Session 1220

# The Design of Optimal Digital Filters Directly from Analog Prototypes

**Aaron Robinson and Fred O. Simons, Jr.**
*High-performance Computing and Simulation (HCS) Research Laboratory*
**Department of Electrical Engineering**
**Florida A&M University and Florida State University**

***Abstract -*** *Generally analog prototype filters are not candidates for the design of optimum digital jilters because the processing requirements to convert from the analog prototype filter to the target digital filter are excessive, However, some optimized bilinear transform algorithms introduced by Simons and Harden to solve differential equation models were found to be adaptable to the problem of designing optimal digital filters without introducing excessive processing requirements. Based on these optimized bilinear transform algorithms, a procedure is derived whereby the coefficients of an analog prototype filter are adjusted in a parameter optimization process, The convergence of this process yields the digital filter that optimizes a cost function specifically formulated to realize desired digital filter goals and specifications. It is important to note that this new class of digital filters can be FIR or IIR with the latter form also guaranteed to be stable,*

## Introduction

The name optimal digital filter implies that filter design has been accomplished by specifying some criteria defined as optimal and the filter has been designed to meet that criteria. In many cases a prototype filter is chosen as a starting point and coefficient values are changed until satisfactory performance is reached. In such cases, digital filters are normally chosen as a starting point because of the large number of calculations needed to transform analog prototypes to discrete domains. An alternate approach, with algorithms, which makes it feasible to design optimal digital filters based on analog filter prototypes is presented by the authors. Much of this alternate design process rest on powerful algorithms, developed by Simons and Harden to solve differential equations. [SIMO88] These algorithms can be adapted to greatly reduce the number of computations required to derive a bilinear transformed digital $H(z)$ model from a prototype $H(s)$ analog model. This reduction in processing makes it feasible to base the optimal digital filter design on an analog prototype and arrive at solutions based on the changing coefficients of the analog filter. The end result is an optimally designed digital filter as well as an analog filter that could be claimed to be optimal in some sense.

In order to verify and converge on a proper filter design, the frequency response of the current filter in the design process must be continuously iterated until convergence to the optimal filter occurs. Frequency response calculations are also based on using new and efficient algorithms that are presented. These algorithms are based on real arithmetic operations, which account for their speed and efficiency[SIMO87].

In addition to providing the algorithms used in the formulation of the filter design, a detailed outline of the process used to arrive at the new class of optimally designed digital filters is provided. Based on the presentations and evaluations of all algorithms and procedures, the authors present conclusions and ideas for future research.

## Optimized Bilinear Transform Algorithms

Calculation of the bilinear transform provides a method of transferring a polynomial in "s" to the "z" domain where digital processing techniques may be used. Since this transform accounts for a large percentage of the computations needed to complete the optimal design, more efficient means of calculating the transform becomes critical to an efficient optimization process.

Simons and Harden presented a treatise on "An Optimized Simulation of Dynamic Continuous Models" in which the authors derived algorithms that provided the basis for structuring an optimized PC program for simulating differential equation models of continuous dynamic subsystem components [SIMO88]. These algorithms were adapted to reduce the number of computations needed to perform the bilinear transform.

Discussion of the bilinear transform process begins with an analog filter since this study is based on designing optimal digital filters directly from analog prototypes. The process assumes the prototype *H(s)* filter model has been specified as a ratio of two decreasing order polynomials in "s". It should be noted that the analog prototype may be specified by supplying the decreasing order polynomials, the poles and zeros, or a combination of both, Nevertheless, by applying commonly used algorithms for real and complex conjugate root accumulation or multiplication, *H(s)* factored forms can be put into the common polynomial form[ALKH86]:

$$H(s) = \frac{\sum_0^M b_m s^m}{\sum_0^N a_n s^n} \qquad [1]$$

Calculation of the bilinear transform then proceeds by making the following substitution:

$$H(z) \triangleq \left. \frac{\sum_0^M b_m s^m}{\sum_0^N a_n s^n} \right|_{s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} \qquad [2]$$

which is used to implement the bilinear transform and results in the common digital filter *H(z)* model consisting of a ratio of two polynomials in z or

$$H(z) \triangleq \frac{N(z)}{D(z)} = \frac{\sum_0^{M=N} c_m z^{-m}}{\sum_0^N d_n z^{-n}} \qquad [3]$$

The following algorithms for performing this transformation have been optimized with respect to processing and memory requirements.

Initially, it should be noted the scale factor of 2/T can be assumed to be one without loss of generality since the factor can be incorporated into the *H(s)* coefficients by making the following substitutions

$$b_m \leftarrow (2/T)^m \, b_m \qquad [4]$$

and

$$a_n \leftarrow (2/T)^n \, a_n \qquad [5]$$

Note that after the bilinear transform is evaluated, $M$ = N and the filter is now specified by a new set of $c_m, d_n$ coefficients. Therefore we seek methods that enable us to generate the $c_m, d_n$ coefficients in equation [3] from the $b_m, a_n$ coefficients in equation [1].

To obtain N(z) and *D(z),* consider the following definitions for just the denominator *D(z).* If *P(s) is* the denominator of the analog transfer function *H(s),* then

$$D(z) = (z+1)^N P\frac{z-1}{(z+1)} \qquad [6]$$

If the following definitions are applied

$$E(z) \triangleq P(z-1) \qquad [7]$$

$$F(z) \triangleq z^N E(1/z) \qquad [8]$$

$$G(z) \triangleq F(z+1/z) \qquad [9]$$

$$J(z) \triangleq z^N G(1/z) \qquad [10]$$

then

$$D(z) \triangleq J(2z) \qquad [11]$$

The preceding substitutions constitute the basis of the Simons-Harden algorithm for deriving a bilinear transformed *H(z).* In words, the algorithm consists of the following steps:

1. Translate the polynomial one unit to the right [DAZZ66].
2. Reverse the order of the coefficients.
3. Translate the polynomial one half unit to the left,
4. Reverse the order of the coefficients.
5. Replace the nth order coefficient $d_n$ with $d_n$ times 2“

This algorithm effectively transforms the denominator of the *H(s)* transfer function into the “z” domain. The process must be repeated for the numerator of the transfer function, which results in a ratio of two polynomials in “z”.

The authors assume that the order of the denominator of the *H(s)* transfer function is greater than that of the numerator so that the prototype filter is realizable. As mentioned earlier, when the bilinear transform is calculated, a ratio of two polynomials of equal order occurs. This happens because the bilinear transform automatically places the needed number of z = -1 zeros in the numerator to achieve this equality. If the analog filter is realizable, and the above algorithm is evaluated only for the numerator and denominator, N(z) and *D(z)* will be of different orders. Therefore to complete the bilinear transform formulation, one more step must be added. After the algorithm is applied to the numerator of the *H(s)* transfer function, the resulting numerator polynomial must be multiplied by $(z+1)^{N-M}$, where N is the order of the transformed denominator and $M$ is the order of the transformed numerator. After the bilinear transform of the analog filter is calculated, the next step in the design is to evaluate the frequency response of the resulting digital filter.

## Real Arithmetic *H(z)* Frequency Response Algorithms

Frequency response evaluation of *H(z)* models becomes an integral part of the design process when error analysis has to be performed. The frequency response of the calculated digital filter must be compared to the response of the ideal filter to determine the magnitude of error. If stopping conditions are met, the optimal filter has been determined; otherwise the direction of coefficient change for the next iteration must be calculated. In most cases, the computer program used for the purpose of frequency response evaluation will be based on some standard FFT algorithm which limits the evaluation to a discrete set of values. The algorithms used in this paper performs the frequency response evaluation using all real arithmetic and the user is not confined to a limited set of discrete evaluation points. In most cases the real arithmetic algorithms require less processing power than FFT based evaluation techniques [SIMO87]. It is assumed to be that the *H(z)* transfer-function form is

$$\frac{Y(z)}{X(z)} \triangleq H(z) = \frac{\sum_0^M c_m z^{-m}}{\sum_0^N d_n z^{-n}} \qquad [12]$$

since the bilinear transformation process yields exactly this form. In the evaluation of an *H(z)* frequency response,

$$z = \varepsilon^{j\omega t} = \cos(\omega T) + j\sin(\omega T) \qquad [13]$$

The magnitude of the response is calculated as a function of frequency ω. Each response calculation requires the evaluation of *H(z)* which is a complex function of a complex variable z. Due to the periodic nature of the $z = \varepsilon^{j\omega T}$ substitution, a complete frequency response evaluation is defined by the range of frequencies from O to $\omega_s/2$, provided the *H(z)* coefficients are real. With the definition

$$P(z) = \sum_{n=0}^N a_n z^n \qquad [14]$$

and $$(z-a-jb)(z-a+jb) \triangleq z^2 + \alpha z + \beta \qquad [15]$$

Clearly $$\alpha = -2\cos(\omega T) \quad \text{and} \quad \beta = 1 \qquad [16]$$

The division of $P(z)$ by $(z^2+\alpha z+\beta)$ can be put into the form

$$P(z) = (z^2+\alpha z+\beta)\sum_0^{N-2} b_n z^n + R_1 z + R_0 \qquad [17]$$

Since $z = \varepsilon^{j\omega T}$ is defined as a root of equation [15], then

$$P(\varepsilon^{j\omega T}) = R_1 \varepsilon^{j\omega T} + R_0 \qquad [18]$$

Thus, the polynomial evaluation can be expressed in terms of its real and imaginary parts as

$$R_e\left[P(\varepsilon^{j\omega T})\right] = R_0 + R_1\cos(\omega T) \text{ and } I_m\left[P(\varepsilon^{j\omega T})\right] = jR_1\sin(\omega T) \qquad [19]$$

This result implies that if we have a method of determining $R_1$ and $R_0$, then we have a method of determining the frequency response by evaluating the numerator and denominator of *H(z)*. Consider the form

$$\frac{P(z)}{z^2+\alpha z+\beta} \triangleq \sum_{-\infty}^{N-2} b_n z^n \triangleq \sum_{0}^{N-2} b_n z^n \text{-}t \frac{R_1 z + R_0}{z^2+\alpha z+\beta} \qquad \textbf{[20]}$$

from which

$$\frac{R_1 z + R_0}{z^2+\alpha z+\beta} = \sum_{n=-\infty}^{-1} b_n z^n \qquad \textbf{[21]}$$

Multiplying both sides by the denominator and equating equal powers of $z$ leads to

$$R_1 = b_{-1} \text{ and } R_0 = b_{-2} + \alpha\, b_{-1} \qquad \textbf{[22]}$$

Thus, with the evaluations of $R_1$ and $R_0$ revealed, the frequency response algorithm consist of evaluating the numerator and denominator of *H(z)* with the following steps:

Step 1: Evaluate $\alpha$ and $\beta$ from &i” [7]

Step 2: From the definition of the *P(z)* form given in equation [20], evaluate

$$b_{n-2} = a_n - \alpha\, b_{n-1} - \beta\, b_n$$

For $n = N, N-1, N-2, \ldots, 0$

with $b_N \triangleq O$ and $b_{N-1} \triangleq O$

Step 3: Evaluate $R_0, R_1 = (b_{-2} + \alpha\, b_1),\ b_1$

Step 4: Finally $P(\varepsilon^{j\omega T}) = R_0 + R_1\cos(\omega T) + jR_1\sin(\omega T)$

With these optimized algorithms on which the design process heavily relies, the complete optimal digital filter design process can be outlined.

## The Design of Optimum Digital Filters Based on Analog Prototypes

In order to best express the ideas developed for the implementation of the optimization process, a reference flow chart has been included at the end of the paper. Explanations on how each step contributes to the process follows.

The first step in the process is the evaluation of the initial guess analog prototype filter. There are a number of options available for analog prototype filter design but they are not treated in this paper since many basic DSP texts cover this topic[PROA88],[ DEFA88]. There are several methods of filter coefficient input, but the authors have chosen to enter the poles of the filter because of the ease of insuring stability.

The second step in the process is to use the bilinear transform to transform the filter from the continuous domain to the discrete domain. The optimized algorithms used to accomplish this task were presented earlier in detail. A frequency response of the digital filter is then performed and compared to the ideal response by calculating the Integral Square Error (ISE) of the differences between the two responses.

Then the gradient is evaluated based on small perturbations in the analog parameters and the magnitude of the ISE. The gradient calculation is an integral part of the parameter optimization process. In parameter optimization, we seek to find the combination of parameters that provides us with the smallest possible error.

Calculation of the gradient gives us an indication of the direction of maximum increase in error. With the gradient calculation completed, the negative of this calculation will give us the direction of maximum decrease from our present position. The process resembles successive parameter optimization in the sense that for each analog parameter changed, the effects of this change is measured by evaluating the ISE and noting the change in the magnitude of the error. Thus, each perturbation and error calculation is equivalent to the calculation of a partial of the error with respect to that parameter. Once the negative of the gradient is calculated from the collective parameter gradients, a good indication of which direction should be taken from the present point to arrive at the optimal parameter solution is known. This approach is termed the method of steepest descent.

Therefore, gradient calculations include calculation of the ISE at the present solution, calculation of ISE after each change, and the differences between the two errors. It should also be noted that each error calculation involves the bilinear transform, frequency response, and ISE calculation.

It was decided to initially limit the size of parameter step to O. 1°A of the average magnitude of the poles in order to avoid the chances of overshooting possible local solutions. After each gradient iteration, the magnitude of the error must be checked to determine if further calculation is needed. If the error criteria is met, the current filter is defined as the optimal solution. If the criteria is not met, further calculations are required.

In the flow chart diagram, the cost function terms $J_n$ and $J_i$ may cause some confusion. These two terms are used to denote two different concepts.

$J_n$ refers to the total gradient calculation for the $n^{th}$ time through the whole process whereas
$J_i$ refers to the component of the gradient that is due to the $i^{th}$ parameter.

In the error criteria iteration, the analog coefficients are changed as a function of error magnitude and the calculated gradient. However, the analog coefficients are in turn functions of pole positions, which defines the analog parameters that are actually iterated. With their direct implication on stability, pole positions are clearly the preferred optimization parameters. Thus pole position parameter optimization continues until the error criteria is met and the optimal solution is determined.

## Conclusions

The optimized algorithms on which the design process is based have been shown to be very powerful and effective in reducing computational requirements. The number of computations needed to perform the bilinear transform with the optimized algorithm has been shown with MATLAB simulations to be 10 times smaller that the number needed to perform the same transformation with direct computations [MED194]. It has also been shown that the frequency response algorithms used in the design reduce the computational requirements over other techniques in most cases [SIMO87]. Therefore the use of these algorithms in formulating this new procedure for designing optimal digital filters results in an extremely efficient process. Also, since parameter optimization techniques have been widely used and documented, the optimal filter design defined herein has a high probability of success and a strong possibility of wide acceptance in the field of digital signal processing.

## Future Research

In order to take full advantage of the algorithms, direct form filter implementations of the digital *H(z)* models are implied. The disadvantage of direct form structures is that they suffer great penalties due to quantization error. Therefore, new implementation algorithms will be sought to circumvent these errors.

## References


[SIMO88] Simons, Jr., F.O. and Harden, R. C., "An Optimized Simulation of Dynamic Continuous Models", Proceedings of the 96th Annual ASEE Conference, Portland, Oregon, June 19-23, 1988.

[SIMO87] Simons, Jr., F.O. and Harden, R. C., "FFT vs Real Argument Polynomial Algorithms for Evaluating H(z) Frequency Responses", Proceedings of the 19th Southeastern Symposium on System Theory, Clemson, SC, March 15-17, 1987.

[DAZZ66] D'Azzp, J.J. and Houpis, L. H., Feedback Control Systems Analysis and Synthesis, 2nd edition, McGraw-Hill Co., 1966

[ALKH86] Al-Khafaji, A.W. and Tooley, J. R., Numerical Methods in Engineering, CBS College Publishing, 1986.

[PROA88] Proakis, J.G. and Manolakis, D. G., Introduction to Digital Signal Processing, Macmillan Publishing Co., 1988.

[DEFA88] Defatta, D.J. et. al, Digital Signal Processing : A System Desire Approach, John Wiley and Sons Inc., 1988.

[MEDI94] Medina, J. Unpublished MATLAB work, HCS Research Lab, FAMU-FSU College of Engineering, 1994.

## AARON L. ROBINSON

Aaron was born on December 5, 1970 in Miami, Florida. He received his B.S. in Electrical Engineering in April 1994 from Florida State University. He should receive his M.S in Electrical Engineering at the end of the Summer 1996. He has served as a project leader for the DSP group in the FAMU-FSU College of Engineering High Performance Computing and Simulations Research Laboratory. He is a McKnight Fellow committed to finishing his Ph.D. degree and becoming a professor of Electrical Engineering.

## FRED O. SIMONS, JR.

Fred O. Simons, Jr. received his BSEE degree in August, 1960 from Mississippi State University, and his MS and Ph.D. degrees in electrical engineering from the University of Florida in 1962 and 1965, respectively. From 1965 to 1972 he was with the University of Florida. From September 1972 to September 1977 he was at the University of Central Florida. In 1978, after a year's leave of absence to direct a Navy Lab, he returned to UCF as a Professor. In 1984 he joined the FAMU/FSU College of Engineering as a Professor of Electrical Engineering. He also serves as the Associate Director of the *High-performance Computing and Simulation (HCS) Laboratory* and as the Director of FEEDS (Florida Engineering Education System) at Florida A&M University and Florida State University. In 1987-88 he was the Interim Chair at FAMU/FSU. He is a member of Tau Beta Pi, Phi Kappa Phi, Eta Kappa Nu, IEEE, SCS, FES and ASEE.

### Optimal Digital Filter Design Directly from Analog Prototypes Flow Chart