Frequency Transformation in Digital Domain

Phuoc S. Nguyen
Department of Communications and Informatics, Victoria University of Technology, Melbourne, Australia
Email: nguyensiphuoc@frequencytransform.com

Abstract—This paper introduces a new algorithm to transform a digital low-pass filter into a digital low-pass, high-pass, band-pass, band-stop and narrow-band filter in the digital domain. The technique of this method is based on bilinear and inverse bilinear z-transformation with pre-warping frequency and the support of Pascal’s triangle. From this, a matrix equation is derived, called ‘Digital low-pass to digital filter Pascal matrix equation’. This equation will make it easier for hand calculations and programming when transformations are made from a digital low-pass filter to another digital filter in the digital domain.

Index Terms—frequency transformation, bilinear z-transformation, digital filters, analog filters, pre-warping frequency, Pascal’s triangle.

I. INTRODUCTION

One method for designing a digital filter is to transform a given analog low-pass prototype into a desired digital filter by using frequency transformation in the s-domain and bilinear z-transformation [1], [2]. The main advantage of this method is stability from mapping the poles and zeros from the stable region in the s-domain into a stable region in the z-domain. However, when applying bilinear z-transformation, the warping frequency will give a non-linear relation between the analog frequency and the digital frequency [3], [4]. To solve this problem, a method called bilinear z-transformation with pre-warping frequency is used. And also from this method, applying inverse bilinear z-transformation [5], the inverse transformation from a digital filter to an analog low-pass prototype can be found. Thus, it is called inverse bilinear z-transformation with pre-warping frequency. This paper presents a new algorithm for frequency transformation in the digital domain using bilinear and inverse bilinear z-transformation with pre-warping frequency, which can be described by a mathematics equation called ‘Digital low-pass to digital filter Pascal matrix equation’.

II. BILINEAR AND INVERSE BILINEAR Z-TRANSFORMATION WITH PRE-WARPING FREQUENCY

Bilinear z-transformation is a method for transforming an analog filter to an equivalent digital filter [1]. It is defined by (1), where T is a sampling time period, e^{j\omega T} and z e^{j\omega}. Let \omega_A be an angular frequency in the s-domain and \omega_D be an angular frequency in the digital domain:

\[ s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \]  

(1)

Equation (1) can be rewritten as:

[1 - e^{-j\omega T}] [2 \tan(j \frac{T}{2} \omega_D)] = j \tan(j \frac{T}{2} \omega_A)

(2)

\[ j \omega_s = \frac{2}{T} \frac{1 - e^{-j\omega T}}{1 + e^{-j\omega T}} = \frac{2}{T} \tan(\frac{T}{2} \omega_D) = j \tan(\frac{T}{2} \omega_A) \]

From (2), the relationship between analog frequency \omega_A and digital frequency \omega_D is a non-linear function caused by the tan function. Pre-warping frequency can be used to overcome this.

Let \[ s = k \frac{1 - z^{-1}}{1 + z^{-1}} \]

where \( k \) is a design parameter and \( \omega_c \)

be a wanted frequency \( \omega_c = k \tan(\frac{T}{2} \omega_C) \), solve for:

\[ k = \frac{\omega_c}{\tan(\frac{T}{2} \omega_c)} \]

The bilinear z-transformation with pre-warping frequency is:

\[ s = \omega_c \cot(\frac{1 - z^{-1}}{1 + z^{-1}}) \]  

(3)

A. Frequency Transformation from the s-Domain to the z-Domain with Bilinear z-Transformation with Pre-Warping Frequency

The transfer function H(z) of a digital filter can be obtained from the transfer function H(s) of an analog low-pass prototype filter by first transforming an analog low-pass prototype to an analog filter that is the same class of the digital filter using frequency transform in the s-domain [6], and then applying the bilinear z-transformation with pre-warping frequency [7]-[9]. Table I illustrates the frequency transformation from the s-domain to the z-domain.

Let \( f_t \) is the sampling frequency, \( f_c \) is the cut-off frequency for the low-pass and high-pass filter, and \( f_b \) and \( f_l \) are the upper and lower frequency for the band-pass, band-stop and narrow-band filter. The parameters of \( c, t, U \) and \( L \) can be found as below:

\[ c = \cot\left(\pi \frac{f_t}{f_c}\right) \]

\[ t = \tan\left(\pi \frac{f_t}{f_c}\right) \]

\[ \left\{ \begin{array}{l}
 c_v = \cot\left(\pi \frac{f_t}{f_v}\right) \\
 t_v = \tan\left(\pi \frac{f_t}{f_v}\right)
\end{array} \right. \Rightarrow \left\{ \begin{array}{l}
 U = \frac{c_v}{1 - c_v t_v} \\
 L = \frac{t_v}{1 - c_v t_v}
\end{array} \right. \]
In the case of a narrow band, if given the center frequency \( f_0 \) and quality factor \( Q \), the upper and lower frequency can be calculated as below:

\[
\begin{cases}
  f_L = f_s \sqrt{1 + \frac{1}{4Q^2} - \frac{1}{2Q}} \\
  f_U = f_s \sqrt{1 + \frac{1}{4Q^2} + \frac{1}{2Q}}
\end{cases}
\]

B. Inverse Bilinear z-Transformation with Pre-Warping Frequency

Inverse bilinear z-transformation with pre-warping frequency is a method used to transform a digital filter back to an analog low-pass prototype [10]. In this paper, it shows the transformation from a digital low-pass to an analog low-pass prototype.

<table>
<thead>
<tr>
<th>Converting types</th>
<th>Frequency transformation in s-domain</th>
<th>Bilinear z-transformation with pre-warping frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low pass to low pass</td>
<td>( \frac{s}{\omega_s} )</td>
<td>( \frac{1 - z^{-1}}{1 + z^{-1}} )</td>
</tr>
<tr>
<td>Low pass to high pass</td>
<td>( \frac{\omega_c}{s} )</td>
<td>( \frac{i}{1 + z^{-1}} )</td>
</tr>
<tr>
<td>Low pass to band pass</td>
<td>( \frac{s^2 + \omega_c^2}{s + \omega_c} )</td>
<td>( \frac{1}{U - z^{-1}} )</td>
</tr>
<tr>
<td>Low pass to band stop</td>
<td>( \frac{(\omega_c - \omega_l)^2}{s + \omega_c} )</td>
<td>( \frac{1}{U - z^{-1} + L + \frac{1 + z^{-1}}{1 - z^{-1}}} )</td>
</tr>
</tbody>
</table>

In Table I, to transform an analog low-pass prototype with transfer function \( H(s) \) to a digital low-pass with transfer function \( H(z) \), \( s \) in \( H(s) \) is replaced by \( \frac{1 - z^{-1}}{1 + z^{-1}} \).

The inverse bilinear z-transformation with pre-warping frequency is then:

\[
s = c \frac{1 - z^{-1}}{1 + z^{-1}} \Rightarrow z^{-1} = \frac{c - s}{c + s} \quad (4)
\]

Equation (4) is an inverse bilinear z-transformation with warping frequency for transforming a digital low-pass to an analog low-pass prototype.

<table>
<thead>
<tr>
<th>TABLE II. FREQUENCY TRANSFORMATION IN DIGITAL DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converting types</td>
</tr>
<tr>
<td>Low pass to low pass</td>
</tr>
<tr>
<td>Low pass to high pass</td>
</tr>
<tr>
<td>Low pass to band pass</td>
</tr>
<tr>
<td>Low pass to band stop</td>
</tr>
</tbody>
</table>

IV. DIGITAL LOW-PASS TO DIGITAL FILTER PASCAL MATRIX EQUATION

The digital low-pass to digital filter Pascal matrix equation is used to transform the coefficients \( a_n \) and \( b_n \) of a digital low-pass filter to the coefficients \( a_d \) and \( b_d \) of a desired digital filter. This matrix equation is multiplied by some matrices, which are found from the coefficient and binomial expansion in Pascal’s triangle, and they are introduced in the next section.

A. Diagonal Matrix \( D_s \)

Matrix \( D_s \) is a diagonal matrix with a size of \( (n+1; n+1) \), where \( n \) is the given \( n \)-th-ordered digital low-pass filter, as shown below:

\[
D_s = \begin{bmatrix}
  x^n & 0 & 0 & \cdots & 0 \\
  0 & x^{n-1} & 0 & \cdots & 0 \\
  0 & 0 & x^{n-2} & \cdots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & 0 & \cdots & 1
\end{bmatrix}
\]

B. Matrix \( P \)

Matrix \( P \) has a size of \( (n+1; n+1) \) and contains the positive and negative coefficients of Pascal’s triangle. There are three different types depending on the conversion type, matrix \( P \) may be \( P_{LP}, P_{HBS} \) or \( P_{BS} \), as shown below:

\[
P_{LP} = \begin{bmatrix}
  (P_{LP})_{00} & (P_{LP})_{01} & (P_{LP})_{02} & \cdots & (P_{LP})_{0n} \\
  (P_{LP})_{01} & (P_{LP})_{11} & (P_{LP})_{12} & \cdots & (P_{LP})_{1n} \\
  (P_{LP})_{02} & (P_{LP})_{12} & (P_{LP})_{22} & \cdots & (P_{LP})_{2n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  (P_{LP})_{0n} & (P_{LP})_{1n} & (P_{LP})_{2n} & \cdots & (P_{LP})_{nn}
\end{bmatrix}
\]

\[
P_{HBS} = \begin{bmatrix}
  (P_{HBS})_{00} & (P_{HBS})_{01} & (P_{HBS})_{02} & \cdots & (P_{HBS})_{0n} \\
  (P_{HBS})_{01} & (P_{HBS})_{11} & (P_{HBS})_{12} & \cdots & (P_{HBS})_{1n} \\
  (P_{HBS})_{02} & (P_{HBS})_{12} & (P_{HBS})_{22} & \cdots & (P_{HBS})_{2n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  (P_{HBS})_{0n} & (P_{HBS})_{1n} & (P_{HBS})_{2n} & \cdots & (P_{HBS})_{nn}
\end{bmatrix}
\]

\[
P_{BS} = \begin{bmatrix}
  (P_{BS})_{00} & (P_{BS})_{01} & (P_{BS})_{02} & \cdots & (P_{BS})_{0n} \\
  (P_{BS})_{01} & (P_{BS})_{11} & (P_{BS})_{12} & \cdots & (P_{BS})_{1n} \\
  (P_{BS})_{02} & (P_{BS})_{12} & (P_{BS})_{22} & \cdots & (P_{BS})_{2n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  (P_{BS})_{0n} & (P_{BS})_{1n} & (P_{BS})_{2n} & \cdots & (P_{BS})_{nn}
\end{bmatrix}
\]
The transformation from a digital low-pass to another digital filter is called the digital low-pass to digital filter Pascal matrix equation, as written below:

\[ P_{85} = \left\{ \begin{array}{c}
(P_{LP})_{i,j} = \binom{n}{j} \\
(P_{LP})_{i,j} = (-1)^{i-j} \binom{n}{j} \\
(P_{LP})_{i,j} = 1 \\
(P_{LP})_{i,j} = -\left( (P_{LP})_{i,j} + (P_{LP})_{i-1,j} - (P_{LP})_{i,j-1} \right) 
\end{array} \right. \]

C. Matrix \( T \)

Two kinds of matrix \( T - T_{UL} \) and \( T_e \) are introduced. One useful application of Pascal’s triangle is the expansion of binomial \((U+L)^n\). Inserting zeroes will make matrix \( T_{UL} \), with a size of \((n+1; 2n+1)\), and matrix \( T_e \), with a size of \((n+1; n+1)\) as shown below:

\[
T_{UL} = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & U & 0 & L & 0 & 0 \\
0 & 0 & U^2 & 0 & 2L & 0 & L^2 & 0 & 0 \\
0 & 0 & U^3 & 0 & 3U^2 & 0 & 3UL & 0 & L^3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \ldots & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[
T_e = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & x^2 & 0 & 0 \\
0 & 0 & x^3 & 0 \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & x^n \\
\end{bmatrix}
\]

The transformation from a digital low-pass to another digital filter is called the digital low-pass to digital filter Pascal matrix equation, as written below:

\[ \begin{align*}
a_j &= [a_{ij}] [P_{LP}] [P_{LP}] [P_{LP}] [P_{LP}] \\
b_j &= [b_{ij}] [P_{LP}] [P_{LP}] [P_{LP}] [P_{LP}] 
\end{align*} \] (5)

V. EXAMPLES FOR DESIGN OF DIGITAL FILTER

This section demonstrates five examples for transforming a digital low-pass to a digital filter using the digital low-pass to digital filter Pascal matrix equation with MATLAB programming as shown in Fig. 1 (low pass to low pass), Fig. 2 (low pass to high pass), Fig. 3 (low pass to band pass), Fig. 4 (low pass to band stop) and Fig. 5 (low pass to Notch). 

A. Example 1: Transforming a Digital Low-Pass to a Digital Low-Pass

Transforming a second-ordered Butterworth digital low-pass with the transfer function \( H(z) \) at corner frequency \( 200 \text{Hz} \) to a digital low-pass with corner frequency \( 400 \text{Hz} \) at sampling frequency \( 2000 \text{Hz} \):

\[
H(z) = \frac{1 + 2z^{-1} + z^{-2}}{14.8246 - 16.9443z^{-1} + 6.1196z^{-2}}
\]

\[ f_c = 200 \text{Hz} \quad f_s = 400 \text{Hz} \quad f_s = 2000 \text{Hz} \]

\[ c = \cot\left( \frac{\pi f_c}{f_s} \right) = 3.0777 \quad cn = \cot\left( \frac{\pi f_c f_s}{f_c f_s} \right) = 1.3764 \]

Apply (5)

\[ \begin{align*}
a_j &= [a_{ij}] [P_{LP}] [T_e] [P_{LP}] \\
b_j &= [b_{ij}] [P_{LP}] [T_e] [P_{LP}] 
\end{align*} \]

B. Example 2: Transforming a Digital Low-Pass to a Digital High-Pass

Transforming a second-ordered Butterworth digital low-pass with transfer function \( H(z) \) at corner frequency \( 200 \text{Hz} \) to a digital high-pass with corner frequency \( 200 \text{Hz} \) at sampling frequency \( 2000 \text{Hz} \):

\[
H(z) = \frac{1 + 2z^{-1} + z^{-2}}{14.8246 - 16.9443z^{-1} + 6.1196z^{-2}}
\]

\[ f_c = 200 \text{Hz} \quad f_s = 2000 \text{Hz} \quad \Rightarrow t = \tan\left( \frac{\pi f_c}{f_s} \right) = 0.3249 \]

Apply (5)

\[ \begin{align*}
a_j &= [a_{ij}] [P_{LP}] [T_e] [P_{LP}] \\
b_j &= [b_{ij}] [P_{LP}] [T_e] [P_{LP}] 
\end{align*} \]

\[
H(z) = \frac{1 + 2z^{-1} + z^{-2}}{14.8246 - 16.9443z^{-1} + 6.1196z^{-2}}
\]

Figure 1. Transforming a digital low-pass to a digital low-pass using the digital low-pass to digital filter Pascal matrix equation.
C. Example 3: Transforming a Digital Low-Pass to a Digital Band-Pass

Transforming a second-ordered Butterworth digital low-pass with transfer function \( H(z) \) at corner frequency 200Hz to a digital band-pass with upper frequency 400Hz and lower frequency 200Hz at sampling frequency 2000Hz:

Apply (5)

\[
\begin{align*}
a_p &= a_2P_{DS_{60}}T_{LS}P_{INS} \\
b_p &= b_2P_{DS_{60}}T_{LS}P_{INS}
\end{align*}
\]

\[
f_0 = 400Hz , f_c = 200Hz , f_f = 2000Hz
\]

\[
\begin{align*}
c_a &= \cot \left( \frac{f_c}{f_f} \right) = 1.3764 & \Rightarrow \quad U = \frac{c_a}{1 - c_a f_f} = 2.4899 \\
t_a &= \tan \left( \frac{f_c}{f_f} \right) = 0.3249 & \Rightarrow \quad L = \frac{t_a}{1 - c_a f_f} = 0.5878
\end{align*}
\]

\[
a_p = [1 \quad 2 \quad 1] = 0.9721 \quad 0 \quad 0.3077
\]

\[
b_p = [14.8246 -16.9443 \quad 6.1196] = 9.4721 \quad 0 \quad 3.0777
\]

\[
H_{BP}(\zeta) = \frac{1 - 2.472\zeta + 3.527\zeta^2 - 2.472\zeta^3 + \zeta^4}{1.565 - 3.04\zeta + 3.166\zeta^2 - 1.904\zeta^3 + 0.646\zeta^4}
\]

D. Example 4: Transforming a Digital Low-Pass to a Digital Band-Stop

Transforming a second-ordered Butterworth digital low-pass with transfer function \( H(z) \) at corner frequency 200Hz to a digital band-stop with upper frequency 400Hz and lower frequency 200Hz at sampling frequency 2000Hz:

Apply (5)

\[
\begin{align*}
a_s &= a_2P_{DS_{60}}T_{LS}P_{INS} \\
b_s &= b_2P_{DS_{60}}T_{LS}P_{INS}
\end{align*}
\]

\[
f_s = 400Hz , f_c = 200Hz , f_f = 2000Hz
\]

\[
\begin{align*}
c_s &= \cot \left( \frac{f_c}{f_f} \right) = 1.3764 & \Rightarrow \quad U = \frac{c_s}{1 - c_s f_f} = 2.4899 \\
t_s &= \tan \left( \frac{f_c}{f_f} \right) = 0.3249 & \Rightarrow \quad L = \frac{t_s}{1 - c_s f_f} = 0.5878
\end{align*}
\]

\[
a_s = [1 \quad 1 \quad 2 \quad 1] = 1.0 \quad 1 \quad 0.3455 \quad 1 \quad 0 \quad 2 \quad 0 \quad 1
\]

\[
b_s = [14.8246 -16.9443 \quad 6.1196] = 9.4721 \quad 0 \quad 3.0777
\]

\[
H_{BS}(\zeta) = \frac{1 - 2.472\zeta + 3.527\zeta^2 - 2.472\zeta^3 + \zeta^4}{1.565 - 3.04\zeta + 3.166\zeta^2 - 1.904\zeta^3 + 0.646\zeta^4}
\]
it is more effective in programs and calculations. All operations imply the matrix multiplication; therefore, the algorithm of this method is simple because computing, when transforming the digital filter in the z-domain. The algorithm of this method is simple because computing, when transforming the digital filter in the z-domain.

A new method was developed for transforming a digital low-pass to digital filter Pascal matrix equation. This idea shows a lot of promise in applications which he absorbed in them and hope to share to whom have the same interests. 

Fig. 5. Transforming a digital low-pass to a digital Notch filter using the digital low-pass to digital filter Pascal matrix equation.

VI. CONCLUSION

A new method was developed for transforming a digital low-pass filter with transfer function $H(z)$ to a digital filter (low-pass, high-pass, band-pass, band-stop and narrow-band). The involvement of Pascal’s triangle in the digital low-pass to digital filter Pascal matrix equation, as presented and demonstrated in the examples, made the work easier for hand calculations and computing, when transforming the digital filter in the z-domain. The algorithm of this method is simple because all operations imply the matrix multiplication; therefore, it is more effective in programs and calculations.

ACKNOWLEDGMENT

This is my own research, and I would like to share my idea of how to apply Pascal’s triangle in filters design. I would appreciate feedback from readers to improve this research.

REFERENCES


