DIGITAL FILTER DESIGN FOR ACCELERATOR DATA AND CONTROL SYSTEMS*

by

Donald R. Machen

Clinton P. Anderson Meson Physics Facility
(LAMPF)
Los Alamos Scientific Laboratory
Los Alamos, New Mexico

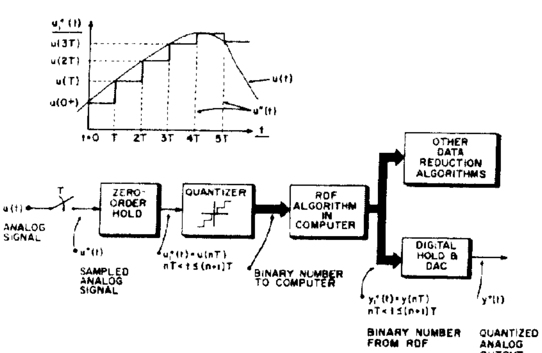
Introduction

The acquisition of real-time analog signals by a digital computer for the purpose of data reduction, supervisory control, or optimization, has become a common technique in today's particle accelerator laboratories. Signal variables of interest most generally contain unwanted noise spectra or are in need of bandwidth limiting prior to use by a computer algorithm.

The recursive digital filter[1,2] (RDF) is a useful method for generating algorithms through which quantized analog-signal sequences are altered with respect to their frequency spectrum. Rather simple filter algorithms can be developed that correspond to commonly used analog filters in both step-response and frequency-response characteristics. The development of the RDF state-equations will be discussed along with an example of filter algorithm design applicable to accelerator development and control problems. In addition, the attenuation and phase characteristics of the example RDF will be compared to that of the counterpart analog filter.

Formulation of the Recursive Digital Filter

The formulation of the RDF computational algorithm can best be understood by consideration of Figure 1. A simplified portion of a large data-acquisition and control system is depicted along with the signal variables needed in the development. In order to define a RDF that corresponds to the characteristics of the desired analog filter, it is necessary to incorporate the mathematical representation of the sampler and quantizer (multiplexer and ADC) together with the continuous state equations defining the desired analog filter. The process of quantizing the continuous equations of motion yields a discrete vector-matrix state-equation of the RDF that may be programmed on a digital computer. The RDF algorithm provides filtered digital-data for subsequent data reduction or control computations.



BLOCK DIAGRAM OF A COMPUTER SYSTEM
SHOWING RDF AND SIGNAL VARIABLES

Figure 1.

It is well known that many useful analog filter transfer relations may be represented by their vector-matrix state-equations.[3] The general state-equation form is:

$$\dot{\underline{x}}(t) = \underline{A}\ \underline{x}(t) + \underline{B}\ \underline{u}(t) \tag{1}$$

and

$$\underline{y}(t) = \underline{C}\ \underline{x}(t) + \underline{D}\ \underline{u}(t) \tag{2}$$

where

$\underline{x}(t)$ is the state vector (n x 1)
$\underline{u}(t)$ is the input vector (r x 1)
$\underline{y}(t)$ is the output vector (m x 1)

and

$\underline{A}$ is the System matrix (n x n)
$\underline{B}$ is the Control matrix (n x r)
$\underline{C}$ is the Output matrix (m x n)
$\underline{D}$ is the Coupling matrix (m x r)

A more common statement of the continuous state-equation is written in terms of a scalar input and a scalar output with no cross coupling term between input and output. The state representation is then:

$$\dot{\underline{x}}(t) = \underline{A}\ \underline{x}(t) + \underline{b}\ u(t) \tag{3}$$

$$y(t) = \underline{c}'\ \underline{x}(t) \tag{4}$$

($\underline{c}'$ indicates the transpose of $\underline{c}$)

and

$$\underline{x}(o) = \underline{o} \tag{5}$$

The scalar input is u(t) and the scalar output is y(t); $\underline{b}$ is a (n x 1) control vector and $\underline{c}$ is a (m x 1) coupling vector. Equation (5) indicates that the initial condition term is a null vector and may be so chosen without loss of generality. The elements of $\underline{A}$, $\underline{b}$, and $\underline{c}$ depend upon the particular analog filter chosen and the method selected for converting the filter differential equations to a state representation.

Solution of the State-Equation

The time-domain solution[3] of Equation (3) is:

$$\underline{x}(t) = e^{\underline{A}t}\underline{x}(o) + \int_o^t e^{\underline{A}(t-\tau)}\underline{b}\ u(\tau)d\tau \tag{6}$$

The matrix $e^{\underline{A}t}$ is known as the state-transition matrix and may be expressed as a power-series expansion of $e^{\underline{A}t}$. More usefully, $e^{\underline{A}t}$ may be found from the solution of Equation (3) in the frequency-domain:

$$\underline{x}(s) = (s\underline{I}-\underline{A})^{-1}\underline{x}(o) + (s\underline{I}-\underline{A})^{-1}\underline{b}\ u(s) \tag{7}$$

where s is the Laplace variable and $\underline{I}$ is the identity matrix.

Thus, using the notation $\underline{\Phi}(t)$ for the state-transition matrix, $e^{\underline{A}t}$ may be evaluated from:

$$\underline{\Phi}(t) = e^{\underline{A}t} = \mathcal{L}^{-1}(s\underline{I}-\underline{A})^{-1} \tag{8}$$

i.e., the inverse Laplace transformation of the inverse matrix, $(s\underline{I}-\underline{A})^{-1}$.

The Discrete State Equation

Referring to Figure 1, the sample period T, of the multiplex switch, is related to the desired filter break frequency, ω_o, by sampling theory.[4] Theoretically, $\omega_s = 2\pi/T$ whould be at least a factor of two greater

*Work performed under the auspices of the U. S. Atomic Energy Commission.

than the highest frequency present in the analog signal u(t). In practice it is advisable to have $\omega_s = 25\omega_o$ or greater, if possible. Unwanted sidebands due to the sampling process may invalidate the RDF results if the sampling frequency is not sufficiently greater than ω_o.

The discrete approximation, $u_1^*(t)$, to the sampled and held analog signal, u(t), is shown in Figure 1. The hold selected is the "zero-order" hold, in common usage in many present-day data-acquisition and control systems. If $u^*(t)$ is the sampled analog signal, then an approximation to the hold output is:

$$u_1^*(t) = u(nT),\ nT < t \le (n+1)T \tag{9}$$

The equation defines a "stair-step" response, changing value at each sample period, nT.

Also, $u_1^*(t) = u(o+)$ may be used in Equation (6) to determine the state-vector $\underline{x}(T)$ when a discrete input is applied for the duration of the initial sample period. For $o < t \le T$, Equation (6) is:

$$\underline{x}(T) = \underline{\Phi}(T)\ \underline{x}(o) + \int_o^T \underline{\Phi}(T-\tau)\ \underline{b}\ u(o+)d\tau$$

Then, for $nT < t \le (n+1)T$, it can be shown that:

$$\underline{x}(n+1)T = \underline{\Phi}(T)\ \underline{x}(nT) + \int_o^T \underline{\Phi}(T-\tau)\ \underline{b}\ u(nT)d\tau \tag{10}$$

The discrete form of the state-transition matrix and control vector may now be defined:

$$\underline{F} \triangleq \Phi(T) = \mathcal{L}^{-1}(s\underline{I}-\underline{A})^{-1}\Big|_{t=T} \tag{11}$$

and

$$\underline{g}_o \triangleq \int_o^T \underline{\Phi}(T-\tau)\ \underline{b}\ d\tau = A^{-1}(\underline{\Phi}(T)-\underline{I})\ \underline{b} \tag{12}$$

The primary RDF state-equation is then:

$$\underline{x}(n+1)T = \underline{F}\ \underline{x}(nT) + \underline{g}_o\ u(nT) \tag{13}$$

$$y(nT) = \underline{c}'\ x(nT) \tag{14}$$

and

$$\underline{x}(oT) = \underline{o} \tag{15}$$

The assumption made in Equation (13) is that the input signal, u(t), is sampled by a "near-impulse" sampler and held by the zero-order hold device. Other methods of sample and hold will produce a different $\underline{g}_o$ and input approximation, $u_1^*(t)$. The various matrix and vector elements are dependent upon the analog filter chosen, the sampling frequency, and the technique used to generate the continuous state-equation. The general form of Equations (13) and (14), however, is valid for any filter and state-representation.[5]

The RDF state-equation (13) may be further reduced[5] and combined with Equations (14) and (15) to yield the output for any time step nT, given the past (n-1)T input sequence.

$$y(nT) = \underline{c}' \sum_{m=0}^{n-1} \underline{F}^m\ \underline{g}_o\ u(n-m-1)T \tag{16}$$

Equation (16) requires the raising of a potentially high-order matrix, $\underline{F}$, to the power of m for $o \le m \le n-1$, and is not a preferred method for use in algorithm design. Equation (13), however, calls for a simple recursive computation for each state-variable as successive samples of the input signal are acquired by the data system.

In order to reduce quantization and round off error accumulation[6] in an RDF algorithm, all multiplication, addition, and subtraction should be implemented by double-precision arithmetic. Currently available hardware on the smallest minicomputer significantly reduces the time required for these operations.

RDF Formulation of a Two-Pole Filter

To illustrate the formulation of a RDF, a two-pole analog filter with gain K and break frequency ω_o will be considered. To simplify the presentation, the filter poles will be placed on the real-axis.

The Laplace transfer relation between input and output is:

$$\frac{y(s)}{u(s)} = \frac{K}{(s+\omega_o)^2} \tag{17}$$

The equivalent time-domain equation of motion may be written as:

$$\ddot{y}(t) + 2\ \omega_o\ \dot{y}(t) + \omega_o^2 y(t) = K\ u(t) \tag{18}$$

Selecting the phase-variables, y(t) and $\dot{y}(t)$, as the state-variables, $x_1(t)$ and $x_2(t)$, respectively, the state-equation is:

$$\underline{\dot{x}}(t) = \begin{bmatrix} 0 & 1 \\ -\omega_o^2 & -2\omega_o \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ K \end{bmatrix} u(t) \tag{19}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \underline{x}(t) \tag{20}$$

From Equation (11), the discrete state-transition matrix is:

$$\underline{F} = \begin{bmatrix} e^{-\omega_o T} + \omega_o T e^{-\omega_o T} & T\ e^{-\omega_o T} \\ -\omega_o^2 T e^{-\omega_o T} & e^{-\omega_o T} - \omega_o T e^{-\omega_o T} \end{bmatrix} \tag{21}$$

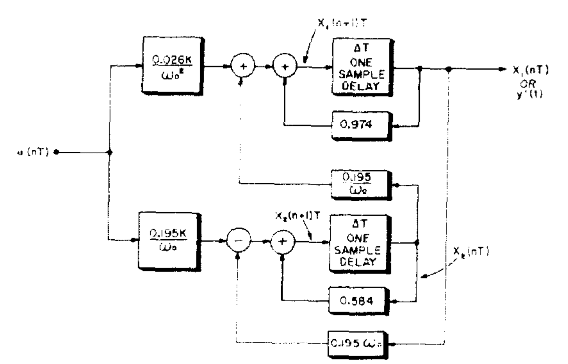
By selecting $\omega_s = 25\omega_o$ or $T \cong 1/4\omega_o$, Equation (21) reduces to:

$$\underline{F} = \begin{bmatrix} 0.974 & 0.195/\omega_o \\ -0.195\omega_o & 0.584 \end{bmatrix} \tag{22}$$

From Equation (12), the control vector is:

$$\underline{g}_o = \begin{bmatrix} 0.026K/\omega_o^2 & 0.195K/\omega_o \end{bmatrix}' \tag{23}$$

Figure 2 shows the flow diagram of the generalized two-pole RDF formulated in the above example. The RDF may be programmed on any digital computer with the gain K and break frequency ω_o set at will. The condition of importance that must be met is that the sample period T vary inversely with the break frequency ω_o, as indicated above. When the time required to recursively compute $y^*(t)$ equals the sample period T, the maximum bandwidth of the RDF will have been reached. Hardware multiply and divide along with efficient coding will allow the maximum bandwidth to be several hundred Hertz.



GENERALIZED TWO-POLE RDF

Figure 2.

Figure 3 indicates the measured and calculated frequency response of the RDF formulated in the example. The attenuation and phase are plotted against normalized frequency in radians per second. The consequences of having frequency components present in u(t) that are greater than the sampling frequency are demonstrated by the periodic frequency response of the RDF above $\omega_s/2$.

Great care must be exercised in the selection of ω_s and in the high frequency pre-filtering of u(t). These restrictions are not excessive however, as the usefulness of a RDF is in the implementation of low frequency, low-pass, and band-pass filters.

The periodic amplitude-response shown in Figure 3 indicates the frequency folding effect for input signal frequencies that are integer multiples of the sampling frequency, ω_s. As the input signal passes through an integer multiple of ω_s, the output, $y^*(t)$, will assume frequencies within the bandpass of the filter.

The phase-response shown in Figure 3 for frequencies above $\omega_s/2$ relates the phase of the (lower) folded-frequency signal to that of the (higher) signal input. Phase-lag below $\omega_s/2$ is the common phase-shift between two signals of the same frequency.

Both the frequency and phase-response of the two-pole RDF below the $10\omega_0$ are identical to that of the counterpart analog filter.

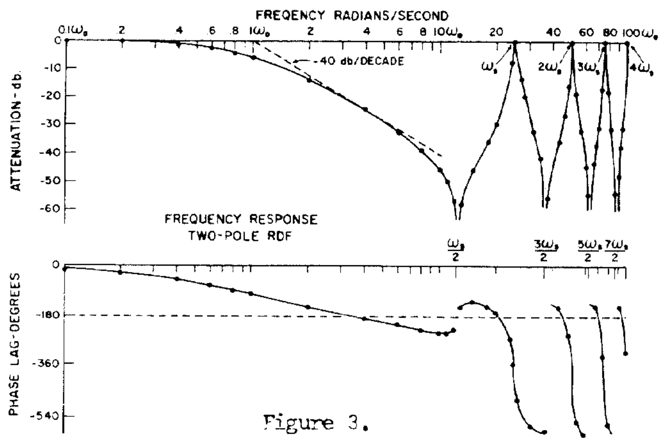


Figure 3.

Figure 4 shows the input and output signals of the two-pole RDF formulated in the example. The top trace is u(t), composed of signal plus noise.

$$u(t) = E_1 \sin \omega_1 t + E_2 \sin \omega_2 t$$

$$\omega_1 = \omega_0 \text{ and } \omega_2 = 10\omega_0.$$

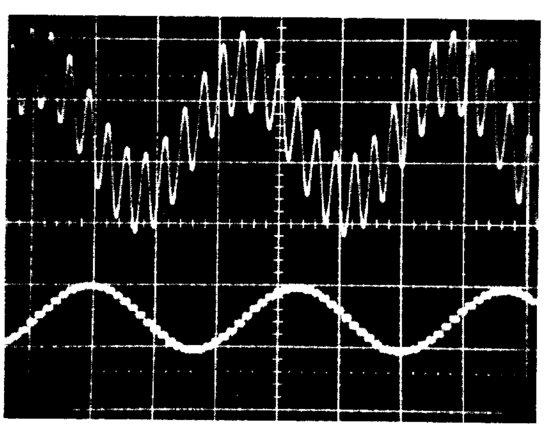


Fig. 4
Two-Pole RDF Input and Output Signals

The bottom trace is $y^*(t)$, the filtered output. The noise component has been removed and the $E_1 \sin \omega_1 T$ component of the input is passed, with an attenuation of 6 db. and a phase shift of 90°.

The example RDF was coded on a NOVA computer designed for data-acquisition and general accelerator development at LAMPF. Input and output quantization was implemented at 12 bits, while RDF computation was carried out in double-precision arithmetic.

Conclusions

The RDF, while not the only technique for digital filter design,[2] is conceptually elegant in theory and directly applicable to programming and use in a digital computer data-acquisition and control system.

Although the RDF exhibits a periodic frequency response above the sampling frequency, the match to the counterpart analog filter below the sampling frequency is excellent. In addition, the step response of the RDF matches exactly the counterpart analog filter at the sample points.

It should be noted that the RDF is useful at low frequencies (low-pass, band-pass) where an analog filter is physically impractical. Use of a RDF at high frequencies is constrained by readily available analog filters and the high required sample-frequency that could tend to demand near 100% of a digital computer's available time.

Acknowledgment

The stimulation and encouragement of Dr. R. A. Gore and Dr. H. B. Demuth of the Los Alamos Scientific Laboratory Staff are gratefully acknowledged.

References

1. Haykin, S. S., "A Unified Treatment of Recursive Digital Filters," IEEE Transactions on Automatic Control, pp. 113-116, February 1972.
2. Huelsman, L. P., Editor, "Active Filters," Chapter 5. (DIGITAL FILTERS, J. V. Wait), Mc-Graw-Hill, 1970.
3. Shultz and Melsa, "State Function and Linear Control Systems," McGraw-Hill, 1967.
4. Jury, "Sampled-Data Control Systems," Wiley, 1958.
5. Machen, D. R., "Digital Filters - The Digital Processing of Analog Signals," LAMPF Internal Report, MP-1-3001, 1971.
6. Machen, D. R., "Digital Filters - Quantization Errors," LAMPF Internal Report, MP-1-3002, 1971.