



FILTER DIGITAL:

FINITE IMPULSE RESPONSE DAN INFINITE IMPULS RESPONSE

FILTER DIGITAL

Filter merupakan suatu sistem yang mempunyai fungsi transfer tertentu untuk meloloskan sinyal masukan pada frekuensi - frekuensi tertentu dan menyaring / memblokir / melemahkan sinyal masukan pada frekuensi-frekuensi yang lain.

Filter digital adalah semua filter elektronik yang bekerja dengan menerapkan operasi matematika digital atau algoritma pada suatu pemrosesan sinyal. Salah satu batasan utama pada filter digital adalah dalam hal keterbatasan kecepatan pemrosesan/waktu komputasi yang sangat tergantung dengan kemampuan mikrokontroler atau komputer yang digunakan.

Filter digital merupakan salah satu bagian dalam teknologi elektronika digital, filter ini biasanya digunakan dalam bidang pengolahan sinyal suara dan telekomunikasi. Filter digital dapat dibagi menjadi dua, yaitu filter digital FIR (Finite Impulse Response) dan filter digital IIR (Infinite Impulse Respons)

CONTOH PENGGUNAAN FILTER DIGITAL PADA PEMROSESAN SINYAL

Filter Rendah Butterworth pada Sinyal EKG ([Tautan Paper](#))

Sinyal EKG (Elektrokardiogram) adalah sinyal yang merekam aktivitas listrik jantung. Filter digital dapat digunakan untuk membersihkan sinyal EKG dari **noise frekuensi tinggi** yang tidak diinginkan.

1. Misalkan kita memiliki sinyal EKG yang mengandung noise frekuensi tinggi dan ingin membersihkannya untuk mendapatkan sinyal yang lebih jelas dan terfokus pada aktivitas jantung.
2. Terapkan filter rendah **Butterworth** pada sinyal EKG. Filter Butterworth merupakan salah satu jenis filter yang sering digunakan dalam pemrosesan sinyal. Filter ini dirancang untuk memberikan respon frekuensi datar di daerah frekuensi yang diinginkan dan memotong frekuensi di atas batas **cutoff**.

Hasil filter rendah Butterworth pada sinyal EKG akan membuat komponen noise frekuensi tinggi akan dikurangi atau dihilangkan, sedangkan komponen frekuensi rendah yang terkait dengan aktivitas jantung akan tetap terjaga. Hasilnya adalah sinyal EKG yang lebih bersih dan lebih mudah untuk dianalisis secara klinis.

Dalam implementasi nyata, penggunaan filter digital melibatkan desain filter dengan mempertimbangkan spesifikasi yang diinginkan, seperti **cutoff frequency, orde filter, dan jenis filter yang digunakan**.

Langkah-langkah pemrosesan sinyal dan penggunaan algoritma filter digital umumnya menggunakan penggunaan perangkat lunak seperti MATLAB atau Python untuk mengimplementasikan filter dan menganalisis hasilnya.

FINITE IMPULSE RESPONSE (FIR)

Filter Digital **Finite Impulse Response** (FIR) adalah salah satu jenis filter digital yang sangat umum digunakan dalam pemrosesan sinyal. FIR filter memiliki respons impuls dengan durasi terbatas, yang berarti filter ini merespons sinyal input dengan mengalikan koefisien filter dengan sinyal input dan menjumlahkannya.

Filter FIR dapat dinyatakan dalam bentuk matematis sebagai berikut:

$$y[n] = b_0 * x[n] + b_1 * x[n-1] + b_2 * x[n-2] + \dots + b_M * x[n-M]$$

Di mana:

- $y[n]$ adalah keluaran filter pada waktu diskret n
- $x[n]$ adalah sinyal input pada waktu diskret n
- b_0, b_1, \dots, b_M adalah koefisien filter FIR yang ditentukan oleh desain filter. Koefisien ini mempengaruhi karakteristik respons frekuensi dan waktu dari filter FIR.
- M adalah panjang filter FIR, yang menentukan jumlah koefisien dalam filter.

ELEMEN UTAMA

1. Koefisien Filter

Koefisien filter FIR menentukan respons frekuensi dan karakteristik filter. Setiap koefisien mengontrol **amplitudo** dan **fase** kontribusi frekuensi tertentu pada sinyal output. Koefisien ini harus dipilih dengan hati-hati untuk mencapai karakteristik filter yang diinginkan, seperti pemotongan frekuensi tinggi atau rendah, penekanan band-pass, atau karakteristik pita-passthrough lainnya.

2. Konvolusi

Proses utama dalam filter FIR adalah konvolusi antara koefisien filter dan sinyal input. Konvolusi dilakukan dengan menggeser koefisien filter sepanjang sinyal input dan melakukan perkalian titik pada setiap titik waktu. Hasil perkalian titik ini kemudian dijumlahkan untuk menghasilkan sinyal output.

Konvolusi antara dua sinyal diskrit $x[n]$ dan $v[n]$ dapat dinyatakan sebagai:

$$x[n] * v[n] = \sum_{i=-\infty}^{\infty} x[i]v[n-i]$$

Bentuk penjumlahan yang ada di bagian kanan pada persamaan (1) disebut sebagai convolution sum. Jika $x[n]$ dan $v[n]$ memiliki nilai 0 untuk semua integer pada $n < 0$, selanjutnya $x[i]=0$ untuk semua integer pada $i < 0$ dan $v[i-n]=0$ untuk semua integer $n - i < 0$ (atau $n < i$). Sehingga jumlahan pada persamaan (1) akan menempati dari nilai $i=0$ sampai dengan $i=n$, dan operasi konvolusi selanjutnya dapat dituliskan sebagai:

$$x[n] * v[n] = \begin{cases} 0 & , n = -1, -2, \dots \\ \sum_{i=0}^n x[i]v[n-i] & , n = 0, 1, 2, \dots \end{cases}$$

3. Jendela Waktu Terbatas

Filter FIR secara intrinsik memiliki jendela waktu terbatas atau respons impuls terbatas. Ini berarti filter hanya merespons sinyal input dalam jendela waktu terbatas, dan tidak ada pengaruh pada waktu sebelumnya atau sesudahnya. Durasi jendela waktu ini ditentukan oleh **panjang koefisien** filter.

Window Filter FIR Causal

- Rectangular:

$$w(n)=1 \quad 0 \leq n \leq M-1$$

- Hann:

$$w(n)=0.5-0.5\cos(2\pi\frac{n}{M-1}) \quad 0 \leq n \leq M-1$$

- Hamming:

$$w(n)=0.54-0.46\cos(2\pi\frac{n}{M-1}) \quad 0 \leq n \leq M-1$$

- Blackman:

$$w(n)=0.42-0.46\cos(2\pi\frac{n}{M-1})+0.08\cos(2\pi\frac{2n}{M-1}) \quad 0 \leq n \leq M-1$$

Metode yang bisa diterapkan pada FIR misalnya **desain windowing, metode desain frekuensi, atau metode desain optimal**. Dalam praktiknya, filter FIR dapat diimplementasikan menggunakan algoritma **konvolusi langsung** atau dengan metode efisien seperti **Fast Fourier Transform (FFT)** dalam implementasi filter FIR berkecepatan tinggi.

- Kelebihan FIR: Karakteristik fase linear dan kestabilan numerik yang baik
- Kelemahan FIR: Kebutuhan komputasi yang tinggi tergantung pada jumlah koefisien filter.

CONTOH PENERAPAN FIR

1. Implementasi pada Matlab

```
% Membuat sinyal input
fs = 1000; % Frekuensi sampel
t = 0:1/fs:1; % Rentang waktu dari 0 hingga 1 detik
f1 = 10; % Frekuensi komponen 1 (Hz)
f2 = 50; % Frekuensi komponen 2 (Hz)
x = sin(2*pi*f1*t) + 0.5*sin(2*pi*f2*t); % Sinyal input dengan
dua komponen frekuensi

% Mendefinisikan koefisien filter FIR
b = [0.2, 0.7, 0.3, 0.1]; % Koefisien filter FIR
```

```
% Menggunakan fungsi filter untuk menerapkan filter FIR pada
sinyal input
y = filter(b, 1, x); % Keluaran filter FIR

% Plot sinyal input dan keluaran filter
subplot(2,1,1);
plot(t, x);
xlabel('Waktu (s)');
ylabel('Amplitudo');
title('Sinyal Input');
subplot(2,1,2);
plot(t, y);
xlabel('Waktu (s)');
ylabel('Amplitudo');
title('Keluaran Filter FIR');
```

Pada contoh di atas, langkah-langkah yang diterapkan adalah sebagai berikut:

1. Membuat sinyal input dengan dua komponen frekuensi, yaitu 10 Hz dan 50 Hz.
2. Mendefinisikan koefisien filter FIR sebagai vektor **b**. Dalam contoh ini, kita menggunakan koefisien **[0.2, 0.7, 0.3, 0.1]**.
3. Menggunakan fungsi **filter** dalam MATLAB untuk menerapkan filter FIR pada sinyal input. Output filter FIR disimpan dalam variabel **y**. Parameter kedua fungsi **filter** adalah 1, yang menunjukkan bahwa tidak ada koefisien feedback yang digunakan dalam filter FIR.
4. Menampilkan plot sinyal input dan keluaran filter FIR menggunakan fungsi **plot**. Pada subplot pertama, ditampilkan sinyal input, sedangkan pada subplot kedua, ditampilkan keluaran filter FIR.

2. Teknik Windowing

Di MATLAB, pengguna dapat menggunakan fungsi **fir1** untuk mendesain filter FIR dengan Teknik Windowing. Berikut adalah contoh penggunaan **fir1** untuk mendesain filter FIR teknik tersebut:

```
% Spesifikasi filter
fs = 1000; % Frekuensi sampel
fc = 100; % Frekuensi cut-off (Hz)
N = 51; % Panjang filter FIR (jumlah koefisien)
window = hamming(N); % Fungsi jendela Hamming

% Mendefinisikan koefisien filter FIR dengan teknik windowing
h = fir1(N-1, fc/(fs/2), window);

% Menampilkan respons frekuensi filter FIR
fvtool(h, 1);
```

Algoritma Hamming Window:

$$w(n) = 0.54 - 0.46 \cos\left(2\pi \frac{n}{N}\right), \quad 0 \leq n \leq N.$$

Panjang Windows $L = N + 1$.

Pada contoh di atas, kita menggunakan fungsi **fir1** dengan argumen sebagai berikut:

- **N-1** menentukan panjang filter FIR (jumlah koefisien). Dalam contoh ini, kita menggunakan $N=51$ sehingga menggunakan **N-1 = 50** sebagai argumen.
- **fc/(fs/2)** adalah frekuensi cut-off relatif terhadap separuh frekuensi sampel (Nyquist frequency). Dalam contoh ini, kita menggunakan frekuensi cut-off 100 Hz dan frekuensi sampel 1000 Hz, sehingga **fc/(fs/2) = 100/500 = 0.2**.
- **window** adalah fungsi jendela yang digunakan dalam desain filter FIR. Dalam contoh ini, kita menggunakan fungsi jendela Hamming dengan panjang $N=51$.

Setelah menjalankan kode di atas, akan didapatkan koefisien filter FIR dalam variabel **h**. Koefisien tersebut digunakan untuk menerapkan filter FIR pada sinyal input menggunakan fungsi **filter**. Sangat penting untuk melakukan spesifikasi filter FIR, panjang filter, frekuensi cut-off, dan fungsi jendela sesuai dengan kebutuhan penerapan.

INFINITE IMPULSE RESPONSE

Filter Digital **Infinite Impulse Response (IIR)** adalah jenis filter digital lain yang banyak digunakan dalam pemrosesan sinyal. IIR filter memiliki respons impuls yang **tak terbatas**, yang berarti filter ini dapat merespons sinyal input dalam rentang waktu tak terbatas.

Filter IIR dapat dinyatakan dalam bentuk matematis sebagai berikut:

$$y[n] = b_0 * x[n] + b_1 * x[n-1] + b_2 * x[n-2] + \dots + b_M * x[n-M] - a_1 * y[n-1] - a_2 * y[n-2] - \dots - a_N * y[n-N]$$

Di mana:

- $y[n]$ adalah keluaran filter pada waktu diskret n
- $x[n]$ adalah sinyal input pada waktu diskret n
- b_0, b_1, \dots, b_M adalah koefisien feedforward yang menghubungkan sampel input dengan keluaran filter. Koefisien ini mempengaruhi karakteristik respons frekuensi filter.

- a_1, a_2, \dots, a_N adalah koefisien feedback yang menghubungkan sampel keluaran filter masa lalu dengan keluaran saat ini. Koefisien ini mempengaruhi karakteristik respons frekuensi dan stabilitas filter.
- M adalah panjang feedforward filter (jumlah koefisien feedforward)
- N adalah panjang feedback filter (jumlah koefisien feedback)

Dalam rumus di atas, setiap koefisien **feedforward b** dihubungkan dengan sampel input x pada waktu sebelumnya, sedangkan setiap koefisien feedback a dihubungkan dengan sampel keluaran y pada waktu sebelumnya. **Koefisien filter IIR** menentukan seberapa banyak kontribusi sampel input dan keluaran masa lalu pada keluaran saat ini.

ELEMEN UTAMA

1. Koefisien Filter

Koefisien filter IIR menentukan **respons frekuensi dan karakteristik filter**. IIR filter diimplementasikan sebagai perpaduan antara koefisien filter feedforward (feedforward coefficients) dan feedback (feedback coefficients). Koefisien ini mempengaruhi amplitudo, fase, dan karakteristik frekuensi filter. Koefisien filter IIR ditentukan melalui desain filter menggunakan metode yang sesuai.

2. Fungsi Transfer

Filter IIR dijelaskan oleh fungsi transfer yang merupakan **rasio antara output filter dengan input filter dalam domain frekuensi**. Fungsi transfer dapat ditulis dalam bentuk persamaan diferensial atau dalam bentuk persamaan perbedaan (difference equation) yang menggambarkan hubungan antara sinyal input, sinyal output, dan koefisien filter.

3. Feedback Loop

Fitur kunci dari filter IIR adalah adanya loop umpan balik (feedback loop). **Loop ini memungkinkan sinyal output kembali ke input filter, menciptakan respons frekuensi yang kompleks dan karakteristik filter yang berbeda.** Loop umpan balik ini memungkinkan filter IIR untuk memiliki respons frekuensi yang tajam dan kemampuan untuk merespons frekuensi dengan durasi panjang.

4. Pemrosesan Rekursif

Filter IIR bekerja dengan melakukan operasi rekursif pada sinyal input dan output sebelumnya. Hal ini berarti bahwa output filter saat ini bergantung pada input saat ini dan output sebelumnya. Pemrosesan rekursif ini

memungkinkan filter IIR untuk memiliki **respons frekuensi** yang kompleks dengan **orde filter** yang relatif rendah.

Dalam desain filter IIR, beberapa metode yang umum digunakan antara lain metode desain **Butterworth, Chebyshev, dan Elliptic**. Desain filter IIR memperhatikan kriteria seperti respons frekuensi, karakteristik pita lintang, redaman pita-henti, dan keseimbangan antara kelancaran fase dan kecepatan transisi.

- Kelebihan: Orde filter yang relatif rendah untuk mencapai respons frekuensi yang kompleks, efisiensi komputasi yang tinggi, dan desain yang fleksibel.
- Kelemahan: karakteristik fase non-linear dan potensi ketidakstabilan numerik jika tidak dirancang dengan baik.

CONTOH PENERAPAN IIR (Butterworth)

Implementasi pada Matlab

```
% Spesifikasi filter
fs = 1000; % Frekuensi sampel (Hz)
fc = 100; % Frekuensi cut-off (Hz)
order = 4; % Orde filter

% Membuat sinyal input
t = 0:1/fs:1; % Rentang waktu
x = sin(2*pi*50*t) + sin(2*pi*150*t); % Sinyal input dengan %
dua komponen frekuensi

% Mendesain filter IIR Butterworth
[b, a] = butter(order, fc/(fs/2));

% Menerapkan filter IIR pada sinyal input
y = filter(b, a, x);

% Plot sinyal input dan keluaran filter
subplot(2,1,1);
plot(t, x);
xlabel('Waktu (s)');
ylabel('Amplitudo');
title('Sinyal Input');
subplot(2,1,2);
plot(t, y);
xlabel('Waktu (s)');
ylabel('Amplitudo');
```



```
title('Keluaran Filter IIR Butterworth');  
  
% Menampilkan respons frekuensi filter IIR  
fvtool(b, a);
```

Pada contoh di atas, digunakan filter IIR Butterworth orde 4 dengan frekuensi cut-off $f_c=100$ Hz. Sinyal input terdiri dari dua komponen frekuensi, yaitu 50 Hz dan 150 Hz. Filter IIR Butterworth dengan fungsi **butter** didesain dengan orde dan frekuensi cut-off yang ditentukan. Kemudian, filter IIR diterapkan pada sinyal input menggunakan fungsi **filter**. Hasilnya disimpan dalam variabel **y**.

Selanjutnya, menampilkan plot sinyal input dan keluaran filter menggunakan fungsi **plot**. Pada subplot pertama, ditampilkan sinyal input yang terdiri dari dua komponen frekuensi. Pada subplot kedua, ditampilkan sinyal keluaran filter IIR Butterworth.

Terakhir, untuk menampilkan respons frekuensi filter IIR digunakan fungsi **fvtool**. Fungsi ini akan menghasilkan diagram respons frekuensi filter IIR yang telah didesain.