April, 1979

LIDS-P-897

THE DIGITAL IMPLEMENTATION OF CONTROL COMPENSATORS: THE COEFFICIENT WORDLENGTH ISSUE

by

Paul Moroney
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139 &
The Charles Stark Draper Laboratory, Inc.
555 Technology Square
Cambridge, Massachusetts 02139

Alan S. Willsky
Laboratory for Information and Decision Systems
Department of Electrical Engineering and
Computer Sciences
Cambridge, Massachusetts 02139

Paul K. Houpt
Laboratory for Information and Decision Systems
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract

There exists a number of mathematical procedures for designing discrete-time compensators. However, the digital implementation of these designs, with a microprocessor for example, has not received nearly as thorough an investigation. The finite-precision nature of the digital hardware makes it necessary to choose an algorithm (computational structure) that will perform 'well-enough' with regard to the initial objectives of the design. This paper describes a procedure for estimating the required fixed-point coefficient wordlength for any given computational structure for the implementation of a single-input single-output LQG design. The results are compared to the actual number of bits necessary to achieve a specified performance index.

* This work was performed in part at the MIT Laboratory for Information and Decision Systems with support provided by NASA Ames under grant NGL-22-009-124 and part at the Charles Stark Draper Laboratory.

1. Introduction

The design of discrete-time compensators through the use of optimal regulators, pole-placement concepts, observer theory, optimal filtering [1,2] and also via classical control theory [3] has received a great deal of attention in the literature. In the past such designs have usually been implemented on large, expensive, floating-point computer systems. However, the number of applications that could effectively use small-scale hardware control systems that work in real time has greatly increased, especially with the advent of the inexpensive microprocessor.

While the recent advances in digital hardware capabilities have opened many new possibilities for control system implementations, they have also raised new issues. A number of these involve the problems that arise in dealing with the fixed-point arithmetic and finite wordlengths (limited storage) of small-scale digital systems. As these problems are not addressed at all in the idealized mathematical design procedures that have been developed to date, a methodology must be established for treating the digital implementation of a design. The mathematical design procedure, only a first step, produces an infinite-precision compensator that is 'ideal,' at least with respect to all finite-precision implementations. The job of the implementation step will be to specify and order the critical computations that must take place in the compensator so that the end result (finite-precision) performs as close to the 'ideal' as is consistent with the expense and

and speed requirements of the application. The implementation step will also include a specification of the hardware architecture and components. It is important to note that the mathematical design and the implementation phases are not totally independent, since the implementation can be very important in determining an acceptable sampling rate and the number of operations that can be performed per sampling period.

Some effort has been directed to the implementation phase of an overall controller design, but it has been quite limited. Knowles and Edwards [4] have considered some roundoff noise questions for a specific classical controller design. Sripad [5] has looked at the roundoff noise and coefficient sensitivity of the Kalman filter. Rink and Chong [6] have derived bounds on the quantization error in floating point regulators.

Our approach will draw on the field of digital signal processing [7], which has generated many results concerning the realistic implementation of digital filters. The finite precision effects of coefficient quantization, limit cycles, and quantization noise have been reviewed (for filters) in [8], [9], and [10]. These results are very important for control applications, since a control system can be viewed as a digital filter (compensator) imbedded in a feedback loop through a plant. Our work is a first step towards bridging the gap between the digital filtering results (no external feedback) and the ideal controller design procedures. This paper bring the techniques for digital filter implementation to bear on the fixed-point compensator coefficient wordlength issue.

Approximating the coefficients of an implementation with a finite number of bits will cause a degradation in the system's performance as compared to the ideal. Assuming that a given quantitative performarce measure is provided, we can measure the tradeoff in the number of bits vs. the degradation. Then, assuming that we specify an acceptable amount of degradation, one must determine the minimum number of coefficient bits needed to meet this goal. Clearly a straightforward way to determine this wordlength is to simply reevaluate the measure of performance over a number of different rounded wordlengths, and to choose the smallest wordlength meeting the design specification. This brute force method can be quite time-consuming. The concept of a (simpler) statistical estimate of the wordlength originated in the study of digital filters with the work of Knowles and Olcayto [11]. Avenhaus [12] applied this idea to the digital filter power transfer function (as a performance measure), and later Crochiere [13,14] used the concept with the filter transfer function magnitude and a wordlength optimization procedure. All three of these studies chose dif- different performance measures, none of which seem to be particularly appropriate for control problems where the compensator phase is critical. In this paper we will adapt the statistical wordlength concept to the steady-state linear-quadratic-Gaussian (LQG) control problem, using the

LQG penalty function as our measure of performance. After discussing the LQG configuration, a notation for specifying different implementations will be presented, followed by the actual statistical wordlength procedure. Examples demonstrating this procedure and comparing it to the brute-force method follow.

2. The LQG Controller Problem

This section will present the single-input single-output LQG control configuration and the mathematical (ideal) design of the compensator. The discretized plant equations are described as follows (assume a given sample rate):

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) + w_1(k) \\ y(k) &= Lx(k) + w_2(k) \end{aligned} \qquad (1)$$

where $\Phi(nxn)$ is the transition matrix, $\Gamma(nx1)$ and $L(1xn)$ are the input and output gains, and w_1 and w_2 are discrete white Gaussian noise sequences with covariance matrices $\Theta_1(nxn)$ and $\Theta_2(1x1)$ respectively. The control law is chosen to minimize the following performance index: (the discretized version of a continuous-time performance index)

$$J = E\left\{\lim_{i\to\infty} \frac{1}{2i} \sum_{k=-i}^{+i} (x'(k)Qx(k) + x'(k)Mu(k) + Ru^2(k))\right\} \qquad (2)$$

where Q is nxn, M is nx1, and R is 1x1. The result is the following regulator/Kalman filter compensator:

$$\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma\hat{u}(k) + K(y(k) - L\hat{x}(k))$$
$$u(k) = -G\hat{x}(k) \tag{3}$$

Note that the equations in (3) base the current control u(k) only on *past* outputs y(k-1), y(k-2) ,...,[1]. A real compensator (one that can be implemented) cannot allow u(k) to depend on y(k), since a finite amount of computation time must elapse before y(k) can affect the output u. Thus y(k) can affect u(k+1) but *not* u(k).

The gains G(1xn) and K(nx1) can be found by solving the following two algebraic Ricatti equations [1]:

$$P = (\Phi-\Gamma R^{-1}M')'P\{I-\Gamma(R+\Gamma'P\Gamma)^{-1}\Gamma'P\}(\Phi-\Gamma R^{-1}M') + Q-MR^{-1}M'$$
$$\Sigma = \Phi\{I - \Sigma L'(\Theta_2+L\Sigma L')^{-1}L\}\Sigma\Phi' + \Theta_1 \tag{4}$$

and

$$G = (R+\Gamma' P\Gamma)^{-1}\Gamma'P(\Phi-\Gamma R^{-1}M') + R^{-1}M' \tag{5}$$
$$K = \Phi\Sigma L'(\Theta_2+L\Sigma L')^{-1}$$

Figure 1 presents a simple block diagram of the system and its (infinite-precision) compensator. This ideal compensator (3) can be described by an infinite-precision map (transfer function) in the digital frequency domain:

$$\frac{U(z)}{Y(z)} = -G(z - \Phi+KL + \Gamma G)^{-1}K \tag{6}$$

The digital filter transfer function (6) must be implemented in finite precision with as little degradation in some system performance measure as possible. In the setting of a steady-state LQG problem, it is convenient to select the performance index J in (2) as the measure of performance, since it reflects the weighted steady-state RMS state and control fluctuations. It would also have been possible to choose a criterion such as phase margin, output noise power, or any combination of stability or noise measures. If the problem under consideration was simply a Kalman filter, then a suitable performance measure would be the trace of the error covariance matrix. We have chosen J in order to present our results in a specific context. These results extend in a simple and direct fashion to other measures. It should also be noted that the selection of a single-input single-output system has only been done for convenience, and the following analysis can be easily extended to the multiple-input multiple-output case.

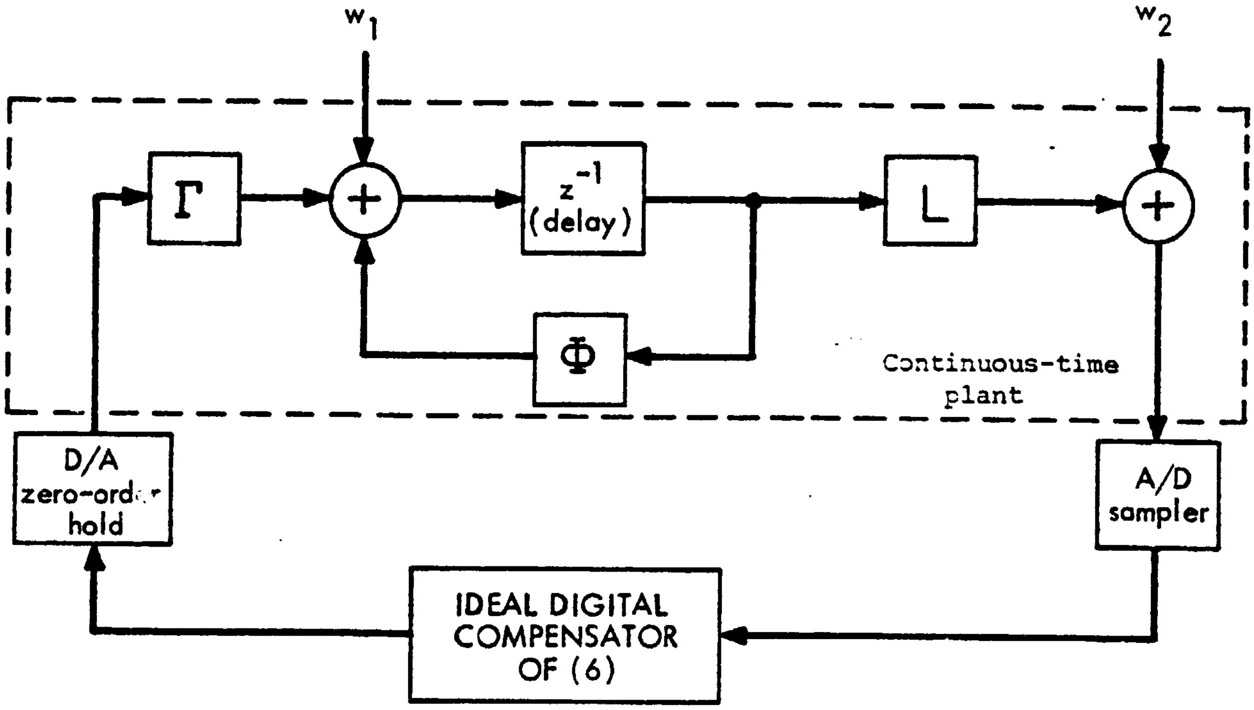


Figure 1: Plant & Compensator

3. Algorithms and Structures

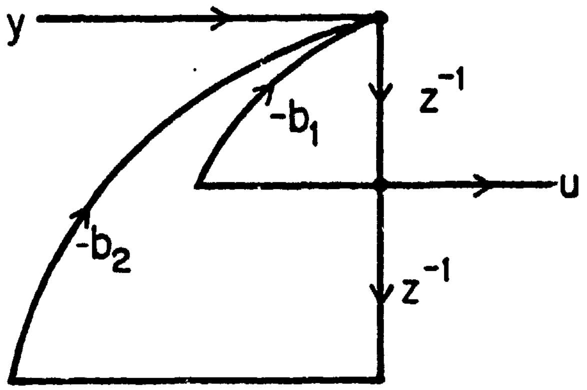
In order to discuss different implementations, one must have an accurate notation that reflects these differences. The term 'algorithm' or 'structure' will be employed to specify the exact finite-precision procedure by which the compensator output samples u are generated from its input samples y. All structures for implementing a given filter or compensator would perform identically under infinite-precision arithmetic, but will produce different quantization noise, coefficient quantization effects, and limit cycles given the (realistic) finite-precision environment. A good review of some of the structures used to implement digital filters can be found in [14], [15], and [16].

In order to demonstrate the finite-precision effects of different structures, consider the following example. Assume that an ideal compensator has been designed, and that its (infinite-precision) transfer function is given in (7).

$$\frac{U(z)}{Y(z)} = \frac{z^{-1}}{1+1.11z^{-1}+0.287z^{-2}} \tag{7}$$

One possible structure for implementing this filter is the direct form II [7]. Figure 2-A shows a signal flow graph of this filter where the coefficients b_1 and b_2 can be read directly from (7), the unfactored ideal transfer function. Given finite precision fixed-point arithmetic (say 10 bits total per coefficient), the ideal

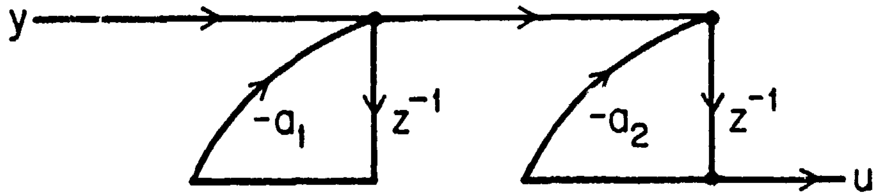
A: <u>Direct Form II</u>



b_1: 1.11 (ideal, ∞ bits)
1.109375 (10 bits)

b_2: 0.287 (ideal)
0.28515625 (10 bits)

B: <u>Cascade Form</u>



a_1: 0.41 (ideal)
0.41015625 (10 bits)

a_2: 0.70 (ideal)
0.69921875 (10 bits)

<u>Figure 2</u>: Example Structures

coefficients values of b_1 and b_2 must be quantized (assume rounding). Reserving two bits for the integral portion of the coefficient word (bits to the left of the binary point) and 8 bits for the fractional portion, the rounded coefficient values would be 1.109375 and 0.28515625.

Figure 2-B shows the flow graph of another common structure, the cascade form. Here we realize (7) by a series cascade of two first-order filter sections. The coefficients a_1 and a_2 can be found by factoring the denominator of (7). Again, the ideal values must be rounded to fit 10 bit words, producing a_1=C.69921875 and a_2=0.41015625.

Now let us examine the performance of these two structures given their respective finite-precision coefficients. The (10-bit) direct form II and the cascade have the transfer functions shown in (8) and (9) respectively.

$$\frac{U(z)}{Y(z)} = \frac{z^{-1}}{1+1.109375z^{-1} + 0.28515625z^{-2}} \tag{8}$$

$$\frac{U(z)}{Y(z)} = \frac{z^{-1}}{1+1.109375z^{-1} + 0.2867889404296875} \tag{9}$$

Clearly these two structures produce slightly different transfer functions under finite precision, and we have not even mentioned their respective quantization noise and limit cycle behavior. Thus different structures will in general result in different finite-precision performance even though their infinite-precision counterparts have equivalent performance (that of the ideal design)

In order to deal with these different structures, it is important to have an accurate way in which to represent the operations involved. The modified state-space of Chan [15] is the most convenient method. Consider a filter (compensator) with input y, output u, and state vector v. Then the coefficierts and the sequence of multiplies and critical additions in any structure can be specified with the following representation:

$$\begin{bmatrix} v(k+1) \\ u(k) \end{bmatrix} = \psi_q \psi_{q-1} \cdots \psi_1 \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} \qquad (10)$$

Two important points make (10) useful:

(1) Each (rounded) coefficient in the structure occurs once and only once as an entry in one of the ψ_i matrices. The remainder of the matrix entries are ones and zeros.

(2) The concept of a precedence to the operations (multiplies, adds, and quantizations) is maintained. The ordering of the ψ matrices implies that the operations in computing $\psi_1 \begin{bmatrix} v(k) \\ y(k) \end{bmatrix}$ are completed first, then $\psi_2 \left[\psi_1 \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} \right]$ next, and so forth. The parameter q specifies the number of such precedence levels.

Consider the example of the last section. The direct form II in Figure 2-A has a one-level modifed state space representation as shown in (11), while the cascade (12) requires two levels to describe its operations (even though its multiplies can be confined to one level):

$$\begin{bmatrix} v(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -b_2 & -b_1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} \qquad (11)$$

$$\begin{bmatrix} v(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -a_1 & 0 & 1 \\ 0 & -a_2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} \qquad (12)$$

Thus any two structures will have associated with them two different sets of ψ matrices. Let the coefficients in each ψ_i matrix be replaced by their infinite-precision counterparts (their values before rounding), and define ψ_∞ to be the infinite-precision product $\psi_q\psi_{q-1} \cdots \psi_1$. This matrix ψ_∞ will then be identical (within a similarity transform) for all structures - it depends strictly on the ideal design and choice of states v(k). It is also notationally convenient to partition ψ_∞ into a state-space representation, with no feedthrough term. (see Section 2.)

$$\begin{bmatrix} v(k+1) \\ u(k) \end{bmatrix}_{\infty} = \begin{bmatrix} \psi_{11} & \psi_{12} \\ \psi_{21} & 0 \end{bmatrix} \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} \qquad (13)$$

Thus (13) represents the ideal compensator's input-output behavior in a state-space form, and any factorization (10) of ψ_{∞}, with the resulting coefficients rounded, will represent a specific finite precision structure for implementing the ideal transfer function.

4. Statistical Wordlength

The need for a coefficient wordlength estimate is twofold. First, the computation of an estimate should be simpler than directly evaluating the performance measure over and over as the number of coefficient bits is varied. More importantly, if the estimate is continuous in nature (not confined to integral numbers of bits) then it is possible to apply simple optimization techniques to synthesize better structures. The statistical wordlength estimate can fulfill both these aims.

The remainder of this section will review the basic development of the statistical wordlength measure. [13] Consider a general measure of performance f. With a finite-precision implementation, the resulting f will then depend on the coefficients $(c_1, c_2, \ldots, c_m)$ of the structure. The value of f associated with any particular finite-precision structure will reflect a degradation in performance as compared to the ideal (unrounded coefficients) case f_∞. This degradation df can be expanded in a Taylor's series about the ideal value. To first order

$$df(c_1,c_2,\ldots,c_m) \approx \sum_{i=1}^{m} \left(\left.\frac{\partial f}{\partial c_i}\right|_\infty dc_i \right) \qquad (14)$$

where c_i is the i^{th} coefficient to be rounded, dc_i is the error due to quantization, and $\left.\frac{\partial f}{\partial c_i}\right|_\infty$ is the first partial derivative of f evaluated with the unrounded coefficient values. Note that coefficients such as $3,2,1,\frac{1}{2}\ldots.$ are not normally affected by rounding and should not be included in the sum (14).

If Δ is the quantization step size (the fraction represented by the least significant bit of the fixed-point coefficient word), then each dc_i must lie between $\pm\frac{\Delta}{2}$ (rounding assumed). Given the partial derivatives in (14), we could then (upper) bound the error df, producing a very pessimistic wordlength estimate.

The basic statistical wordlength idea is to treat an ensemble of structures. Over this ensemble, the coefficient errors dc_i can be described as uniformly-distributed zero-mean uncorrelated random variables, each of variance $\Delta^2/12$. The error df is therefore also zero-mean, with a variance:

$$\sigma_{df}^2 = \frac{\Delta^2}{12} \sum_{i=1}^{m} \left(\frac{\partial f}{\partial c_i} \bigg|_{\infty} \right)^2 \tag{15}$$

For large m, the central limit theorem can be applied to justify a Gaussian distribution for df. Thus with a given confidence level (probability), say 95%, one can predict the variance σ_{df}^2 needed for the error df to remain within some prescribed bound. In other words, 95 out of 100 of the structures in the ensemble will result in systems where df remains within this bound.

From a table of the Gaussian distribution,

$$\Pr[|df| \leq 2\sigma_{df}] = .954 \tag{16}$$

If the quantity of interest f is constrained to lie within ± E_o of the ideal $f_{\infty,}$ then (16) implies that σ_{df} equal $E_o/2$. This result can be combined with (15) to produce an estimate of the parameter Δ:

$$\Delta = \frac{\sqrt{3}\ E_o}{\sqrt{\sum_{i=1}^{m} \left(\frac{\partial f}{\partial c_i} \bigg|_{\infty} \right)^2}} \tag{17}$$

Given Δ, the statistical wordlength can be defined to be

$$\text{SWL} = \ell + \log_2 \frac{1}{\Delta} \tag{18}$$

The first term in (18) represents the number of bit necessary to represent the integer portion of the coefficients (bits to the left of the fixed-point binary word 'decimal point') and the second term gives the number of bits necessary for the fractional portion of the coefficient word (bits to the right of the binary point).

In the digital filter area, Crochiere [13,14,16] presents a number of results comparing the statistical wordlength of structures using the transfer function magnitude as the performance measure f. Since this choice of f is frequency-dependent, the resulting estimate is also frequency-dependent. The final wordlength can be selected as the maximum of the estimates over the frequency range of interest. In the examples treated by Crochiere, the statistical wordlength estimate was always 1 to 3 bits conservative as compared to the actual minimum number c˜ bits necessary to just meet the transfer function error limit. In a related work by Chan and Rabiner [17], which considered a large number of finite-impulse-response filters and a similar statistical approach to coeff'.cient wordlength, the resulting 95% confidence level estimates were also observed to be conservative. Crochiere [13] :was also able to use statistical wordlength as the basis for an optimization procedure involving the filter-specification filter-order tradeoff.

5. Statistical Wordlength and the Performance Index J

As mentioned above, it is convenient to use the performance index J in (2) as the measure of performance f in an LQG setting. Using the approach of the previous section, the change in J would be estimated by:

$$dJ(c_1,c_2,\ldots,c_m) \approx \sum_{i=1}^{m} \left(\left.\frac{\partial J}{\partial c_i}\right|_{\infty} dc_i \right) \tag{19}$$

However, the optimal nature of the LQG control problem forces all the sensitivities $\frac{\partial J}{\partial c_i}$ to be zero. Therefore a higher-order approximation is necessary:

$$dJ \approx \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\left.\frac{\partial^2 J}{\partial c_i \partial c_j}\right|_{\infty} dc_i dc_j \right) \tag{20}$$

The use of second-order terms (not seen in digital filter analysis) will make the statistical wordlength expression for LQG compensators unique, and as will be shown, quite complex to compute.

Proceeding from (20), the mean of dJ will no longer be zero:

$$E(dJ) = \sum_{i=1}^{m} \left.\frac{\partial^2 J}{\partial c_i^2}\right|_{\infty} E[(dc_i)^2] \tag{21}$$

For convenience, define the random variable ϵ to be the square of dc_i. Its mean and variance can be shown to be $\Delta^2/12$ and $\Delta^4/180$. The second moment and variance of dJ can now be found:

$$E[(dJ)^2] = \overline{e^2} \sum_{i=1}^{m} \left(\frac{\partial^2 J}{\partial c_i^{\,2}} \bigg|_{\infty} \right) + (\overline{e})^2 \sum_{\substack{i=1 \\ i \neq k}}^{m} \sum_{k=1}^{m} \left(\frac{\partial^2 J}{\partial c_i^{\,2}} \bigg|_{\infty} \right) \left(\frac{\partial^2 J}{\partial c_j^{\,2}} \bigg|_{\infty} \right)$$

$$+ 2(\overline{e})^2 \sum_{\substack{i=1 \\ i \neq j}}^{m} \sum_{j=1}^{m} \left(\frac{\partial^2 J}{\partial c_i \partial c_j} \bigg|_{\infty} \right)^2 \qquad (22)$$

$$\sigma_{dj}^2 = \sigma_{\epsilon}^2 \sum_{i=1}^{m} \left. \frac{\partial^2 J}{\partial c_i^{\,2}} \right|_{\infty}^{2} + 4(\overline{e})^2 \sum_{\substack{i=1 \\ i > j}}^{m} \sum_{j=1}^{m} \left(\frac{\partial^2 J}{\partial c_i \partial c_j} \bigg|_{\infty} \right)^2 \qquad (23)$$

The same Gaussian assumption and confidence level approach can be applied to this higher-order formulation, as shown in Figure 3. Since the value of J can only increase under coefficient quantization, we need only have a specification on its maximum allowed value $J_\infty + E_o$. If we choose two standard deviations around the mean, then we can write

$$J_\infty + E_o = J_\infty + \overline{dJ} + 2\sigma_{dJ} \qquad (24)$$

This choice of σ_{dJ} gives a 97.5% confidence level in terms of remaining below the allowed deviation E_o. Combining (22), (23), and (24) we can derive an expression for Δ^2:

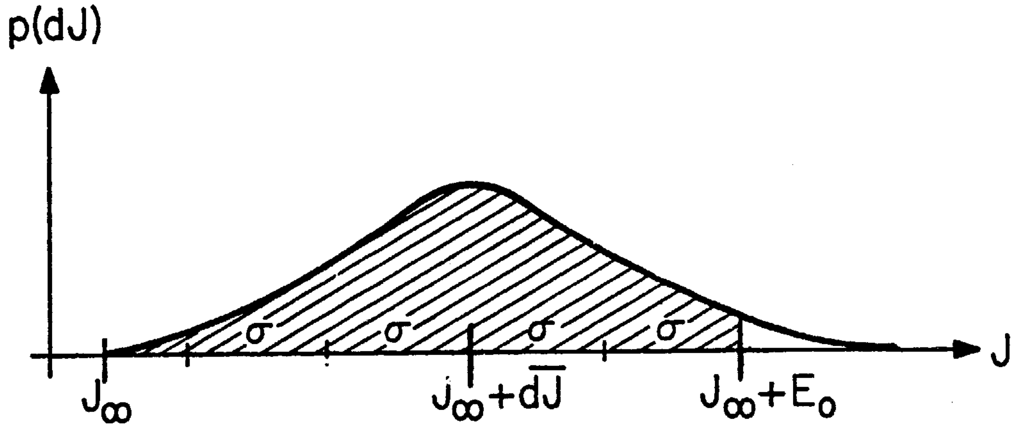


Figure 3: Probability Density of dJ

$$\Delta^2 = \frac{1}{3E_o}\sqrt{\sum_{\substack{i=1\\i>j}}^{m}\sum_{j=1}^{m}\left(\left.\frac{\partial^2 J}{\partial c_i\,\partial c_j}\right|_\infty\right)^2 + \frac{1}{5}\sum_{i=1}^{m}\left(\left.\frac{\partial^2 J}{\partial c_i^{\,2}}\right|_\infty\right)} + \frac{1}{12E_o}\sum_{i=1}^{m}\left(\left.\frac{\partial^2 J}{\partial c_i^{\,2}}\right|_\infty\right) \qquad (25)$$

Using (18), the SWL can then be written:

$$\mathrm{SWL} = \ell + \frac{1}{2}\log_2\frac{1}{\Delta} \qquad (26)$$

The use of second-partial derivatives in approximating dJ in (20) has given rise to a complex expression for the statistical wordlength. Efficient methods for evaluating (26) will be discussed in the next section.

6. Computational Procedure

In order to compute the derivatives of J_∞, the infinite-precision (ideal) performance index, it is convenient to use the trace form of equation (2): [18]

$$J_\infty = \mathrm{trace}\,[S\ Z] \qquad (27)$$

The 2nx2n matrices S and Z are defined by (28) and (29):

$$S = \begin{bmatrix} Q & M\psi_{21} \\ \psi_{21}'M' & \psi_{21}'R\psi_{21} \end{bmatrix} \qquad (28)$$

$$Z = E\left\{\begin{bmatrix} x(k) \\ v(k) \end{bmatrix} [x'(k) \quad v'(k)]\right\} \tag{29}$$

where Q,M, and R are the performance index parameters described in (2) and ψ_{21} is the lower left-hand portion of ψ_∞ as described in (13). The matrix Z, the covariance matrix for plant and compensator states, can be shown to satisfy the following Lyapunov equation:

$$Z = AZA' + \begin{bmatrix} \Theta_1 & 0 \\ 0 & \psi_{12}\Theta_2\psi_{12}' \end{bmatrix} \tag{30}$$

where

$$A = \begin{bmatrix} \Phi & \Gamma\psi_{21} \\ \psi_{12}L & \psi_{11} \end{bmatrix}$$

Note that (27)-(30) depend on the infinite-precision (ideal) compensator and on the selection of compensator state variables v. The resulting J_∞ will be independent of structure. However, the partial derivatives of J_∞ (evaluated for ideal coefficients) will depend on the structure since each coefficient c_i resides in one of the structure's ψ_i matrices. Taking the partial derivatives of (27) will produce: (assume all partials are evaluated at the ideal values of the coefficients)

$$\frac{\partial^2 J}{\partial c_i \partial c_j} = \text{trace}\left(\frac{\partial^2 S}{\partial c_i \partial c_j} Z\right) + \text{trace}\left(\frac{\partial S}{\partial c_i}\,\frac{\partial Z}{\partial c_j} + \frac{\partial S}{\partial c_j}\,\frac{\partial Z}{\partial c_j}\right) + \text{trace}\left(S\,\frac{\partial^2 Z}{\partial c_i \partial c_j}\right) \tag{31}$$

At first glance, (31) represents a great deal of computation. The first term requires the solution of (30) for Z. However, the second trace term involves the first partial derivatives of Z:

$$\frac{\partial Z}{\partial c_i} = A\,\frac{\partial Z}{\partial c_i}\,A' + \tilde{Q}_i \tag{32}$$

where

$$\tilde{Q}_i = \frac{\partial A}{\partial c_i}\,ZA' + AZ\,\frac{\partial A'}{\partial c_i} + \begin{bmatrix} 0 & 0 \\ 0 & \left(\frac{\partial \psi_{12}}{\partial c_i}\,\Theta_2\psi'_{12} + \psi_{12}\Theta_2\,\frac{\partial \psi'_{12}}{\partial c_i}\right) \end{bmatrix}$$

Evaluation of the second trace term in (31) for all i or j will imply solving m Lyapunov equations of the form shown in (32). The final term of (31) requires second partials of Z:

$$\frac{\partial^2 Z}{\partial c_i \partial_{c_j}} = A\,\frac{\partial^2 Z}{\partial c_i \partial c_j}\,A' + \tilde{C}_{ij} \tag{33}$$

where the 2nx2n matrix $\tilde{C}_{ij}$ involves partial derivatives of A, Z, and ψ_∞ with respect to the i^{th} and j^{th} coefficients. Solving (33) for all i and j would require m(m+1)/2 more Lyapunov solutions.

Fortunately, this burden can be substantially reduced. Specifically, the concept of adjoint operators can be used [1] to simplify the last term of (31). If we take the trace of the product of two matrices to be an inner product on the space of matrices, and $\mathcal{L}$ to be a matrix operator, then:

$$\mathrm{trace}(\mathcal{L}(X)\ U) = \mathrm{trace}(X \quad \mathcal{L}^*(U)) \tag{34}$$

where $\mathcal{L}^*$ is the adjoint operator of $\mathcal{L}$. For $\mathcal{L}(X) = X-AXA'$, the operator $\mathcal{L}^*$ can be derived from (34):

$$\begin{aligned}\mathrm{trace}((X-AXA')\ U) &= \mathrm{trace}(XU) - \mathrm{trace}(AXA'U)\\ &= \mathrm{trace}(XU) - \mathrm{trace}(XA'UA)\\ &= \mathrm{trace}(X(U-A'UA))\end{aligned} \tag{35}$$

Thus $\mathcal{L}^*(u) = U-A'UA$. This adjoint operator can be used to simplify the last term of (31) if X is $\frac{\partial^2 Z}{\partial c_i \partial c_j}$ and $\mathcal{L}^*(U)$ equals S. Since $\mathcal{L}\left(\frac{\partial^2 Z}{\partial c_i \partial c_j}\right)$ equals $\tilde{C}_{ij}$, we can rewrite (31):

$$\frac{\partial^2 J}{\partial c_i \partial c_j} = \mathrm{trace}\left(\frac{\partial^2 S}{\partial c_i \partial c_j} Z\right) + \mathrm{trace}\left(\frac{\partial S}{\partial c_i}\ \frac{\partial Z}{\partial c_j} + \frac{\partial S}{\partial c_j}\ \frac{\partial Z}{\partial c_i}\right) + \mathrm{trace}(U\tilde{C}_{ij}) \tag{36}$$

where U satisfies U-A'UA=S. Thus the last term of (36) requires only one Lyapunov solution.

There is still the problem of the m Lyapunov solutions needed in term 2. By using the Lyapunov solution method of Barraud [19], this computation can also be simplified. Consider the general Lyapunov equation (37):

$$X = FXF' + C \tag{37}$$

Barraud's method breaks into two distinct parts, one which transforms F into the upper Schur form, and one which back substitutes using the transformed F matrix and C. The major portion of this computation involves the initial F transformation. Thus, if there exists several Lyapunov equations with identical F matrices but different C matrices, then the F transformation need be done only once. This is exactly the situation for the Lyapunov equations (30) and (32) needed for the first two terms of (31). Typically, 50-90% of the Lyapunov computation time can be saved, depending on the particular matrices.

Further computational time savings are possible. Certain partial derivatives involved in the $\tilde{Q}_i$, $\tilde{C}_{ij}$, and $\frac{\partial \varepsilon}{\partial c_i}$ expressions are known to be zero and need not be computed. As an example, the term $\frac{\partial^2 \psi_\infty}{\partial c_i \partial c_j}$ <u>must</u> be zero if the i^{th} and j^{th} coefficients are in the same precedence level. Suppose ψ_∞ equals $\psi_1\psi_2\psi_3$ (three precedence levels exist). The nature of the modified state-space representation guarantees that each of these coefficients may be a single entry in only one precedence level. Assume that c_i and c_j are both in ψ_2.

Taking the partial derivative with respect to c_i:

$$\frac{\partial \psi_\infty}{\partial c_i} = \psi_1 \frac{\partial \psi_2}{\partial c_i} \psi_3 \qquad (38)$$

The matrix $\frac{\partial \psi_2}{\partial c_i}$ must be an all-zero matrix excluding a single unit entry at the same location as c_i in ψ_2. The expression in (38) is now independent of c_j, implying that $\frac{\partial^2 \psi_\infty}{\partial c_j \partial c_i}$ equals zero.

The specific details of the computational procedure (heavily involving the use of trace identities to simplify expressions) and the program itself can be found in the appendices of [20].

7. An LQG Example

The following sixth-order example was chosen to test the statistical wordlength algorithm. It is adapted from the bngitudinal control system design done for the F8 digital fly-by-wire flighter [21].

Continuous Time System Parameters:

$$A = \begin{bmatrix} -0.6696 & 5.7\text{x}10^{-4} & -9.01 & 0 & -15.77 & 0 \\ 0 & -0.01357 & -14.11 & -32.2 & -0.433 & 0 \\ 1 & -1.2\text{x}10^{-4} & -1.214 & 0 & -0.1394 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -12 & 12 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$$

$$C = [\,1 \quad 0.003091 \quad 31.28 \quad 1 \quad 3.592 \quad 0]$$

Continuous-Time Performance Index Parameter:

$$\hat{Q} = \begin{bmatrix} 6.637 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.6554\text{x}10^{-7} & 2.686\text{x}10^{-3} & 0 & 3.085\text{x}10^{-4} & 0 \\ 0 & 2.686\text{x}10^{-3} & 27.174 & 0 & 3.121 & 0 \\ 0 & 0 & 0 & 27.174 & 0 & 0 \\ 0 & 3.085\text{x}10^{-4} & 3.121 & 0 & 0.3585 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\hat{R} = 5.252$$

Continuous-Time Noise Covariances

$$\Xi_1 = \text{diag}[0 \quad 0 \quad 0 \quad 0 \quad 10^{-6} \quad 10^{-6}]$$

$$\Xi_2 = 0.00368825$$

This continuous-time system was discretized at a sample rate of 10 HZ and the optimal regulator and Kalman filter designed. The double-precision parameters Φ, Γ, L, Q, M, R, Θ_1, Θ_2, G, and K can be found in [20].

Four structures for implementing the ideal compensator transfer function (6) were examined. The first three are regular filter structures - the direct form II, the cascade form, and the parallel form. The coefficients of the direct form II structure (recall Figure 2) come directly from the unfactored transfer function (39); the 12 coefficients and one precedence level ψ_1 are shown in (40).

$$H(z) = \frac{a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4}+a_5z^{-5}+a_6z^{-6}}{1+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}+b_5z^{-5}+b_6z^{-6}} \tag{39}$$

$$\psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -b_6 & -b_5 & -b_4 & -b_3 & -b & -b_1 & 1 \\ a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & 0 \end{bmatrix} \tag{40}$$

The actual a_i and b_i values, and the ideal coefficient values for the other 3 structures can be found in [20].

The second structure, the cascade (see Figure 2), derives its coefficients from a multiplicative factorization of (39) and breaks into 3 series direct form II second-order sections. The factored transfer function (twelve coefficients) and the two precedence level matrices ψ_1 and ψ_2 are shown in (41) and (42):

$$H(z) = \frac{(r_1z^{-1}+r_2z^{-2})(1+r_3z^{-1}+r_4z^{-2})(1+r_5z^{-1}+r_6z^{-2})}{(1+c_1z^{-1}+c_2z^{-2})(1+c_3z^{-1}+c_4z^{-2})(1+c_5z^{-1}+c_6z^{-2})} \tag{41}$$

$$\psi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{42}$$

$$\psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -c_2 & -c_1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ r_2 & r_1 & -c_4 & -c_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & r_4 & r_3 & -c_6 & -c_5 & 0 \\ 0 & 0 & 0 & 0 & r_6 & r_5 & 0 \end{bmatrix}$$

The third structure, the parallel form, corresponds to a partial-fraction expansion of (39) and breaks into parallel direct form II first and second-order sections. The expanded transfer function (also 12 coefficients) and the one precedence level ψ_1 are shown in (43) and (44):

$$H(z) = \frac{e_1z^{-1}+e_2z^{-2}}{1+c_1z^{-1}+c_2z^{-2}} + \frac{e_3z^{-1}}{1+d_3z^{-1}} + \frac{e_4z^{-1}}{1+d_4z^{-1}} + \frac{e_5z^{-1}}{1+d_5z^{-1}} + \frac{e_6z^{-1}}{1+d_6z^{-1}} \tag{43}$$

$$\psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ -c_2 & -c_1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -d_3 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -d_4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -d_5 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -d_6 & 1 \\ e_2 & e_1 & e_3 & e_4 & e_5 & e_6 & 0 \end{bmatrix} \qquad (44)$$

The fourth structure (herein referred to as the 'simple' form) is taken directly from the original LQG compensator equations (3). The parameters of Φ, Γ, K, L and G are taken to be the coefficients of this structure. The form of the transfer function containing these coefficients is shown in (45), and the modified state-space representation of the structure (two precedence levels) is shown in (46):

$$H(z) = -G(z-\Phi+KL+\Gamma G)^{-1}K \qquad (45)$$

$$\psi_2\psi_1 = \left[\begin{array}{c|c|c} \Phi & -K & -\Gamma \\ \hline 0\ 0\ 0\ 0\ 0\ 0 & 0 & -1 \end{array}\right] \left[\begin{array}{c|c} I_6 & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix} \\ \hline L & -1 \\ \hline G & 0 \end{array}\right] \qquad (46)$$

Table 1 presents data concerning the statistical wordlength estimate for the four structures described above. For this system, a five percent degradation was specified as the maximal deterioration allowable in the measure of performance J.

		SWL	TWL	
Structure (eqn)	ℓ	bits (time)	bits (time)	coefficients
direct-II(40)	3	26.68(.75)	27(1.18)	12
cascade(42)	1	16.78(.81)	15(1.34)	12
parallel(44)	1	12.65(.71)	12(.77)	12
simple(46)	5	22.50(4.2)	21(.76)	47

TABLE 1: SWL Results for the F8 Example

The effect of structure on coefficient wordlength is evident from Table 1. The direct form II structure requires by far the most bits, while the cascade and parallel forms require the least. Both of these results are also typical of digital filters [7]. The simple form structure derived directly from the LQG compensator equations requires an intermediate number of bits, but its most undesireable property is its 47 coefficients, implying many hardware multipliers or a long calculation time (low system sample rate).

As an estimate, the statistical wordlength for this LQG example is between -.32 and +1.78 bits of the true wordlength (TWL). With this error range, the statistical wordlength estimate is quite useful both for the comparison of different structures and for the determination of an acceptable design wordlength. For comparison, the digital filter examined by Crochiere [14] has statistical wordlength estimates (based on transfer function magnitude) that were between 1 and 3 bits conservative.

Before interpreting the computation times listed in parentheses in Table 1, the method for determining the true wordlength must be described. The performance index J is roughly a monotonic function in the number of coefficient bits. This fact allows a binary search type of algorithm to be used,re-evaluating the index J until the degradation specification is met with a minimum number of bits. Unfortunately, there are several problems that can arise. First, when rounded coefficients produce an unstable closed-loop system, J can be below its J_∞ value and even be negative. Even when this situation does not occur, J is not necessarily montonic; certain values of J can be slightly smaller then the J value using 1 more coefficient bit. These two problems can slow down (or 'tie up') the search algorithm used in determining a true required wordlength, and explains the 1.18 and 1.34 second computation times for the direct and cascade structures' true wordlength.

Comparing the computation times for the statistical and true wordlengths, we see that the statistical wordlength is somewhat faster to compute in all cases except the simple form. This exception is due to the strong computational dependence of the statistical estimate on the number of coefficients. However, as mentioned above, this simple form would probably never be considered due to the hardware implications of computing 47 multiplies per sample period.

8. Conclusion

This paper constitutes a first step in examining the issues involved in the digital implementation of control compensators. To deal with these issues, we have sought to ally the fields of digital signal processing and control and estimation, a fairly novel approach.

More specifically, this paper treats the statistical coefficient wordlength issue for the LQG compensator using fixed-point arithmetic. After reviewing the LQG design procedure and defining the notion of an implementation structure, the statistical wordlength concept for digital filters was described. In adapting this concept to a control and estimation problem, we stressed the importance of selecting a good performance measure. The index J was chosen for the LQG problem, although the method readily extends to other measures (for example, the covariance matrix trace for Kalman filter problems). Finally an efficient computational method was discussed and an illustrative example presented.

Our results demonstrate the feasibility of using the statistical approach in determining a sufficient LQG compensator coefficient wordlength. One application of this technique would be in the comparison of different structures for implementing a design. In addition, the statistical wordlength is also an accurate criterion for selecting the wordlength once a specific structure is chosen.

Perhaps of more importance, the continuous 'closed-form' nature of the statistical wordlength estimate makes it possible to *synthesize* minimum coefficient wordlength structures in a straightforward manner. Chan [15] has described such a technique, using the modified state-space notation, for digital filters. This idea can be easily extended to the LQG statistical wordlength estimate presented in this paper. [20]

Finally, as a general technique, the statistical measure of coefficient wordlength can be applied to a variety of control and estimation problems, using whatever measure of performance seems appropriate (gain margin, phase margin, transfer function magnitude *and* phase, a covariance matrix trace, etc.). Within the computational formulation of sections 4 and 5, *suboptimal* LQG compensators or Kalman filters can be considered simply by including first derivative terms in the analysis (with a moderate increase in computation). These and related questions are considered in more detail in [20].

REFERENCES

[1] H. Kwakernaak and R. Sivan, Linear Optimal Control Systems, J. Wiley & Sons, New York, 1972.

[2] J.C. Willems and S.K. Mitter, "Controllability, Observability, Pole Allocation, and State Reconstruction," IEEE Trans. on Aut. Control, V. AC-17, Dec. 1971, pp. 582-595.

[3] B.C. Kuo, Analysis and Synthesis of Sampled-Data Control Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

[4] J.B. Knowles and R. Edwards, "Effect of a Finite-Word-Length Computer in a Sampled-Data Feedback Systems," Proc. IEE, V.112, No.6, June 1965, pp. 1197-1207.

[5]. A.B. Sripad, "Models for Finite Precision Arithmetic, with Application to the Digital Implementation of Kalman Filters," Sc.D. Dissertation, Washington Univ. Sever Institute, Jan. 1978.

[6] R.E. Rink and H.Y. Chong, "Performance of State Regulator Systems with Floating-Point Computation," to be published.

[7] A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[8] T.A.C.M. Claasen, W.F.G. Mecklenbräuker, and J.B.H. Peek, "Effects of Quantization and Overflow in Recursive Digital Filters," IEEE Trans. Acoustics, Speech, and Sig. Processing, V.ASSP-24, No.6, Dec. 1976, pp. 517-529.

[9] J.F. Kaiser, "On the Limit Cycle Problem, "Proc. IEEE Inter. Conf. Acoustics, Speech, and Sig. Processing, 1976, pp. 642-644.

[10] A.V. Oppenheim and C.J. Weinstein, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," Proc. IEEE, V. 60, August 1972, pp. 957-976.

[11] J.B. Knowles and E.M. Olcayto, "Coefficient Accuracy and Digital Filter Response," IEEE Trans. Circuits and Systems, V. CAS-15, March 1968, pp. 31-41.

[12] E. Avenhaus, "On the Design of Digital Filters with Coefficients of Limited Word Length," IEEE Trans. Audio & Electroacoustics, V. AU-20, Aug. 1972, pp. 206-212.

[13] R.E. Crochiere, "A New Statistical Approach to the Coefficient Word Length Problem for Digital Filters," IEEE Trans. Circuits and Systems, V. CAS-22, No.3, March 1975, pp. 190-196.

[14] R.E. Crochiere, "Digital Network Theory and Its Application to the Analysis and Design of Digital Filters," Ph.D. Dissertation, MIT, Dept. of EE, April, 1974.

[15] D.S.K. Chan, "Theory and Implementation of Multidimensional Discrete Systems for Signal Processing," Ph.D. Dissertation, MIT, Dept. of EE & CS, May 1978.

[16] R.E. Crochiere and A.V. Oppenheim, "Analysis of Linear Digital Networks," Proc. IEEE, V.63, No. 4, April 1975, pp. 581-595.

[17] D.S.K. Chan and L.R. Rabiner, "Analysis of Quantization Errors in the Direct Form for Finite Impulse Response Digital Filters," IEEE Trans. Audio Electroacoustics, V. AU-21, August 1973, pp.354-366.

[18] G.K. Roberts, "Consideration of Computer Limitations in Implementing On-Line Controls," MIT ESL-R-665, Cambridge, Ma., June 1976.

[19] A.Y. Barraud, "A Numerical Algorithm to Solve $A^TXA-X=Q$," IEEE Trans. Aut. Control, V. AC-22, No.5, Oct. 1977, pp. 883-885.

[20] P. Moroney, "Issues in the Digital Implementation of Control Compensators," Ph.D. Dissertation, MIT, Dept. of EE & CS, in progress.

[21] A.E. Bryson,Jr., Guest Ed. Mini-Issue on the F-8 DFBW, IEEE Trans. Aut. Control, V. AC-22, No. 5, Oct. 1977, pp. 752-806.