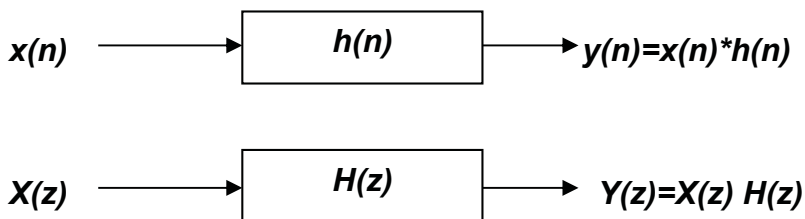


11

Filter Digital

Setelah kita mempelajari konsep filter dan filter analog pada Bab 9 dan 10 maka sekarang kita akan mempelajari tentang konsep filter digital. Setelah mempelajari bab ini maka mahasiswa akan dapat mengidentifikasi kelebihan dan kekurangan dari filter digital jika dibandingkan dengan filter analog, menentukan parameter dari filter digital, menggambarkan respon frekuensi dari filter digital, dan dapat merancang implementasi filter digital dalam bentuk perangkat keras.



Gambar 11.1

Filter digital dalam bentuk impulse respons dan transfer function

Filter digital adalah sistem diskrit (Bab 8) yang berfungsi sebagai filter. Karena merupakan sistem diskrit maka filter

digital dapat dinyatakan dalam bentuk *impulse response*, $h(n)$, atau *transfer function*, $H(z)$, seperti pada Gambar 11.1. Filter digital juga memiliki jenis yang sama dengan filter analog yaitu LPF, HPF, BPF dan BSF.

11.1 Kelebihan Filter Digital

Filter analog berupa persamaan yang diimplementasikan dalam bentuk rangkaian elektronika. Filter digital berisi angka-angka yang diproses secara aritmetika (penjumlahan, pengurangan, perkalian, dan pembagian). Filter digital ini diimplementasikan dalam bentuk program komputer atau proses digital di komputer atau mikroprosesor. Karena perbedaan ini maka filter digital memiliki beberapa kelebihan dibandingkan dengan filter analog antara lain:

- Karena hanya berupa angka dan proses aritmetika maka filter digital mudah diimplementasikan sekalipun dengan orde yang sangat tinggi. Penambahan satu orde pada filter analog berakibat pada penambahan beberapa komponen elektronika yang akan membuat fisik rangkaian menjadi lebih besar. Penambahan orde pada filter digital hanya akan menambah beberapa angka pada program komputer atau prosesor. Kita akan jarang menjumpai filter analog dengan orde lebih dari 6, sedangkan kita akan sering menjumpai filter digital ber-orde puluhan bahkan ratusan.
- Filter digital bisa memiliki respon frekuensi yang sangat bervariasi. Filter analog terbatas pada bentuk respon frekuensi seperti Butterworth, Chebyshev, dan Elliptic karena batasan ketersediaan jenis dan nilai komponen elektronika. Karena hanya berupa angka maka respon frekuensinya dapat divariasikan tanpa batasan fisik komponen. Kita dapat mendesain filter digital dengan fase yang benar-benar linier. Hal ini tidak dimungkinkan oleh filter analog.

- Filter digital *tidak dipengaruhi oleh suhu* atau gangguan luar lainnya. Nilai komponen dari filter analog dapat berubah jika suhu berubah. Hal ini tidak terjadi pada filter digital karena angka-angkanya ditentukan oleh program yang dituliskan pada prosesor.
- *Parameter dari filter digital bisa diubah dengan gampang* karena hanya berupa angka-angka pada program komputer atau mikroprosesor. Perubahan parameter pada filter analog harus dilakukan dengan mengganti komponen elektroniknya. Kelebihan ini juga memungkinkan implementasi dari adaptive filter.
- Filter digital *dapat disimpan (save) dan diduplikasi* dengan tidak ada perubahan performansi. Hal ini tidak dijamin oleh filter analog karena nilai dari komponen elektronika selalu bervariasi.
- Filter digital *dapat digunakan pada frekuensi yang sangat rendah*. Hal ini sulit dilakukan oleh filter analog karena membutuhkan nilai kapasitor yang sangat besar.

11.2 Kelemahan Filter Digital

Disamping kelebihan-kelebihan di atas, filter digital juga memiliki beberapa kelemahan jika dibandingkan dengan filter analog, antara lain:

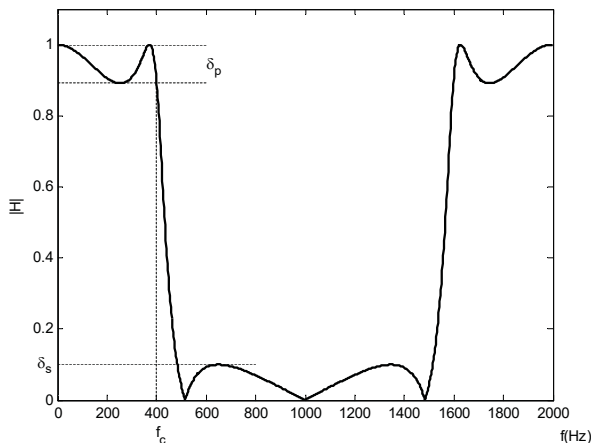
- Filter digital *memiliki range frekuensi yang terbatas* dan cenderung lebih rendah dari filter analog. Range frekuensi filter digital hanya dibatasi sampai setengah dari frekuensi samplinya. Frekuensi sampling sangat ditentukan oleh kecepatan dari ADC dan mikroprosesor yang digunakan.
- Filter digital memiliki *noise yang disebabkan oleh pembulatan angka-angka* di dalam prosesnya. Semakin

tinggi orde dari suatu filter digital maka semakin besar pula efek dari pembulatan khususnya bila ada proses recursive.

- Filter digital tidak dapat digunakan sebagai filter untuk anti aliasing karena filter ini harus diletakkan di daerah analog.
- Filter digital tidak bisa digunakan untuk memfilter sinyal dengan daya tinggi misalnya filter tegangan jala-jala.

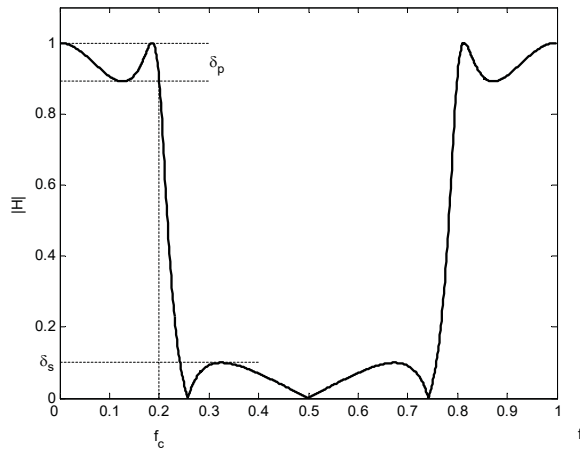
11.3 Parameter Filter Digital

Filter digital memiliki parameter yang sama dengan filter analog tetapi mengalami pencerminan pada $f_s / 2$ seperti dijelaskan pada Bab 4. Gambar 11.2 adalah filter digital LPF dengan $f_c = 400 \text{ Hz}$ dan frekuensi sampling, f_s , sebesar 2 kHz . Jika kita tidak menyadari adanya pencerminan respon frekuensi pada spektrum diskrit maka kita bisa keliru menganggap bahwa gambar ini adalah respon frekuensi dari BSF.



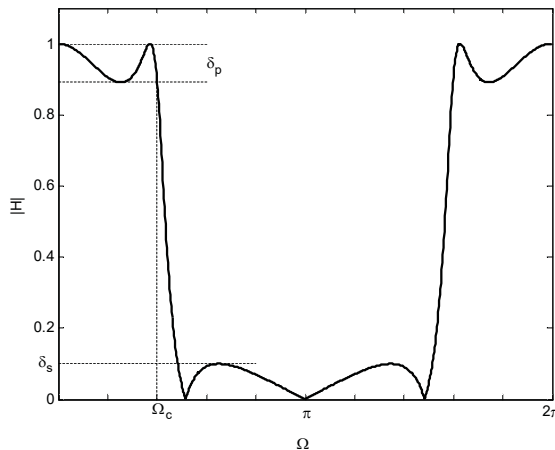
Gambar 11.2
Respon frekuensi filter digital LPF

Respon frekuensi dari filter digital sering digambarkan dalam frekuensi yang sudah dinormalisasi dimana frekuensi sampling bernilai 1 seperti pada Gambar 11.3, atau dalam skala radian ternormalisasi dimana frekuensi sampling adalah 2π dan setengah frekuensi sampling menjadi π seperti pada Gambar 11.4.



Gambar 11.3

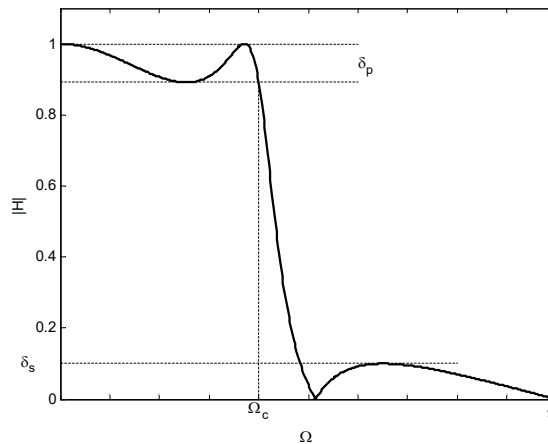
Respon frekuensi filter digital LPF dengan frekuensi ternormalisasi



Gambar 11.4

Respon frekuensi filter digital LPF dengan frekuensi radian ternormalisasi

Karena respon frekuensi pada Gambar 11.4 bersifat pencerminan pada $\Omega = \pi$, maka pada umumnya hanya digambarkan sampai dengan π seperti pada Gambar 11.5.



Gambar 11.5

Respon frekuensi filter digital LPF dengan $\Omega_c = 0.4 \pi$ (sama dengan $0.2 f_s$)

Catatan penting untuk pengguna Matlab:

Desain filter digital pada Matlab menggunakan input berupa frekuensi yang sudah dinormalisasi (seperti Gambar 11.3), tetapi frekuensi 1 bukan mewakili f_s melainkan mewakili $\frac{1}{2} f_s$.

Jika kita menggunakan Matlab untuk mendesain LPF dengan $f_c = 400 \text{ Hz}$ pada $f_s = 2000 \text{ Hz}$, maka f_c ternormalisasi adalah 0.4 (bukan 0.2).

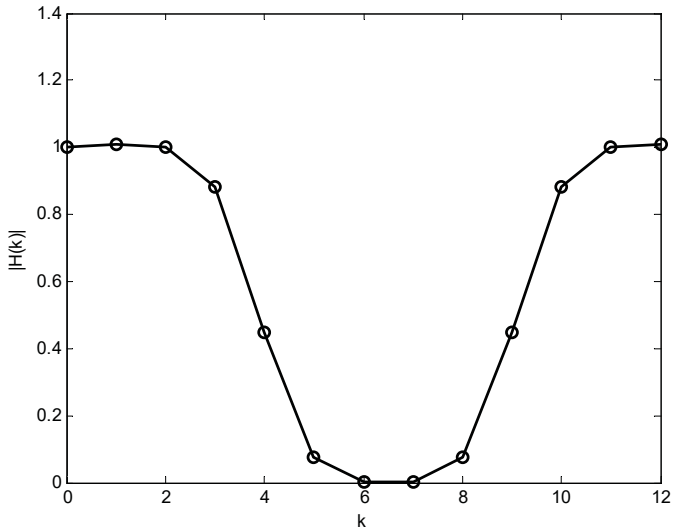
11.4 Menemukan Respon Frekuensi dari Filter

Gambar-gambar di atas adalah kurva respon frekuensi dari filter digital. Bagaimanakah kita bisa menemukan kurva respon frekuensi dari suatu filter? Jika kita mengetahui *impulse respons*-nya, $h(n)$, maka respon frekuensinya, $H(k)$ dapat dihitung dengan menggunakan DFT seperti pada Bab 4. Jika kita mengetahui transfer functionnya, $H(z)$, maka terlebih dahulu kita harus menemukan $h(n)$ dari filter tersebut kemudian baru menghitung respon frekuensinya, $H(k)$, dengan menggunakan DFT seperti pada Bab 4. Cara menemukan $h(n)$ sudah dijelaskan di Bab 9. Sebagai contoh kita akan menemukan kurva respon frekuensi dari filter dengan $h(n)$ seperti pada Tabel 11.1

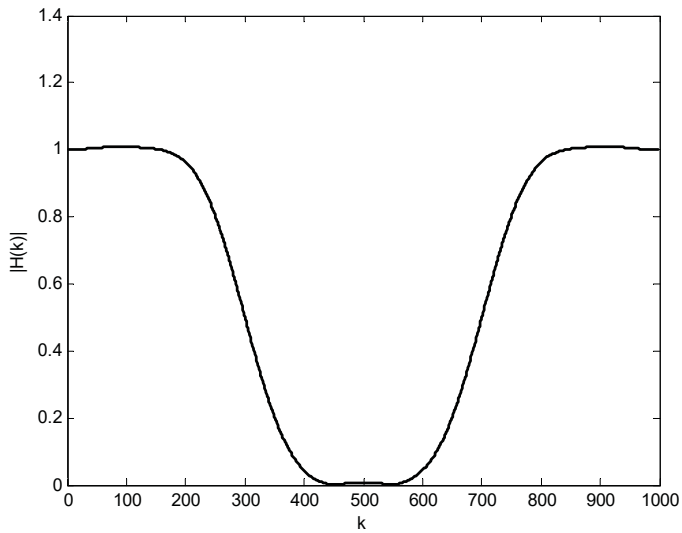
Tabel 11.1
Impulse response dari sebuah filter

n	$h(n)$
0	-0.0041
1	0.0000
2	0.0236
3	-0.0338
4	-0.0724
5	0.2853
6	0.6027
7	0.2853
8	-0.0724
9	-0.0338
10	0.0236
11	0.0000
12	-0.0041

Perhitungan DFT menghasilkan $|H(k)|$ sebanyak 12 titik seperti digambarkan pada Gambar 11.6a. Gambar respon frekuensi ini dapat dibuat lebih baik dengan *zero padding*. Gambar 11.6b menunjukkan respon frekuensi yang dihitung menggunakan DFT dengan *zero padding*.



a)

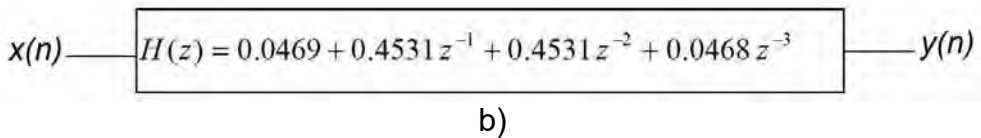
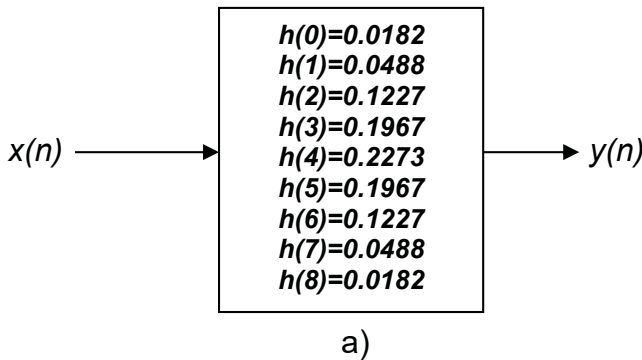


b)

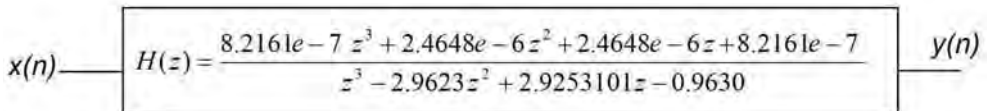
Gambar 11.6
Respon frekuensi filter pada Tabel 11.1 a) dihitung dengan DFT; b) dihitung dengan DFT dan zero padding

11.5 Tipe dari Filter Digital

Secara umum, filter digital dibedakan atas dua tipe yaitu *Finite Impulse Response* (FIR) dan *Infinite Impulse Response* (IIR). Sesuai dengan namanya, filter FIR memiliki *impulse response*, $h(n)$, dalam jumlah yang tertentu dan terbatas, sedangkan filter IIR memiliki $h(n)$ yang sangat banyak. Filter FIR bisa dinyatakan dalam bentuk *impulse response* atau dalam bentuk *transfer function*, sedangkan filter IIR hanya dinyatakan dalam bentuk *transfer function* karena memiliki $h(n)$ yang sangat banyak. Gambar 11.7 menunjukkan contoh filter FIR sedangkan Gambar 11.8 menunjukkan contoh filter IIR.



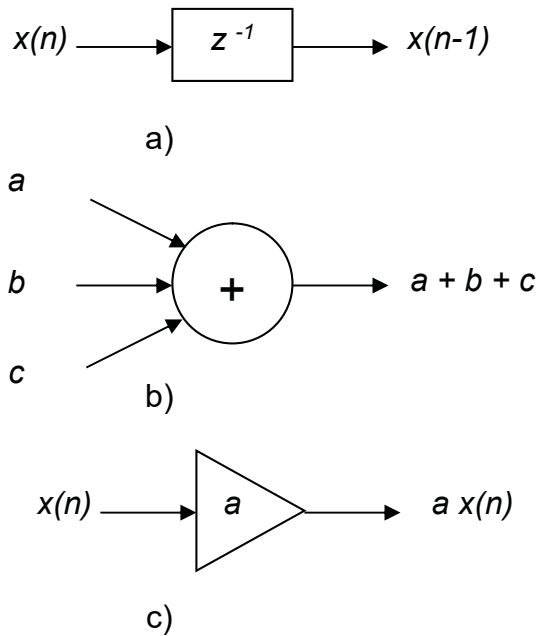
Gambar 11.7
Contoh filter FIR a) dengan impulse respons, b) dengan transfer function



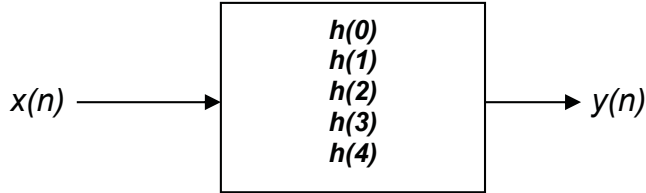
Gambar 11.8
Contoh filter IIR

11.6 Implementasi Filter FIR

Baik filter FIR maupun filter IIR dapat diimplementasikan dalam bentuk program komputer atau perangkat keras digital. Implementasi dalam bentuk program komputer sama dengan implementasi sistem diskrit (Bab 8). Implementasi dalam bentuk perangkat keras menggunakan tiga komponen utama yaitu *delay*, penjumlah, dan pengali. Komponen *delay* menunda input sebesar 1 hitungan. Komponen penjumlah menjumlahkan semua input, sedangkan komponen pengali mengeluarkan output yang merupakan hasil perkalian antara input dengan faktor pengalinya, seperti ditunjukkan pada Gambar 11.9.



Gambar 11.9
Komponen dasar filter digital

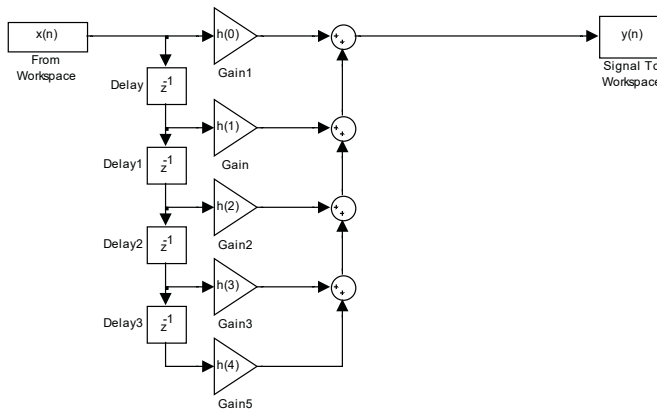


Gambar 11.10
Contoh Filter FIR orde 4

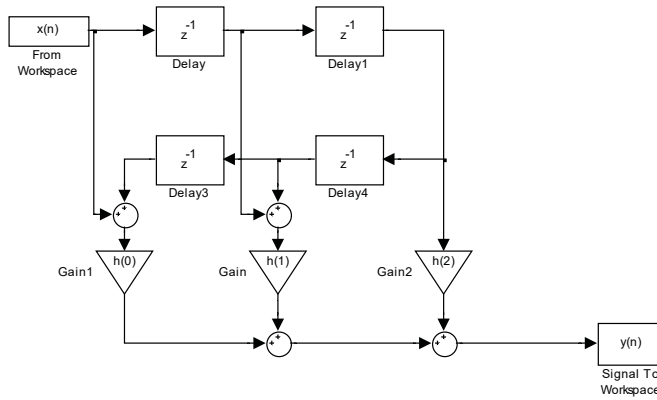
Sebagai contoh, sebuah filter FIR seperti pada Gambar 11.10, dapat dinyatakan dalam bentuk impulse response sehingga outputnya dihitung dengan konvolusi (lihat Bab 6) yaitu:

$$y(n) = h(0) x(n) + h(1) x(n-1) + h(2) x(n-2) + h(3) x(n-3) + \dots + h(N) x(n-N)$$

Filter tersebut dapat diimplementasikan dengan struktur transversal (Gambar 11.11) atau struktur fase linier (Gambar 11.12). Struktur fase linier lebih menghemat proses perhitungan karena filter FIR dengan fase linier memiliki $h(0)$ yang selalu sama dengan $h(N)$, $h(1) = h(N-1)$ dan seterusnya seperti terlihat pada filter di Gambar 11.7a.



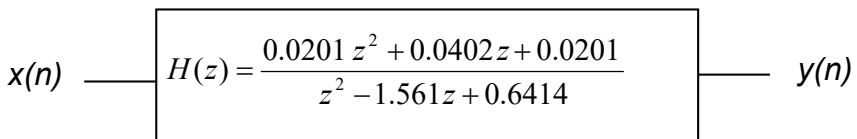
Gambar 11.11
Implementasi FIR dengan struktur transversal



Gambar 11.12
Implementasi FIR dengan struktur fase linier

11.7 Implementasi Filter IIR

Filter IIR juga diimplementasikan dengan menggunakan komponen yang sama seperti pada Gambar 11.9. Filter IIR dapat diimplementasikan dengan beberapa struktur. Struktur yang paling umum dipakai adalah Direct Form I dan Direct Form II.

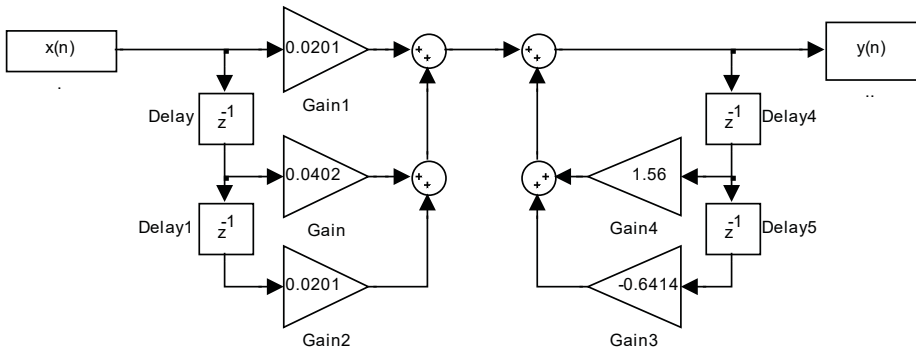


Gambar 11.13
Contoh Filter IIR orde 2

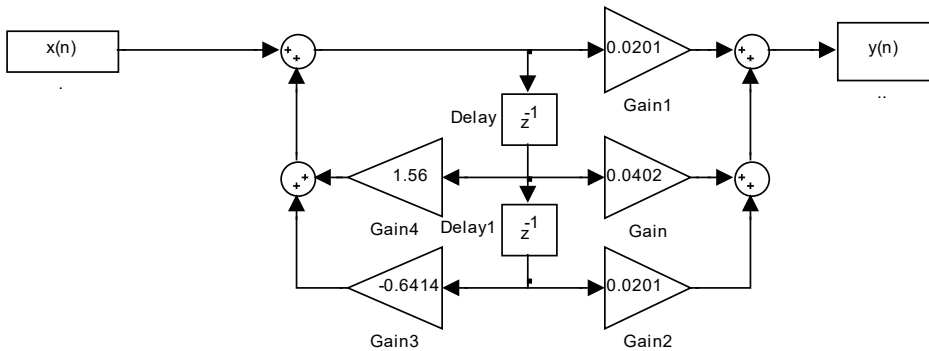
Output dari filter IIR seperti pada Gambar 11.13 dapat dihitung dengan persamaan diferensial:

$$y(n) = 0.0201x(n) + 0.0402x(n-1) + 0.0201x(n-2) + 1.56y(n-1) - 0.6414y(n-2)$$

Implementasi filter tersebut dengan struktur Direct Form I ditunjukkan pada Gambar 11.14, sedangkan implementasi dengan struktur Direct Form II ditunjukkan pada Gambar 11.15. Direct Form II lebih menghemat penggunaan proses *delay*.



Gambar 11.14
Implementasi IIR dengan struktur Direct Form I



Gambar 11.15
Implementasi IIR dengan struktur Direct Form II

11.8 Perbandingan antara FIR dan IIR

Ketika kita merancang filter digital, kita harus memilih tipe apa yang akan digunakan, FIR atau IIR. Kedua tipe ini memiliki kelebihan dan kekurangan. Berikut ini adalah perbandingan antara kedua filter tersebut.

- Filter FIR dapat memiliki respon fase yang benar-benar linier sehingga tidak merusak bentuk sinyal seperti dijelaskan pada Bab 10. Filter IIR tidak memiliki respon fase yang linier. Filter FIR digunakan untuk aplikasi yang sangat mementingkan bentuk sinyal misalnya pada transmisi data digital, biosignal (ECG, EEG, dsb), audio digital, dan pemrosesan citra digital. Jika kita mentransmisi data digital (gelombang kotak) melewati sebuah filter IIR maka outputnya tidak lagi berbentuk kotak. Distorsi ini tidak terjadi pada filter FIR karena memiliki respon fase yang linier.
- Filter FIR selalu stabil karena output hanya tergantung dari input, sedangkan output dari filter IIR tergantung dari input dan output sebelumnya sehingga bisa tidak stabil.
- IIR menggunakan proses umpanbalik (recursive) sehingga rawan terhadap pembulatan angka pada proses perhitungan output. Filter FIR tidak terlalu terpengaruh oleh pembulatan angka.
- Untuk mendapat *transition band* yang curam diperlukan filter FIR dengan orde yang sangat tinggi. Filter IIR tidak memerlukan orde yang tinggi untuk mendapat *transition band* yang curam.
- Filter IIR bisa mempunyai respon frekuensi yang persis sama dengan filter analog. Jika kita ingin mengganti filter analog pada suatu sistem dengan filter digital tanpa

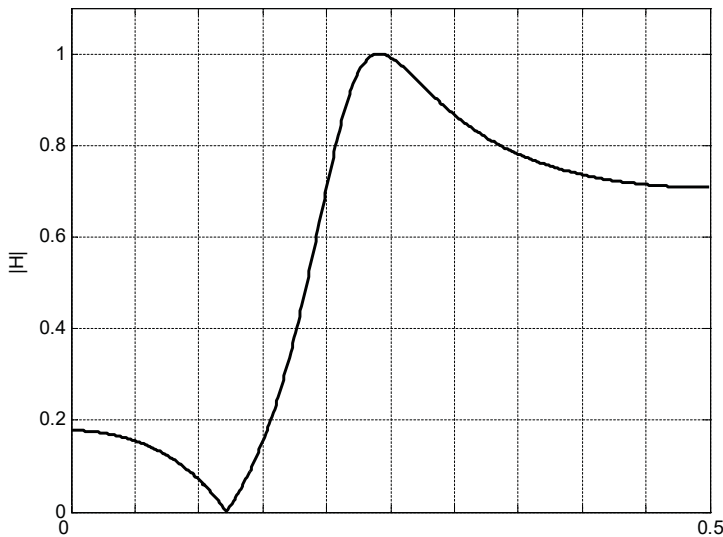
merubah respon frekuensinya maka kita harus memiliki filter IIR.

- Proses desain filter FIR sangat memerlukan bantuan komputer karena biasanya filter FIR ber-orde sangat tinggi sehingga sangat banyak angka yang harus dihitung. Filter IIR dapat didesain secara manual tanpa bantuan komputer.
- Karena memiliki orde yang sangat tinggi maka filter FIR mengalami *delay* yang sangat panjang dibandingkan dengan filter IIR

Gunakan filter IIR jika filter kita hanya mementingkan kecuraman transition band. Gunakan FIR jika kita harus menghindari distorsi fase dan jika kita memiliki komputer atau prosesor yang dapat menghitung filter dengan orde sangat tinggi.

SOAL LATIHAN

1. Gambarlah respon frekuensi dari suatu filter HPF digital dengan $\Omega_c = 0.1\pi$, $\delta_p = 2$ dB, dan $\delta_s = -10$ dB.
2. Gambarlah respon frekuensi dari suatu filter LPF digital (dalam bentuk frekuensi radian ternormalisasi) yang memiliki $f_c = 1$ kHz, $f_s = 5$ kHz, $\delta_p = 3$ dB, dan $\delta_s = -20$ dB.
3. Respon frekuensi dari suatu filter digital digambarkan dalam bentuk frekuensi ternormalisasi seperti di bawah ini. Tentukan jenis dan parameter filter tersebut dalam Hz dan dB.



4. Implementasikan filter dengan $h(n)$ berikut ini pada Simulink (dengan modifikasi file `soal10_4.mdl`) menggunakan struktur transversal. Gunakan filter tersebut untuk input s yang terdapat dalam file `sinyal10_4.mat`. Amati bentuk input dan outputnya tentukan jenis dan parameter dari filter tersebut.

n	$h(n)$
0	0.0068
1	0.0152
2	0.0395
3	0.0780
4	0.1212
5	0.1553
6	0.1682
7	0.1553
8	0.1212
9	0.0780
10	0.0395
11	0.0152
12	0.0068

- Ulangi soal nomor 4 di atas dengan menggunakan struktur fase linier.
- Implementasikan filter dengan $H(z)$ berikut ini pada Simulink dengan menggunakan struktur Direct Form I. Gunakan filter tersebut untuk input x yang terdapat dalam file `sinyal10_4.mat`. Amati bentuk input dan outputnya tentukan jenis dan parameter dari filter tersebut.

$$H(z) = \frac{0.0191z^3 - 0.0185z^2 - 0.0185z + 0.0191}{z^3 - 2.8665z^2 + 2.7534z - 0.8856}$$

- Ulangi soal nomor 6 dengan menggunakan struktur Direct Form II.
- Hitung dan gambarlah respon frekuensi dari filter pada nomor 4 dan 6 di atas. Gunakan Matlab.
- Gantilah input pada soal nomor 4 dengan suara dari mikrophone. Gunakan file Simulink `soal10_9.mdl`. Amati efek dari filter tersebut terhadap suara anda.

Catatan: Semua file `*.mdl` dan `*.mat` dapat diunduh di situs buku ini.

Perintah Matlab adalah:

```
>> F=[0 2/9 4/9 6/9 8/9 1];  
>> A=[1 1 1 0 0 0];  
>> N=8;  
>> B=fir2(N,F,A,boxcar(N+1))
```

F adalah frekuensi dari titik-titik sample, A adalah nilai dari titik-titik sample tersebut, dan N adalah orde dari filter yang dirancang. Output dari perintah tersebut adalah B sebagai berikut:

```
B =  
Columns 1 through 6  
0.0363 -0.0762 -0.0503 0.3072 0.5557 0.3072  
  
Columns 7 through 9  
-0.0503 -0.0762 0.0363
```

maka

```
h(0) = 0.0363  
h(1) = -0.0762  
h(2) = -0.0503  
h(3) = 0.3072  
h(4) = 0.5557  
h(5) = 0.3072  
h(6) = -0.0503  
h(7) = -0.0762  
h(8) = 0.0363
```

Nilai ini berbeda dengan hasil perhitungan manual di atas karena Matlab menuntut kita untuk harus menyelipkan satu titik pada $\Omega = 1$ yang setara dengan $f_s/2$.

Untuk memastikan respon frekuensi dari filter yang baru didesain, maka $h(n)$ tersebut ditransformasi ke domain frekuensi dengan FFT seperti pada Gambar 12.6.

SOAL LATIHAN

1. Rancanglah suatu filter HPF orde 8 dengan $\Omega_c = 4/9 \pi$ dengan metode sampling frekuensi.
2. Ulangi soal nomor 2 di atas dengan menggunakan Matlab.
3. Rancanglah suatu filter LPF orde 9 dengan $f_c = 2 \text{ KHz}$, $f_{stop} = 3 \text{ KHz}$ dan $f_s = 10 \text{ kHz}$ dengan metode sampling frekuensi.
4. Ulangi soal nomor 3 di atas dengan menggunakan Matlab.
5. Implementasikan filter tersebut dengan memodifikasi file `soal111_5.m`. Berikan beberapa sinyal sinusoidal dengan frekuensi 500 Hz , 1 kHz , 2 kHz , 4 kHz dan 4 kHz (gunakan $f_s = 10 \text{ kHz}$) sebagai inputnya. Amati apakah filter tersebut berfungsi sebagaimana mestinya.
6. Matlab memiliki fungsi 'filter' seperti terlihat pada file `soal111_6.m`. Ulangi soal nomor 5 di atas dengan menggunakan fungsi `filter`. Amati apakah filter tersebut berfungsi sebagaimana mestinya.
7. Implementasikan filter di atas dengan menggunakan Simulink untuk sinyal input seperti pada soal nomor 5.