Nama : Fakhri Ahmad Kurnia

NIM : 5312422016

Prodi : Teknik Komputer

Filter adalah sistem linier dan invarian waktu. Ini berarti mereka memenuhi sifat-sifat berikut:
jika $F(x(n))$ adalah fungsi filter sinyal masukan $x(n)$, maka kita punya

Linearitas : untuk 2 sinyal $x_1(n)$ dan $x_2(n)$

$F(x_1(n) + x_2(n)) = F(x_1(n)) + F(x_2(n))$

dengan faktor a : $F(a.x(n)) = a.F(x(n))$

yang berarti kita dapat mengeluarkan jumlah dan faktor dari fungsi kita

Time invariance, jika

$$y(n) = F(x(n))$$

kemudian kita punya, untuk penundaan n_0 : $y(n+n_0) = F(x(n+n_0))$
yang berarti fungsi kita tetap sama kapan pun kita menerapkannya

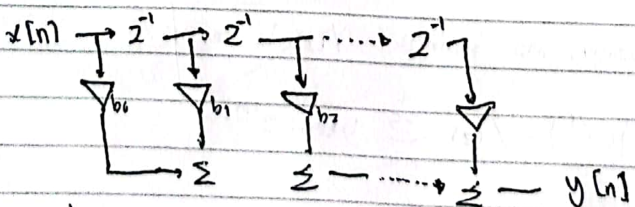
=> FIR Filter

Filter Finite Impulse Response (FIR) sederhana memiliki persamaan perbedaan seperti berikut, dengan $x(n)$ masukan dari filter dan $y(n)$ sebagai output :

$$y(n) = \sum_{m=0}^{L} b(m)\, x(n-m)$$

Perhatikan bahwa ini adalah konvolusi sinyal $x(n)$ dengan $b(n)$. Di sini, $b(m)$ adalah koefisien filter atau respon impulsnya. Ini juga bisa disebut 'taps', karena sistem ini dapat dipandang sebagai 'taps' pada garis tunda seperti terlihat pada gambar dibawah.

Persamaan perbedaan ini juga merupakan cara filter biasanya diimplementasikan di Matlab / Python dan bahasa pemrograman lainnya. Diagram bloknya :



Perhatikan bahwa disini blok dengan z^{-1} diimplementasikan dengan penundaan sebesar 1 interval pengambilan sampel, bukan perkalian z^{-1}, seperti yang dilakukan pada domain z.
Setelah blok penundaan pertama z^{-1}, kita mempunyai $x[n-1]$, setelah blok penundaan kedua kita mempunyai $x[n-2]$ dan seterusnya. Setiap blok penundaan menyimpan nilai dari kiri untuk satu siklus jam sampel dan melepaskannya ke kanan pada

Dipindai dengan CamScanner

siklus jam berikutnya.

Traporman 2 dari persamaan perbedaan konvolusi

$$y(n) = \sum_{m=0}^{L} b(m)\, x(n-m)$$

adalah (menggunakan linearitas 2 Transform) :

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

Sekarang kita bisa menghitung fungsi transfer yang didefinisikan sebagai keluaran dibagi masukan.

$$H(z) := \frac{Y(z)}{X(z)} = \sum_{m=0}^{L} b(m) \cdot z^{-m}$$

Sekarang kita bisa memperoleh respons frekuensi sehingga kita dapat melihat frekuensi mana yang dilemahkan dan mana yang tidak dari fungsi transfer filter dengan mengganti z, dengan $e^{j\Omega}$

$$H(e^{j\Omega}) = \sum_{m=0}^{L} b(m) \cdot e^{-j\Omega \cdot m}$$

Karena e bilangan kompleks, respons frekuensi H juga kompleks. Oleh karena itu bilangan kompleks untuk setiap frekuensi Ω.

→ IIR Filter

Persamaan dari bess sistem ini adalah :

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

Disini terdapat dua konvolusi. Ada umpan balik dari keluaran y kembali ke input dalam penjumlahan ini. dan feedback juga dimulai dengan delay r = 1. Ini untuk menghindari loop tanpa penundaan

Transformasi 2 persamaan dari bess adalah

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

Untuk mendapatkan fungsi alihnya, kita memindahkan Y(z) ke satu sisi

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

Maka fungsi yang ~~dihasilkan~~ dihasilkan adalah

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{m=0}^{L} b(m) \cdot z^{-m}$$

SIDU

CS Dipindai dengan CamScanner

Disini kita dapat melihat bahwa kita mendapatkan polinomial di penyebut fungsi alih dan ~~jika semua pole ini~~ karena muncul pole. Filter dapat menjadi tidak stabil, zero dan polinomial penyebut ini menjadi ~~fungsi~~ pole dari fungsi alih. Jika semua pole ini berada dlm satu lingkaran satuan, maka filter stabil.

> Combined FIR - IIR structure used in the python

Karena pen delay adalah operator linear, kita bisa menggesernya setelah penjumlahan, sehingga bisa menggabungkan rumus persamaan umum bagian FIR dan IIR.

Kita hanya perlu sistem dengan pole pada posisi p. Dalam persamaan sebelumnya kita mendapatkan dengan menetapkan $b(0) = 1$ dan $a(1) = p$. Maka kita mendapat persamaan kedua sederhana :

$$y(n) = 1 \cdot x(n) + p \cdot y(n-1)$$

Jika n adalah pulsa satuan, keluaran adalah urutan yang ~~menyebar~~ meluruh secara eksponensial :

$$1, p, p^2, p^3, \ldots$$

Dalam domain z ini adalah :

$$Y(z) = X(z) + p \cdot z^{-1} \cdot Y(z)$$

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

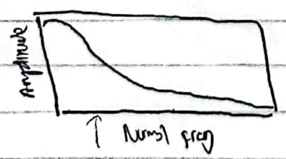
Dalam struktur ini, kita sekarang dapat melihat loop umpan balik. Ini sama dengan Transformasi z deret eksponensial sebelumnya. Ini berarti, ketika kita mentransformasikan ini kembali ke domain waktu, kita mendapatkan fungsi eksponensial yang merupakan respon impuls filter.

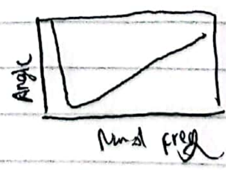
Hasil dari inverse z - transform : $1, p, p^2, p^3, \ldots$

Contoh :

Jika kita memilih $a(1) = p = 0{,}9$, maka didapatkan $A = [1, -0{,}9]$ dan $B = [1]$

Dengan menggunakan python yg didapatkan :





Perhatikan bahwa sumbu horizontal frekuensi ternormalisasi, sisi kanan adalah π, yang merupakan frekuensi Nyquist atau setengah dari frekuensi sampling.

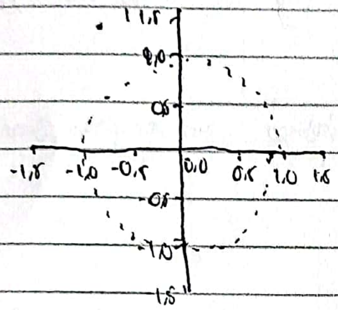
Respon frekuensi ini memiliki karakter low pass.

Dipindai dengan CamScanner

Kita memiliki

$$H(z) = \frac{1}{1 - p \cdot z^{-1}}$$

Dan np.roots mengharapkan argumen berupa koefisien yang terurut dari polinomial kita dalam z^{-1} (yang membuatnya kompatibel dengan lfilter) Oleh karena itu, posisi pole kita adalah hasil dari np.roots (a) dan zero pada np.roots (b)



Zero ditandai dengan "O" dan pole ditandai dengan x, disini kita melihat pole pada lokasi $z = 0{,}9$.
Secara umum, semakin dekat sebuah pole dengan lingkaran satuan, semakin besar puncak yang berasosiasi dalam magnitudo respon frekuensi pada frekuensi ter normalisasi yang identik dengan sudut pole terhadap titik asal.