All-pass and Fractional-delay Filters

Ivan W. Selesnick

October 30, 2019

1 All-pass filters

Consider a recursive digital filter with a transfer function of the form,

$$H(z) = \frac{z^{-L}\, D(1/z)}{D(z)}. \tag{1}$$

For example, if

$$D(z) = 1 + d_1 z^{-1} + d_2 z^{-2}$$

and $L = 2$, then

$$H(z) = \frac{d_2 + d_1 z^{-1} + z^{-2}}{1 + d_1 z^{-1} + d_2 z^{-2}}.$$

For a transfer function of this form, the numerator coefficients are the denominator coefficients but in reverse order.

Note that

$$H(z)\, H(1/z) = \frac{z^{-L}\, D(1/z)}{D(z)}\, \frac{z^{L}\, D(z)}{D(1/z)} \tag{2}$$

$$= 1. \tag{3}$$

Equivalently, in the time-domain:

$$h(n) * h(-n) = \delta(n)$$

and, equivalently, in the frequency-domain:

$$H^f(\omega)\, \overline{H^f(\omega)} = 1$$

which implies that $|H^f(\omega)|^2 = 1$ and

$$|H^f(\omega)| = 1.$$

Therefore, discrete-time LTI systems of the form (1) are *all-pass* systems. Such systems pass all frequencies with equal amplitude.

2 Fractional-delay filters

The delay of a discrete-time signal by a whole number of samples is simple — it requires only a enough delay elements. However, some applications require that a signal be delayed by a *fractional* number of samples, like 3.3 samples.

One way to delay a discrete-time signal by a fractional number of samples is to convert the discrete-time signal into a continuous-time (analog) signal, then delay the analog signal by the correct number of seconds, and then resample the delayed analog signal to obtain a new discrete-time signal. In this case, one needs to design a continuous-time system, which presents its own problem; and furthermore, the conversion of the discrete-time signal to an analog signal and back to a discrete-time signal is inefficient and unnecessary. The fractional-delay can be accomplished entirely with a fully discrete-time LTI system.

How can we design an discrete-time LTI system that delays a discrete-time signal by a fractional number of samples?

Consider a discrete-time signal $x(n)$ and its DTFT $X(\omega)$,

$$x(n) \quad \longleftrightarrow \quad X(\omega).$$

When n_1 is an integer, the shift-property of the discrete-Fourier transform states that:

$$x(n-n_1) \quad \longleftrightarrow \quad X(\omega)\,\mathrm{e}^{-\mathrm{j}n_1\omega}, \qquad n_1 \in \mathbb{Z}.$$

The property can be extended to non-integer shifts in order to define a non-integer signal delay:

$$x(n-\tau) \quad \longleftrightarrow \quad X(\omega)\,\mathrm{e}^{-\mathrm{j}\tau\omega}, \qquad |\omega| < \pi, \quad \tau \in \mathbb{R}$$

Therefore, the frequency response of the (ideal) LTI fractional-delay system is

$$H(\omega) = \mathrm{e}^{-\mathrm{j}\tau\omega}, \quad |\omega| < \pi. \tag{4}$$

Note that this is an all-pass system because $|H(\omega)| = 1$. The phase response is

$$\angle H(\omega) = -\tau\omega, \quad |\omega| < \pi.$$

and the group delay response is $H_{\mathrm{grpd}}(\omega) = \tau$. The impulse response corresponding to (4) is

$$h(n) = \mathrm{sinc}(n-\tau),$$

illustrated in Fig. 1. Being samples of the sinc function, the impulse response is infinite in both negative and positive directions. Therefore the system is non-causal. Like the ideal low-pass filter, *the ideal fractional-delay system can not be exactly realized.*

The impulse response for $\tau = 3.3$ is illustrated in Fig. 1(c) as samples of the shifted sinc function. Note that when τ is an integer, then sampling the shifted sinc function gives the delta function, $\delta(n)$.

In practice, we use an FIR or IIR filter that is designed to approximate the ideal fractional-delay system. As the ideal fractional-delay system is an all-pass system, it is quite natural to use a recursive all-pass filter to approximate the ideal fractional-delay filter. (Although, FIR filters can also be used, for example see [1] and references therein.)

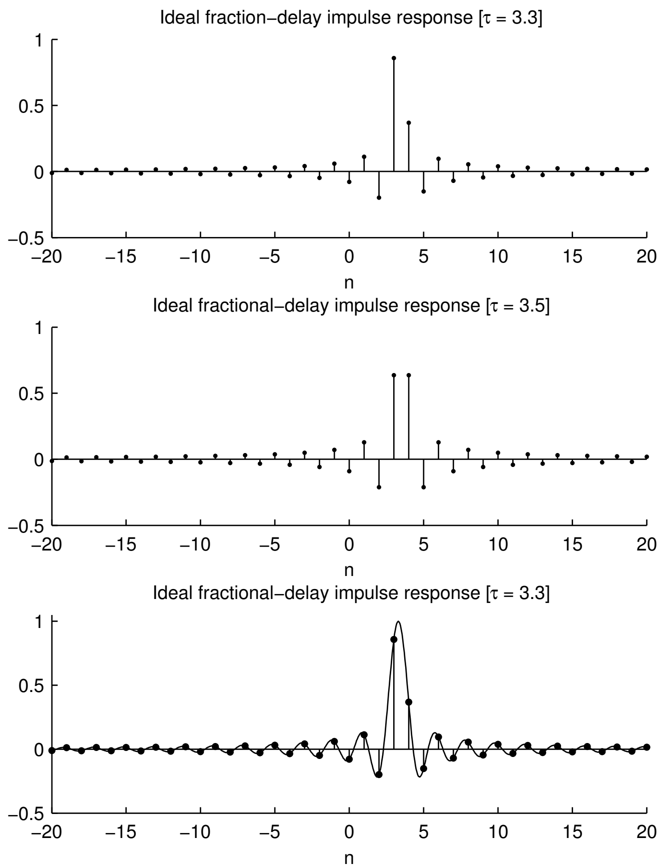

Figure 1: The impulse responses of the ideal fractional-delay system for a delay of (a) 3.3 and (b) 3.5 samples. The impulse response of the ideal fractional-delay system as samples of a shifted-sinc function (c).

3 Fractional-delay all-pass filters

The ideal fractional-delay system is a specific kind of all-pass filter. Several authors have addressed the design of discrete-time all-pass systems that approximate a fractional delay [1, 2, 3]. The following formula for the maximally-flat delay all-pass filter is adapted from Thiran's formula for the maximally-flat delay all-pole filter [4]. The maximally-flat approximation to a delay of τ samples is given by

$$H(z) = \frac{z^{-L}\, D(1/z)}{D(z)}$$

where

$$D(z) = 1 + \sum_{n=1}^{L} d(n)\, z^{-n}$$

with

$$d(n) = (-1)^n \binom{L}{n} \frac{(\tau - L)_n}{(\tau + 1)_n} \tag{5}$$

where $\binom{L}{n}$ is the binomial coefficient:

$$\binom{L}{n} = \frac{L!}{n!\,(L-n)!},$$

and $(x)_n$ represents the rising factorial:

$$(x)_n := \underbrace{(x)\,(x+1)\,\cdots\,(x+n-1)}_{n\text{ terms}}.$$

With this $D(z)$ we have the approximation

$$H^f(\omega) \approx \mathrm{e}^{-j\omega\tau} \quad \text{around } \omega = 0$$

or equivalently,

$$H(z) \approx z^{-\tau} \quad \text{around } z = 1.$$

The coefficients $d(n)$ in (5) can be computed very efficiently using the ratio:

$$\frac{d(n+1)}{d(n)} = -\frac{\binom{L}{n+1}}{\binom{L}{n}} \cdot \frac{(\tau-L)_{n+1}}{(\tau-L)_n} \cdot \frac{(\tau+1)_n}{(\tau+1)_{n+1}} \tag{6}$$

$$= \frac{(L-n)(L-n-\tau)}{(n+1)(n+1+\tau)} \tag{7}$$

From this ratio, it follows that the filter $d(n)$ can be generated as:

$$d(0) = 1 \tag{8}$$

$$d(n+1) = d(n) \cdot \frac{(L-n)(L-n-\tau)}{(n+1)(n+1+\tau)}, \quad 0 \leqslant n \leqslant L-1. \tag{9}$$

This can be implemented in Matlab as:

```
n = 0:L-1;
d = cumprod([1, (L-n).*(L-n-tau)./(n+1)./(n+1+tau)]);
```

For example, with $\tau = 3.5$ and $L = 3$, we have

$$d(n) = \{1,\ -1/3,\ 1/11,\ -5/429\}, \quad \text{for} \quad n = 0, 1, 2, 3.$$

The all-pass filter $H(z) = z^{-3}D(1/z)/D(z)$ is illustrated in Figs. 3 and 2.

3.1 Example

To illustrate the effect of a fractional-delay all-pass filter when it is applied to a signal, it is applied to the test signal in Fig. 4. The test signal is a cosine pulse with frequency 0.2π, created in MATLAB using the command:

```
x = blackman(N) .* cos(0.2*pi*(0:N-1)');
```

When we filter the test signal using the all-pass system illustrated in Fig. 3, we obtain the output signal illustrated in the figure. As the figure illustrates, the output signal has the same shape as the input signal, except it is delayed by 3.5 samples.

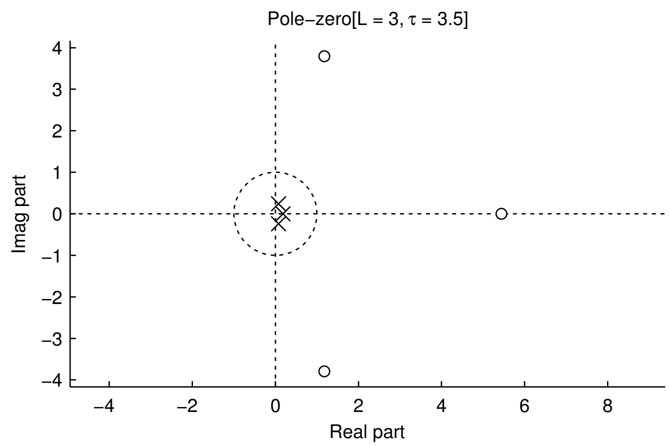

Figure 2: Pole-zero diagram.

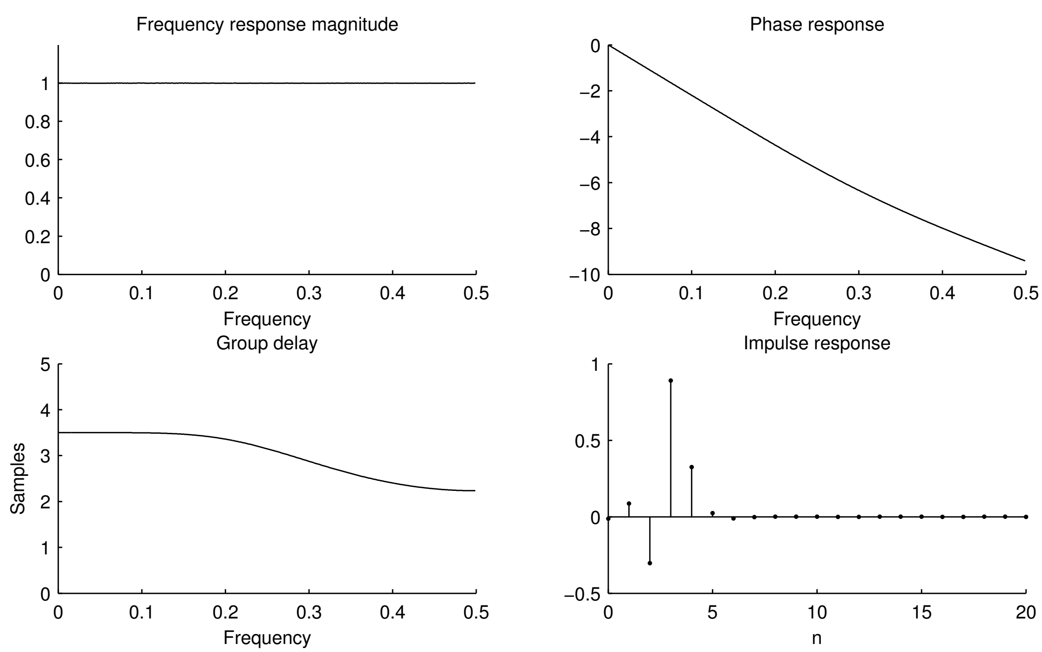

Figure 3: Frequency and impulse responses of the maximally-flat fractional-delay all-ass filter with $L = 3$ and $\tau = 3.5$.

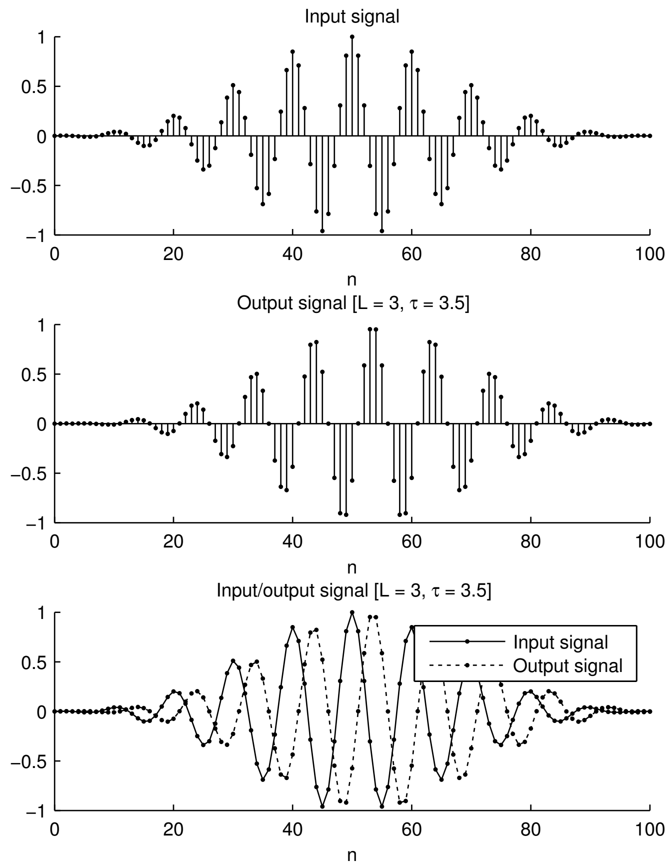

Figure 4: Test signal and output signal of all-pass filter illustrated in Fig. 3 (with $L = 3$ and $\tau = 3.5$).

4 Exercise

Ideal fractional delay system. The ideal fractional delay system has the frequency response

$$H^f(\omega) = \mathrm{e}^{-\mathrm{j}\tau\omega} \tag{10}$$

for $|\omega|| < \pi$. This system delays the input signal by τ samples. The value of τ does not have to be an integer.

a) Sketch the magnitude and phase response for $|\omega| < 2\pi$.

b) Use the inverse DTFT to find the impulse response $h(n)$ of the ideal fractional delay system with parameter τ.

c) Can the ideal fractional delay system be implemented using a difference equation? Explain.

References

[1] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine. Splitting the unit delay. *IEEE Signal Processing Magazine*, 13(1):30–60, January 1996.

[2] M. Lang. Allpass filter design and applications. *IEEE Trans. Signal Process.*, 46(9):2505–2514, September 1998.

[3] H. W. Schüßler and P. Steffen. On the design of allpasses with prescribed group delay. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume 3, pages 1313–1316, Albuquerque, April 3-6 1990.

[4] J. P. Thiran. Recursive digital filters with maximally flat group delay. *IEEE Trans. on Circuit Theory*, 18(6):659–664, November 1971.