Lecture 19: Cascaded LSI Systems

Mark Hasegawa-Johnson

ECE 401: Signal and Image Analysis

1. Review: DTFT, Square Wave, Ideal Filters, and DT Processing of CT Signals
2. Response of a Filter when the Input is Periodic
3. A Pure-Delay "Filter"
4. Example: Delaying a Square Wave
5. Cascaded LSI Systems
6. Summary

Outline

1. Review: DTFT, Square Wave, Ideal Filters, and DT Processing of CT Signals
2. Response of a Filter when the Input is Periodic
3. A Pure-Delay "Filter"
4. Example: Delaying a Square Wave
5. Cascaded LSI Systems
6. Summary

DTFT

The DTFT (discrete time Fourier transform) of any signal is $X(\omega)$, given by

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

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega$$

Properties worth knowing include:

- Time Shift: $x[n-n_0] \leftrightarrow e^{-j\omega n_0} X(\omega)$
- Filtering is Convolution:

$$y[n] = h[n] * x[n] \leftrightarrow Y(\omega) = H(\omega)X(\omega)$$

Square Wave, Rectangular Window

Periodic $x(t)$, with period T_0:

$$x(t) = \begin{cases} 1 & -\frac{L}{2} \le n \le \frac{L}{2} \\ 0 & \text{otherwise} \end{cases}$$

$$X_k = \begin{cases} \frac{L}{T_0} & k = 0 \\ \frac{\sin(\pi k L/T_0)}{\pi k} & k \neq 0 \end{cases}$$

Aperiodic $x[n]$:

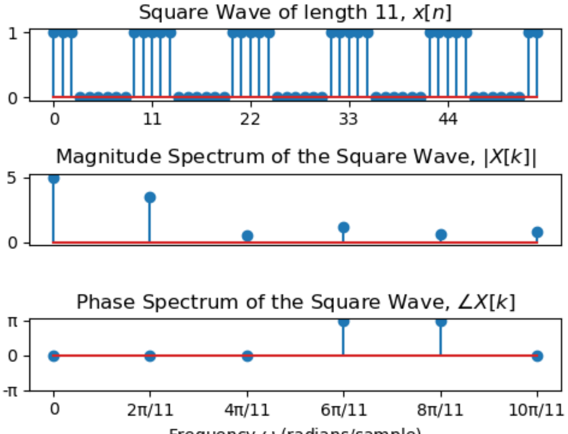
$$d_L[n] = \begin{cases} 1 & -\frac{(L-1)}{2} \le n \le \frac{(L-1)}{2} \\ 0 & \text{otherwise} \end{cases}$$

$$D_L(\omega) = \frac{\sin(\omega L/2)}{\sin(\omega/2)}$$

Discrete-Time Square Wave, Rectangular Window

Periodic $x[n]$, with period T_0:

$$x[n] = \begin{cases} 1 & -\frac{L-1}{2} < n < \frac{L-1}{2} \\ 0 & \text{otherwise} \end{cases} \quad \leftrightarrow \quad X_k = \begin{cases} \frac{L}{T_0} & k = 0 \\ \frac{\sin(\pi k L/T_0)}{T_0 \sin(\pi k/T_0)} & k \neq 0 \end{cases}$$



Ideal Lowpass Filter

$$H_{LP}(\omega) = \begin{cases} 1 & |\omega| \le \omega_c, \\ 0 & \omega_c < |\omega| \le \pi. \end{cases} \quad \leftrightarrow \quad h_{LP}[m] = \frac{\omega_c}{\pi}\text{sinc}(\omega_c n)$$

Review: DT Processing of CT Signals

A bandlimited periodic signal $x(t)$ can be sampled, filtered, then sinc-interpolated to create:

$$x(t) = \sum_{k=-N}^{N} X_k e^{j2\pi kF_0 t}$$

$$x[n] = \sum_{k=-N}^{N} X_k e^{jk\omega_0 n},$$

$$y[n] = \sum_{k=-N}^{N} Y_k e^{jk\omega_0 n},$$

$$y(t) = \sum_{k=-N}^{N} Y_k e^{j2\pi kF_0 n},$$

where $\omega_0 = \frac{2\pi F_0}{F_s}$, and $N = \lfloor \frac{F_s/2}{F_0} \rfloor$, and $Y_k = H(k\omega_0)X_k$.

Outline

1. Review: DTFT, Square Wave, Ideal Filters, and DT Processing of CT Signals
2. Response of a Filter when the Input is Periodic
3. A Pure-Delay "Filter"
4. Example: Delaying a Square Wave
5. Cascaded LSI Systems
6. Summary

Response of a Filter when the Input is Periodic

Now we're ready to ask this question:

What is the output of a filter when the input, $x[n]$, is periodic with period N_0?

Response of a Filter when the Input is Periodic

1. **Fourier Series:** If the input is periodic, then we can write it as

$$x[n] = \sum_{k=-N_0/2}^{(N_0-1)/2} X_k e^{j2\pi kn/N_0}$$

2. **Frequency Response:** If the input is $e^{j\omega n}$, then the output is

$$y[n] = H(\omega)e^{j\omega n}$$

3. **Linearity (of convolution, and of frequency response):** If the input is $x_1[n] + x_2[n]$, then the output is

$$y[n] = y_1[n] + y_2[n]$$

Response of a Filter when the Input is Periodic

Putting all those things together, if the input is

$$x[n] = \sum_{k=-N_0/2}^{(N_0-1)/2} X_k e^{j2\pi kn/N_0}$$

...then the output is

$$y[n] = \sum_{k=-N_0/2}^{(N_0-1)/2} X_k H(k\omega_0) e^{j2\pi kn/N_0}$$

...where $\omega_0 = \frac{2\pi}{N_0}$ is the fundamental frequency.

Outline

1. Review: DTFT, Square Wave, Ideal Filters, and DT Processing of CT Signals
2. Response of a Filter when the Input is Periodic
3. **A Pure-Delay “Filter”**
4. Example: Delaying a Square Wave
5. Cascaded LSI Systems
6. Summary

A Pure-Delay "Filter"

One thing we can do to a signal is to just delay it, by n_0 samples:

$$y[n] = x[n - n_0]$$

Even this very simple operation can be written as a convolution:

$$y[n] = g[n] * x[n]$$

where the "filter," $g[n]$, is just

$$g[n] = \delta[n - n_0] = \begin{cases} 1 & n = n_0 \\ 0 & \text{otherwise} \end{cases}$$

Frequency Response of A Pure-Delay "Filter"

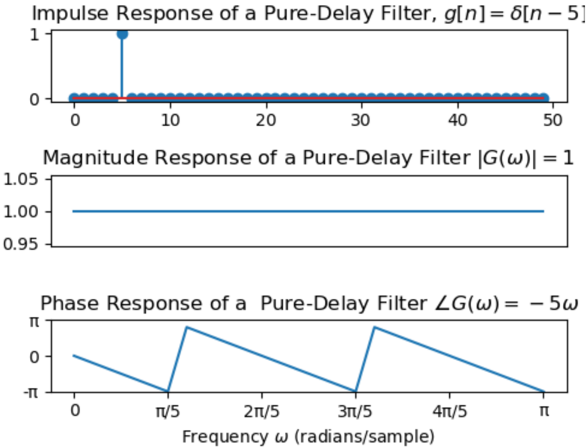
$$g[n] = \begin{cases} 1 & n = n_0 \\ 0 & \text{otherwise} \end{cases}$$

The frequency response is

$$G(\omega) = \sum_m g[m] e^{-j\omega m} = e^{-j\omega n_0}$$

Impulse Response of A Pure-Delay "Filter"

Here is the impulse response of a pure-delay "filter" (and the magnitude and phase responses, which we'll talk about next).



Magnitude and Phase Response of A Pure-Delay "Filter"

$$G(\omega) = \sum_m g[m] e^{-j\omega m} = e^{-j\omega n_0}$$

Notice that the magnitude and phase response of this filter are

$$|G(\omega)| = 1$$
$$\angle G(\omega) = -\omega n_0$$

So, for example, if have an input of $x[n] = \cos(\omega n)$, the output would be

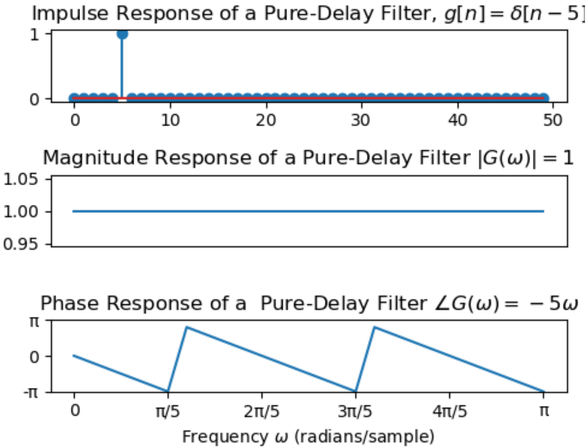
$$y[n] = |G(\omega)| \cos\left(\omega n + \angle G(\omega)\right) = \cos\left(\omega n - \omega n_0\right)$$

Outline

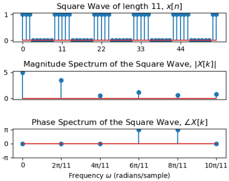
1. Review: DTFT, Square Wave, Ideal Filters, and DT Processing of CT Signals
2. Response of a Filter when the Input is Periodic
3. A Pure-Delay "Filter"
4. Example: Delaying a Square Wave
5. Cascaded LSI Systems
6. Summary

Magnitude and Phase Response of A Pure-Delay "Filter"

Here are the magnitude and phase response of the pure delay filter.



Spectrum of a Square Wave



Here are the Fourier series coefficients of an $N_0 = 11$, $L = 5$, even-symmetric square wave:

$$X_k = \left.\frac{\sin(\omega L/2)}{\sin(\omega/2)}\right|_{\omega=\frac{2\pi k}{N_0}} = \frac{\sin(2\pi kL/N_0)}{\sin(2\pi k/N_0)}$$

Response of a Filter when the Input is Periodic

And here's what happens when we pass a periodic signal through a filter $g[n]$:

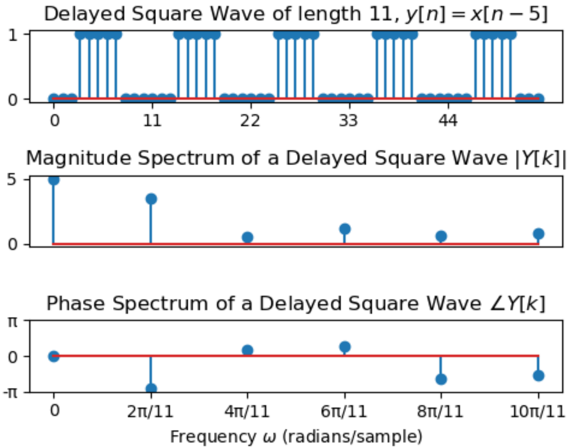
$$x[n] = \sum_{k=-N_0/2}^{(N_0-1)/2} X_k e^{j2\pi kn/N_0}$$

$$y[n] = \sum_{k=-N_0/2}^{(N_0-1)/2} X_k G(k\omega_0) e^{j2\pi kn/N_0}$$

...where $\omega_0 = \frac{2\pi}{N_0}$ is the fundamental frequency.

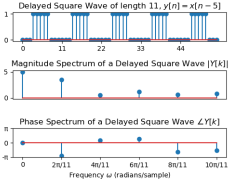
Spectrum: Delayed Square Wave

And here's the result. This is the square wave, after being delayed by the pure-delay filter:



You can see that magnitude's unchanged, but phase is changed.

Spectrum of a Delayed Square Wave



The Fourier series coefficients of a square wave, delayed by n_0 samples, are

$$Y_k = \frac{\sin(kL\omega_0/2)}{\sin(k\omega_0/2)} e^{-jk\omega_0 n_0}$$

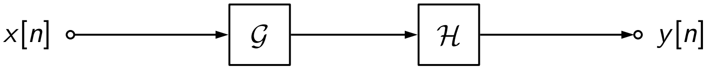
where $\omega_0 = \frac{2\pi}{N_0}$.

Outline

1. Review: DTFT, Square Wave, Ideal Filters, and DT Processing of CT Signals
2. Response of a Filter when the Input is Periodic
3. A Pure-Delay "Filter"
4. Example: Delaying a Square Wave
5. Cascaded LSI Systems
6. Summary

Cascaded LSI Systems

What happens if we pass the input through two LSI systems, in cascade?



Cascaded filters

Suppose I pass the signal through filter $g[n]$, then pass it through another filter, $h[n]$:

$$y[n] = h[n] * (g[n] * x[n]),$$

we get a signal $y[n]$ whose spectrum is:

$$Y(\omega) = H(\omega)G(\omega)X[k]$$

Convolution is Commutative

Notice that

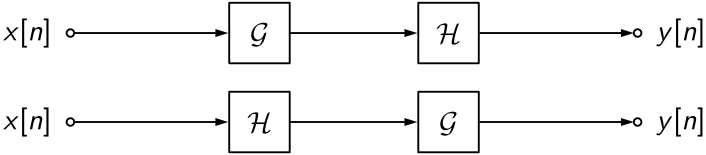
$$Y(\omega) = H(\omega)G(\omega)X(\omega) = G(\omega)H(\omega)X(\omega)$$

and therefore:

$$y[n] = h[n] * (g[n] * x[n]) = g[n] * (h[n] * x[n])$$

Convolution is Commutative

Since convolution is commutative, these two circuits compute exactly the same output:



Quiz

Go to the course webpage, and try the quiz!

Example: Differenced Square Wave

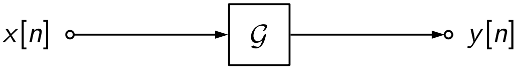
Suppose we define $x[n]$ to be an 11-sample square wave, $g[n]$ to be a delay, and $h[n]$ to be a first difference:

$$x[n] = \begin{cases} 1 & -2 \le n \le 2 \\ 0 & 3 \le n \le 8 \end{cases}$$

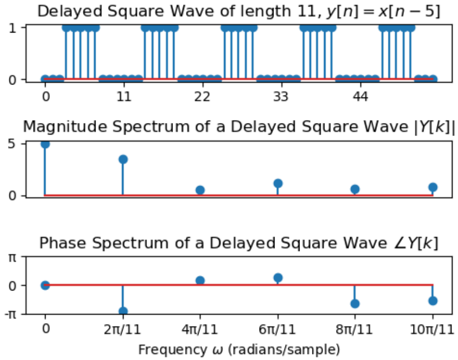
$$x[n] \stackrel{\mathcal{G}}{\longrightarrow} z[n] = x[n-5]$$

$$z[n] \stackrel{\mathcal{H}}{\longrightarrow} y[n] = z[n] - z[n-1]$$

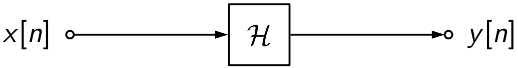
Delayed Square Wave



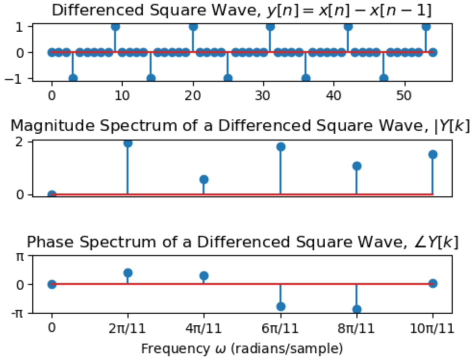
Here's what we get if we just **delay** the square wave:



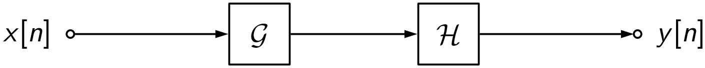
Differenced Square Wave



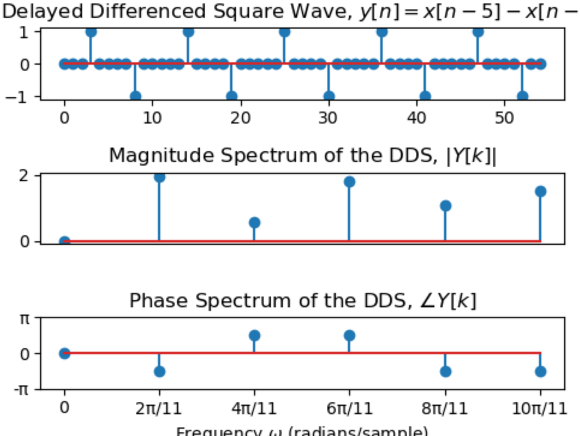
Here's what we get if we just **difference** the square wave:



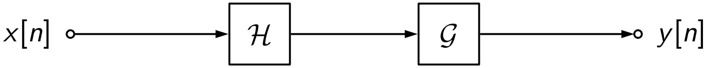
Example: Differenced Delayed Square Wave



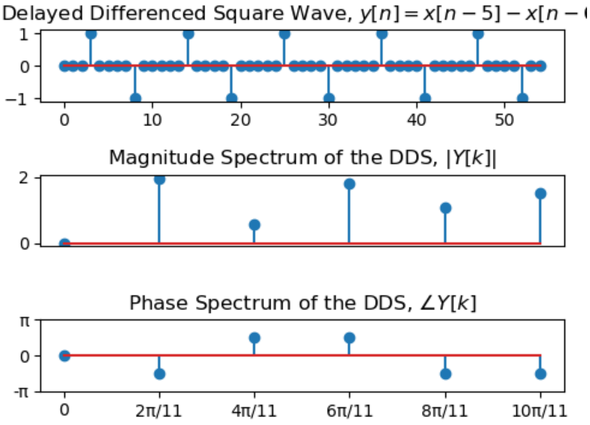
Here's what we get if we **delay** and then **difference** the square wave:



Example: Delayed Differenced Square Wave



Here's what we get if we **difference** and then **delay** the square wave (hint: it's exactly the same as the previous slide!!)



Magnitude and Phase of Cascaded Frequency Responses

In general, when you cascade two LSI systems, the magnitudes multiply:

$$|Y_k| = |H(\omega)||G(\omega)||X_k|,$$

but the phases add:

$$\angle Y_k = \angle H(\omega) + \angle G(\omega) + \angle X_k$$

That's because:

$$H(\omega)G(\omega) = |H(\omega)|e^{j\angle H(\omega)}|G(\omega)|e^{j\angle G(\omega)} = |H(\omega)||G(\omega)|e^{j(\angle H(\omega)+\angle G(\omega)}$$

Outline

1. Review: DTFT, Square Wave, Ideal Filters, and DT Processing of CT Signals
2. Response of a Filter when the Input is Periodic
3. A Pure-Delay "Filter"
4. Example: Delaying a Square Wave
5. Cascaded LSI Systems
6. Summary

Summary

- **Periodic inputs:** If the input of an LSI system is periodic,

$$x[n] = \sum_{k=-N_0/2}^{(N_0-1)/2} X_k e^{j2\pi kn/N_0}$$

...then the output is

$$y[n] = \sum_{k=-N_0/2}^{(N_0-1)/2} X_k H(k\omega_0) e^{j2\pi kn/N_0}$$

- **Cascaded LTI Systems** convolve their impulse responses, equivalently, they multiply their frequency responses:

$$y[n] = h[n] * g[n] * x[n], \quad Y_k = H(k\omega_0)G(k\omega_0)X_k$$