Abstract

This paper describes new proposed algorithms for constructing the transfer function of nth order Butterworth digital LPF using the idea of the cascade combination of second order and first order digital filters. Computer-aided design is used with new algorithms to achieve z-domain transfer function (for conjugate poles and single real pole). C++ program is used for designing procedure by linear programming. The proposed algorithms are very fast, flexible, and exact. The program allows users to design different (sampling, cut-off) frequencies, and order of Butterworth digital LPF with high flexibility.

Keywords: Programming algorithms, Digital filter design, IIR filter.

1. Introduction

This paper introduces new proposed algorithms for nth order Butterworth digital LPF using the idea of cascade combination of digital filters. This work is to simplify digital filter design which can be used for printed digital circuits or integrated circuits easily and with high flexibility of using (sampling, cut-off) frequencies and changing the order easily as we want. Designing an IIR filter involves choosing the coefficients to satisfy a given specification, usually a magnitude response specification.

Digital filtering is one of the most powerful tools of DSP. Apart from the obvious advantages of virtually eliminating errors in the filter associated with passive component fluctuations over time and temperature. A digital filter can be defined as a discrete-time, discrete-amplitude convolving machine. Filtering is in essence the multiplication of the signal spectrum by the frequency domain impulse response of the filter [1].

Detection of a wanted signal may be impossible if unwanted signals and noise are not removed sufficiently by filtering. Electronic filters allow some signals to pass, but stop others. To be more precise, filters allow some signal frequencies applied at their input terminals to pass through to their output terminals with little or no reduction in signal level [2].

A digital filter uses a digital processor to perform its operation. A digital filter is programmable (its operation is determined by a program stored in the processor’s memory). This means the digital filter can be easily changed without affecting the circuitry (hardware) [3].

Chen [1] has provided a lot of information in filters with significant additions in the areas of computer-aided design of digital filters.

Thede [3] has written programs in C Language for IIR filter design. He has used bilinear transform to convert transfer function from s-domain to z-domain.
Winder [4] has written simple programs for filter design like “Super FILTER” and “Filter Master”. He has written “Digital F” as an digital filter design program.

2. Butterworth Low-Pass Filters

A filter which approximates the ideal low-pass filter with a relatively flat pass band characteristic is Butterworth filter. Its amplitude response [4] is:

\[ |H(j\omega)| = \frac{k}{\sqrt{1+(\omega/\omega_c)^{2n}}} \]

Where \( n \) is the order of the filter.

It may be seen that the filter is improved as \( n \) increases. The Butterworth filter has the advantage of maximally flat response in the pass band. The poles of the \( n \)th order Butterworth filter can be calculated [2] by:

\[ p_k = -\sin\left(\frac{(2k-1)\pi}{2n}\right) + j\cos\left(\frac{(2k-1)\pi}{2n}\right) \]

For \( k=1, 2, ..., n \)

3. Converting the Transfer Function from s-domain to z-domain

For the first order transfer function [4]:

\[ H(s) = \frac{1}{s+a} \]

\[ \Rightarrow H(z) = \frac{1}{1-e^{-aT}z^{-1}} \]

Where \( T \) is the sampling period.

The second order transfer function (with conjugate poles) [5]:

\[ H(s) = \frac{1}{(s+\alpha+j\beta)(s+\alpha-j\beta)} \]

\[ \Rightarrow H(s) = \frac{j/2\beta}{s+\alpha+j\beta} - \frac{j/2\beta}{s+\alpha-j\beta} \]

Now using the transformation of Equation (3) to Equation(4) in Equation(6), we get:

\[ H(z) = \frac{j/2\beta}{1-e^{-(\alpha+j\beta)T}z^{-1}} - \frac{j/2\beta}{1-e^{-(\alpha-j\beta)T}z^{-1}} \]
Then that leads to:

\[ H(z) = \frac{e^{-\alpha T} \sin(\beta T) z^{-1}/\beta}{1 - 2e^{-\alpha T} \cos(\beta T) z^{-1} + e^{-2\alpha T} z^{-2}} \]  

(8)

4. General Circuits and Equations

If we make a general IIR circuit for the following first order transfer function:

\[ H(s) = \frac{1}{s + a} \Rightarrow H(z) = \frac{1}{1 - e^{-\alpha T} z^{-1}} \]  

(9)

Now let \( b = e^{-\alpha T} \)  

(10)

Then \( H(z) = \frac{1}{1 - b z^{-1}} \)  

(11)

Then the IIR circuit for above Equation is:

Figure (1): First order IIR circuit

Now the general IIR circuit for the following second order transfer function [1]:

\[ H(z) = \frac{a_1 z^{-1}}{1 - b_1 z^{-1} - b_2 z^{-2}} \]  

(12)

Figure (2): Second order IIR circuit for Equation (12).
Now by comparing Equation (8) with Equation (12), we get:

\[ a_1 = \frac{e^{-aT} \sin(\beta T)}{\beta} \]  \hspace{1cm} (13)

\[ b_1 = 2e^{-aT}\cos(\beta T) \]  \hspace{1cm} (14)

\[ b_2 = -e^{-2aT} \]  \hspace{1cm} (15)

5. Design Principles

The nth order Butterworth digital filter can be implemented using multistage circuit. Each two conjugate poles represent a stage. If the order is odd, then there will be a real pole \( s = -1 \) which can be represented like circuit of Figure (1) with:

\[ b = e^{-aT} = e^{-T} \]  \hspace{1cm} (14)

For example if the designed filter is 7th order then the stages will be:

\[ \text{Figure (3) Stages of 7th Order Digital LPF} \]

6. Realization of the Prototype Filter

The designed filter for the calculated poles is prototype filter with cut-off frequency \( w_c \) equals to 1rad./sec. Here the equations can be transformed to real cut-off frequency poles by changing each \( s \) in \( H(s) \) to \( (s/w_c) \). That \( w_c \) is the cut-off frequency [3,6,7,8].

7. Frequency Transformation

LPF is discussed in this paper while other types also can be designed easily by simple frequency transformation to get: (High-pass, band-pass and band-stop) [2,5].

8. Design Algorithms

Flowchart (1) shows the algorithms of nth order digital filter which gives flexibility to choose (sampling, cut-off) frequencies and the order of this digital filter.
Flowchart(1): The algorithms of nth order digital filter

Start

Input Order (n), cut-off frequency (f_c), Sampling frequency (f_s).

Yes

Is \( f_s < 2f_c \) ?

No

k = 1

Calculate \( \alpha, \beta \) for kth conjugate poles.

k = k+1

Calculate the Coefficients of Second Order IIR circuit for the kth Stage

Print the results of kth stage.

Yes

k \geq n/2 ?

No

Calculate the Coefficients of First Order IIR Circuit

Print the Results of First Order Stage

Yes

Integer(n/2)=n/2 ?

No

Stop

Print "Change f_s, or f_c to avoid aliasing error using the condition f_s \geq 2f_c"
9. Design Example Results

Example results of the program (in C++ Language) which follows the algorithms of Flowchart (1) are as follows:

- For third order with sampling freq.=100kHz and with selected cut-off frequency $f_c=26kHz$
  - The first stage: $b_1=0.703429$, $b_2=5.122426$, $a_1=2.581659$
  - The second stage: $b_1=0.19522$
  - Fig.(4) shows above third order filter

![Third order LPF](image)

- For sixth order with sampling freq.=100kHz and with selected cut-off frequency $f_c=26kHz$
  - The first stage: $b_1=0.021878$, $b_2=2.329441$, $a_1=1.58005$
  - The second stage: $b_1=2.563609$, $b_2=10.077439$, $a_1=4.107168$
  - The third stage: $b_1=8.836823$, $b_2=23.474795$, $a_1=7.681326$
  - Fig.(5) shows above sixth order filter

![Sixth order LPF](image)

- For tenth order with sampling freq.=100kHz and with selected cut-off frequency $f_c=26kHz$
  - The first stage: $b_1=0.110283$, $b_2=1.667143$, $a_1=1.306081$
  - The second stage: $b_1=0.482728$, $b_2=4.407481$, $a_1=2.340588$
  - The third stage: $b_1=2.563609$, $b_2=10.077439$, $a_1=4.107168$
  - The fourth stage: $b_1=6.321985$, $b_2=18.377861$, $a_1=6.378674$
  - The fifth stage: $b_1=9.714762$, $b_2=25.204713$, $a_1=8.112535$
  - Fig.(6) shows above tenth order filter
Fig. (6) Tenth order LPF

10. Conclusions

The program is very simple to use and does not require knowledge in filtering circuits design in order to run it. The performance of each designed filter was evaluated using “Electronic Workbench 10”, which resulted that the design is exact and convenient for any particular application. The main outputs from this program are the coefficients of each stage of the filter.

11. References


