

A Method for Finding the Phase Shift and Amplitude Response of a Digital Filter

S. A. Tretter
October 21, 2020

Last week in class a method for determining the phase shift of a digital filter from the time delay of the output relative to the input sine waves. It may be difficult to determine this time shift accurately since the input and output signals are sequences of samples. This document presents a method of accurately determining the phase shift and amplitude response of the filter.

Suppose one input to the filter is $x_c(nT) = \cos \omega nT$ and the resulting output is $y_c(nT)$. Let another input be $x_s(nT) = \sin \omega nT$ with the corresponding output $y_s(nT)$. The input can be represented using complex notation as

$$x(nT) = \cos \omega nT + j \sin \omega nT = e^{j\omega nT} \quad (1)$$

From sinusoidal steady state analysis we know that the output is the complex signal

$$y(nT) = y_c(nT) + jy_s(nT) = H(\omega)e^{j\omega nT} = A(\omega)e^{j\theta(\omega)}e^{j\omega nT} \quad (2)$$

Now let

$$z(nT) = z_R(nT) + jz_I(nT) = y(nT)\overline{x(nT)} = y(nT)e^{-j\omega nT} = A(\omega)e^{j\theta(\omega)} \quad (3)$$

You know the sequences $\cos \omega nT$, $\sin \omega nT$, $y_c(nT)$, and $y_s(nT)$. So you can compute

$$\begin{aligned} z(nT) &= [y_c(nT) + jy_s(nT)][\cos \omega nT - j \sin \omega nT] \\ &= [y_c(nT) \cos \omega nT + y_s(nT) \sin \omega nT] + j[y_s(nT) \cos \omega nT - y_c(nT) \sin \omega nT] \end{aligned} \quad (4)$$

Thus

$$z_R(nT) = y_c(nT) \cos \omega nT + y_s(nT) \sin \omega nT \quad (5)$$

and

$$z_I(nT) = y_s(nT) \cos \omega nT - y_c(nT) \sin \omega nT \quad (6)$$

and

$$\theta(\omega) = \arctan[z_I(nT)/z_R(nT)] \quad (7)$$

Notice that the time nT does not appear in the final result. Of course, you need to let the filter output reach steady state before computing the phase shift angle.

The magnitude of the input $x(nT) = e^{j\omega nT}$ is 1. Thus the squared amplitude response is just

$$A^2(\omega) = |y(nT)|^2 = y_c^2(nT) + y_s^2(nT) \quad (8)$$

MATLAB can do complex arithmetic. The equations above have expanded the complex computations in terms of the real signals. You could let MATLAB do this for you.