

APLIKASI INTERFACE SPTOOLS MATLAB DALAM MENDESAIN IIR DAN FIR DIGITAL FILTER

Khoe Yao Tung

Fakultas Teknik Jurusan Elektro, Universitas Kristen Krida Wacana
Tanjung Duren 4 – Jakarta 14770
E-mail : yaotung@satumail.com

Abstract

This paper discusses examples of lowpass, highpass and bandpass filter digital designs by using a variety of filter design algorithm are available in Matlab for both IIR and FIR filter. Matlab provides different options for digital filter design, which include function call to filter algorithm and a graphical user interface called Sptools Interfacing. Interface Sptools (Signal processing tools) has more user friendly environment, more quicker display and simpler option than the option of making calls to the filter algorithms. However, the shortcoming of sptools is that it only displays magnitude response of the filter, not the phase response.

1. Pendahuluan

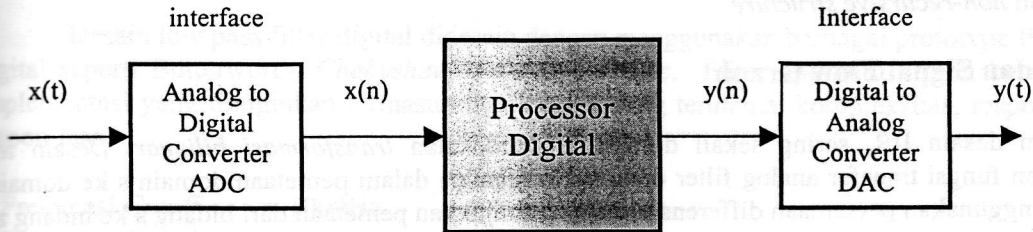
Aplikasi proses sinyal digital yang teresenting adalah filter. Filter berkaitan langsung dengan manipulasi spektrum sinyal, Untuk membangun sebuah filter digital dibutuhkan tiga komponen utama yaitu: *adder* (penambah), *multiplier* (pengali) dan *delay* (penunda). Penambah mempunyai dua input dan satu output yang hasilnya menambahkan masukan dari kedua input tersebut. Pengali adalah elemen penguat dan akan mengalikan sinyal input dengan suatu besaran konstanta tertentu. Penunda akan menunda satu cuplikan yang masuk.

Pengolahan sinyal digital menggunakan transformasi diskrit, transformasi yang sering digunakan adalah transformasi z yang merupakan prosedur deret sinyal masukan $x(n)$ menjadi deret sinyal keluaran $y(n)$. Filter digital bekerja berdasarkan data masukan diskrit dari cuplikan-cuplikan sinyal kontinu, yang kemudian diubah oleh konverter analog ke digital menjadi data digital biner, data digital inilah yang nanti dapat dimanipulasi kinerja dan spektrum sinyalnya dengan prosesor digital. Hasil dari data digital ini dikembalikan ke dalam bentuk analog bila diinginkan dengan konverter *digital to analog*. Penerapannya filter digital pada pengolahan sinyal dapat digunakan dalam *noise reduction*, *image processing*, *antialiasing* dan menghilangkan *pseudoimages* pada *multirate processing*, *matched filtering*, *osilator digital*.

Gambar 2. Block Diagram dan Signal Flow Graph dari tiga elemen utama filter digital

2. Desain IIR dan FIR

Terdapat dua jenis filter digital yang disebabkan karena adanya perbedaan penggunaan feedback dalam mendesain suatu filter digital. Yang pertama jenis IIR (*Infinite Impulse Response*) menggunakan struktur *feedback* yang sering juga disebut dengan *recursive structure*, sedangkan



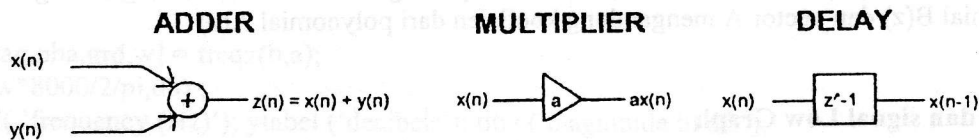
Gambar 1. Proses pengolahan dalam filter digital, dengan pengubahan

Beberapa keunggulan dari filter digital setelah melalui proses pengolahan sinyal adalah

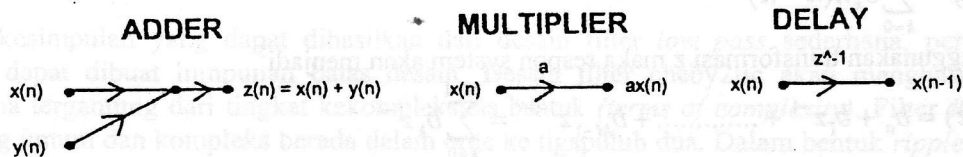
- Pengaturan frekuensi cuplikan sehingga daerah kerja yang dapat dipilih sangat lebar (meliputi frekuensi rendah dan frekuensi tinggi). Termasuk frekuensi sangat rendah sehingga dapat digunakan untuk aplikasi elektronika biomedis.
- Respon fasa yang benar-benar linear.
- Karena menggunakan *programmable processor*, respon frekuensi dapat dipilih secara langsung secara otomatis.
- Beberapa sinyal masukan dapat disimpan untuk keperluan selanjutnya.
- Berkembangnya *teknologi pico* memungkinkan penggunaan *hardware* yang lebih kecil, konsumsi daya yang kecil, menekan biaya produksi, dan *single chip*.

Dalam implementasi filter digital kita dapat menggunakan *block diagram* atau *signal flow graph*. Seperti yang nampak dalam gambar di bawah ini

Block Diagram of Filter Elements



Signal Flow Graph of Filter Elements



Gambar 2. Block Diagram dan Signal Flow Graph dari tiga elemen utama filter digital

2. Desain IIR dan FIR

Terdapat dua jenis filter digital yang disebabkan karena adanya perbedaan penggunaan *feedback* dalam mendesain suatu filter digital. Yang pertama Jenis IIR (*Infinite Impulse Response*) yang menggunakan struktur *feedback* yang sering juga disebut dengan *recursive structure*, sedangkan

yang kedua jenis FIR (*Finite Impulse Response*) yang tidak menggunakan *feedback* yang sering disebut dengan *non-recursive structure*.

2.1 FIR dan Signal Flow Graph

Dalam desain IIR, sering sekali digunakan pendekatan *transformasi bilinear*. Desain ini dimulai dengan fungsi transfer analog filter dan menyajikannya dalam pemetaan domain s ke domain z . Dengan menggunakan persamaan differensial dapat ditunjukkan pemetaan dari bidang s ke bidang z . Dengan transformasi dikenal dengan *Tustin's bilinear transformation*²

$$s = \frac{J}{2} \left(\frac{1+z^{-1}}{1-z^{-1}} \right)$$

Pemetaan ini hasil dari bentuk umum filter IIR dengan sejumlah pole dan zero sembarang. Respon sistem dan persamaan differensial dari filter ini akan mengikuti perumusan sebagai berikut, dengan menggunakan *operator shift time* dari z maka filter dari H akan dapat direpresentasikan dengan fungsi rasional³:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{n=0}^M b_n z^{-n}}{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}} = \frac{1 + a_1 z^{-1} + \dots + a_N z^{-N}}{\sum_{n=0}^N a_n z^{-n}}$$

$$y(n) = \sum_{m=0}^M b^m x(n-m) - \sum_{n=N}^N a^n y(n-m)$$

dalam implementasi sptools Matlab dapat digantikan dengan fungsi *filter(B,A,x)* yang artinya melakukan implementasi $H(z)=B(z)/A(z)$ dari input signal x . Vektor B mengandung koefisien dari polynomial $B(z)$ dan vector A mengandung koefisien dari polynomial $A(z)$

2.2 FIR dan signal Low Graph

Dan filter dari FIR akan memiliki *difference equation*⁴ dari

$$y(n) = \sum_{k=0}^{M-1} b^k x(n-k)$$

Dengan menggunakan transformasi z maka respon system akan menjadi⁵

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_{M-1} z^{-M-1} = \sum_{k=0}^{M-1} b^k z^{-k}$$

¹ Proakis, John G. and Manolakis, Dimitris G. *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd Edition, Prentice Hall, Upper River, NL, 1996, p. 677
² Brian Adtran, Breiner Moshes, *MATLAB for engineers*, Addison Wesley, Wokingham, England, 1995, p.575
³ Ingle, Vinay K. and Proakis, John G. *Digital Signal Processing Using Matlab*, PWS Publishing Company, 1997, p.183
⁴ Proakis, John G. and Manolakis, Dimitris G. *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd Edition, Prentice Hall, Upper River, NL, 1996, p. 620
⁵ Ingle, Vinay K. and Proakis, John G. *Digital Signal Processing Using Matlab*, PWS Publishing Company, 1997, p.197

3. Desain Low Pass Filter

Desain low pass filter digital didesain dengan menggunakan berbagai prototype tiga jenis filter digital seperti Butterworth, *Chebyshev tipe 1* dan *elliptic*. Filter optimum dipilih berdasarkan dari implementasi yang diinginkan termasuk tiga kriteria yang terutama: kompleksitas, respon besaran dan respon fasa.

- Frekuensi *Cutoff* = 1000Hz
- Frekuensi *Sample* = 8000Hz
- Ripple passband = 0,5dB
- Stopband attenuasi = 60dB
- Band Transisi = 100Hz

Matlab Code (Chebyshev):

```
% Lowpass digital filter with chebyshev-I analog prototype
%
% Digital Filter specifications:
wp = 0.125*2*pi; % digital passband frequency in Hz (normalized)
ws = 0.1375*2*pi; % digital stopband frequency in Hz (normalized)
Rs = 0.5; % passband ripple in dB
As = 20; % stopband attenuation in dB
% Analog Prototype Specifications:
Fs = 1; T = 1/Fs;
OmegaP = (2/T)*tan(wp/2); % prewarped prototype passband frequency
OmegaS = (2/T)*tan(ws/2); % prewarped prototype stopband frequency
% Analog Chebyshev-1 Prototype Filter Calculation:
[c,d] = cheby1(OmegaP,OmegaS,Rs,As,'stop');
% Bilinear Transformation:
[b,a] = bilinear(cs,ds,Fs);
%
[db,mag,pha,grd,w] = freqz(b,a);
plot(w*8000/2/pi,db);
xlabel('frequency (Hz)'); ylabel('decibels'); title('magnitude in dB');
```

Mathlab juga menyajikan tampilan orde filter dan koefisien filter. Dalam contoh ini, filter Chebyshev dapat ber-orde sembilan. *elliptic filter* memiliki lima orde dan filter butterworth tiga puluh dua orde.

Beberapa kesimpulan yang dapat dihasilkan dari desain filter *low pass* sederhana, pertama pada umumnya dapat dibuat himpunan batas desain. Desain filter cheby2tic akan menghasilkan filter tersederhana tergantung dari tingkat kekompleksitas bentuk (*terms of complexity*). Filter Butterworth yang paling umum dan kompleks berada dalam orde ke tigapuluh dua. Dalam bentuk *ripple passband*, filter *butterworth* memberikan respon optimum. Dalam passband, hampir tidak terdapat *ripple* hanya sebagai monotonik. *elliptic* dan *chebyshev* keduanya mempunyai lebih banyak *ripple* dalam passband. Inilah yang menjadi pilihan dari filter-filter yang digunakan. Dalam hal respon besaran dan kompleksitas, *filter elliptic* akan menjadi pilihan yang terutama, akan tetapi *ripple elliptic* mempunyai respon fasa yang lebih *non linear* dibandingkan filter Chebyshev dan Butterworth. Oleh karenanya ketajaman *cutoff* dengan kompleksitas rendah harus jatuh pada filter *elliptic*. Jika respon fasa yang dibutuhkan linear, maka filter Chebyshev atau Butterworth akan menjadi pilihan.

Filter IIR didesain dengan menggunakan rutin yang dibuat oleh MH Akroyd's dalam publikasinya tentang Butterworths *digital filter*. Filter *Butterworth* digunakan secara maksimal dalam

periode keatarannya dalam passband, kondisinya pada monoton *cut-off*. Filter Butterworth dalam kondisi ternormalisasi dapat diimplementasikan sampai dengan derajat ke enam. Chebyshev tipe I memiliki *ripple* pada *passband*, monotonic *cut-off* dari frekuensi *cut-off* pada stopband. Pada implementasi derajat ke enam dengan *ripple* sebesar 1 dB (maksimum deviasi dari normal sebesar kurang lebih satu dB bukan satu dB dari besaran amplituda *peak to peak*)
 Chebyshev tipe II memiliki flat maksimal dari passband dan *ripple* dalam stopband. Maksimum *ripple* 40 dB dari stopband dan memiliki transisi 3% dari frekuensi *cut-off*.

4. Desain High Pass Filter

Matlab menyediakan fungsi implementasi untuk konversi *low pass* ke *high pass* dan konversi *low pass* ke *band pass*. Dengan menyediakan perangkat orde untuk filter, passband ripple dan frekuensi *cut-off* 3 dB ke fungsi *cheby1()*, sebuah filter *high pass* dapat didesain. Orde filter akan ditemukan dengan menggunakan fungsi *chebord()*. Untuk prototipe Butterworth, fungsi dari *butter()* dan *butterd()*. Untuk prototipe dari *cheby2()* kita menggunakan *cheby2()* dan *cheby2ord()*
 Di bawah ini adalah kode program Matlab untuk desain Chebyshev highpass digital filter dengan band transisi passband dengan frekuensi 1100hz.

```
% highpass Chebyshev Digital Filter
ws = [0.125 * 2*pi]; % digital stopband frequency in rad/s
wp = [0.1375 * 2*pi]; % digital passband frequency in rad/s
Rs = 0.5; % passband ripple in dB
As = 20; % stopband attenuation in dB
[N, wn] = cheblord(wp/pi, ws/pi, Rs, As);
[b, a] = cheby1(N, Rs, wn, 'high');
[db, mag, pha, grd, w] = freqz_m(b, a);
plot(w*8000/2/pi, db);
axis ([800 1200 -22 1]);
```

Band pass filter ditentukan dengan menggunakan dua fungsi. Untuk filter bandpass frekuensi ditentukan dengan *wp* dan *ws* yaitu frekuensi passband dan frekuensi stopband yang keduanya merupakan dua elemen vektor yang dapat menyebabkan dua frekuensi passband dan dua frekuensi stopband, Matlab di bawah ini mendesain filter bandpass digital cheby2tik

```
% Bandpass Elliptic Digital Filter
ws = [0.3*pi 0.75*pi]; %Stopband edge frequency
ws = [0.4*pi 0.6*pi]; %Passband edge frequency
Rs = 0.5; %Passband ripple in dB
As = 20; %Stopband attenuation in dB
[N, wn] = cheby2ord(wp/pi, ws/pi, Rs, As);
[b, a] = cheby2(N, Rs, As, 'wn');
```

```
[db, mag, pha, grd, w] = freqz_m(b, a);
plot(w*8000/2/pi, db);
axis ([5000 3500 -22 1]);
xlabel('Frequency (Hz)'); ylabel('decibels'); title('magnitude in db Respons of cheby2tic Filter');
```

5. Signal Processing Tools

Signal processing tool atau Sptools pada Matlab digunakan sebagai alat visualisasi untuk desain dan menganalisis filter digital. Sptool merupakan *graphical user interface* yang mampu menganalisis dan memanipulasi signal, filter dan spektrum frekuensi. Desain filter dengan Sptools memperbolehkan pengguna untuk memilih algoritma desain filter yang digunakan ketika membuat filter. Algoritma desain ini diantaranya terbagi untuk filter FIR dan filter IIR. Untuk filter FIR terbagi dalam *equiripple*, *least squares*, *Kaiser Windows* dan untuk filter IIR adalah *butterworth*, *chebyshev* tipe 1 dan 2, serta *elliptic*. Pengguna juga diperbolehkan menspesifikasikan filter *lowpass*, *bandpass*, *high pass* atau *bandstop*. Desain dari sptools filter akan ditampilkan dalam respon besaran dan tampilan orde filter.

Berikut ini adalah contoh besaran respon dari bandpass filter yang didesain dalam kode program Matlab.

Filter 4

Appendix

```

Function [b, a] = chb1 {Wp, Ws, Rs, As) ;
% Analog Lowpass Filter Design: Chebyshev-1
%
% [b, a] = chb1 (Wp, Ws, Rs, As) ;
% b = Numerator coefficients of Ha (s)
% a = Denominator coefficients of Ha (s)
% Wp = Passband edge frequency in rad/sec
% Ws = Stopband edge frequency in rad/sec
% Rs = Passband ripple in dB
% As = stopband attenuation in dB
%
if wp <= 0
    error ('Passband edge must be larger than 0 ')
end
if Ws <= Wp
    error ('Stopband edge must be larger than Passband edge ')
end
if (Rs <= 0) | (As < 0)
    error ('PB ripple and/or SB attenuation must be larger than 0 ')
end
ep = sqrt (10^(Rs/10)-1) ;
A = 10^(As/20) ;
OmegaC = Wp;
OmegaR = Ws/Wp;
g = sqrt (A*A -1) /ep;
N = ceil (log10 (g+sqrt (g*g-1) ) /log10 (OmegaR+sqrt (omegar*Aomegar-1) ) ) ;
fprintf ('\n*** Chebyshev-1 Filter Order = %2.Of \n',N) ;
[b, a] = ap_chb1(N, Rs, OmegaC) ;

function [b, a] = ap_chb1(N, Rs, OmegaC) ;
% Chebyshev-1 Analog Lowpass Filter Prototype
%
```

```

% [b, a] = ap_cheb1(N, Rs, OmegaC) ;
% b = numerator polynomial coefficients
% a = denominator polynomial coefficients
% N = Order of the cheby2ical filter
% Rs = passband Ripple in dB
% OmegaC = Cutoff frequency in rad/sec
%
[z, p, k] = chebap (N, Rs) ;
a = real (poly (p)) ;
aN = a (N+1) ;
p = p*omegaC;
a = real (poly (p)) ;
aNu = a (N+1) ;
k = k*aNu/aNn;
b0 = k;
B = real (poly (z)) ;
b = k*B;
    
```

6. Argumen Sptools

Untuk membuat filter butterworth di bawah ini adalah fungsi-fungsi yang disediakan oleh sptools Matlab untuk memanipulasi desain digital filter butterworth dimana

$$H(z) = B(z)/A(z)$$

[B,A] = butter[N,Wn] Menghitung koefisien dari low pass filter. Wn adalah frekuensi

cutoff dalam frekuensi ternormalisasi.

Menghitung koefisien dari highpass butterworth filter. Wn

adalah frekuensi cutoff dalam frekuensi ternormalisasi

Menghitung koefisien dari bandpass filter. Wn adalah vector

yang berisi dari dua frekuensi cutoff ternormalisasi dari passband

dalam urutan naik

Menghitung koefisien dari bandstop butterworth filter, Wn

adalah vector yang mengandung dua frekuensi cutoff

ternormalisasi dalam stopband dalam urutan naik.

Untuk desain dari chebyshev tipe I kita menggunakan variasi dari cheby1 dengan fungsi komputasi

$$\text{dari koefisien filter order keN dengan filter } H(z) = B(z)/A(z)$$

[B,A] = cheby1[N,Rp,Wn] Menghitung koefisien dari low pass filter Chebyshev tipe I. Rp

mewakili ripple passband dan Wn adalah frekuensi cutoff dalam

frekuensi ternormalisasi.

Menghitung koefisien dari highpass chebyshev tipe I filter. Rp

mewakili ripple passband dan Wn adalah frekuensi cutoff dalam

frekuensi ternormalisasi

Menghitung koefisien dari bandpass ripple, dengan Rp mewakili

passband ripple dan Wn adalah vector yang berisi dari dua

frekuensi cutoff ternormalisasi dari passband dalam urutan naik

Menghitung koefisien dari bandstop chebyshev tipe filter,

dimana Rp mewakili dari passband ripple. Wn adalah vector

[B,A] = cheby1(N,Rp,Wn,'stop')

[B,A] = cheby1(N,Rp,Wn)

[B,A] = cheby1(N,Rp,Wn,'high')

[B,A] = cheby1[N,Rp,Wn]

yang mengandung dua frekuensi *cutoff* ternormalisasi dalam stopband dalam urutan naik.

Untuk desain dari chebyshev tipe II kita menggunakan variasi dari cheby2 dengan fungsi komputasi dari koefisien filter order keN dengan filter $H(z)=B(z)/A(z)$

- [B,A] = cheby2(N,Rs,Wn) Menghitung koefisien dari low pass filter Chebyshev tipe 2. Rs mewakili stopband *ripple* dan Wn adalah frekuensi *cutoff* dalam frekuensi ternormalisasi.
- [B,A] = cheby2(N,Rs,Wn,'high') Menghitung koefisien dari highpass chebyshev tipe 2 filter. Rs mewakili ripple stopband dan Wn adalah frekuensi cutoff dalam frekuensi ternormalisasi
- [B,A] = cheby2(N,Rs,Wn) Menghitung koefisien dari bandpass dengan Ws adalah stopband ripple dan Wn adalah vector yang berisi dari dua frekuensi cutoff ternormalisasi dari passband dalam urutan naik
- [B,A] = cheby2(N,Rs,Wn,'stop') Menghitung koefisien dari bandstop chebyshev tipe filter, dimana Rs mewakili dari stopband ripple. Wn adalah vektor yang mengandung dua frekuensi *cutoff* ternormalisasi dalam stopband dalam urutan naik.

Untuk desain dari chebyshev tipe II kita menggunakan variasi dari cheby2 dengan fungsi komputasi dari koefisien filter order keN dengan filter $H(z)=B(z)/A(z)$

- [B,A] = ellip(N,Rp, Rs, Wn) Menghitung koefisien dari low pass filter Chebyshev tipe 2. Rp mewakili ripple passband, Rs mewakili stopband *ripple* dan Wn adalah frekuensi *cutoff* dalam frekuensi ternormalisasi.
- [B,A] = ellip(N,Rp,Rs,Wn,'high') Menghitung koefisien dari highpass chebyshev tipe 2 filter. Rp mewakili ripple passband, Rs mewakili ripple stopband dan Wn adalah frekuensi cutoff dalam frekuensi ternormalisasi
- [B,A] = ellip(N,Rp,Rs,Wn) Menghitung koefisien dari bandpass dengan Rp mewakili ripple passband dengan Ws adalah stopband ripple dan Wn adalah vector yang berisi dari dua frekuensi cutoff ternormalisasi dari passband dalam urutan naik
- [B,A] = ellip(N,Rp,Rs,Wn,'stop') Menghitung koefisien dari bandstop chebyshev tipe filter, dimana Rp mewakili ripple passband, Rs mewakili dari spassband ripple. Wn adalah vektor yang mengandung dua frekuensi *cutoff* ternormalisasi dalam stopband dalam urutan naik.

7. Daftar Pustaka

- [1] A.V. Oppenheim and R.W. Schaffer, "Digital Signal Processing", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.
- [2] Biran Adrian, Breiner Moshe; *MATLAB for engineers*, Adison wesley, Wokingham, England, 1995.
- [3] L.R. Rabiner & R.W. Schaffer, "Digital Processing of Speech Signals", Prentice Hall, 1978, ISBN 0-13-213603-1.
- [4] R. E. Crochiere & L. R. Rabiner, "Multirate Digital Signal Processing", Prentice-Hall, 1983, ISBN 0136051626.

- [5] Ingle, Vinay K. and Proakis, John G. *Digital Signal Processing Using Matlab*. PWS Publishing Company, 1997.
- [6] Proakis, John G. and Manolakis, Dimitris G. *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd Edition. Prentice Hall. Upper River, NJ, 1996.
- [7] Porat Boaz, "A course in digital signal processing", John Wiley & Sons, Inc New York 1997.
- [8] Ziemer, Rodger E., Tranter, William H., and Fanning, D. Ronald. *Signal and Systems: Continuous and Discrete*, 3rd Edition. Macmillan Publishing Company, 1993.
- [9] Eter, Delores M., *Engineering Problem Solving with Matlab*, second edition, Prentice Hall, New Jersey 1997.
- [10] http://dolphin.wmin.ac.uk/filter_design.html
- [11] <http://www.isip.msstate.edu>
- [12] <http://www.dsp.rice>