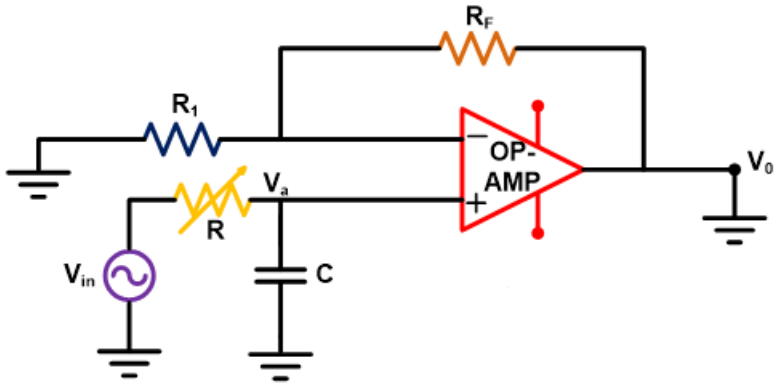
Filtrare Analogica cu Filtre Butterworth

Necula Leonard-Gabriel

Ianuarie 2021

1 Introducere

Filtrul **Butterworth** este unul dintre cele mai utilizate filtre în procesarea semnalelor analogice. Pentru a realiza un filtru Butterworth de ordinul 1 avem nevoie de câteva rezistenţe, un condensator şi un amplificator operaţional.



Pentru filtrul de mai sus putem să definim urmatoarele elemente caracteristice:

- A_f - Amplificarea in banda de trecere
- f_c - Frecventa de tăiere a filtrului

$$A_f = 1 + \frac{R_F}{R_1} \tag{1}$$

$$f_c = \frac{1}{2\pi RC} \tag{2}$$

2 Funcţia de Transfer unui FTJ Butterworth

După cum aţi văzut şi la curs un FTJ de ordin n de tip Butterworth cu pulsaţa de tăiere egală cu 1 rad/s are drept poli radacinile de ordin n ale unitaţii. Forma complexă a acestui filtru nu este totuşi cea mai utilizată în implementarile numerice pentru testare. Din acest motiv o să utilizăm forma reală a acestui filtru.

Forma reală a numitorilor funcţiei de transfer în funcţie de paritatea ordinului filtrului:

$$B_n(s) = \prod_{k=1}^{\frac{n}{2}} \left[s^2 - 2s \cdot cos\left(\frac{2k+n-1}{2n}\pi \right) + 1 \right] \quad n = par \tag{3}$$

$$B_n(s) = (s+1)\prod_{k=1}^{\frac{n}{2}}\left[s^2 - 2s\cdot cos\left(\frac{2k+n-1}{2n}\pi\right)+1\right] \quad n = impar \tag{4}$$

Filtele definite cu ajutorul numitorilor prezentaţi mai sus au pulsaţia de tăiere $\omega_c = 1$. Pentru a modifica această pulsaţie de tăiere trebuie să utilizăm următoarea schimbare de variabilă: $s = \frac{s}{w_{c_{nou}}}$, unde $w_{c_{nou}}$ reprezintă noua pulsaţie de tăiere a filtrului.

3 Transformări ale unui FTJ într-un FTS, FTB, FSB

Transformarea unui FTJ de tip Butterworth într-un altfel de filtru se face asemănător cu modul de schimbare a pulsaţiei de tăiere, deci tot ce trebuie să facem este să realizăm o schimbare de variabilă.

1. FTJ $\rightarrow$ FTS
$$s = \frac{\omega_c}{s} \tag{5}$$
2. FTJ $\rightarrow$ FTB
$$s = \frac{s^2 + \omega_{c_1}\omega_{c_2}}{(\omega_{c_2} - \omega_{c_1})\, s}, \qquad \omega_{c_2} > \omega_{c_1} \tag{6}$$
3. FTJ $\rightarrow$ FSB
$$s = \frac{(\omega_{c_2} - \omega_{c_1})\, s}{s^2 + \omega_{c_1}\omega_{c_2}}, \qquad \omega_{c_2} > \omega_{c_1} \tag{7}$$

4 Anexă cod Matlab

```
1
2 %% Designing a Butterworth filter
3
4 % If we set wc to be exactly the frequency of the first sinusoid we'll
5 % still modify it's value, we have to keep in mind the -3dB
6 order = [2, 3, 2, 3];
7 wc = [2 350 5 0.05]; % cutoff freq's
8 wc_band = wc * 100;
9
10 % Create an input
11 % Time domain
12 t = 0: 0.01: 180;
13 % Frequency vector
14 w = [1, 10, 100, 200, 300, 400]';
15 arg = w * t;
16 % Matrix with components on rows
17 y = sin(arg);
18 % Adding all the components, result should be a row vector with length(t)
19 % elements
20 y = sum(y);
21
22 % Checking the filter response
23
24 y = y(:);
25 H = cell(numel(order), 1);
26 y_out = cell(numel(order),1);
27 line = factor(numel(order));
28 col = line(1);
29 line = prod(line(2:end));
30
31 type = {'lowpass', 'highpass', 'stopband', 'passband'};
32 fig_time_response = figure();
33 fig_freq_filter_response = figure();
34
35 for i = 1:numel(order)
36
37     % Design a butterworth filter
38     [H{i}, ~] = designButter(order(i), type{i}, wc(i), wc_band(i));
39
40     % Subplot setup
41     ax = subplot(line, col, i, 'Parent', fig_time_response);
42     ax_freq = subplot(line, col, i, 'Parent', fig_freq_filter_response);
43
44
45     % Frequency response
46     [amp, phase, w] = bode(H{i});
47     % Vectorize bode outputs
48     amp = amp(:);
49     phase = phase(:);
50     semilogx(ax_freq, w, 20 * log10(abs(amp)), 'Color', 'm', 'LineWidth', 3)
51     grid(ax_freq, 'on');
52     xlabel('Frequency');
53     ylabel('dB Magnitude');
54
55
```

```
    % The response to a sum o sinusoids
    y_out{i} = lsim(H{i}, y, t);
    y_out{i} = y_out{i}(:);
    plot(ax, t, y, 'LineWidth', 0.5, 'Color', 'c');
    hold(ax, 'on');
    plot(ax, t, y_out{i});
    legend(ax, 'Input Signal', 'Filtered Data');
    xlabel(ax, 'Time');
    ylabel(ax, 'Signal Amplitude');


    % Setting up titles
    if checkType(type{i}, [3, 4])
        title(ax, [upper(type{i}(1)) type{i}(2:end)  ' filter of order '....
            num2str(order(i)) ' with wc_{start} = ' num2str(wc(i)) ....
            ' and wc_{stop} = ' num2str(wc_band(i)) ' time response']);

        title(ax_freq, [upper(type{i}(1)) type{i}(2:end)  ' filter of order '....
            num2str(order(i)) ' with wc_{start} = ' num2str(wc(i)) ....
            ' and wc_{stop} = ' num2str(wc_band(i)) ' frequency response']);
    else
         title(ax, [upper(type{i}(1)) type{i}(2:end)  ' filter of order '....
            num2str(order(i)) ' with wc = ' num2str(wc(i)) ' time response']);

        title(ax_freq, [upper(type{i}(1)) type{i}(2:end)  ' filter of order '....
            num2str(order(i)) ' with wc = ' num2str(wc(i)) ' frequency response']);
    end


end
```