The Frequency Response Computation of $H(z)$ Models that Include Digital Error Effects

1984

Carol R. Smith
University of Central Florida

Find similar works at: https://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Engineering Commons

STARS Citation

https://stars.library.ucf.edu/rtd/4708

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
THE FREQUENCY RESPONSE COMPUTATION OF H(Z) MODELS
THAT INCLUDE DIGITAL ERROR EFFECTS

BY

CAROL R. SMITH
B.S.E., University of South Florida, 1980

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Spring Term
1984
ABSTRACT

Pulse responses are not suitable for evaluating $H(z)$ frequency responses that include digital errors. Digital error effects, however, are conveniently included in $H(z)$ frequency response calculations based on the step response. A non-real time algorithm for computing the frequency response of real time digital filters based on the filter step response is developed and verified through an example.
# TABLE OF CONTENTS

INTRODUCTION ............................................. 1

WHY PULSE RESPONSES ARE NOT SUITABLE TO EVALUATE H(Z) FREQUENCY RESPONSES THAT INCLUDE DIGITAL ERRORS ........................................... 4

DIGITAL ERRORS ARE CONVENIENTLY INCLUDED IN H(Z) FREQUENCY RESPONSE EVALUATIONS BASED ON THE STEP RESPONSE ..................... 6

Step Invariant Approach .................................... 6
Z-Transform Approach ....................................... 10

AN EXAMPLE OF AN H(Z) FREQUENCY RESPONSE FROM THE STEP RESPONSE THAT INCLUDES DIGITAL EFFECTS ........................................ 13

CONCLUSIONS ............................................. 16

FUTURE DIRECTIONS ....................................... 17

APPENDIX .................................................... 18

Fortran Implementation of an H(z) Frequency Response Algorithm ........................................ 18

REFERENCES .............................................. 21
INTRODUCTION

When a digital filter is implemented as a computer algorithm or in special-purpose hardware all relevant filter parameters must be represented using finite-length registers. Since infinite precision cannot be used the filter performance will be affected. Some of the digital error effects are as follows:

1. Inaccuracies due to coefficient quantization.
3. Uncorrelated roundoff noise.
5. Nonlinearities due to quantization of arithmetic results.

Several factors influence how these digital errors affect the filter performance. Included among these factors are the type of arithmetic used, the method of quantization, and the exact filter structure used.

Since the frequency response of an implemented filter may differ appreciably from the original design, it is desirable to develop a direct method of evaluating these architectural and roundoff effects.

Discrete dynamic linear shift-invariant (LSIV) systems can be modeled by the LSIV single-input single-output difference equation of the form:

$$\sum_{n=0}^{N} d_n y(k-n) = \sum_{m=0}^{M} c_m x(k-m)$$  (1)
The transfer function model comes from the z-transform of (1):

\[
H(z) = \frac{\sum_{m=0}^{M} c_m z^{-m}}{\sum_{n=0}^{N} d_n z^{-n}}
\]  

The frequency response of the H(z) model can be obtained by evaluating H(z) with z replaced by \( e^{j\omega T} \) where T is the sample period. Because of periodicity and symmetry, the radian frequency needs only to be evaluated over the range defined by \( 0 \leq \omega \leq \omega_s / 2 \) where \( \omega_s = 2\pi / T \).

An indirect method of evaluating the frequency response of a digital filter is to drive the actual implementation with white noise. A digital spectrum analyzer can then be used to estimate and average the frequency characteristics of H(z). This approach includes all digital quantization affects but requires building a prototype filter, or a more difficult implementation of a software spectrum analyzer.

Another method which would include these effects is to run sinusoidal input simulations until steady state is reached. This procedure must be done for each frequency of interest—which is time consuming and extremely costly!

It is possible to calculate the proper initial conditions for the system so that it will be in steady state when a sinusoidal input is applied. In this case, it would be necessary to run the simulation for only half of a cycle for each frequency. The drawbacks with this
frequency of interest and that a different procedure for calculating
the initial conditions is needed for each architecture.

All frequency response information is contained within any
transient response. Thus it seems that the best method would utilize
a transient solution obtained from a simulation which includes all
quantization effects. Procedures based on the pulse response \( h(n) \)
and the step response \( h_1(n) \) shall be investigated. An algorithm
for computing the frequency response from the step response shall be
derived. This algorithm was used in designing a program to test and
evaluate the frequency response example which will be presented.
Theoretical and experimental computer results are compared and
discussed.
WHY PULSE RESPONSES ARE NOT SUITABLE TO EVALUATE $H(z)$

FREQUENCY RESPONSES THAT INCLUDE DIGITAL ERRORS

The basic definition for the $z$-transform function $H(z)$ of a causal system is given by

$$H(z) = \sum_{n=0}^{\infty} h(n)z^{-n}$$

(3)

This form suggests a method of calculating the frequency response. The transient pulse response $h(n)$ obtained from an accurate simulation that includes all quantization affects could be substituted in (3). Then $H(z)$ could be evaluated out to some $n$ for adequate convergence.

For high sample rates, however, a single pulse input $x(n) = \delta(n)$ produces an extremely low amplitude output and thus varies over an extremely small dynamic range. A higher signal to noise ratio could be obtained by appropriate scaling of $x(n)$. The input $x(n)$ should be scaled such that $h(n)$ will have a wide dynamic range. Generally, $H(z)$ models are scaled to accommodate a unit step input, that is, $H(z)$ is adjusted so that $H_{dc} = H(1)$ is 1/2 to allow for transient overshoot.

For a system modeled by

$$y(k) = A[x(k) + \sum_{m=1}^{M} c_m x(k-m)] - \sum_{n=1}^{N} d_n y(k-n)$$

(4)

it appears that the input should be scaled by a factor of $1/A$. 

However, consider a real simulation model given by

\[
H(z) = \frac{0.04044(1 + z^{-1})}{1.0 - 0.91911z^{-1}} \times \frac{0.07119(1.0 - 1.73770z^{-1} + z^{-2})}{1.0 - 1.91077z^{-1} + 0.94811z^{-2}} \times \frac{0.15445(1.0 - 1.88554z^{-1} + z^{-2})}{1.0 - 1.84912z^{-1} + 0.86680z^{-2}}
\]

Taking the inverse z-transform of \( H(z) \) yields

\[
y(k) = 4.4465 \times 10^{-4}[x(k) + \sum_{m=1}^{5} c_m x(k-m)] - \sum_{n=1}^{5} d_n y(k-n)
\]

(5)

It then appears that the pulse input should be scaled by a factor of \( 1/4.4465 \times 10^{-4} \). Applying this input to the first stage gives

\[
H(z) \times [1/4.4465 \times 10^{-4}] = \frac{0.04044}{4.4465 \times 10^{-4}} \times \frac{0.07119(1.0 - z^{-1})}{1.0 - 0.91911z^{-1}} \times \frac{90(1.0 + z^{-1})}{1.0 - 0.91911z^{-1}}
\]

(7)

Thus the output of the first section would exceed fixed point scaled limits.

It is possible to solve for \( y(k-m) \) initial conditions such that \( y(k) = h(k) \) would have a wide dynamic range. This again requires a separate calculation for each architecture, not to mention the overflow dangers in fixed point arithmetic implementations. These approaches could be applied to floating-point arithmetic implementations but a more general approach which is applicable to both fixed and floating point architectures is desired.
DIGITAL ERRORS ARE CONVENIENTLY INCLUDED IN H(Z) FREQUENCY RESPONSE EVALUATIONS BASED ON THE STEP RESPONSE

Since the scaling of H(z) models can be based on a unit step response, it seems that the next transient response approach to be investigated should be the step response. Two different approaches are presented which both lead to the development of an effective algorithm for evaluating quantization effects on the frequency response.

Step Invariant Approach

\[
\begin{align*}
X(z) & \xrightarrow{H(z)} H_1(z) \xrightarrow{G(z)} Y(z) \\
\text{Fig. 1. Model for step invariant approach}
\end{align*}
\]

Consider the model of figure 1, where H(z) is a given transfer function model and the input is a unit step given by

\[
\begin{align*}
x(n) &= \delta_1(n) \\
X(z) &= \frac{z}{z - 1}
\end{align*}
\]

This model can be used to evaluate the frequency response of H(z) if G(z) is selected such that the output Y(z) is the pulse response, i.e., Y(z) = H(z). The frequency response could then be computed as the Fourier transform of the pulse response.

The output of the model is given by:

\[
Y(z) = X(z) H(z) G(z)
\]

For Y(z) = H(z),

\[
G(z) = \frac{1}{X(z)} = 1 - z^{-1}
\]
The pulse response can be written in terms of the step response as

\[ h(n) = h_1(n) - h_1(n-1) \]  \hspace{1cm} (11)

using only superposition. It can be easily verified that \( y(n) \) is indeed the pulse response. Let the input \( x(n) \) be a unit step. The output of the first stage is the step response \( h_1(n) \) of the given filter. The system output \( y(n) \) is given by

\[ y(n) = h_1(n) \ast g(n) \]  \hspace{1cm} (12)

Taking the inverse z-transform of \( G(z) \) yields

\[ g(n) = \delta(n) - \delta(n-1) \]  \hspace{1cm} (13)

Thus

\[ y(n) = h_1(n) \ast [\delta(n) - \delta(n-1)] \]

\[ y(n) = \sum_{m=-\infty}^{\infty} h_1(n-m) \{ \delta(m) - \delta(m-1) \} \]

\[ y(n) = h_1(n) - h_1(n-1) = h(n) \]  \hspace{1cm} (14)

Therefore, \( y(n) \) is simply the pulse response of \( H(z) \) for the given \( G(z) \) and a step input.

The frequency response can now be found as the Fourier transform of \( y(n) \). The Fourier transform pair is

\[ h(n) = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} H(e^{j\omega T})e^{j\omega n} \, d\omega \]

\[ H(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n} \]  \hspace{1cm} (15)

The Fourier transform of \( y(n) \) is

\[ Y(e^{j\omega T}) = H_1(e^{j\omega T}) G(e^{j\omega T}) \]  \hspace{1cm} (16)

Assuming that \( H(z) \) represents a causal system and that for \( n > N_D \), \( h_1(n) \) is at its steady state value \( h_{1SS} \) then \( h_1(n) \) is
given by:

\[
    h_1(n) = \begin{cases} 
    0 & n < 0 \\
    h_1(n) & 0 \leq n \leq N_D \\
    h_{1SS} & N_D < n 
    \end{cases} 
\]  

Thus

\[
    H_1(e^{j\omega T}) = \sum_{n=0}^{N_D} h_1(n) e^{-j\omega n T} + \sum_{n=N_D+1}^{\infty} h_{1SS} e^{-j\omega n T} 
\]  

(17)

The Fourier transform of \( g(n) \) is

\[
    G(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} [\delta(n) - \delta(n-1)] e^{-j\omega n T} 
\]

(19)

\[
    G(e^{j\omega T}) = 1 - e^{-j\omega T} 
\]

(18)

\[
    Y(e^{j\omega T}) \text{ is computed as} 
\]

\[
    Y(e^{j\omega T}) = \left( \sum_{n=0}^{N_D} h_1(n)e^{-j\omega n T} + \sum_{n=N_D+1}^{\infty} h_{1SS} e^{-j\omega n T} \right)[1 - e^{-j\omega T}] 
\]

(20)

Letting

\[
    Y_N(e^{j\omega T}) = [1 - e^{-j\omega T}]\left( \sum_{n=0}^{N_D} h_1(n)e^{-j\omega n T} \right) 
\]

(21)

and

\[
    Y_{SS}(e^{j\omega T}) = [1 - e^{-j\omega T}][h_{1SS} \sum_{n=N_D+1}^{\infty} e^{-j\omega n T}] 
\]

(22)

then

\[
    Y(e^{j\omega T}) = Y_N(e^{j\omega T}) + Y_{SS}(e^{j\omega T}) 
\]

(23)

Now, \( Y_{SS}(e^{j\omega T}) \) can be simplified as follows.

\[
    Y_{SS}(e^{j\omega T}) = h_{1SS} \sum_{n=N_D+1}^{\infty} e^{-j\omega n T} - h_{1SS} \sum_{n=N_D+1}^{\infty} e^{-j\omega T(n+1)} 
\]

(24)
Letting $m = n + 1$,

\[
Y_{SS}(e^{j\omega T}) = h_{1SS} \sum_{n=ND+1}^{\infty} e^{-j\omega nT} - h_{1SS} \sum_{m=ND+2}^{\infty} e^{-j\omega mT}
\]

\[
= [h_{1SS} e^{-j\omega (ND+1)} + h_{1SS} \sum_{n=ND+2}^{\infty} e^{-j\omega nT}]
\]

\[
- h_{1SS} \sum_{m=ND+2}^{\infty} e^{-j\omega mT}
\]

(25)

Thus

\[
Y_{SS}(e^{j\omega T}) = h_{1SS} e^{-j\omega (ND+1)}
\]

(26)

Now, let $w = k \Delta w$ where $\Delta w = w_s/N = 2\pi/NT$. This gives $\omega T = k(2\pi/N)$ and

\[
Y(e^{j\omega T}) = Y(e^{j2\pi k/N}) = Y(k)
\]

(27)

Assuming that $N = ND + 1$, $Y_{SS}(k)$ can be simplified further.

\[
Y_{SS}(k) = h_{1SS} e^{-j[2\pi k/(ND+1)](ND+1)} = h_{1SS} e^{-j2\pi k}
\]

(28)

But, $e^{-j2\pi k} = 1$, so

\[
Y_{SS}(k) = h_{1SS}
\]

(29)

Thus

\[
Y(k) = [1 - e^{-j2\pi k/N}][\sum_{n=0}^{N-1} h_1(n)e^{-j2\pi nk/N} + h_{1SS}]
\]

(30)

The discrete Fourier transform (DFT) pair is defined as

\[
X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}
\]

\[
x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi nk/N}
\]

(31)

Using the DFT, the frequency response can be found by

\[
Y(k) = H_1(k)[1 - e^{-j2\pi k/(ND+1)}] + h_{1SS}
\]

(32)
where $H_1(k)$ is the $N_D + 1$ point DFT of the step response of the given $H(z)$ model.

Z-Transform Approach

![Diagram of Z-Transform Approach](image)

Fig. 2. Model for z-transform approach

Consider the model of figure 2 for a given transfer function $H(z)$ when the input is a unit step, i.e.,

$$X(z) = 1/(1-z^{-1})$$ (33)

In this case, $Y(z)$ is the z-transform of the step response $h_1(n)$, and

$$H_1(z) = X(z) H(z)$$ (34)

or

$$H_1(z) = [1/(1-z^{-1})] H(z)$$ (35)

Thus

$$H(z) = (1 - z^{-1}) H_1(z)$$ (36)

Applying the definition of the z-transform to $H_1(z)$,

$$H(z) = (1 - z^{-1}) \sum_{n=0}^{\infty} h_1(n)z^{-n}$$ (37)

Assuming that for $n \leq N_D$, $h_1(n)$ is at steady state

$$h_1(n) = \begin{cases} h_1(n) & 0 \leq n \leq N_D \\ h_{1SS} & N_D < n \end{cases}$$ (38)

then,

$$H(z) = [1 - z^{-1}][\sum_{n=0}^{N_D} h_1(n)z^{-n} + \sum_{n=N_D+1}^{\infty} h_{1SS}z^{-n}]$$

$$= [1 - z^{-1}][\sum_{n=0}^{N_D} h_1(n)z^{-n}] + [1 - z^{-1}][\sum_{n=N_D+1}^{\infty} h_{1SS}z^{-n}]$$ (39)
The second term, call it $H_{1SS}(z)$, can be simplified.

$$H_{1SS}(z) = (1 - z^{-1}) \sum_{n=0}^{\infty} h_{1SS} z^{-n}$$

$$= \sum_{n=N_D+1}^{\infty} h_{1SS}(z^{-n} - z^{-(n+1)})$$

Letting $m = n + 1$,

$$H_{1SS}(z) = \sum_{n=N_D+1}^{\infty} h_{1SS} z^{-n} - \sum_{m=N_D+2}^{\infty} h_{1SS} z^{-m}$$

$$= [h_{1SS} z^{-(N_D+1)} + \sum_{n=N_D+2}^{\infty} h_{1SS} z^{-n}] - \sum_{m=N_D+2}^{\infty} h_{1SS} z^{-m}$$

$$H_{1SS}(z) = h_{1SS} z^{-(N_D+1)}$$

Thus

$$H(z) = (1 - z^{-1}) \sum_{n=0}^{N_D} h_1(n)z^{-n} + h_{1SS} z^{-(N_D+1)}$$

The frequency response of $H(z)$ can be found by replacing $z$ with $e^{j\omega T}$. This result allows the calculation of the frequency response for any frequency $\omega$.

Further simplification of equation (42) is possible. Letting $w = k \Delta w$ where $\Delta w = w_s/N = 2 \pi /NT$, then $\omega T = 2 \pi k/N$. Now

$$H(z) = H(e^{j\omega T}) = H(e^{j2 \pi k/N}) = H(k)$$

Letting $N = N_D + 1$

$$H_{1SS}(k) = h_{1SS} e^{-j[2 \pi k/(N_D+1)][N_D+1]}$$

$$= h_{1SS} e^{-j2 \pi k} = h_{1SS}$$

Thus

$$H(k) = [1 - e^{-j2 \pi k/N}] \left[ \sum_{n=0}^{N-1} h_1(n)e^{-j2 \pi nk/N} \right] + h_{1SS}$$
Recognizing the summation as the DFT of $h_1(n)$,

$$H(k) = H_1(k)[1 - e^{-j2\pi nk/N}] + h_{1SS}$$  \hspace{1cm} (46)

This is identical to equation (32), the result from the step invariant approach. It should be noted that while equation (42) allows for calculating the frequency response for any $w$, the simplified result of (46) allows for finding only $N$ values of $H(k)$ at frequencies of $w = k\Delta w$. For higher resolution, $H(k)$ can be computed for $N > N_D + 1$ points. In this case, $h_1(n) = h_{1SS}$ for $N_D + 1 \leq n \leq N$. 
AN EXAMPLE OF AN H(Z) FREQUENCY RESPONSE FROM THE STEP RESPONSE THAT INCLUDES DIGITAL ERROR EFFECTS

The frequency response of a fifth order filter has been evaluated to demonstrate the effectiveness of this technique. The filter was implemented with the Intel 2920 Signal Processor which has 9 bit I/O and carries 25 bits internally. The unit step response was obtained from a simulation run on the Intel 2920 simulator.

The H(z) model of the filter is

\[
H(z) = \frac{0.040442658(1+z^{-1})}{1.0-0.91911468z^{-1}} \times \frac{0.071185498(1.0-1.7377013z^{-1}+z^{-2})}{1.0-1.9107708z^{-1}+0.94811434z^{-2}} \times \frac{0.15445013(1.0-1.8855376z^{-1}+z^{-2})}{1.0 - 1.8491228z^{-1} + 0.86680162z^{-2}}
\]

The sampling frequency is 10K hertz.

To calculate the frequency response, the algorithm of (46) was used, i.e.,

\[
H(k) = H_1(k)[1 - e^{-j2\pi k/N}] + h_{1SS}
\]  

Using the trigonometric form for the exponential this becomes

\[
H(k) = H_1(k)[1 - \cos(2\pi k/N) + j \sin(2\pi k/N)] + h_{1SS}
\]

This algorithm was easily programmed in Fortran, making use of the library FFT subroutine for the computation of \(H_1(k)\). The complete program listing is given in the Appendix.
In this example, a 512 point FFT was used to compute the DFT of the step response. The step response was evaluated by the simulation for 300 points. For \( n > 300 \), the step response was set equal to \( h_{1SS} \).

The frequency response was computed for both the 9 bit truncated step response and for the step response prior to truncation. The gain for these two cases is given in figure 3, along with the theoretical frequency response which assumes no digital errors.

Although the truncated step response is specified at 9 bits, one bit is lost due to the sign and there is always one bit of uncertainty in conversion. Therefore the step response is only accurate to 7 bits and so the gain calculated from the truncated step response should be down from zero db in the order of \( 7 \times 6 = 42 \text{db} \).

From figure 3 it is seen that this interpretation agrees with the expected results.

The gain computed using the step response prior to truncation is below 54db. However, there appear to be three small errors around 700hz, 1400hz, and 3000hz. These errors could be due to aliasing and/or the quantization of \( w \), i.e., \( w \) is \( k \Delta w \) (\( w \) is not continuous due to utilizing the FFT calculation routine). The gain for the pre-truncation case is more accurate than the gain calculated from the truncated step response. However, there is no reason to assert that errors before truncation must be in the order of \( 25 \times 6 = 150 \text{db} \) below zero db because quantization effects in digital filters at internal full precision always affect the output precision with much more significance. Thus 80 to 90 db down is not unreasonable.
Fig. 3. Gain of frequency response example.

Theoretical
25 bits
9 bits

Gain (db)

5000
(f/2)

1000

100

-120
-100
-80
-60
-40
-20
0
CONCLUSIONS

The method that has been presented provides a quick and effective means of calculating an $H(z)$ frequency response which includes all digital quantization errors. Although an accurate simulation of the filter implementation which includes all digital error effects is necessary, only one simulation run of the step response is required. This is a vast improvement over running sinusiodal input simulations for each frequency of interest.

The algorithm in the form

$$H(e^{j\omega T}) = (1 - e^{-j\omega T}) \sum_{n=0}^{N_D} h_1(n)e^{-j\omega nT} + h_{1SS} e^{-j\omega(N_D+1)T}$$

allows for the frequency response to be calculated for any frequency $\omega$. The simplified version,

$$H(k) = H_1(k)(1 - e^{-j2\pi k/N}) + h_{1SS}$$

makes use of existing software for FFT algorithms. When an $N$ point FFT is used only $N$ values of the frequency response $H(k)$ are obtained. These values are at the frequencies defined by $\omega = k \Delta \omega$, $\Delta \omega = 2\pi/NT$. This resolution can be increased, however, by letting $N > N_D + 1$, where $N_D$ is the point where steady state is reached. In this case, the step response is set equal to its steady state value for $N \geq N_D + 1$. 
FUTURE DIRECTIONS

The purpose of this report was to develop and test a method of evaluating the $H(z)$ frequency responses of linear shift invariant digital filters such that all digital quantization errors of the filters would be included in the frequency responses. This objective has been met. From the test results it is seen that further work could be done to explain the effect that digital errors have on frequency responses. In particular, "Can the random nature of the ADC truncation be verified theoretically?" or "Why does the internal 25 bit response become so apparently organized?"
APPENDIX

FORTRAN IMPLEMENTATION OF AN H(Z) FREQUENCY RESPONSE ALGORITHM

The H(z) Frequency response can be computed using the following:

\[ H(k) = H_1(k)[1 - \cos(2 \pi k/N) + j \sin(2 \pi k/N)] + h_{1SS} \]  \hspace{1cm} (A-1)

where \(H_1(k)\) is the DFT of the step response, \(h_{1SS}\) is the steady state value, and \(N\) is the number of points used to compute \(H_1(k)\).

A Fortran program has been developed to implement this algorithm.

The program requires the user to supply two subroutines:

1. \textsc{Geth1}(H1,NDELT,FS,H1SS): \(H1\) – the step response values prior to reaching steady state

   \(NDELT\) – the number of points prior to steady state

   \(FS\) – the sampling frequency

   \(H1SS\) – the steady state value of \(h(n)\)

2. \textsc{Fft}(H1,N,X): Fast Fourier transform to perform an N point DFT on \(H1\) and store the result in complex array \(X\)

The frequency response is only computed for the first \(N/2\) values. This gives a frequency range of \(0 \leq \omega \leq \omega_s/2\).

The program listing is given in figure A-1. A sample output is shown in figure A-2 for the system modeled by:

\[ H(z) = \frac{0.040442658(1-z^{-1})}{1.0-0.91911468z^{-1}} \times \frac{0.071185498(1.0-1.7377013z^{-1}+z^{-2})}{1.0-1.9107708z^{-1}+0.94811434z^{-2}} \times \frac{0.15445013(1.0-1.8855376z^{-1}+z^{-2})}{1.0-1.8491228z^{-1}+0.86680162z^{-2}} \]  \hspace{1cm} (A-2)
Fig. A-1. Program for computation of H(z) frequency response algorithm
### Fig. A-2. Sample output for a fifth order filter

<table>
<thead>
<tr>
<th>FREQUENCY</th>
<th>RAD</th>
<th>MAG</th>
<th>GAIN</th>
<th>DP</th>
<th>PHASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.05</td>
<td>0.49</td>
<td>0.04</td>
<td>-4.08</td>
<td>-4.48</td>
</tr>
<tr>
<td>0.1</td>
<td>0.15</td>
<td>0.45</td>
<td>0.05</td>
<td>-3.98</td>
<td>-4.28</td>
</tr>
<tr>
<td>0.2</td>
<td>0.25</td>
<td>0.41</td>
<td>0.06</td>
<td>-3.88</td>
<td>-4.18</td>
</tr>
<tr>
<td>0.3</td>
<td>0.35</td>
<td>0.37</td>
<td>0.07</td>
<td>-3.78</td>
<td>-4.08</td>
</tr>
<tr>
<td>0.4</td>
<td>0.45</td>
<td>0.33</td>
<td>0.08</td>
<td>-3.68</td>
<td>-3.98</td>
</tr>
<tr>
<td>0.5</td>
<td>0.55</td>
<td>0.29</td>
<td>0.09</td>
<td>-3.58</td>
<td>-3.88</td>
</tr>
<tr>
<td>0.6</td>
<td>0.65</td>
<td>0.25</td>
<td>0.10</td>
<td>-3.48</td>
<td>-3.78</td>
</tr>
<tr>
<td>0.7</td>
<td>0.75</td>
<td>0.21</td>
<td>0.11</td>
<td>-3.38</td>
<td>-3.68</td>
</tr>
<tr>
<td>0.8</td>
<td>0.85</td>
<td>0.17</td>
<td>0.12</td>
<td>-3.28</td>
<td>-3.58</td>
</tr>
<tr>
<td>0.9</td>
<td>0.95</td>
<td>0.13</td>
<td>0.13</td>
<td>-3.18</td>
<td>-3.48</td>
</tr>
<tr>
<td>1.0</td>
<td>1.05</td>
<td>0.09</td>
<td>0.14</td>
<td>-3.08</td>
<td>-3.38</td>
</tr>
<tr>
<td>1.1</td>
<td>1.15</td>
<td>0.05</td>
<td>0.15</td>
<td>-2.98</td>
<td>-3.28</td>
</tr>
<tr>
<td>1.2</td>
<td>1.25</td>
<td>0.01</td>
<td>0.16</td>
<td>-2.88</td>
<td>-3.18</td>
</tr>
<tr>
<td>1.3</td>
<td>1.35</td>
<td>0.07</td>
<td>0.17</td>
<td>-2.78</td>
<td>-3.08</td>
</tr>
<tr>
<td>1.4</td>
<td>1.45</td>
<td>0.13</td>
<td>0.18</td>
<td>-2.68</td>
<td>-2.98</td>
</tr>
<tr>
<td>1.5</td>
<td>1.55</td>
<td>0.19</td>
<td>0.19</td>
<td>-2.58</td>
<td>-2.88</td>
</tr>
<tr>
<td>1.6</td>
<td>1.65</td>
<td>0.24</td>
<td>0.20</td>
<td>-2.48</td>
<td>-2.78</td>
</tr>
<tr>
<td>1.7</td>
<td>1.75</td>
<td>0.29</td>
<td>0.21</td>
<td>-2.38</td>
<td>-2.68</td>
</tr>
<tr>
<td>1.8</td>
<td>1.85</td>
<td>0.34</td>
<td>0.22</td>
<td>-2.28</td>
<td>-2.58</td>
</tr>
<tr>
<td>1.9</td>
<td>1.95</td>
<td>0.39</td>
<td>0.23</td>
<td>-2.18</td>
<td>-2.48</td>
</tr>
<tr>
<td>2.0</td>
<td>2.05</td>
<td>0.44</td>
<td>0.24</td>
<td>-2.08</td>
<td>-2.38</td>
</tr>
</tbody>
</table>

*Note: The table above represents a sample output for a fifth order filter. The columns correspond to frequency, magnitude, gain, and phase, respectively.*
REFERENCES


