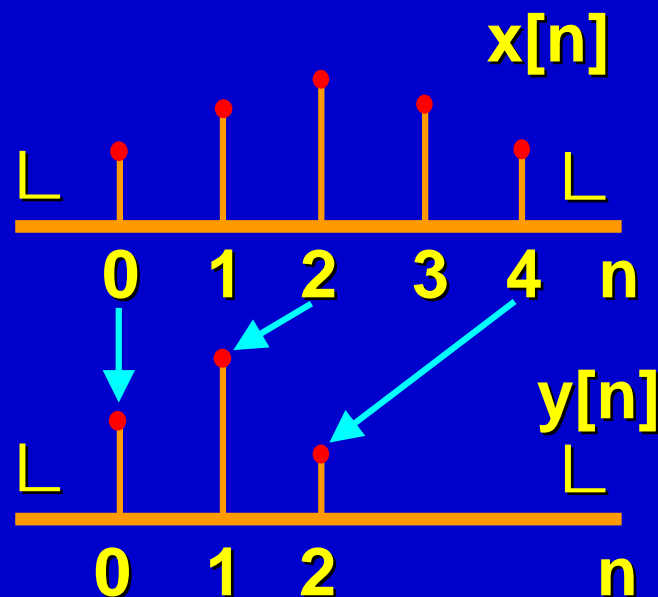
$$= \sum_{m \text{ even}} x[m]e^{-i\omega m/2}$$

$$= \frac{1}{2} \sum_m \{1 + (-1)^m\} x[m]e^{-i\omega m/2}$$

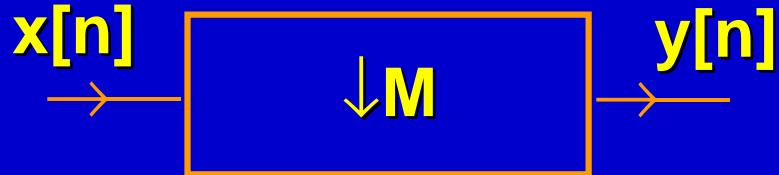
$$= \frac{1}{2} \left\{ \sum_m x[m]e^{-i\omega m/2} + \sum_m x[m]e^{-i(\frac{\omega}{2} + \pi)m} \right\};$$

$$(-1)^m = e^{-i\pi m}$$

$$= \frac{1}{2} \{X(\omega/2) + X(\omega/2 + \pi)\}$$



Downsampling by M



$$y[n] = x[Mn]$$

$$Y(\omega) = \sum_{m=nM} x[m]e^{-i\omega m/M}$$

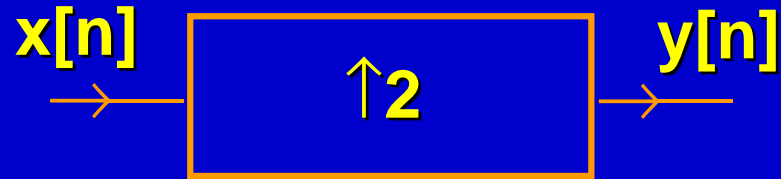
$$= \frac{1}{M} \sum_m \left\{ \sum_{k=0}^{M-1} e^{-i\frac{2\pi}{M}km} \right\} x[m]e^{-i\omega m/M};$$

$$\frac{1}{M} \sum_{k=0}^{M-1} (e^{-i\frac{2\pi}{M}m})^k = \begin{cases} 1 & \text{if } m = nM \\ 0 & \text{if } m \neq nM \end{cases}$$

$$= \frac{1}{M} \sum_{k=0}^{M-1} X\left(\frac{\omega + 2\pi k}{M}\right)$$

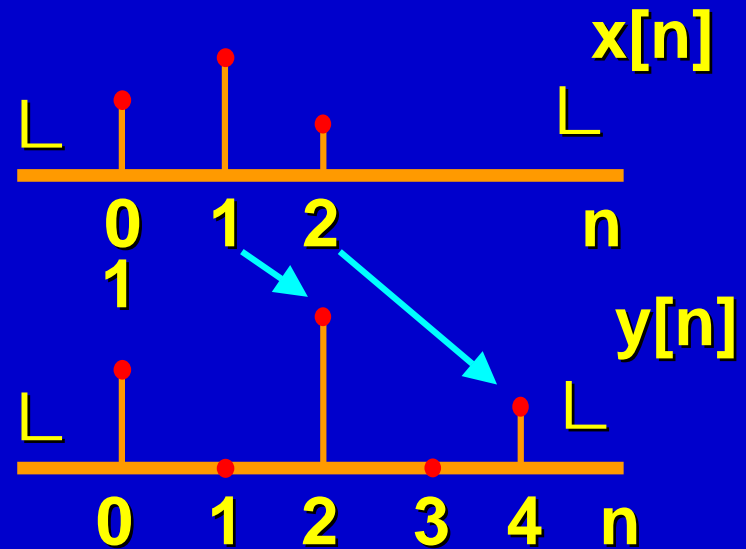
Upsampling

Upsampling by 2



$$y[n] = \begin{cases} x[n/2] & ; n \text{ even} \\ 0 & ; n \text{ odd} \end{cases}$$

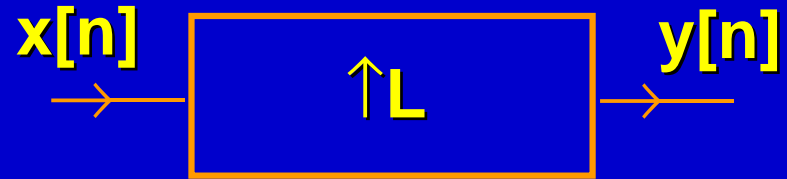
$$\begin{aligned} Y(\omega) &= \sum_{n \text{ even}} x[n/2] e^{-i\omega n} \\ &= \sum_m x[m] e^{-i\omega 2m} \\ &= X(2\omega) \end{aligned}$$



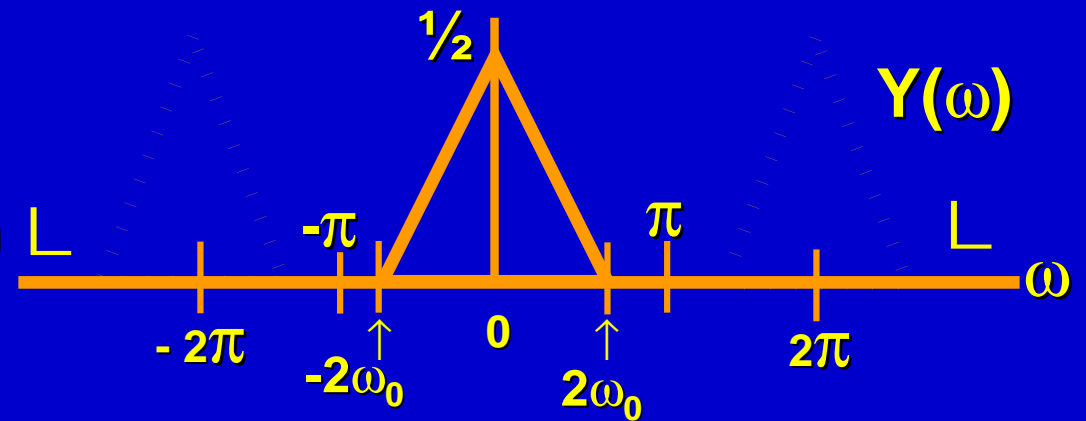
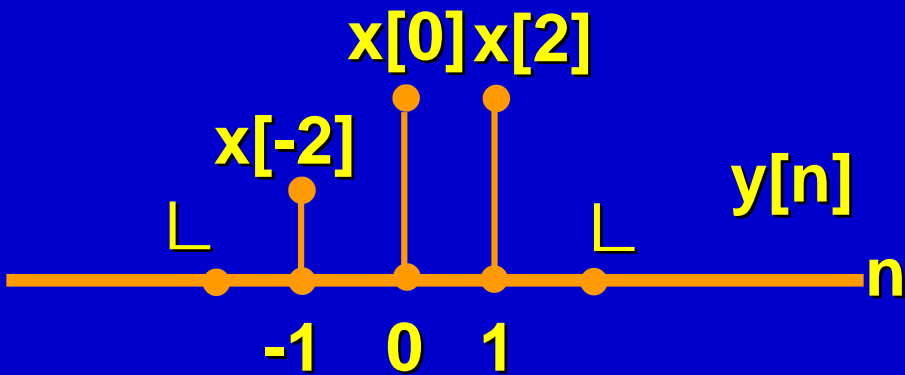
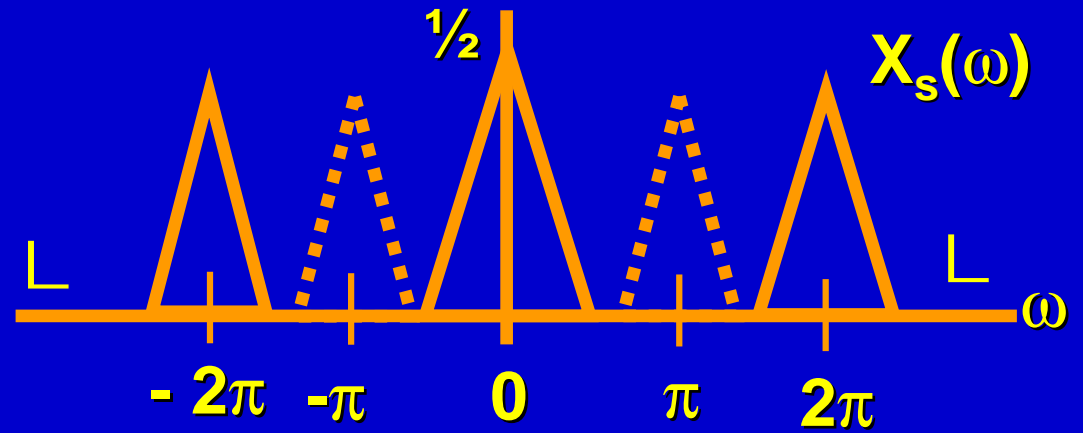
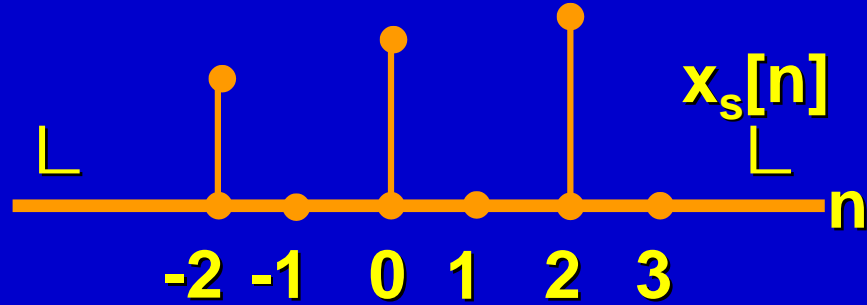
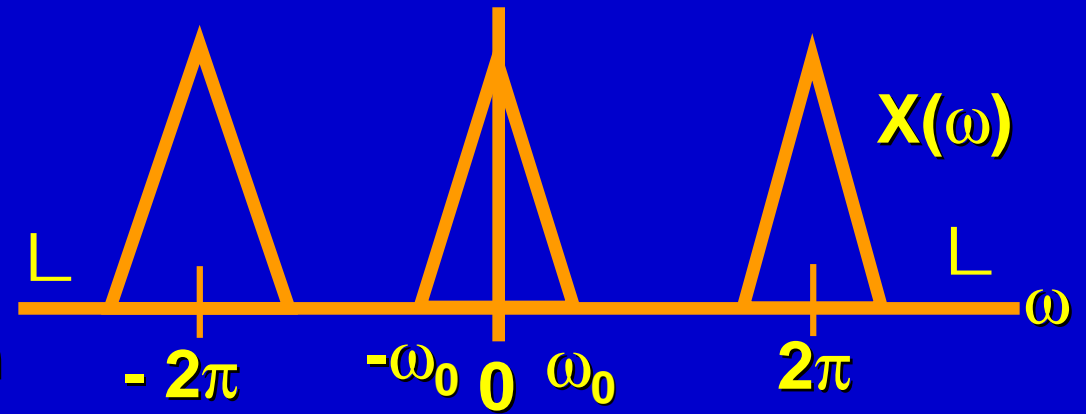
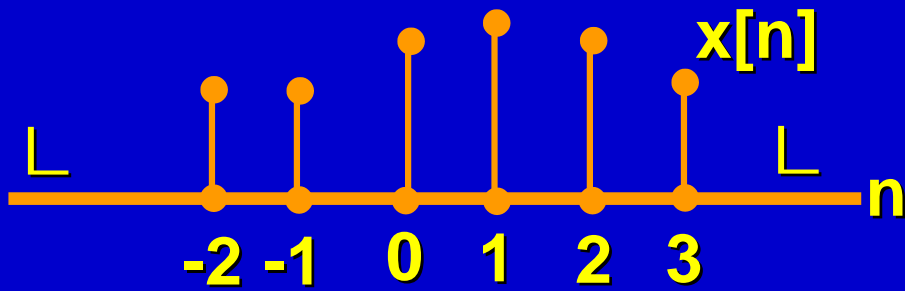
Upsampling by L

$$y[n] = \begin{cases} x[n/L] & ; n = mL \\ 0 & ; n \neq mL \end{cases}$$

$$\begin{aligned} Y(\omega) &= \sum_{n=mL} x[n/L] e^{-i\omega n} \\ &= X(L\omega) \end{aligned}$$



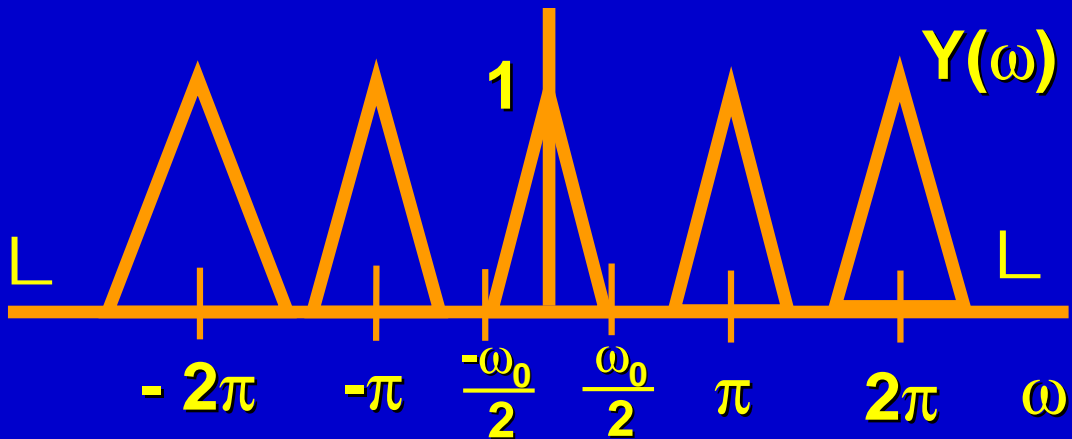
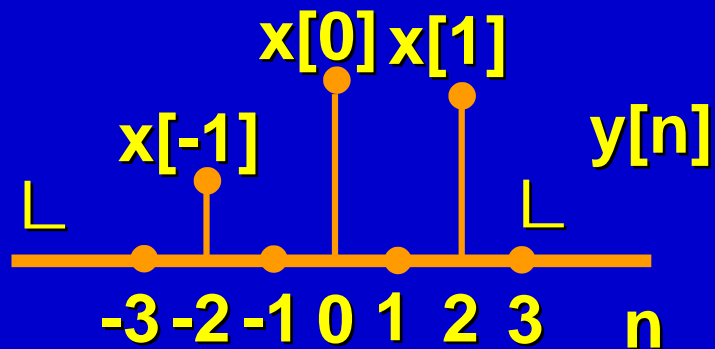
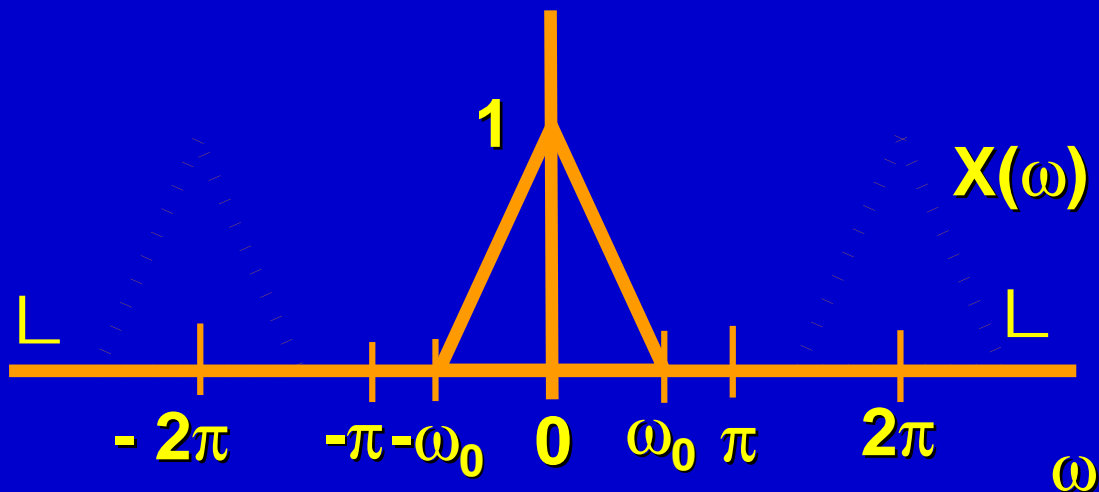
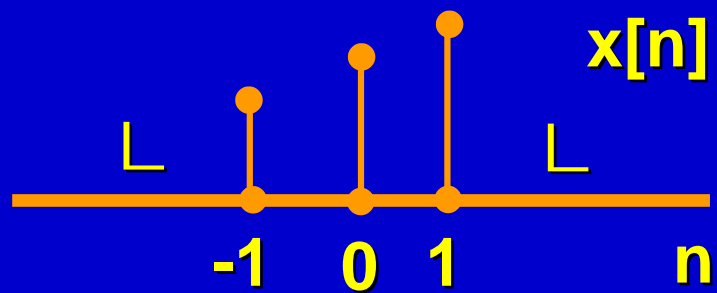
Downsampling



$$y[n] = (\downarrow 2) x[n] = x[2n]$$

$$Y(\omega) = \frac{1}{2} \left\{ X\left(\frac{\omega}{2}\right) + X\left(\frac{\omega}{2} + \pi\right) \right\}$$

Upsampling

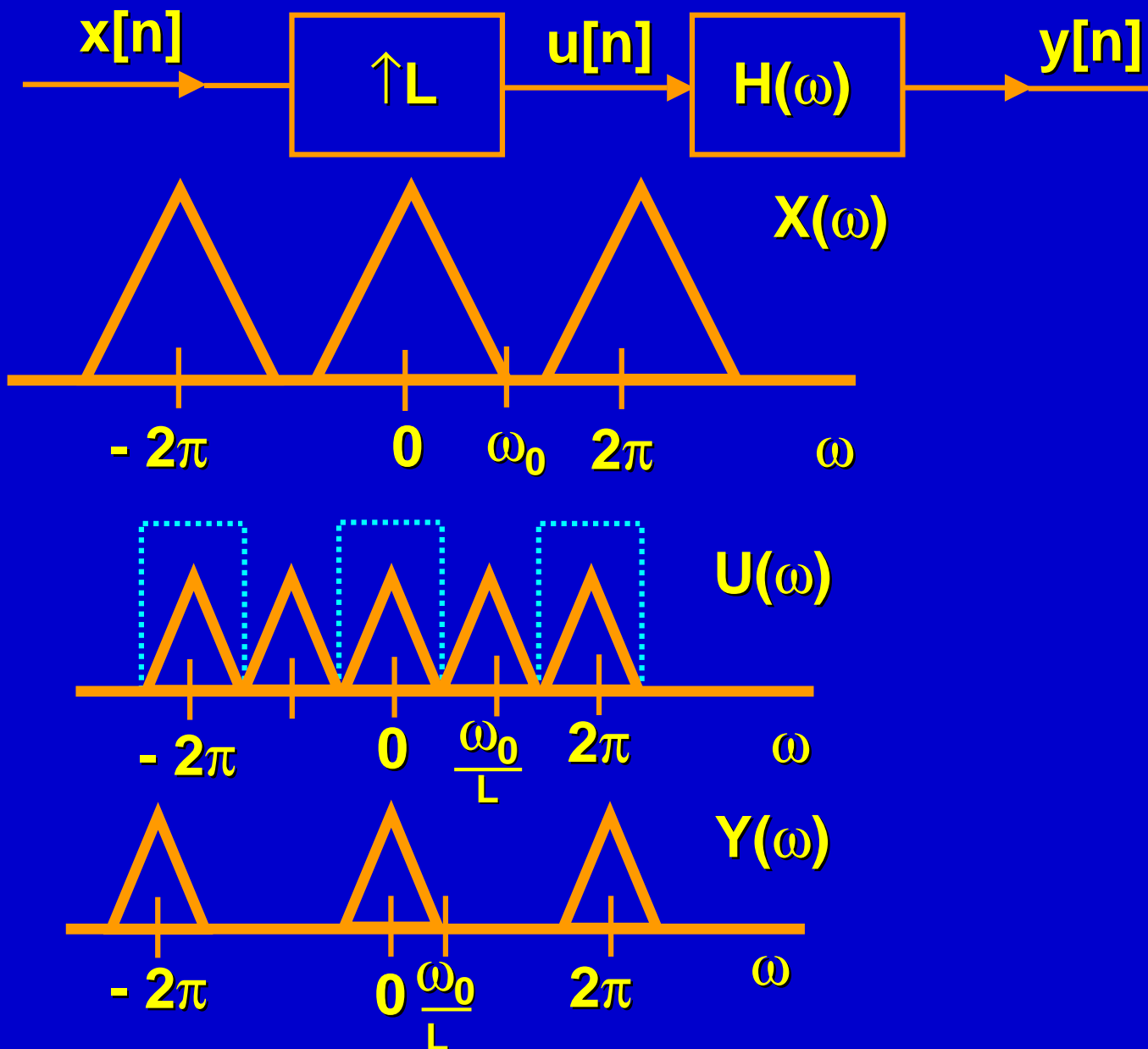


$$y[n] = \begin{cases} x[n/2] & ; n \text{ even} \\ 0 & ; n \text{ odd} \end{cases}$$

$$Y(\omega) = X(2\omega)$$

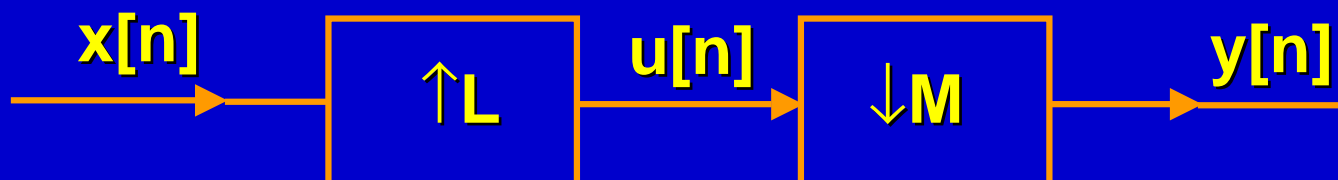
Interpolation

Use lowpass filter after upsampling



Fractional Sampling

Consider



$$\begin{aligned} Y(\omega) &= \frac{1}{M} \sum_{k=0}^{M-1} U\left(\frac{\omega + 2\pi k}{M}\right) \\ &= \frac{1}{M} \sum_{k=0}^{M-1} X\left(\frac{\omega + 2\pi k}{M} L\right) \end{aligned}$$

What about

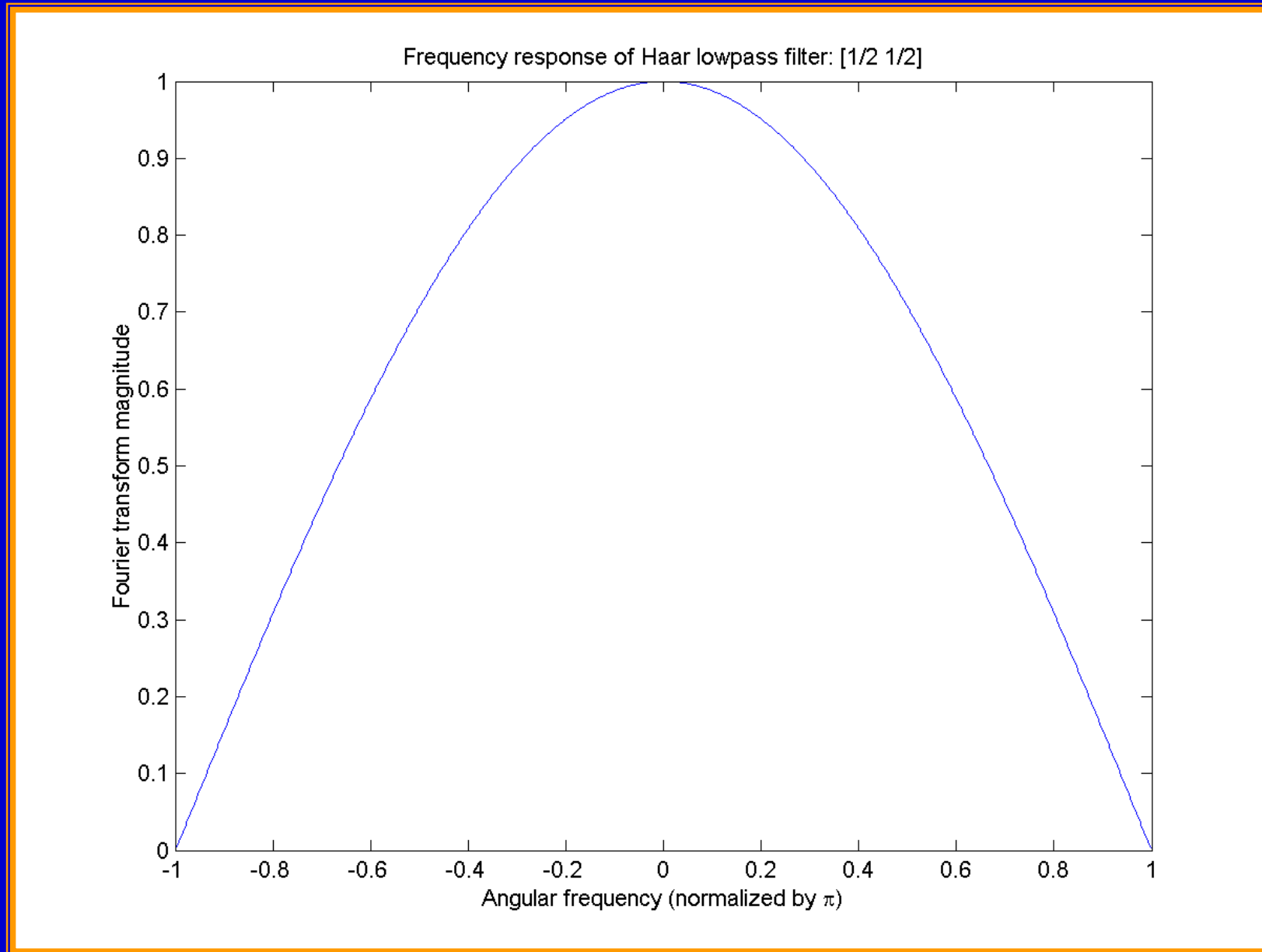


$$\begin{aligned} Y(\omega) &= D(\omega L) \\ &= \frac{1}{M} \sum_{k=0}^{M-1} X\left(\frac{\omega L + 2\pi k}{M}\right) \end{aligned}$$

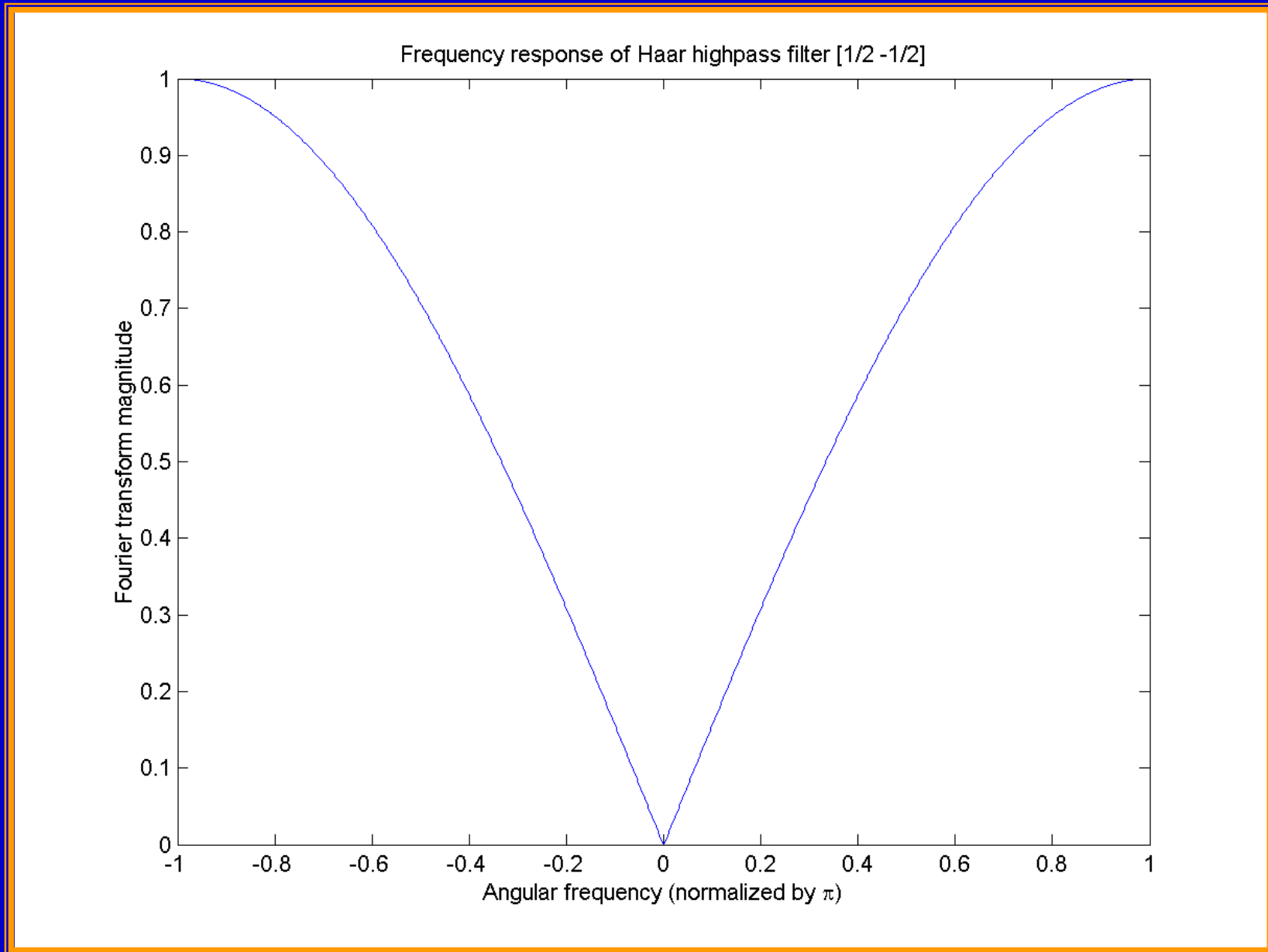
Matlab Example 1

Basic filters, upsampling and downsampling.

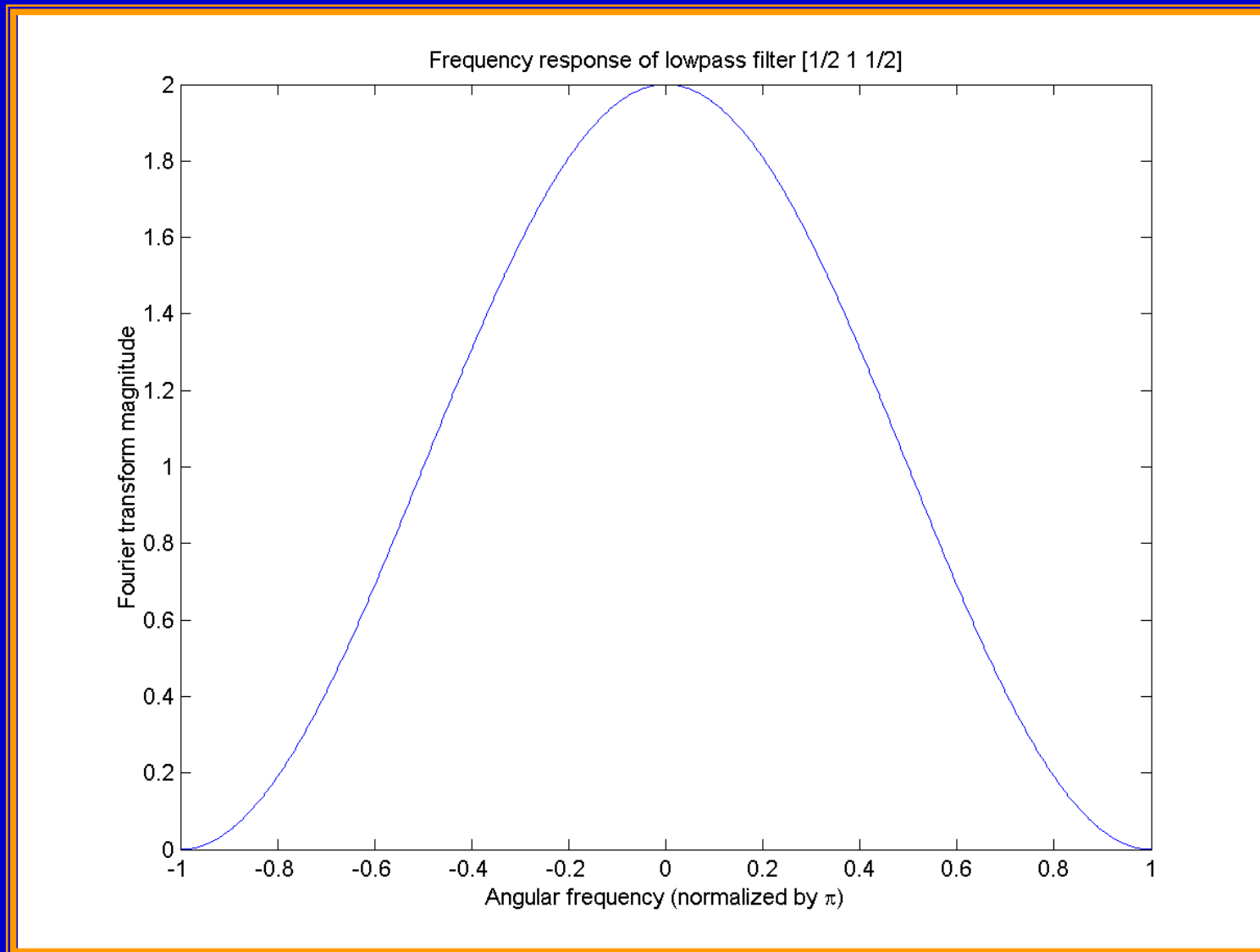
Lowpass filter



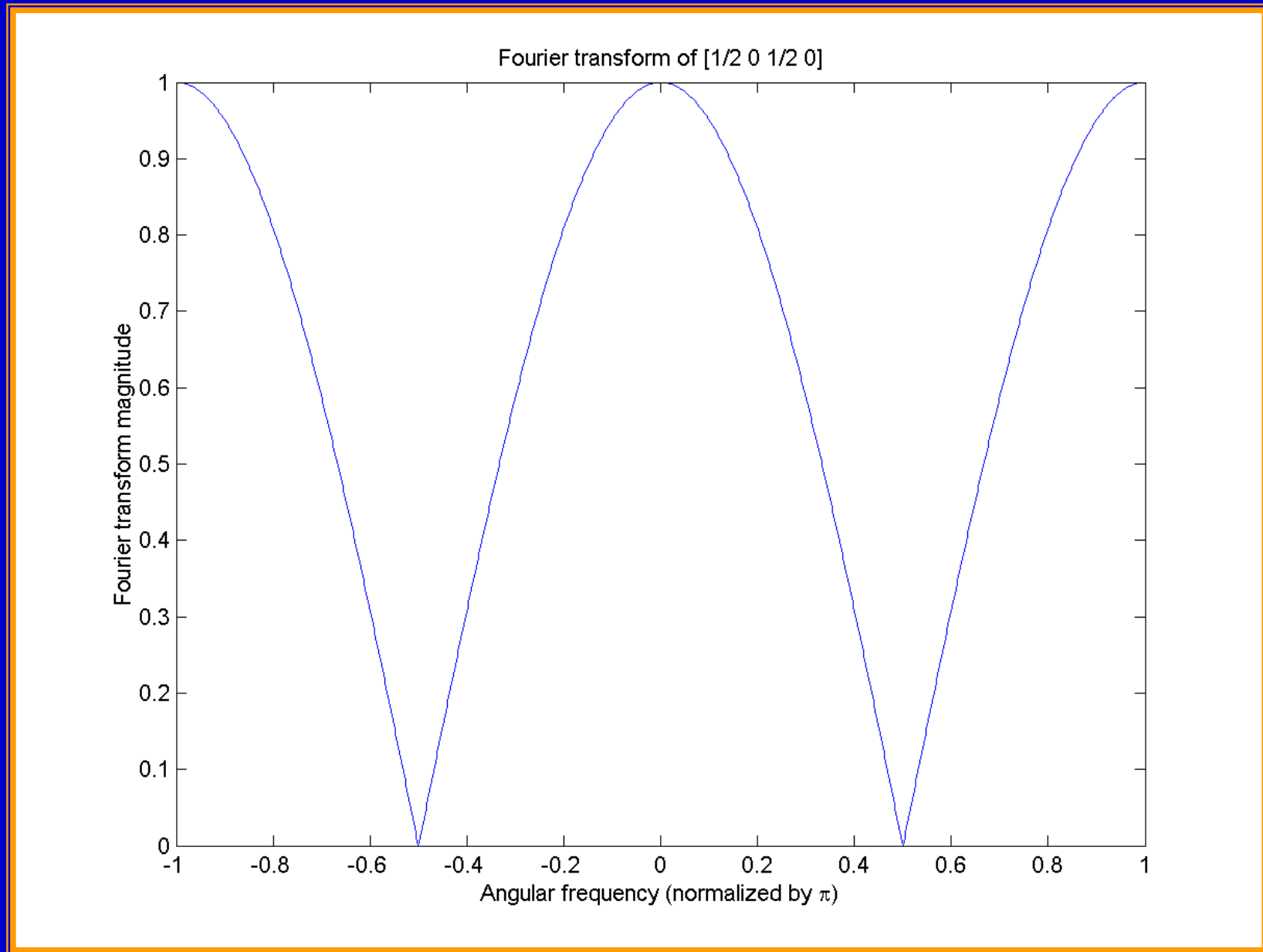
Highpass filter



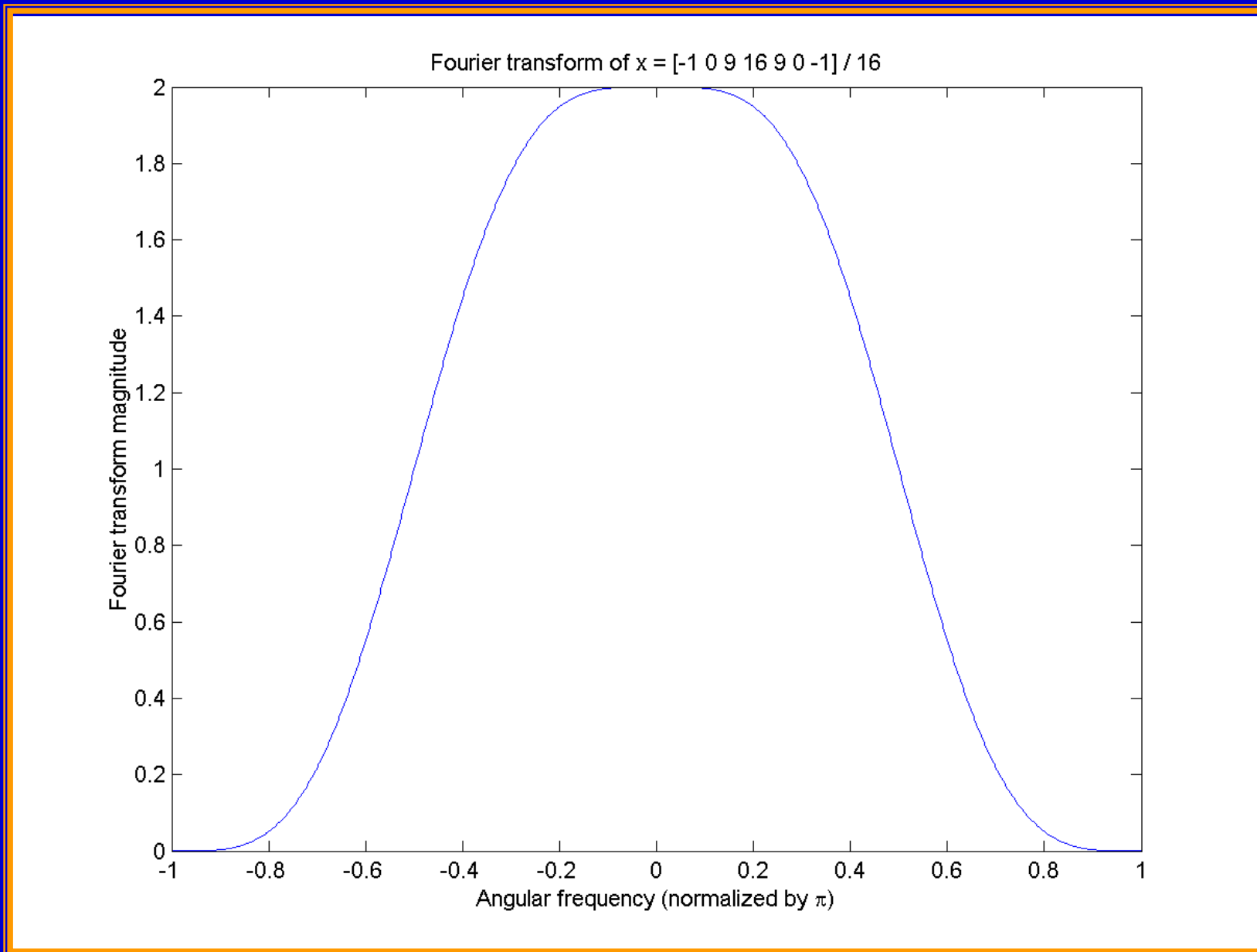
Linear interpolating lowpass filter



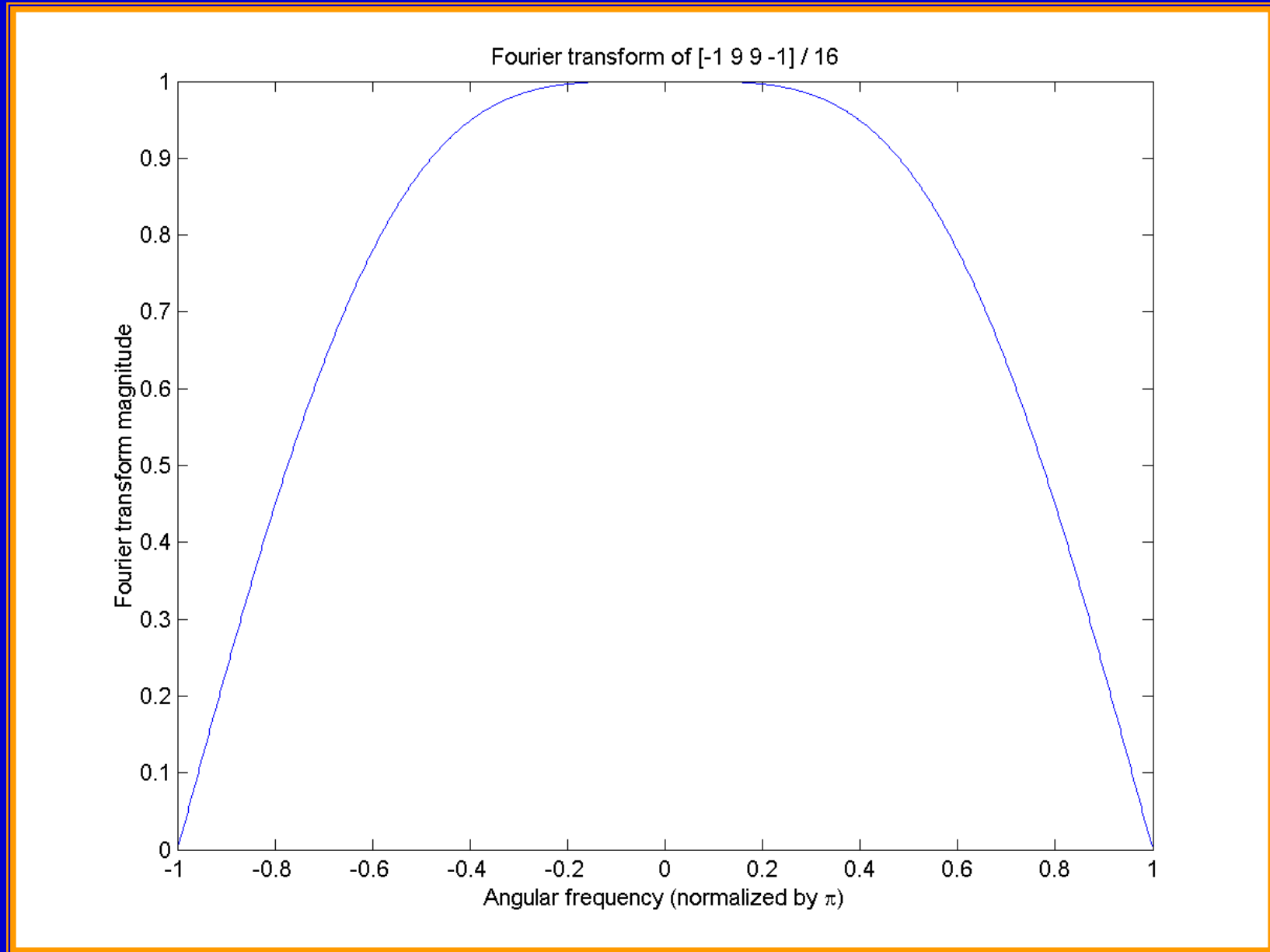
Upsampling



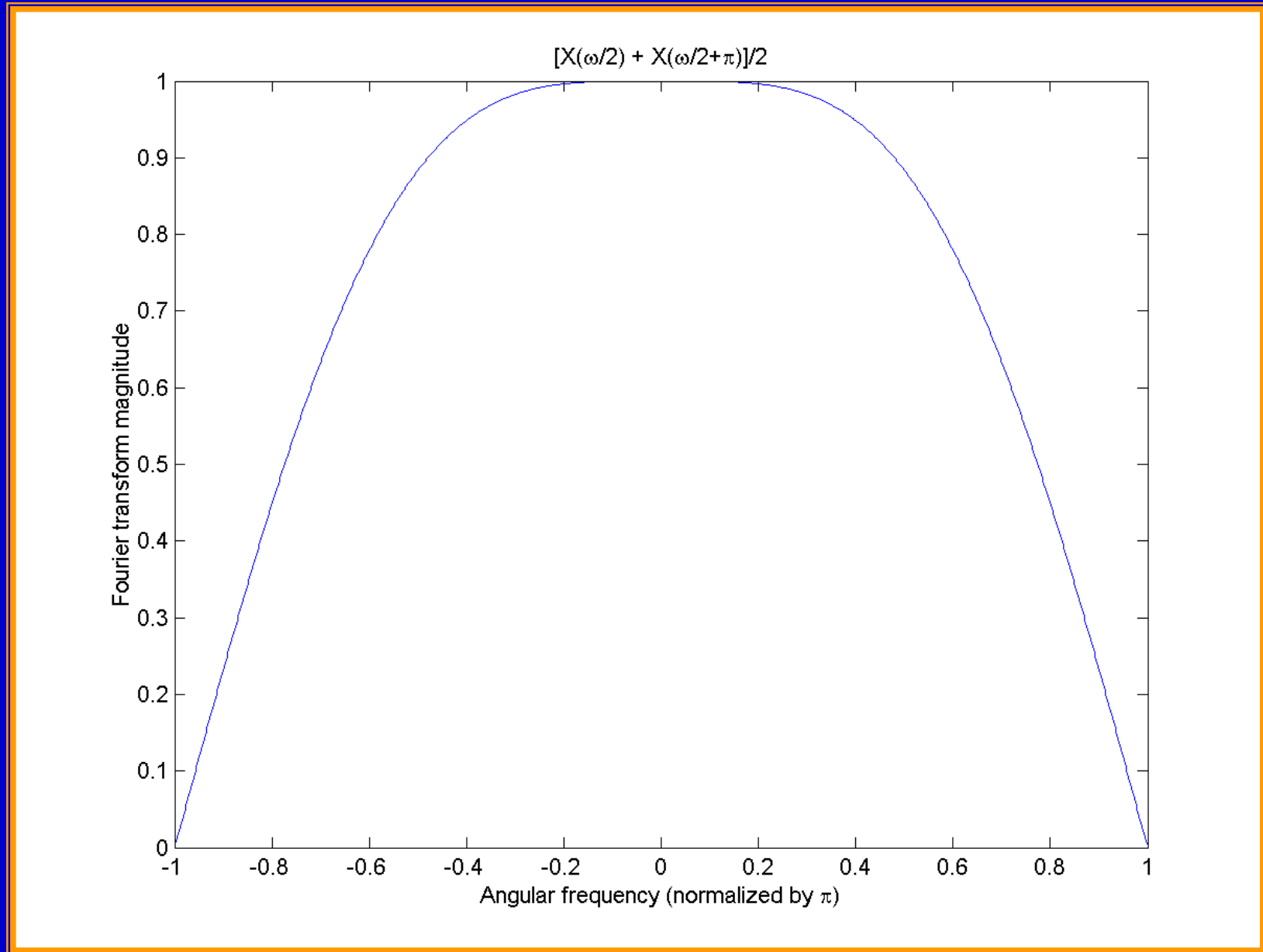
Downsampling



Downsampling



Downsampling



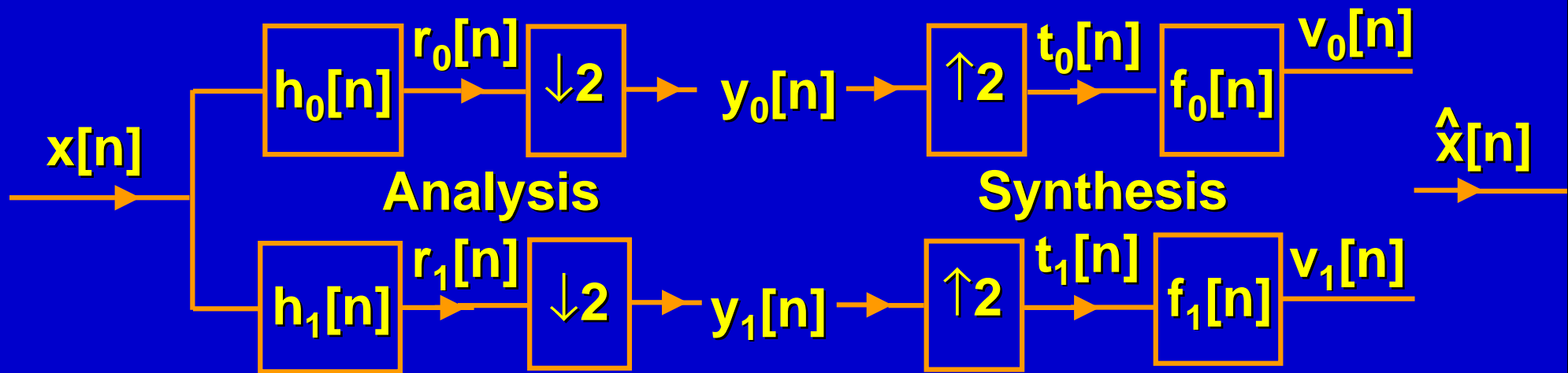
Course 18.327 and 1.130

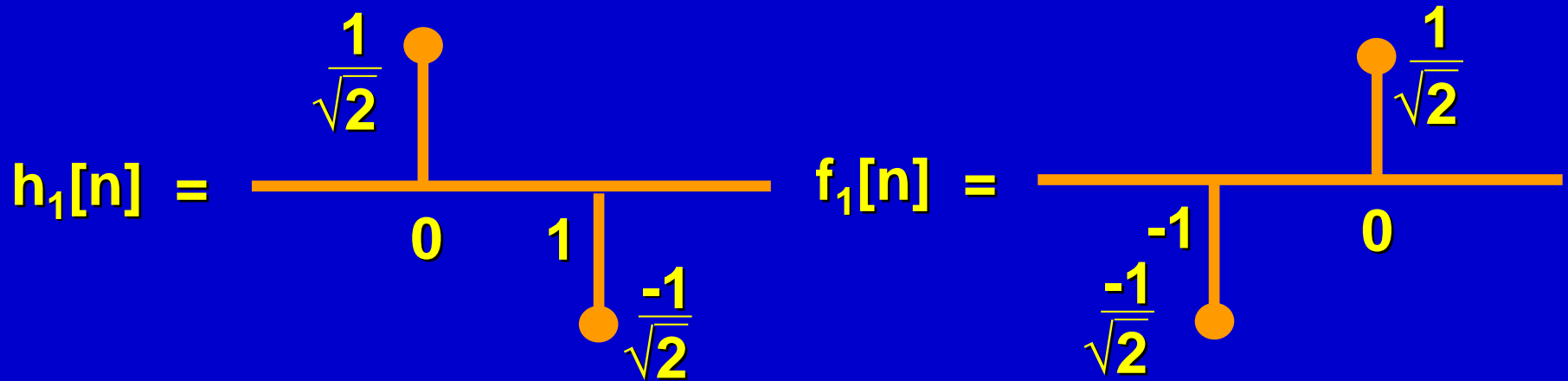
Wavelets and Filter Banks

**Filter Banks: time domain
(Haar example) and frequency domain;
conditions for alias cancellation
and no distortion**

Haar Filter Bank

Simplest (non-trivial) example of a two channel FIR perfect reconstruction filter bank.





Analysis:

$$r_0[n] = \frac{1}{\sqrt{2}} (x[n] + x[n-1])$$

lowpass filter

$$y_0[n] = r_0[2n]$$

downsampler

$$y_0[n] = \frac{1}{\sqrt{2}} (x[2n] + x[2n-1])$$

-----j

Similarly

$$y_1[n] = \frac{1}{\sqrt{2}} (x[2n] - x[2n-1])$$

-----k

Matrix form

$$\begin{bmatrix}
 \text{M} \\
 y_0[0] \\
 y_0[1] \\
 \vdots \\
 \vdots \\
 y_1[0] \\
 y_1[1] \\
 \text{M}
 \end{bmatrix}
 = \frac{1}{\sqrt{2}}
 \begin{bmatrix}
 \text{L} & 1 & 1 & \text{M} & 0 & 0 & \text{L} \\
 \text{L} & 0 & 0 & & 1 & 1 & \text{L} \\
 \hline
 \text{L} & -1 & 1 & & 0 & 0 & \text{L} \\
 \text{L} & 0 & 0 & & -1 & 1 & \text{L} \\
 & & & \text{M} & & &
 \end{bmatrix}
 \begin{bmatrix}
 \vdots \\
 x[-1] \\
 x[0] \\
 x[1] \\
 \\
 x[2] \\
 \vdots \\
 \vdots
 \end{bmatrix}$$

$$\begin{bmatrix}
 y_0 \\
 \hline
 y_1
 \end{bmatrix}
 = \begin{bmatrix}
 \text{L} \\
 \hline
 \text{B}
 \end{bmatrix}
 \mathbf{x}
 \quad \text{-----}$$

Synthesis

$$t_0[n] = \begin{cases} y_0[n/2] & n \text{ even} \\ 0 & n \text{ odd} \end{cases} \quad \text{upsampler}$$

$$v_0[n] = \frac{1}{\sqrt{2}} (t_0[n+1] + t_0[n]) \quad \text{lowpass filter}$$

$$= \begin{cases} \frac{1}{\sqrt{2}} y_0[n/2] & n \text{ even} \\ \frac{1}{\sqrt{2}} y_0[\frac{n+1}{2}] & n \text{ odd} \end{cases}$$

Similarly

$$v_1[n] = \begin{cases} \frac{1}{\sqrt{2}} y_1[n/2] & n \text{ even} \\ \frac{1}{\sqrt{2}} y_1[\frac{n+1}{2}] & n \text{ odd} \end{cases}$$

So, the reconstructed signal is

$$\hat{x}[n] = v_0[n] + v_1[n]$$
$$= \begin{cases} \frac{1}{\sqrt{2}} (y_0[n/2] + y_1[n/2]) & n \text{ even} \\ \frac{1}{\sqrt{2}} (y_0[\frac{n+1}{2}] - y_1[\frac{n+1}{2}]) & n \text{ odd} \end{cases}$$

i.e.

$$\hat{x}[2n-1] = \frac{1}{\sqrt{2}} (y_0[n] - y_1[n]) = x[2n-1]$$

from j and k

$$\hat{x}[2n] = \frac{1}{\sqrt{2}} (y_0[n] + y_1[n]) = x[2n]$$

So $\hat{x}[n] = x[n] \Rightarrow$ Perfect reconstruction!

In general, we will make all filters causal, so we will have

$$\hat{x}[n] = x[n - n_0] \Rightarrow \text{PR with delay}$$

Matrix form

$$\begin{bmatrix} \overset{M}{\hat{x}[-1]} \\ \overset{M}{\hat{x}[0]} \\ \overset{M}{\hat{x}[1]} \\ \overset{M}{\hat{x}[2]} \\ \overset{M}{\phantom{\hat{x}[2]}} \end{bmatrix} = \frac{1}{\sqrt{2}} \left[\begin{array}{cc|cc} \overset{M}{1} & \overset{M}{0} & \overset{M}{-1} & \overset{M}{0} \\ \overset{M}{1} & \overset{M}{0} & \overset{M}{1} & \overset{M}{1} \\ \overset{L}{0} & \overset{L}{1} & \overset{L}{0} & \overset{L}{-1} \\ \overset{L}{0} & \overset{L}{1} & \overset{L}{-1} & \overset{L}{1} \\ \overset{M}{} & \overset{M}{} & \overset{M}{} & \overset{M}{} \end{array} \right] \begin{bmatrix} \overset{M}{y_0[0]} \\ \overset{M}{y_0[1]} \\ \overset{M}{} \\ \overset{M}{y_1[0]} \\ \overset{M}{y_1[1]} \\ \overset{M}{} \end{bmatrix}$$

$$\overset{M}{\hat{x}} = \left[\begin{array}{c|c} \overset{L}{L^T} & \overset{L}{B^T} \end{array} \right] \begin{bmatrix} \overset{M}{y_0} \\ \hline \overset{M}{y_1} \end{bmatrix} \quad \text{-----} \overset{M}{m}$$

Perfect reconstruction means that the synthesis bank is the inverse of the analysis bank.

$$\hat{\mathbf{x}} = \mathbf{x} \Rightarrow \left[\begin{array}{c|c} \mathbf{L}^T & \mathbf{B}^T \\ \hline 1 & 2 & 3 \end{array} \right] \left[\begin{array}{c} \mathbf{L} \\ \hline \mathbf{B} \\ 1 & 2 & 3 \end{array} \right] = \mathbf{I}$$

\mathbf{W}^{-1} \mathbf{W}

(Wavelet transform matrix)

In the Haar example, we have the special case

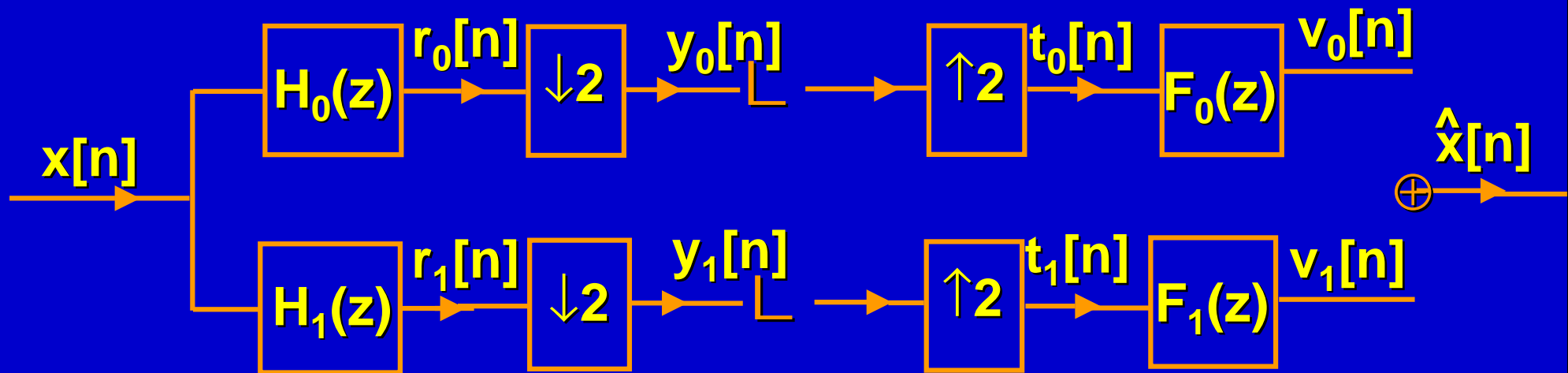
$$\mathbf{W}^{-1} = \mathbf{W}^T \rightarrow \text{orthogonal matrix}$$

So we have an orthogonal filter bank, where
 Synthesis bank = Transpose of Analysis bank

$$\begin{array}{lcl} f_0[n] & = & h_0[-n] \\ f_1[n] & = & h_1[-n] \end{array}$$

Perfect Reconstruction Filter Banks

General two-channel filter bank



z-transform definition:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

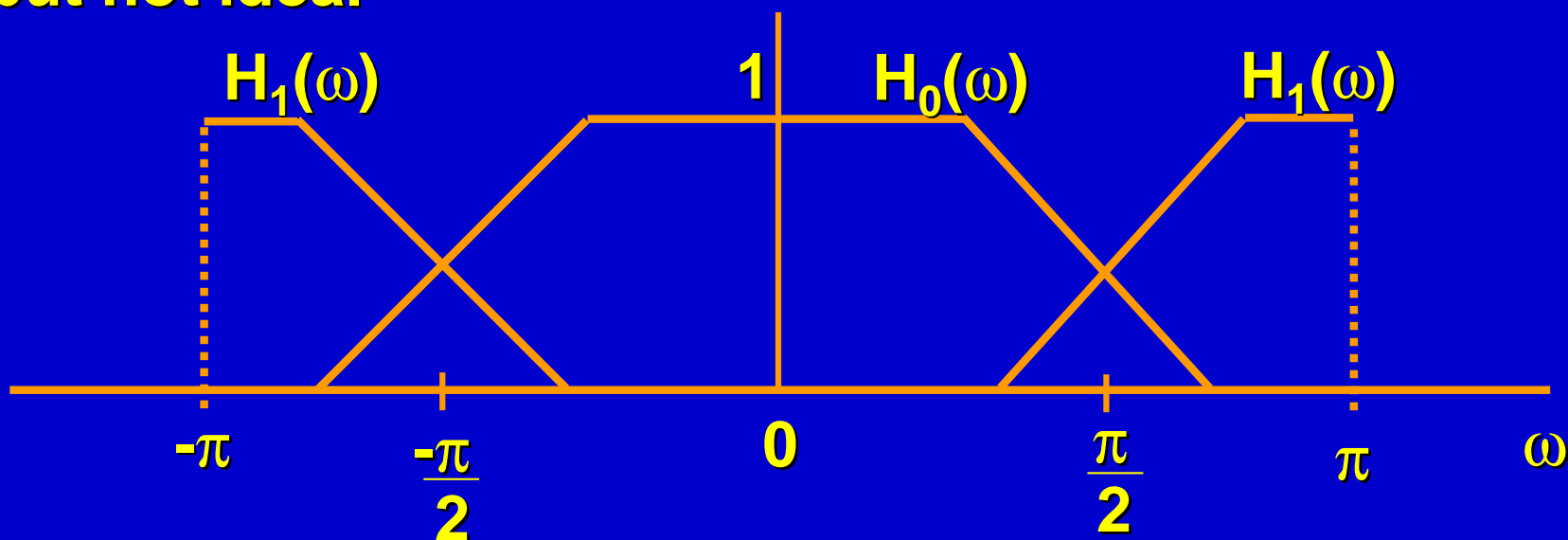
Put $z = e^{j\omega}$ to get DTFT

Perfect reconstruction requirement:

$$\hat{x}[n] = x[n - l] \quad (l \text{ time delays})$$

$$\hat{X}(z) = z^{-l} X(z)$$

$H_0(z)$ and $H_1(z)$ are normally lowpass and highpass, but not ideal



⇒ Downsampling operation in each channel can produce aliasing

Let's see why:

Lowpass channel has

$$\begin{aligned} Y_0(z) &= \frac{1}{2}\{R_0(z^{1/2}) + R_0(-z^{1/2})\} \quad (\text{downsampling}) \\ &= \frac{1}{2}\{H_0(z^{1/2})X(z^{1/2}) + H_0(-z^{1/2})X(-z^{1/2})\} \end{aligned}$$

In frequency domain:

$$X(z) \rightarrow X(\omega) \quad \text{or } X(e^{i\omega})$$

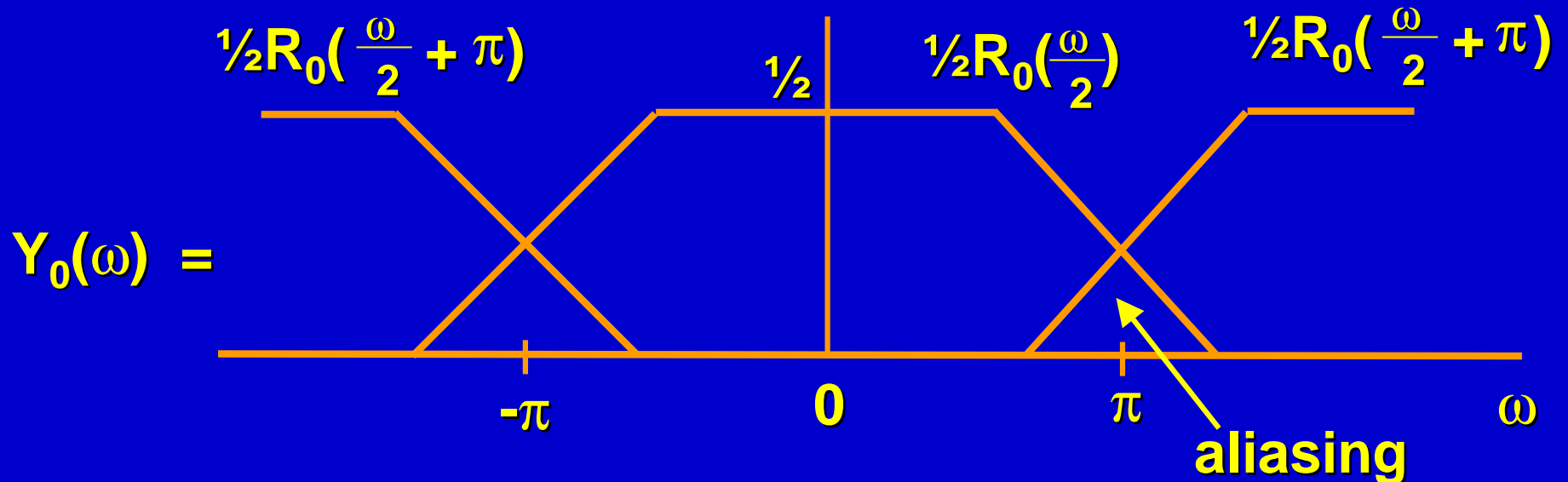
$$X(-z) \rightarrow X(\omega + \pi)$$

$$X(z^{1/2}) \rightarrow X\left(\frac{\omega}{2}\right)$$

$$Y_0(\omega) = \frac{1}{2}\left\{H_0\left(\frac{\omega}{2}\right)X\left(\frac{\omega}{2}\right) + H_0\left(\frac{\omega}{2} + \pi\right)X\left(\frac{\omega}{2} + \pi\right)\right\}$$

Suppose $X(\omega) = 1$ (input has all frequencies)

Then $R_0(\omega) = H_0(\omega)$, so that after downsampling we have



Goal is to design $F_0(z)$ and $F_1(z)$ so that the overall system is just a simple delay - with no aliasing term:

$$V_0(z) + V_1(z) = z^{-1} X(z)$$

$$\begin{aligned}
V_0(z) &= F_0(z) T_0(z) \\
&= F_0(z) Y_0(z^2) && \text{(upsampling)} \\
&= \frac{1}{2} F_0(z) \{ H_0(z) X(z) + H_0(-z) X(-z) \} \\
V_1(z) &= \frac{1}{2} F_1(z) \{ H_1(z) X(z) + H_1(-z) X(-z) \}
\end{aligned}$$

So we want

$$\begin{aligned}
&\frac{1}{2} \{ F_0(z) H_0(z) + F_1(z) H_1(z) \} X(z) \\
&\quad + \\
&\frac{1}{2} \{ F_0(z) H_0(-z) + F_1(z) H_1(-z) \} X(-z) \\
&\hspace{15em} = z^{-1} X(z)
\end{aligned}$$

Compare terms in $X(z)$ and $X(-z)$:

- 1) Condition for no distortion (terms in $X(z)$ amount to a delay)**

$$F_0(z) H_0(z) + F_1(z) H_1(z) = 2z^{-j} \quad \text{-----}j$$

- 2) Condition for alias cancellation (no term in $X(-z)$)**

$$F_0(z) H_0(-z) + F_1(z) H_1(-z) = 0 \quad \text{-----}k$$

To satisfy alias cancellation condition, choose

$$\begin{aligned} F_0(z) &= H_1(-z) \\ F_1(z) &= -H_0(-z) \end{aligned} \quad \text{-----}l$$

What happens in the time domain?

$$\begin{aligned} F_0(z) &= H_1(-z) \\ &= \sum_n h_1[n] (-z)^{-n} \end{aligned}$$

$$= \sum_n (-1)^n h_1[n] z^{-n}$$

$$F_0(\omega) = H_1(\omega + \pi)$$

So the filter coefficients are

$$f_0[n] = (-1)^n h_1[n]$$

$$f_1[n] = (-1)^{n+1} h_0[n]$$

alternating signs
rule

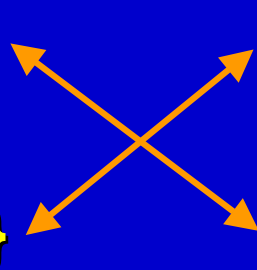
Example

$$h_0[n] = \{a_0, a_1, a_2\}$$

$$h_1[n] = \{b_0, b_1, b_2\}$$

$$f_0[n] = \{b_0, -b_1, b_2\}$$

$$f_1[n] = \{-a_0, a_1, -a_2\}$$



Product Filter

Define

$$P_0(z) = F_0(z) H_0(z) \text{ -----}m$$

Substitute $F_1(z) = -H_0(-z)$, $H_1(z) = F_0(-z)$
in the zero distortion condition (Equation j)

$$F_0(z) H_0(z) - F_0(-z) H_0(-z) = 2z^{-l}$$

i.e. $P_0(z) - P_0(-z) = 2z^{-l} \text{ -----}n$

Note: l must be odd since LHS is an odd function.

Normalized Product Filter

Define

$$P(z) = z^l P_0(z) \quad \text{-----} \circ$$

$$P(-z) = -z^l P_0(-z) \quad \text{since } l \text{ is odd}$$

So we can rewrite Equation n as

$$z^{-l} P(z) + z^{-l} P(-z) = 2z^{-l}$$

$$\text{i.e.} \quad P(z) + P(-z) = 2 \quad \text{-----} \rho$$

This is the condition on the normalized product filter for Perfect Reconstruction.

Design Process

1. Design $P(z)$ to satisfy Equation ρ . This gives $P_0(z)$. Note: $P(z)$ is designed to be lowpass.
2. Factor $P_0(z)$ into $F_0(z) H_0(z)$. Use Equations 1 to find $H_1(z)$ and $F_1(z)$.

Note: Equation ρ requires all even powers of z (except z^0) to be zero:

$$\sum_n p[n]z^{-n} + \sum_n p[n](-z)^{-n} = 2$$

$$\Rightarrow p[n] = \begin{cases} 1 & ; \quad n = 0 \\ 0 & ; \quad \text{all even } n \text{ (} n \neq 0 \text{)} \end{cases}$$

For odd n , $p[n]$ and $-p[n]$ cancel.

The odd coefficients, $p[n]$, are free to be designed according to additional criteria.

Example: Haar filter bank

$$H_0(z) = \frac{1}{\sqrt{2}} (1 + z^{-1}) \quad H_1(z) = \frac{1}{\sqrt{2}} (1 - z^{-1})$$

$$F_0(z) = H_1(-z) = \frac{1}{\sqrt{2}} (1 + z^{-1})$$

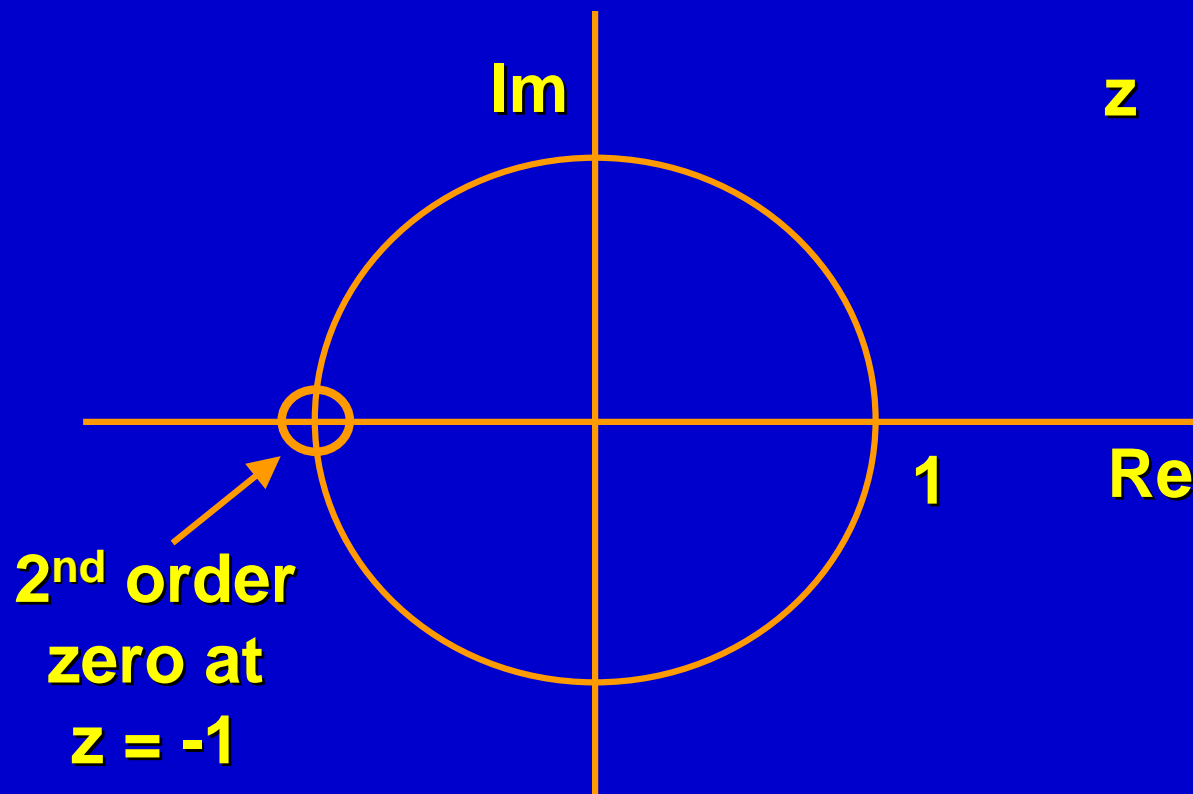
$$F_1(z) = -H_0(-z) = \frac{-1}{\sqrt{2}} (1 - z^{-1})$$

$$P_0(z) = F_0(z) H_0(z) = \frac{1}{2} (1 + z^{-1})^2$$

So the Perfect Reconstruction requirement is

$$\begin{aligned} P_0(z) - P_0(-z) &= \frac{1}{2}(1 + 2z^{-1} + z^{-2}) - \frac{1}{2}(1 - 2z^{-1} + z^{-2}) \\ &= 2z^{-1} \quad \Rightarrow \quad |z| = 1 \end{aligned}$$

$$P(z) = z^l P_0(z) = \frac{1}{2}(1 + z)(1 + z^{-1})$$



Zeros of $P(z)$:

$$1 + z = 0$$

$$1 + z^{-1} = 0$$

Course 18.327 and 1.130

Wavelets and Filter Banks

Filter Banks (contd.): perfect reconstruction; halfband filters and possible factorizations.

Product Filter

Example: Product filter of degree 6

$$P_0(z) = \frac{1}{16} (-1 + 9z^{-2} + 16z^{-3} + 9z^{-4} - z^{-6})$$

$$P_0(z) - P_0(-z) = 2z^{-3}$$

⇒ Expect perfect reconstruction with a 3 sample delay

Centered form:

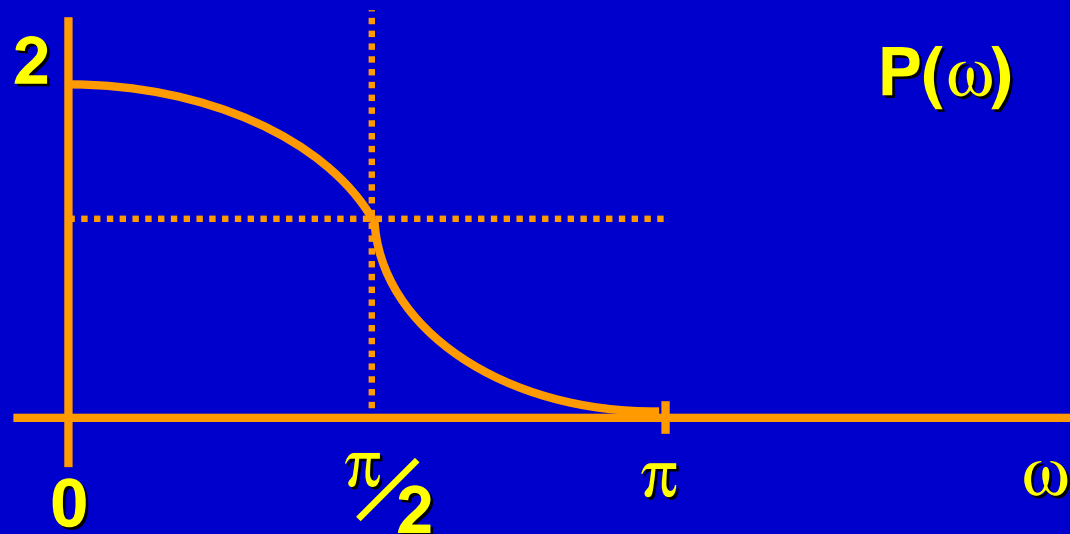
$$P(z) = z^3 P_0(z) = \frac{1}{16} (-z^3 + 9z + 16 + 9z^{-1} - z^{-3})$$

$$P(z) + P(-z) = 2 \quad \text{i.e. even part of } P(z) = \text{const}$$

In the frequency domain:

$$P(\omega) + P(\omega + \pi) = 2$$

Halfband Condition



**Note antisymmetry
about $\omega = \pi/2$**

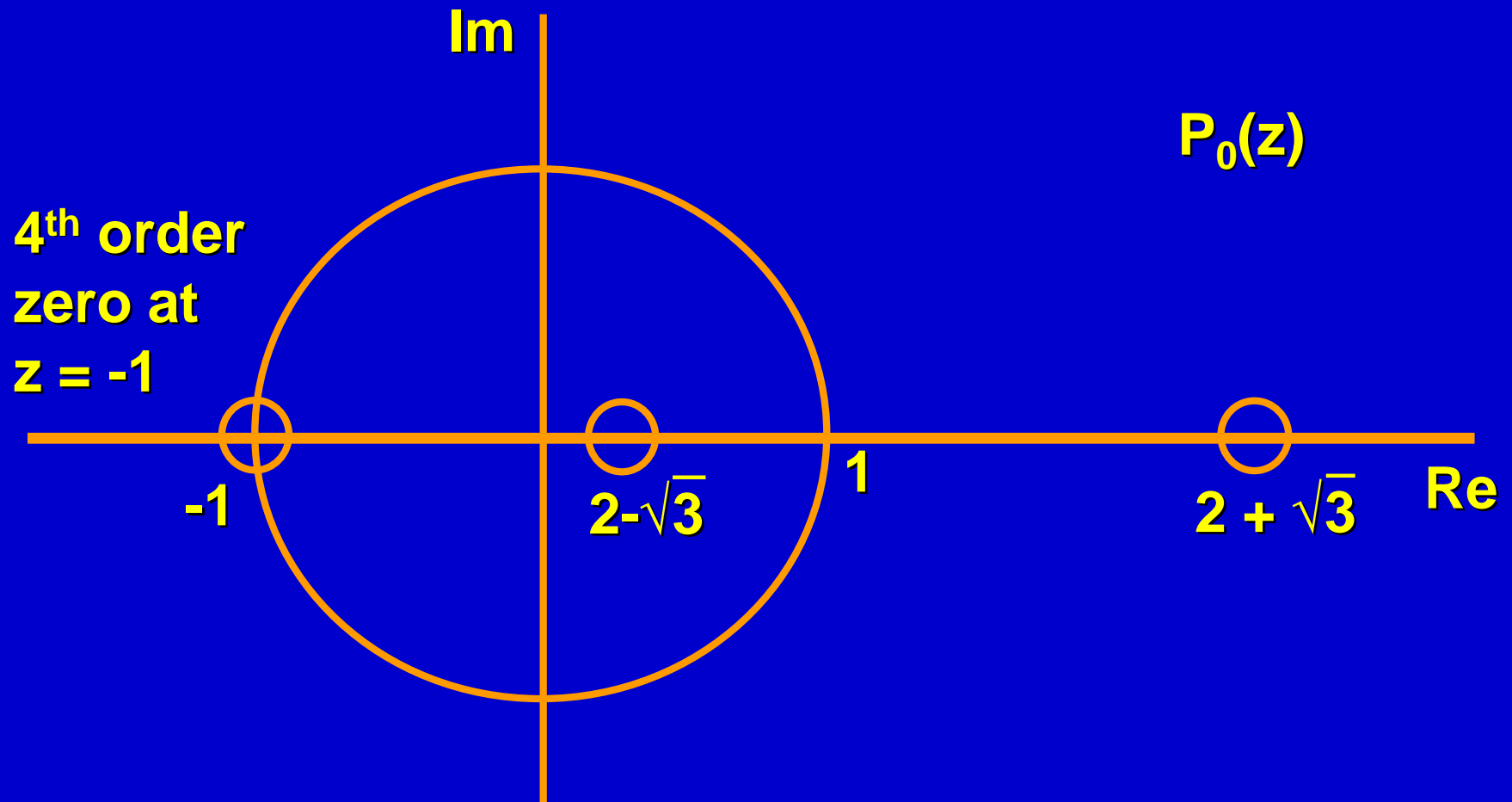
$P(\omega)$ is said to be a halfband filter.

How do we factor $P_0(z)$ into $H_0(z) F_0(z)$?

$$\begin{aligned}
 P_0(z) &= 1/16(1 + z^{-1})^4(-1 + 4z^{-1} - z^{-2}) \\
 &= -1/16(1 + z^{-1})^4(2 + \sqrt{3} - z^{-1})(2 - \sqrt{3} - z^{-1})
 \end{aligned}$$

So $P_0(z)$ has zeros at
 $z = -1$ (4th order)
 $z = 2 \pm \sqrt{3}$

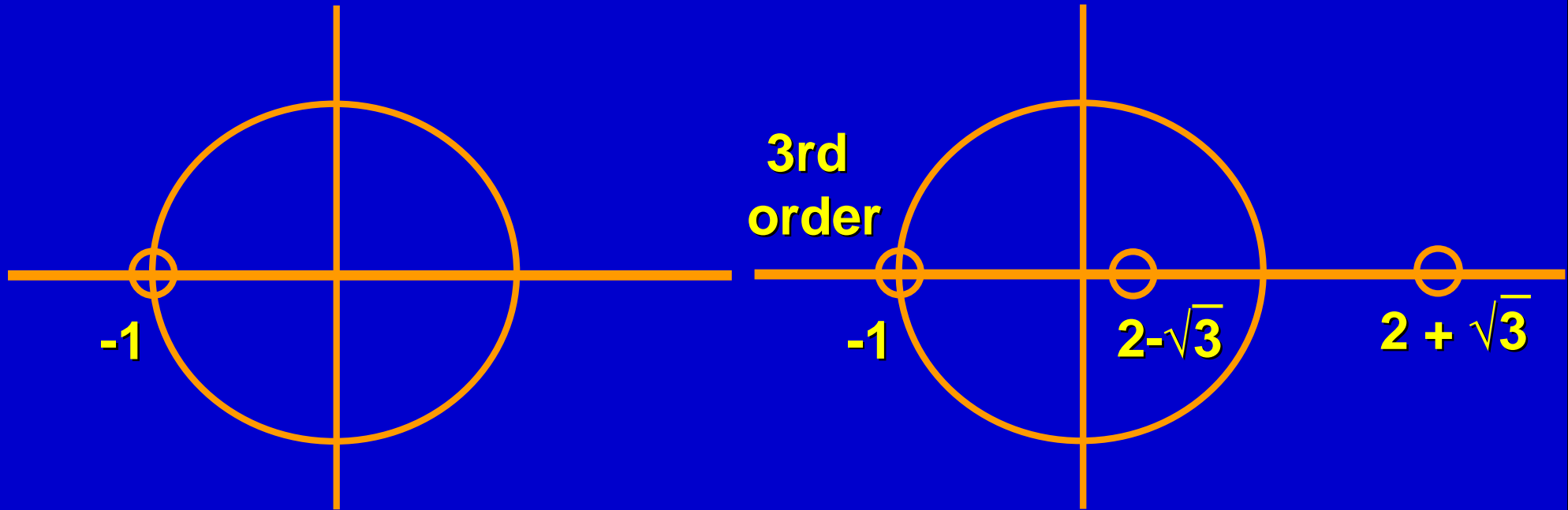
Note: $2 + \sqrt{3} = \frac{1}{2 - \sqrt{3}}$



Some possible factorizations

	$H_0(z)$ (or $F_0(z)$)	$F_0(z)$ (or $H_0(z)$)
(a)	1	$-1/16(1 + z^{-1})^4(2 + \sqrt{3} - z^{-1})(2 - \sqrt{3} - z^{-1})$
(b)	$1/2(1 + z^{-1})$	$-1/8(1 + z^{-1})^3(2 + \sqrt{3} - z^{-1})(2 - \sqrt{3} - z^{-1})$
(c)	$1/4(1 + z^{-1})^2$	$-1/4(1 + z^{-1})^2(2 + \sqrt{3} - z^{-1})(2 - \sqrt{3} - z^{-1})$
(d)	$1/2(1 + z^{-1})(2 + \sqrt{3} - z^{-1})$	$-1/8(1 + z^{-1})^3(2 - \sqrt{3} - z^{-1})$
(e)	$1/8(1 + z^{-1})^3$	$-1/2(1 + z^{-1})(2 + \sqrt{3} - z^{-1})(2 - \sqrt{3} - z^{-1})$
(f)	$\frac{(\sqrt{3}-1)}{4\sqrt{2}}(1 + z^{-1})^2(2 + \sqrt{3} - z^{-1})$	$\frac{-\sqrt{2}}{4(\sqrt{3}-1)}(1 + z^{-1})^2(2 - \sqrt{3} - z^{-1})$
(g)	$1/16(1 + z^{-1})^4$	$-(2 + \sqrt{3} - z^{-1})(2 - \sqrt{3} - z^{-1})$

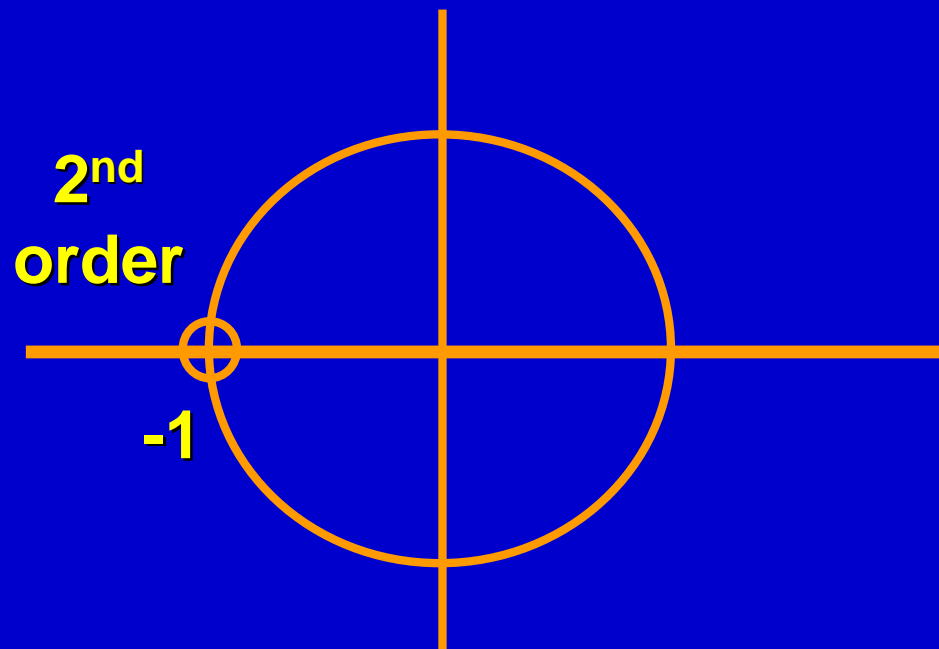
Case (b) -- Symmetric filters (linear phase)



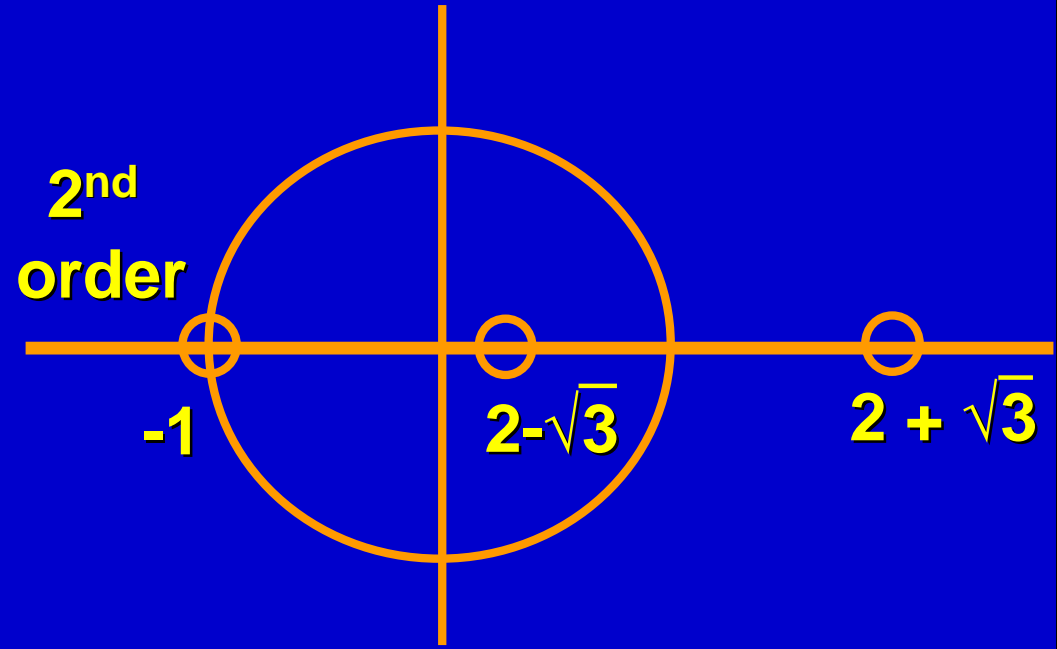
filter length = 2
 $\frac{1}{2}\{1, 1\}$

filter length = 6
 $\frac{1}{8}\{-1, 1, 8, 8, 1, -1\}$

Case (c) -- Symmetric filters (linear phase)

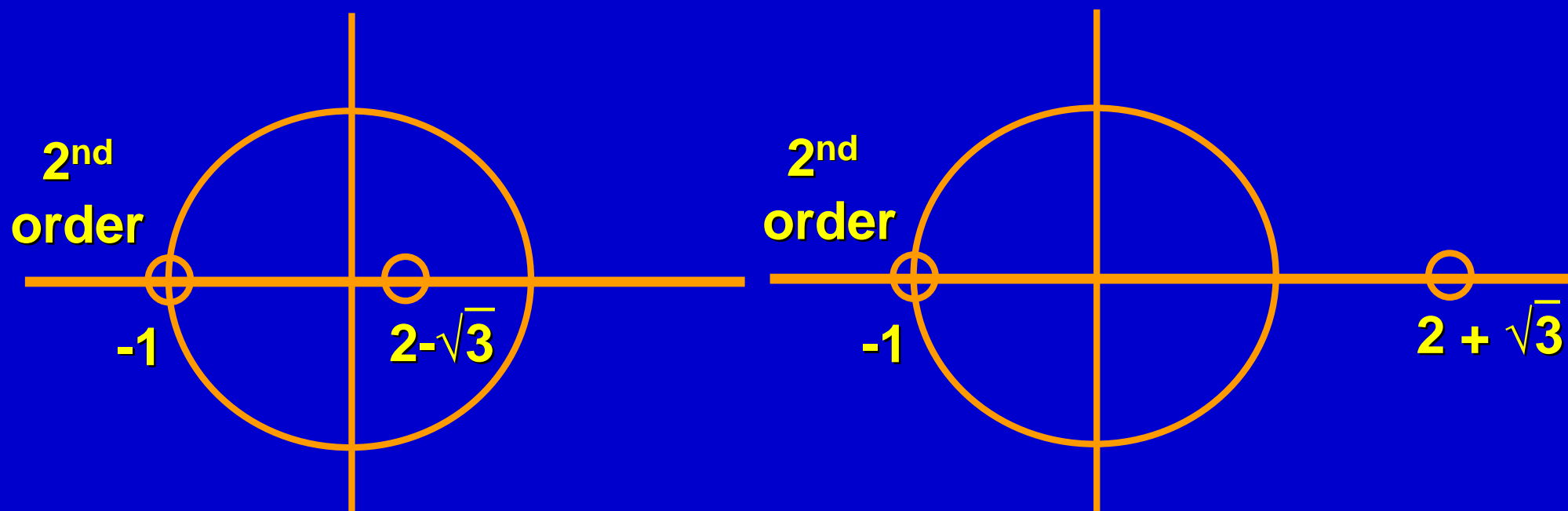


filter length = 3
 $\frac{1}{4} \{ 1, 2, 1 \}$



filter length = 5
 $\frac{1}{4} \{ -1, 2, 6, 2, -1 \}$

Case (f) -- Orthogonal filters (minimum phase/maximum phase)



filter length = 4

$$\frac{1}{4\sqrt{2}} \begin{matrix} \infty \\ \nearrow \\ 1+\sqrt{3}, 3+\sqrt{3}, 3-\sqrt{3}, 1-\sqrt{3} \\ \searrow \\ 0 \end{matrix}$$

filter length = 4

$$\frac{1}{4\sqrt{2}} \begin{matrix} \infty \\ \nearrow \\ 1-\sqrt{3}, 3-\sqrt{3}, 3+\sqrt{3}, 1+\sqrt{3} \\ \searrow \\ 0 \end{matrix}$$

Note that, in this case, one filter is the flip (transpose)

of the other: $f_0[n] = h_0[3 - n]$

$$F_0(z) = z^{-3} H_0(z^{-1})$$

General form of product filter (to be derived later):

$$P(z) = 2 \left(\frac{1+z}{2} \right)^p \left(\frac{1+z^{-1}}{2} \right)^p \sum_{k=0}^{p-1} \binom{p+k-1}{k} \left(\frac{1-z}{2} \right)^k \left(\frac{1-z^{-1}}{2} \right)^k$$

$$P_0(z) = z^{-(2p-1)} P(z)$$

$$= (1+z^{-1})^{2p} \frac{1}{2^{2p-1}} \sum_{k=0}^{p-1} \binom{p+k-1}{k} (-1)^k z^{-(p-1)+k} \left(\frac{1-z^{-1}}{2} \right)^{2k}$$

**Binomial
(spline)
filter**

Q(z)

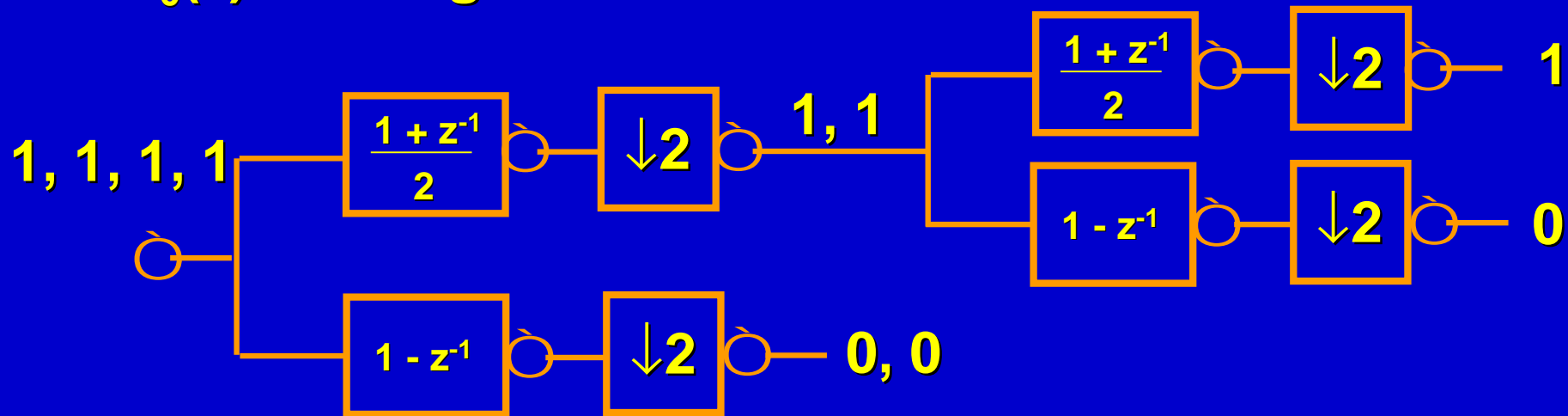
**Cancels all odd powers
except $z^{-(2p-1)}$**

$P_0(z)$ has $2p$ zeros at π (important for stability of iterated filter bank.)

Q(z) factor is needed to ensure perfect reconstruction.

$p = 1$

$P_0(z)$ has degree 2 \rightarrow leads to Haar filter bank.



$$F_0(z) = 1 + z^{-1}, \quad H_0(z) = \frac{1 + z^{-1}}{2}$$

Synthesis lowpass filter has 1 zero at π

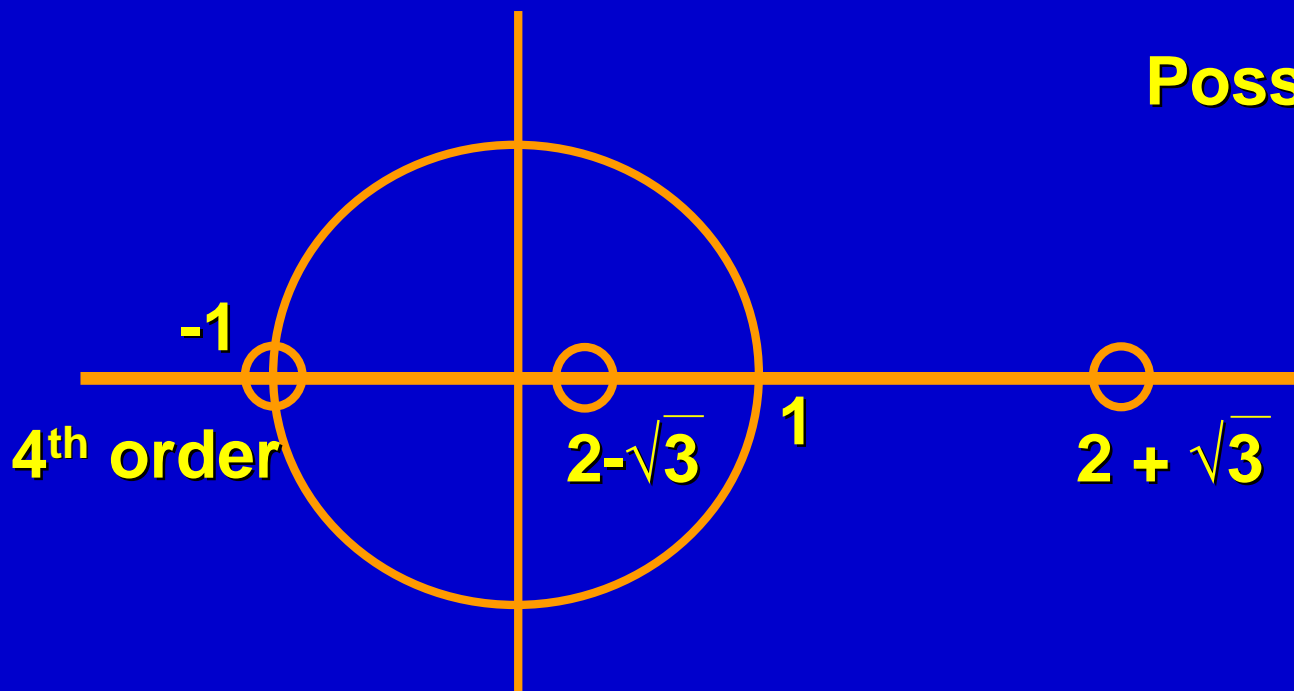
\rightarrow Leads to cancellation of constant signals in analysis
highpass channel.

Additional zeros at π would lead to cancellation of
higher order polynomials.

$$p = 2$$

$P_0(z)$ has degree $4p - 2 = 6$

$$\begin{aligned} P_0(z) &= (1 + z^{-1})^4 \frac{1}{8} \left\{ \binom{1}{0} z^{-1} - \binom{2}{1} \left(\frac{1-z^{-1}}{2}\right)^2 \right\} \\ &= \frac{1}{16} (1 + z^{-1})^4 (-1 + 4z^{-1} - z^{-2}) \\ &= \frac{1}{16} \{-1 + 9z^{-2} + 16z^{-3} + 9z^{-4} - z^{-6}\} \end{aligned}$$



Possible factorizations

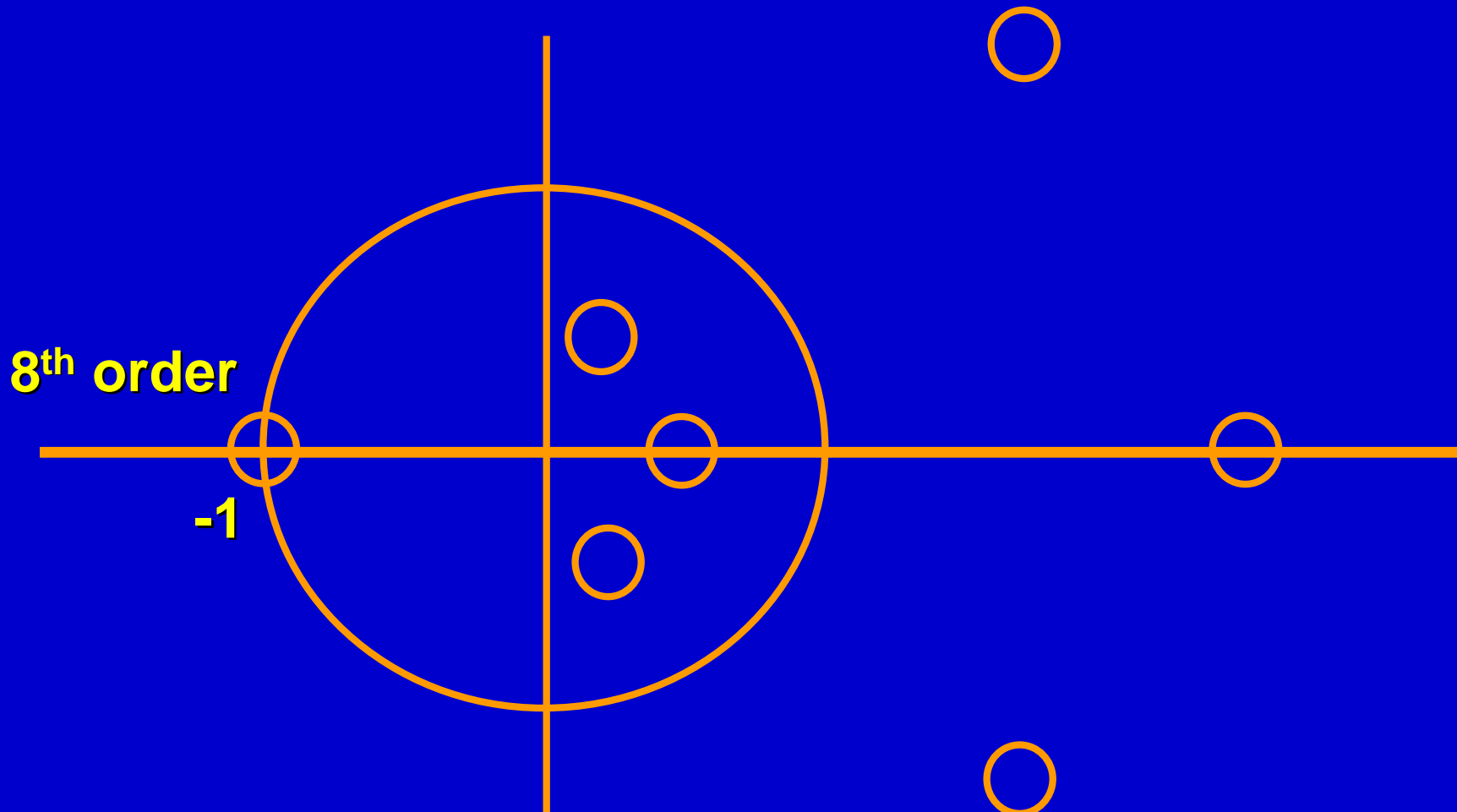
1/8 trivial

2/6 $\begin{matrix} \circ \\ \searrow \\ \infty \end{matrix}$ linear phase

4/4 orthogonal
(Daubechies-4)

$$p = 4$$

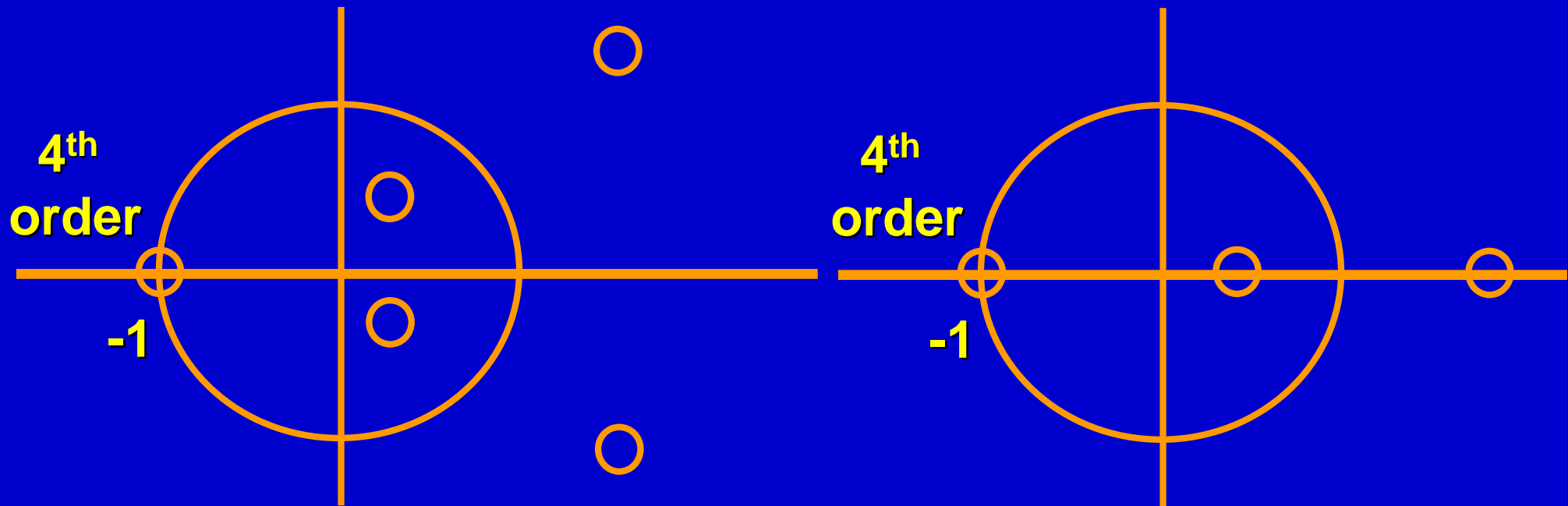
$$P_0(z) \text{ has degree } 4p - 2 = 14$$



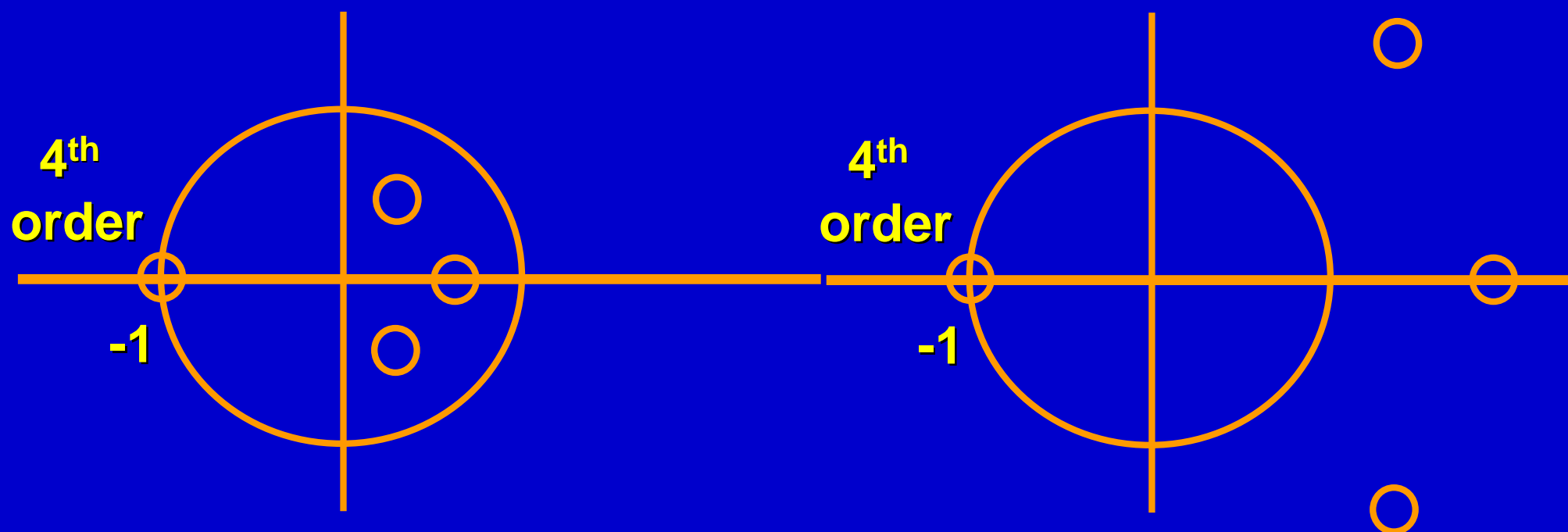
Common factorizations ($p = 4$):

(a) $9/7$

Known in Matlab
as `bior4.4`



(b) 8/8 (Daubechies 8) -- Known in Matlab as db4

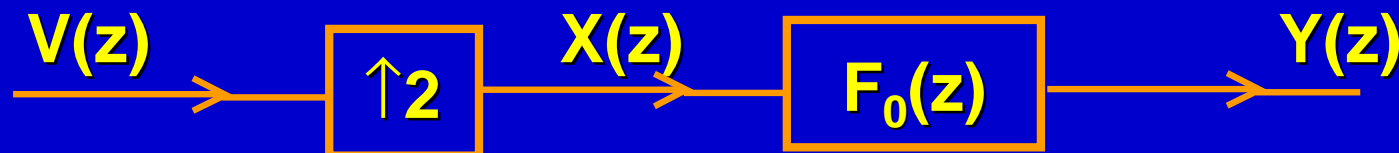


Why choose a particular factorization?

Consider the example with $p = 2$:

i. One of the factors is halfband

The trivial $1/8$ factorization is generally not desirable, since each factor should have at least one zero at π . However, the fact that $F_0(z)$ is halfband is interesting in itself.



Let $F_0(z)$ be centered, for convenience. Then

$$F_0(z) = 1 + \text{odd powers of } z$$

Now

$$X(z) = V(z^2) = \text{even powers of } z \text{ only}$$

So

$$Y(z) = F_0(z) X(z)$$

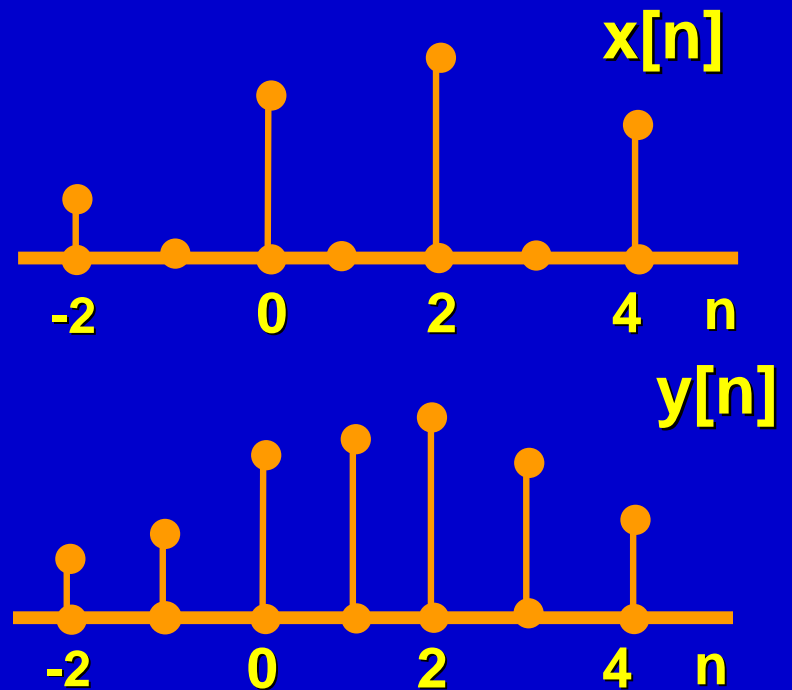
$$= X(z) + \text{odd powers}$$

$$y[n] = \sum_{k \text{ even}} x[n-k] \quad ; \quad n \text{ even}$$

$$\sum_{k \text{ odd}} f_0[k] x[n-k] \quad ; \quad n \text{ odd}$$

$\Rightarrow f_0[n]$ is an interpolating filter

Another example: $f_0[n] = \frac{\sin(\frac{\pi}{2})n}{\pi n}$
(ideal bandlimited
interpolating filter)



ii. Linear phase factorization e.g. 2/6, 5/3

Symmetric (or antisymmetric) filters are desirable for many applications, such as image processing. All frequencies in the signal are delayed by the same amount i.e. there is no phase distortion.

$$h[n] \text{ linear phase} \Rightarrow A(\omega)e^{-i(\omega\alpha + \theta)}$$

Diagram illustrating the components of the linear phase factorization:

- $A(\omega)$ is real
- $e^{-i(\omega\alpha + \theta)}$ delays all frequencies by α samples
- θ is 0 if symmetric, $\frac{\pi}{2}$ if antisymmetric

Linear phase may not necessarily be the best choice for audio applications due to preringing effects.

iii. Orthogonal factorization

This leads to a minimum phase filter and a maximum phase filter, which may be a better choice for applications such as audio. The orthogonal factorization leads to the Daubechies family of wavelets – a particularly neat and interesting case. 4/4 factorization:

$$\begin{aligned} H_0(z) &= \frac{\sqrt{3}-1}{4\sqrt{2}} (1+z^{-1})^2 [(2+\sqrt{3})-z^{-1}] \\ &= \frac{1}{4\sqrt{2}} \{(1+\sqrt{3}) + (3+\sqrt{3})z^{-1} + (3-\sqrt{3})z^{-2} + (1-\sqrt{3})z^{-3}\} \end{aligned}$$

$$\begin{aligned} F_0(z) &= \frac{-\sqrt{2}}{4(\sqrt{3}-1)} (1+z^{-1})^2 [(2-\sqrt{3})-z^{-1}] \\ &= \frac{\sqrt{3}-1}{4\sqrt{2}} z^{-3} (1+z^2) [(2+\sqrt{3})-z] \\ &= z^{-3} H_0(z^{-1}) \end{aligned}$$

$$P(z) = z^l P_0(z)$$

$$= H_0(z) H_0(z^{-1})$$

From alias cancellation condition:

$$H_1(z) = F_0(-z) = -z^{-3} H_0(-z^{-1})$$

$$F_1(z) = -H_0(-z) = z^{-3} H_1(z^{-1})$$

Special Case: Orthogonal Filter Banks

Choose $H_1(z)$ so that

$$H_1(z) = -z^{-N} H_0(-z^{-1})$$

N odd

Time domain

$$h_1[n] = (-1)^n h_0[N - n]$$

$$F_0(z) = H_1(-z) = z^{-N} H_0(z^{-1})$$

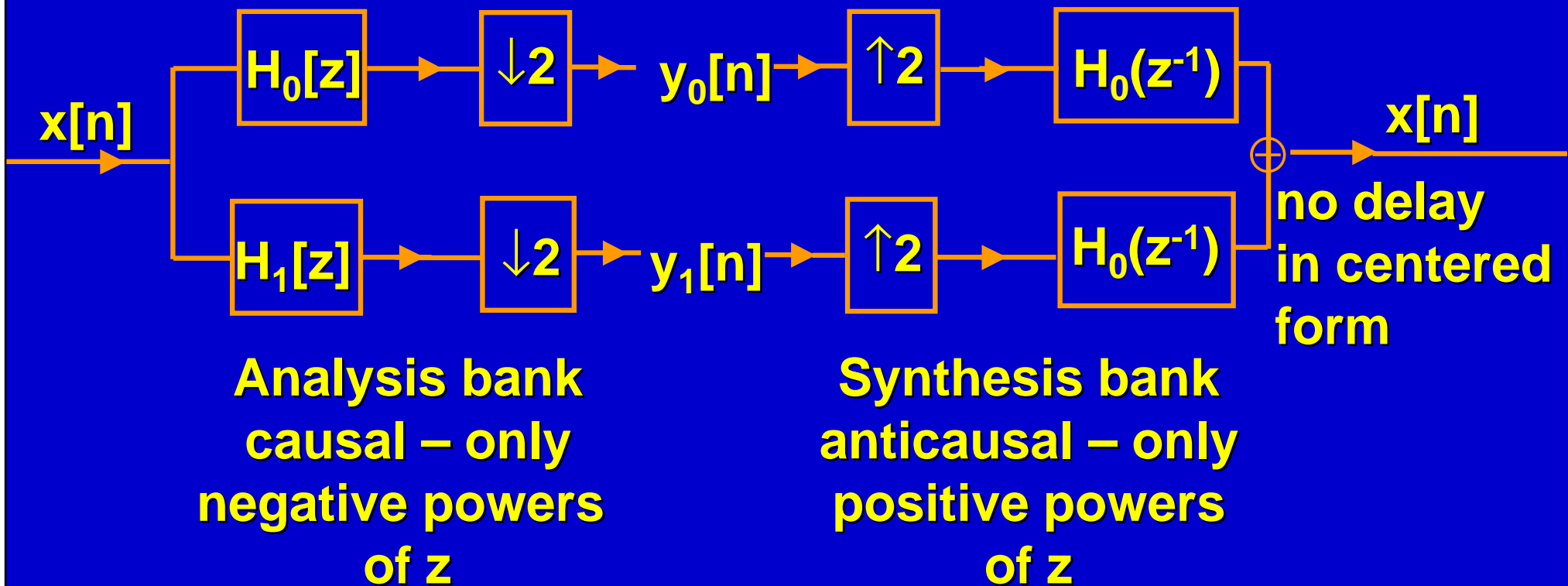
$$\Rightarrow f_0[n] = h_0[N - n]$$

$$F_1(z) = -H_0(-z) = z^{-N} H_1(z^{-1})$$

$$\Rightarrow f_1[n] = h_1[N - n]$$

So the synthesis filters, $f_k[n]$, are just the time-reversed versions of the analysis filters, $h_k[n]$, with a delay.

**Why is the Daubechies factorization orthogonal?
Consider the centered form of the filter bank:**



In matrix form:

Analysis

$$\begin{bmatrix} y_0 \\ \hline y_1 \end{bmatrix} = \begin{bmatrix} L \\ \hline B \\ 1 \ 2 \ 3 \end{bmatrix} \begin{bmatrix} x \end{bmatrix}$$

W

Synthesis

$$\begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} L^T & | & B^T \\ \hline 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} y_0 \\ \hline y_1 \end{bmatrix}$$

W^T

So

$$x = W^T W x \text{ for any } x$$

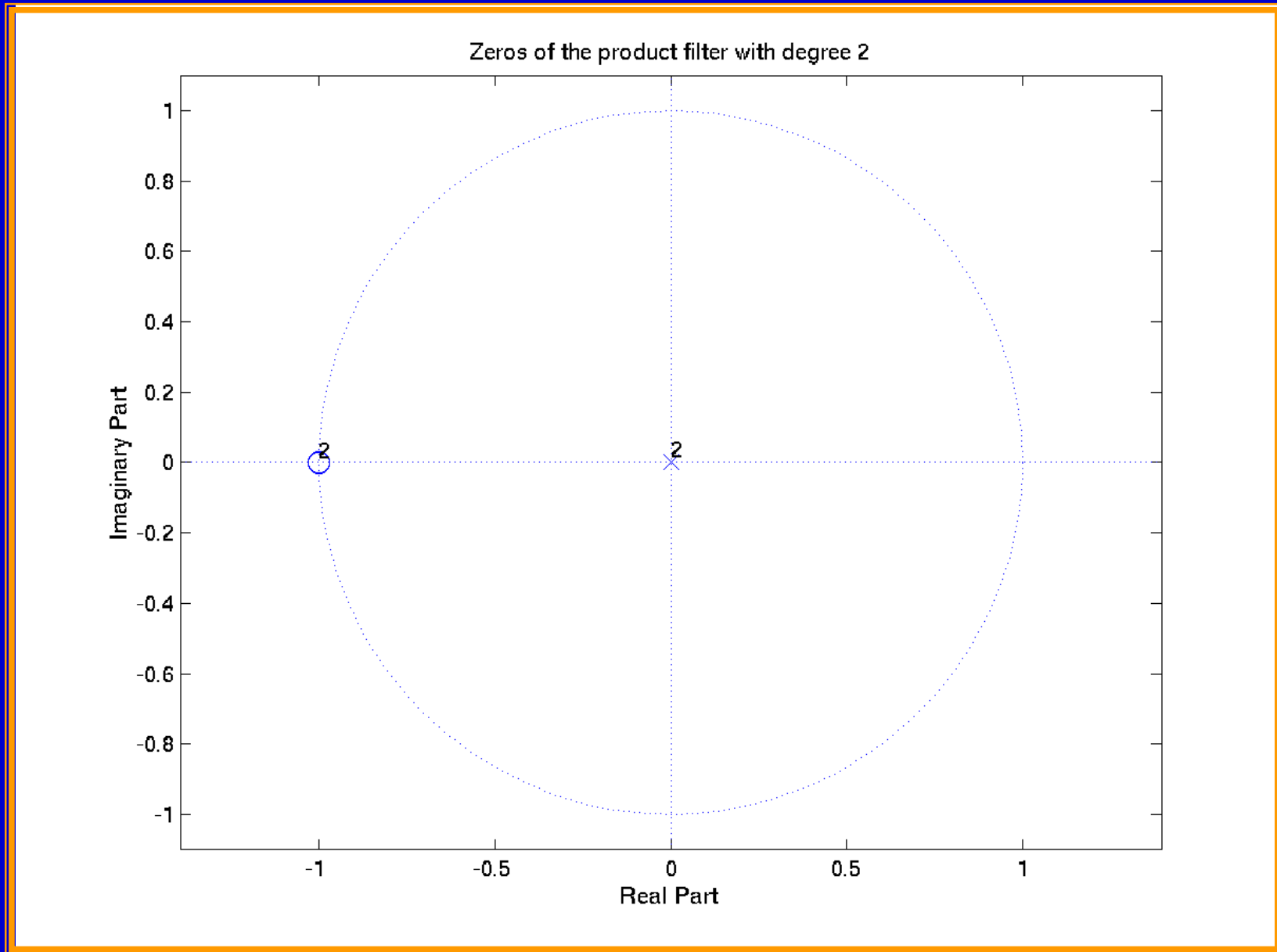
$$W^T W = I = W W^T$$

An important fact: symmetry prevents orthogonality

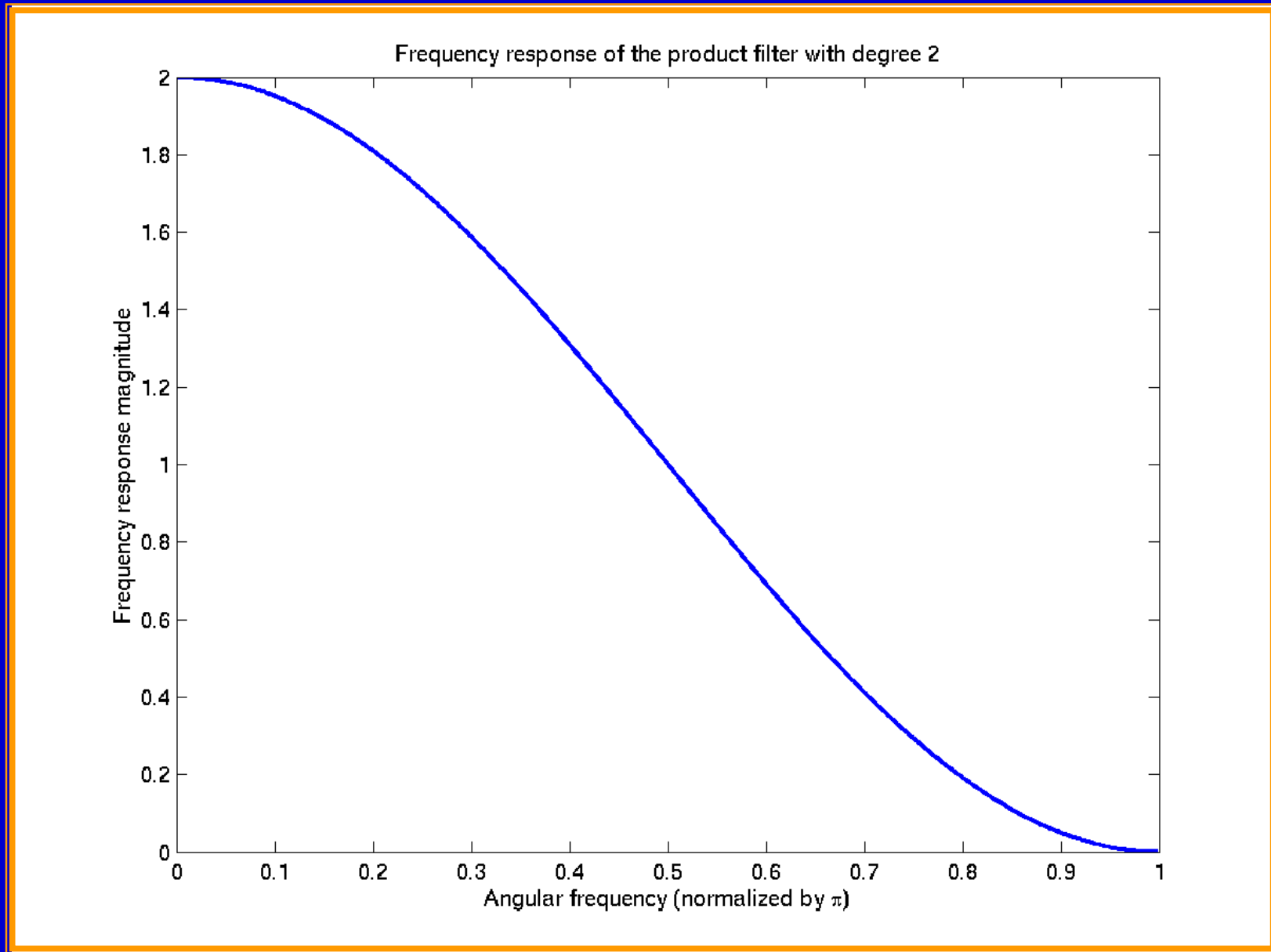
Matlab Example 2

1. Product filter examples

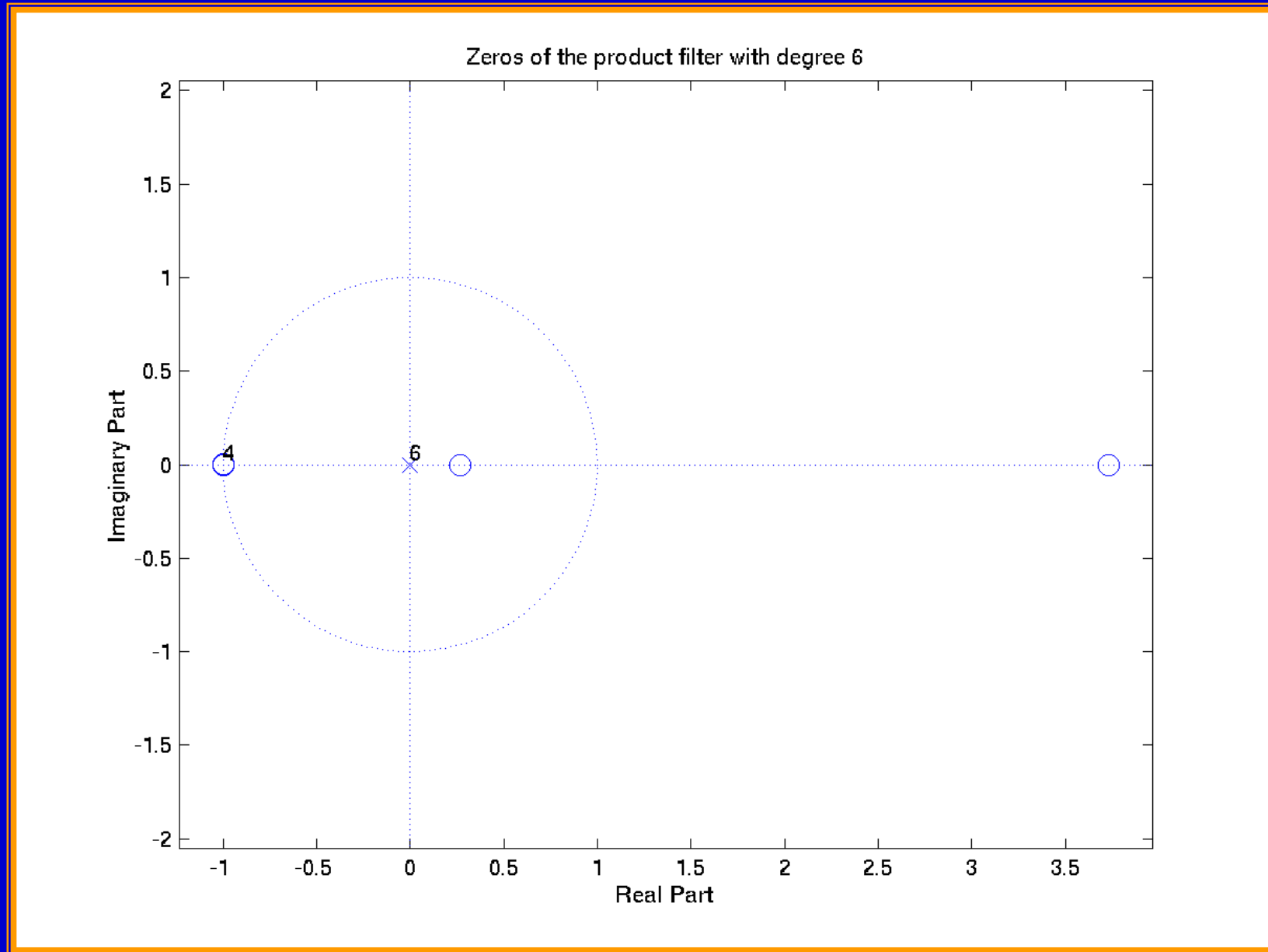
Degree-2 (p=1): pole-zero plot



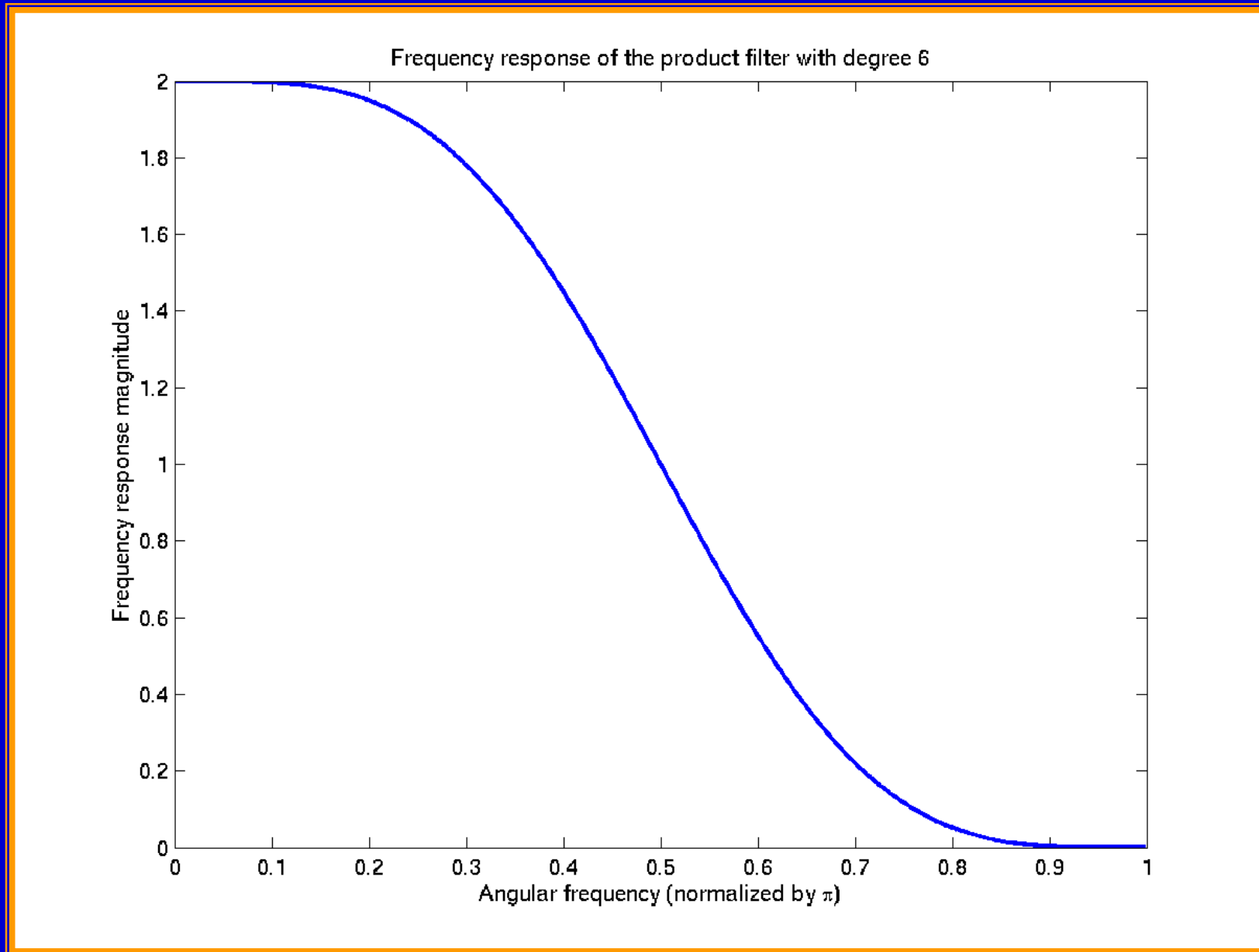
Degree-2 (p=1): Freq. response



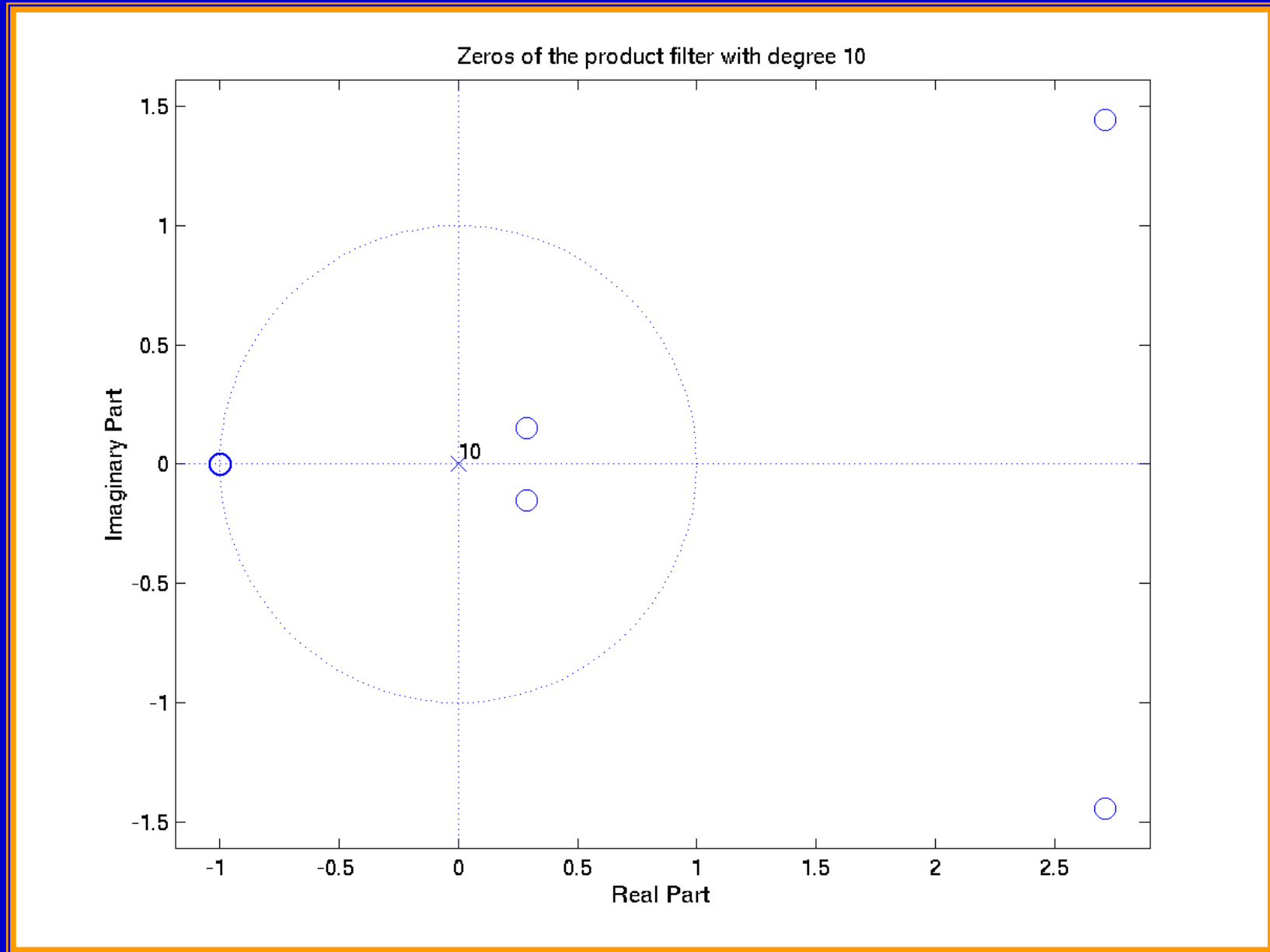
Degree-6 (p=2): pole-zero plot



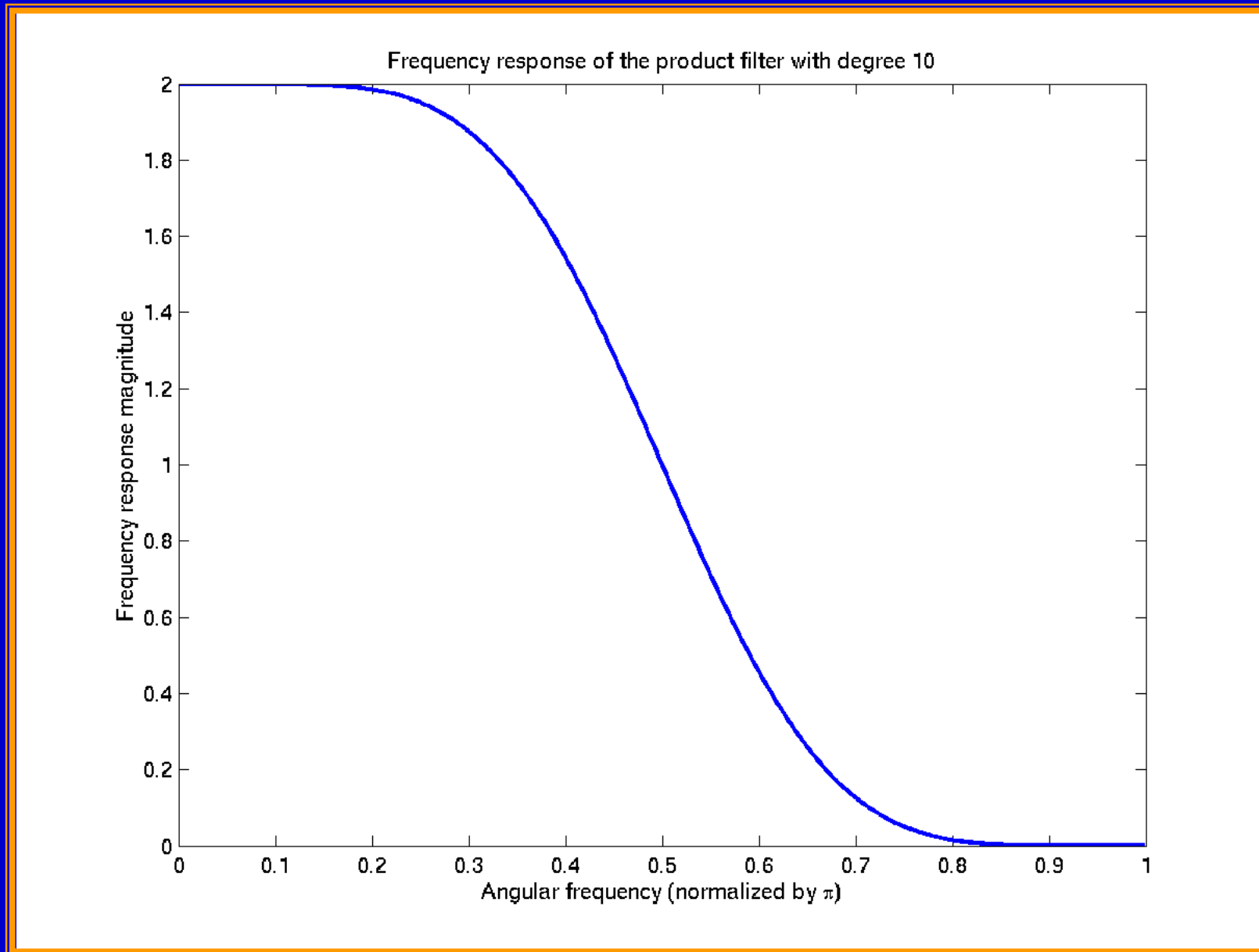
Degree-6 (p=2): Freq. response



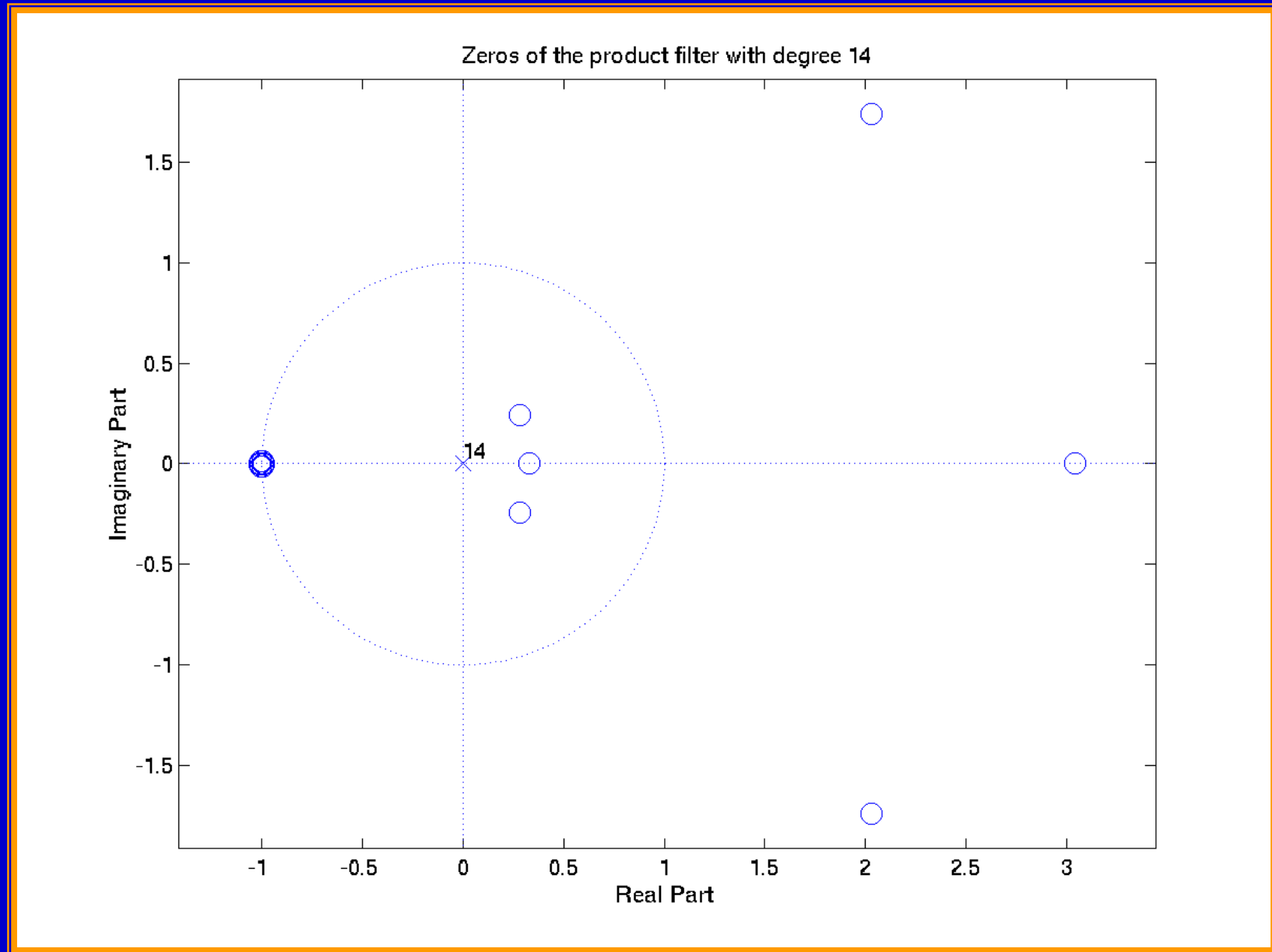
Degree-10 (p=3): pole-zero plot



Degree-10 (p=3): Freq. response



Degree-14 (p=4): pole-zero plot



Degree-14 (p=4): Freq. response

