Filter Digital

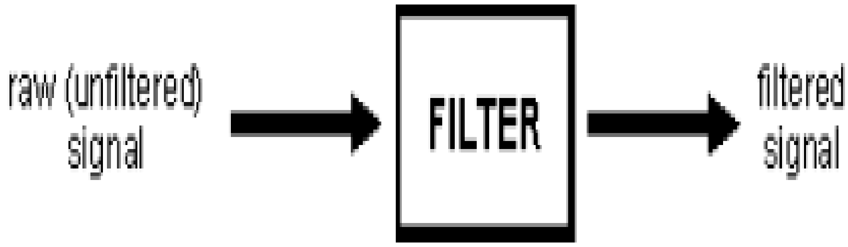
Bahan Ajar Sinyal dan Sistem
Pascasarjana Terapan
P E N S

Introduction to digital filters

Analog and digital filters

In signal processing, the function of a *filter* is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range.

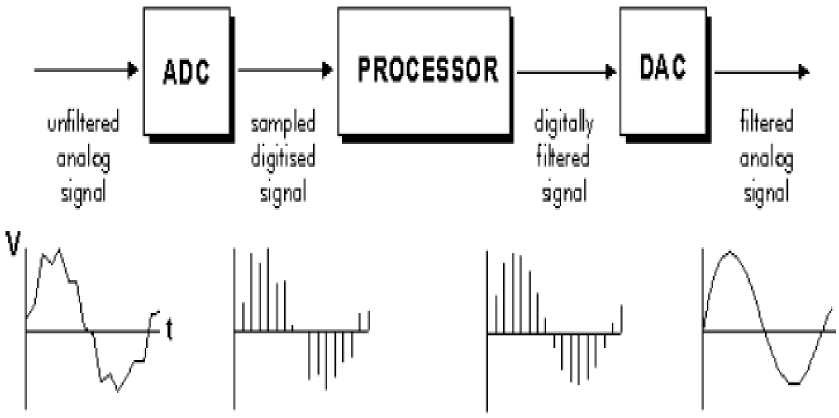
Blok Diagram



Analog & Digital Filter

- An **analog** filter uses analog electronic circuits made up from components such as resistors, capacitors and op amps to produce the required filtering effect. Such filter circuits are widely used in such applications as noise reduction, video signal enhancement, graphic equalisers in hi-fi systems, and many other areas.
- A **digital** filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialised DSP (Digital Signal Processor) chip.

Blok Diagram Filter Digital



Advantages of using digital filters (1)

- A digital filter is *programmable*, i.e. its operation is determined by a program stored in the processor's memory. This means the digital filter can easily be changed without affecting the circuitry (hardware). An analog filter can only be changed by redesigning the filter circuit.
- Digital filters are easily *designed*, *tested* and *implemented* on a general-purpose computer or workstation.
- The characteristics of analog filter circuits (particularly those containing active components) are subject to drift and are dependent on temperature. Digital filters do not suffer from these problems, and so are extremely *stable* with respect both to time and temperature.

Advantages of using digital filters (2)

- Unlike their analog counterparts, digital filters can handle *low frequency* signals accurately. As the speed of DSP technology continues to increase, digital filters are being applied to high frequency signals in the RF (radio frequency) domain, which in the past was the exclusive preserve of analog technology.
- Digital filters are very much more *versatile* in their ability to process signals in a variety of ways; this includes the ability of some types of digital filter to adapt to changes in the characteristics of the signal.
- Fast DSP processors can handle complex combinations of filters in parallel or cascade (series), making the hardware requirements relatively *simple* and *compact* in comparison with the equivalent analog circuitry.

Examples of simple digital filters

1. Unity gain filter:

$$y_n = x_n$$

Each output value y_n is exactly the same as the corresponding input value x_n:

$$y_0 = x_0$$
$$y_1 = x_1$$
$$y_2 = x_2$$
$$.....$$
$$etc$$

This is a trivial case in which the filter has no effect on the signal.

Examples of simple digital filters

2. Simple gain filter:

$$y_n = Kx_n$$

where K = constant.

This simply applies a gain factor K to each input value.

$K > 1$ makes the filter an amplifier, while $0 < K < 1$ makes it an attenuator. $K < 0$ corresponds to an inverting amplifier.

Examples of simple digital filters

3. Pure delay filter:

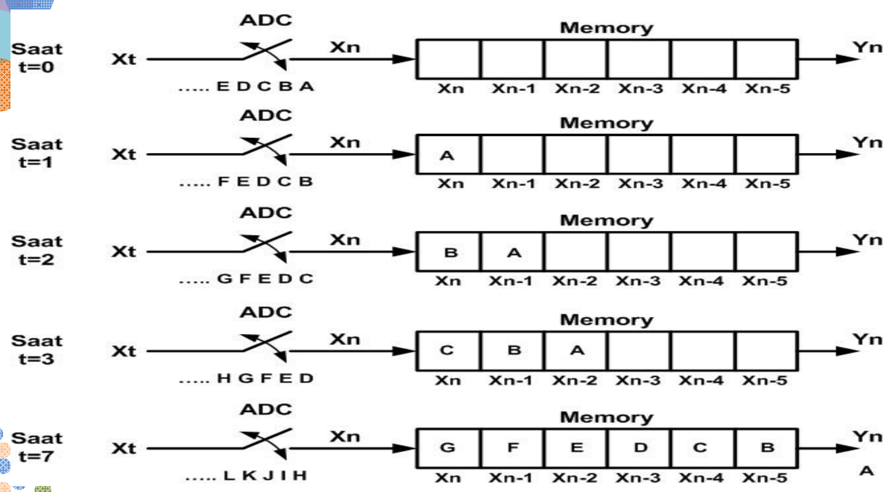
$$y_n = x_{n-1}$$

The output value at time $t = nh$ is simply the input at time $t = (n-1)h$, i.e. the signal is delayed by time h:

$$y_0 = x_{-1}$$
$$y_1 = x_0$$
$$y_2 = x_1$$
$$y_3 = x_2$$
$$\ldots$$
$$\text{etc}$$

Note that as sampling is assumed to commence at $t = 0$, the input value x_{-1} at $t = -h$ is undefined. It is usual to take this (and any other values of x prior to $t = 0$) as zero.

Delay Filter



413

Examples of simple digital filters

4. Two-term difference filter:

$$y_n = x_n - x_{n-1}$$

The output value at $t = nh$ is equal to the difference between the current input *xn* and the previous input *xn-1*:

$$y_0 = x_0 - x_{-1}$$
$$y_1 = x_1 - x_0$$
$$y_2 = x_2 - x_1$$
$$y_3 = x_3 - x_2$$
... etc

i.e. the output is the *change* in the input over the most recent sampling interval *h*. The effect of this filter is similar to that of an analog differentiator circuit.

414

Examples of simple digital filters

5. Two-term average filter:

$$y_n = \frac{x_n + x_{n-1}}{2}$$

The output is the average (arithmetic mean) of the current and previous input:

$$y_0 = \frac{x_0 + x_{-1}}{2}$$

$$y_1 = \frac{x_1 + x_0}{2}$$

$$y_2 = \frac{x_2 + x_1}{2}$$

$$..... etc$$

This is a simple type of low pass filter as it tends to smooth out high-frequency variations in a signal.

(We will look at more effective low pass filter designs later).

415

Examples of simple digital filters

6. Three-term average filter:

$$y_n = \frac{x_n + x_{n-1} + x_{n-2}}{2}$$

This is similar to the previous example, with the average being taken of the current and two previous inputs:

$$y_0 = \frac{x_0 + x_{-1} + x_{-2}}{2}$$

$$y_1 = \frac{x_1 + x_0 + x_{-1}}{2}$$

$$y_2 = \frac{x_2 + x_1 + x_0}{2}$$

$$..... etc$$

As before, *x-1* and *x-2* are taken to be zero.

416

Examples of simple digital filters

7. Central difference filter:

$$y_n = \frac{x_n - x_{n-2}}{2}$$

This is similar in its effect to example (4). The output is equal to half the change in the input signal over the previous two sampling intervals:

$$y_0 = \frac{x_0 + x_{-2}}{2}$$

$$y_1 = \frac{x_1 + x_{-1}}{2}$$

$$y_2 = \frac{x_2 + x_0}{2}$$

$$\ldots\ldots etc$$

Order of a digital filter

The *order* of a digital filter is the number of *previous* inputs (stored in the processor's memory) used to calculate the current output.

Digital filter coefficients

All of the digital filter examples given above can be written in the following general forms:

Zero order: $y_n = a_0.x_n$

First order: $y_n = a_0.x_n + a_1.x_{n-1}$

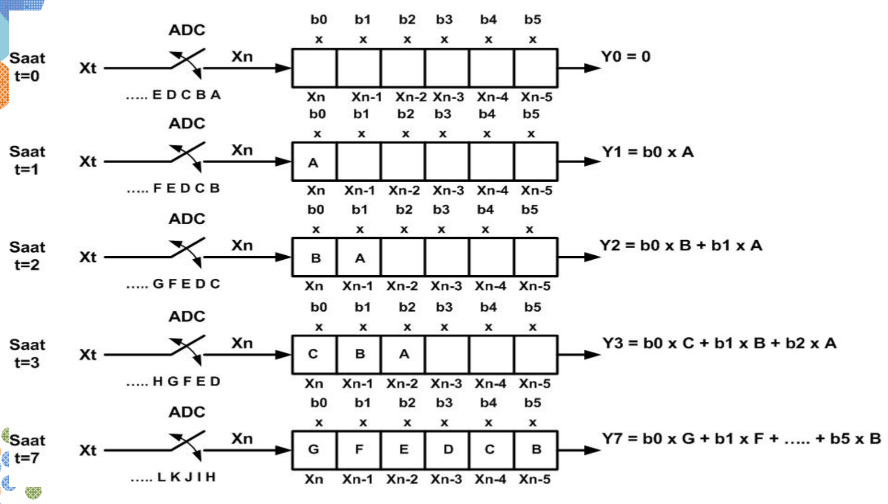
Second order: $y_n = a_0.x_n + a_1.x_{n-1} + a_2.x_{n-2}$

Similar expressions can be developed for filters of any order.

The constants *a0, a1, a2, ...* appearing in these expressions are called the *filter coefficients*. It is the values of these coefficients that determine the characteristics of a particular filter.

419

Proses Kerja FIR Filter



420

Program FIR

```
float buff[]={0,0,0};   // mengosongkan buffer memory

void FIR(int data)
{
     int orde = 2, i;
     float b[] = {0.5, 1, 0.5};
     float buff[10], temp,y;

     for(i=orde+1;i>=0;i--)          // Menggeser isi buffer memory
             buff[i]=buff[i-1];
     buff[0]=data;

     temp=0;
     for(i=0;i<orde+1;i++)           // Perhitungan konvolusi output
             temp+=buff[i]*b[i];
     y=temp;
}
```

421

The transfer function of a digital filter

- In the last section, we used two different ways of expressing the action of a digital filter: a form giving the output *yn* directly, and a "symmetrical" form with all the output terms on one side and all the input terms on the other.
- In this section, we introduce what is called the *transfer function* of a digital filter. This is obtained from the symmetrical form of the filter expression, and it allows us to describe a filter by means of a convenient, compact expression. We can also use the transfer function of a filter to work out its frequency response.
- First of all, we must introduce the *delay operator*, denoted by the symbol z^{-1}.

422

The transfer function of a digital filter

- When applied to a sequence of digital values, this operator gives the *previous* value in the sequence. It therefore in effect introduces a delay of one sampling interval.
- Applying the operator z^{-1} to an input value (say x_n) gives the previous input (x_{n-1}):

$$z^{-1} \cdot x_n = x_{n-1}$$

423

The transfer function of a digital filter

- Contoh :

$$x_0 = 5$$
$$x_1 = -2$$
$$x_2 = 10$$

$$z^{-1}.x_1 = x_0 = 5$$
$$z^{-1}.x_2 = x_1 = -2$$

424

The transfer function of a digital filter

- Similarly, applying the z^{-1} operator to an output gives the previous output:

$$z^{-1}.y_n = y_{n-1}$$

- Applying the delay operator z^{-1} twice produces a delay of two sampling intervals:

$$z^{-1}.(z^{-1}.x_n) = z^{-1}.x_{n-1} = x_{n-2}$$

- We adopt the (fairly logical) convention :

$$z^{-1}.\ z^{-1} = z^{-2}$$

425

The transfer function of a digital filter

- **Aplikasi z^{-1} dalam IIR Filter design :**

$$a_0 \cdot y_n + a_1 \cdot y_{n-1} + a_2 \cdot y_{n-2} = b_0 \cdot x_n + b_1 \cdot x_{n-1} + b_2 \cdot x_{n-2}$$

$$a_0 \cdot y_n + a_1 \cdot y_n \cdot z^{-1} + a_2 \cdot y_n \cdot z^{-2} = b_0 \cdot x_n + b_1 \cdot x_n \cdot z^{-1} + b_2 \cdot x_n \cdot z^{-2}$$

$$(a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}) \cdot y_n = (b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}) \cdot x_n$$

$$H(z) = \frac{y_n}{x_n} = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}}$$

426

The transfer function of a digital filter

- **Aplikasi z^{-1} dalam FIR Filter design :**

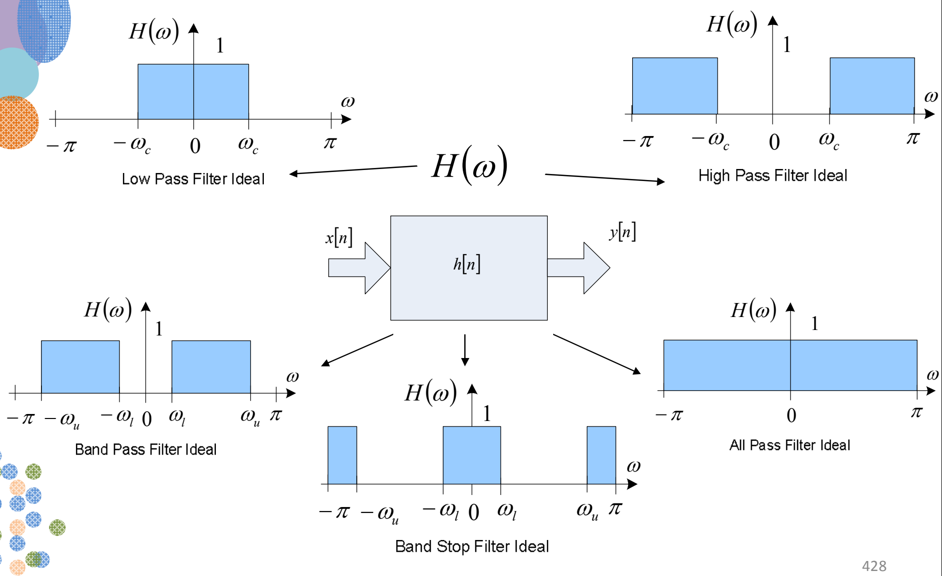
$$a_0 \cdot y_n = b_0 \cdot x_n + b_1 \cdot x_{n-1} + b_2 \cdot x_{n-2}$$

$$a_0 \cdot y_n = b_0 \cdot x_n + b_1 \cdot x_n \cdot z^{-1} + b_2 \cdot x_n \cdot z^{-2}$$

$$a_0 \cdot y_n = (b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}) \cdot x_n$$

$$H(z) = \frac{y_n}{x_n} = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{a_0}$$

Sistem Sebagai Filter



Filter digital sebagai implementasi LCCDE

Persamaan perbedaan (Difference Equation)

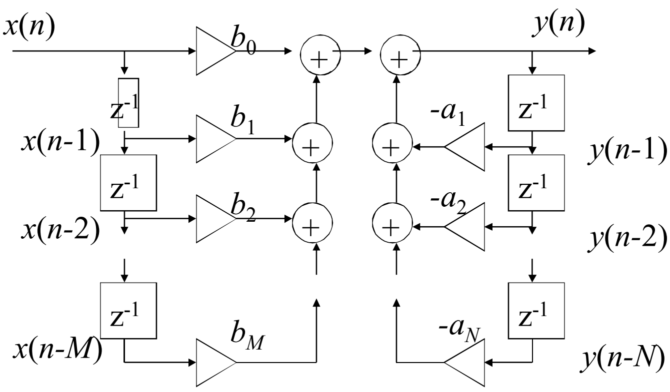
$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \qquad H(\omega) = \frac{\sum_{k=0}^{M} b_k e^{-j\omega k}}{1 + \sum_{k=1}^{N} a_k e^{-j\omega k}}$$

429

Implementasi Sistem Waktu Diskrit

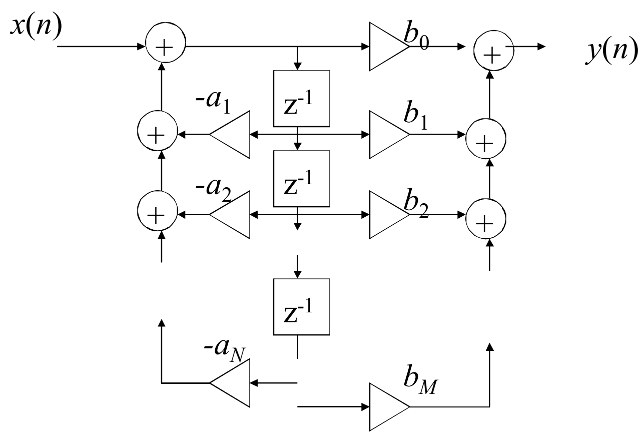
Direct Form Tipe 1



430

Implementasi Sistem Waktu Diskrit

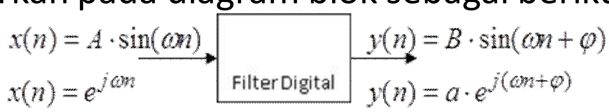
Direct Form Tipe 2

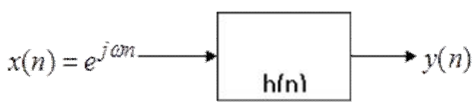


431

Respon Frekuensi Filter Digital

Sebuah sistem filter digital secara umum digambarkan pada diagram blok sebagai berikut:





Sistem menerima masukan berupa sinyal dengan frekuensi tertentu → $x(n) = e^{j\omega n}$

432

Respon Frekuensi Filter Digital

Jika konvolusi sbb: $y(n) = \sum_{k=-\infty}^{\infty} h(k) \cdot x(n-k) \quad ==> domainwaktu$

Mendapatkan masukan: $x(n) = e^{j\omega n}$

Maka: $y(n) = \sum_{k=-\infty}^{\infty} h(k) \cdot e^{j\omega(n-k)}$

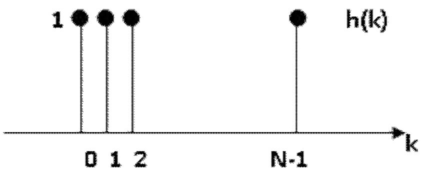
$$y(n) = e^{j\omega n} \cdot \sum_{k=-\infty}^{\infty} h(k) \cdot e^{-j\omega k}$$

$$y(n) = e^{j\omega n} \cdot H(e^{j\omega})$$

Sinyal Semula — Harga amplitudo yang bergantung pada frekuensi sinyal input ω

Contoh

- Respon frekuensi LPF digital



Jika diketahui respon impulse h(k), maka carilah respon frekuensinya!

Contoh

- Jawaban: $H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h(k) \cdot e^{-j\omega k}$

$$H(e^{j\omega}) = \sum_{k=0}^{N-1} 1 \cdot e^{-j\omega k}$$

$$H(e^{j\omega}) = \sum_{k=0}^{N-1} e^{-j\omega k}$$

$$H(e^{j\omega}) = \sum_{k=0}^{N-1} e^{-[j\omega]k}$$

Jika $\sum_{k=0}^{N} a^k = \dfrac{1-a^{N+1}}{1-a}$, maka :

435

Contoh

$$H(e^{j\omega}) = \sum_{k=0}^{N} [e^{-j\omega}]k = \frac{1-e^{-j\omega N}}{1-e^{-j\omega}}$$

$$H(e^{j\omega}) = \frac{e^{-j\omega N/2}[e^{j\omega N/2} - e^{-j\omega N/2}]}{e^{-j\omega/2}[e^{j\omega/2} - e^{-j\omega/2}]}$$

$$H(e^{j\omega}) = e^{-j(N-1)\omega/2} \cdot \frac{[e^{j\omega N/2} - e^{-j\omega N/2}]/2j}{[e^{j\omega/2} - e^{-j\omega/2}]/2j}$$

Jika $\sin\alpha = \dfrac{e^{j\alpha} - e^{-j\alpha}}{2j}$, maka :

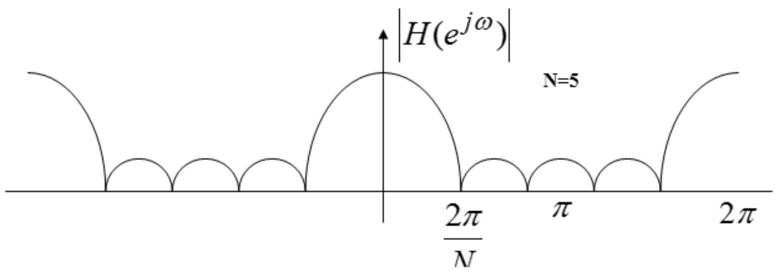
$$H(e^{j\omega}) = e^{-j(N-1)\omega/2} \cdot \frac{\sin(\omega N/2)}{\sin(\omega/2)}$$

436

Contoh

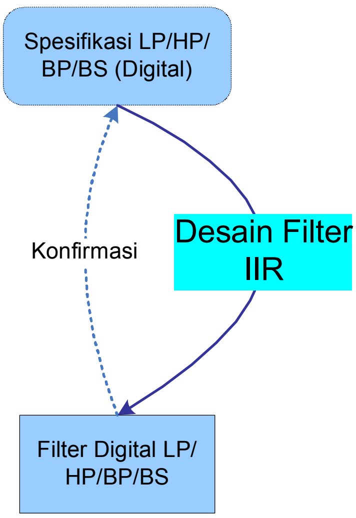
Sehingga magnitude respon frekuensi :

$$H(e^{j\omega}) = \left| \frac{\sin(\omega N/2)}{\sin(\omega/2)} \right| \quad ==> Fungsi\ kontinyu$$



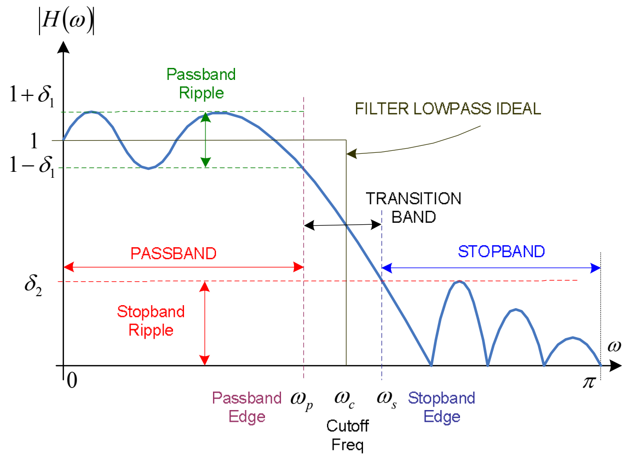
Proses Desain IIR

- Mulai dengan spesifikasi respons frekuensi
- Cari koefisien {ak,bk} sehingga terbentuk LCCDE dengan respons frekuensi yang diinginkan
- Konfirmasi bahwa respons frekuensi yang dihasilkan LCCDE ini memenuhi spesifikasi



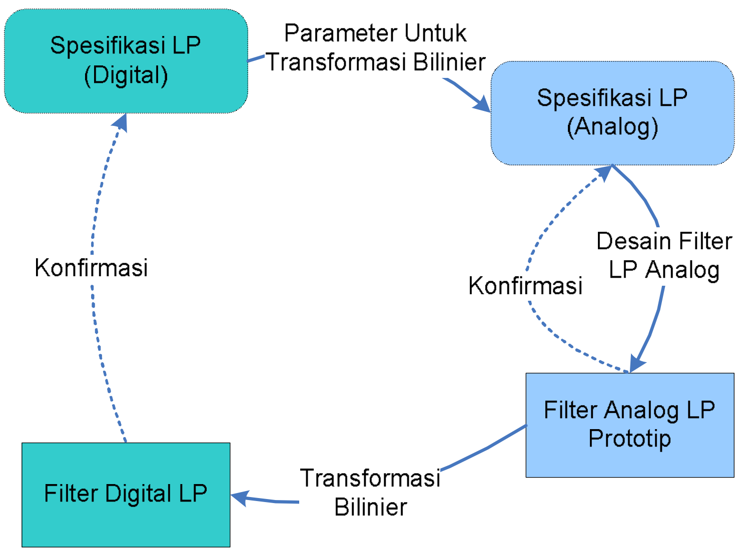
Problem Desain LCCDE

- Cari $\{a_k\}$ dan $\{b_k\}$ agar $|H(\omega)|$ dari LCCDE mendekati filter ideal



439

Proses Desain Filter LP Digital



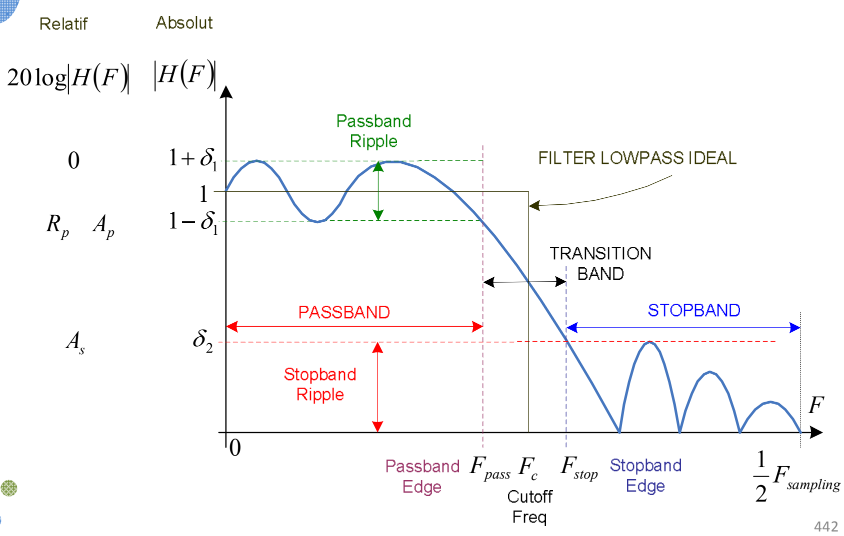
440

Analisa Lowpass Filter Analog

Contoh: LPF ini memiliki pass band edge dan stopband edge masing-masing $F_{pass} = 800$ Hz dan $F_{stop} = 1200$ Hz. Ripple pada passband dan stopband masing masing adalah δ_1 dan δ_2, dan diinginkan $R_p \leq 1$ dB serta $A_s > 50$ dB.

441

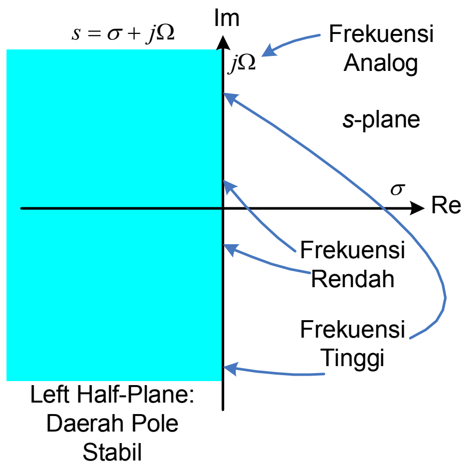
Spesifikasi Filter Analog Lowpass



442

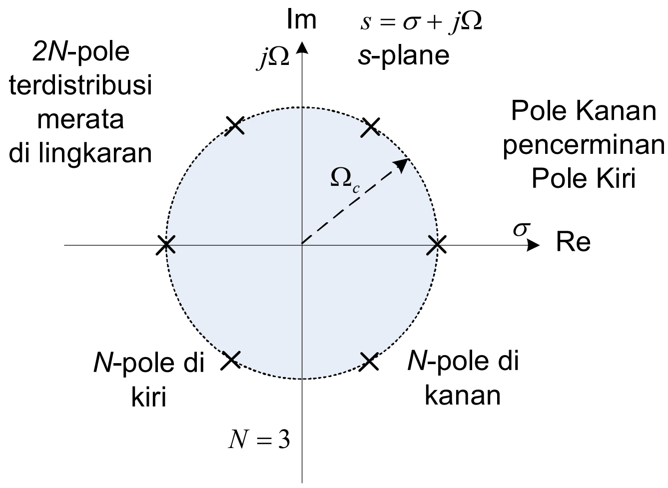
s-plane

- Domain untuk Laplace Transform
- Bagian Imajiner merepresentasikan frekuensi analog
- Filter Analog dinyatakan dengan fungsi sistem $H(s)$
- Left Half-Plane merupakan lokasi pole $H(s)$ yang stabil
- $H(s)$ di evaluasi pada sumbu imajiner adalah respons frekuensinya



443

Filter LP Analog Butterworth



444

Filter Analog Butterworth

$$H_a(s) = \frac{\Omega_c^N}{\prod\limits_{Pole\,Kiri} (s - p_k)}$$

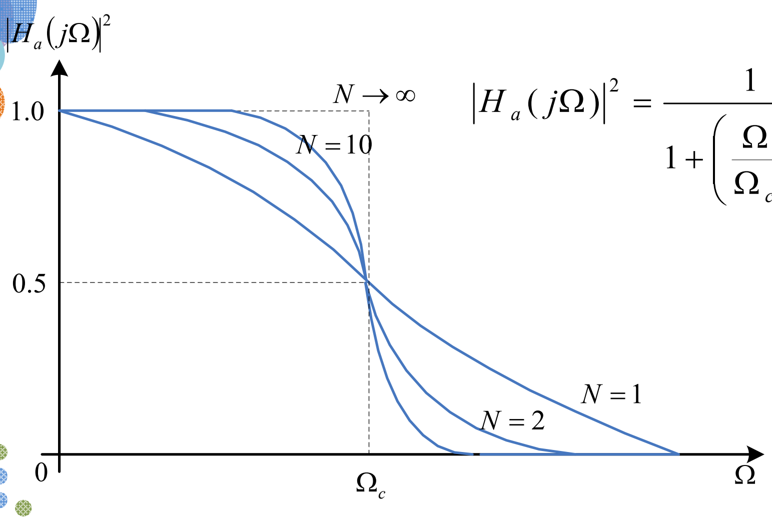
$$|H_a(j\Omega)|^2 = H_a(j\Omega)H_a^*(j\Omega) = H_a(j\Omega)H_a(-j\Omega) = H_a(s)H_a(-s)\big|_{s=j\Omega}$$

$$H_a(s)H_a(-s) = |H_a(j\Omega)|^2\Big|_{\Omega = s/j}$$

$$|H_a(j\Omega)|^2 = \frac{\Omega_c^{2N}}{\Omega^{2N} + \Omega_c^{2N}}$$

$$|H_a(j\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}}$$

Karakteristik LP Butterworth



$$|H_a(j\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}}$$

Soal Lowpass Filter Analog

Contoh: LPF Butterworth ini memiliki pass band edge dan stopband edge masing-masing $F_{pass} = 800$ Hz dan $F_{stop} = 1200$ Hz. Ripple pada passband dan stopband masing masing adalah δ_1 dan δ_2, dan diinginkan $R_p \le 1$ dB serta $A_s > 50$ dB. Cari N

447

Jawaban

$$R_p = -10\log_{10}\left|H_a(j\Omega_p)\right|^2 \qquad R_p = -10\log_{10}\left(\frac{1}{1+\left(\frac{\Omega_p}{\Omega_c}\right)^{2N}}\right)$$

$$A_s = -10\log_{10}\left|H_a(j\Omega_s)\right|^2 \qquad A_s = -10\log_{10}\left(\frac{1}{1+\left(\frac{\Omega_s}{\Omega_c}\right)^{2N}}\right)$$

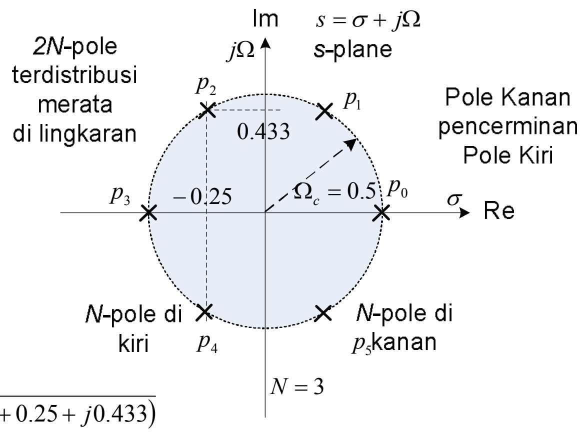
$$N = \frac{\log_{10}\left[\left(10^{R_p/10}-1\right)\left(10^{A_s/10}-1\right)\right]}{2\log_{10}\left(\Omega_p/\Omega_s\right)}$$

$$\Omega_c = \frac{\Omega_p}{\sqrt[2N]{10^{\frac{R_p}{10}}-1}} \qquad \Omega_c = \frac{\Omega_s}{\sqrt[2N]{10^{\frac{A_s}{10}}-1}}$$

448

Filter Untuk *N*=3

$$H_a(s) = \frac{\Omega_c^3}{(s-p_2)(s-p_3)(s-p_4)}$$

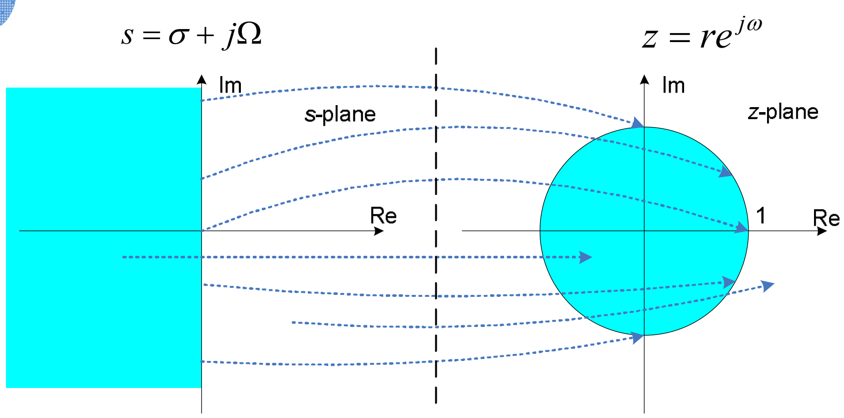


$$H_a(s) = \frac{0.5^3}{(s+0.25-j0.433)(s+0.5)(s+0.25+j0.433)}$$

$$H_a(s) = \frac{0.125}{(s+0.5)(s^2+0.5s+0.25)}$$

449

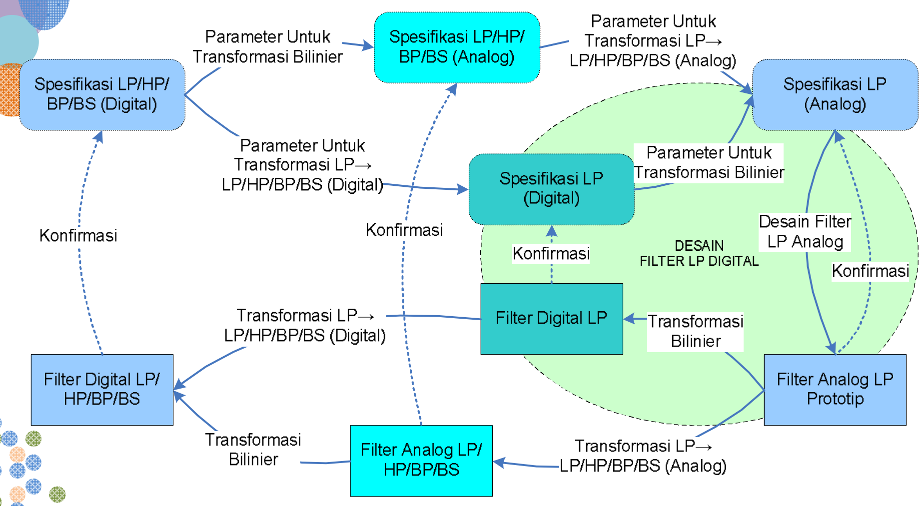
Transformasi Bilinier



$$s = \frac{2}{T}\frac{z-1}{z+1} = \frac{2}{T}\frac{re^{j\omega}-1}{re^{j\omega}+1}$$

450

Detail Proses Desain Filter IIR



451