BAB I
PENGANTAR PENGOLAHAN SINYAL

Tujuan:

1. Menjelaskan definisi sinyal
2. Menjelaskan klasifikasi sinyal
3. Menjelaskan aplikasi pengolahan sinyal

PENDAHULUAN

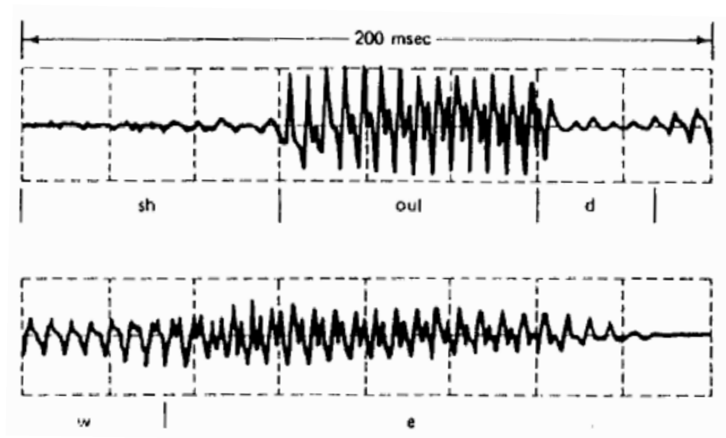
Sinyal memegang peranan penting dalam kehidupan modern, karena saat ini masyarakat tidak lepas dari telekomunikasi terutama handphone, yang mana piranti ini sarat dengan pengolahan sinyal. Tanpa disadari di alam, sinyal juga dapat ditemukan di sekitar manusia dalam bentuk sinyal elektromagnetik tubuh makhluk hidup.

Agar sinyal dapat bermanfaat sesuai kebutuhan manusia dengan efisien dan optimal, maka diperlukan pengolahan sinyal dengan menggunakan suatu sistem elektronika analog maupun yang digital.

Diambil dari berbagai sumber, pengertian sinyal sangat bermacam, antara lain :

- Fungsi satu variabel atau lebih yang menunjukkan informasi dalam fisik fenomena alam.
- System berupa arus data yang mengalir melalui jalur transmisi
- Suatu indikator yang digunakan sebagai alat komunikasi
- Suatu impuls atau fluktuasi besaran listrik seperti tegangan, arus, kuat medan listrik, yang mengkodekan informasi.
- Suatu impuls elektronik atau gelombang radio yang dikirim atau diterima

- Suatu kuantitas/besaran yang berubah-ubah. Seperti contoh di bawah, menggambarkan tegangan mikrofon sebagai fungsi waktu. Tegangan listrik yang dihasilkan oleh mikrofon sebagai respon terhadap ucapan 'should' dan 'we'. Tegangan tersebut bersesuaian dengan tekanan akustik pada telinga, yang merupakan reaksi terhadap perubahan tekanan, seperti Gambar di bawah.



Dalam proses pengolahan sinyal analog, sinyal input masuk ke Analog Signal Processing (ASP), diberi berbagai perlakukan (misalnya pemfilteran, penguatan,dsb.) dan outputnya berupa sinyal analog.

Proses pengolahan sinyal secara digital memiliki bentuk sedikit berbeda. Komponen utama system ini berupa sebuah processor digital yang mampu bekerja apabila inputnya berupa sinyal digital. Untuk sebuah input berupa sinyal analog perlu proses awal yang bernama digitalisasi melalui perangkat yang bernama *analog-to-digital conversion* (ADC), dimana sinyal analog harus melalui proses sampling, quantizing dan coding. Demikian juga output dari processor digital harus melalui perangkat *digital-to-analog conversion* (DAC) agar outputnya kembali menjadi bentuk analog. Ini bisa kita amati pada perangkat seperti PC, digital sound system, dsb.

KLASIFIKASI SINYAL

Klasifikasi sinyal dapat dibeda-bedakan menurut:

- Sinyal Analog dan sinyal Digital

Sinyal analog adalah sinyal yang mempunyai nilai untuk setiap waktu, sinyal ini bersifat kontinyu terhadap waktu.

Sinyal digital adalah sinyal yang tidak untuk setiap waktu terdefinisi, sinyal ini bersifat diskrit terhadap waktu. Sinyal digital berasal dari sinyal analog yang disampling, yang artinya mengambil nilai suatu sinyal analog mulai t=0, t=Δt, t=2Δt, t=3Δt dan seterusnya. Untuk mendapatkan sinyal waktu diskrit yang mampu mewakili sifat sinyal aslinya, proses sampling harus memenuhi syarat Nyquist:

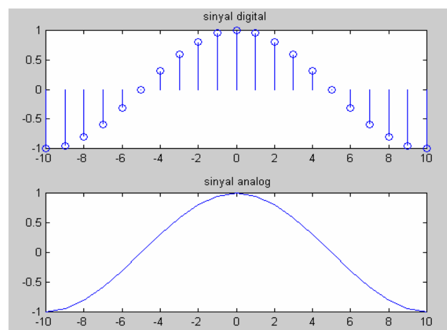
$$f_s > 2\, f_i$$

dimana f_s = frekuensi sinyal sampling

f_i = frekuensi sinyal informasi yang akan disampel

Dalam MATLAB (m-file)

```
n=-10:10;
y=cos(pi*n/10);
subplot(2,1,1);
stem (n,y);
subplot(2,1,2);
plot (n,y);
```



- Sinyal Riil dan sinyal kompleks

  Sinyal riil merupakan sinyal yang bersifat riil untuk semua variabel.

  Sinyal kompleks merupakan sinyal yang mempunyai nilai yang kompleks ada faktor nilai imajiner.

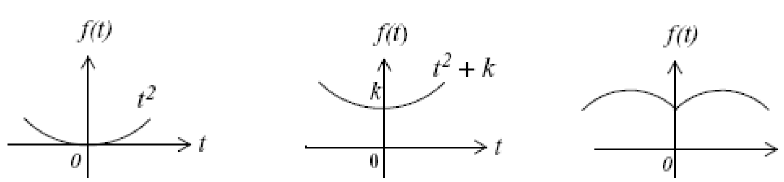
- Sinyal Ganjil dan Genap

  Sinyal genap mempunyai sifat:

  - $f(-t) = f(t)$
  - Polinomial dengan pangkat yang genap

    $$\cos t = 1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \frac{t^6}{6!} + \ldots$$

  - Contoh:

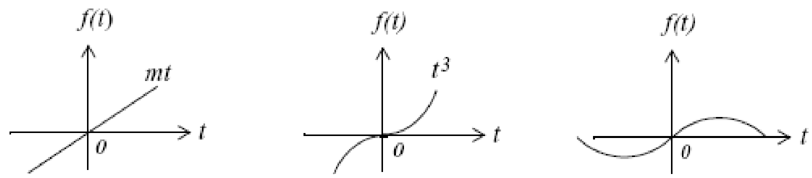


Sinyal ganjil mempunyai sifat:

- $-f(-t) = f(t)$
- Polinomial dengan pangkat yang ganjil

$$\sin t = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \dots$$

- Contoh:

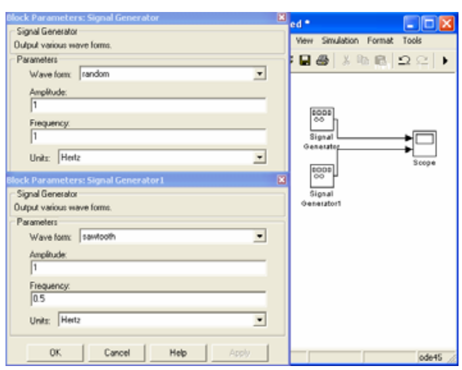


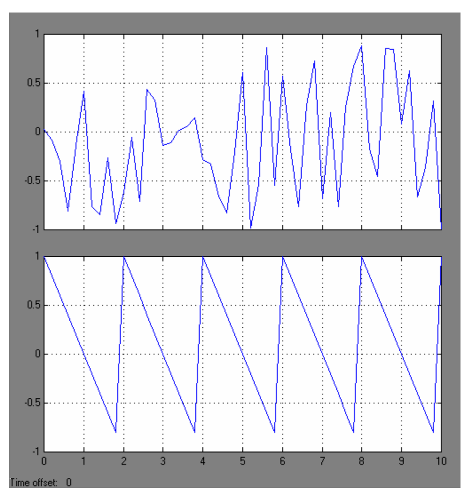
- Sinyal Deterministik dan sinyal Random

  Sinyal deterministik merupakan sinyal yang nilainya secara lengkap untuk semua titik waktu sudah dikenal.

  Sinyal random mempunyai nilai random untuk waktu yang diberikan. Nilai-nilai sinyal random untuk setiap titiknya tidak diketahui dengan pasti, sehingga sinyal random hanya dibahas berdasarkan karakter statistik, misalnya nilai rata-rata dan nilai tengah.

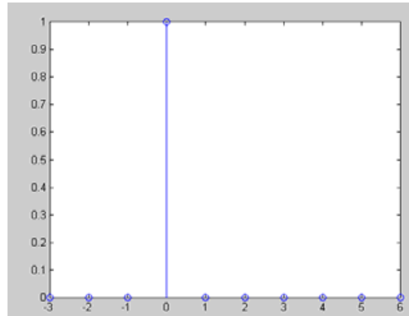
Dalam Matlab (Simulink)





Bentuk – bentuk sinyal:

- Impuls



$$x(t) = \frac{du(t)}{dt} \equiv \delta(t)$$

$$\int_{-\infty}^{+\infty} \delta(x)dx = 1$$

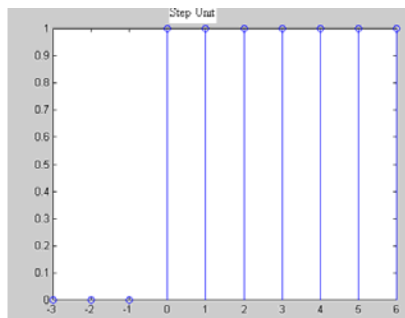
$$\delta(x) = \begin{cases} \infty & \text{untuk } x = 0 \\ 0 & \text{untuk } x \neq 0 \end{cases}$$

$$\int_{-\infty}^{+\infty} \delta(x) f(x) dx = f(0)$$

Dimana f(x) segala bentuk sinyal.

- Step (tangga)

Waktu diskrit:

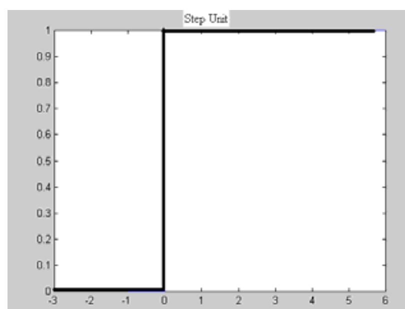


$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0' \end{cases}$$

Waktu kontinu

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0' \end{cases}$$

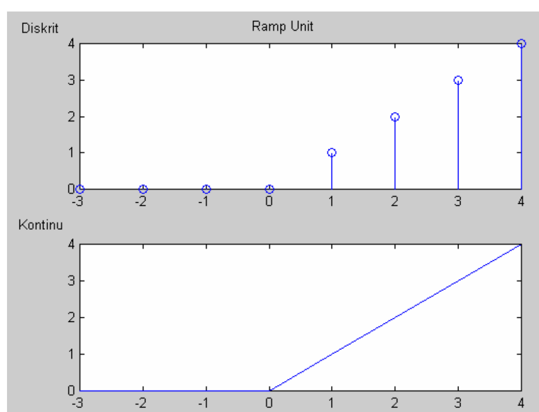
m-file (Matlab):




```
(diskrit)  n=-3:6;
           u=[zeros(1,3),ones(1,7)];
           subplot(2,1,1);
           stem(n,u);
(kontinu)
           n=[-3 -2 0 0.0001 1 2 3 4 5 6];
           u=[zeros(1,3),ones(1,7)];
           subplot(2,1,1);
           plot(n,u);
```

- Ramp

Waktu diskrit



$$r[n] = \begin{cases} n, & n \geq 0 \\ 0, & n < 0' \end{cases}$$

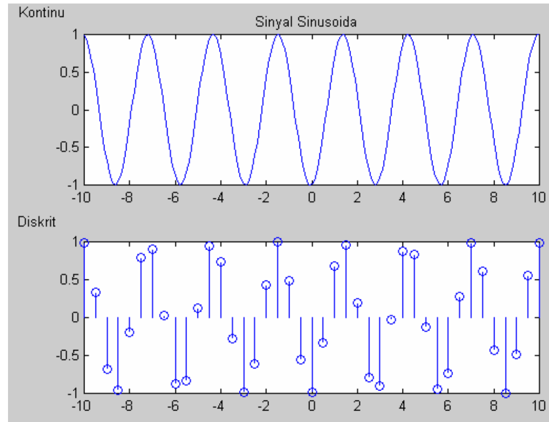
Waktu Kontinu

$$r(t) = \begin{cases} t, & t \geq 0 \\ 0, & t < 0' \end{cases}$$

- Sinusoida

  Waktu Kontinu

  $x(t) = \cos(\omega\ t + \alpha)$



  Waktu diskrit

  $x[n] = A\cos(\Omega.n + \alpha)$

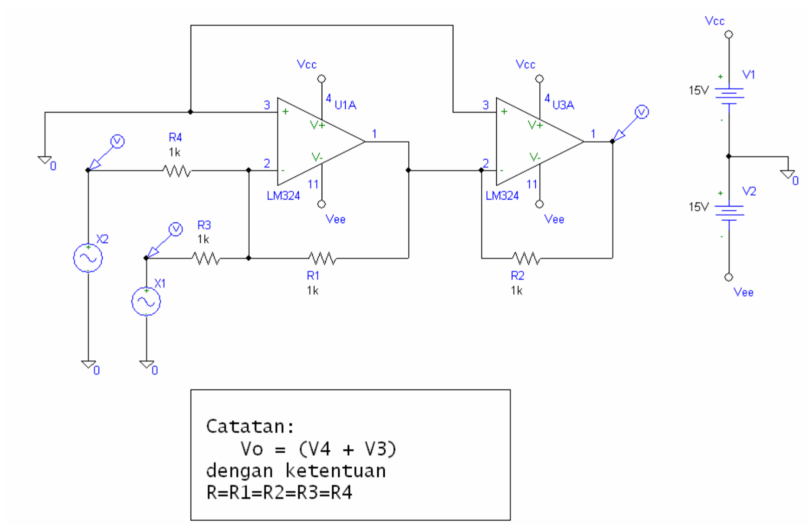
Operasi dasar pada sinyal:

- Penambahan sinyal

  Waktu kontinu

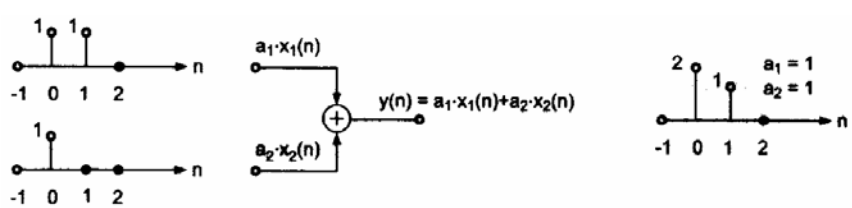
  $x(t) = x_1(t) + x_2(t)$

  Dalam aplikasi rangkaian analog dengan menggunakan op amp, simulasi dengan menggunakan PSPICE / EWB



  Waktu diskrit

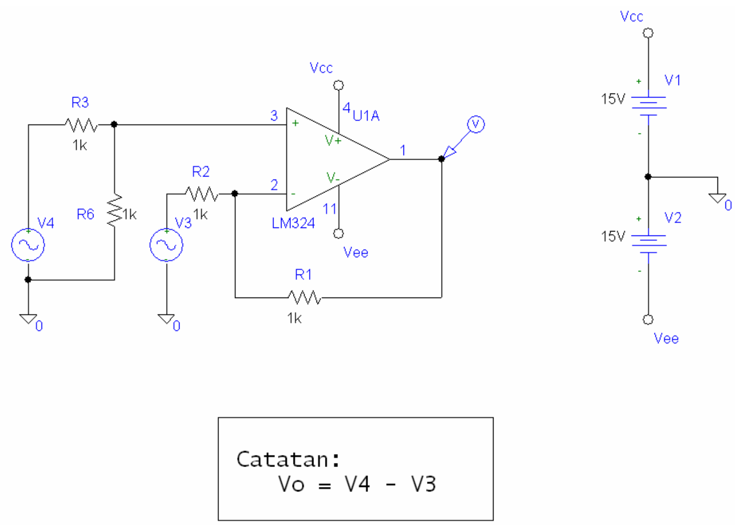
  $x[n] = x_1[n] + x_2[n]$



- Pengurangan sinyal

  $x(t) = x_1(t) - x_2(t)$

  Dalam aplikasi rangkaian analog dengan menggunakan op amp

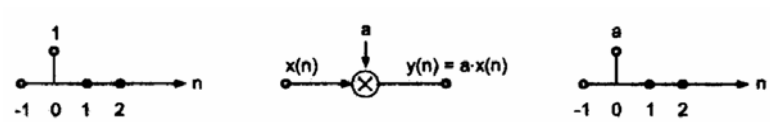


  Waktu diskrit

  $x[n] = x_1[n] - x_2[n]$

- Perkalian Sinyal

Waktu kontinu	Waktu diskrit
$x(t) = x_1(t).x_2(t)$	$x[n] = x_1[n].x_2[n]$



  Dalam aplikasinya, perkalian sinyal juga termasuk bagian dari modulasi

- Penskalaan Waktu

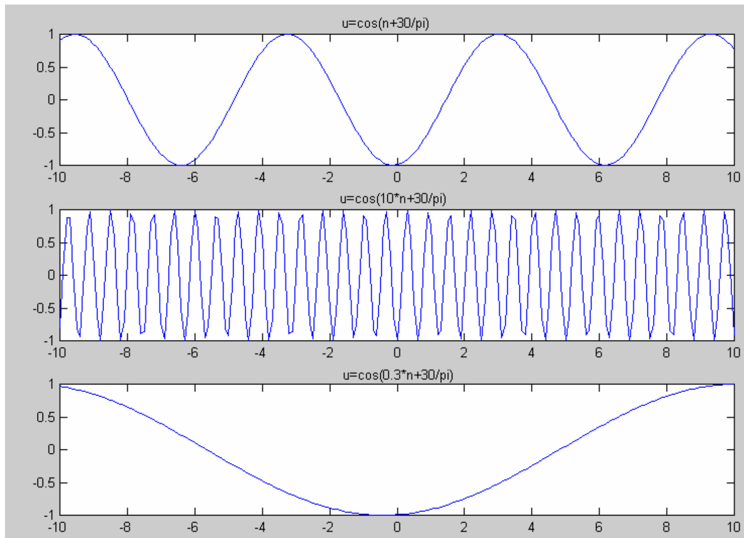
  Pada persamaan sinyal sinusoida/logaritma ada istilah penskalaan waktu, dimana nilai t bisa dikalikan dengan bilangan riil. Penskalaan waktu berlaku pada waktu kontinu dan diskrit.

  Contoh:

  m-file (MATLAB)

```
n=-10:0.1:10;            ➔ horisontal antara -10 s/d 10 dengan selisih 0.1
u=cos(n+30/pi);
subplot(3,1,1);          ➔ plot dibagi 3 , kolom 1, baris 1
plot(n,u);
u=cos(10*n+30/pi);       ➔ skala waktu dikali 10
subplot(3,1,2);          ➔ plot dibagi 3 , kolom 1, baris 2
```

```
plot(n,u);
u=cos(0.3*n+30/pi);     ➔ skala waktu dikali 0.3
subplot(3,1,3);         ➔ plot dibagi 3 , kolom 1, baris 3
plot(n,u);
```

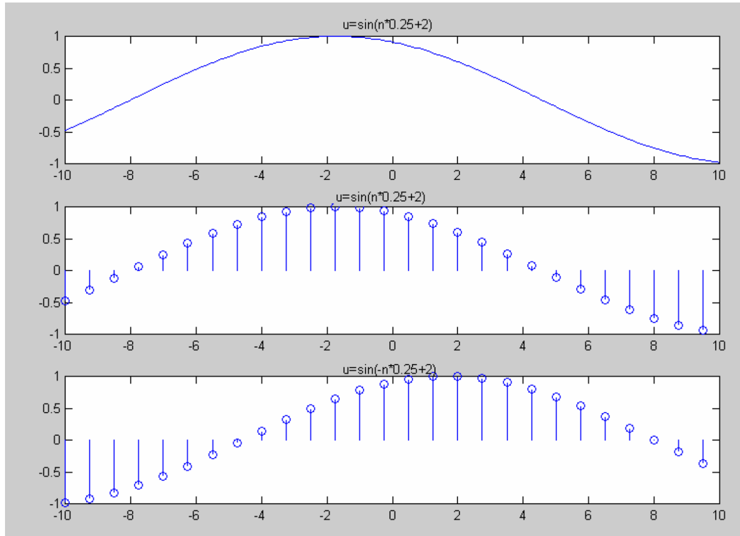


- Refleksi atau pencerminan

  Refleksi atau pencerminan umumnya digunakan saat pengolahan sinyal konvolusi diskrit. Pencerminan terhadap sumbu y.

  Contoh:

```
n=-10:0.1:10;
u=sin(n*0.25+2);
subplot(3,1,1);
plot(n,u)               ➔ bentuk kontinu
n=-10:0.75:10;
u=sin(n*0.25+2);        ➔ sebelum dicerminkan (n)
subplot(3,1,2);
stem(n,u)               ➔ bentuk diskrit
n=-10:0.75:10;
u=sin(-n*0.25+2);       ➔ setelah dicerminkan (–n)
subplot(3,1,3);
stem(n,u)               ➔ bentuk diskrit
```



- Delay atau digeser

  Delay pada :

  - sinyal diskrit : bergeser sebesar berapa kali sampling (ke kanan dan ke kiri) ➔ besarnya berupa bilangan interger
  - sinyal analog : bergeser sebesar sudut fasa ➔ besarnya derajat sudut fasa bisa berupa bilangan riil/integer

  Contoh:

  Diketahui sinyal diskrit :

  x[n] = 1 untuk n → -7 s/d -1

  x[n] = 0 untuk n → -7 > n dan n > -1

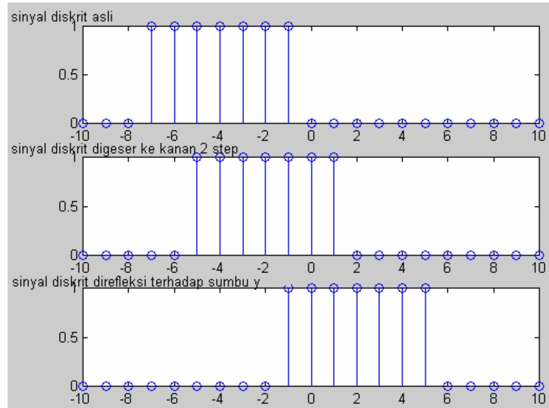
  Proses sinyal tersebut jika diinginkan:

  a. y[n] = x[n-2] ➔ sinyal digeser ke kanan 2 kali

  b. y[n] = x[-[n-2]] = x[-n+2] ➔ dicerminkan terhadap sumbu y.

  Jawab:

```
n=-10:10;
u=[zeros(1,3),ones(1,7),zeros(1,11)];
subplot(3,1,1);
stem(n,u);
n=-10:10;
u=[zeros(1,5),ones(1,7),zeros(1,9)];
subplot(3,1,2);
stem(n,u);
n=-10:10;
u=[zeros(1,9),ones(1,7),zeros(1,5)];
subplot(3,1,3);
stem(n,u);
```



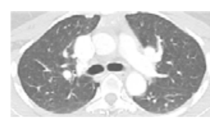
APLIKASI

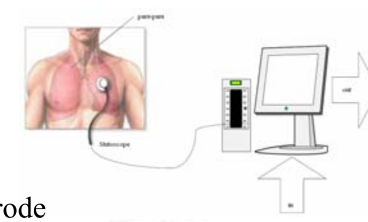
Dalam kehidupan modern, pengolahan sinyal sangat bermanfaat dalam segala bidang, antara lain:

- Bidang telekomunikasi

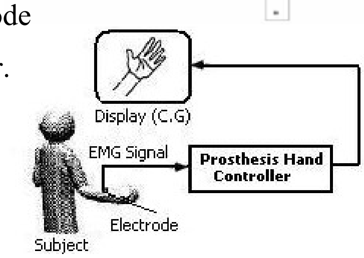
  Contoh:
  - Pengiriman sinyal handphone menuju BTS
  - Sistem telekomunikasi berbasis listrik PLN
- Bidang kedokteran

  Contoh:
  - Pengolahan sinyal dari sensor menjadi suatu gambar / image untuk mengidentifikasi apakah ada penyakit pada otak / paru-paru





  - Pengolahan sinyal dari sensor elektrode (EMG) diubah menjadi suatu gambar.



- Bidang Militer

  Contoh:
  - Pengiriman sinyal radar pesawat
  - Sistem kendali NASA

TUGAS

1. Jelaskan yang dimaksud dengan sinyal
2. Jelaskan perbedaan sinyal analog dan sinyal digital
3. Diketahui sinyal diskrit :

    x[n] = 3 untuk n → -3 s/d 0

    x[n] = 0 untuk n → -3>n>0

    Proses sinyal tersebut dengan menggunakan MATLAB jika diinginkan

    y[n] = x[n+3] ➔ sinyal digeser ke kiri 2 kali

    y[n] = x[-n-3] ➔ sinyal dicerminkan

4. Beri 2 contoh aplikasi pengolahan sinyal pada bidang komunikasi data komputer

BAB II
SISTEM LINIER DAN INVARIAN TERHADAP WAKTU

Tujuan:

1. Menjelaskan sistem linier
2. Menjelaskan sistem invarian terhadap waktu
3. Menjelaskan konvolusi

PENDAHULUAN

Sistem merupakan alat/algoritma yang memproses sinyal masukan untuk menghasilkan sinyal keluaran.

Sistem dikatakan linier apabila memenuhi syarat:

- Dapat dijumlah (*additivity*)

  Jika mempunyai sebuah sinyal masukan berbentuk $x(t) = x_1(t) + x_2(t)$,

  maka sinyal keluarannya adalah

  $$y(t) = T[x_1(t) + x_2(t)] = T[x_1(t)] + T[x_2(t)] = y_1(t) + y_2(t)$$

  $y_1(t)$ dan $y_2(t)$ adalah sinyal keluaran masing-masing dari $x_1(t)$ dan $x_2(t)$.

- Homogen

  Sinyal masukan:

  $$x(t) = a.x(t)$$

  Sinyal keluaran:

  $$y(t) = T[a.x_1(t)] = a.T[x_1(t)] = a.y_1(t)$$

  Jika sistem dikatakan linier dengan masukan

  $$x(t) = a.x_1(t) + b.x_2(t)$$

  Sistem tersebut akan bereaksi dengan mengeluarkan sinyal respon

$y(t) = T[a.x_1(t) + b.x_2(t)]$

$y(t) = T[a.x_1(t)] + T[b.x_2(t)]$ → dengan sifat additif

$y(t) = a.T[x_1(t)] + b.T[x_2(t)]$ → dengan sifat homogen

maka

$y(t) = a.y_1(t) + b.y_2(t)$

Sistem Invarian terhadap Waktu

Sistem ini mempunyai sifat, jika

$y(t) = T[x(t)]$,

maka untuk parameter bebas t_0

$y(t\text{-}t_0) = T[x(t\text{-}t_0)]$

Sebuah sistem yang invarian terhadap waktu, akan bereaksi sama jika dibeikan masukan yang sama, tidak tergantung kapan sinyal masukan itu diberikan (t_0 bebas).

Contoh:

Ada suatu persamaan $y(t) = a.x(t) + b$

Dimana $x(t)$ = sinyal masukan

$y(t)$ = sinyal keluaran

$a = b$ = konstanta

Tentukan apakah persamaan di atas linier atau tidak linier?

Jawab:

- Untuk menentukan linier atau tidak dicoba additif / tidak:

  Dengan 2 sinyal masukan $x_1(t)$ dan $x_2(t)$.

  Jika 2 sinyal masukan diatas dimasukkan secara bergantian, didapat

  $y_1(t) = T[x_1(t)] = a.x_1(t) + b$

  $y_2(t) = T[x_2(t)] = a.x_2(t) + b$

  Jumlah keduanya menjadi

  $y(t) = y_1(t) + y_2(t) = (a.x_1(t) + a.x_2(t)) + 2.b$ (1)

  jika sinyal diatas dimasukkan secara bersamaan dengan

  $x(t) = x_1(t) + x_2(t)$

  Maka

  $y(t) = T[x(t)] = a.(x_1(t) + x_2(t)) + b$ (2)

dapat dilihat persamaan (1) dan (2) berbeda, sehingga dikatakan sistem ini tidak additif.

- Homogen atau tidak

  Dengan sinyal masukan $x_1(t) = 2.x(t)$, didapat sinyal keluaran

  $y_1(t) = T[2.x(t)] = 2.ax(t) + b \neq y_1(t) = 2y(t) = 2.ax(t) + 2b$

  sistem ini tidak homogen

Dari uji additif dan homogen, diambil kesimpulan sistem $y(t) = a.x(t) + b$ bukan sistem linier.

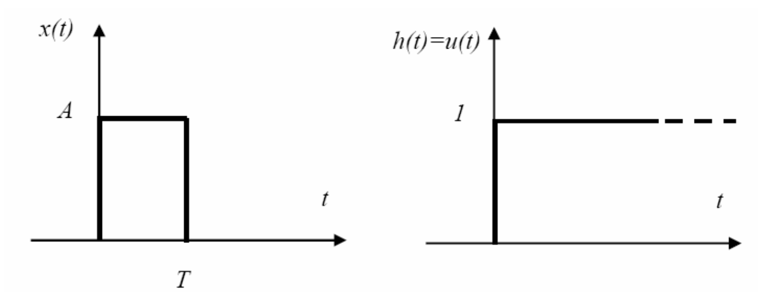
KONVOLUSI

Sinyal kontinu

Untuk menentukan hasil dari suatu sinyal masukan ke sistem dapat menggunakan teknik konvolusi.

$$y(t) = \int_{-\infty}^{+\infty} h(t-\tau)x(\tau)d\tau = h(t) * x(t)$$

Contoh:



Sinyal masukan x(t) adalah sinyal segiempat, dengan lebar T dan amplitudo A. Respon impuls dari sistemnya adalah sinyal step/tangga.

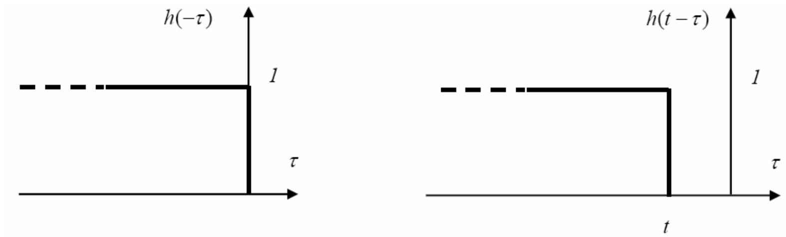
Tentukan keluaranya.

Jawab:

Dengan rumus

$$y(t) = \int_{-\infty}^{+\infty} h(t-\tau)x(\tau)d\tau = h(t) * x(t)$$

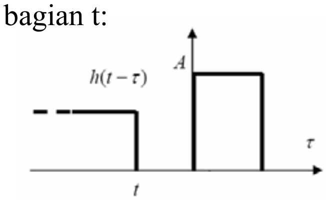
$h(-\tau)$ didapat dengan mencerminkan fungsi $h(\tau)$ terhadap sumbu vertikal. $h(t-\tau)$ dihasilkan dari $h(-\tau)$ dengan menggeser / refleksi (bab 1) sejauh t. Jika t bernilai negatif, maka fungsi bergeser ke kiri. Jika t bernilai positif, maka fungsi bergeser ke kanan.



Untuk menghitung integral konvolusi, dibagi beberapa bagian t:

- $-\infty < t \le 0$

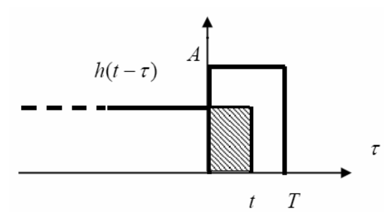
  Karena tidak ada bagian dari x atau h yang saling tumpang tindih



- $0 \le t \le T$

  $h(t-\tau).x(\tau) = A$ untuk $0 \le \tau \le t$ (bidang yang diarsir), selain itu bernilai nol.

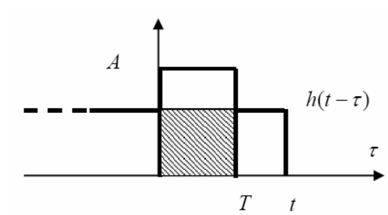
  $$y(t) = \int_{-\infty}^{+\infty} h(t-\tau)x(\tau)d\tau = \int_0^t A d\tau = A\int_0^t d\tau = At$$



- $T \le t$

  $h(t-\tau).x(\tau) = A$ untuk $0 \le \tau \le t$ (bidang yang diarsir), selain itu bernilai nol.

  $$y(t) = \int_{-\infty}^{+\infty} h(t-\tau)x(\tau)d\tau = \int_0^T A d\tau = A\int_0^T d\tau = AT$$



Sehingga keseluruhan nilai y(t):

$$y(t) = \begin{cases} 0 & \text{untuk } t \le 0 \\ A\,t & \text{untuk } 0 \le t \le T \\ A\,T & \text{untuk } T \le t \end{cases}$$

<u>Sinyal Diskrit</u>

Konvolusi antara dua sinyal diskrit x[n] dan h[n] dapat dinyatakan sebagai

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

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

$$x[n] * h[n] = \begin{cases} 0 & , n = -1, -2, -3.... \\ \sum_{i=0}^{n} x[i].h[n-i] & , n = 0,1,2,... \end{cases}$$

Langkah-langkah penyelesaian hampir sama dengan konvolusi pada sinyal kontinu.

- merubah *discretetime index* n sampai dengan i dalam sinyal x[n] dan h[n].
- Sinyal yang dihasilkan x[i] dan h[i] menjadi sebuah fungsi *discrete-time index* i.
- Menentukan h[n-i] dan membentuk pencerminan terhadap sinyal h[i]. Lebih tepatnya h[-i] merupakan pencerminan dari h[i] yang diorientasikan pada sumbu vertikal (axis), dan h[n-i] merupakan h[-i] yang digeser ke kanan dengan step n.
- Saat pertama kali *product* (hasil kali) x[i].h[n-i] terbentuk, nilai pada konvolusi x[n]*h[n] pada titik n dihitung dengan menjumlahkan nilai x[i].h[n-i] sesuai rentang i pada sederetan nilai integer tertentu.

Contoh:

Ada system diskrit sbb:

Sinyal pertama: x[i]= 1 2 3

Sinyal kedua: h[i]= 2 1 3

Cari konvolusi 2 sinyal tersebut secara manual dan dengan MATLAB.

Jawab:

Langkah 1: pembalikan sinyal kedua, h[n] sehingga didapatan kondisi :

Sinyal pertama: x[i] = 1 2 3

Sinyal kedua: h[-i] = 3 1 2

Langkah 2: pergeseran dan penjumlahan

Sinyal pertama: 1 2 3

Sinyal kedua: 3 1 2

------------------ x

product and sum: 0 0 2 0 0 = 2

Langkah 3: pergeseran satu step dan penjumlahan

Sinyal pertama: 1 2 3

Sinyal kedua: 3 1 2

--------------------- x

product and sum: 0 1 4 0 = 5

langkah 4: pergeseran satu step dan penjumlahan

Sinyal pertama: 1 2 3

Sinyal kedua: 3 1 2

------------------- x

product and sum: 3 2 6 = 11

Langkah 5: pergeseran satu step dan penjumlahan

Sinyal pertama: 1 2 3

Sinyal kedua: 3 1 2

------------------- x

product and sum: 0 6 3 0 = 9

Langkah 6: pergeseran satu step dan penjumlahan

Sinyal pertama: 1 2 3

Sinyal kedua: 3 1 2

------------------- x

product and sum: 0 0 9 0 0 = 9

Langkah 7: pergeseran satu step dan penjumlahan

Sinyal pertama: 1 2 3

Sinyal kedua: 3 1 2

------------------- x

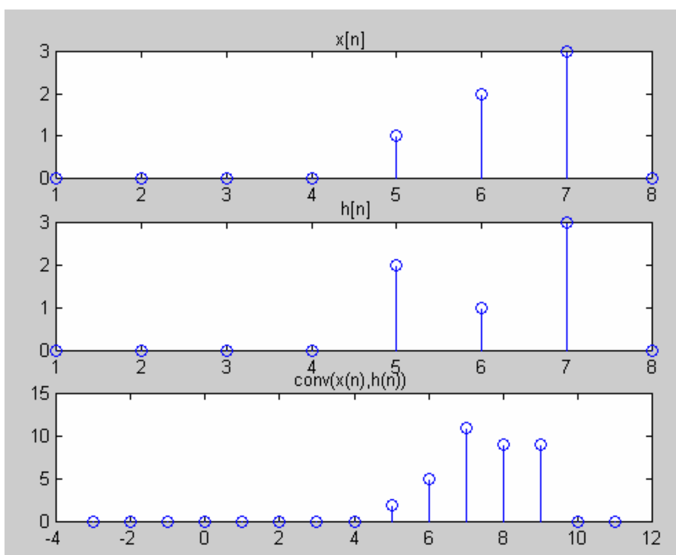
product and sum: 0 0 0 0 0 0 = 0

Dari hasil product and sum tersebut hasilnya dalam bentuk deret : 2 5 11 9 9

Hasil dicek dengan menggunakan m-file (matlab):

```
% contoh konvolusi
n=1:1:8
h=[0 0 0 0 1 2 3 0];
x=[0 0 0 0 2 1 3 0];
hx=conv (x,h);
subplot(3,1,1);
stem(n,h);
subplot(3,1,2);
stem(n,x);
nn=-3:1:11;          → 15 didapat dari jumlah h[n] + x[n] – 1  → 8+8 -1 =15
                     → dimulai dari -3 karena nilai (h[n-k] kali x[k]) ≠ 0 saat n=5
subplot(3,1,3);
stem(nn,hx);
```

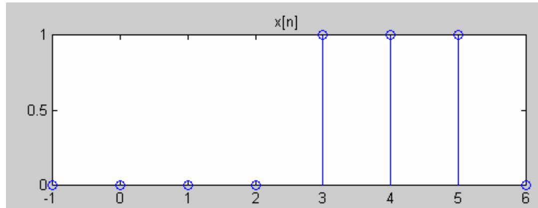
```
hasil
ans =
 Columns 1 through 15

   0    0    0    0    0    0    0    0    2    5   11    9    9    0    0
```



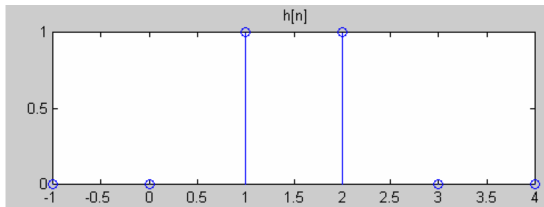
Contoh:

Ada system diskrit sbb:

Sinyal pertama:



Sinyal kedua:



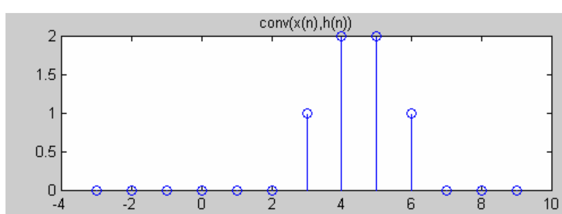
Cari konvolusi 2 sinyal tersebut dengan MATLAB.

Jawab:

m-file

```
h=[0 0 0 0 1 1 1 0];
x=[0 0 1 1 0 0];
hx=conv (x,h);

n=-1:1:6;
subplot(3,1,1);
stem(n,h);

n1=-1:1:4;
subplot(3,1,2);
stem(n1,x);

nn=-3:1:9;
subplot(3,1,3);
stem(nn,hx);
```

Hasil konvolusi :



TUGAS

1. Jelaskan yang dimaksud sistem invarian terhadap waktu
2. Ada suatu persamaan y(t) = 3.x(t) - z

   Dimana x(t) = sinyal masukan

   y(t) = sinyal keluaran

   z = konstanta

   Tentukan apakah persamaan di atas linier atau tidak linier?
3. Jelaskan prinsip kerja konvolusi (waktu diskrit)
4. Ada system diskrit sbb:

   Sinyal pertama: x[i]= [-1 -1 0 0 0]

   Sinyal kedua: h[i]= [0 0 0 1 1]

   Cari konvolusi 2 sinyal tersebut secara manual dan dengan MATLAB.

BAB III
TRANSFORMASI FOURIER

Tujuan:

1. Menjelaskan sinyal dalam domain waktu
2. Menjelaskan sinyal dalam domain frekuensi
3. Menjelaskan transformasi sinyal waktu diskrit (DTF)

DERET FOURIER

Deret fourier dipakai sebagai perangkat untuk menghitung spektrum dari sinyal periodik. Untuk sinyal bukan periodik, perangkat yang digunakan adalah transformasi fourier.

Definisi dari deret Fourier eksponensial kompleks adalah:

$$y(t) = \sum_{n=-\infty}^{+\infty} c_n e^{jn\omega_o t}$$

Dimana $\omega_o = 2\pi f_o = 2\pi / T_o$ disebut frekuensi sudut fundamental

c_n = koefisien Fourier

$$c_n = \frac{1}{T_o} \int_{t_o}^{t_o+T_o} y(t) e^{-jn\omega_o t} dt$$

t_0 = parameter yang bebas dipilih

TRANSFORMASI FOURIER

Satu bentuk transformasi yang umum digunakan untuk merubah sinyal dari domain waktu ke domain frekuensi adalah dengan transformasi Fourier:

$$X(\omega) = \mathscr{F}\{x(t)\} = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt$$

Persamaan di atas merupakan bentuk transformasi Fourier yang siap dikomputasi secara langsung dari bentuk sinyal x(t). Dimana x(t) merupakan fungsi yang tidak periodik terhadap waktu t.

Jika X(ω) diketahui, maka dapat diperoleh nilai x(t) dari persamaan invers transformasi fourier

$$x(t) = \mathscr{F}^{-1}\{X(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} X(\omega)e^{j\omega t}d\omega$$

Contoh:

1. Transformasikan ke dalam TF:
   a. unit impuls
   b. persamaan:

$$x(t) = p_a(t) = \begin{cases} 1 & |t| < a \\ 0 & |t| > a \end{cases}$$

Jawab:

a.

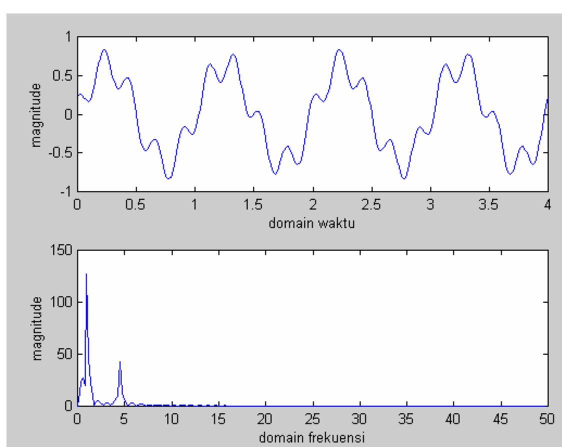
$$X(\omega) = \int_{-\infty}^{+\infty} \delta(t)e^{-j\omega t}dt = 1$$

b.

$$X(\omega) = \int_{-\infty}^{+\infty} p_a(t)e^{-j\omega t}dt = \int_{-a}^{+a} e^{-j\omega t}dt = \left[\frac{1}{-j\omega}e^{-j\omega t}\right]_{-a}^{+a} = \frac{1}{j\omega}\left(e^{+j\omega a} - e^{-j\omega a}\right)$$

$$X(\omega) = \frac{1}{j\omega}2j\sin(a\omega) = 2a\frac{\sin(a\omega)}{a\omega}$$

Sebagai contoh, ada sinyal sinus dengan frekuensi 5 Hz dan amplitudo 1 Volt. Dalam domain waktu anda akan melihat seperti pada Gambar di bawah bagian atas. Sementara dalam domain frekuensi akan didapatkan seperti pada bagian bawah. Untuk memperoleh hasil seperti gambar tersebut dapat memanfaatkan *library fft* yang tersedia pada Matlab.

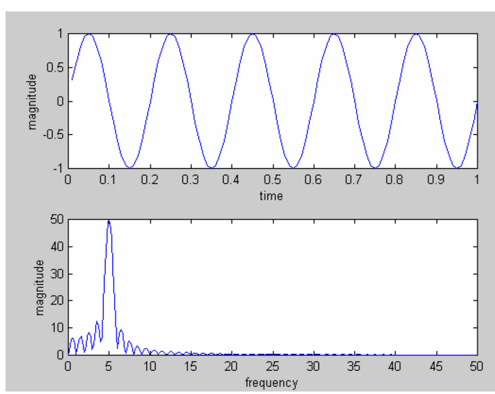


Untuk menghitung frekuensi dari suatu sinyal, sebuah implementasi diskrit dari analisa Fourier dapat digunakan, yang kemudian lebih disempurnakan dengan suatu algoritma yang kita kenal sebagai *Fast Fourier transform* (FFT). Secara umum teknik ini merupakan pendekatan yang terbaik untuk transformasi. Dalam hal ini input sinyal ke window ditetapkan memiliki panjang 2^m. Anda dapat memilih analisis window yang akan digunakan. Output dari syntax FFT(x,n) merupakan sebuah vector komplek, dengan n amplitudo komplek dari 0 Hz sampai dengan sampling frekuensi yang digunakan.

Contoh:

1. membangkitkan sinyal sinus dan spektrum frekuensinya dengan menggunakan m-file

```
Fs=100;                          %  ketelitian pixel grafik
t=(1:100)/Fs;
f=5;                             %  frekuensi =  5
s=sin(2*pi*f*t);                 %  sinyal sinus, amplitudo = 1
subplot(2,1,1)
plot(t,s)
xlabel('time')
ylabel('magnitude')

S=fft(s,512);                    %  spektrum frekuensi
w=(0:255)/256*(Fs/2);
subplot(2,1,2)
plot(w,abs(S(1:256)))
xlabel('frequency')
ylabel('magnitude')
```

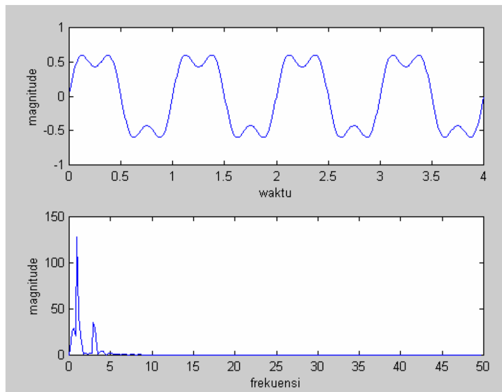


2. membangkitkan 2 sinyal sinus yang dijumlahkan dan spektrum frekuensinya dengan f1: amplitudo 2/pi dengan frekuensi 1 Hz, sinyal 2 f2: amplitudo pi*2/3 frekuensi 3 Hz. Tentukan keluaran 2 sinyal tersebut dalam domain waktu dan domain frekuensi.

   Jawab:

```
Fs=100;
```

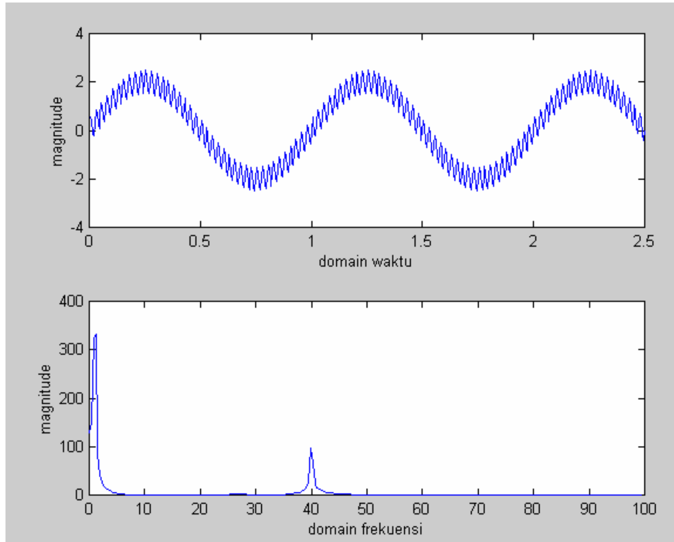
```
t=(1:400)/Fs;
f1=1;                              % frekuensi = 1
s1=(2/pi)*sin(2*pi*f1*t);          % amplitudo = 2/pi
f2=3;                              % frekuensi = 3
s2=(2/3/pi)*sin(2*pi*f2*t);        % amplitudo = 2/3/pi
s=s1+s2;
subplot(2,1,1)
plot(t,s)
xlabel('waktu')
ylabel('magnitude')

S=fft(s,512);
w=(0:255)/256*(Fs/2);
subplot(2,1,2)
plot(w,abs(S(1:256)))
xlabel('frekuensi')
ylabel('magnitude')
```



3. membangkitkan 2 sinyal sinus yang dijumlahkan dan spektrum frekuensinya dengan f1: amplitudo 2 dengan frekuensi 1 Hz, sinyal noise f2: amplitudo 0.5 frekuensi 40 Hz. Tentukan keluaran 2 sinyal tersebut dalam domain waktu dan domain frekuensi.

   Jawab:

```
Fs=200;
t=(1:500)/Fs;
f1=1;                                % frekuensi = 1
s1=(2)*sin(2*pi*f1*t);               % amplitudo = 2
f2=40;                               % frekuensi = 40
s2=(0.5)*sin(2*pi*f2*t);             % amplitudo = 0.5
s=s1+s2;
subplot(2,1,1)
plot(t,s)
xlabel('domain waktu')
ylabel('magnitude')

S=fft(s,512);
w=(0:255)/256*(Fs/2);
subplot(2,1,2)
plot(w,abs(S(1:256)))
xlabel('domain frekuensi')
ylabel('magnitude')
```



4.

DISCRETE FOURIER TRANSFORM (DFT)

DFT dibentuk dengan menggantikan integral berhingga dengan sederetan jumlahan pada suatu nilai berhingga:

$$X(\omega_k) \triangleq \sum_{n=0}^{N-1} x(t_n) e^{-j\omega_k t_n} \qquad k = 0, 1, 2, \ldots\ldots, N-1$$

Simbol $\triangleq$ memiliki arti *equal by definition* atau menyatakan bahwa sisi kiri secara definisi akan senilai dengan sisi kanan. Sementara $x(t_n)$ juga sebagai x(n), yang merupakan notasi sample ke-n pada sinyal input. $X(\omega_k)$ juga dapat dijumpai sebagai X(k) yang merupakan spectral sample ke-k.

Parameter lain yaitu:

- *j* = merupakan dasar dari bilangan komplek.
- $e \triangleq \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n = 2{,}718281828\ldots$
- $\omega_k = k\Omega$ = merupakan sample frekuensi ke-k. Sedangkan Ω merupakan interval sampling dalam radian dan memiliki nilai $\Omega = 2\pi/NT$.
- N = merupakan sample frekuensi yang digunakan.
- $T = 1/f_s$ = 1/(sampling rate).

DFT memiliki basis sinyal sinusoda dan merupakan bentuk komplek. Sehingga representasi domain frekuensi yang dihasilkan juga akan memiliki bentuk komplek. Dengan demikian dapat dilihat adanya bagian real dan imajiner, dan bisa juga hasil transformasi direpresentasikan dalam bentuk nilai absolute yang juga dikenal sebagai magnitudo respon frekuensinya dan magnitudo respon fase.

Selanjutnya untuk proses pengolahan sinyal digital, DFT diperlukan karena akan berhubungan dengan sinyal waktu diskrit, yang merupakan bentuk tersampel dari sinyal waktu kontinyu.

Contoh:

Jika ada suatu persamaan dalam bentuk

$$X(k) = \sum_{n=0}^{N-1} (3\cos(0{,}02\pi n))(\cos(k\omega_0 n) - j\sin(k\omega_0 n))$$

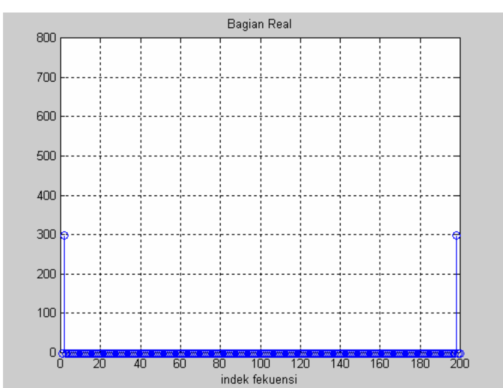
Carilah respon riil dan imajiner persamaan diatas terhadap domain frekuensi

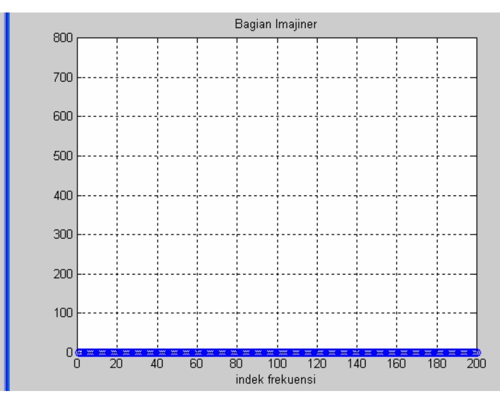
Jawab:

Dengan menggunakan m-file dapat diperoleh:

```
clear all;
N=200;
nn=N-1;
for k=1:200;
x_n=0.0;
for n=1:nn
   x_n = (3*cos(0.02*pi*n)).*(exp(-j*k*2*pi*n/200)) + x_n;
end
yR(k)=real(x_n);
yI(k)=imag(x_n);
magni_k(k)=sqrt(real(x_n).*real(x_n) +imag(x_n).*imag(x_n));
end
figure(1)
stem(yR)
axis([0 200 0 800])
xlabel('indek fekuensi')
title('Bagian Real')
grid;
figure(2)
stem(yI)
axis([0 200 0 800])
xlabel('indek frekuensi')
title('Bagian Imajiner')
grid;
```

Hasil





APLIKASI

Dalam kehidupan sehari hari, TF yang paling sering digunakan adalah DFT dimana transformasi ini sangat banyak digunakan dalam elektronika telekomunikasi. Contoh paling sederhana yaitu pengolahan sinyal radio, lebih spesifikasinya di – modulasi amplitudo.

Contoh:

Ada sinyal informasi yang sudah dimodulasi, sbb:

sinyal informasi : `sin(2*pi*t*100)`

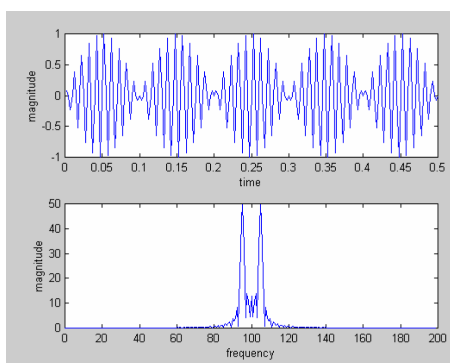
sinyal termodulasi : `sin(2*pi*f*t)`

Carilah domain frekuensi sinyal diatas.

Jawab:

Dengan menggunakan matlab,

```
Fs=400;                          %   ketelitian pixel grafik
t=(1:200)/Fs;
f=5;                             %   frekuensi
s=(sin(2*pi*t*100)).*(sin(2*pi*f*t));  %   sinyal sinus yang dibangkitkan
subplot(2,1,1)
plot(t,s)
xlabel('time')
ylabel('magnitude')

S=(fft(s,512));                  %   spektrum frekuensi
w=(0:255)/256*(Fs/2);
subplot(2,1,2)
plot(w,abs(S(1:256)))
xlabel('frequency')
ylabel('magnitude')
```



TUGAS

1. Apa perbedaan antara transformasi fourier dan deret fourier
2. Membangkitkan 3 sinyal sinus yang dijumlahkan, sinyal 1: amplitudo 2 dengan frekuensi 1 Hz, sinyal 2: amplitudo 1 dengan frekuensi 10 Hz, dan sinyal 3: amplitudo 0.5 frekuensi 60 Hz. Tentukan keluaran 3 sinyal tersebut dalam domain waktu dan domain frekuensi.
3. Apa yang dimaksud dengan DTF
4. Apa fungsi FFT dan aplikasinya

BAB IV
MODULASI

Tujuan:

1. Menjelaskan pengertian modulasi
2. Menjelaskan jenis-jenis modulasi
3. Menjelaskan multiplekser

MODULASI

Modulasi merupakan suatu proses dimana properti atau parameter dari suatu gelombang divariasikan secara proporsional terhadap gelombang yang lain. Parameter yang diubah tergantung dari modulasi yang diberikan.

Manfaat dari modulasi, antara lain:

1. untuk menjaga kerahasiaan suatu informasi maka diperlukan suatu modulasi terutama dalam bidang militer
2. untuk menggandakan saluran kanal dalam telekomunikasi
3. untuk meyederhanakan data

Dalam proses modulasi, melibatkan 2 sinyal yang berkaitan yaitu sinyal informasi / sinyal pemodulasi dan sinyal pembawa (*carrier*). Pada proses modulasi, sinyal pembawa seolah-olah membawa sinyal informasi yang biasanya berbentuk sinusoida (analog), dengan persamaan sebagai berikut:

$$V = A \sin (\omega.t + \theta)$$

Dimana

A	=	amplitudo
θ	=	sudut fasa
ω	=	2*pi*f

Dari persamaan diatas, sinyal pembawa tergantung 3 variabel bebas yang bisa dimodifikasi dan menjadi dasar jenis-jenis modulasi.

JENIS MODULASI

Jenis dasar dari modulasi:

1. modulasi amplitudo
2. modulasi frekuensi
3. modulasi phasa

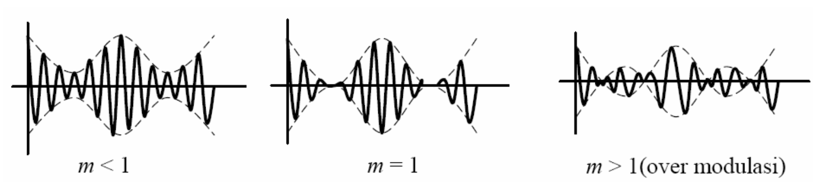
Pada perkembangannya, modulasi dasar diatas dimodifikasi menjadi FSK (Frequency Shift Keying), QAM (Quadrature Amplitude Modulation), PSK (Phase Shift Keying).

MODULASI AMPLITUDO

Modulasi amplitudo mempergunakan amplitudo sinyal pembawa untuk membedakan amplitudo sinyal informasi (jika sinyal analog) atau membedakan kondisi 0/1 (jika sinyal informasi berupa sinyal digital).

Bentuk gelombang sinyal AM bisa diperoleh dari persamaan:

$$\phi_{AM}(t) = A\,[\,1 + m \cos \omega_m t\,]\cos \omega_c t$$



Kerapatan spektrum dari sinyal AM adalah :

$$\Phi_{AM}(\omega) = ½\, F(\omega+\omega_c) + ½\, F(\omega-\omega_c) + \pi A\,\delta(\omega+\omega_c) + \pi A\,\delta(\omega-\omega_c)$$

AM merupakan cara modulasi yang paling mudah, namun mudah dipengaruhi oleh keadaan media transmisinya.

MODULASI FREKUENSI

Modulasi ini mempergunakan frekuensi dari sinyal pembawa untuk menyatakan sinyal indormasi. Pada FM, amplitudo dan phasenya tetap, sedangkan yang berubah-ubah adalah frekuensinya.

Frekuensi sesaat proporsional terhadap sinyal input $f(t)$:

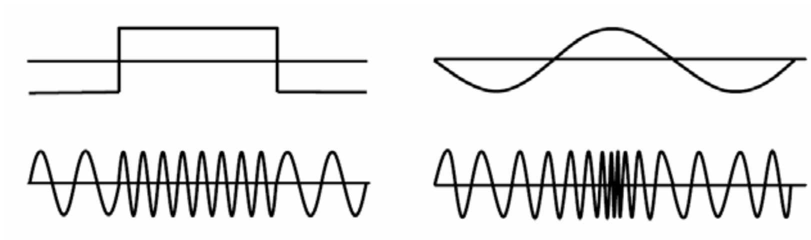
$$\omega_i = \omega_c + k_f f(t)$$

Dimana ω_c , θ_0 dan k_f merupakan konstanta.

Karena frekuensi berhubungan secara linear terhadap *f*(*t*), modulasi seperti ini disebut *Frequency Modulation* (FM).

Sudut phase dari sinyal FM adalah :

$$\theta(t) = \int_0^t \omega_i(\tau)d\tau = \omega_c t + \int_0^t k_f f(\tau)d\tau + \theta_o$$



Pada Gambar di atas dapat dilihat bahwa frekuensi hasil modulasi berubah lebih besar saat sinyal informasi menjadi beramplitudo lebih besar / positif. Cara modulasi frekuensi lebih sukar dari pada AM, tetapi tidak terlalu mudah dipengaruhi keadaan transmisinya.

Keuntungan digunakannya FM antara lain:

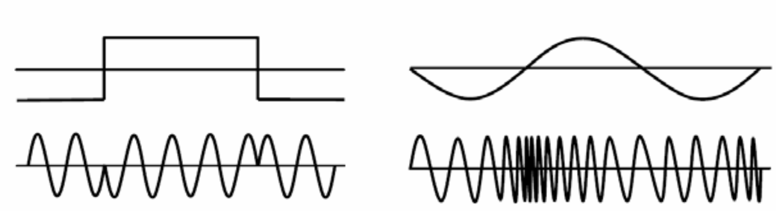
- meningkatkan redaman terhadap noise yang cukup baik dibandingkan AM
- daya pancar yang dibutuhkan lebih kecil dari AM, untuk mendapatkan redaman yang sama baik dengan AM.
- Peralatan pemancar lebih efisien

Kekurangan digunakannya FM antara lain:

- Gelombang FM memerlukan bandwidth besar dibandingkan AM
- Sistem FM lebih rumit daripada AM

MODULASI PHASE

Modulasi Phase mempergunakan perbedaan sudut fasa dari sinyal pembawa untuk menyatakan sinyal informasi. Pada PM, phase gelombang pembawa proporsional terhadap sinyal informasi, serta amplitude dan frekuensinya tetap hanya nilai phasenya yang berubah-ubah.

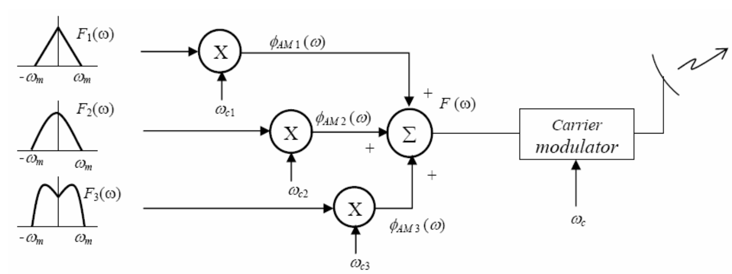


Dari gambar di atas bagian kiri, dapat dilihat bahwa setiap perubahan nilai dari positif dan negatif maka sinyal hasil modulasi akan berubah phase. Di antara ketiga modulasi di atas yang paling sulit dan rangkaiannya rumit adalah PM.

MULTIPLEKSER

Multiplexing adalah pengiriman secara simultan beberapa sinyal informasi dengan menggunakan satu kanal. Dengan *multiplexing* sistem akan menjadi lebih efisien.
Multiplekser dalam telekomunikasi yang paling banyak digunakan yaitu FDM (Frequency Division Multiplexing). FDM sering digunakan pada jaringan sambungan telpon, siaran radio dan televisi

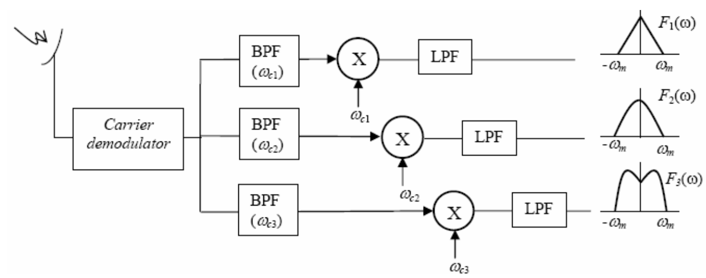
Contoh:



DEMULTIPLEKSER

Demultiplekser merupakan kebalikan dari mutiplekser, dimana prinsip dasar demultiplekser mempunyai sebuah saluran masukan dipilah-pilah menjadi beberapa saluran keluaran dengan pengendali yang berguna untuk menghubungkan saluran masukan dihubungkan ke saluran keluaran yang mana.

Contoh:



TUGAS

1. Apa yang dimaksud dengan modulasi
2. Sebut dan jelaskan jenis-jenis modulasi dasar
3. Jelaskan yang dimaksud dengan multiplekser pada sistem telekomunikasi
4. Sebutkan 4 aplikasi modulasi di sekitar anda

BAB V
FILTER

Tujuan:

1. Menjelaskan pengertian filter
2. Menjelaskan jenis-jenis filter
3. Menjelaskan aplikasi filter

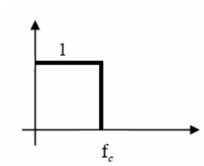
PENDAHULUAN

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.

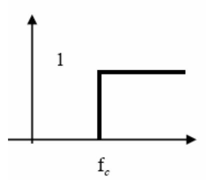
Filter diklasifikasikan :

- Filter analog dan filter digital
    - Filter analog : sinyal masukan berupa sinyal analog
        - Filter pasif dan filter aktif
            - Filter pasif : filter yang hanya disusun komponen tahanan, induktor dan kapasitor
            - Filter aktif : filter yang disusun komponen op amp atau transistor ditambah tahanan, induktor, dan kapasitor
    - Filter digital : sinyal masukan berupa sinyal diskrit
        - FIR (Finite Impulse Response)
        - IIR (Infinite Impulse Response)
- Filter menurut frekuensi yang disaring:
    - Filter low pass:

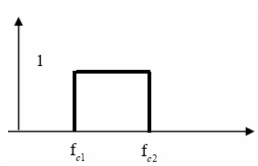
- Filter yang tegangan keluarannya tetap sampai ke suatu frekuensi cut off (fc) . Bersama naiknya frekuensi di atas fc, tegangan keluarannya melemah/menghilang. Frekuensi cut off / frekuensi 3 dB / frekuensi 0,707 .



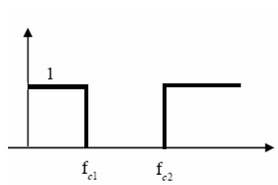
- Filter high pass
  - Filter yang memperlemah tegangan keluaran untuk frekuensi di bawah nilai frekuensi cut off fc. Di atas fc besar tegangan keluaran tetap.



- Filter band pass
  - Filter yang hanya melewatkan sinyal keluaran yang berfrekuensi tertentu (yang dititik beratkan di 1 nilai fc) dan melemahkan tegangan sinyal keluaran semua frekuensi selain frekuensi tertentu.

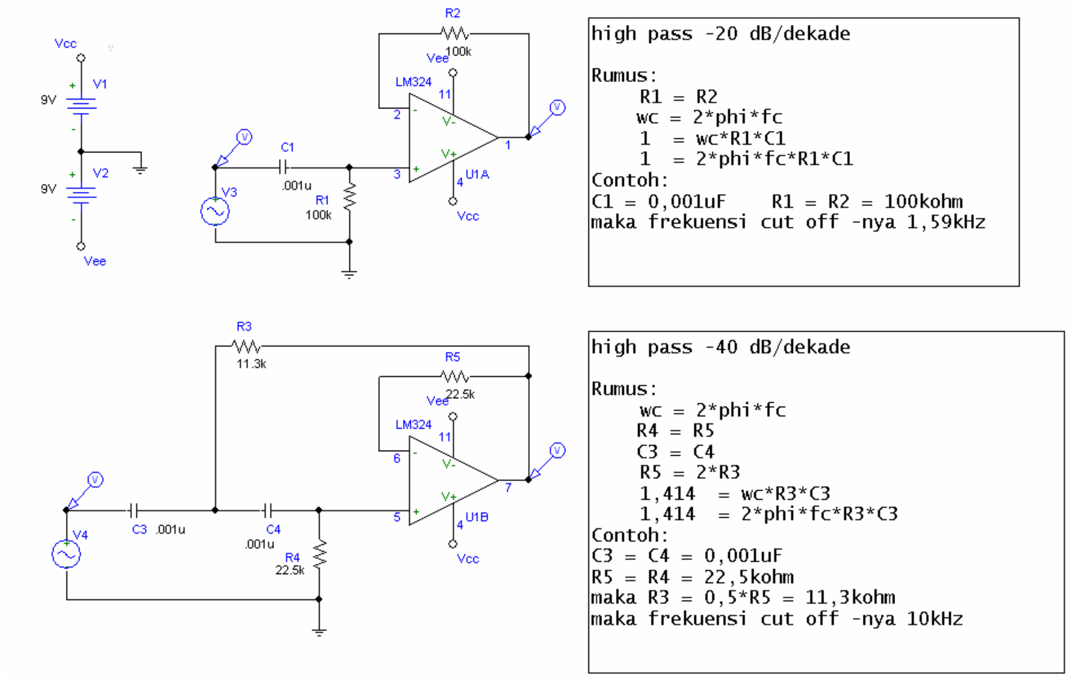


- Filter band elimination
  - Filter yang menyaring / melemahkan tegangan sinyal keluaran yang memiliki suatu nilai frekuensi tertentu dan meloloskan sinyal keluaran yang lain.

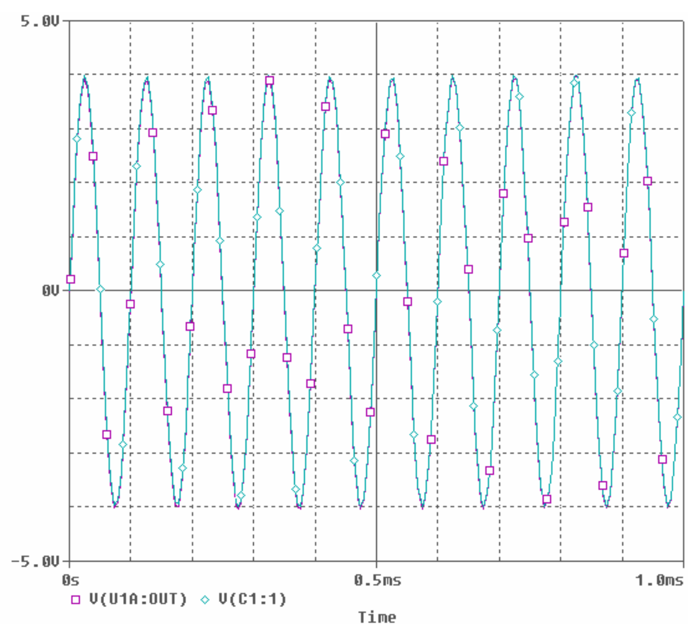


FILTER ANALOG

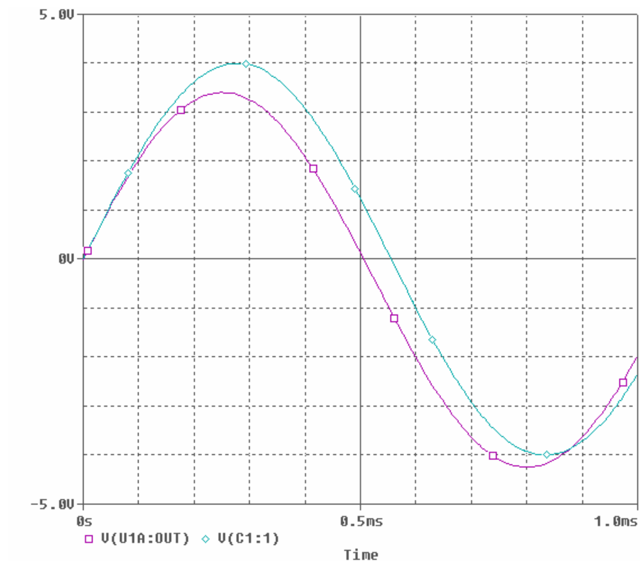
Filter high pass



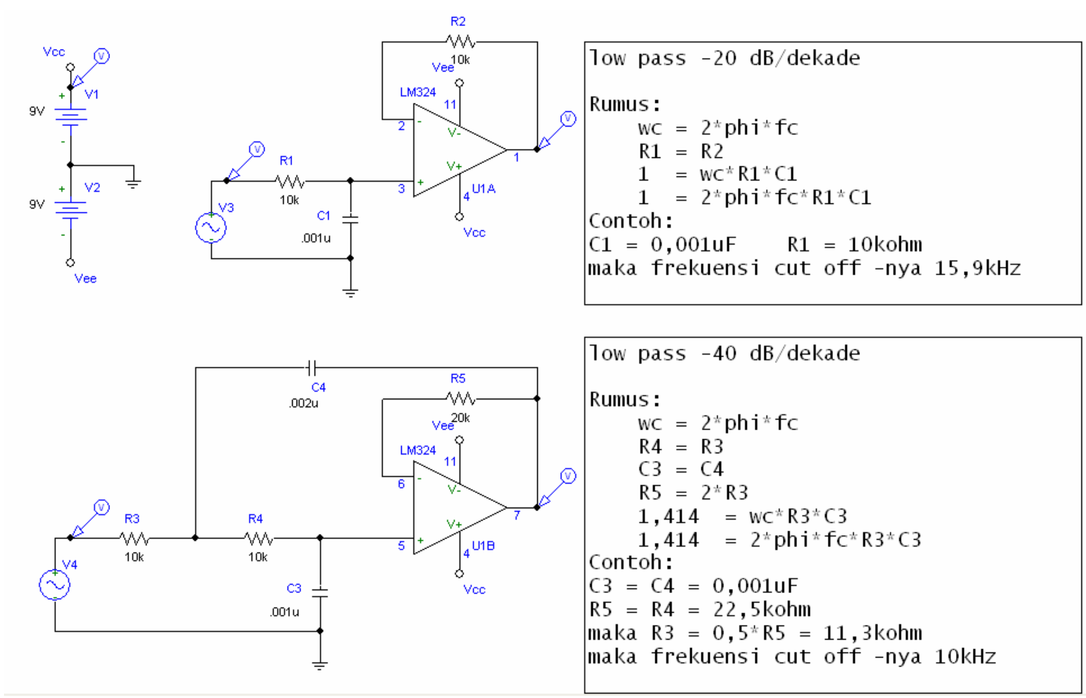
Pada filter high pass diberi masukan V3 sebesar 10 kHz, sinyal keluaran sama dengan sinyal masukan (meloloskan), seperti grafik di bawah.



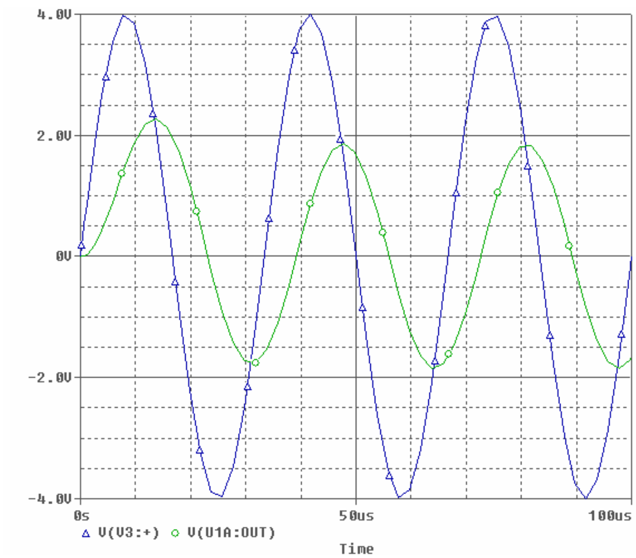
Pada filter high pass diberi masukan V3 sebesar 1 kHz, sinyal keluaran mempunyai amplitudo lebih kecil daripada sinyal masukan (melemahkan) , seperti grafik di bawah.



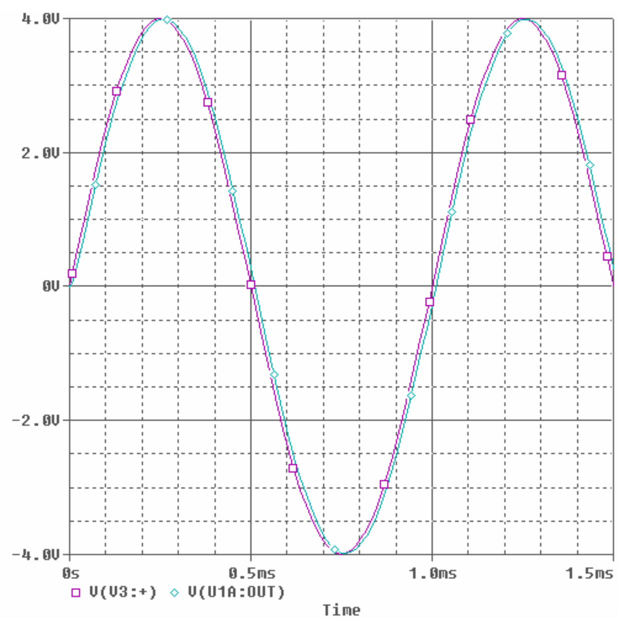
Filter low pass



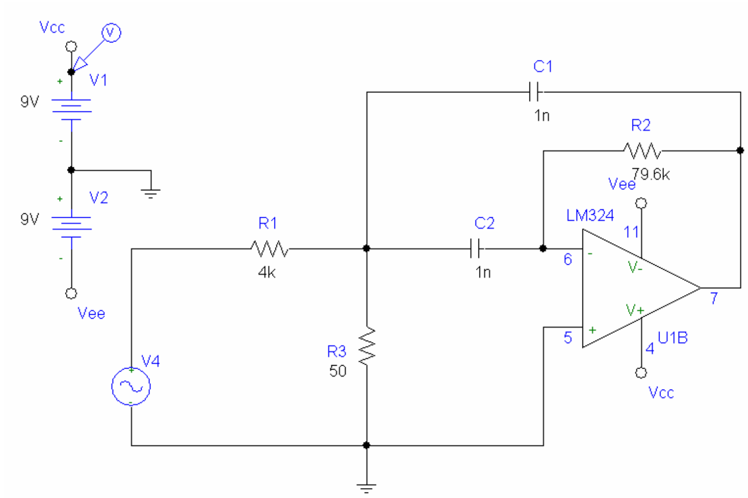
Pada filter high pass diberi masukan V3 sebesar 30 kHz, sinyal keluaran mempunyai amplitudo lebih kecil daripada sinyal masukan (melemahkan) , seperti grafik di bawah.



Pada filter high pass diberi masukan V3 sebesar 10 kHz, sinyal keluaran sama dengan sinyal masukan (meloloskan) , seperti grafik di bawah.

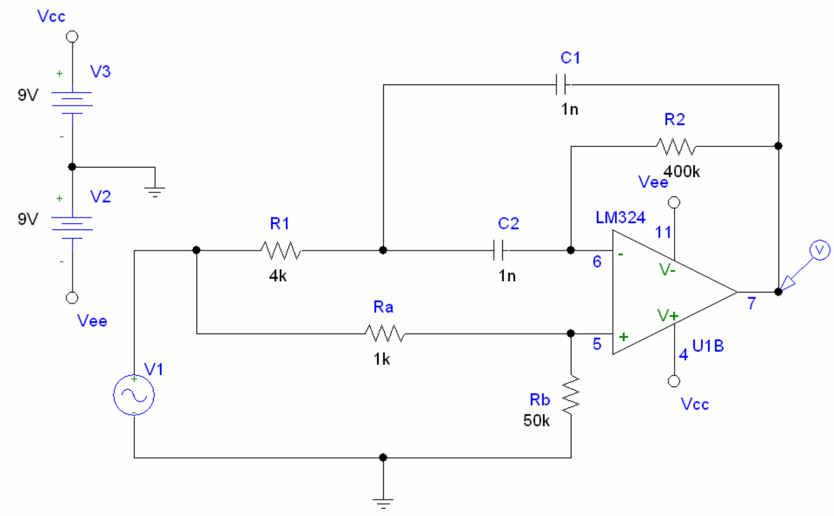


Filter band pass




```
band pass
Dibagi 2, yaitu:
    # filter pita sempit
        B < 0.1 wr
        Q > 10
    # filter pita lebar
        B > 0.1 wr
        Q < 10

Keterangan variabel
    Ar = gain tegangan maksimum
    Vmaks = tegangan keluaran maksimum
    fh = frekuensi yang gain tegangannya 0,707 Ar,
     dimana frekuensi ini lebih besar daripada fr
    fl = frekuensi yang gain tegangannya 0,707 Ar,
     dimana frekuensi ini lebih kecil daripada fr
    fr = frekuensi resonan
    B  = lebar pita (rad/dtk)
    Q  = faktor kualitas
    C  = nilai kapasitor yang digunakan (C=C1=C2)

Rumus:
    wr = 2*phi*fr (rad/dtk)
    B  = wh - wl (rad/dtk)
    Q  = wr / B
    R2 = 2 / (B*C)
    R1 = R2 / (Ar*2)
    R3 = R2 / (4*Q*Q - 2*Ar)

Ditentukan:
    C1 = C2 = 0,001uF        Ar= 10
    fr = 80kHz  (pita sempit Q = 20)
Dihitung:
    wr = 2*pi*80k = 502,4 krad/dtk
    B  = 502,4 k / 20 = 25,12krad/dtk
    R2 = 79,6 kohm
    R1 = 4 kohm
    R3 = 50 ohm
```

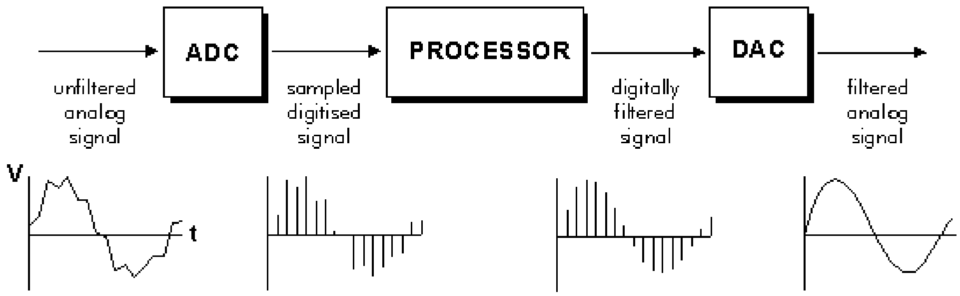
Filter Band Elimination




```
band elimination
Dibagi 2, yaitu:
    # filter pita sempit
        B < 0.1 wr
        Q > 10
    # filter pita lebar
        B > 0.1 wr
        Q < 10

Keterangan variabel
    Ar = gain tegangan maksimum
    Vmaks = tegangan keluaran maksimum
    fh = frekuensi yang gain tegangannya 0,707 Ar,
     dimana frekuensi ini lebih besar daripada fr
    fl = frekuensi yang gain tegangannya 0,707 Ar,
     dimana frekuensi ini lebih kecil daripada fr
    fr = frekuensi resonan
    B  = lebar pita (rad/dtk)
    Q  = faktor kualitas
    C  = nilai kapasitor yang digunakan (C=C1=C2)

Rumus:
    wr = 2*phi*fr (rad/dtk)
    B  = wh - wl (rad/dtk)
    Q  = wr / B
    C1 = C2
    R2 = 2 / (B*C)
    R1 = R2 / (4*Q*Q)
    Rb = (2*Q*Q*Ra)

Ditentukan:
    C1 = C2 = 0,01uF
    fr = 400kHz  (pita sempit Q = 5)
Dihitung:
    wr = 2*pi*400k = 2,51 krad/dtk
    B  = 2,51 k / 5 = 500 krad/dtk
    R2 = 400 kohm
    R1 = 4 kohm
    Rb = 50 ohm
```

FILTER DIGITAL

Pada sistem digital, filter dirancang berupa software dan dapat diaplikasikan seperti pada blok di bawah.



Dibandingkan dengan filter analog filter digital mempunyai kelebihan antara lain:

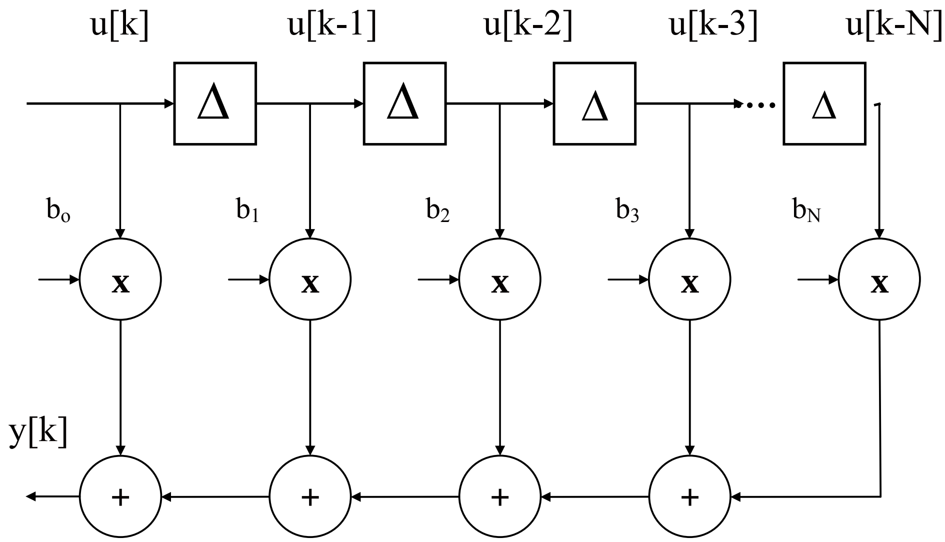
- Filter digital lebih mudah diubah-ubah dan *programmable*. Filter digital membutuhkan memori-prosessor dan filter analog membutuhkan rangkaian filter, sehingga untuk mengubah filter berarti mengubah rangkaian.
- Filter digital lebih mudah di design, tes, dan diimplementasikan/dihubungkan pada komputer
- Karakteristik rangkaian analog yang lebih mudah terkena gangguan (noise / temperatur). Filter digital lebih stabil – tergantung komputer
- Dalam praktek, filter digital menggunakan DSP (PLC/mikrokontroller/DSP card) relatif lebih simpel dan *compact* dibandingkan rangkaian analog.

FIR

Filter FIR adalah salah satu tipe dari filter digital yang dipakai pada aplikasi Digital Signal Processing (DSP). FIR kepanjangan dari Finite Impulse Response. Mengapa disebut respons impulsnya terbatas (finite)? Karena *tidak ada feedback* didalam filter, jika memasukkan sebuah impulse (yaitu sebuah sinyal ′1′ diikuti dengan banyak sinyal ′0′), sinyal nol akan keluar setelah sinyal 1 melewati semua delay line dengan koefisiennya.

Keuntungan filter FIR antara lain adalah stabil dan memiliki phasa yang linier. Sedangkan kerugiannya adalah filter FIR terkadang membutuhkan lebih banyak memory dan/atau perhitungan untuk mencapai karakteristik respon filter yang diberikan. Dan juga, respon tertentu tidak mudah dilaksanakan untuk diimplementasikan dengan filter FIR.

Blok dari filter FIR seperti pada Gambar di bawah



Bentuk respon impuls dari filter FIR dinyatakan

$$h[0] = b_0, h[1] = b_1, ..., h[N] = b_N, h[N+1] = 0, ...$$

$$y[k] = b_0.u[k] + b_1.u[k-1] + ... + b_N.u[k-N]$$

Fungsi alih dari filter FIR

$$H(z) = \frac{B(z)}{z^N} = b_0 + b_1 z^{-1} + ... + b_N z^{-N}$$

Filter FIR menggunakan MATLAB

b=fir1(n,Wn,type,window), windowed linear-phase FIR design, n is filter order, Wn defines band-edges, type is \`high',\`stop',…

b=fir2(n,f,m,window), windowed FIR design based on inverse Fourier transform with frequency points f and corresponding magnitude response m

b=remez(n,f,m), equiripple linear-phase FIR design with Parks-McClellan (Remez exchange) algorithm

Contoh:

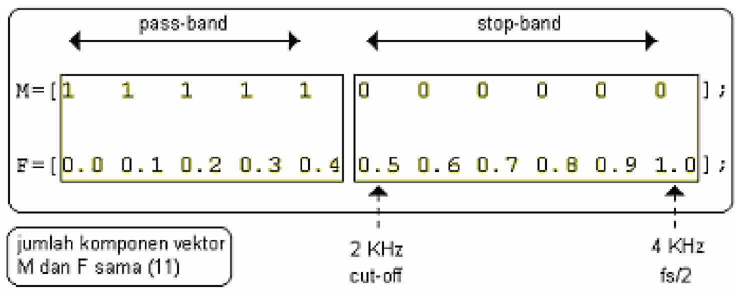
Menghitung koefisien filter low-pass dengan frekuensi cut-off 2 KHz yang mempunyai frekuensi sampling dengan menggunakan Matlab.

Jawab:

Menghitung nilai frekuensi cut-off pada domain frekuensi

Dirancang :

- setengah dari frekuensi sampling ➔8kHz : 2 = 4kHz (pada sumbu fekuensi bernilai 1 saat 4kHz)
- saat frekuensi cut off 2kHz ➔ 2 kHz : 4 kHz = 0,5 (pada sumbu frekuensi bernilai 0,5 saat 4 kHz)



Untuk menghitung nilai koefisien dari filter dengan MATLAB dapat menggunakan perintah :

X=fir2(orde,F,M,blackman(orde+1));

Dimana

x = nilai koefisien dari filter yang dirancang

orde = nilai orde dari filter yang dirancang

F = nilai fekuensi

M = nilai magnitude

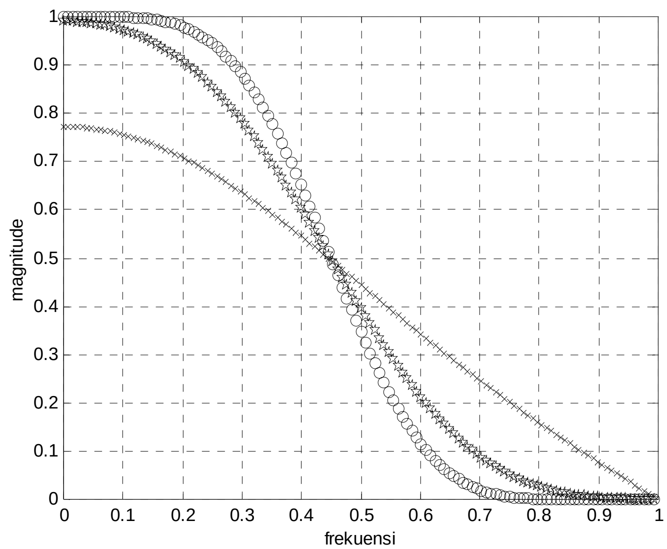
Blackman = jenis membangkitkan blackman windows (selain blackman ada juga hamming, chebwin, hanning, dan lainnya)

m-file

```
M=[1 1 1 1 1 0 0 0 0 0 0];
F=[0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1];

% orde 15
nx1=fir2(15,F,M,blackman(16));
[y1,w1]=freqz(nx1,1,128);
% orde 10
nx2=fir2(10,F,M,blackman(11));
[y2,w2]=freqz(nx2,1,128);
% orde 5
nx3=fir2(5,F,M,blackman(6));
[y3,w3]=freqz(nx3,1,128);
```

```
plot(w1/pi,abs(y1),'kO',w2/pi,abs(y2),'kp',w3/pi,abs(y3),'kx'  );
grid on;
ylabel('magnitude');
xlabel('frekuensi');
```

hasil



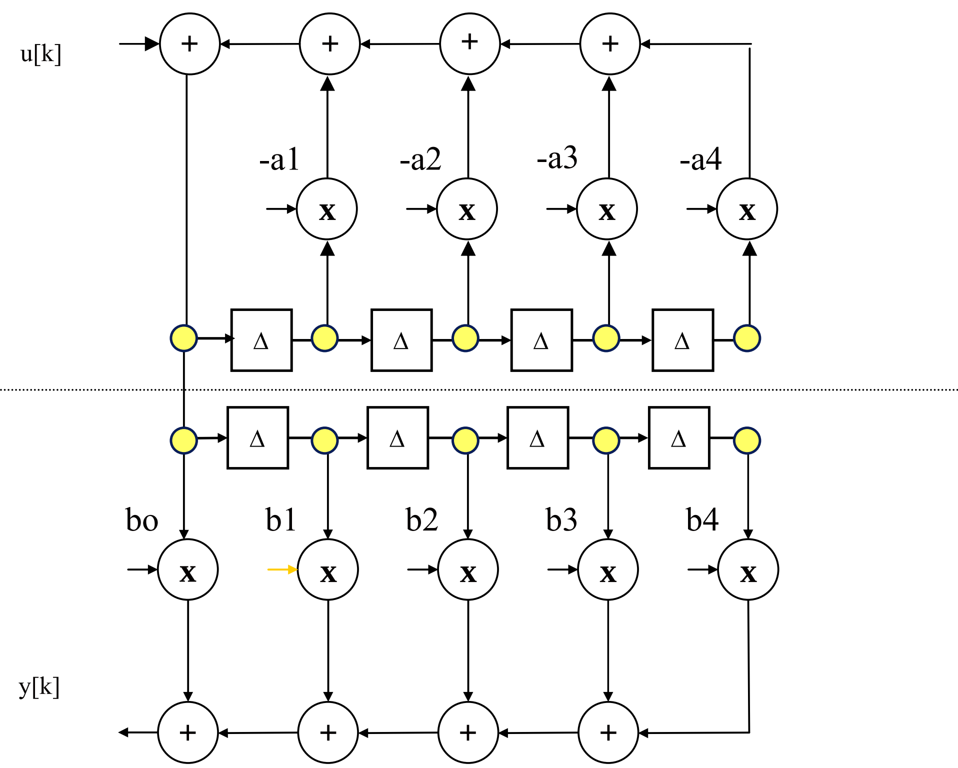
Dari gambar diatas dapat dibandingkan semakin besar orde yang digunakan maka semakin curam filter low pass (semakin mendekati ideal).

IIR

Filter IIR adalah salah satu tipe dari filter digital yang dipakai pada aplikasi Digital Signal Processing (DSP). IIR kepanjangan dari Infinite Impulse Response. Mengapa disebut response impulsnya tak terbatas (infinite)? Karena *adanya feedback* didalam filter, jika anda memasukkan sebuah impulse (yaitu sebuah sinyal ′1′ diikuti dengan banyak sinyal ′0′), maka pada outputnya akan terus menerus berosilasi karena adanya umpan balik, walaupun pada prakteknya akan hilang pada suatu saat.

Keuntungan filter IIR antara lain adalah membutuhkan koefisien yang lebih sedikit untuk respon frekuensi yang curam sehingga dapat mengurangi jumlah waktu komputasi.

Blok dari filter IIR seperti pada Gambar di bawah



Bentuk respon impuls dari filter IIR dinyatakan

$$y[k]+a_1.y[k-1]+\ldots+a_N.y[k-N]=b_0.u[k]+b_1.u[k-1]+\ldots+b_N.u[k-N]$$

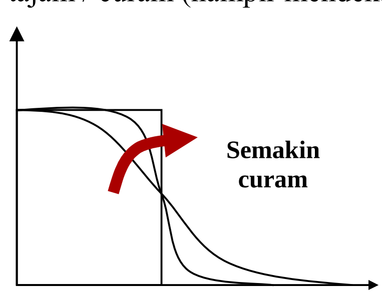
$$y[k]=\underbrace{b_0.u[k]+b_1.u[k-1]+\ldots+b_N.u[k-N]}_{\text{`MA'}}\underbrace{-a_1.y[k-1]-\ldots-a_N.y[k-N]}_{\text{`AR'}}$$

Fungsi alih dari filter IIR

$$H(z)=\frac{B(z)}{A(z)}=\frac{b_0z^N+b_1z^{N-1}+\ldots+b_N}{z^N+a_1z^{N-1}+\ldots+a_N}=\frac{b_0+b_1z^{-1}+\ldots+b_Nz^{-N}}{1+a_1z^{-1}+\ldots+a_Nz^{-N}}$$

Kelebihan IIR dibandingkan FIR

- Pada filter IIR Orde rendah dapat menghasilkan respon frekuensi yang tajam / curam (hampir mendekati ideal). Contoh gambar low pass filter IIR



- Lebih simple perhitungannya

Kekurangan IIR dibandingkan FIR

- Design lebih sulit
- Kestabilan perlu diperiksa
- Respon phasa tidak mudah dikontrol

Filter FIR menggunakan MATLAB

[b,a]=butter/cheby1/cheby2/ellip(n,…,Wn),

IIR LP/HP/BP/BS design based on analog prototypes, pre-warping, bilinear transform, …

Contoh:

Desain filter lowpass dengan frekuensi cut off 5 kHz pada fekuensi sampling sebesar 16 kHz. Tentukan nilai koefisiennya

Jawab:

Diketahui frekuensi sampling filter = 16 kHz

Setengah dari frekuensi sampling filter = 8 kHz

Frekuensi cut off filter = 5 kHz

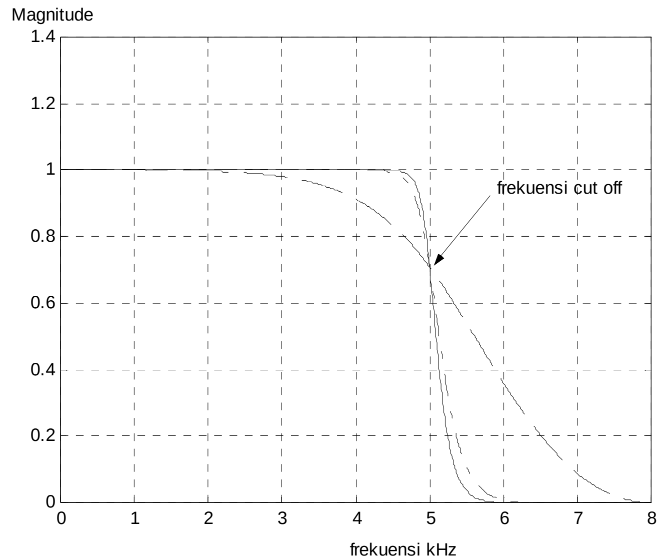
Nilai wn sebagai frekuensi cut off bernilai = (5 kHz / 8 kHz) = 0,625

Dirancang filter low pass IIR orde 2, orde 10, dan orde 15 sebagai perbandingan

[a,b]=butter(orde,frekuensi_cut_off);

m-file

```
f=8;
n1=2;
n2=10;
n3=15;

wn=0.625;
[a1,b1]=butter(n1,wn);
[a2,b2]=butter(n2,wn);
[a3,b3]=butter(n3,wn);

[h1,w1]=freqz(a1,b1,1024);
[h2,w2]=freqz(a2,b2,1024);
[h3,w3]=freqz(a3,b3,1024);

%plot((w*f)/pi,20*log(abs(h)));
plot((w1*f)/pi,(abs(h1)),'k',(w2*f)/pi,(abs(h2)),'b',(w3*f)/pi,
(abs(h3)),'k');
grid on;
```



Dari Gambar di atas, semakin besar ordenya, maka filter lowpass IIR semakin curam/mendekati ideal.

Untuk mendapatkan koefisien filter dengan metode butterworth. Pada Matlab digunakan untuk mendesain filter IIR low pass menggunakan metode butterworth orde N dengan frekuensi cut-off pada wn, dan menghasilkan koefisien filter pada vector B (numerator) dan vector A (denumerator) sebanyak N+1.

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 z^N + b_1 z^{N-1} + \ldots + b_N}{z^N + a_1 z^{N-1} + \ldots + a_N} = \frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}}$$

Nilai dari frekuensi cut off wn harus bernilai 0 < wn < 1, dimana 1 menunjukkan setengah dari frekuensi sampling.

[a,b]=butter(orde,frekuensi_cutoff,'high'); ➔ untuk filter high pass

[a,b]=butter(orde,[frek_cutoff1 frek_cutoff2]); ➔ untuk filter band pass

[a,b]=butter(orde,[frek_cutoff1 frek_cutoff2],'stop'); ➔ untuk filter stop pass

TUGAS

1. Jelaskan yang dimaksud filter
2. Jelaskan yang dimaksud filter digital IIR dan FIR
3. Desain filter highpass dengan frekuensi cut off 40 kHz pada fekuensi sampling sebesar 100 kHz. Tentukan nilai koefisiennya dengan menggunakan IIR
4. Desain filter lowpass dengan frekuensi cut off 15kHz dengan menggunakan opamp

LAMPIRAN

Laplace Transform:	$x(t) \Leftrightarrow X(s)$	dimana $X(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt$
Continuous-Time Fourier Transform:	$x(t) \Leftrightarrow X(j\omega)$	dimana $X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$
z Transform:	$x[n] \Leftrightarrow X(z)$	dimana $X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$
Discrete-Time Fourier Transform:	$x[n] \Leftrightarrow X(e^{j\Omega})$	dimana $X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}$

$s = j\omega$ **dan** $z = e^{j\Omega}$

Transformasi laplace

$$\mathscr{L}\{1\}(s) = \frac{1}{s} \qquad \mathscr{L}\{\cos bt\}(s) = \frac{s}{s^2+b^2}$$

$$\mathscr{L}\{e^{at}\}(s) = \frac{1}{s-a} \qquad \mathscr{L}\{e^{at}t^n\}(s) = \frac{n!}{(s-a)^{n+1}}, \quad n = 1, 2, \ldots$$

$$\mathscr{L}\{t^n\}(s) = \frac{n!}{s^{n+1}}, \quad n = 1, 2, \ldots \qquad \mathscr{L}\{e^{at}\sin bt\}(s) = \frac{b}{(s-a)^2+b^2}$$

$$\mathscr{L}\{\sin bt\}(s) = \frac{b}{s^2+b^2} \qquad \mathscr{L}\{e^{at}\cos bt\}(s) = \frac{s-a}{(s-a)^2+b^2}$$

$$\mathscr{L}\{f+g\} = \mathscr{L}\{f\} + \mathscr{L}\{g\}$$

$$\mathscr{L}\{cf\} = c\mathscr{L}\{f\} \quad \text{for any constant } c$$

$$\mathscr{L}\{e^{at}f(t)\}(s) = \mathscr{L}\{f\}(s-a)$$

$$\mathscr{L}\{f'\}(s) = s\mathscr{L}\{f\}(s) - f(0)$$

$$\mathscr{L}\{f''\}(s) = s^2\mathscr{L}\{f\}(s) - sf(0) - f'(0)$$

$$\mathscr{L}\{f^{(n)}\}(s) = s^n\mathscr{L}\{f\}(s) - s^{n-1}f(0) - s^{n-2}f'(0) - \cdots - f^{(n-1)}(0)$$

$$\mathscr{L}\{t^n f(t)\}(s) = (-1)^n \frac{d^n}{ds^n}\big(\mathscr{L}\{f\}(s)\big)$$

$$\mathscr{L}\{f(t-a)u(t-a)\}(s) = e^{-as}\mathscr{L}\{f(t)\}(s)$$

$$\mathscr{L}\{g(t)u(t-a)\}(s) = e^{-as}\mathscr{L}\{g(t+a)\}(s)$$

Transformasi Fourier

Function, f(t)	**Fourier Transform, F(ω)**
Definition of Inverse Fourier Transform $f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega$	*Definition of Fourier Transform* $F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$
$f(t-t_0)$	$F(\omega)e^{-j\omega t_0}$
$f(t)e^{j\omega_0 t}$	$F(\omega-\omega_0)$
$f(\alpha t)$	$\frac{1}{\|\alpha\|}F(\frac{\omega}{\alpha})$
$F(t)$	$2\pi f(-\omega)$
$F(t)$	$2\pi f(-\omega)$
$\frac{d^n f(t)}{dt^n}$	$(j\omega)^n F(\omega)$
$(-jt)^n f(t)$	$\frac{d^n F(\omega)}{d\omega^n}$
$\int_{-\infty}^{t} f(\tau)d\tau$	$\frac{F(\omega)}{j\omega} + \pi F(0)\delta(\omega)$
$\delta(t)$	1
$e^{j\omega_0 t}$	$2\pi\delta(\omega-\omega_0)$
$\cos(\omega_0 t)$	$\pi\left[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\right]$
$\sin(\omega_0 t)$	$\frac{\pi}{j}\left[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)\right]$
$u(t)\cos(\omega_0 t)$	$\frac{\pi}{2}\left[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\right]+\frac{j\omega}{\omega_0^2-\omega^2}$
$u(t)\sin(\omega_0 t)$	$\frac{\pi}{2j}\left[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)\right]+\frac{\omega^2}{\omega_0^2-\omega^2}$
$u(t)e^{-\alpha t}\cos(\omega_0 t)$	$\frac{(\alpha+j\omega)}{\omega_0^2+(\alpha+j\omega)^2}$

Penyederhanaan persamaan

$\cos(x) = \dfrac{e^{jx} + e^{-jx}}{2}$
$\sin(x) = \dfrac{e^{jx} - e^{-jx}}{2j}$
$\cos(x \pm y) = \cos(x)\cos(y) \mp \sin(x)\sin(y)$
$\sin(x \pm y) = \sin(x)\cos(y) \pm \cos(x)\sin(y)$
$\cos(2x) = \cos^2(x) - \sin^2(x)$
$\sin(2x) = 2\sin(x)\cos(x)$
$2\cos^2(x) = 1 + \cos(2x)$

Integral

$\int \cos(x)dx$	$\sin(x)$
$\int \sin(x)dx$	$-\cos(x)$
$\int x\cos(x)dx$	$\cos(x) + x\sin(x)$
$\int x\sin(x)dx$	$\sin(x) - x\cos(x)$
$\int x^2 \cos(x)dx$	$2x\cos(x) + (x^2 - 2)\sin(x)$
$\int x^2 \sin(x)dx$	$2x\sin(x) - (x^2 - 2)\cos(x)$
$\int e^{\alpha x} dx$	$\dfrac{e^{\alpha x}}{a}$

Daftar Pustaka

http://nic.unud.ac.id/~wiharta/siskom_1/BAB II MODULASI AMPLITUDO.pdf
http://nic.unud.ac.id/~wiharta/siskom_1/BAB III MODULASI SUDUT.pdf
homes.kuleuven.be/sista/~moonen/

Coughlin, Driscoll. `Penguat Operasional dan Rangkaian Terpadu Linier`. Erlangga. Jakarta 1983.

Haykin, Simon. `Signals and Systems`. Wiley. Singapore. 2004.

Saydam, Gouzali.`Teknologi Telekomunikasi`. ALFABETA. Bandung. 2005.