FIR Filters

Mitch Withers, Res. Assoc. Prof., Univ. of Memphis

See Aster and Borchers, Time Series Analysis, chapter 5.

THE UNIVERSITY OF MEMPHIS
Center for Earthquake Research and Information

Ut quod ali cibus est aliis fuat acre venenum.
What is food to one, is to others bitter poison.
Titus Lucretius Carus (c. 99 BC – 55 BC), Roman poet and philosopher.

An alternative interpretation: One person’s noise is another person’s signal.

Digital Filtering is a very broad topic some spend entire careers on. We'll be taking a cursory glance over a few short weeks.

Digital Filtering is a very broad topic some spend entire careers on. We'll be taking a cursory glance over a few short weeks.

Filtering provides a means to alter the amplitude and/or phase of different frequency components of the data.

Digital Filtering is a very broad topic some spend entire careers on. We'll be taking a cursory glance over a few short weeks.

Filtering provides a means to alter the amplitude and/or phase of different frequency components of the data.

1. Noise rejection/Signal Enhancement

Digital Filtering is a very broad topic some spend entire careers on. We'll be taking a cursory glance over a few short weeks.

Filtering provides a means to alter the amplitude and/or phase of different frequency components of the data.

1. Noise rejection/Signal Enhancement

2. Remove instrument response from data

Digital Filtering is a very broad topic some spend entire careers on. We'll be taking a cursory glance over a few short weeks.

Filtering provides a means to alter the amplitude and/or phase of different frequency components of the data.

1. Noise rejection/Signal Enhancement

2. Remove instrument response from data

3. Differentiate or integrate

Digital Filtering is a very broad topic some spend entire careers on. We'll be taking a cursory glance over a few short weeks.

Filtering provides a means to alter the amplitude and/or phase of different frequency components of the data.

1. Noise rejection/Signal Enhancement
2. Remove instrument response from data
3. Differentiate or integrate
4. Change sampling rate

Digital Filtering is a very broad topic some spend entire careers on. We'll be taking a cursory glance over a few short weeks.

Filtering provides a means to alter the amplitude and/or phase of different frequency components of the data.

1. Noise rejection/Signal Enhancement

2. Remove instrument response from data

3. Differentiate or integrate

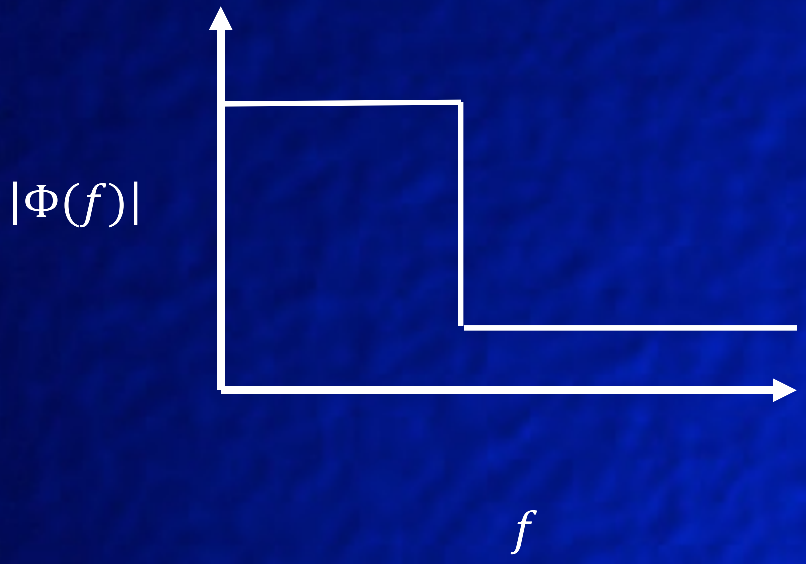
4. Change sampling rate

5. Creative effects in audio and video

One way to describe some filters we work with most frequently is what they do to the amplitude

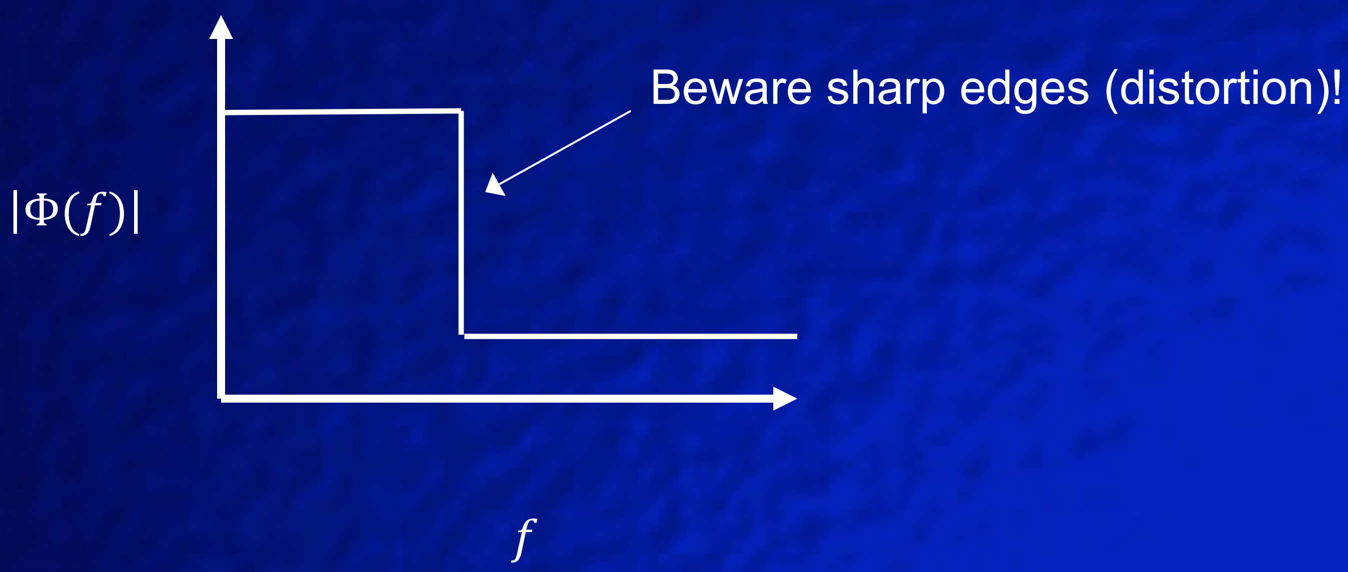
One way to describe some filters we work with most frequently is what they do to the amplitude

A low pass filter keeps low frequencies and rejects high frequencies.



One way to describe some filters we work with most frequently is what they do to the amplitude

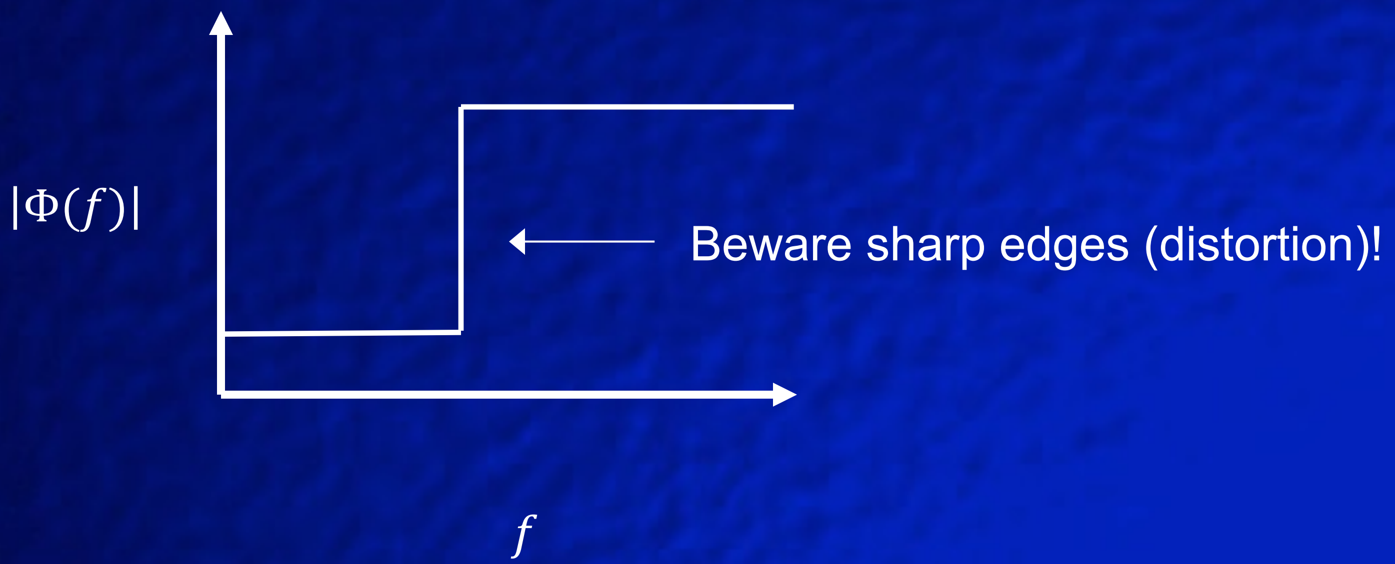
A low pass filter keeps low frequencies and rejects high frequencies.



THE UNIVERSITY OF MEMPHIS
Center for Earthquake Research and Information

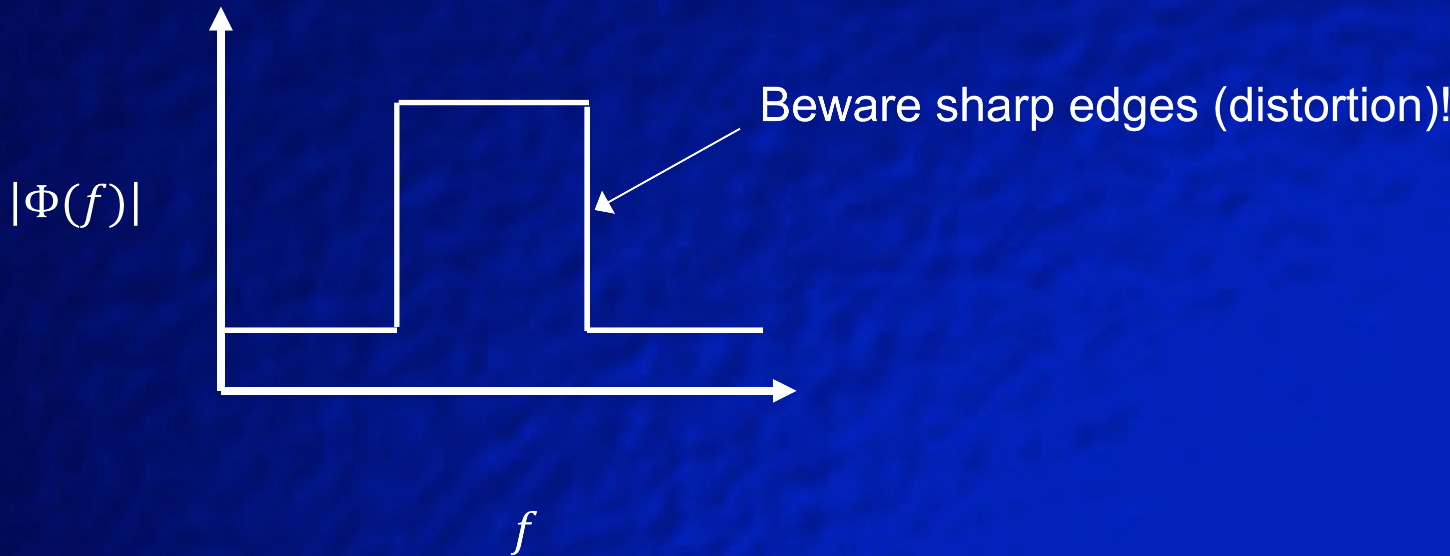
One way to describe some filters we work with most frequently is what they do to the amplitude

A high pass filter rejects low frequencies and keeps high frequencies.



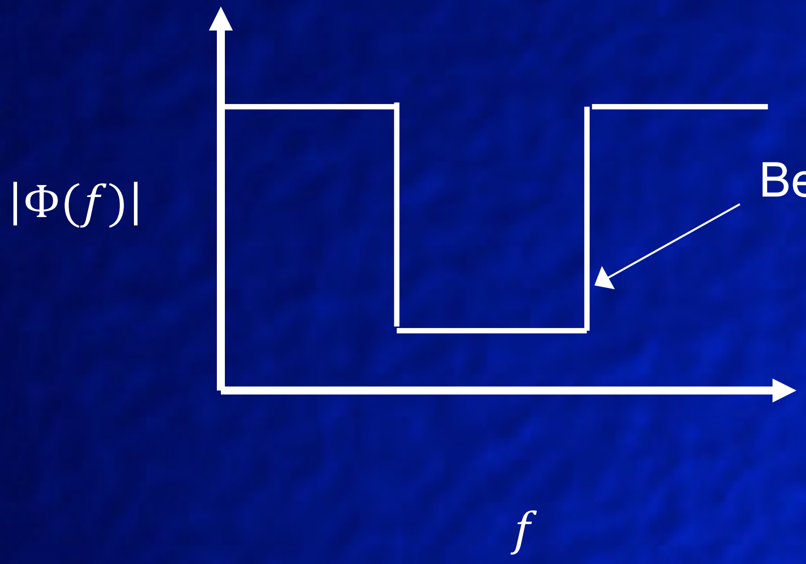
One way to describe some filters we work with most frequently is what they do to the amplitude

A band pass filter keeps frequencies within a specified range and rejects those outside the range.



One way to describe some filters we work with most frequently is what they do to the amplitude

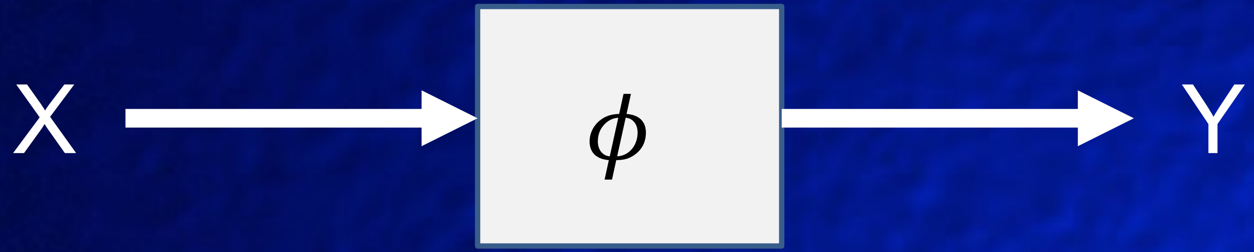
A notch or band gap filter rejects frequencies within a specified range and keeps those outside the range.



The key in the design is enhancing the desired portion of the signal while minimizing distortion.

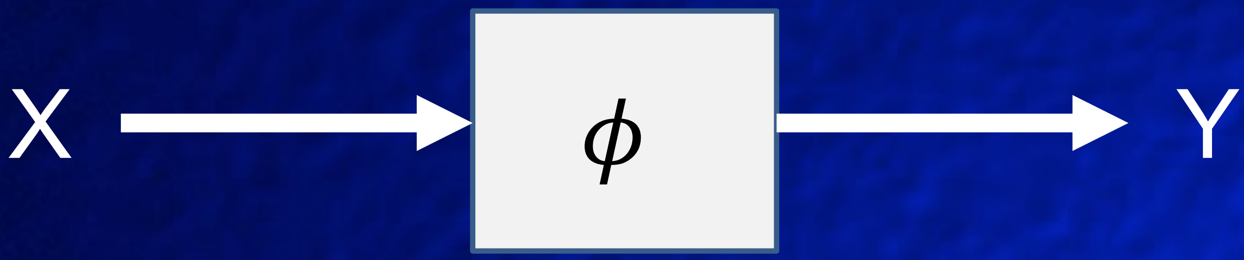
The key in the design is enhancing the desired portion of the signal while minimizing distortion.

Most filters we encounter are also Linear Time Invariant systems (LTI).



The key in the design is enhancing the desired portion of the signal while minimizing distortion.

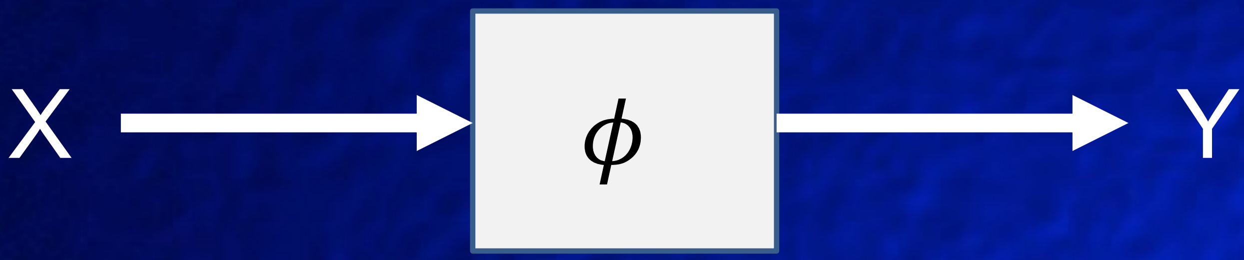
Most filters we encounter are also Linear Time Invariant systems (LTI).



Allows us to take advantage of all the tools we’ve learned so far.

The key in the design is enhancing the desired portion of the signal while minimizing distortion.

Most filters we encounter are also Linear Time Invariant systems (LTI).



Allows us to take advantage of all the tools we've learned so far.

Beware the unstable filter (no poles on the RHS of the S-plane).

THE UNIVERSITY OF MEMPHIS.
Center for Earthquake Research and Information

One of the most simple, though not the best, ways to filter is by direct manipulation in the frequency domain.

One of the most simple, though not the best, ways to filter is by direct manipulation in the frequency domain.

Computing the FT may be impractical for very long time series.

One of the most simple, though not the best, ways to filter is by direct manipulation in the frequency domain.

Computing the FT may be impractical for very long time series.

Not real-time.

One of the most simple, though not the best, ways to filter is by direct manipulation in the frequency domain.

Computing the FT may be impractical for very long time series.

Not real-time.

Windowing cautions apply.

One of the most simple, though not the best, ways to filter is by direct manipulation in the frequency domain.

Computing the FT may be impractical for very long time series.

Not real-time.

Windowing cautions apply.

Run matlab command, filterdemo_mac

FIR → Finite Impulse Response

Finite duration time-domain impulse response, often ten to a few hundred points, applied with direct time-domain convolution.

FIR → Finite Impulse Response

Finite duration time-domain impulse response, often ten to a few hundred points, applied with direct time-domain convolution.

Fast → no FT required.

FIR → Finite Impulse Response

Finite duration time-domain impulse response, often ten to a few hundred points, applied with direct time-domain convolution.

Fast → no FT required.

Real-time → Can be done in real-time using a small buffer (which means it can be acausal if you're not careful).

FIR → Finite Impulse Response

Finite duration time-domain impulse response, often ten to a few hundred points, applied with direct time-domain convolution.

Fast → no FT required.

Real-time → Can be done in real-time using a small buffer (which means it can be acausal if you're not careful).

Stable → no recursive components. Time domain impulse response goes to 0 within finite number of points.

FIR → Finite Impulse Response

Finite duration time-domain impulse response, often ten to a few hundred points, applied with direct time-domain convolution.

Fast → no FT required.

Real-time → Can be done in real-time using a small buffer (which means it can be acausal if you're not careful).

Stable → no recursive components. Time domain impulse response goes to 0 within finite number of points.

Linear phase response → minimal distortion

In the construction of a FIR filter, we typically select some desired frequency response, $\Omega(f)$, and a desired number of points, M.

In the construction of a FIR filter, we typically select some desired frequency response, $\Omega(f)$, and a desired number of points, M.

Choice of M is key and can either be done by truncation (aka, windowing with a boxcar), or by windowing in the time domain to M-points.

In the construction of a FIR filter, we typically select some desired frequency response, $\Omega(f)$, and a desired number of points, M.

Choice of M is key and can either be done by truncation (aka, windowing with a boxcar), or by windowing in the time domain to M-points.

Alternatively, choose the desired frequency-domain resolution, Δf, to determine M.

In the construction of a FIR filter, we typically select some desired frequency response, $\Omega(f)$, and a desired number of points, M.

Choice of M is key and can either be done by truncation (aka, windowing with a boxcar), or by windowing in the time domain to M-points.

Alternatively, choose the desired frequency-domain resolution, Δf, to determine M.

Both approaches accomplish the same thing, choosing M, but depend on which aspect is most important to the design; time-length or frequency resolution.

Consider the running mean filter, an easy and fast FIR filter.

$$w_n = \frac{1}{M}\Pi_m = \begin{cases} \frac{1}{M}, & |n| \le {}^{(M-1)}\!/_{2} \\ 0, & |n| > {}^{(M-1)}\!/_{2} \end{cases}$$

Consider the running mean filter, and easy and fast FIR filter.

$$w_n = \frac{1}{M}\Pi_m = \begin{cases} \frac{1}{M}, & |n| \leq {}^{(M-1)}/_{2} \\ 0, & |n| > {}^{(M-1)}/_{2} \end{cases}$$

Where $w_n = w_0, w_1, w_2, \cdots, w_{M-1}$ are the time-domain filter weights.

Consider the running mean filter, an easy and fast FIR filter.

$$w_n = \frac{1}{M}\Pi_m = \begin{cases} \frac{1}{M}, & |n| \leq {}^{(M-1)}/_{2} \\ 0, & |n| > {}^{(M-1)}/_{2} \end{cases}$$

Where $w_n = w_0, w_1, w_2, \cdots, w_{M-1}$ are the time-domain filter weights.

Convolution of the data with the filter weights, w_n, gives a weighted sum of the input within Π_M or in this case, an average, since all the weights are equal.

Consider the running mean filter, an easy and fast FIR filter.

$$w_n = \frac{1}{M}\Pi_m = \begin{cases} \frac{1}{M}, & |n| \leq {}^{(M-1)}\!/_{2} \\ 0, & |n| > {}^{(M-1)}\!/_{2} \end{cases}$$

Where $w_n = w_0, w_1, w_2, \cdots, w_{M-1}$ are the time-domain filter weights.

Convolution of the data with the filter weights, w_n, gives a weighted sum of the input within Π_M or in this case, an average, since all the weights are equal.

The output is then, $z_n = y_n * w_n = \sum_{l=-\infty}^{\infty} y_l w_{n-l}$

Consider the running mean filter, possible the most common FIR filter.

$$w_n = \frac{1}{M}\Pi_m = \begin{cases} \frac{1}{M}, & |n| \le {(M-1)}/{2} \\ 0, & |n| > {(M-1)}/{2} \end{cases}$$

Where $w_n = w_0, w_1, w_2, \cdots, w_{M-1}$ are the time-domain filter weights.

Convolution of the data with the filter weights, w_n, gives a weighted sum of the input within Π_M or in this case, an average, since all the weights are equal.

The output is then, $z_n = y_n * w_n = \sum_{l=-\infty}^{\infty} y_l w_{n-l} = \frac{1}{M} \sum_{l=-(M-1)/2}^{(M-1)/2} y_l$

For example, let $M = 5$

$$z[10] = \frac{y[8] + y[9] + y[10] + y[11] + y[12]}{5}$$

For example, let $M = 5$

$$z[10] = \frac{y[8] + y[9] + y[10] + y[11] + y[12]}{5}$$

$$z[11] = \frac{y[9] + y[10] + y[11] + y[12] + y[13]}{5}$$

For example, let $M = 5$

$$z[10] = \frac{y[8] + y[9] + y[10] + y[11] + y[12]}{5}$$

$$z[11] = \frac{y[9] + y[10] + y[11] + y[12] + y[13]}{5}$$

$$= y[10] + \frac{y[13] - y[8]}{5}$$

THE UNIVERSITY OF MEMPHIS
Center for Earthquake Research and Information

For example, let $M = 5$

$$z[10] = \frac{y[8] + y[9] + y[10] + y[11] + y[12]}{5}$$

$$z[11] = \frac{y[9] + y[10] + y[11] + y[12] + y[13]}{5}$$

$$= z[10] + \frac{y[13] - y[8]}{5}$$

→ Very fast to implement in code.

For example, let $M = 5$

$$z[10] = \frac{y[8] + y[9] + y[10] + y[11] + y[12]}{5}$$

$$z[11] = \frac{y[9] + y[10] + y[11] + y[12] + y[13]}{5}$$

$$= z[10] + \frac{y[13] - y[8]}{5}$$

→ Very fast to implement in code.

We use the DFT to see the frequency domain response of this filter.

THE UNIVERSITY OF MEMPHIS. Center for Earthquake Research and Information

$$Z_k = Y_k \cdot DFT[w_n]$$

THE UNIVERSITY OF MEMPHIS
Center for Earthquake Research and Information

$$Z_k = Y_k \cdot DFT[w_n] \; = Y_k \frac{1}{M} \sum_{n=-(M-1)/2}^{(M-1)/2} e^{-i2\pi kn/N}$$

$$Z_k = Y_k \cdot DFT[w_n] = Y_k \frac{1}{M} \sum_{n=-{(M-1)}/{2}}^{{(M-1)}/{2}} e^{-i2\pi kn/N}$$

The DFT of a boxcar

$$Z_k = Y_k \cdot DFT[w_n] \quad = Y_k \frac{1}{M} \underbrace{\sum_{n=-(M-1)/2}^{(M-1)/2} e^{-i2\pi kn/N}}_{\text{The DFT of a boxcar}}$$

The DFT of a boxcar

$$= Y_k \frac{1}{M} \frac{\sin\left(m\pi k/_N\right)}{\sin\left(\pi k/_N\right)}$$

$$Z_k = Y_k \cdot DFT[w_n] = Y_k \frac{1}{M} \sum_{n=-(M-1)/2}^{(M-1)/2} e^{-i2\pi kn/N}$$

The DFT of a boxcar

$$= Y_k \frac{1}{M} \frac{\sin\left(m\pi k/_N\right)}{\sin\left(\pi k/_N\right)}$$

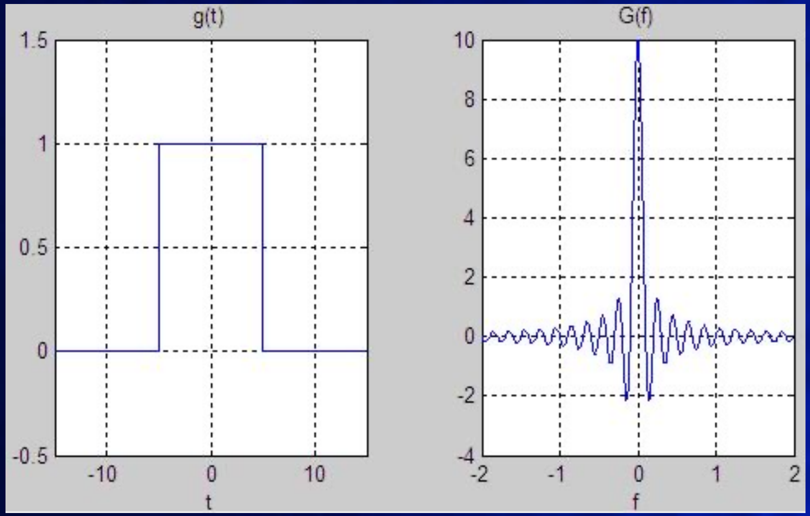
→ The discrete equivalent of a sync function.

$$Z_k = Y_k \cdot DFT[w_n] \;\; = Y_k \frac{1}{M} \sum_{n=-{(M-1)}/{2}}^{{(M-1)}/{2}} e^{-i2\pi kn/N}$$

The DFT of a boxcar

$$= Y_k \frac{1}{M} \frac{\sin\left({m\pi k}/{N}\right)}{\sin\left({\pi k}/{N}\right)}$$

The discrete equivalent of a sync function.



A low pass filter with a lot of ripple in the stop band.

In general, an M-point FIR filter an be written in the frequency domain as,

$$W_k = \sum_{n=0}^{M-1} w_n e^{-i2\pi kn/N}$$

In general, an M-point FIR filter an be written in the frequency domain as,

$$W_k = \sum_{n=0}^{M-1} w_n e^{-i2\pi kn/N}$$

If w_n is symmetric and real, it has linear phase response.

$$W_k = \underbrace{e^{-i2\pi k(M-1)/N}}_{\text{Phase shift}} \underbrace{\sum_{n=-(M-1)/2}^{(M-1)/2}}_{\text{Time shift}} w_n e^{-i2\pi kn/N}$$

Recall the shift property of the FT

In general, an M-point FIR filter an be written in the frequency domain as,

$$W_k = \sum_{n=0}^{M-1} w_n e^{-i2\pi kn/N}$$

If w_n is symmetric and real, it has linear phase response.

$$W_k = \underbrace{e^{-i2\pi k(M-1)/N}}_{\text{Phase shift}} \underbrace{\sum_{n=-(M-1)/2}^{(M-1)/2}}_{\text{Time shift}} w_n e^{-i2\pi kn/N}$$

w_n is symmetric (even function)

$$= e^{-i2\pi k(M-1)/N} \left(2 \sum_{n=1}^{(M-1)/2} w_n \cos(2\pi kn/N) + w_0 \right)$$

In general, an M-point FIR filter an be written in the frequency domain as,

$$W_k = \sum_{n=0}^{M-1} w_n e^{-i2\pi kn/N}$$

If w_n is symmetric and real, it has linear phase response.

$$W_k = \underbrace{e^{-i2\pi k(M-1)/N}}_{\text{Phase shift}} \underbrace{\sum_{n=-(M-1)/2}^{(M-1)/2}}_{\text{Time shift}} w_n e^{-i2\pi kn/N}$$

w_n is symmetric (even function)

$$= e^{-i2\pi k(M-1)/N} \left(2 \sum_{n=1}^{(M-1)/2} w_n \cos(2\pi kn/N) + w_0 \right)$$

$$= P(k) \cdot A(k)$$

$$W_k = P_k \cdot A_k$$

P_k is a linear phase shift just like the 15f phase shift from before

$$W_k = P_k \cdot A_k$$

P_k is a linear phase shift just like the 15f phase shift from before

Time shifts of each frequency component are all the same relative to each other.

$$W_k = P_k \cdot A_k$$

P_k is a linear phase shift just like the 15f phase shift from before

Time shifts of each frequency component are all the same relative to each other.

No distortion.

$$W_k = P_k \cdot A_k$$

P_k is a linear phase shift just like the 15f phase shift from before

Time shifts of each frequency component are all the same relative to each other.

No distortion.

A_k is the real valued frequency domain amplitude response of the filter.

$$W_k = P_k \cdot A_k$$

P_k is a linear phase shift just like the 15f phase shift from before

Time shifts of each frequency component are all the same relative to each other.

No distortion.

A_k is the real valued frequency domain amplitude response of the filter.

The matlab **conv($\boldsymbol{x}, \boldsymbol{w}$)** command is useful to convolve the filter sequence (or weights), w with the input sequence (or data), x.

$$W_k = P_k \cdot A_k$$

P_k is a linear phase shift just like the 15f phase shift from before

Time shifts of each frequency component are all the same relative to each other.

No distortion.

A_k is the real valued frequency domain amplitude response of the filter.

The matlab **conv(**$\boldsymbol{x}, \boldsymbol{w}$) command is useful to convolve the filter sequence (or weights), w with the input sequence (or data), x.

This will produce an extra $M - 1$ points on the output which is the time it takes the filter response to decay after the last data point.

$$W_k = P_k \cdot A_k$$

P_k is a linear phase shift just like the 15f phase shift from before

Time shifts of each frequency component are all the same relative to each other.

No distortion.

A_k is the real valued frequency domain amplitude response of the filter.

The matlab **conv(**$\boldsymbol{x}, \boldsymbol{w}$) command is useful to convolve the filter sequence (or weights), w with the input sequence (or data), x.

This will produce an extra $M - 1$ points on the output which is the time it takes the filter response to decay after the last data point.

Just cut the extra points if not needed, y=y(1:N);

The matlab **filter** command may also be used to implement your FIR filter. It’s designed for IIR filters (more later) which uses two vectors but works just as well with 1 vector.

```
y=filter(w,[1],x);
```

One method of FIR filter design is to begin with an ideal continuous frequency domain response, $\Omega(f)$, and inverse FT on the Nyquist interval to get the time domain FIR weights.

One method of FIR filter design is to begin with an ideal continuous frequency domain response, $\Omega(f)$, and inverse FT on the Nyquist interval to get the time domain FIR weights.

$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df$$

where f_s is the sample rate.

THE UNIVERSITY OF MEMPHIS
Center for Earthquake Research and Information

One method of FIR filter design is to begin with an ideal continuous frequency domain response, $\Omega(f)$, and inverse FT on the Nyquist interval to get the time domain FIR weights.

$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df$$

where f_s is the sample rate.

It is frequently more intuitive to design a filter in the frequency domain.

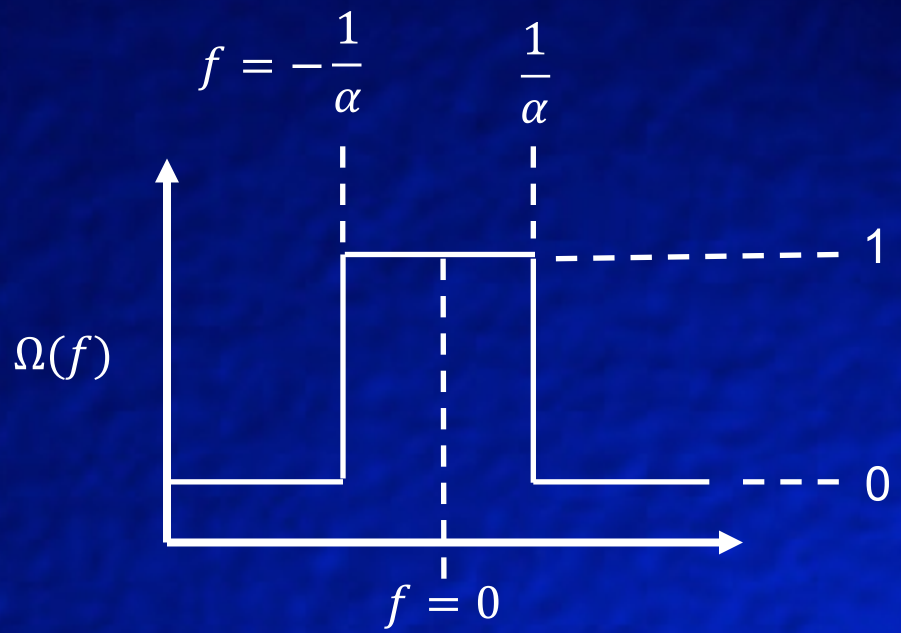
The problem here is there may be an infinite number of non-zero weights, w_n.

THE UNIVERSITY OF MEMPHIS
Center for Earthquake Research and Information

Consider the ideal low pass filter, a boxcar.

$$\Omega(f) = \Pi\left(\frac{\alpha f}{2}\right)$$

Transitions at $\frac{\alpha f}{2} = \pm\frac{1}{2}$

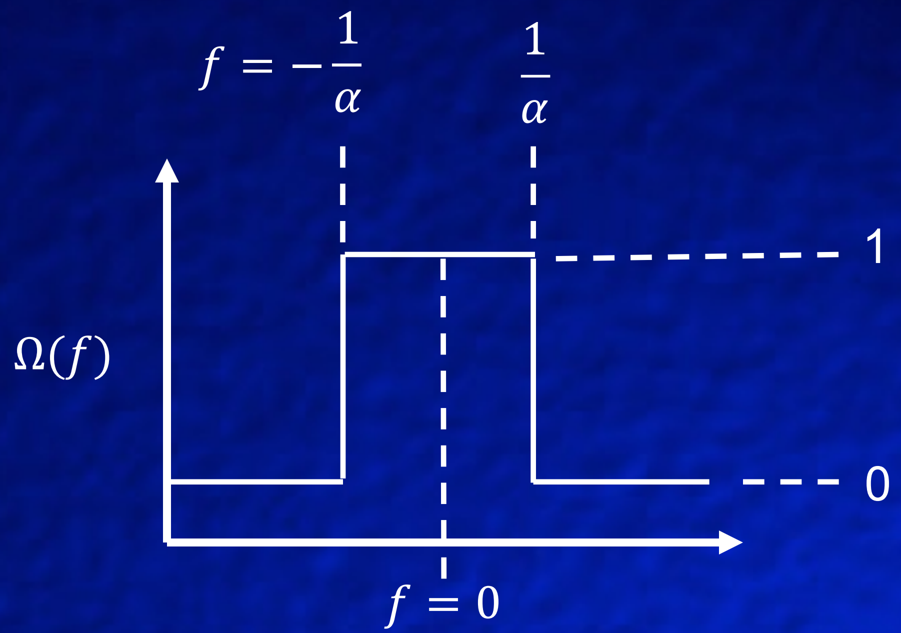


Consider the ideal low pass filter, a boxcar.

$$\Omega(f) = \Pi\left(\frac{\alpha f}{2}\right)$$

Transitions at $\frac{\alpha f}{2} = \pm\frac{1}{2}$

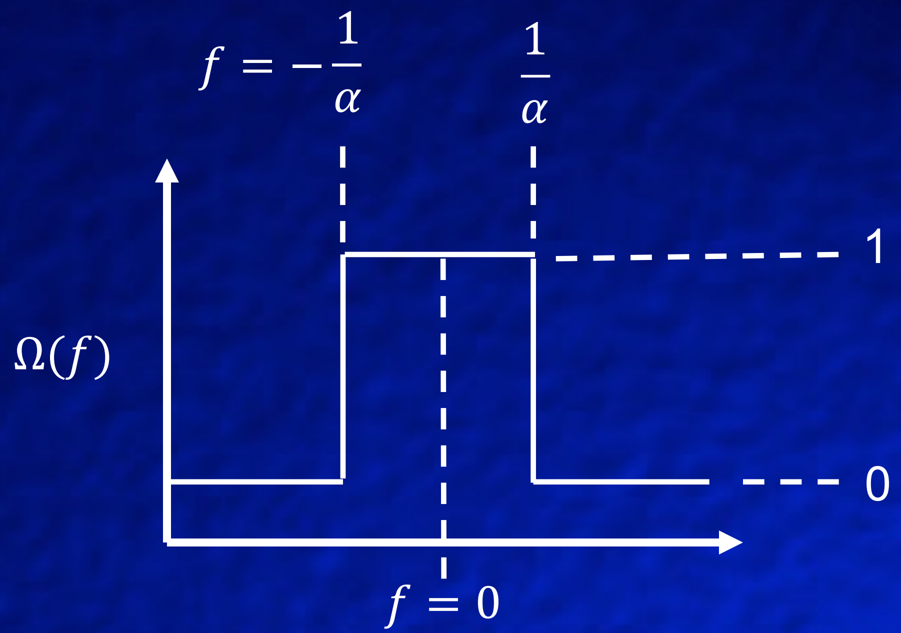
$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df$$



Consider the ideal low pass filter, a boxcar.

$$\Omega(f) = \Pi\left(\frac{\alpha f}{2}\right)$$

Transitions at $\frac{\alpha f}{2} = \pm\frac{1}{2}$



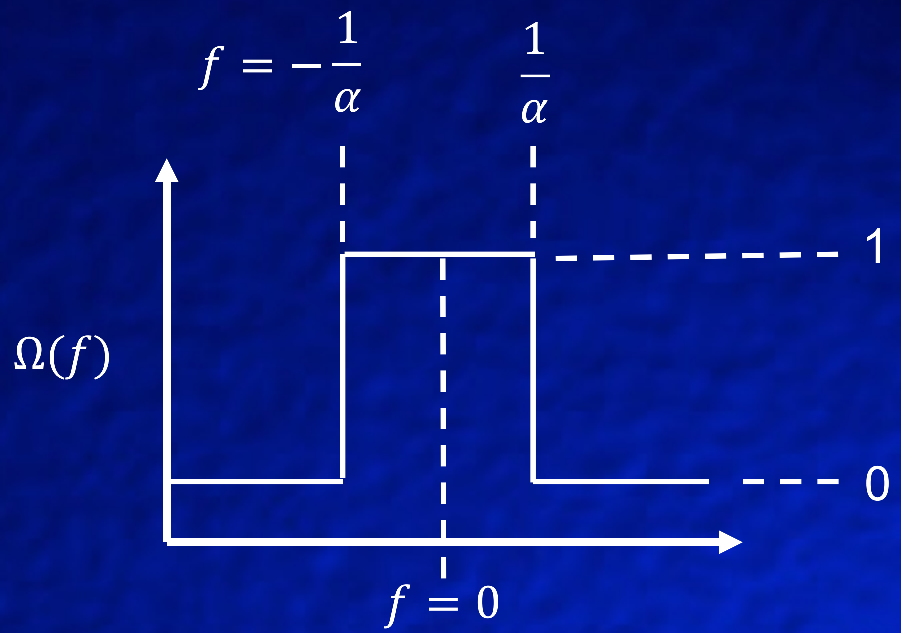
$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df = \int_{-f_s/\alpha}^{f_s/\alpha} e^{i2\pi f n} df$$

For simplicity, let $f_s = 1$

Consider the ideal low pass filter, a boxcar.

$$\Omega(f) = \Pi\left(\frac{\alpha f}{2}\right)$$

Transitions at $\frac{\alpha f}{2} = \pm\frac{1}{2}$



$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df = \int_{-f_s/\alpha}^{f_s/\alpha} e^{i2\pi f n} df$$

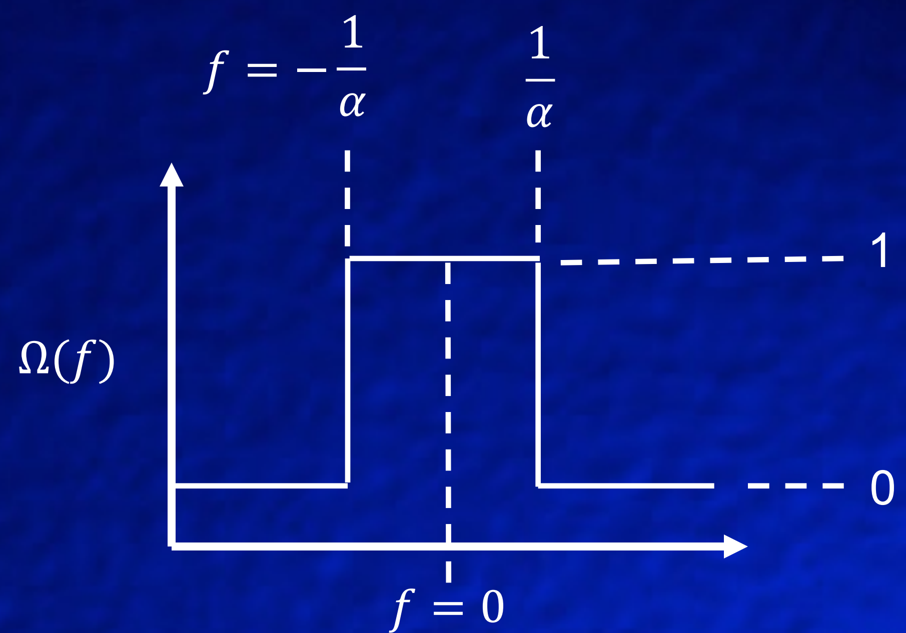
For simplicity, let $f_s = 1$

$$= \int_{-1/\alpha}^{1/\alpha} e^{i2\pi f n} df$$

Consider the ideal low pass filter, a boxcar.

$$\Omega(f) = \Pi\left(\frac{\alpha f}{2}\right)$$

Transitions at $\frac{\alpha f}{2} = \pm\frac{1}{2}$



$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df = \int_{-f_s/\alpha}^{f_s/\alpha} e^{i2\pi f n} df$$

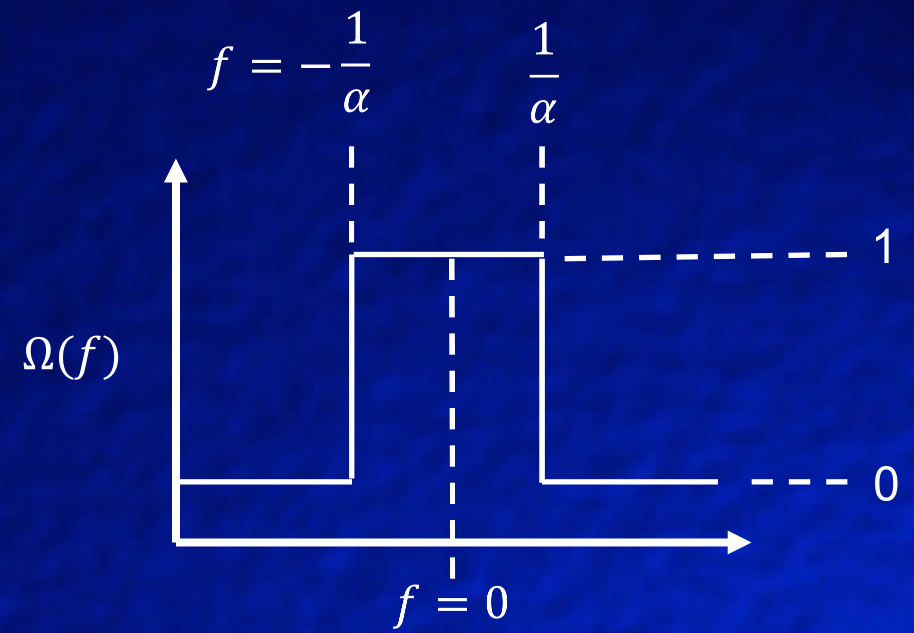
For simplicity, let $f_s = 1$

$$= \int_{-1/\alpha}^{1/\alpha} e^{i2\pi f n} df = 2\int_{0}^{1/\alpha} \cos(2\pi f n) df$$

Consider the ideal low pass filter, a boxcar.

$$\Omega(f) = \Pi\left(\frac{\alpha f}{2}\right)$$

Transitions at $\frac{\alpha f}{2} = \pm\frac{1}{2}$



$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df = \int_{-f_s/\alpha}^{f_s/\alpha} e^{i2\pi f n} df$$

For simplicity, let $f_s = 1$

$$= \int_{-1/\alpha}^{1/\alpha} e^{i2\pi f n} df = 2\int_{0}^{1/\alpha} \cos(2\pi f n) df = \frac{2}{\alpha} sinc\left(\frac{2n}{\alpha}\right)$$

$w_n = \frac{2}{\alpha} sinc\left(\frac{2n}{\alpha}\right)$, are the time domain weights of our filter for the ideal low pass response.

$w_n = \frac{2}{\alpha} sinc\left(\frac{2n}{\alpha}\right)$, are the time domain weights of our filter for the ideal low pass response.

The weights decay as $\frac{1}{n}$ and only goes to 0 in the limit as $n \rightarrow \infty$.

$w_n = \frac{2}{\alpha} sinc\left(\frac{2n}{\alpha}\right)$, are the time domain weights of our filter for the ideal low pass response.

The weights decay as $\frac{1}{n}$ and only goes to 0 in the limit as $n \rightarrow \infty$.

We can truncate the weights using a boxcar from $\pm\, {}^{(M-1)}/_{2}$, but as we found with windowing in our discussion on power spectra, this creates spectral leakage in the frequency domain by convolving the ideal filter, $\Omega(f)$ with a sinc function.

THE UNIVERSITY OF MEMPHIS. Center for Earthquake Research and Information

$w_n = \frac{2}{\alpha} sinc\left(\frac{2n}{\alpha}\right)$, are the time domain weights of our filter for the ideal low pass response.

The weights decay as $\frac{1}{n}$ and only goes to 0 in the limit as $n \rightarrow \infty$.

We can truncate the weights using a boxcar from $\pm\, {}^{(M-1)}/_{2}$, but as we found with windowing in our discussion on power spectra, this creates spectral leakage in the frequency domain by convolving the ideal filter, $\Omega(f)$ with a sinc function.

This leads us right back to windowing.

Run matlab program aster_fig54ff.m

THE UNIVERSITY OF MEMPHIS
Center for Earthquake Research and Information

We designed our FIR filter by first selecting the desired ideal frequency response, $\Omega(f)$, then converting it to the discrete time domain with the integral transform,

$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df$$

where f_s is the sample rate.

This, incidentally, is similar to our asymmetric transform pair we found in equation 3.32 in Aster and Borchers while developing the DFT.

THE UNIVERSITY OF MEMPHIS. Center for Earthquake Research and Information

We designed our FIR filter by first selecting the desired ideal frequency response, $\Omega(f)$, then converting it to the discrete time domain with the integral transform,

$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df$$

where f_s is the sample rate.

We then *windowed* w_n to the desired number of weights (128 points in our example).

We designed our FIR filter by first selecting the desired ideal frequency response, $\Omega(f)$, then converting it to the discrete time domain with the integral transform,

$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{i2\pi f n} df$$

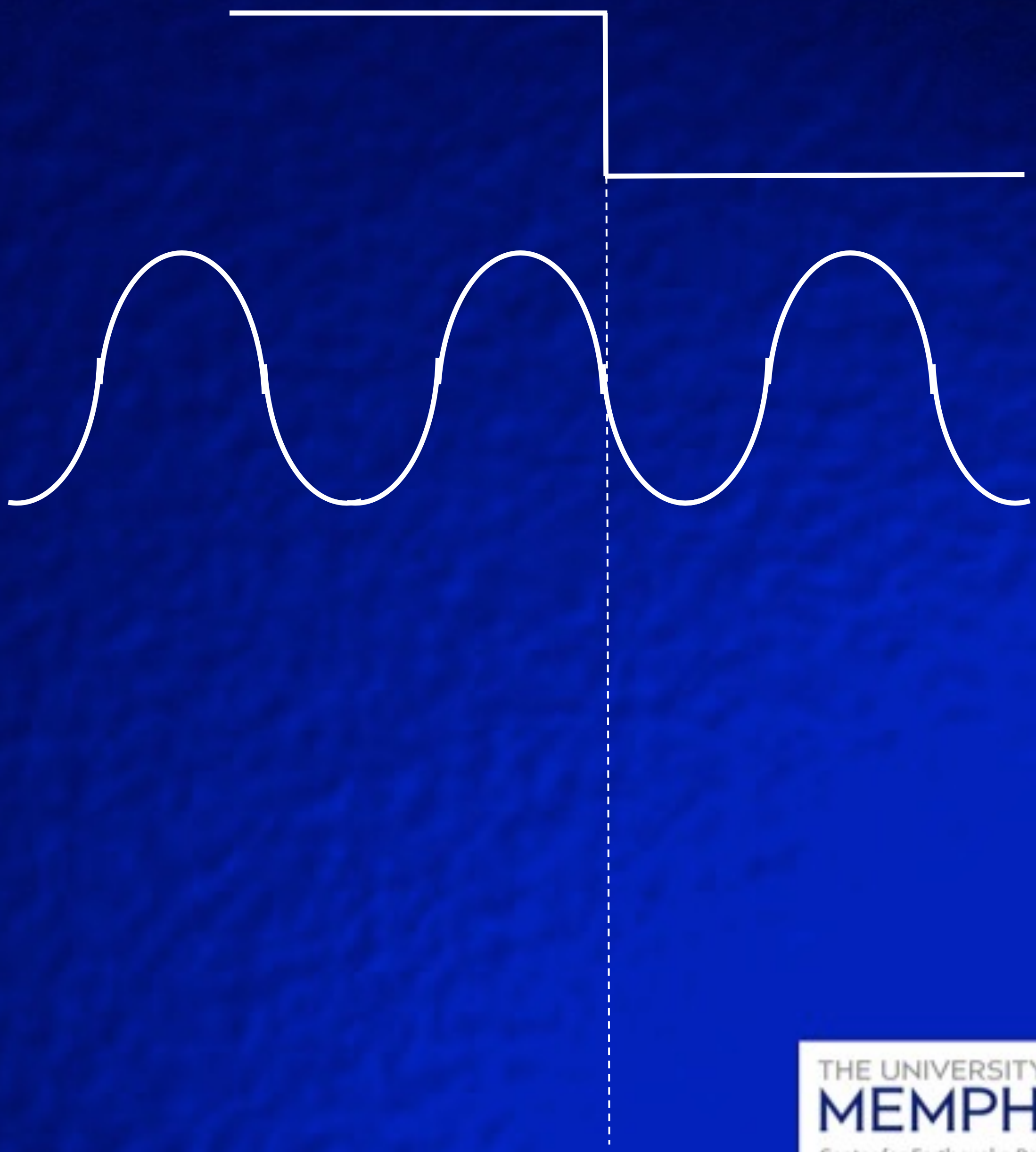
where f_s is the sample rate.

We then *windowed* w_n to the desired number of weights (128 points in our example).

This is not the best way to design a FIR filter but it is the easiest.

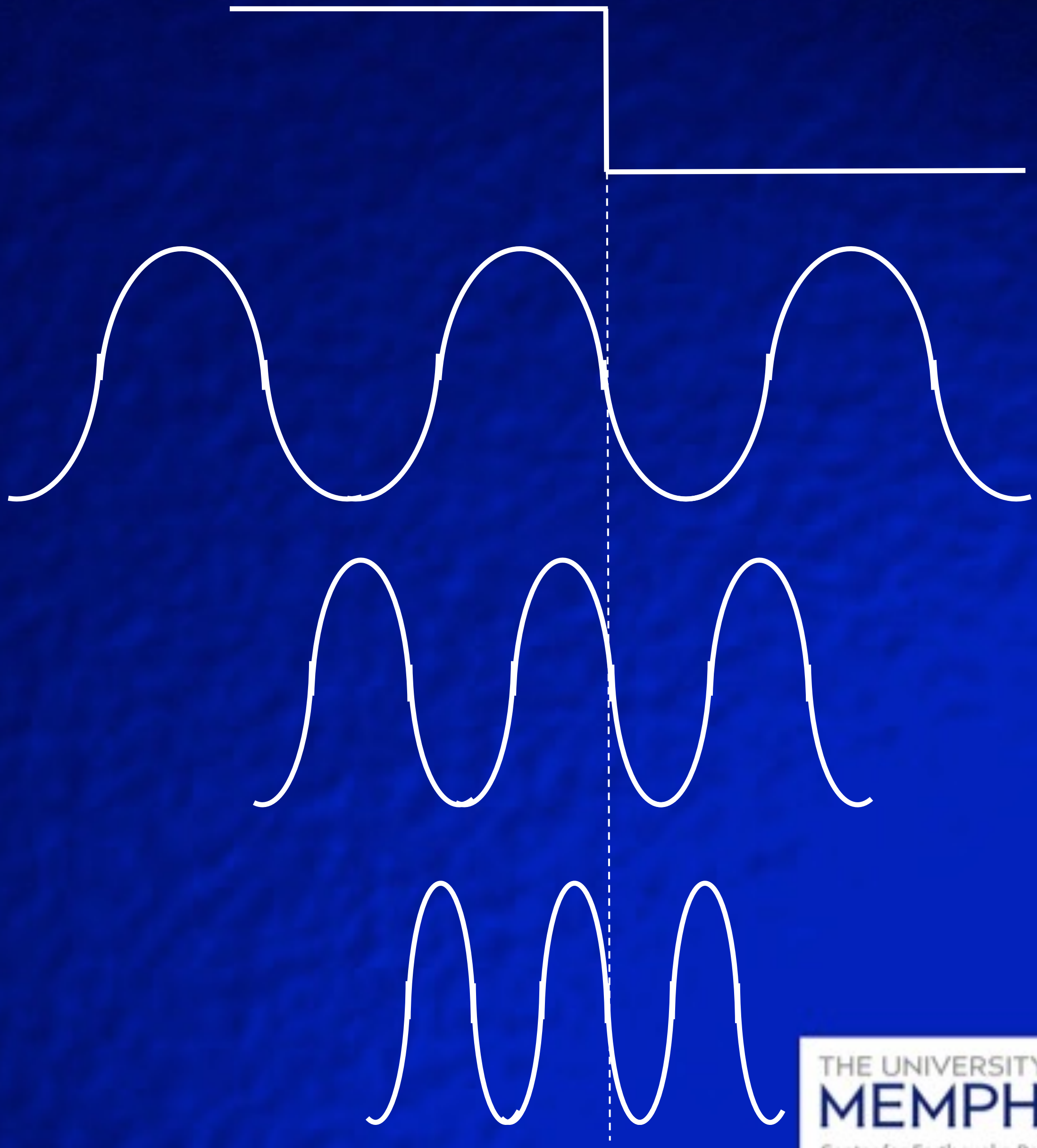
Run matlab program firpmdemo.m

Consider how successive superpositions of precisely aligned (think phase) cosines sum to create a discontinuity.



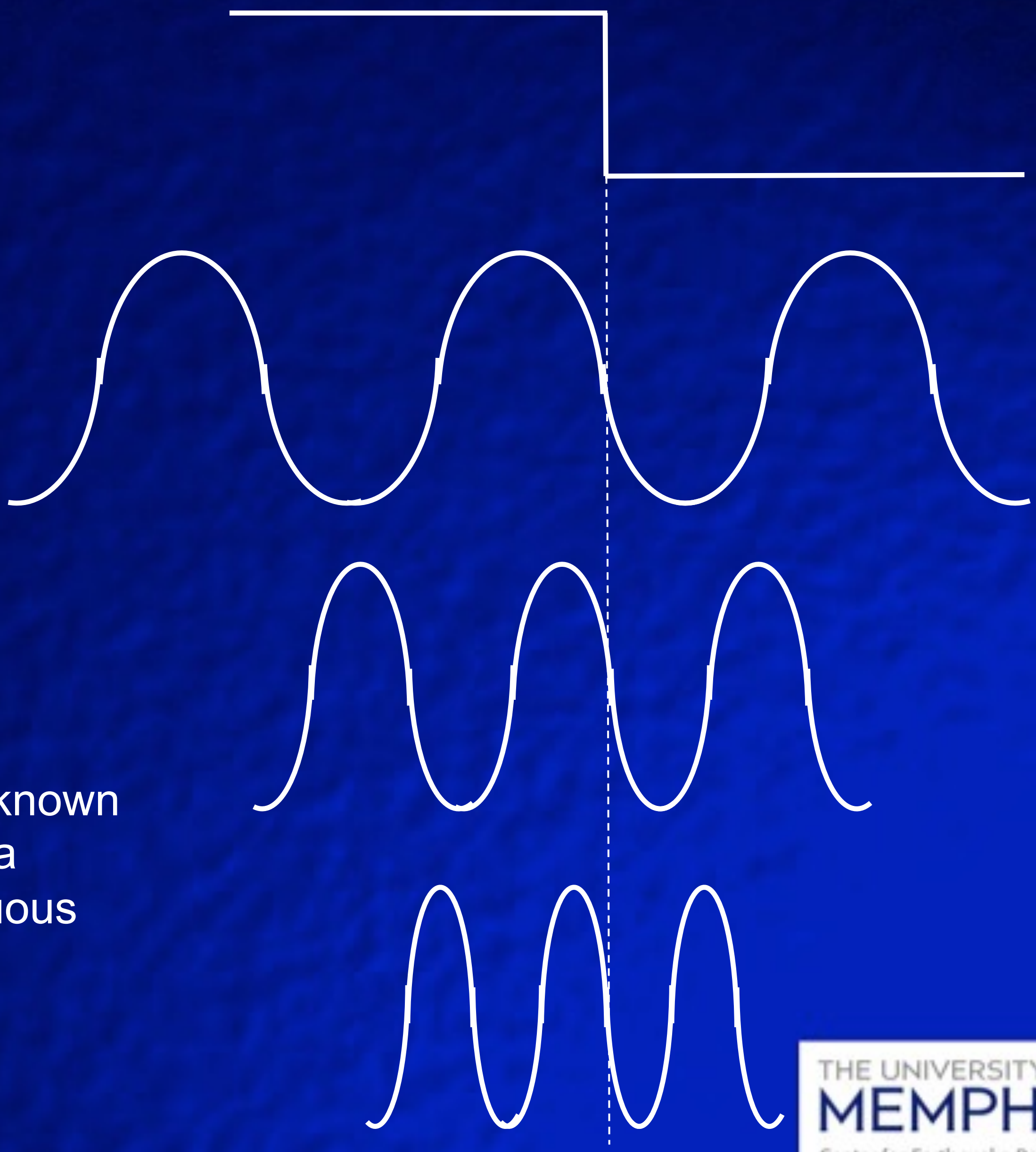
Consider how successive superpositions of precisely aligned (think phase) cosines sum to create a discontinuity.

Long periods are needed to make the flat parts and high frequencies are needed to construct the discontinuity.



THE UNIVERSITY OF MEMPHIS. Center for Earthquake Research and Information

Consider how successive superpositions of precisely aligned (think phase) cosines sum to create a discontinuity.



Long periods are needed to make the flat parts and high frequencies are needed to construct the discontinuity.

This leads us to a phenomena known as Gibbs Phenomena which is a result of constructing discontinuous signals with a finite number of discrete frequencies.

THE UNIVERSITY OF MEMPHIS
Center for Earthquake Research and Information

On the one hand, we need the lower frequencies to construct the flat portion of the function, and high frequencies aligned to create the discontinuity.

This creates overshoot at the discontinuity. We can make the overshoot more narrow by adding additional higher frequencies but the amplitude of the overshoot remains at 9%. This is Gibb's phenomena.

Run gibbs.m