ECE 3640 - Discrete-Time Signals and Systems
Hilbert Transform

Jake Gunther

Spring 2015

Department of Electrical & Computer Engineering

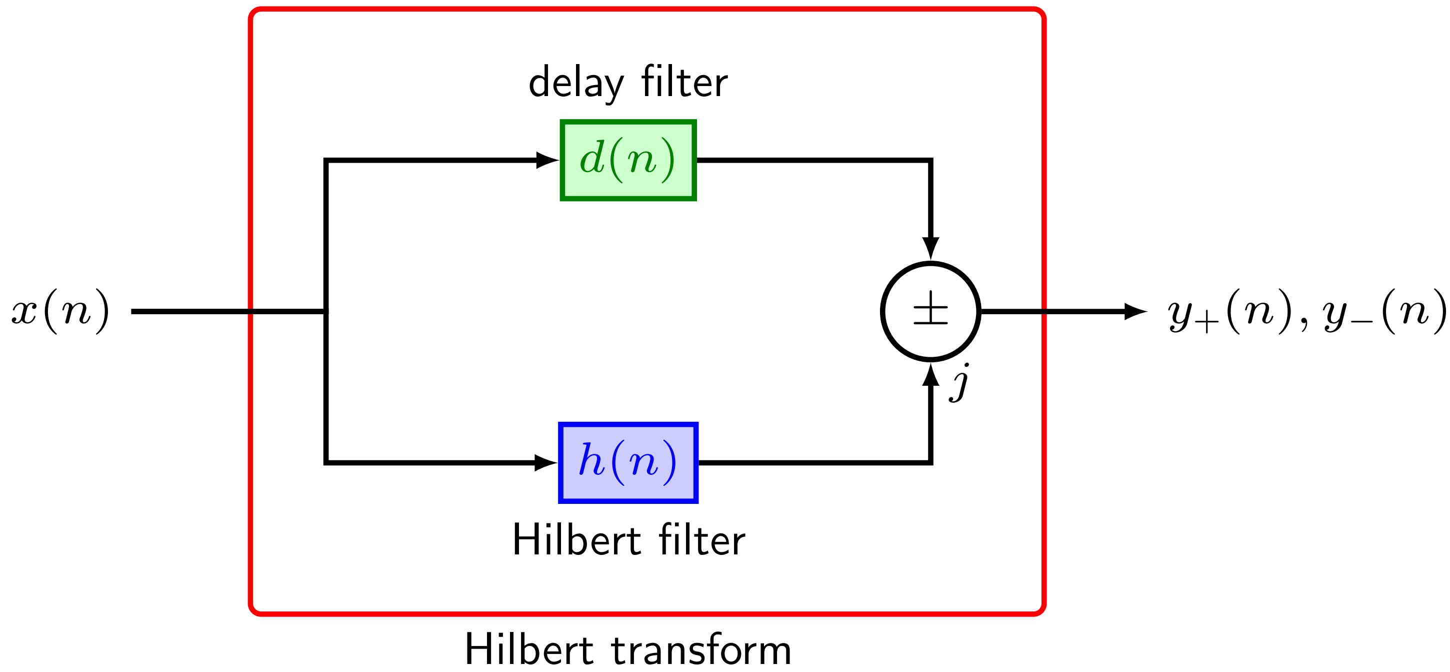
outline

- Hilbert transform
- envelope and phase
- phase unwrapping
- instantaneous frequency
- assignment

Hilbert transform

- the Hilbert transform is a filtering operation that consists of two filters
- the input $x(n)$ may be real or complex
- the output $y(n)$ is always complex
- $h(n)$ is the Hilbert filter
- $d(n)$ is a delay (all-pass) filter (same group delay as Hilbert filter)
- (see block diagram on the next page)

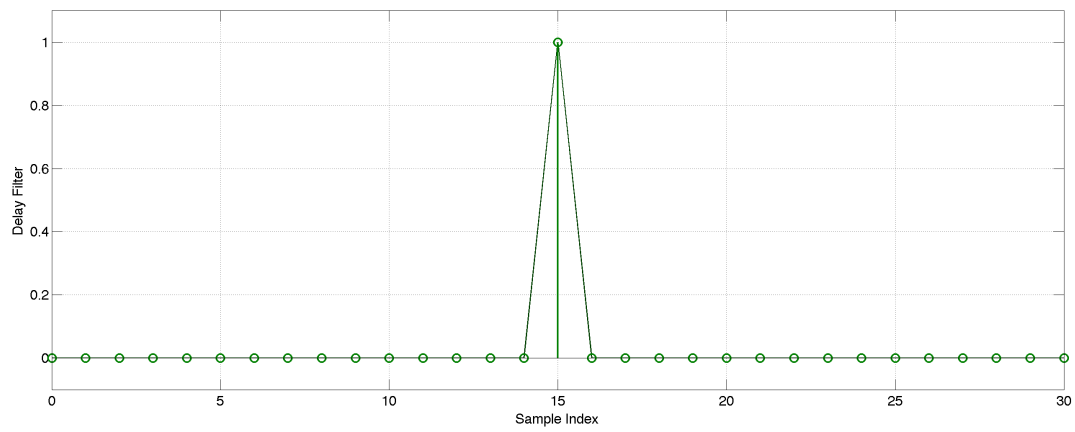
Hilbert transform block diagram

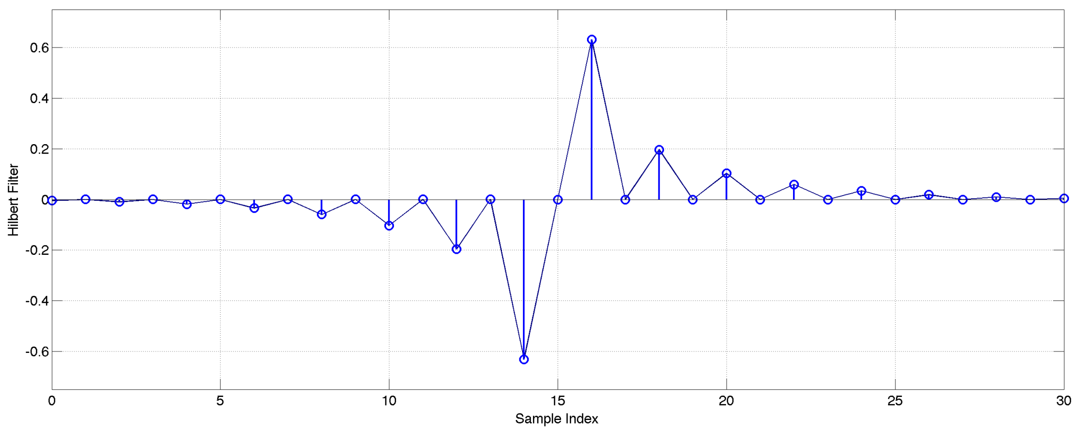


$$y_+(n) = [d(n) + jh(n)] * x(n) = [d(n) * x(n)] + j\,[x(n) * h(n)]$$
$$y_-(n) = [d(n) - jh(n)] * x(n) = [d(n) * x(n)] - j\,[x(n) * h(n)]$$

- a complex signal is really two real signals, i.e. a two channel signal
- (see filter impulse responses and frequency responses on the next few slides)

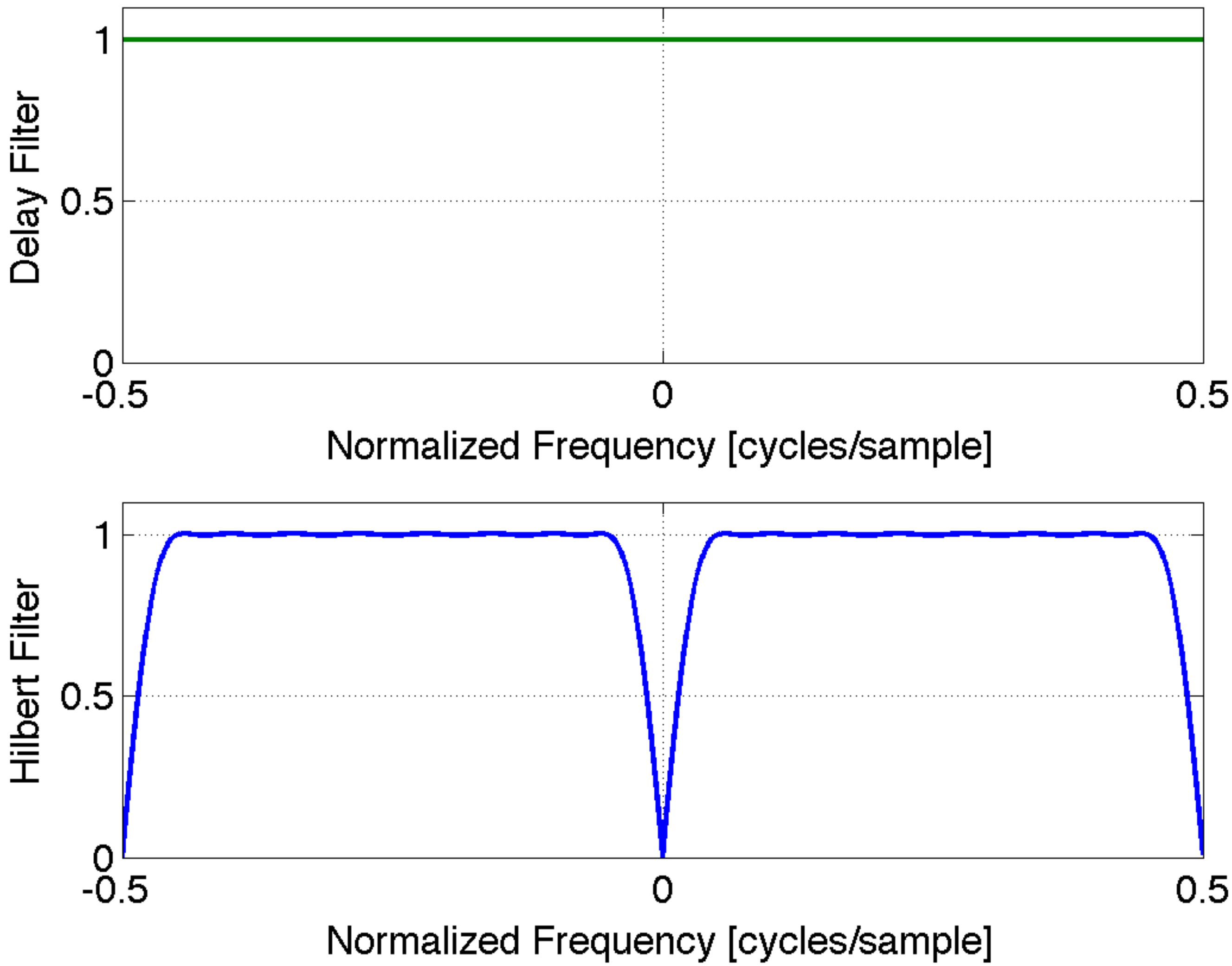
filter impulse responses



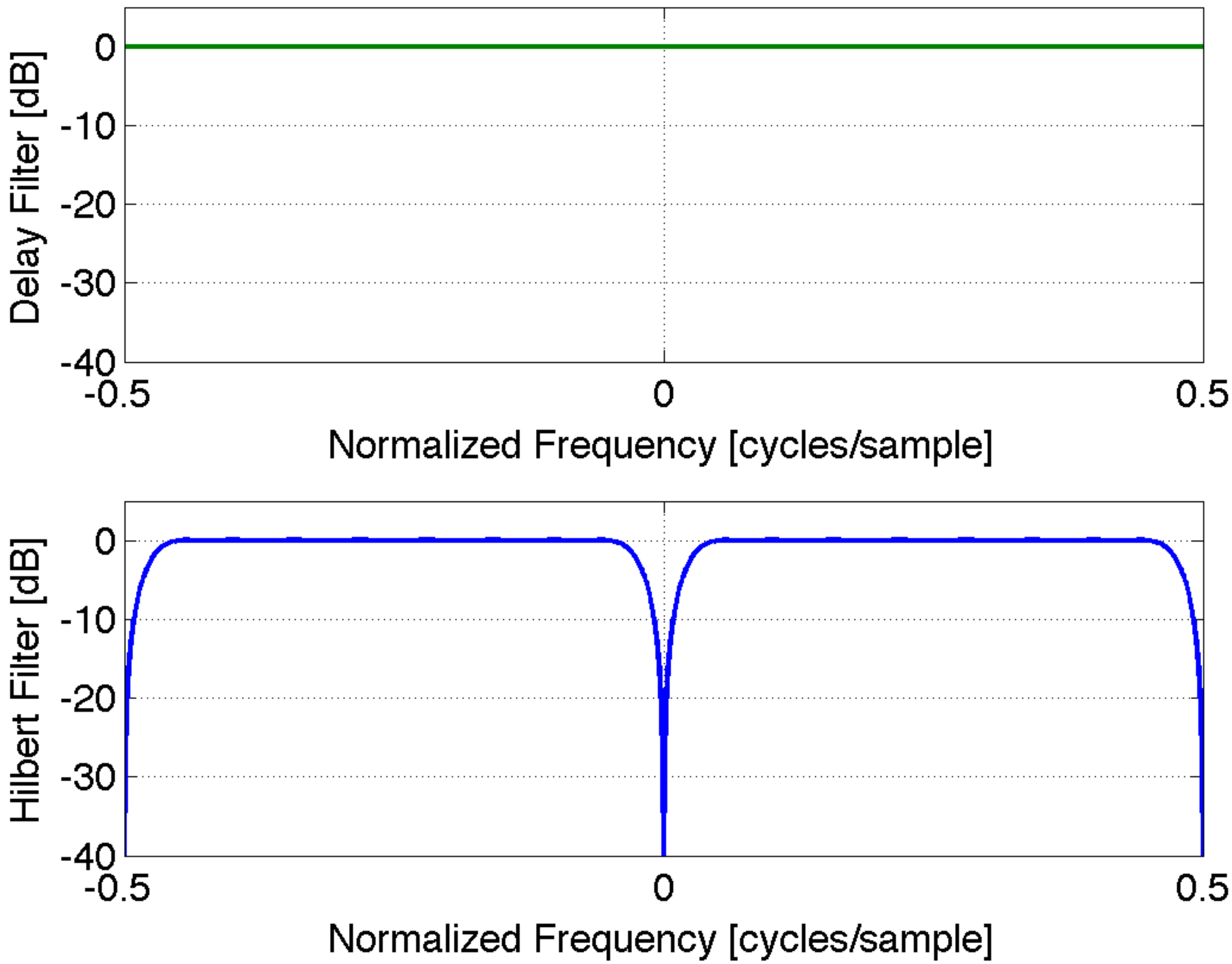


- Hilbert filter has odd length, odd symmetry about midpoint
- delay filter impulse located at symmetry point (zero crossing) of Hilbert filter

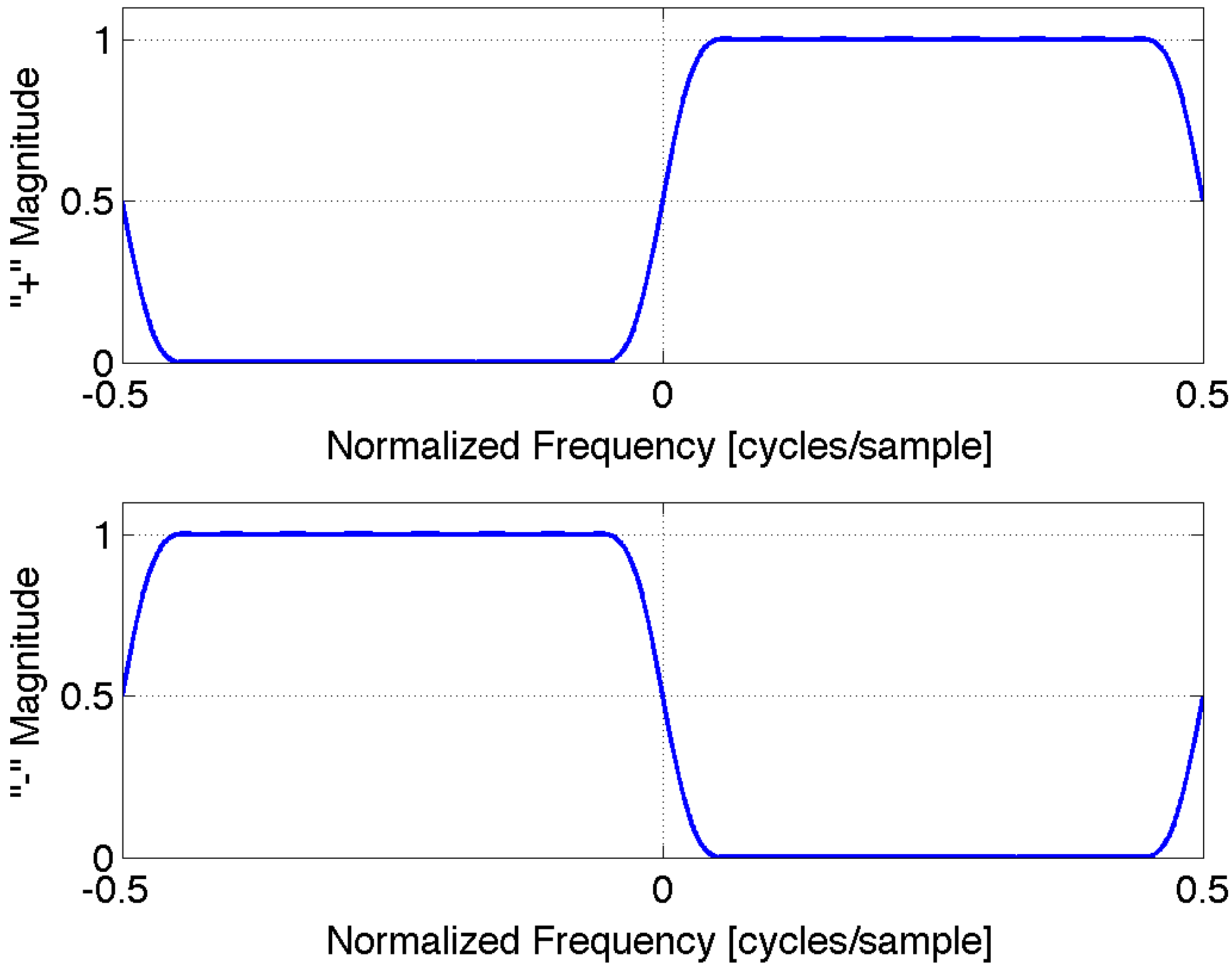
filter magnitude responses



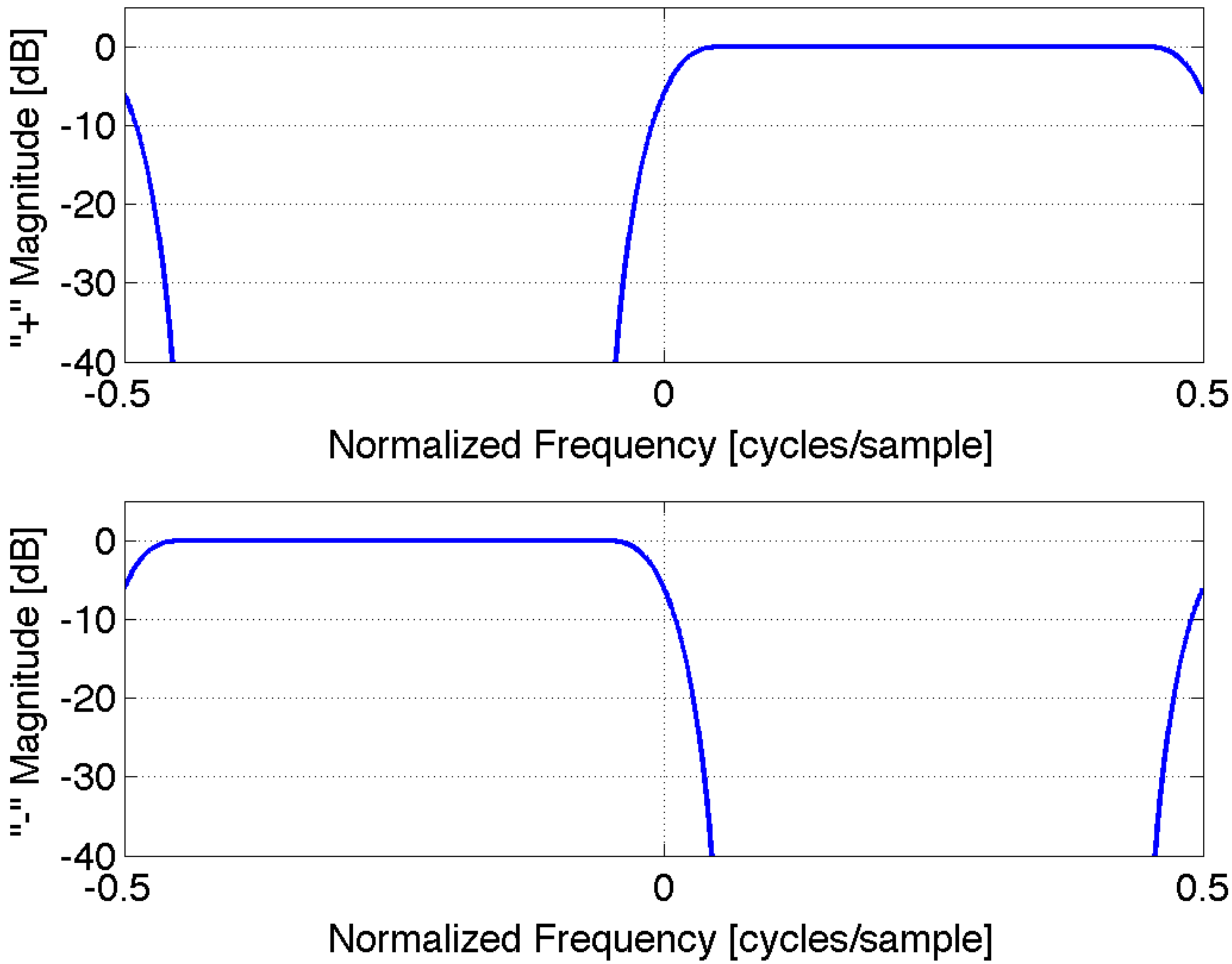
filter magnitude responses in dB



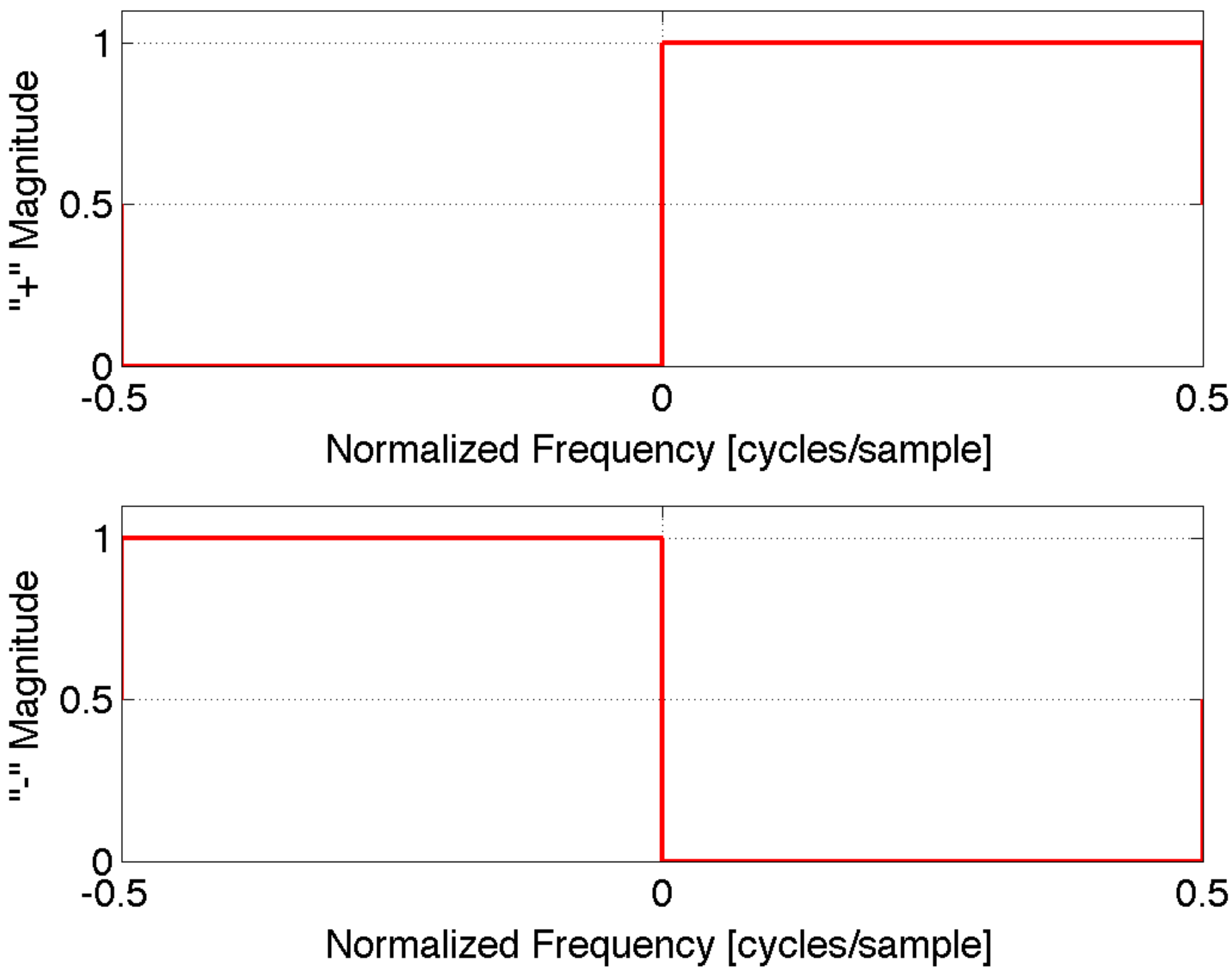
Hilbert magnitude responses



Hilbert magnitude responses in dB



ideal Hilbert magnitude responses



$$h[n] = \frac{2}{\pi}\frac{\sin^2(\pi n)}{n}, \qquad -\infty < n < \infty$$

action of Hilbert transform

- $[d(n) + jh(n)]$
  - pass positive frequency components
  - set negative frequency components to zero

- $[d(n) - jh(n)]$
  - pass negative frequency components
  - set positive frequency components to zero

- practical Hilbert transformers have transition bands

design Hilbert transform in Matlab

```
len = 321;
h = firpm(len-1,[0.005 0.995],[1 1],'Hilbert'); h = h(:);
d = zeros(len,1); d((len+1)/2) = 1;
```

applications of Hilbert transform

- radio receivers
- radar receivers
- single sideband modulation
- medical instruments (MRI, ultrasound, etc.)
- audio effects (pitch shifting)
- signal analysis
- if $x(n) = \cos(\omega n)$, then $y(n) = \cos(\omega n) \pm j\sin(\omega n) = \exp(\pm j\omega n)$
- if $x(n) = A(n)\cos\varphi(n)$, then $y(n) = A(n)\exp(j\varphi(n))$
- Hilbert transform enables decomposition of the input signal into its envelope $A(n)$ and phase $\varphi(n)$
- given the phase $\varphi(n)$, the instantaneous frequency can be computed by differentiation

$$f_i(t) = -\frac{1}{2\pi}\frac{d\varphi(t)}{dt} \quad \longleftrightarrow \quad f_i(n) \approx -\frac{f_s}{2\pi}[\varphi(n) - \varphi(n-1)]$$

extracting envelope and phase

1. given $x(n)$, compute the Hilbert transform $y(n)$

2. the envelope $A(n)$ is the magnitude of $y(n)$

$$A(n) = \sqrt{(\Re\{y(n)\})^2 + (\Im\{y(n)\})^2}$$

3. the phase $\varphi(n)$ is the unwrapped phase of $y(n)$

$$\varphi(n) = \tan^{-1}\frac{\Im\{y(n)\}}{\Re\{y(n)\}}, \qquad \text{(4-quadrant arc-tangent, atan2 in C)}$$

4. the instantaneous frequency is $f_i(n) = -(\varphi(n) - \varphi(n-1))(F_s/2\pi)$ [Hz]

5. the phase is unwrapped and instantaneous frequency computed by:

```
pz = atan2(yi,yr); // Compute the phase (four-quadrant arc-tangent)
fi = -(pz-pzold);  // Compute the phase difference (derivative approx.)
if     (fi >  M_PI) { fi -= 2.0*M_PI; } // Phase unwrap
else if(fi < -M_PI) { fi += 2.0*M_PI; } // Phase unwrap
pzold = pz;       // Save phase for next iteration
```

6. re-synthesize the signal by:

$$x(n) = \cos(\varphi(n))$$

```
x = cos(pz);
```

assignment

- the file to be processed is `fireflyintro.wav`
- design an odd length Hilbert filter and save it to a binary file (or use `hilbert_filter.bin`)
- write a C program to:
  - compute the real and imaginary components of the Hilbert transform of the input (do this by modifying the filtering program that you wrote previously)
  - use the `atan2` function to compute the phase of the Hilbert transformed signal
  - unwrap the phase
  - compute the instantaneous frequency
  - filter the instantaneous frequency through a 301-point normalized Gaussian filter (parameter $= 2$), see file `gaussian_2_filter.bin`
  - save the filtered instantaneous frequency
- make a spectrogram of the original signal ($N_{\mathsf{FFT}} = 2^{12}$, Hamming window, 90% overlap)
- overlay the filtered instantaneous frequency over the top of the spectrogram

- describe the relationship between the instantaneous frequency and the spectrogram

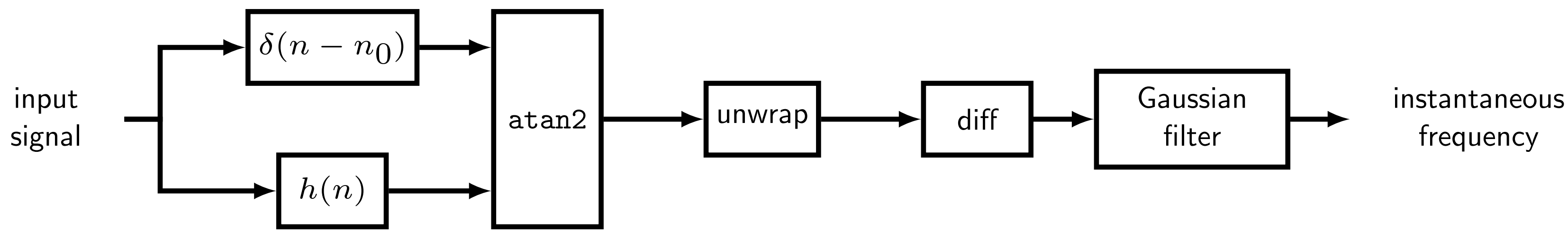
- plot the magnitude and phase response for the Hilbert filter $h(n)$ and the delay filter $d(n)$

- plot the magnitude and phase of the Hilbert transformer $d(n) + jh(n)$

- describe in words the action of the Hilbert transformer based on the frequency response

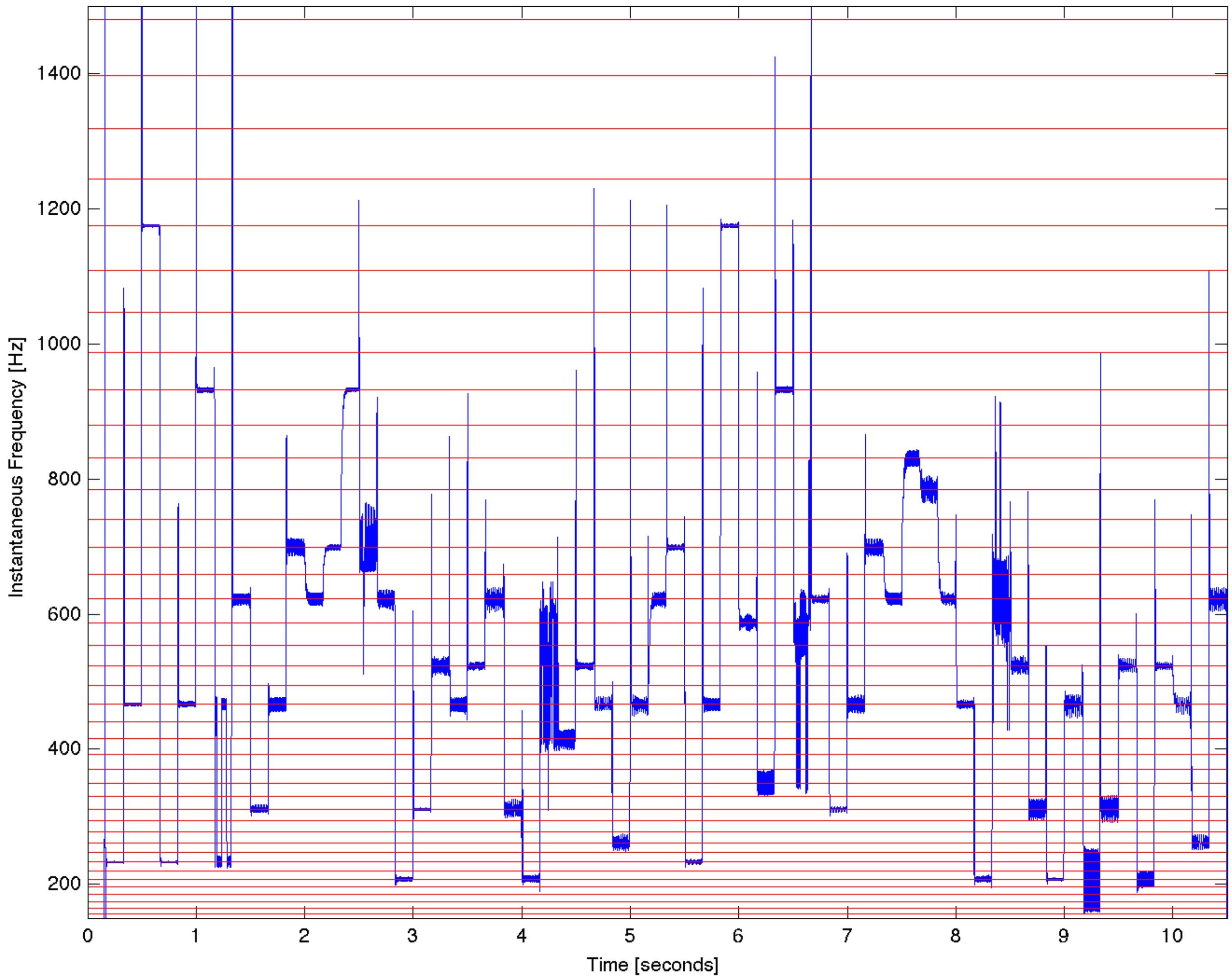
- plot the magnitude and phase of the cascade combination of the 1st-order difference filter and the Gaussian filter

processing diagram



- linear time invariant systems and their impulse responses
  - delay: $\delta(n-n_0)$
  - Hilbert filter: $h(n)$
  - 1st-difference (derivative approximation): $\delta(n)-\delta(n-1)$
  - Gaussian filter: normalized Gaussian function (`w=gausswin(301,2); w=w/sum(w);` in Matlab)
- other operations
  - calculate phase (four-quadrant arc-tangent): `atan2`
  - phase unwrapping

instantaneous frequency - linear scale - notes in red



instantaneous frequency - log scale - notes in red