

Lecture 5: Discrete-System Implementation

In this section, several simple digital signal processing techniques are presented to illustrate the practical applications based on the original concepts in the continuous-time domain.

The discrete versions are implemented through the equation-conversion technique as well as the impulse-invariance method. The use of the bilinear transformation for digital filter design will be presented later, after the introduction of Butterworth and Chebyshev filters.

1. Change detection

The most logical and practical approach to change detection is differentiation. And the simplest example is the 1st-order differentiation, as shown

$$y(t) = c \frac{d}{dt} x(t)$$

where c is a real and positive scaling factor. The frequency response of this simple operator is

$$H(j\omega) = c \cdot j\omega$$

The magnitude of the frequency response is a linear function of the frequency

$$|H(j\omega)| = c \cdot |\omega|$$

Thus, this operator is known as a high-pass filter, enhancing the high-frequency components linearly.

Recognizing the definition of the differentiation operator,

$$\frac{d}{dt}x(t) = \lim_{\Delta t \rightarrow 0} \frac{x(t) - x(t - \Delta t)}{\Delta t}$$

in the discrete mode, we can interpret it as the difference between two adjacent pixels and rewrite it in the form of a difference equation,

$$y(n) = \hat{c} [x(n) - x(n-1)]$$

where \hat{c} is also a scaling factor for similar purpose. The corresponding transfer function is

$$\hat{H}(z) = \hat{c} (1 - z^{-1})$$

and the frequency response is

$$\begin{aligned} \hat{H}(e^{j\theta}) &= \hat{c} (1 - e^{-j\theta}) = \hat{c} e^{-j\theta/2} (e^{+j\theta/2} - e^{-j\theta/2}) \\ &= 2 j \hat{c} e^{-j\theta/2} \sin(\theta/2) \end{aligned}$$

The magnitude of the frequency response also shows a high-pass operation,

$$| \hat{H}(e^{j\theta}) | = 2 \hat{c} | \sin(\theta/2) | \quad \text{for } -\pi \leq \theta \leq +\pi$$

Associated with the transfer function, the impulse response is

$$h(n) = \hat{c} [\delta(n) - \delta(n-1)]$$

The concept of differentiation has been the core element of change detection. And, based on the mathematical structure of the impulse response, in one-dimensional change detection, the normalized operator is commonly represented in the simple matrix form

$$A = [-1, +1]$$

Equivalent to the convolution with the impulse response of the change detector, we glide this simple 2-pixel operator through a given sequence produces the *change profile*. For symmetry, the change-detection operator can be modified into the 3-pixel format. This is to compute for the difference between two neighboring pixels

$$B = [-1, 0, +1]$$

which is corresponding to the difference equation,

$$y(n) = \hat{c} [x(n+1) - x(n-1)]$$

and equivalent to the differentiation operator

$$y(t) = \frac{d}{dt} x(t) = \lim_{\Delta t \rightarrow 0} \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t}$$

2. Edge detection in image analysis

Edge detection is one of the most common operations in image processing for the purpose of image enhancement. The simplest form of edge-detection operator in image processing is the 2×2 operator for the 1^{st} - order edge detection. In the horizontal direction, it is in the form

$$C = \begin{bmatrix} -1 & +1 \\ -1 & +1 \end{bmatrix}$$

The 3×3 first-order edge detection operator has been widely applied because of the symmetry

$$D = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$$

Over the years, various weighting schemes have been utilized to modify the matrix operators in different applications for improved effectiveness. One of the most common versions is use smaller weighting scale for the four farther pixels at the corners,

$$E = \begin{bmatrix} -1/\sqrt{2} & 0 & +1/\sqrt{2} \\ -1 & 0 & +1 \\ -1/\sqrt{2} & 0 & +1/\sqrt{2} \end{bmatrix}$$

Similarly, the weighted 3×3 1st-order edge detection in the vertical direction is in the form

$$F = \begin{bmatrix} +1/\sqrt{2} & +1 & +1/\sqrt{2} \\ 0 & 0 & 0 \\ -1/\sqrt{2} & -1 & -1/\sqrt{2} \end{bmatrix}$$

For improved computation efficiency, we can combine the edge-detection operators and formulate it in the complex form

$$\begin{aligned} G = E + jF &= \begin{bmatrix} -1/\sqrt{2} & 0 & +1/\sqrt{2} \\ -1 & 0 & +1 \\ -1/\sqrt{2} & 0 & +1/\sqrt{2} \end{bmatrix} + (j) \begin{bmatrix} +1/\sqrt{2} & +1 & +1/\sqrt{2} \\ 0 & 0 & 0 \\ -1/\sqrt{2} & -1 & -1/\sqrt{2} \end{bmatrix} \\ &= \begin{bmatrix} (-1+j)/\sqrt{2} & +j & (+1+j)/\sqrt{2} \\ -1 & 0 & +1 \\ (-1-j)/\sqrt{2} & -j & (+1-j)/\sqrt{2} \end{bmatrix} = \begin{bmatrix} \exp(j\frac{3\pi}{4}) & +j & \exp(j\frac{\pi}{4}) \\ -1 & 0 & +1 \\ \exp(j\frac{5\pi}{4}) & -j & \exp(j\frac{7\pi}{4}) \end{bmatrix} \end{aligned}$$

This operator produces a complex profile. The real part is the edge profile in the horizontal direction and the imaginary part is the profile in the vertical direction. This processing is often referred to as the *Gradient operator*, of which the vector form is

$$y(\alpha, \beta) = c \cdot \left[\frac{d}{d\alpha} x(\alpha, \beta), \quad \frac{d}{d\beta} x(\alpha, \beta) \right]$$

It is interesting to note that the weightings for the eight surrounding pixels are in the form of the neighboring pixels' angular positions,

$$w_n = \exp(jn\Delta\theta) = \exp(jn(\frac{2\pi}{8})) = \exp(jn\pi/4)$$

where $n = 0, 1, \dots, 7$.

3. Peak detection

The concept for peak detection derives from the 2^{nd} -order derivative

$$y(t) = c \frac{d^2}{dt^2} x(t)$$

The frequency response is

$$H(j\omega) = -c \cdot \omega^2$$

And the corresponding transfer function of the equivalent operation in the discrete format is

$$\hat{H}(z) = \hat{c} (1 - z^{-1})^2 = -\hat{c} (1 - 2z^{-1} + z^{-2})$$

The symmetrical version, which is commonly used in various applications, is in the form

$$\hat{H}(z) = -\hat{c} (-z + 2 - z^{-1})$$

corresponding to the difference equation

$$h(n) = -\hat{c} [-\delta(n+1) + 2\delta(n) - \delta(n-1)]$$

By extending this concept, we can formulate the normalized 2^{nd} -order change-detection operator as

$$P_3 = [-1/2, 1, -1/2]$$

Conceptually, it can be described as the difference between the value at the center pixel and the average value of the neighboring pixels. Based on this concept, the 3×3 peak-detection operator is in the form

$$P_{3 \times 3} = \left(\frac{1}{8} \right) \begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

This process is equivalent to *the Laplacian operator*.

$$y(\alpha, \beta) = c \cdot \nabla^2 x(\alpha, \beta) = c \cdot \left[\frac{d^2}{d\alpha^2} x(\alpha, \beta) + \frac{d^2}{d\beta^2} x(\alpha, \beta) \right]$$

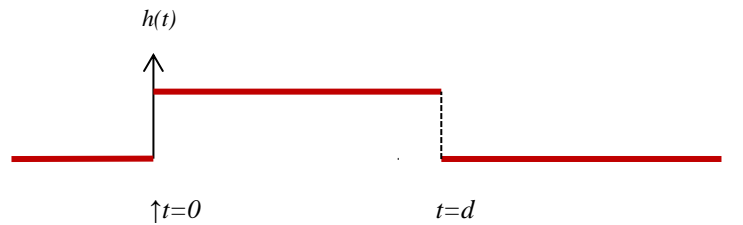
where α and β denote the defining variables of the horizontal and vertical direction. We can also extend the concept with smaller weightings for the four farther pixels and rewrite it in the form

$$Q_{3 \times 3} = \left(\frac{1}{6} \right) \begin{bmatrix} -1/2 & -1 & -1/2 \\ -1 & +6 & -1 \\ -1/2 & -1 & -1/2 \end{bmatrix}$$

4. Moving-average filters

Moving-average filtering is convolution process and the impulse response is a finite-duration pulse function with pulse duration d ,

$$h(t) = \begin{cases} +1/d & 0 \leq t \leq d \\ 0 & \text{elsewhere} \end{cases}$$



The magnitude of the pulse is set to $1/d$ such that the total area underneath the pulse is unity. The low-pass frequency response is

$$H(j\omega) = e^{-j\omega d/2} \text{sinc}(\omega d/2)$$

According to the impulse-invariance conversion method, the impulse response of the discrete system is in the form

$$h(n) = \begin{cases} +1/N & 0 \leq n \leq N-1 \\ 0 & \text{elsewhere} \end{cases}$$

And the low-pass frequency response is

$$\hat{H}(e^{j\theta}) = e^{-j(N-1)\theta/2} \frac{\sin(N\theta/2)}{N \sin(\theta/2)} \quad \text{for } -\pi \leq \theta \leq +\pi$$

Therefore, the convolution kernel of an N -point moving-average filter is

$$m_N = \left(\frac{1}{N}\right) [1, 1, 1, \dots, 1, 1]$$

It is to slide an N -point window along a sequence and the output is the average value of the N pixels within the window. As an example, based on the concept, the matrix representation of the convolution kernel of the 3-point moving average filter is

$$m_3 = \left(\frac{1}{3}\right) [1, 1, 1]$$

It is common to give different weightings to the neighboring pixels. An example of the weighted 3-point moving average operator is in the form

$$w_3 = \left(\frac{1}{4}\right) [1, 2, 1]$$

The moving-average filters have been widely used as an analysis tool in the finance community. Several versions, corresponding to $N = 5, 20$, and 65 associated with the number of trading days, are popularly used to produce the profiles, which are known as the weekly, monthly (4-week), and quarterly (13-week) lines. While we are computing for the average within the observation period, other statistical parameters can be calculated without significantly increasing the computation complexity. Using the mean, from the moving-average procedure, and the standard deviation to set the upper and lower bounds, a lane-like profile can be produced. The upper and lower bounds of the lane are known as *the Bollinger Bands*. Statistically, the probability of staying within the Bollinger Band is 67%. This simple application turned out to be extremely useful for estimation and prediction in financial analysis.

5. Moving average filters for image processing

The concept of moving-average filtering can be applied to image processing by extending the convolution kernel to an $N \times N$ matrix. The basic 2×2 moving-average filter for image processing is

$$M_{2 \times 2} = \left(\frac{1}{4}\right) \begin{bmatrix} +1 & +1 \\ +1 & +1 \end{bmatrix}$$

And similarly the 3×3 moving average filter is in the form

$$M_{3 \times 3} = \left(\frac{1}{9}\right) \begin{bmatrix} +1 & +1 & +1 \\ +1 & +1 & +1 \\ +1 & +1 & +1 \end{bmatrix}$$

Then the weighted 3×3 moving average filters can be written in the form

$$W_{3 \times 3} = \left(\frac{1}{16}\right) \begin{bmatrix} +1 & +2 & +1 \\ +2 & +4 & +2 \\ +1 & +2 & +1 \end{bmatrix}$$

The frequency responses can vary responding to coefficient-weighting schemes. Yet, the low-pass characteristics remain largely the same.

Summary:

1. Edge and peak detection techniques are high-pass enhancement methods commonly applied in image processing, commonly known as the *gradient* and *Laplacian* operators. These operators are equivalent to the 1^{st} and 2^{nd} -order *differentiation operations* in the analog form.
2. Moving average filtering is a low-pass filtering technique, converted through the *impulse invariance method*, corresponding to analog filters with finite-length impulse responses.
3. Different conversion techniques produce different discrete systems from the same analog system. Thus, the equivalent discrete system through the conversion of differential equation is not the same as the one produced through impulse invariance method.
4. The sampling rate is a governing parameter in both the equation-conversion and impulse invariance methods.