Chap. 6 A Framework for Digital Filter Design

6.1 Introduction to digital filters

6.2 Types of digital filters: FIR and IIR filters

6.3 Chose between FIR and IIR filters

6.4 Filter design steps

6.1 Introduction to digital filters

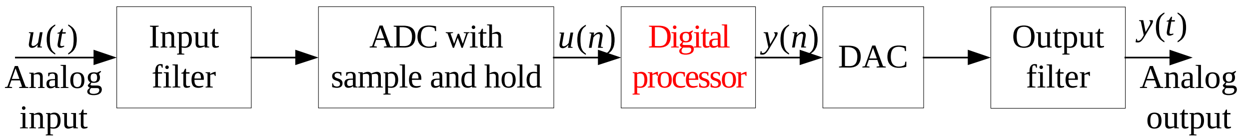
Objectives of Filter

1. To selectively change the wave shape, amplitude-frequency and/or phase-frequency characteristics of a signal in a desired manner.
2. To improve the quality of a signal.
3. To extract information from signals or to separate two or more signals in an available communication channel.

A digital filter

1. A mathematical algorithm implemented in hardware and/or software to manipulate a digital input signal and produce a digital output signal for the purpose of achieving a filtering objective.
2. A specific hardware or software routine performs the filtering algorithm.

A simplified block diagram of a real-time digital filter



1. Input Filter:

   An analog signal is shaped in the form of band-limited signal by the input filter.

2. ADC:

   The band-limited analog signal is sampled periodically and converted into a series of digital samples, $u(n), n = 0, 1, 2, \cdots$.

3. The digital processor:

To implement the filtering operation, mapping the input sequence $u(n)$ into the output sequence $y(n)$ in accordance with a computational algorithm for the filter.

4. DAC

   To convert the digitally filtered output into analog values.

5. The output filter

   To smooth and remove unwanted high frequency components.

🕮 Advantages of digital filter

1. Guaranteed accuracy (depends on the number of bits)
1. Perfect reproducibility (easily reproduced)
2. No drift in performance with aging and environmental variation.
3. Easily implemented in the IC technology (smaller size, higher speed, and lower power consumption)
4. Greater flexibility (easily programmed and reprogrammed without modifying the hardware)
5. Superior performance

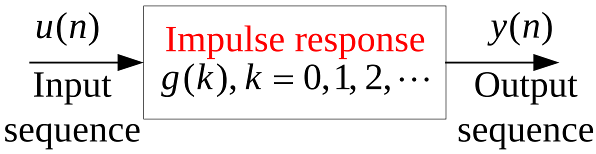
🕮 Disadvantages of digital filter

1. Speed limitations:
   - The analog-digital-analog conversion process introduce a speed constrains on the digital filter performance.
   - The conversion time of the ADC and the settling time of the DAC limit the highest frequency of filtering process.
   - The speed of digital filter operation depends on the speed of digital processors and the number of arithmetic operation.
2. Design time: Time-to-market, Cost for market
3. Finite word-length problems
   - The ADC noise results from the quantization of an analog signal.

- The round-off noise comes from computation.

6.2 Types of digital filters: FIR and IIR filters

Conceptual representation of a digital filter



The digital filter is described in the form of impulse response sequence $g(k)$, $k=0,1,2,\cdots$, where $u(n)$ is the input sequence and $y(n)$ is the output sequence.

- Finite Impulse response (FIR)

$$y(n)=\sum_{k=0}^{N-1} g(k)u(n-k)$$

The impulse response of FIR is of finite duration with N values. The transfer function of FIR is given as

$$G(z)=\sum_{k=0}^{N-1} g(k)z^{-k}$$

- Infinite Impulse response (IIR)

$$y(n)=\sum_{k=0}^{\infty} g(k)u(n-k)$$

The impulse response of IIR is in the form of infinite duration. The IIR filtering equation can be expressed in a recursive form shown as follows.

$$y(n)=\sum_{k=0}^{\infty} g(k)u(n-k)=\sum_{k=0}^{N} b_k u(n-k)-\sum_{i=1}^{M} a_i y(n-i)$$

The transfer function of IIR is given as

$$G(z)=\frac{\sum_{k=0}^{N} b_k z^{-k}}{1+\sum_{i=0}^{M} a_i z^{-i}}$$

6.3 Choice between FIR and IIR filters

- Consideration between FIR and IIR filters

1. FIR filter has an exact linear phase response. The phase response of IIR filter is nonlinear, especially at the band edges. FIR filter is good for data transmission, biomedicine, digital audio, and image processing.
2. FIR filter is non-recursively realized by direct evaluation and the result is always stable. IIR filter is implemented by recursive operation and the stability of IIR filter cannot be guaranteed.
3. The effects on the word-length of implementing filters are much less server in FIR than IIR, such as round-off noise and coefficient quantization.
4. FIR requires more coefficients for sharp cutoff filter than IIR. Given a amplitude response specification, more processing time and storage are required for FIR implementation. The computational speed of FFT and multi-rate techniques can improve significantly the efficiency of FIR implementation.
5. Analog filter can be readily transformed into equivalent IIR digital filter with similar specifications.
6. FIR is algebraically more difficult to synthesize without CAD support.

- Broad guideline FIR and IIR filters

1. IIR will give fewer coefficients than FIR when the important requirements are sharp cutoff filters and high throughput.
2. It is better to use FIR if the number of filter coefficients is not too large and no phase distortion is desired.

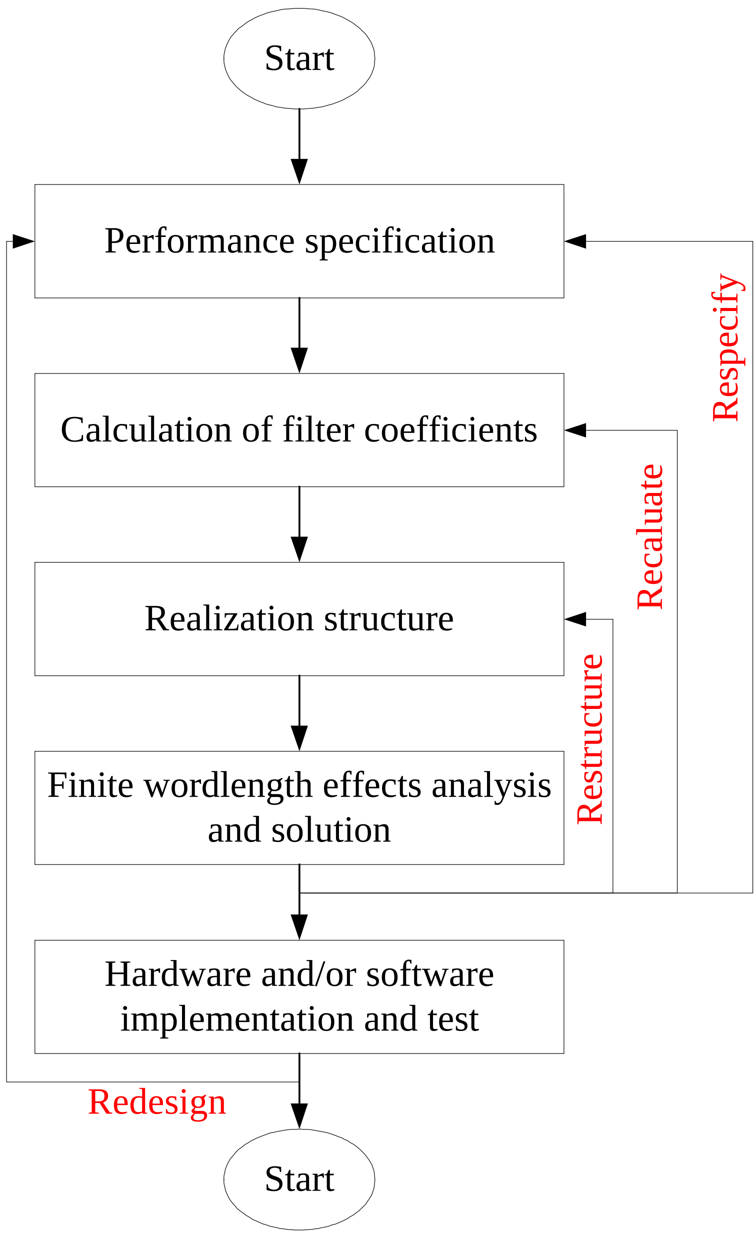
6.4 Filter Design Steps

- Steps of filter design

1. Filter specification:

   Specification of the filter requirements includes the type of filter, the desired amplitude and/or phase response and their tolerance, the sampling frequency, and the word-length of the input data.

2. Coefficient calculation: Calculation of suitable filter coefficients.
3. Realization: The transfer function of the filter is expressed by a suitable filter network or structure.
4. Analysis of the finite word-length effects: The quantization of the filter coefficients and the input data as well as the filtering operation with finite word-length will have the effect on filter's performance.
5. Implementation of filter in software and/or hardware.



- Specification of the filter requirements

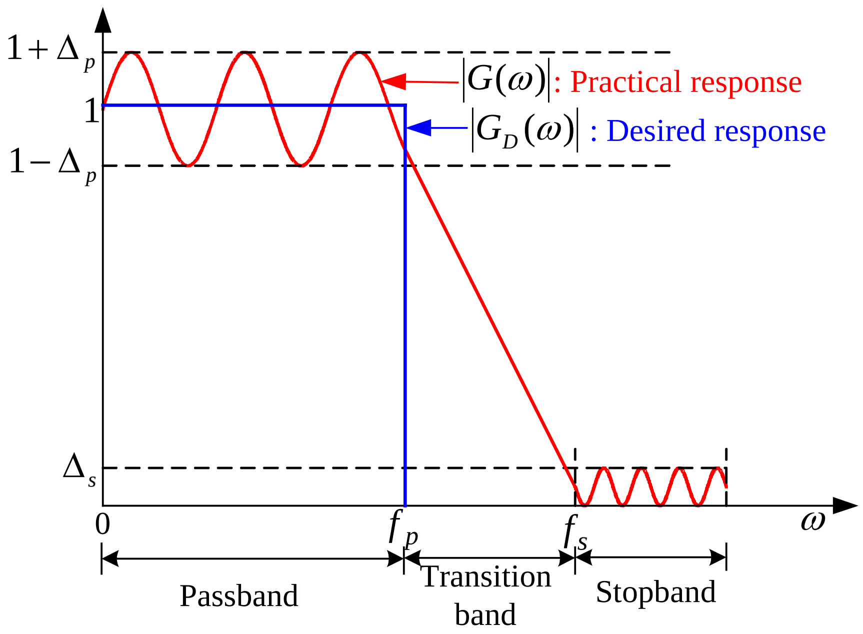
📖 Required specifications

1. Signal characteristics: types of signal source and sink, I/O interface, data rates and width, and highest frequency of interest.
2. The characteristics of filter: desired amplitude and/or phase response and their tolerances, the speed of operation and modes of filtering (real time or batch).
3. The manner of implementation: a high-level language routine in a computer or a DSP processor-based system, choice of signal processor.
4. Other design constraints: cost, Time-to-Market

📖 The characteristics of digital filters are often specified in frequency domain with the form of tolerance schemes. For the phase response, the positive symmetry or negative symmetry is required.

📖 Tolerance scheme for a lowpass filter

1. The magnitude response of a lowpass filter



where Δ_p is the passband deviation, Δ_s is the stopband deviation, f_p is the passband edge frequency, and f_s is the stopband edge frequency. For FIR filter, the peak passband ripple A_p and the minimum stopband attenuation A_s is given as

$$A_p = 20\log_{10}(1+\Delta_p)$$

and

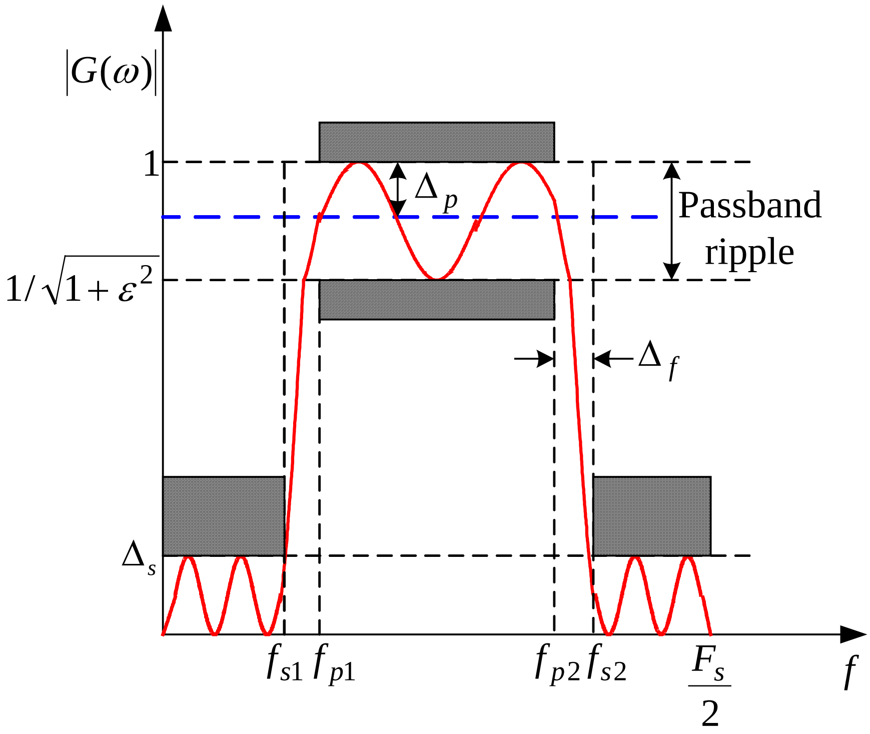
$$A_s = -20\log_{10}\Delta_s$$

2. In the passband, the magnitude response has a peak deviation. In the stopband, it has a maximum deviation. The shaded horizontal lines indicate the tolerance limits.
3. The sharpness of the filter depends on the width of the transition band.

4. The phase response of digital filter is often not specified as the magnitude response.

- IIR bandpass filter

📖 Tolerance scheme for IIR bandpass filter.



where ε^2: parameter ripple parameter, Δ_p: passband deviation, Δ_s: stopband deviation, f_{p1} and f_{p2}: passband edge frequencies, f_{s1} and f_{s2}: stopband edge frequencies, and the shaded horizontal lines indicate tolerance limits.

📖 The passband ripple and stopband attenuation in decibels:

$$A_p = 10\log_{10}(1+\varepsilon^2) = -20\log_{10}(1-\Delta_p)$$

$$A_s = -20\log_{10}(\Delta_s)$$

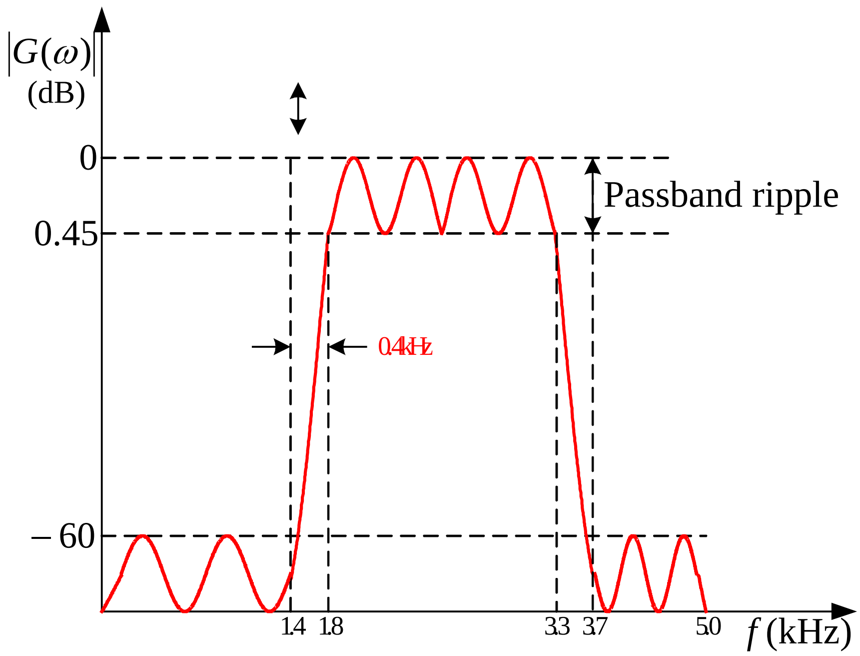
📖 Example

Given the frequency response specifications of an FIR bandpass filter,

1. Passband (normalized): 0.18~0.33
2. Transition width: (normalized): 0.04
3. Stopband deviation: 0.001
4. Passband deviation: 0.05
5. Sampling frequency: 10kHz

To express the filter bandedge frequency in the standard unit of kHz and the stopband and passband deviation in dB

1. Passband: 1.8~3.3kHz
2. Transition width: 0.4kHz
3. Stopbands: 0~1.4kHz and 3.7~5kHz
4. Stopband attenuation: $A_s = -20\log(0.001) = 60(\text{dB})$
5. Passband ripple: $A_p = -20\log(1-0.05) = 0.45(\text{dB})$



- Calculation of suitable filter coefficients

📖 Calculation of FIR filter coefficients

1. The methods to calculate the FIR coefficients are the window, frequency sampling and the optimal (Parks-McClellan algorithm) ones.
2. The optimal method is the first choice

📖 Window method

1. The window method offers a very simple way and a minimum amount of computing FIR coefficients.
2. Its drawback is that it does not offer flexible and adequate control over the filter parameters.

📖 Frequency sampling method

1. It is a recursive realization of FIR and the computation of filter coefficients is very efficient.
2. It lacks flexibility in specifying or controlling filter parameters.

📖 Optimal method

1. The optimal method is now widely used in industry with the availability of an efficient and easy-to-use program.
2. For FIR, the optimal method is the first choice.

📖 Calculation of IIR filter coefficients

1. The coefficient calculations of IIR filter are based on the transformation of known analog filter characteristics into equivalent digital filters by the impulse invariant and the bilinear transformation methods.
2. The pole-zero placement method can be used to calculate the coefficients of simple IIR filters with "trial and error".
3. The bilinear method is sufficient for most cases.

📖 Impulse invariant method

1. Digitalizing the analog filter
2. The impulse response of the original analog filter is preserved.
3. The method is not appropriate for highpass or bandstop filter because of inherent aliasing.
4. The impulse invariant method is good for simulating analog system.

📖 Bilinear transformation method

1. The bilinear method preserves the magnitude-frequency response characteristics of the analog filter, not the time domain responses.
2. It allows the design of digital filters with classical characteristics such as Butterworth, Chebyshev, and elliptic.
3. It is suitable for the calculation of coefficients of frequency selective filters.

4. The bilinear method is best for frequency selective IIR filters.

- Representation of the filter by a suitable structure (realization)

🕮 Realization involves converting a given transfer function $G(z)$ into a suitable filter structure with block or flow diagrams.

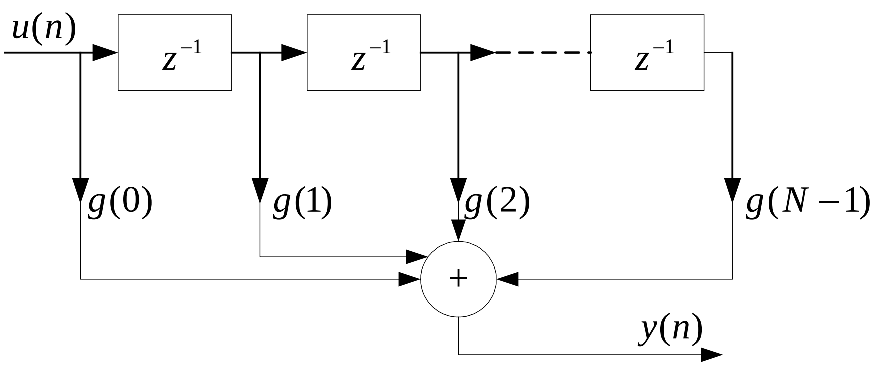
🕮 The block or flow diagrams show the computational procedure for implementing the digital filters.

🕮 For FIR filters, the direct form, frequency sampling structure, and fast convolution methods are common techniques.

🕮 Direct form

1. The most widely used structure for FIR is the direct form with simple implementation.
2. Example: Transversal filter

Given $G(z)=\dfrac{Y(z)}{U(z)}$, $Y(z)=G(z)U(z)$, $y(n)=g(n)\otimes u(n)=\sum_{i=0}^{n} g(i)u(n-i)$



🕮 Frequency sampling structure

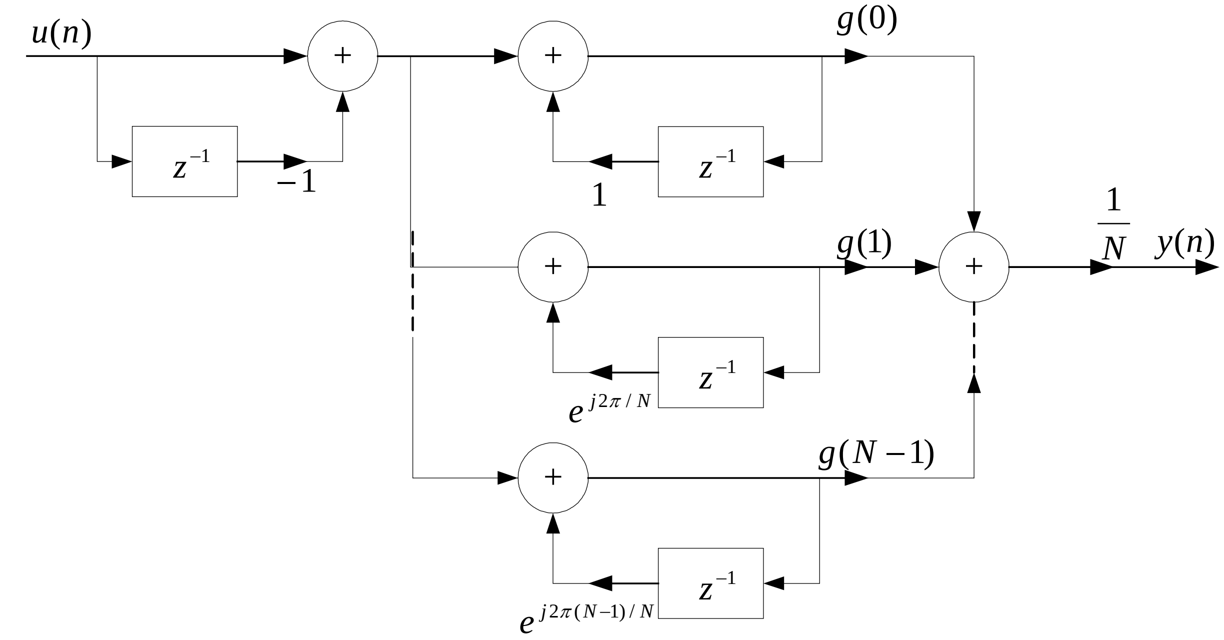
1. The frequency sampling structure can be computationally more efficient with fewer coefficients.
2. Example:

Given $G(z)=\dfrac{Y(z)}{U(z)}$, $Y(z)=G(z)U(z)$,

The transfer function $G(z)$ of FIR can be expressed in the form of frequency domain as follows.

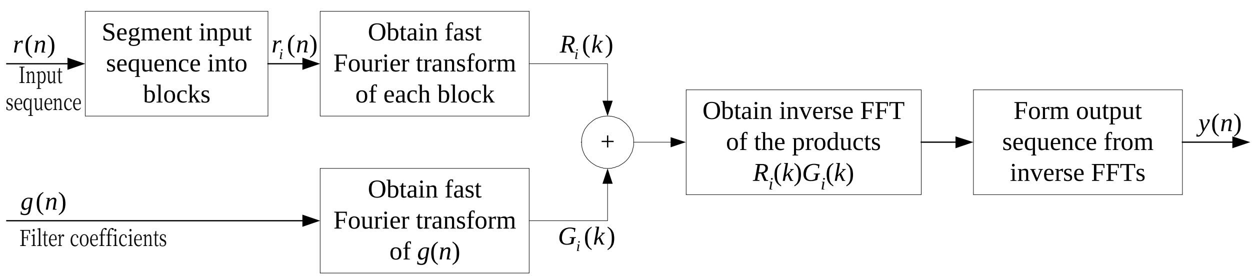
$$G(\omega) = \sum_{k=0}^{N} g(k) e^{-jk\omega}$$

where $\omega = \frac{2\pi}{N} T$



🕮 Fast convolution

1. The fast convolution uses the computational advantage of the Fast Fourier Transform (FFT) and also provides the power spectrum of signal.
2. Example



🕮 For IIR filters, the direct, cascade, and parallel forms are common forms.

🕮 Direct method

1. The direct form is simply a straightforward representation of the IIR transfer function.
2. The gains of the realizations are coefficients in the transfer function polynomials.

Such as: Controllable canonical form and observable canonical form.

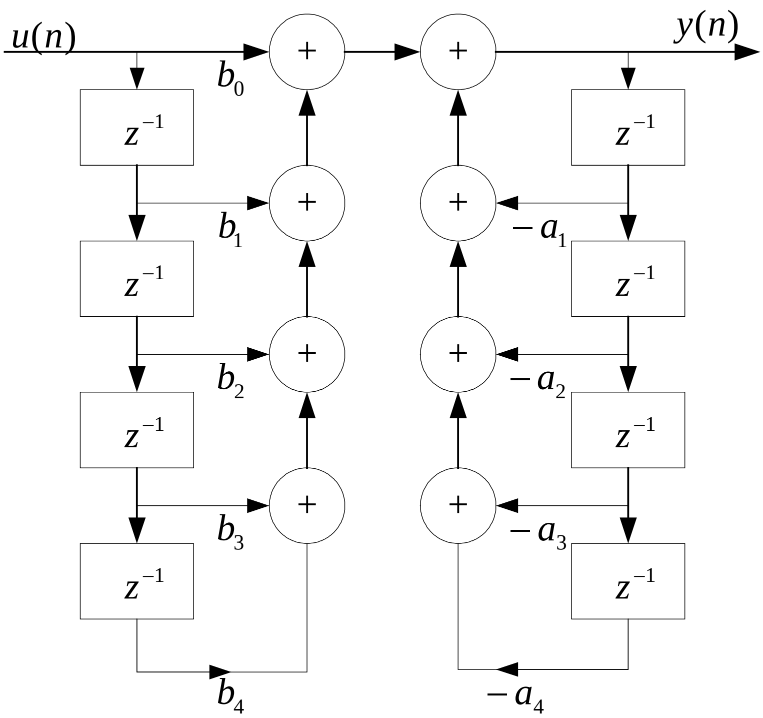
3. The direct structure suffers from severe coefficients sensitivity problem. It should be avoided to use the direct form for high-order filters.
4. Example: Direct realization of a fourth-order IIR filter

Assume that the transfer function of a fourth-order IIR is given as

$$G(z)=\frac{\sum_{k=0}^{4} b_k z^{-k}}{1+\sum_{i=0}^{4} a_i z^{-i}}$$

The IIR filtering equation can be expressed in a recursive form shown as follows.

$$y(n)=\sum_{k=0}^{4} b_k u(n-k)-\sum_{i=1}^{4} a_i y(n-i)$$



📖 Direct canonical form:

✧ Controllable canonical form: (Phase-variable canonical form)

Given $G(z)=\frac{Y(z)}{U(z)}=\frac{b_3 z^3+b_2 z^2+b_1 z+b_0}{z^3+a_2 z^2+a_1 z+a_0}$, we have

$$G(z) = \frac{Y(z)}{U(z)} = \frac{(b_3 z^3 + b_2 z^2 + b_1 z + b_0) X(z)}{(z^3 + a_2 z^2 + a_1 z + a_0) X(z)}$$

$$Y(z) = (b_3 z^3 + b_2 z^2 + b_1 z + b_0) X(z),$$

$$y(kT) = b_3 x\left[(k+3)T\right] + b_2 x\left[(k+2)T\right] + b_1 x\left[(k+1)T\right] + b_0 x(kT)$$

and

$$u(kT) = x\left[(k+3)T\right] + a_2 x\left[(k+2)T\right] + a_1 x\left[(k+1)T\right] + a_0 x(kT)$$

Set $x_1(kT) = x(kT)$,

$$x_2(kT) = x_1\left[(k+1)T\right] = x\left[(k+1)T\right],$$

$$x_3(kT) = x_2\left[(k+1)T\right] = x\left[(k+2)T\right]$$

we have

$$\begin{aligned} x_3\left[(k+1)T\right] = x\left[(k+3)T\right] &= -a_2 x\left[(k+2)T\right] - a_1 x\left[(k+1)T\right] - a_0 x(kT) + u(kT) \\ &= -a_2 x_3(kT) - a_1 x_2(kT) - a_0 x_1(kT) + u(kT) \end{aligned}$$

and

$$\begin{aligned} & y(kT) = b_3 x\left[(k+3)T\right] + b_2 x\left[(k+2)T\right] + b_1 x\left[(k+1)T\right] + b_0 x(kT) \\ & = b_3\left[-a_2 x_3(kT) - a_1 x_2(kT) - a_0 x_1(kT) + u(kT)\right] + b_2 x_3(kT) + b_1 x_2(kT) + b_0 x_1(kT) \\ & = (b_2 - b_3 a_2) x_3(kT) + (b_1 - b_3 a_1) x_2(kT) + (b_0 - b_3 a_0) x_1(kT) + b_3 u(kT) \end{aligned}$$

In the state-variable formulation, the discrete-data system is described by the following dynamic equation in the form of controllable canonical form

$$\begin{bmatrix} x_1\left[(k+1)T\right] \\ x_2\left[(k+1)T\right] \\ x_3\left[(k+1)T\right] \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_2 & -a_1 & -a_0 \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \\ x_3(kT) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(kT)$$

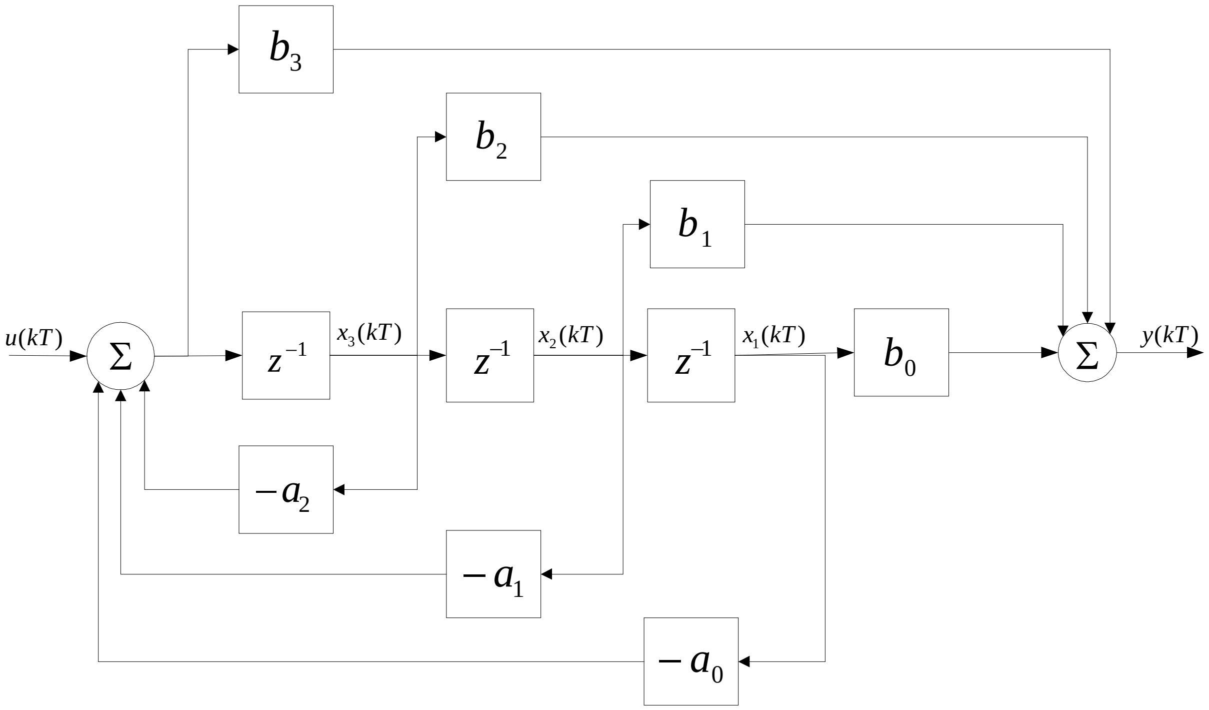
$$X\left[(k+1)T\right] = A_C X(kT) + B_C u(kT)$$

and

$$y(kT) = \begin{bmatrix} b_2 - b_3 a_2 & b_1 - b_3 a_1 & b_0 - b_3 a_0 \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \\ x_3(kT) \end{bmatrix} + b_3 u(kT)$$

$$y(kT) = C_C X(kT) + D_C u(kT)$$

Block Diagram of Controllable Canonical Form



Note: The time delay units z^{-1} are connected in series and the state variables are set form right to left.

✧ Observable canonical form

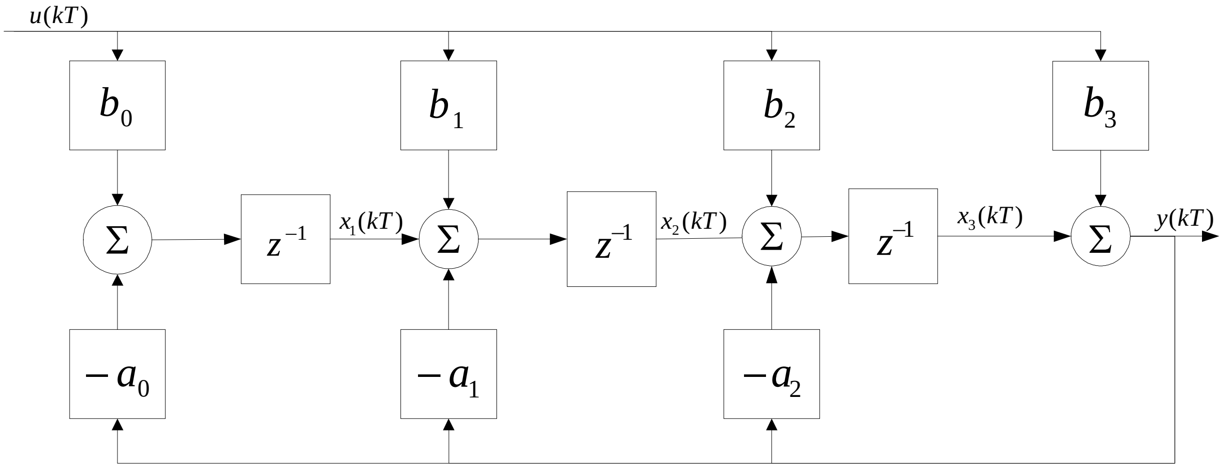
Given $G(z) = \dfrac{Y(z)}{U(z)} = \dfrac{b_3 z^3 + b_2 z^2 + b_1 z + b_0}{z^3 + a_2 z^2 + a_1 z + a_0}$, we have

$$(z^3 + a_2 z^2 + a_1 z + a_0)Y(z) = (b_3 z^3 + b_2 z^2 + b_1 z + b_0)U(z)$$

$$(1 + a_2 z^{-1} + a_1 z^{-2} + a_0 z^{-3})Y(z) = (b_3 + b_2 z^{-1} + b_1 z^{-2} + b_0 z^{-3})U(z)$$

$$Y(z) = b_3 U(z) + z^{-1}\left[b_2 U(z) - a_2 Y(z)\right] + z^{-2}\left[b_1 U(z) - a_1 Y(z)\right] + z^{-3}\left[b_0 U(z) - a_0 Y(z)\right]$$

The Block Diagram of Controllable Canonical Form is described as follows



and we have

$$x_1[(k+1)T] = b_0u(kT) - a_0y(kT) = -a_0x_3(kT) + (b_0 - a_0b_3)u(kT)$$
$$x_2[(k+1)T] = x_1(kT) + b_1u(kT) - a_1y(kT) = x_1(kT) - a_1x_3(kT) + (b_1 - a_1b_3)u(kT)$$
$$x_3[(k+1)T] = x_2(kT) + b_2u(kT) - a_2y(kT) = x_2(kT) - a_2x_3(kT) + (b_2 - a_2b_3)u(kT)$$
$$\begin{bmatrix} x_1[(k+1)T] \\ x_2[(k+1)T] \\ x_3[(k+1)T] \end{bmatrix} = \begin{bmatrix} 0 & 0 & -a_0 \\ 1 & 0 & -a_1 \\ 0 & 1 & -a_2 \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \\ x_3(kT) \end{bmatrix} + \begin{bmatrix} b_0 - a_0b_3 \\ b_1 - a_1b_3 \\ b_2 - a_2b_3 \end{bmatrix} u(kT)$$
$$X[(k+1)T] = A_O X(kT) + B_O u(kT)$$

and

$$y(kT) = x_3(kT) + b_3u(kT) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \\ x_3(kT) \end{bmatrix} + b_3u(kT)$$
$$y(kT) = A_O X(kT) + D_O u(kT)$$

Note: The time delay units z^{-1} are connected in series and the state variables are set form left to right.

📖 Cascade form

1. The cascade and parallel structures are the most widely used for IIR. They are simpler filtering algorithm and less sensitive to the effect of wordlength than direct form.
2. In the cascade form, the transfer function $G(z)$ of IIR filter is factored and

expressed as the product of second –order section.

3. Example: Cascade realization of a fourth-order IIR filter

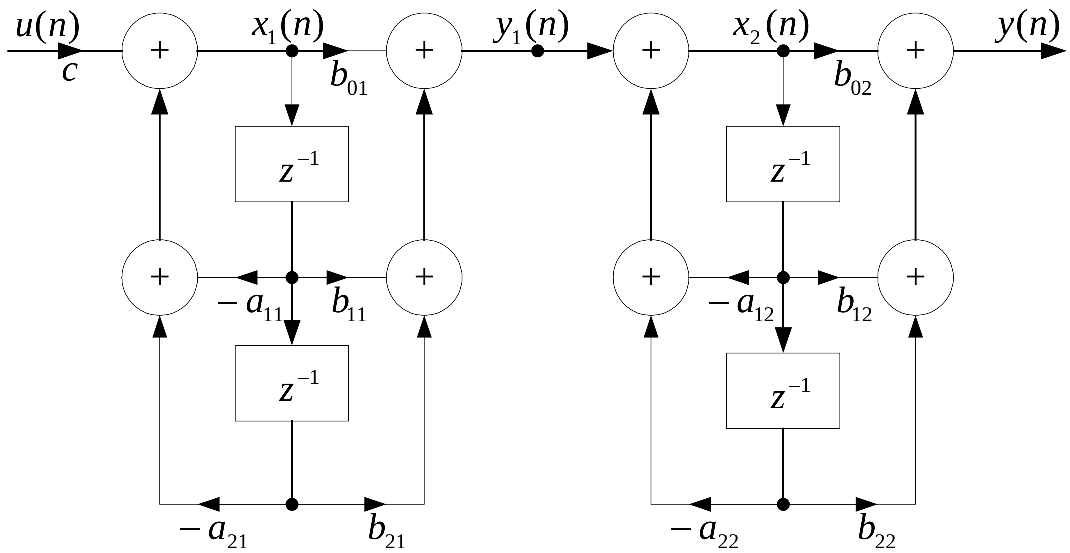
Given $G(z)=\dfrac{Y(z)}{U(z)}=c\prod_{i=1}^{2}\dfrac{b_{0i}+b_{1i}z^{-1}+b_{2i}z^{-2}}{1+a_{1i}z^{-1}+a_{2i}z^{-2}}$, we have

$$x_1(n)=cu(n)-a_{11}x_1(n-1)-a_{21}x_1(n-2),$$

$$y_1(n)=b_{01}x_1(n)+b_{11}x_1(n-1)+b_{21}x_1(n-2)$$

$$x_2(n)=y_1(n)-a_{12}x_2(n-1)-a_{22}x_2(n-2)$$

$$y(n)=b_{02}x_2(n)+b_{12}x_2(n-1)+b_{22}x_2(n-2)$$



📖 Parallel method

1. In the parallel form, the transfer function $G(z)$ of IIR filter is expanded as the sum of second –order section with the partial fractions.

2. Example: Parallel realization of a four-order IIR filter

Given $G(z)=\dfrac{Y(z)}{U(z)}=c+\sum_{i=1}^{2}\dfrac{b_{0i}+b_{1i}z^{-1}}{1+a_{1i}z^{-1}+a_{2i}z^{-2}}$, we have

$$x_1(n)=r(n)-a_{11}x_1(n-1)-a_{21}x_1(n-2)$$

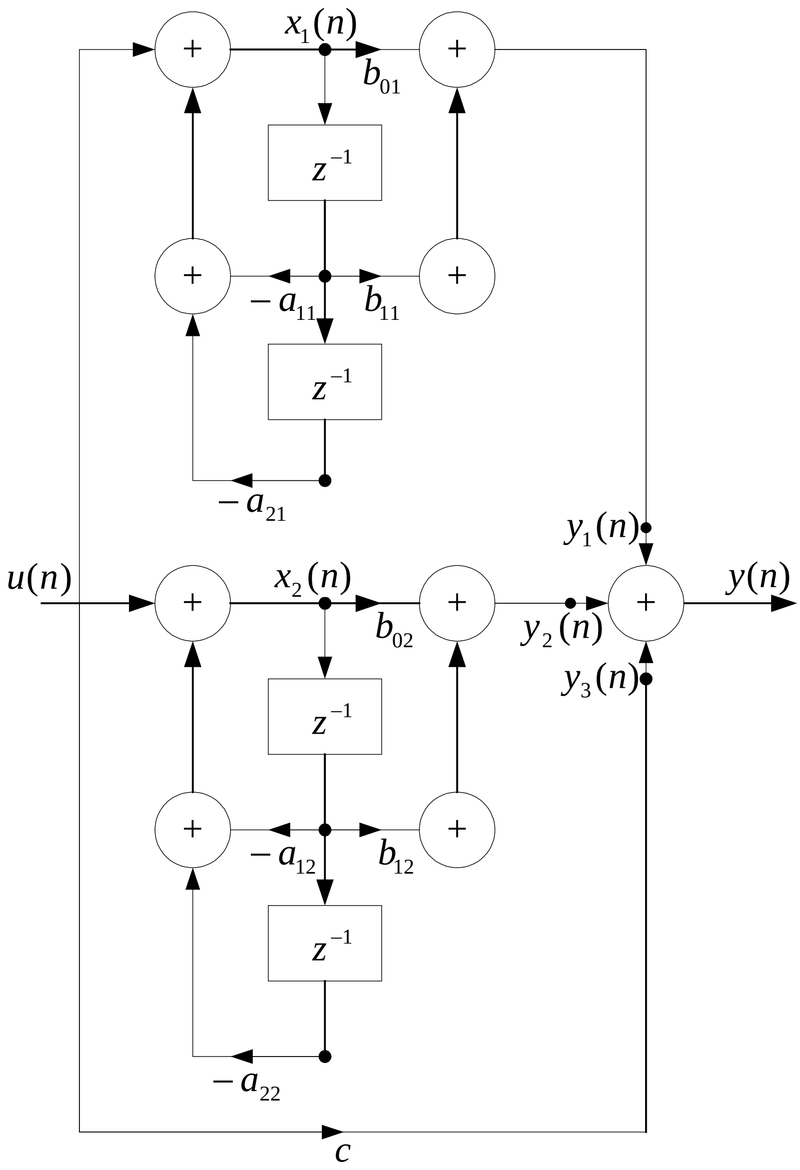
$$x_2(n)=r(n)-a_{12}x_2(n-1)-a_{22}x_2(n-2)$$

$$y_1(n) = b_{01}x_1(n) + b_{11}x_1(n-1)$$

$$y_2(n) = b_{02}x_2(n) + b_{12}x_2(n-1)$$

$$y_3(n) = cu(n)$$

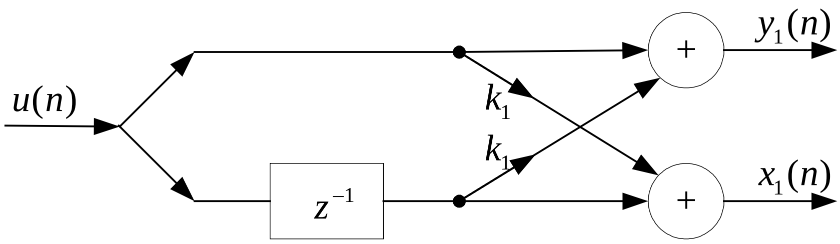
$$y(n) = y_1(n) + y_2(n) + y_3(n)$$



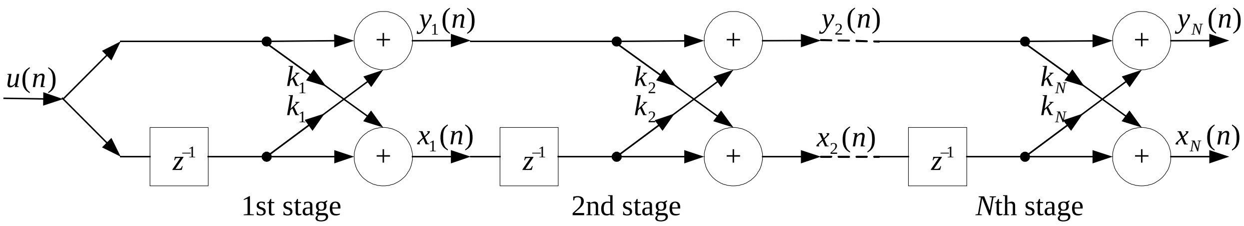
📖 Lattice structure

1. The lattice structure is used in speech processing and linear prediction application.
2. The lattice structure may be used to represent FIR and IIR filters.
3. Example

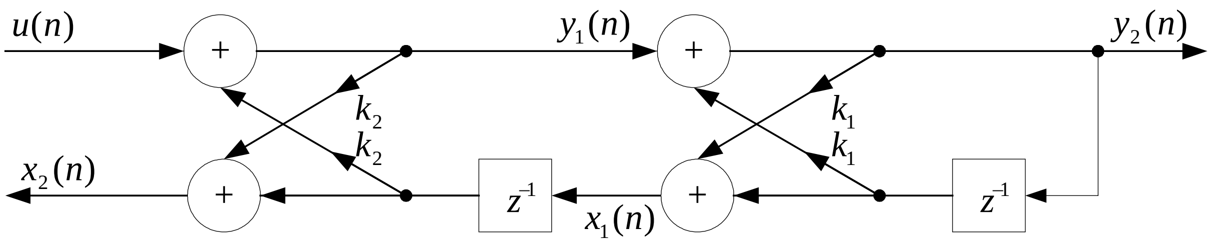
➢ Basic lattice structure



➢ N-stage FIR lattice structure



➢ Two-stage all-pole IIR lattice structure



- Analysis of the effects of finite wordlength on filter performance

🕮 The effects of using a finite number of bits will degrade the performance of filter.

🕮 Main sources of performance degradation in digital filter

1. Input/output signal quantization: ADC noise
2. Coefficient quantization: This leads to deviation in the frequency of FIR and IIR filters, especially for IIR filter.
3. Arithmetic roundoff errors: The finite precision arithmetic to perform filtering operation yields roundoff noise with rounding. This leads instability in IIR filter.
4. Overflow: The result of an addition exceeds permissible wordlength. This leads to wrong output and instability in IIR filters.

🕮 Factors of digital filter performance

1. The wordlength and type of arithmetic used to perform the filtering operation.
2. The method used to quantize filter coefficients and variables to the chosen length.
3. The filter structure.

- Implementation of filter in software and/or hardware

📖 Basic building blocks to implement a filter

1. Memory for storing filter coefficients: ROM
2. Memory for storing the present and past inputs $\{r(n), \quad r(n-1), \quad \cdots\}$ and outputs $\{y(n), \quad y(n-1), \quad \cdots\}$: RAM
3. Hardware or software multiplier(s)
4. Adder or arithmetic logic unit.

📖 Batch processing

1. The entire data is available in some memory device.
2. The filter is implemented in a high-level language and runs in a general-purpose computer.
3. Batch processing may be described as a purely software implementation.

📖 Real-time processing

1. The filter is required either to operate on the present input $r(n)$ and produce the current output $y(n)$ within the intersample period or to operate on an input block of data with FFT algorithms and produce an output block of data within a period proportional to the block length.
2. Real-time filters require fast and dedicated hardware with high sampling rate or high order.