```
400 IERR=IERR+1
401 IERR=IERR+1
    IF (IORD.EQ.0) GO TO 420
    IF (IORD.EQ.MORD) IORD=IORD-1
    IF (KCH.EQ.KCHO) GO TO 415
C  FIND WHICH DATA IN YYD WILL BE USEFUL
    K=0
    W=YYD(KCHO,1)
    DO 405 I=1,IORD
      W1=YYD(KCHO,I)
      IF (ABS(W1).LT.0.4) GO TO 410
      IF (W1*W.LT.0.0) GO TO 410
      K=I
      W1=ABS(W1)
      DO 403 J=1,NP1
403   YYD(J,I)=YYD(J,I)/W1
405 CONTINUE
C  ONLY YYD(*,1) - YYD(*,K) ARE USEFUL FOR NEXT PREDICTION
410 IORD=K
    IF (IORD.EQ.0) GO TO 420
C  NOW PUSH DOWN DATA IN YYD
415 DO 418 J=1,IORD
      K=IORD+1-J
      K1=K+1
      K1P=K1+MORD
      KP=K+MORD
      DO 418 I=1,NP1
        YYD(I,K1P)=YYD(I,KP)
418   YYD(I,K1)=YYD(I,K)
C  UPDATE IORD AND KCH AND COMPLETE THE BACK SUBSTITUTION TO
C  COMPUTE YDOT : SCALE SO THAT YDOT(KCH) = + OR - 1
420 IORD=IORD+1
    KCH=KCHO
    YYD(NP1,1)=1.0
    YYD(NP1,ID2)=Y(NP1)
    IF (IORD.GT.1) YYD(NP1,1)=YYD(KCH,2)
    DO 440 I=1,N
    YYD(I,ID2)=Y(I)
      K=N+1-I
      KP1=K+1
      W=0.0
      DO 430 J=KP1,NP1
430   W=W+B(K,J)*YYD(J,1)
440 YYD(K,1)=-W/B(K,K)
C  NOW PIVOT TO COMPUTE NEW YDOT
    DO 460 I=1,N
      J=N+1-I
      K=ID(J)
      W=YYD(K,1)
      YYD(K,1)=YYD(J,1)
460 YYD(J,1)=W
    IF (IORD.GT.1) GO TO 490
C  FIRST TIME THROUGH SET UP DIRECTION SPECIFIED IN IDIR
    IF (FLOAT(IDIR)*YYD(NP1,1) .GT. 0.0) GO TO 495
    DO 480 I=1,NP1
480 YYD(I,1)=-YYD(I,1)
    GO TO 495
C  SET OUTPUT DETAILS IN INFO
490 IDIR=1
    IF (YYD(NP1,1).LT.0.0) IDIR=-1
495 IF (KCH.EQ.NP1) GO TO 500
    I=1
    IF (YYD(KCH,1).LT.0.0) I=-1
    IDET=IDET*IDIR*I
500 INFO(2)=IDET
900 INFO(1)=IERR
    INFO(3)=NS
    INFO(4)=MNS
    RETURN
    END
C
C
    FUNCTION QNORM(V,K)
C
C  COMPUTES NORM OF V = SUM OF ABSOLUTE MAGNITUDES OF COMPONENTS
C
    DIMENSION V(K)
    W=0.0
    DO 10 I=1,K
 10 W=W+ABS(V(I))
    QNORM=W
    RETURN
    END
C
    FUNCTION MOPT(N)
C
C  COMPUTES, FOR ARBITRARY N, THE OPTIMUM NUMBER OF MODIFIED NEWTON
C  ITERATIONS TO BE TAKEN BETWEEN FULL NEWTON ITERATIONS FOR A
C  PROBLEM OF DIMENSION N
C
    E=0.0
    JP1=N+1
    DO 10 K=1,JP1
      T=ALOG(FLOAT(K+1))/FLOAT(K+N)
      IF (T.LT.E) GO TO 20
      M=K
 10 E=T
 20 MOPT=M
    RETURN
    END
```

Book review

Systems: Decomposition, Optimisation and Control by M. G. Singh and A. Titli, 1978; 645 pages. *Pergamon*, £10

This book is a curious and possibly unique mixture of, on the one hand, standard textbook material on optimisation techniques and, on the other hand, more advanced material on the authors' own specialisation of hierarchical decomposition methods for solving large optimisation problems. In both cases the exposition is very clear and includes many examples, on such topics as traffic control, river pollution control, power system control. The main theme of the book is indeed the application of optimisation to control, both to the static case of steady state set point control and to the control of dynamic plant behaviour. The first three chapters present a lucid, though in some cases terse, introduction to optimisation methods, including the hierarchical approach, and to the static optimisation problem and its solution via linear programming. The hierarchical approach, presented in more detail for the static case in Chapter 4, assumes that a decomposition of the large complex system into more computationally manageable subsystems has already been obtained and this aspect of large system theory has only a brief mention in the book (rather ambiguously as it turns out, as both Analysis (section 1.2.2.) and Decomposition (section 1.2.4.) appear to be used to define this problem). The main emphasis is on solution of the resulting decomposed optimisation method. Chapter 5 introduces the optimisation problem for low order dynamic systems, with particular emphasis on the case of linear dynamics and quadratic cost function. The hierarchical extension of this topic in the large system case is developed in Chapter 6 and several nice examples are quoted. As the authors state, however, the general approach does not lead to the only practically implementable solution, i.e. closed loop control, though they provide extensions to yield this solution in the linear quadratic case. One cannot feel confident, however, in the practicality of the 20 × 20 feedback gain matrix (Table 6.6) obtained for a multi-machine power system control example. Nonlinear dynamic optimisation is treated in the same format, traditional textbook material followed by the large system hierarchical approach, in Chapters 7 and 8. The remaining chapters deal with state and parameter estimation and the stochastic control problem, again both from a tutorial and large system viewpoint. Finally robust hierarchical control schemes are studied.

The book is a valuable collection of results in the hierarchical large scale optimisation field. It is also a useful introduction to optimisation. In this sense the reader is getting two books for the price of one, which, incidentally, is reasonable owing to the direct use of the authors' typescript, which I found very readable. An excellent range of references is given and, less common, an equally excellent set of problems, unfortunately with no reference to where solutions might be obtained.

On a more technical final note, I did not find any convincing evidence in the book that the decomposition approach to large optimisation problems is more efficient in computer time or storage requirements than the global approach, save perhaps in the case of solution on truly parallel processing machines, of which few presently exist. The reference to implementation on a 1973 multi-minicomputer system at Cambridge, and to 'the price of minicomputers getting cheaper every year', I found a trifle out of date in this age of the microprocessor!

M. J. DENHAM (Kingston upon Thames)