Lecture 21. Tuesday 15th December, 2020.

0.1 Comb filters

Comb filters

The simple filters discussed so far are characterized either by a single passband and/or a single stopband. There are applications where these kind of stuff is not sufficient, so filters with multiple passbands and stopbands are required. The so called **comb filters** are an example of such instruments.
A comb filter is an LTI digital filter such that:

- its output to a scaled sum of input digital signals is equal to the scaled sum of the outputs to every one of these input signals (i.e., the filter satisfies the **superposition principle**);
- for any input signal that has a given delay, the output undergoes the same delay as the input.

Features in general form

In its most general form, a comb filter has a frequency response that is a periodic function of ω, with a period $\frac{2\pi}{L}$, where L is a positive integer. Moreover, if $H(z)$ is a filter with a single passband and/or a single stopband, a comb filter can be easily generated from it by replacing each delay in its realization with L delays, resulting in a structure with a transfer function given by:

$$G(z) = H(z^L) \tag{1}$$

In particular, if $\left|H(e^{j\omega})\right|$ exhibits a peak at ω_p, then $\left|G(e^{j\omega})\right|$ will exhibits L peaks at $\frac{\omega_p k}{L}$, with $0 \le k \le L-1$, in the frequency range $0 \le \omega \le 2\pi$. Likewise, $\left|H(e^{j\omega})\right|$ has a notch at ω_0, then $\left|G(e^{j\omega})\right|$ will have L notches at $\frac{\omega_0 k}{L}$, with $0 \le k \le L-1$, in the frequency range $0 \le \omega \le 2\pi$.
Another important thing to remark is that a comb filter can be generated from either an FIR or an IIR prototype filter.

Examples of comb filters

Example 1: Comb filters from FIR/IIR prototype filter

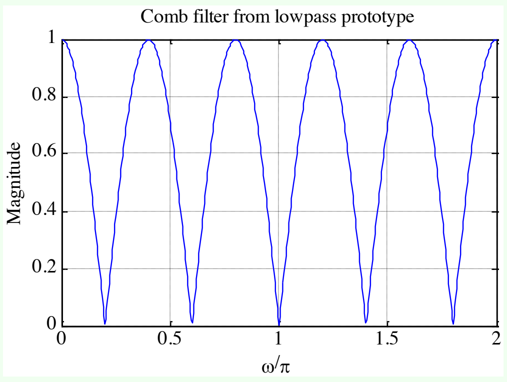
The comb filter generated from the prototype highpass FIR filter:

$$H_0(z) = \frac{1}{2}\left(1 + z^{-1}\right) \tag{2}$$

has a transfer function:

$$G_0(z) = H_0(z^L) = \frac{1}{2}\left(1 + z^{-L}\right) \tag{3}$$

$\left|G_0(e^{j\omega})\right|$ has L notches at $\omega = \frac{(2k+1)\pi}{L}$ and L peaks at $\omega = \frac{2\pi k}{L}$, with $0 \le k \le L-1$, in the frequency range $0 \le \omega < 2\pi$.



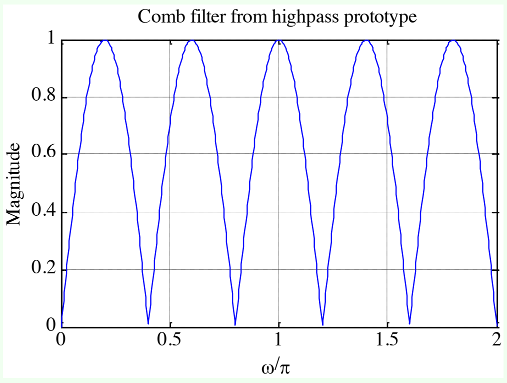
On the other hand, the comb filter generated from the prototype highpass FIR filter:

$$H_1(z) = \frac{1}{2}(1 - z^{-1}) \tag{4}$$

has a transfer function:

$$G_0(z) = H_1(z^L) = \frac{1}{2}(1 - z^{-L}) \tag{5}$$

$\left|G_0(e^{j\omega})\right|$ has L peaks at $\omega = \frac{(2k+1)\pi}{L}$ and L notches at $\omega = \frac{2\pi k}{L}$, with $0 \le k \le L-1$, in the frequency range $0 \le \omega < 2\pi$.



Depending on the application, comb filters with other types of periodic magnitude responses can be easily generated by appropriately choosing the prototype average filter.

Example 2: Comb filter from M-point moving average

The M-point moving average filter:

$$H(z) = \frac{1 - z^{-M}}{M(1 - z^{-1})} \tag{6}$$

can be used as a prototype. This filter has a peak magnitude at $\omega = 0$, and $M-1$ notches at $\omega = \frac{2\pi\ell}{M}$, with $1 \le \ell \le M-1$. The corresponding comb filter will have a transfer function:

$$G(z) = \frac{1 - z^{-LM}}{M(1 - z^{-L})} \tag{7}$$

whose magnitude has L peaks at $\omega = \frac{2\pi k}{LM}$, with $1 \le k \le L(M-1)$. By choosing L and M appropriately, peaks and notches can be created at desired locations.

0.2 Digital filter structures

Digital filter structures

Now, before going on with the discussion, let us return to basic concepts of an LTI discrete-time system. Let us consider:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} x[n-k]h[n] \tag{8}$$

The formula in Eq. 8 expresses the n^{th} output sample as a convolution sum of the input with the impulse response of the system. This can be considered as the characterization of the system and, therefore, the convolution sum can in principle be used to implement the system. It involves additions, multiplications and delays, which are simple operations. In particular, the input-output relation involves a finite sum of products:

$$y[n] = -\sum_{k=1}^{N} d_k y[n-k] + \sum_{k=0}^{M} p_k x[n-k] \tag{9}$$

On the other hand, an FIR system can be implemented using the convolution sum, which is a finite sum of products:

$$y[n] = \sum_{k=0}^{N} h[k]x[n-k] \tag{10}$$

Now, the actual implementation of an LTI digital filter can be either in software or hardware form, depending on the application. So, in either case, the signal variables and the filter coefficients cannot be represented with infinite precision.
A direct implementation based on either the difference equation or the finite convolution sum may not provide satisfactory performance due to the finite precision arithmentic. Thus, it is of practical interest to develop alternate realizations and choose the structure that provides satisfactory performances under this framework.
In this context, a **structural representation** using interconnected basic buiding blocks is the first step in the hardware or software implementation of an LTI digital filter. The structural representation provides the key relations between some pertinent internal variables with the input and output that in turn provides the key to the implementation.

Structural representation

0.2.1 Block diagram representation

Block diagram representation of computational algorithms

We have seen that, in time domain, the input-output relations of an LTI digital filter is given by the convolution sum in Eq. 8 or by the linear constant coefficient difference Eq. 9. For the implementation, the input-output relationship must be descirbed by a **valid computational algorithm**. To illustrate what we are meaning, let us consider the causal first-order LTI digital filter showed in Figure 1.

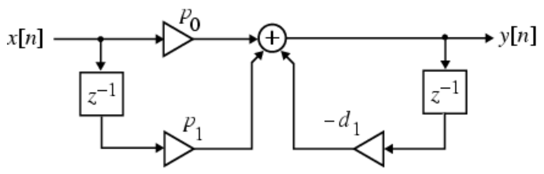


Figure 1: Block diagram representation of a causal first-order LTI digital filter.

Example with a causal first-order LTI digital filter

This filter is described by the difference equation:

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1] \tag{11}$$

Using Eq. 11, we can compute $y[n]$ for $n \geq 0$ knowing the initial condition $y[-1]$ and the input $x[n]$ for $n \geq -1$:

$$\begin{aligned}
y[0] &= -d_1 y[-1] + p_0 x[0] + p_1 x[-1] && (12)\\
y[1] &= -d_1 y[0] + p_0 x[1] + p_1 x[0] && (13)\\
y[2] &= -d_1 y[1] + p_0 x[2] + p_1 x[1] && (14)\\
\cdots &= \cdots
\end{aligned}$$

and so on and so forth. Each step of the calculation requires the knowledge of the previously calculated value of the output sample (delayed value of the output), the present value of the input sample and the previous one (delayed value of the input). As a result, the first-order difference equation can be interpreted as a valid computational algorithm.

Basic building blocks

The computational algorithm of an LTI digital filter can be conveniently represented in **block diagram form** using the **basic building blocks** showed in Figure 2.

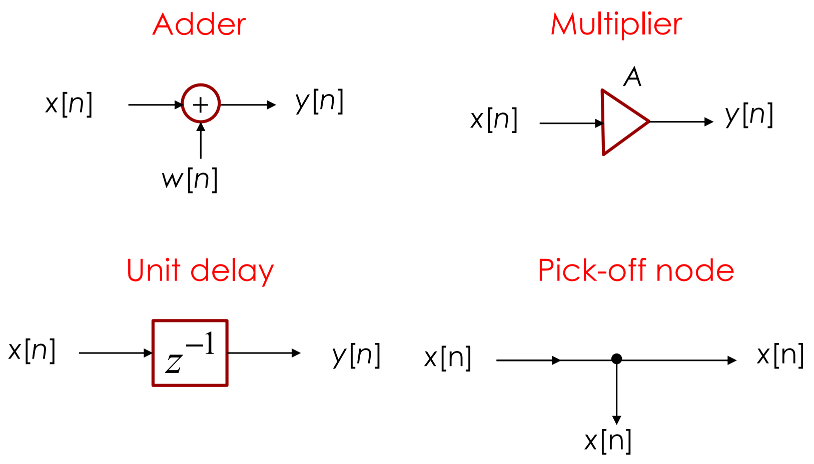


Figure 2: Basic building blocks of a block diagram.

Advantages of block diagram representation

There are several advantages when adopting this representation. In fact, it is:

- easy to write down the computational algorithm by inspection;
- easy to analyze the block diagram to determine the explicit relation between output and input;
- easy to manipulate a block diagram to derive other "equivalent" block diagrams yielding different computational algorithms;
- easy to determine the hardware requirements;
- easier to develop block diagram representations from the transfer function directly.

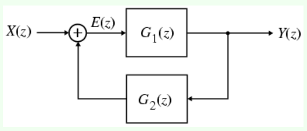
Analysis block diagrams

The **analysis of block diagrams** is carried out by writing down the expressions for the output signals of each adder as a sum of its input signals, and developing a set of equations relating the fiter input and output signals in terms of all internal signals. Eliminating the unwanted internal variables, then, results in the expression for the output signal as a function of the input signal and the filter parameters, that are the multiplier coefficients.

Example 3: Analysis of block diagrams

Example with a single-loop feedback structure

We consider the **single-loop feedback structure** showed below.



The output $E(z)$ of the adder is:

$$E(z) = X(z) + G_2(z)Y(z) \tag{15}$$

However, from the scheme we can see that:

$$Y(z) = G_1(z)E(z) \tag{16}$$

Eliminating $E(z)$ from Eqs. 15 and 16, we arrive at:

$$[1 - G_1(z)G_2(z)]Y(z) = G_1(z)X(z) \tag{17}$$

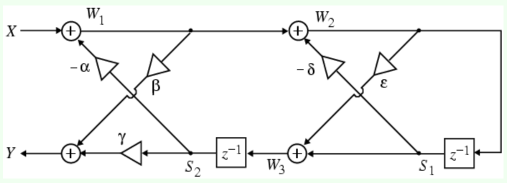
which leads to:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{G_1(z)}{1 - G_1(z)G_2(z)} \tag{18}$$

Example 4: Analysis of block diagrams

We analyze the **cascaded lattice structure** showed in the scheme below, where the z-dependence of signal variables is not showed for brevity.

Example with a cascaded lattice structure



The output signals of the four adders are given by:

$$W_1 = X - \alpha S_2 \tag{19}$$
$$W_2 = W_1 - \delta S_1 \tag{20}$$
$$W_3 = S_1 + \varepsilon W_2 \tag{21}$$
$$Y = \beta W_1 + \gamma S_2 \tag{22}$$

Moreover, from the scheme we can observe:

$$S_2 = z^{-1}W_3 \tag{23}$$
$$S_1 = z^{-1}W_2 \tag{24}$$

Substituting the last two relations in the first four equations, we get:

$$W_1 = X - \alpha z^{-1}W_3 \tag{25}$$
$$W_2 = W_1 - \delta z^{-1}W_2 \tag{26}$$
$$W_3 = z^{-1}W_2 + \varepsilon W_2 \tag{27}$$
$$Y = \beta W_1 + \gamma z^{-1}W_3 \tag{28}$$

Therefore, we get:

$$W_2 = \frac{W_1}{1 + \delta z^{-1}} \tag{29}$$
$$W_3 = \left(\varepsilon + z^{-1}\right)W_2 \tag{30}$$

Combining the last two equations we get:

$$W_3 = \frac{\varepsilon + z^{-1}}{1 + \delta z^{-1}}W_1 \tag{31}$$

Substituting Eq. 31 the ones for W_1 and Y, we finally arrive at:

$$H(z) = \frac{Y}{X} = \frac{\beta + (\beta\delta + \gamma\varepsilon)z^{-1} + \gamma z^{-2}}{1 + (\delta + \alpha\varepsilon)z^{-1} + \alpha z^{-2}} \tag{32}$$

0.2.2 The delay-free loop problem

Necessary condition for physical realizability of the structure

For the physical realizability of the digital filter structure, it is necessary that the block representation contains no delay-free loops. To illustrate the **delay-free loop problem**, we consider the structure in Figure 3.

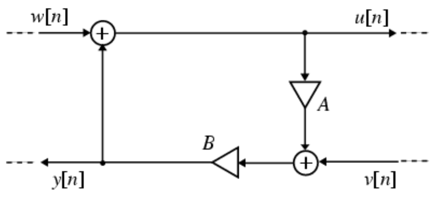


Figure 3: Example of structure with a delay loop.

From the analysis of this structure, we get:

$$u[n] = w[n] + y[n] \tag{33}$$
$$y[n] = B(v[n] + Au[n]) \tag{34}$$

which, when combined, results in:

$$y[n] = B(v[n] + A(w[n] + y[n])) \tag{35}$$

The determination of the current value of $y[n]$ requires the knowledge of the same value. However, this is physically impossible to achieve due to the finite time required to carry out all arithmetic operations on a digital machine.

Solving the delay-free loop problem

There exist methods to detect the presence of delay-free loops in an arbitrary structure, along with methods to locate and remove these loops without the overall input-output relation. One possibility of removal is achieved by replacing the portion of the overall structure containing the delay-free loops by an equivalent realization with no delay-free loops. To illustrate the process, let us consider again Eqs. 33 and 34, characterizing the structure in Figure 3. Substituing Eq. 34 into Eq. 33, we arrive at:

$$u[n] = w[n] + Bv[n] + ABu[n] \tag{36}$$

Solving Eq. 36, we get:

$$u[n] = \frac{1}{1 - AB}(w[n] + Bv[n]) \tag{37}$$

By reordering Eq. 33, we get:

$$y[n] = u[n] - w[n] \tag{38}$$

Therefore, we have obtained a delay-free loop realization based on Eqs. 37 and 38, and the scheme of the new structure is showed in Figure 4.

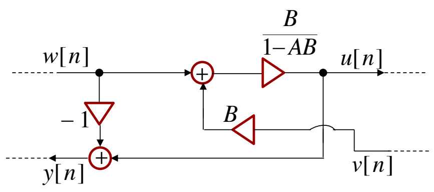


Figure 4: Structure of Figure 3 after removal of delay-free loops.

0.2.3 Canonic and non-canonic structures

Canonic and non-canonic structures

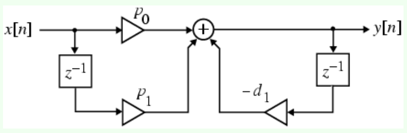
A digital filter structure is said to be **canonic** if the number of delays in the block diagram representation is equal to the order of the transfer function. Otherwise, it is a **non-canonic structure**.

Example 5: Canonic and non-canonic structures

The structure characterized by:

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1] \tag{39}$$

and showed in the scheme below is non-canonic, as it employs two delays to realize a first-order difference equation.


$x[n]$
p_0
$+$
$y[n]$
z^{-1}
p_1
$-d_1$
z^{-1}