Proc. IFAC Symp. on Theory &
Application of Digital
Control, January 5-7, 1982.

LINEAR COMPENSATOR DESIGN WITH BLOCK PROCESSING CONSTRAINTS

Brian D. O. Anderson[1]

Department of Electrical and Computer Engineering, University of Newcastle, N.S.W. 2308, Australia.

James C. Kolanek

Raytheon Co., Electromagnetic Systems Division, 6380 Hollister Avenue, Goleta, CA 93117

Sanjit K. Mitra

Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106

Abstract. The implementation of a discrete feedback compensator using block processing is discussed. In this scheme, a block of L measurements are accumulated before the processing is initiated. A block of L compensation samples are then computed and released serially into the system being compensated. Compensator implementations are developed both for the ARMA and the state-variable representations.

Keywords. Control system synthesis; controllers; digital control; feedback; observers; signal processing.

INTRODUCTION

A conventional feedback compensator might be implemented as shown in Fig. 1. For the discrete case, the serial processor accepts a sample of the plant output $y(n)$ to compute a single compensation value $v(n+1)$. This output is summed with external input $r(n+1)$ to provide the plant input $u(n+1)$ at the next sample instant. At sample time $n+1$ the plant output $y(n+1)$ is processed by the compensator to start another cycle.

An alternate approach to the processing of signals by the compensator is via block processing, a concept which was first proposed by Gold and Jordan (1968) for the implementation of digital filters. In this scheme, illustrated in Fig. 2, the plant output samples are accumulated without processing in a serial to parallel converter. In general, a block of L samples $\{y(kL), y(kL+1),\ldots, y(kL+L-1)\}$ will be accumulated to form the vector $Y(k)$. That is

$$Y(k) = \begin{bmatrix} y(kL) \\ y(kL+1) \\ \vdots \\ y(kL+L-1) \end{bmatrix} \qquad (1.1)$$

For a single input, single output plant, this vector will be of dimension L. The resulting vectors are then processed all at once to form a block of compensation samples $V(k+1)$, also of dimension L, which is defined as follows:

$$V(k+1) = \begin{bmatrix} v(kL+L) \\ v(kL+L+1) \\ \vdots \\ v(kL+2L-1) \end{bmatrix} \qquad (1.2)$$

These samples are then released in serial fashion to be summed with the external input to provide the plant input.

In spite of the built-in delay, up to L sample intervals, which can result in this structure, block processing offers several advantages which has caused the increasing recent interest in its application. These include the possibility of high speed operation using array processors, a reduction in computational complexity in certain cases, and a reduction in the effects of finite word lengths.

The main purpose of this paper is to apply block processing methods to the realization of feedback compensators with the structure shown in Fig. 2. The compensator can be implemented by blocking either the discrete difference (ARMA) equations which result

[1] Part of this work was carried out at the address of the third author.

from an input/output development or the discrete state variable equations which result from a state feedback development. The difference equation approach is discussed in Section 2 while the state variable approach is discussed in Section 3.

DIFFERENCE EQUATION REALIZATION

Suppose that the compensator has been specified in the difference equation notation:

$$\sum_{i=0}^{M} d_i v(n-i) = \sum_{i=0}^{M} c_i y(n-i) \qquad (2.1)$$

Using the blocking procedure developed by Ananthakrishna (1981), with $q = \lfloor M/L \rfloor$, the block form of the compensator equation becomes

$$\sum_{i=0}^{q} D_i V(k-i) = \sum_{i=0}^{q} C_i Y(k-i) \qquad (2.2)$$

where the block parameter matrices C_i and D_i have the form

$$C_i = \begin{bmatrix} c_{iL} & c_{iL-1} \cdots & c_{iL-L+1} \\ c_{iL+1} & \cdot & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ c_{iL+L-1} \cdot & \cdots & c_{iL} \end{bmatrix} \qquad (2.3)$$

and

$$D_i = \begin{bmatrix} d_{iL} & d_{iL-1} \cdots & d_{iL-L+1} \\ d_{iL+1} & \cdot & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ d_{iL+L-1} \cdot & \cdots & d_{iL} \end{bmatrix} \qquad (2.4)$$

with c_j and $d_j = 0$ for $j < 0$ or $j > M$.

The problem with implementing (2.2) is that in order to compute $V(k)$ at the sample time $kL-1$, knowledge of the plant output $Y(k) = \{y(kL+L), y(kL+L+1) \ldots y(kL+2L-1)\}$ is required but which is not yet available. Since the block compensator must operate using only block measurements taken during block $k-1$ or earlier, an approach will now be developed to do this by incorporating the characteristics of the plant into the compensator equations.

Let the difference equations of the plant be given by

$$\sum_{i=0}^{N} b_i y(n-i) = \sum_{i=1}^{N} a_i u(n-i) \qquad (2.5)$$

These equations can also be blocked which produces

$$\sum_{i=0}^{p} B_i Y(k-i) = \sum_{i=1}^{p} A_i U(k-i) \qquad (2.6)$$

where $p = \lfloor N/L \rfloor$ and $Y(k)$, $U(k)$, A_i and B_i have the same structure as the corresponding elements in the compensator equations.

By combining equations (2.2) and (2.6), and noting that $U(k) = R(k) + V(k)$, an expression for $V(k)$ can be obtained which is independent of $Y(k)$.

$$[D_0 - C_0 B_0^{-1} A_0] V(k) = \sum_{i=1}^{p} [C_i - C_0 B_0^{-1}] Y(k-i) + \sum_{i=1}^{p} [C_0 B_0^{-1} A_i - D_i] V(k-i) + \sum_{i=1}^{p} C_0 B_0^{-1} A_i R(k-i) \qquad (2.7)$$

which has the structure

$$E_0 V(k) = \sum_{i=1}^{p} E_i V(k-i) + \sum_{i=1}^{p} F_i Y(k-i) + \sum_{i=0}^{p} G_i R(k-i) \qquad (2.8)$$

This is the design equation of the blocked compensator. The following points should be noted.

1. The lower triangular Toeplitz structure of A_0, B_0, C_0, D_0 ensures that B_0^{-1} and E_0^{-1} exist.

2. The compensator requires knowledge at time $(kL-1)$, the time by which $V(k)$ is to become available, of the external inputs $R(k)$, i.e. $r(kL)$ through $r(kL+L-1)$. This means external inputs must be known for the whole of a block at the start of a block. For a regulator problem, or a problem involving tracking of a reference trajectory, this may present no difficulty. On other occasions however, this could interfere with use of a blocked compensator.

3. The degree of the characteristic polynomial of the compensator is easily checked. This characteristic polynomial is the determinant of $E_0 - E_1 z^{-1} \ldots - E_p z^{-p}$, and has degree N in general.

4. One of the advantages of using block processing is the possibility of computational savings; these are achieved by using fast convolution algorithms to compute products of certain matrices and vectors due to their Toeplitz structure (Burrus, 1972; Mitra, 1978). The use of a conventional compensator over L time instants requires L separate processing actions through an M dimensional system; use of the blocked compensator requires one processing action through a N dimensional system.

5. The adjustments to the above ideas in the case of multiple-input, multiple-output systems are trivial.

6. The case of series or cascade compensation can be handled in much the same way. Suppose that the plant is still given by (2.5), while the scalar compensator equation is

$$\sum_{i=0}^{M} d_i u(n-i) = \sum_{i=0}^{M} c_i[r(n-i) - y(n-i)]$$
$$\overset{\Delta}{=} \sum_{i=0}^{m} c_i z(n-i) \qquad (2.9)$$

Suppose the block compensator has available to it at time kL-1 the vector Z(k-1) and by the time kL is to produce U(k). Blocking of the plant and compensator equations produces

$$\sum_{i=0}^{p} B_i Y(k-i) = \sum_{i=0}^{p} A_i U(k-i) \qquad (2.10)$$

and

$$\sum_{i=0}^{q} D_i U(k-i) = \sum_{i=0}^{q} C_i Z(k-i) \qquad (2.11)$$

Once again, U(k) can be expressed in terms of U(i), Z(i) for $i \le k-i$ and R(i) for $i \le k$. The fact that R(k) is blocked implies that at time kL-1, not only is r(kL) known, but also r(kL+1), ..., r(kL+L-1), i.e. there is the same look ahead requirement as with feedback compensators.

BLOCKING A STATE FEEDBACK LAW AND BLOCKED OBSERVERS

An alternate approach to compensator implementation using state feedback with a state observer will now be studied. Suppose that the plant is described by

$$x(n+1) = ax(n) + bu(n) \qquad (3.1a)$$

$$y(n) = cx(n) \qquad (3.1b)$$

(where lower case a, b and c are used for convenience to differentiate between the conventional state space quantities and the block state space quantities) and the feedback law where

$$v(n) = kx(n) \qquad (3.2)$$

is to be implemented. Suppose further that it is desired to generate v(kL), ..., v(kL+L-1) all at time kL. Observe that

$$v(kL) = kx(kL)$$

$$v(kL+1) = kx(kL+1)$$
$$= kax(kL) + kbu(kL)$$

$$v(kL+2) = kx(kL+2)$$
$$= ka^2x(kL) + kabu(kL) + kbu(kL+1)$$
$$\vdots$$

$$v(kL+L-1) = kx(kL+L-1)$$
$$= ka^{L-1}x(kL) + ka^{L-2}bu(kL)$$
$$+ ka^{L-3}bu(kL+1) + \dots + kbu(kL-L-2)$$

Using upper case letters to denote the block vectors and matrices as in the last section, certain matrices H_1 and H_2 can be defined such that

$$V(k) = H_1X(k) + H_2U(k) \qquad (3.4)$$

where
$$X(k) = x(kL)$$

and H_2 is lower triangular Toeplitz with zero on the diagonal. Using the fact that U(k) = R(k) - V(k)

$$V(k) = [I-H_2]^{-1}H_1X(k) + [I-H_2]^{-1}H_2R(k) \qquad (3.5)$$

Since X(k) is not available until time kL, and U(k) is supposed to be available at time kL, equation (3.5) presents some computational difficulties. As with the difference equation representation, knowledge of the plant equations can be incorporated into the compensator to resolve the problem. The block form of the plant state equations is given by (Meyer, 1975; Mitra, 1978; Barnes, 1980)

$$X(kL+1) = AX(k) + BU(k) \qquad (3.6)$$

where

$$A = a^L \qquad (3.7)$$

$$B = [a^{L-1}b,\ a^{L-2}b,\ \dots,b] \qquad (3.8)$$

Combining (3.5) and (3.6)

$$V(k) = [I-H_2]^{-1}H_1AX(k-1) + [I-H_2]^{-1}H_1BU(k-1)$$
$$+ [I-H_2]^{-1}H_2R(k)] = [I-H_2]^{-1}H_1AX(k-1)$$
$$+ [I-H_2]^{-1}H_1BR(k-1) + [I-H_2]^{-1}H_2R(k-1)$$
$$+ [I-H_2]^{-1}H_1BV(k-1) \qquad (3.9)$$

Now suppose that measurements of the state of (3.1) are not directly available from the plant output. Then we require a procedure for estimating X(k), k=0, 1, ..., rather than x(n) for all n; furthermore, the scheme should preferably itself operate on a block basis.

There are two methods of obtaining such a scheme

(a) We can write down block equations for the plant; then, the sequence of state vectors for these equations is

$X(0)$, $X(L)$, $X(2L)$ i.e. the state vector sequence for the blocked plant equations is a subsequence of the state vector for the actual plant. Then we can build an observer for this blocked system.

(b) We can build a conventional observer for the plant, and then block this observer.

Approach (a)

The block equation of the plant output is (Meyer, 1975; Mitra, 1978; Barnes, 1980)

$$Y(k) = CX(k) + DU(k) \tag{3.10}$$

where the block matrices C and D are given by

$$C = \begin{bmatrix} c \\ ca \\ \cdot \\ \cdot \\ ca^{L-1} \end{bmatrix} \tag{3.11a}$$

$$D = \begin{bmatrix} 0 & \dots & & 0 \\ cb & & & \\ cab & & & \\ \vdots & & & \\ ca^{L-1}b & \dots\ cab & cb & 0 \end{bmatrix} \tag{3.11b}$$

An observer for this blocked plant produces an estimate $X_e(k)$ for $X(k)$ via an equation of the form

$$X_e(k+1) = AX_e(k) + BU(k) + K_e[Y(k) - DU(k) - CX_e(k)] \tag{3.12}$$

where K_e is chosen so that $A-K_eC$ has all its eigenvalues well inside the unit circle.

Approach (b)

A conventional observer for (3.1) is provided by

$$x_0(n+1) = ax_0(n) + bu(n) + k_0[y(n) - cx_0(n)] \tag{3.13}$$

with $a-k_0c$ chosen to have appropriate eigenvalues. Now regard (3.13) simply as a linear system with two inputs, viz. $y(n)$ and $u(n)$, and write down the corresponding block equation. This is

$$X_0(kx+1) = (a-k_0c)^L X_0(k) + [(a-k_0c)^{L-1}k_0 \ \dots\ (a-k_0c)k_0 \ \ k_0]Y(k) + [(a-k_0c)^{L-1}b \ \dots\ (a-k_0c)b \ \ b]U(k) \tag{3.14}$$

Comparing (3.10) and (3.12), we see that the two approaches would yield the same result if we had

$$a^L - K_eC = (a - k_0c)^L \tag{3.15a}$$

$$K_e = [(a-k_0c)^{L-1}k_0 \dots (a-k_0c)k_0 \ \ k_0] \tag{3.15b}$$

$$B-K_eD = [(a-k_0c)^{L-1}b_0 \dots (a-b_0c)b_o \ \ b_0] \tag{3.15c}$$

If one starts with k_0 in approach (b), and defines K_e by (3.15b) then some algebra will verify (3.15a) and (3.15c). This means that every observer definable by approach (b) could equally well have been developed by approach (a). On the other hand, if K_e is any matrix for which a^L-K_eC has the correctly located eigenvalues, it is not true that in general a k_0 can be found satisfying (3.15b), (there are fewer entries in k_0 than in K_e). The observer of (3.12) will then not compare to an observer of the form of (3.14), i.e. observers described by approach (b). Actually, if in (3.13) one permits k_0 to be periodic with period L, the two approaches lead to the same class of blocked observers as can be easily shown.

In summary, in this section, we have shown how a state feedback law can be blocked [see (3.5)] and how a blocked observer can be used to calculate estimates of those values of the state needed to implement the blocked state feedback law [see (3.12) and (3.14)]. We also note

1. The same sort of approach will allow blocking of reduced order (Luenberger) observers.

2. In L sampling intervals, the blocked observer is updated once, the unblocked L times. This is the key to achieving computational savings.

3. Another approach to the ideas of this section would be to consider a conventional observer cascaded with a feedback control law as a 2-input [$u(n)$ and $y(n)$] and 1-output [$kx(n)$] system. Such a system can be described with a difference equation (ARMA equation) which can be blocked. The ideas are similar to those of the preceding section. Analysis of, for example, sensitivity to parameter variations, noise and the like would however proceed very differently for the two arrangements.

4. If the eignevalues of $(a-k_0c)$ are well away from (but inside) the boundary of the unit circle, those of $(a-k_0c)^L$, being the L-th power, may be very close to the origin. As a result, savings in word length in implementing (3.10) or (3.12) if it is equivalent, may well be possible.

Accommodating the External Inputs

A characteristic of the schemes of Section 2 and Section 3 was that some foreknowledge of external inputs was required. Here, we discuss what one might do without this knowledge. Recall that by time kL, the compensator has generated the vector $V(k)$, which is then released in a serial fashion, with $v(i)$ becoming available at time i. The signal $v(i)$ is added to $r(i)$ and $u(i) = v(i) + r(i)$ is provided as the plant input. As (2.8) shows, $V(k)$ depends on $R(k)$.

For the purposes of running the compensator, one can assume a set of values for $r(n)$, call them $\tilde{r}(n)$, based on some sort of extrapolation of past $r(i)$, or some knowledge of the statistics of external inputs. For example, if $r(i)$ consists mainly of constant signals, *the level of which change rarely, one might* assume that the $r(i)$ values during a block were the same as the value at the start of the block. Now using $\tilde{R}(k)$, one can compute a *value for* $V(k)$, *call it* $\tilde{V}(k)$ *and then one can* provide an input to the plant as follows:

$$\tilde{U}(k) = \tilde{V}(k) + R(k) \qquad (4.1)$$

during the k-th block. Of course, $\tilde{R}(k)$ is stored along with $\tilde{U}(k)$ in the compensator. In every block, the wrong $u(i)$ is applied to the plant. However, as we now show, in block k+1 we can correct for all past errors in the inputs applied, so that the only source of errors in the plant inputs and outputs during any one block are those associated with incorrect estimates of the external inputs during that block.

The correction is achieved in the following way. Suppose the compensator is given by (2.8). Let us also suppose that at time k we know $R(i)$ for $i \leq k-1$, but we are forced to use $\tilde{R}(k)$. Then we would compute $\tilde{V}(k+1)$ according to

$$E_0\tilde{V}(k) = \sum_{i=1}^{p} E_iV(k-i) + \sum_{i=1}^{p} F_iY(k-i) + \sum_{i=1}^{p} G_iR(k-i) + G_0\tilde{R}(k) \qquad (4.2)$$

We would apply $\tilde{U}(k+1)$ given by (4.1) to the plant, and if the plant had received correct inputs through block k-1, the output would satisfy

$$B_0\tilde{Y}(k) + \sum_{i=1}^{p} B_iY(k-1) = A\,\tilde{U}(k) + \sum_{i=1}^{p} A_iY(k-i) \qquad (4.3)$$

Now advance to the next block. By the start *of block (k+1), we know what R(k) was.* If we had been able to use the correct $R(k)$, then we would have had $V(k)$ and $Y(k)$ differing from $\tilde{V}(k)$ and $\tilde{Y}(k)$; in fact (2.8) and (4.2) imply

$$V(k) - \tilde{V}(k+1) = E_0^{-1}G_0[R(k) - \tilde{R}(k)] \qquad (4.4)$$

while (2.8) and (4.3) with $U(k)$ and $Y(k)$ replacing $\tilde{U}(k)$ and $\tilde{Y}(k)$ imply

$$Y(k) - \tilde{Y}(k) = B_0^{-1}A_0[V(k) - \tilde{V}(k+1)] \qquad (4.5)$$

$$= B_0^{-1}A_0E_0^{-1}G_0[R(k) - \tilde{R}(k)]$$

Accordingly, the compensator at time k+1 replaces the internally stored vector $\tilde{R}(k)$ by $R(k)$, computes the quantity $V(k)$ from $\tilde{V}(k)$ using the correction (4.4), computes $Y(k)$ *from the measurements of the plant output* and the correction (4.5), and then uses (4.2) updated by one time instant.

While the above analysis is for the case when *feedback compensation is used, it should be* realized that similar procedures will cope with other forms of controller operating in a block processing mode.

CONCLUSION

We have shown how the known procedures for blocking systems described by difference and state equations can be modified when such systems are part of a feedback loop. The main consequences are: 1) computational savings are still provided by blocking; 2) the resulting control arrangement alternates between L-1 time instants of open loop operation and 1 time instant of closed loop operation.

The ideas, though presented for single-input, single-output plants, extend trivially to multivariable plants. It is also easy to make adjustments for the delays caused by computations at the start of each block, and for different forms of observers.

REFERENCES

Ananthakrishna, P. (1981). A general theory of block digital filters and low noise realizations. Ph.D. Dissertation, University of California, Santa Barbara, CA, September 1981.

Barnes, C. W., and S. Shinnaka (1980). Block-shift invariance and block implementation of discrete-time filters, IEEE Trans. Circuits Systems, Vol. CAS-27, pp. 667-672.

Meyer, R. A., and C. S. Burrus (1975). A unified analysis of multivariate and periodically time-varying digital filters, IEEE Trans. Circuits Systems, Vol. CAS-22, pp. 162-168.

Mitra, S. K., and R. S. Ganasekaran (1978). Block implementation of recursive digital filters-New structures and properties, IEEE Trans. Circuits Systems, Vol. CAS-25, pp. 200-207.

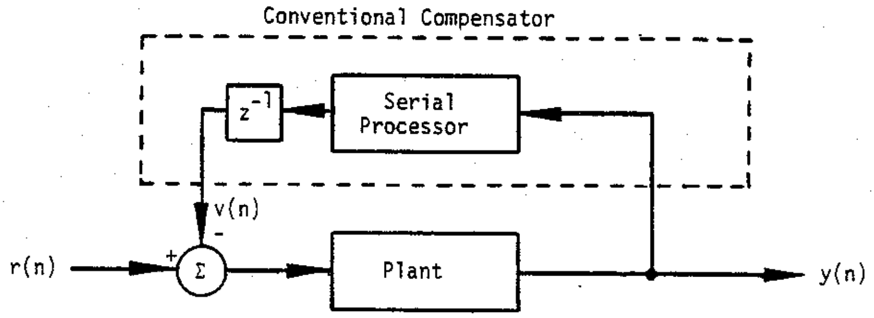

Fig. 1. Conventional feedback compensator

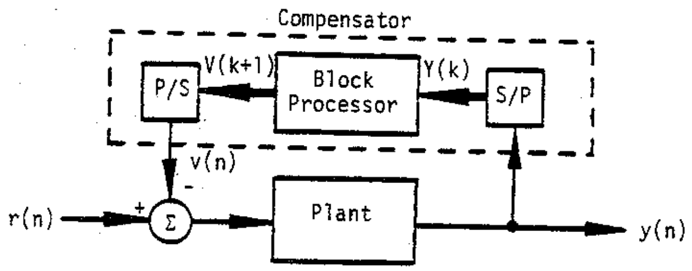

Fig. 2. Blocked feedback compensator