

A recursive algorithm for the reduction of block diagrams

TROND ANDRESEN†

Keywords: *Block diagram reduction, recursive algorithm, transfer matrix, time delay.*

A recursive algorithm to compute input-output relationships in a network of rational transfer functions (a block diagram) is derived. Initially all transfer functions are disconnected. For each step in the algorithm, a new transfer function in the network is connected to the system. One application of the algorithm is to find the transfer matrix of a linear multivariable system given on state space form. The algorithm is advantageous when one needs repeated computations of a transfer matrix in a multivariable system for successive changes in one or a few parameters in the system. The transfer matrix for a multivariable linear system with one time delay, and the exact frequency response in the case of several time delays, are derived.

1. Introduction

For analysis of a multivariable linear system one frequently needs to compute the transfer matrix $H(s)$ ($H(z)$ for a discrete linear system) where

$$y(s) = H(s)x(s) \quad (1)$$

One procedure for finding $H(s)$ is first to formulate a state space representation of the linear system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Dx \end{aligned} \quad (2)$$

and then calculate $H(s)$ as

$$H(s) = D(sI - A)^{-1}B \quad (3)$$

There are several methods of this class. All of them are based on the calculation of the eigenvalues of the system (Ogata, 1967).

A different approach is based on a description of the system as a network of monovariate transfer functions (a 'block diagram'). Such a network may then be reduced to a transfer matrix by Mason's rule (Horowitz, 1963). This procedure, however, is very cumbersome if the different parts of the system are strongly interconnected. This paper will present a way to computerize Mason's rule in the form of a recursive algorithm. A by-product is the derivation of the transfer matrix for a multivariable system with one time delay.

Received 15 December 1990.

† Division of Engineering Cybernetics, Norwegian Institute of Technology, N-7034 Trondheim, Norway.

2. Derivation of the algorithm

To derive the algorithm the following notation is defined.

- H^{k-1} Transfer matrix for a quadratic system with $k-1$ inputs and $k-1$ outputs (for simplicity's sake, the dependence on s is omitted)
- P^{k-1} Matrix of nominator polynomials in s
- q_{k-1} Common denominator polynomial in s . We have $H^{k-1} = P^{k-1}/q_{k-1}$
- S_{k-1} Name tag for the system given by H^{k-1}
- u^{k-1} Input vector to S_{k-1}
- y_{k-1} Output vector from S_{k-1}
- B_k Name tag for a monovariate block which is to be connected to S^{k-1}
- h_k Transfer function for B_k
- t_k, n_k Nominator and denominator polynomial for B_k , $h_k = t_k/n_k$
- y_k Output from B_k
- u_k Input to B_k
- c_u^{k-1} Vector of coefficients connecting the inputs of S_{k-1} to the output of B_k
- $c_y^{T,k-1}$ Row vector of coefficients connecting the outputs of S_{k-1} to the input of B_k

The connection between the system S_{k-1} and the monovariate block B_k may be expressed as in Fig. 1: This again is equivalent to the system S_k as shown in Fig. 2.

We can express the transfer matrix H^k as a function

$$H^k = \alpha_k(H^{k-1}, c_y^{T,k-1}, c_u^{k-1}, h_k) \quad (4)$$

which is recursive.

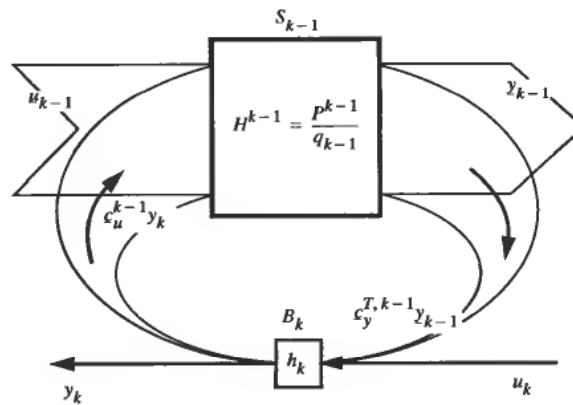


Figure 1.

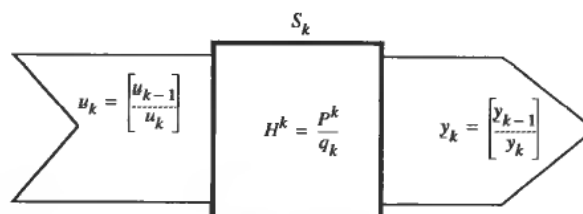


Figure 2.

We want to find the elements in the matrix $H^k = P^k/q_k$. The nominator matrix P^k may be partitioned as

$$P^k = \begin{bmatrix} \tilde{P}^k & p_u^k \\ p_y^{T,k} & p_{kk}^k \end{bmatrix} \quad (5)$$

where \tilde{P}^k is a $(k-1) \times (k-1)$ matrix of polynomials in s , $p_y^{T,k}$ and p_u^k are similar row and column vectors of dimension $(k-1)$, and p_{kk}^k is a scalar polynomial in s . Our first step is to find expressions for p_{kk}^k and q_k . From Fig. 1 we have

$$y_{k-1} = H^{k-1} u_{k-1} + H^{k-1} c_u^{k-1} y_k \quad (6)$$

Furthermore

$$y_k = h_k u_k + h_k c_y^{T,k-1} y_{k-1} \quad (7)$$

Setting $u_{k-1} = 0$ in (6) and substituting (6) for y_{k-1} in (7) we get

$$y_k = h_k u_k + h_k c_y^{T,k-1} H^{k-1} c_u^{k-1} y_k \quad (8)$$

which leads to

$$\frac{y_k}{u_k} = \frac{h_k}{1 - h_k c_y^{T,k-1} H^{k-1} c_u^{k-1}} \quad (9)$$

We want polynomials in nominator and denominator and multiply both by $n_k q_{k-1}$, getting

$$\frac{y_k}{u_k} = \frac{t_k q_{k-1}}{n_k q_{k-1} - t_k c_y^{T,k-1} P^{k-1} c_u^{k-1}} = \frac{p_{kk}^k}{q_k} \quad (10)$$

which implies

$$p_{kk}^k = t_k q_{k-1} \quad (11)$$

and

$$q_k = n_k q_{k-1} - t_k c_y^{T,k-1} P^{k-1} c_u^{k-1} \quad (12)$$

We now have recursive expressions (11) and (12) for two of the polynomials in H^k . (Note: The expression $h_k c_y^{T,k-1} H^{k-1} c_u^{k-1}$ in (9) is the loop transfer function or the sum of all parallel transfer functions in the new loops that results when the monovariate block B_k is connected to the system S_{k-1} . This is illustrated in Fig. 3 where $S_{k-1} = S_3$ is a 3×3 system and we have 9 loops.)

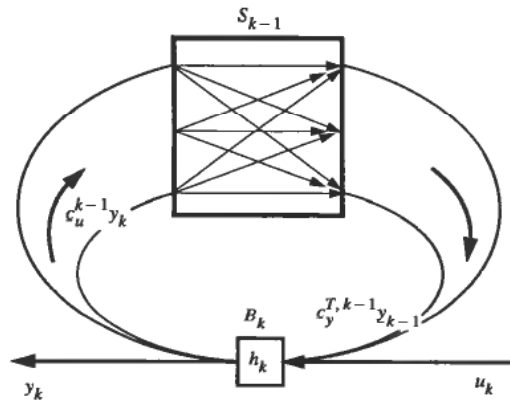


Figure 3.

The lower row vector of polynomials in P^k , $p_y^{T,k}$, can be found by setting $u_k=0$ in (7) and substituting (6) for y_{k-1} in (7).

$$y_k = h_k c_y^{T,k-1} H^{k-1} u_{k-1} + h_k c_y^{T,k-1} H^{k-1} c_u^{k-1} y_{k-1} \quad (13)$$

Solving for y_k and multiplying the nominator and denominator by $n_k q_{k-1}$, we get

$$y_k = \frac{t_k c_y^{T,k-1} P^{k-1}}{n_k q_{k-1} - t_k c_y^{T,k-1} P^{k-1} c_u^{k-1}} u_{k-1} = \frac{p_y^{T,k}}{q_k} u_{k-1} \quad (14)$$

which implies

$$p_y^{T,k} = t_k c_y^{T,k-1} P^{k-1} \quad (15)$$

By a similar procedure, we get the rightmost column vector in P^k

$$p_u^k = t_k P^{k-1} c_u^{k-1} \quad (16)$$

What remains now is the matrix \tilde{P}^k . It may be found by setting $u_k=0$ in (7) and substituting (7) for y_k in (6).

$$y_{k-1} = H^{k-1} u_{k-1} + h_k H^{k-1} c_u^{k-1} c_y^{T,k-1} y_{k-1} \quad (17)$$

which, solved for y_{k-1} , yields

$$y_{k-1} = (I - h_k H^{k-1} c_u^{k-1} c_y^{T,k-1})^{-1} H^{k-1} u_{k-1} = \tilde{H}^k u_{k-1} \quad (18)$$

Using the matrix inversion lemma on (18), we get

$$\tilde{H}^k = H^{k-1} + h_k H^{k-1} c_u^{k-1} (1 - h_k c_y^{T,k-1} H^{k-1} c_u^{k-1})^{-1} c_y^{T,k-1} H^{k-1} \quad (19)$$

Which may be written as

$$\tilde{H}^k = \frac{1}{q_k} \frac{P^{k-1} q_k + t_k P^{k-1} c_u^{k-1} c_y^{T,k-1} P^{k-1}}{q_{k-1}} = \frac{\tilde{P}^k}{q_k} \quad (20)$$

which implies

$$\tilde{P}^k = \frac{P^{k-1} q_k + P^{k-1} c_u^{k-1} c_y^{T,k-1} P^{k-1}}{q_{k-1}} \quad (21)$$

Alternatively, using (12) for q_k in (21)

$$\tilde{P}^k = P^{k-1} n_k + t_k \frac{P^{k-1} c_u^{k-1} c_y^{T,k-1} P^{k-1} - (c_y^{T,k-1} P^{k-1} c_u^{k-1}) P^{k-1}}{q_{k-1}} \quad (22)$$

(Note: \tilde{P}^k is a matrix of polynomials. This implies that (21) and the nominator in (22) may be divided by q_{k-1} . This, then, will always be the case when \tilde{H}^{k-1} is a transfer matrix. Also note: The expression to the right in the nominator in (22) is not associative. The parentheses are necessary.)

We now have the necessary recursive formulae corresponding to (4), and may sum them up as (23)–(28):

$$q_k = n_k q_{k-1} - t_k c_y^{T,k-1} P^{k-1} c_u^{k-1} \quad (23)$$

$$p_{kk}^k = t_k q_{k-1} \quad (24)$$

$$p_y^{T,k} = t_k c_y^{T,k-1} P^{k-1} \quad (25)$$

$$p_u^k = t_k P^{k-1} c_u^{k-1} \quad (26)$$

$$\tilde{P}^k = \begin{cases} \frac{P^{k-1} q_k + P^{k-1} c_u^{k-1} c_y^{T,k-1} P^{k-1}}{q_{k-1}}, & \text{or} \\ P^{k-1} n_k + t_k \frac{P^{k-1} c_u^{k-1} c_y^{T,k-1} P^{k-1} - (c_y^{T,k-1} P^{k-1} c_u^{k-1}) P^{k-1}}{q_{k-1}} \end{cases} \quad (27)$$

$$P^k = \begin{bmatrix} \tilde{P}^k & p_u^k \\ p_y^{T,k} & p_{kk}^k \end{bmatrix} \quad (28)$$

This algorithm will from now on be labeled the RBDR algorithm. (RBDR = recursive block diagram reduction.)

3. Applications and examples

3.1. The transfer matrix for a state space model

A system given by

$$\dot{x} = Ax + u \quad (29)$$

$$y = x$$

has a transfer matrix

$$H(s) = (sI - A)^{-1} \quad (30)$$

(We omit the matrices B and D since the difficult task is to compute the matrix $(sI - A)^{-1}$.) The system (29) may be viewed as a network of monovariate blocks, each with a transfer function

$$h_k(s) = \frac{1}{s - a_{kk}}, \quad k = 1, \dots, n \quad (31)$$

The vectors $c_y^{T,k}$ and c_u^k may then be extracted from the matrix A as shown in Fig. 4. (Note: The order of calculations may be freely chosen. Figure 4 implies that we start with the system with the transfer 'matrix' $h_1(s)$ and connect this system to the block given by $h_2(s)$, then connect the resulting 2×2 -system to the block given by $h_3(s)$, and so on. One may, however, start with any block, and continue connecting the remaining blocks in any order. This of course requires an interchange of coefficients in the matrix A before extracting the vectors c_y^T and c_y . This freedom of choice will be exploited later.)

A disadvantage with this way of computing the matrix $(sI - A)^{-1}$ is that the eigenvalues of the system are not found. They may be found afterwards as the roots of the equation $q_n(s) = 0$, but this is not a good procedure numerically.

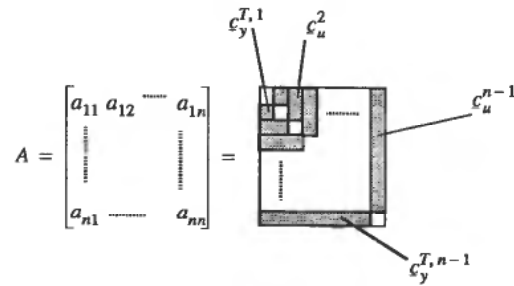


Figure 4.

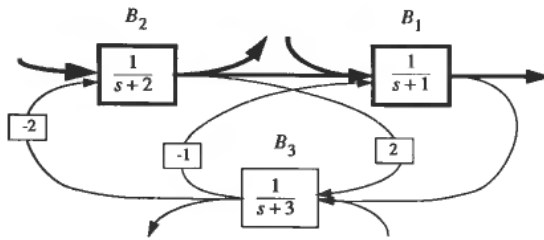


Figure 5.

We will now look at an example to illustrate the RBDR algorithm applied to a state space model. We have the system (29) with the matrix A given by

$$A = \begin{bmatrix} -1 & 1 & -1 \\ 0 & -2 & -2 \\ 1 & 2 & -3 \end{bmatrix} \quad (32)$$

or, drawn as a block diagram in Fig. 5. Inputs and outputs for the total system S_3 are shown in the figure. The sub-system S_2 is shown with thick lines and arrows. The transfer matrix of S_2 is trivial and can be found by inspection of Fig. 5.

$$H^2 = \frac{1}{s^2 + 3s + 2} \begin{bmatrix} s+2 & 1 \\ 0 & s+1 \end{bmatrix} = \frac{1}{q_2} P^2 \quad (33)$$

Finding the transfer matrix H^3 by inspection of the block diagram is not so trivial. Let us therefore do a step of the RBDR algorithm to find H^3 . First we extract vectors from the matrix A as shown in Fig. 4.

$$c_u^3 = \begin{bmatrix} -1 \\ -2 \end{bmatrix} \quad c_y^{T,3} = [1 \quad 2] \quad (34)$$

Using (23) we get

$$q_3 = (s+3)(s^2 + 3s + 2) - 1[1 \quad 2] \begin{bmatrix} s+2 & 1 \\ 0 & s+1 \end{bmatrix} \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

or

$$q_3 = s^3 + 6s^2 + 16s + 14 \quad (35)$$

Using (24), we get

$$p_{33}^3 = q_2 = s^2 + 3s + 2 \quad (36)$$

From (25), we get

$$p_y^{T,3} = 1[1 \ 2] \begin{bmatrix} s+2 & 1 \\ 0 & s+1 \end{bmatrix} = [s+2 \ 2s+3] \quad (37)$$

And (26) gives

$$p_u^3 = 1 \begin{bmatrix} s+2 & 1 \\ 0 & s+1 \end{bmatrix} \begin{bmatrix} -1 \\ -2 \end{bmatrix} = \begin{bmatrix} -s-4 \\ -2s-2 \end{bmatrix} \quad (38)$$

From the second part of (27), and from (37) and (38) we have

$$\tilde{P}^3 = \begin{bmatrix} s+2 & 1 \\ 0 & s+1 \end{bmatrix} (s+3) + \frac{\begin{bmatrix} -s-4 \\ -2s-2 \end{bmatrix} [s+2 \ 2s+3] - (-5s-8) \begin{bmatrix} s+2 & 1 \\ 0 & s+1 \end{bmatrix}}{s^2+3s+2}$$

or

$$\tilde{P}^3 = \begin{bmatrix} s+2 & 1 \\ 0 & s+1 \end{bmatrix} (s+3) + \frac{\begin{bmatrix} 4(s+1)(s+2) & -2(s+1)(s+2) \\ -2(s+1)(s+2) & (s+1)(s+2) \end{bmatrix}}{(s+1)(s+2)}$$

which reduces to

$$\tilde{P}^3 = \begin{bmatrix} s^2+5s+10 & s+1 \\ -2 & s^2+4s+4 \end{bmatrix} \quad (39)$$

From (35)–(39) and (28), we finally get the transfer matrix H^3 .

$$H^3 = \frac{P^3}{q^3} = \frac{1}{s^3+6s^2+16s+14} \begin{bmatrix} s^2+5s+10 & s+1 & -s-4 \\ -2 & s^2+4s+4 & -2s-2 \\ s+2 & 2s+3 & s^2+3s+2 \end{bmatrix} \quad (40)$$

3.2. The transfer matrix for a general network of monovisible blocks

The procedure can be explained through an example. The system is identical to the system in Fig. 5, except for each block being a higher-order monovisible block. The new system is shown in Fig. 6.

To find the transfer matrix as $H^3(s) = D(sI - A)^{-1}B$, one must initially describe the system given in Fig. 6 as a 7th order state space model. Using the RBDR algorithm, the ‘dimension’ of the problem is still 3, as in the example in section 3.1, and the calculations go exactly as in (35)–(40), except that initially the first-order blocks are substituted by

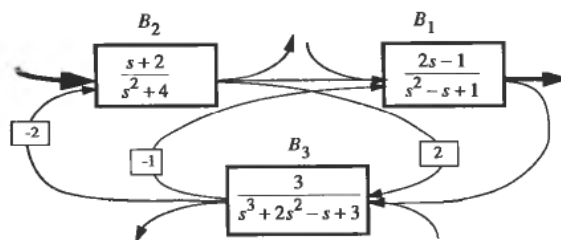


Figure 6.

the corresponding higher-order blocks. Running the algorithm on a computer gives the transfer matrix H^3 . Only one element h_{12}^3 in this matrix is given here (corresponding to input 2 and output 1 marked in thick lines in Fig. 6).

$$h_{12}^3 = \frac{2s^5 + 7s^4 + 2s^3 - 13s^2 - 7s + 6}{s^7 + s^6 + 2s^5 + 10s^4 + 6s^3 + 48s^2 + 14s + 12} \quad (41)$$

which is of order 7, but still an element in a 3×3 system, since the number of inputs and outputs are still 3 as in section 3.1.

3.3. The transfer matrix for a non-quadratic system

Until now it has been assumed that the number of inputs equals the number of outputs. This restriction is not necessary, and was used only for convenience. If we initially start with a transfer matrix with r inputs and m outputs instead of a monovaryable block, we can label this system $S_{m0,r0}$ and the transfer matrix $H^{m0,r0}$. Monovaryable blocks may then be successively connected to this system. The system remains non-quadratic. We may express this in a similar way as in (4):

$$H^{m,r} = \alpha_{m,r}(H^{m-1,r-1}, c_y^{T,m-1}, c_u^{r-1}, h_{m-m0+1}) \quad (42)$$

Note that now $\dim(c_y^{T,m-1}) = m-1$ and $\dim(c_u^{r-1}) = r-1$. Deriving the 'non-quadratic' versions of formulae (23)–(28) is straightforward and will not be done here. (For convenience, quadratic systems ($r=m=k$) will still be used in the remaining sections of this paper.)

3.4. Repetitive transfer matrix calculations for parameter changes

The RBDR algorithm may be time-saving in a situation with repetitive transfer matrix calculations for parameter changes. Let us say one wants a Nyquist plot of a high-order system S_k with respect to one varying parameter. One may do this by defining the monovaryable block containing the parameter as the *last* block in the RBDR algorithm. Then one has to do the calculation of the sub-system S_{k-1} only once, and only one iteration from S_{k-1} to S_k for each value of the parameter.

3.5. The transfer matrix for a linear system with one time delay

One may have a network of blocks with rational transfer functions and time delays. Here we will give the result for a system S_k with one time delay. The system is shown in Fig. 7. Somewhere inside the system there is a time delay. We assume that the number of outputs and inputs are both k . The method is first to disconnect the time delay and calculate a transfer matrix based on the remaining network (labeled \bar{S}_{k+1}) of

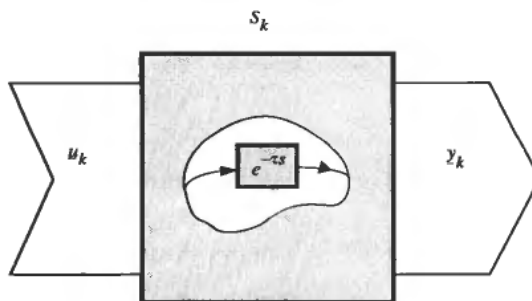


Figure 7.

rational monovariate blocks. This is illustrated in Fig. 8. This system is expanded with an 'artificial' input and output to connect the time delay to get a total system which is equivalent to the system in Fig. 7. Now we can use the RBDR algorithm to find the resulting transfer matrix of the expanded system. We are only interested in the first k inputs and outputs of this matrix. The transfer functions (= elements in the transfer matrix) will no longer be rational functions in s . We may now use the formulae (23) and (27) (second version), to find elements in the transfer matrix. (23) gives

$$\bar{q}_{k+2} = \bar{q}_{k+1} - e^{-\tau s} \bar{p}_{k+1, k+1}^{k+1} = q_k \tag{43}$$

We may now find the matrix \bar{P}^{k+2} . We are not interested in the lower row and rightmost column in this matrix, which correspond to the 'artificial' in- and outputs $k+1$ shown in Fig. 8. The remaining upper leftmost part of \bar{P}^{k+2} is the sought-after nominator matrix P^k . One component p_{ij}^k of this matrix, using (27), is then

$$p_{ij}^k = \bar{p}_{ij}^{k+1} + e^{-\tau s} \frac{\bar{p}_{i, k+1}^{k+1} \bar{p}_{k+1, j}^{k+1} - \bar{p}_{k+1, k+1}^{k+1} \bar{p}_{ij}^{k+1}}{\bar{q}_{k+1}}, \quad i, j \leq k \tag{44}$$

Observe that the common denominator q_k and the nominators p_{ij}^k of the transfer matrix of the system S_k all are of the form

$$(polynomial_1) + e^{-\tau s} (polynomial_2).$$

This holds generally for a linear multivariable system with one time delay.

Equation (44) may also be derived using the rule for reduction of a block diagram with only one loop. This is due to the fact that time delay is only connected to one input and one output of the sub-system \bar{S}_{k+1} . Figure 9 depicts the change between input j and

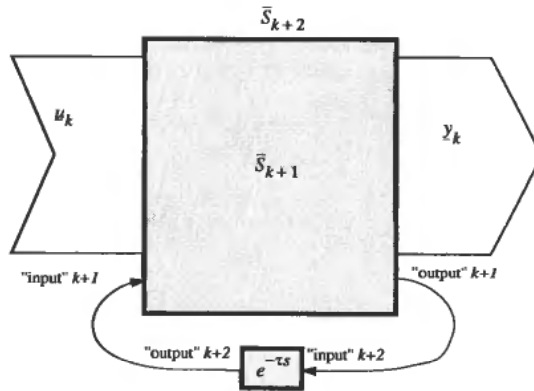


Figure 8.

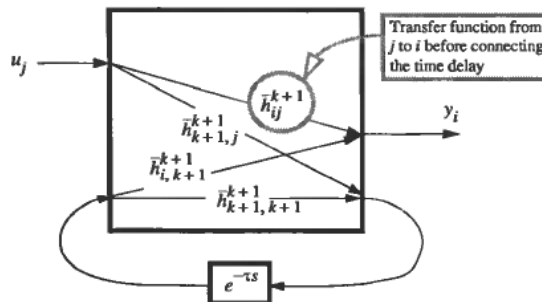


Figure 9.

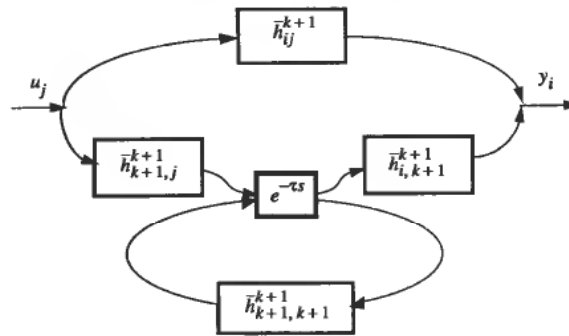


Figure 10.

output i due to the time delay and may be redrawn as shown in Fig. 10. We have two parallel forward paths, and one of these is associated with a feedback loop. Inspection of Fig. 10 gives

$$h_{ij}^k = \bar{h}_{ij}^{k+1} + \frac{\bar{h}_{k+1,j}^{k+1} e^{-\tau s} \bar{h}_{i,k+1}^{k+1}}{1 + e^{-\tau s} \bar{h}_{k+1,k+1}^{k+1}}, \quad i, j \leq k$$

Multiplying with (43), $q_k = \bar{q}_{k+1} - e^{-\tau s} \bar{p}_{k+1,k+1}^{k+1}$, on both sides gives

$$p_{ij}^k = \frac{\bar{p}_{ij}^{k+1}}{\bar{q}_{k+1}} (\bar{q}_{k+1} - e^{-\tau s} \bar{p}_{k+1,k+1}^{k+1}) + \frac{\bar{p}_{k+1,j}^{k+1} e^{-\tau s} \bar{p}_{i,k+1}^{k+1}}{\bar{q}_{k+1}}, \quad i, j \leq k \quad (45)$$

which is identical to (44).

To sum up: (43) and (44) is the rule for the change in any input-output relationship in a linear multivariable system when a time delay is introduced anywhere into the system network. One does not have to use Padé approximations or suchlike; the end result is an exact analytical expression for the (non-rational) transfer matrix.

3.6. The transfer matrix for a linear system with several time delays

The analytical expressions for the elements in the transfer matrix will be too complicated when a linear multivariable system, in addition to rational transfer functions, contains several time delays. One may, however, apply the RBDR algorithm in a 'numerical' way as now explained.

If the need for example is to do Bode or Nyquist plots, one may substitute $s = j\omega$ for a given value of ω in all monovaryable blocks (each block will now contain a complex number instead of a transfer function), and run through the RBDR algorithm. This will give one Bode or Nyquist plot point (a matrix of complex numbers if the system is multivariable) for that specific value of ω . The RBDR algorithm must then be repeated for a number of other values of ω to complete a plot. Such plots will exactly reflect the influence of the time delays in the system, as opposed to plots based on the use of (Padé) approximations.

4. Computer program

The RBDR algorithm has been implemented in FORTRAN and tried out in double precision on different computers. Up to this date, no numerical problems have been encountered, and the algorithm seems to run reasonably fast.

REFERENCES

- HOROWITZ, I. M. (1963). *Synthesis of Feedback Systems* (Academic Press, N.Y. and London).
OGATA, K. (1967). *State Space Analysis of Control Systems* (Prentice-Hall, N.-J.).