Nama : Naufal Hilmi Fathul Ihsan
NIM : 5312422039
Matkul : Pengolahan Sinyal Digital

"Resume Video"

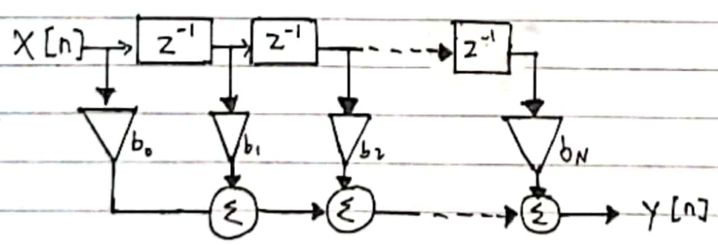
* Filters

Filter adalah sistem linear dan tidak bergantung pada waktu. Artinya filter memenuhi sifat-sifat berikut :
Jika $F(X(n))$ adalah fungsi filter dari sinyal input $x(n)$, maka kita memiliki :
Linearitas : untuk 2 sinyal $x_1(n)$ dan $x_2(n)$
Dengan a faktor a : $F(x_1(n) + x_2(n)) = F(x_1(n)) + F(x_2(n))$
$F(a \cdot x(n)) = a \cdot F(x(n))$
Artinya kita dapat "mengeluarkan" jumlah dan faktor dari fungsi kita.
Invariansi Waktu, jika $y(n) = F(X(n))$
maka kita punya, untuk penundaan n_0 : $y(n+n_0) = F(X(n+n_0))$
Yang berarti fungsi kita tetap sama kapanpun kita menerapkannya.

* FIR Filters. Video : "ADSP-07b Filters-02 Finite Impluse Respons (FIR) Filters"

Fiter FIR memiliki persamaan perbedaan seperti berikut :
$y(n) = \sum_{m=0}^{L} b(m)\, x(n-m)$
Dimana :
- $Y(n)$ adalah output filter pada waktu n
- $X(n)$ adalah input filter pada waktu n
- $b(m)$ adalah koefisien filter
- m adalah indeks penundaan

Implemetasi Filter FIR, diagram blok khas filter FIR :



Perhatikan bahwa disini blok dengan z^{-1} diimplementasikan dengan penundaan sebesar 1 interval sampling, bukan perkalian dengan z^{-1}, seperti yang kita lakukan pada z-domain. Setelah blok penundaan pertama z^{-1} kita memiliki $x(n-1)$, setelah blok penundaan kedua kita memiliki $x[n-2]$ dan seterusnya. Setiap blok penundaan "menghafal" nilai dari kiri untuk satu siklus clock sampel, dan melepaskannya ke kanan pada siklus clock sampel berikutnya.

Dipindai dengan CamScanner

Oleh karena itu, mereka menunda sampel sebanyak 1 siklus clock sampel.

Transformasi-Z dan Respons Frekuensi

Transformasi-Z adalah alat yang digunakan untuk menganalisis filter FIR dalam z-domain. Respons frekuensi filter FIR dapat diperoleh dengan mengganti z dengan $e^{j\omega}$ dalam persamaan transfer fungsi linear. Respon frekuensi menunjukkan bagaimana filter memperkuat atau melemahkan sinyal pada frekuensi yang berbeda.

Filter FIR bisa dirancang dengan memilih koefisien filter $b(m)$, metode untuk mendesain filter FIR seperti metode desain berdasarkan pernyataan frekuensi, berdasarkan respon impuls dan metode berdasarkan optimasi.

Filter FIR bisa diimplementasikan menggunakan perangkat keras/perangkat lunak. Implementasi perangkat keras lebih cepat dan efisien daripada perangkat lunak, tetapi lebih mahal dan kurang fleksibel. Implementasi perangkat lunak lebih fleksibel tapi lebih lambat dan kurang efisien.

Aplikasi Filter FIR yaitu : pemrosesan sinyal audio, pemrosesan gambar, telekomunikasi dan kontrol sistem.

* IIR Filters. video : "ADSP-076 Filters-03 Infinite Implus Respons (IIR) Filters."

Filter Infinite Impluse Respons (IIR) berbeda dengan filter Finite Impluse Response (FIR) yang dibahas sebelumnya. Perbedaan utama terletak pada persamaan diferensialnya, yaitu :

$$y(n) = \sum_{m=0}^{L} b(m) \cdot x(n-m) + \sum_{r=1}^{R} a(r) \cdot y(n-r)$$

Dimana:

- $y(n)$ = output dan $x(n)$ = input filter waktu n
- $b(m)$ adalah koefisien filter maju (feedforward)
- m yaitu indeks penundaan untuk koefisien filter maju
- $A(r)$ yaitu koefisien filter balik (feedback)
- r adalah indeks penundaan untuk koefisien filter balik

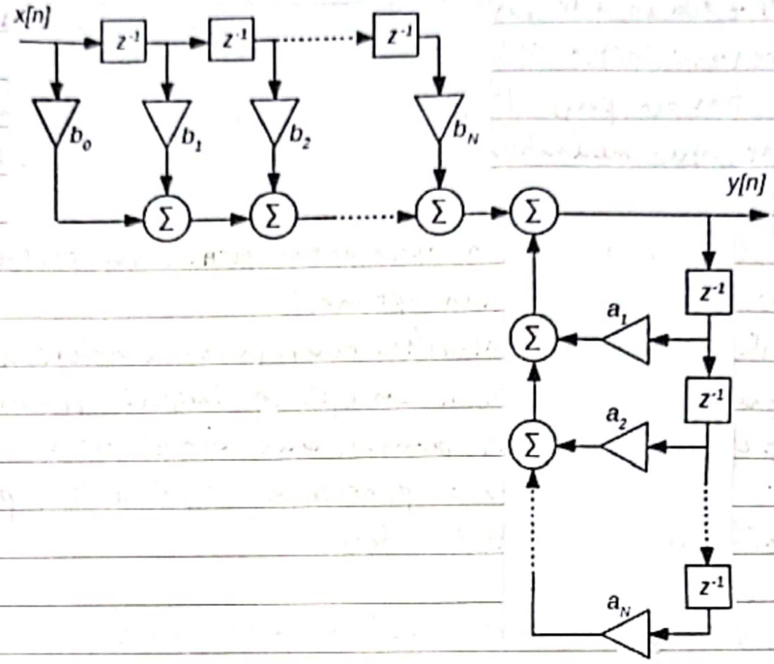
Persamaan ini menunjukkan bahwa output filter IIR bergantung pada input filter saat ini ($x(n)$) dan input filter sebelumnya ($x(n-m)$), dan output filter sebelumnya ($y(n-r)$). Koefisien filter $b(m)$ dan $a(r)$ menentukan input dan output filter dikombinasikan untuk menghasilkan output filter saat ini. Persamaan diferensial filter IIR memiliki konvolusi ganda dan umpan balik dari output y ke input dalam penjumlahan. Umpan balik ini memiliki penundaan $R=1$ untuk menghindari loop T-Leyla.

Persamaan diferensial filter IIR memiliki 2 komponen utama :

1. Konvolusi ganda : Filter IIR melibatkan konvolusi ganda antara input $X(n)$ dan koefisien filter maju $b(m)$ pada bagian pertama persamaan $\sum_{m=0}^{L}(b(m) \cdot x(n-m))$
2. Umpan balik : Bagian kedua persamaan $\sum_{r=1}^{R} a(r) \cdot y(n-r)$, mewakili umpan balik dari output filter $y(n)$ ke input.

Dipindai dengan CamScanner

Persamaan diferensial filter IIR dapat diimplementasikan menggunakan MATLAB, Octave, atau Scilab. Diagram blok filter IIR sebagai berikut:



Sekali lagi, disini kotak dengan z^{-1} melambangkan penundaan 1 periode sampling, dan segitiga melambangkan perkalian dengan faktor sebelumnya.
Kita dapat menyederhanakan struktur ini dengan menggabungkan penjumlahan, sebagai berikut:

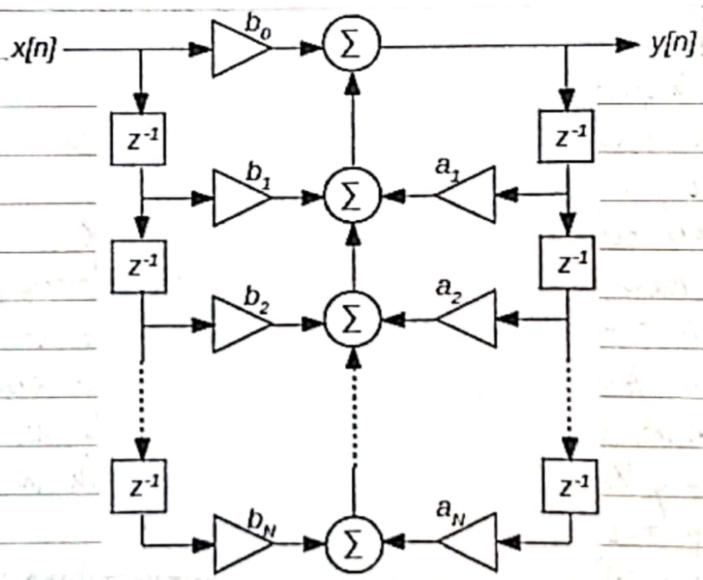


Diagram blok menunjukkan langkah-langkah berikut:

- Input : Sinyal input x(n) dimasukkan ke dalam sistem
- Blok penundaan : Sinyal input dan output disimpan ke blok penundaan untuk mempertimbangkan nilai sebelumnya.
- Perkalian : Koefisien b(m) dikalikan dengan nilai input dan output yang tertunda.

Dipindai dengan CamScanner

• Penjumlahan : hasil perkalian dijumlahkan dan digabungkan dengan input $x(n)$.
• output : hasil penjumlahan menjadi output filter $y(n)$
• umpan balik : Output $y(n)$ diumpankan kembali ke input dalam penjumlahan menciptakan loop umpan balik.

Transformasi Z dari persamaan sebelumnya (1) adalah :

$$Y(z) = \sum_{m=0}^{L} b(m) \cdot X(z) \cdot z^{-m} + \sum_{r=1} a(r) \cdot Y(z) \cdot z^{-r}$$

Transformasi-z digunakan untuk menganalisis filter IIR dalam domain-z. Transformasi ini mengubah sinyal waktu diskrit menjadi polinomial dalam variabel kompleks z.

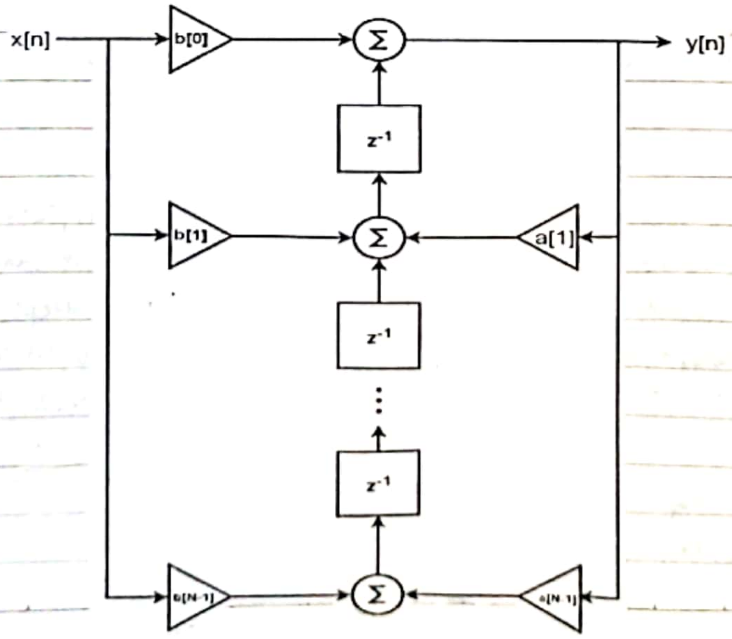
Fungsi transfer : $$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{L} b(m) \cdot z^{-m}}{1 - \sum_{r=1}^{R} a(r) \cdot z^{-r}}$$

Fungsi transfer filter IIR diperoleh dengan membagi transformasi-z output filter dengan transformasi-z input filter. Fungsi transfer ini mewakili respons filter terhadap sinyal input pada frekuensi yang berbeda. Fungsi transfer filter IIR memiliki polinomial dipenyebutnya, yang berarti kutub dan nol dapat menentukan stabilitas filter.

Stabilitas filter : Filter IIR dikatakan stabil jika semua kutubnya berada didalam lingkaran satuan dalam bidang kompleks z. Jika kutub berada diluar lingkaran satuan, filter akan berosilasi dan tidak stabil. Stabilitas filter IIR sangat penting untuk memastikan keluaran yang benar dan mencegah ledakan sinyal.

* Video : "Combined FIR-IIR Structure used in the Python "lFilter" Function :

Struktur gabungan FIR-IIR, yang merupakan kombinasi dari fiter FIR dan IIR. Karena penundaan adalah operator linear, dapat menggesernya setelah penjumlahan, dan karena dapat menggabungkan rantai penundaan untuk bagian FIR dan IIR. Hal ini mengurangi kebutuhan memori untuk implementasi, dan mengarah ke struktur berikut :



CS Dipindai dengan CamScanner

Implementasi struktur gabungan FIR-IIR dapat diimplementasikan dengan menggabungkan filter FIR dan IIR secara kasakade. Cara lain dengan menggunakan struktur paralel, dimana input dibagi menjadi dua jalur yang diproses oleh filter FIR dan IIR, lalu outputnya digabungkan. Struktur gabungan FIR dan IIR adalah alat yang ampuh untuk desain filter yang menawarkan fleksibilitas, efisiensi, dan kemampuan desain yang lebih besar dibandingkan filter FIR atau IIR saja.

* Filter Example. video : "ADSP-07b Filters-05 Filter Example : Exponential Decaying Signal"

Cara mengimplementasikan sinyal peluruhan eksponensial menggunakan sistem dengan kutub pada posisi p. Sinyal. Sinyal peluruhan eksponensial adalah sinyal yang nilainya berkurang secara eksponensial seiring waktu.

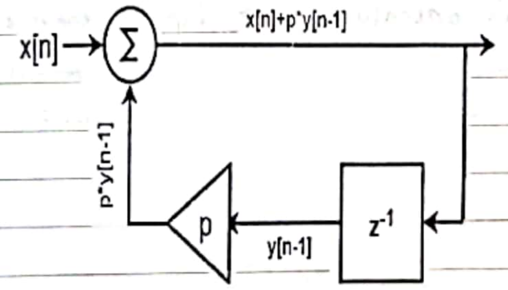
Langkah-langkah implementasi:

1.) Mendefinisikan sistem : Sistem memiliki kutub pada posisi P., Persamaan selisih untuk sistem diperoleh dengan menetapkan $b(0)=1$ dan $a(1)=p$.
Persamaan selisihnya adalah : $y(n) = 1 \cdot x(n) + p \cdot y(n-1)$

2.) Menafsirkan persamaan selisih : Persamaan selisih menunjukkan bahwa output sistem $y(n)$ adalah kombinasi linear dari input saat ini $x(n)$ dan output yang tertunda satu sampel $y(n-1)$.

3.) Menjelaskan respons impuls : Jika input adalah pulsa unit, output adalah urutan peluruhan eksponensial. Hal ini karena pulsa unit memicu respons sistem, dan respons tersebut kemudian meluruh secara eksponensial karena umpan balik negatif dalam sistem.

4.) Mempresentasikan sistem sebagai diagram blok, berikut ini :



- Diagram blok menunjukkan input, penundaan satu sampel, pengali, penjumlahan dan output.
- Penundaan satu sampel mewakili keterlambatan antara input saat ini dan output yang tertunda.
- Pengali memodifikasi input dan output yang tertunda dengan koefisien yang sesuai.
- Penjumlahan menggabungkan input dan output yang tertunda untuk menghasilkan output akhir.

5.) Menganalisis sistem di Z-domain, yaitu : $Y(z) = X(z) + p \cdot z^{-1} \cdot Y(z)$

Fungsi Transfer $\rightarrow$ $H(z) = \dfrac{Y(z)}{X(z)} = \dfrac{1}{1 - p \cdot z^{-1}}$

Transformasi Z diterapkan pada persamaan selisih untuk mendapatkan fungsi transfer sistem di Z-domain

Dipindai dengan CamScanner

6.) Menafsirkan fungsi transfer : Fungsi transfer menunjukkan hubungan antara input dan output dalam Z domain. Kutub sistem pada P terlihat dalam penyebut fungsi transfer.

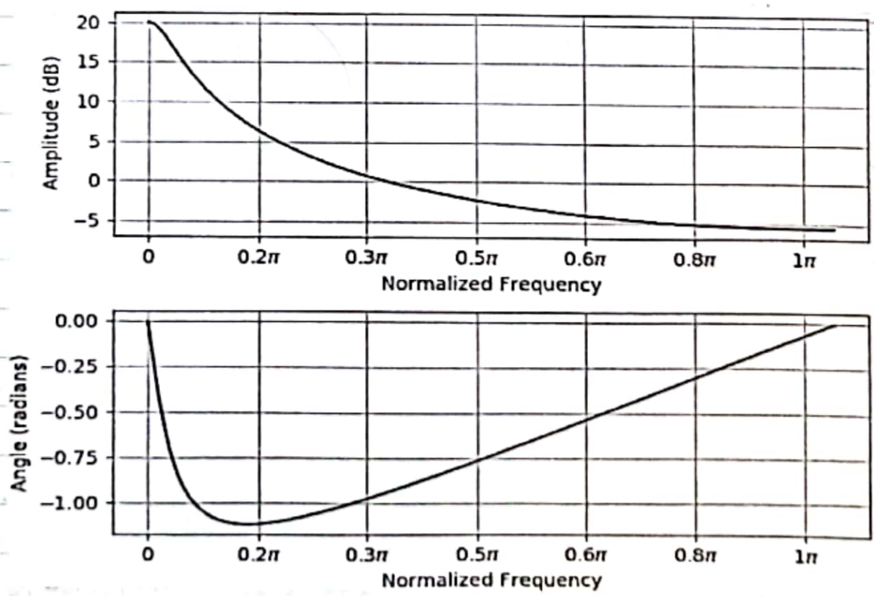
7.) Mengubah fungsi transfer ke domain waktu : Transformasi Z terbalik diterapkan pada fungsi transfer untuk mendapatkan respons impuls sistem di domain waktu. Respon impulsnya adalah urutan peluruhan eksponensial dengan konstanta peluruhan P. Hasil invers Z-Transform dari fungsi transfer kita adalah $1, p, p^2, p^3, \ldots$

* Video : "Computing the Resulting Frequency Response"

Dalam video ini membahas tentang cara menghitung respons frekuensi dari filter digital. Respons frekuensi adalah grafik yang menunjukkan bagaimana filter mempengaruhi frekuensi yang berbeda dari sinyal input, menunjukkan bagaimana filter bakal merubah sinyal dengan frekuensi yang berbeda. Contoh : Koefisien 'a' = (1, -0.9) dan koefisien 'b' = (1), ini berarti filternya tipe low-pass, yang melewati frekuensi rendah tapi memblok frekuensi tinggi. Angka 0.9 menunjukkan seberapa kuat filter meredam frekuensi tinggi.

Cara menghitungnya dengan fungsi yang namanya 'freqz', inputnya adalah koefisien filter 'a' dan 'b' yang disusun dalam bentuk vektor. Koefisien menentukan bagaimana filter bekerja. Fungsi 'freqz' akan mengeluarkan respons frekuensi yang bisa dilihat dalam bentuk grafik. Grafik ini menunjukkan bagaimana amplitudo sinyal berubah tergantung frekuensinya setelah melewati filter.

Berikut plot respons frekuensinya :



1.) Grafik respons Magnitudo (dB):

- Sumbu X : Frekuensi dinormalisasi (π), yaitu frekuensi tertinggi yang bisa ditangani filter itu adalah π radian/sampel (atau setengah dari frekuensi sampling).
- Sumbu Y : Magnitudo dalam desibel (dB), menunjukkan seberapa besar sinyal dengan frekuensi

Dipindai dengan CamScanner

tertentu dilemahkan atau dikuatkan dengan filter.

• Bentuk grafiknya menunjukkan penurunan magnitudo dari frekuensi rendah ke tinggi, ciri khas dari filter low-pass. Artinya filter ini melewati sinyal frekuensi rendah dengan baik, tapi meredam sinyal frekuensi tinggi

2.) Grafik Respons Fase (radian):

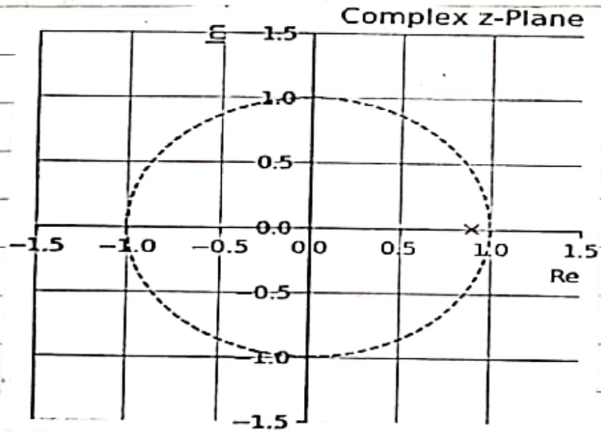
• Sumbu x : sama dengan grafik magnitudo, frekuensi dinormalisasi (π).

• Sumbu y : Fase dalam radian, menunjukkan seberapa besar pergeseran waktu sinyal dengan frekuensi tertentu setelah melewati filter.

• Bentuk grafiknya menunjukkan bagaimana fase sinyal berubah tergantung frekuensinya. Grafik ini memberikan gambaran lengkap tentang bagaimana filter digital akan merubah sinyal yang masuk, dari segi magnitudo sama fasenya.

Dapat juga menggunakan perintah "z plane" (dalam matlab) untuk memplot lokasi nol dan kutub pada bidang z yang kompleks. Pertama menghitung posisi tiang nol dengan fungsi python np.roots yaitu: $H(z) = \frac{1}{1 - p \cdot z^{-1}}$.

dan akar sebagai argumen koefisien terurut dari polinomial di z^{-1} (kompatibel dengan l filter). Posisi kutub adalah np.roots (a) dang angka nol pada np.roots (b).



Nol ditandai dengan "o" dan kutub ditandai dengan "x". Tiang lokasi $z = 0{,}9$.
Semakin dekat kutub ke lingkaran satuan, semakin besar puncak besaran respons frekuensi pada frekuensi yang dinormalisasi yang identik dengan sudut kutub ke titik asal.
Bisa dilihat menggunakan z transform ke DTFT dimana z diganti dengan $e^{j\Omega}$ (Ω = frekuensi dinormalisasi).
Jika melewati disepanjang lingkaran satuan, semakin dekat ke kutub, semakin tinggi magnitudo respons frekuensi. Semakin dekat ke nol, semakin kecil magnitudo respons frekuensi.
Pemfilteran, diberikan sinyal input dalam vektor x dan koefisien filter dalam vektor A dan B, output yang difilter y dari filter kita sederhana adalah:

$$y = lfilter(B, A, x)$$