NAG Library Routine Document

G13BJF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of **bold italicised** terms and other implementation-dependent details.

1 Purpose

G13BJF produces forecasts of a time series (the output series) which depends on one or more other (input) series via a previously estimated multi-input model for which the state set information is not available. The future values of the input series must be supplied. In contrast with G13BHF the original past values of the input and output series are required. Standard errors of the forecasts are produced. If future values of some of the input series have been obtained as forecasts using ARIMA models for those series, this may be allowed for in the calculation of the standard errors.

2 Specification

```
SUBROUTINE G13BJF (MR, NSER, MT, PARA, NPARA, KFC, NEV, NFV, XXY, LDXXY,        &
                   KZEF, RMSXY, MRX, PARX, LDPARX, FVA, FSD, STTF, ISTTF,       &
                   NSTTF, WA, IWA, MWA, IMWA, IFAIL)

INTEGER          MR(7), NSER, MT(4,NSER), NPARA, KFC, NEV, NFV, LDXXY,          &
                 KZEF, MRX(7,NSER), LDPARX, ISTTF, NSTTF, IWA,                  &
                 MWA(IMWA), IMWA, IFAIL
REAL (KIND=nag_wp) PARA(NPARA), XXY(LDXXY,NSER), RMSXY(NSER),                   &
                 PARX(LDPARX,NSER), FVA(NFV), FSD(NFV), STTF(ISTTF),            &
                 WA(IWA)
```

3 Description

G13BJF has two stages. The first stage is essentially the same as a call to the model estimation routine G13BEF, with zero iterations. In particular, all the parameters remain unchanged in the supplied input series transfer function models and output noise series ARIMA model. The internal nuisance parameters associated with the pre-observation period effects of the input series are estimated where requested, and so are any backforecasts of the output noise series. The output components z_t and n_t, and residuals a_t are calculated exactly as in Section 3 in G13BEF, and the state set for forecasting is constituted.

The second stage is essentially the same as a call to the forecasting routine G13BHF. The same information is required, and the same information is returned.

Use of G13BJF should be confined to situations in which the state set for forecasting is unknown. Forecasting from the original data is relatively expensive because it requires recalculation of the state set. G13BJF returns the state set for use in producing further forecasts using G13BHF, or for updating the state set using G13BGF.

4 References

Box G E P and Jenkins G M (1976) *Time Series Analysis: Forecasting and Control* (Revised Edition) Holden–Day

5 Parameters

1: MR(7) – INTEGER array   *Input*

   On entry: the orders vector (p, d, q, P, D, Q, s) of the ARIMA model for the output noise component.

   p, q, P and Q refer respectively to the number of autoregressive (ϕ), moving average (θ), seasonal autoregressive (Φ) and seasonal moving average (Θ) parameters.

d, D and s refer respectively to the order of non-seasonal differencing, the order of seasonal differencing and the seasonal period.

Constraints:

$$p, d, q, P, D, Q, s \geq 0;$$
$$p + q + P + Q > 0;$$
$$s \neq 1;$$
$$\text{if } s = 0, \ P + D + Q = 0;$$
$$\text{if } s > 1, \ P + D + Q > 0;$$
$$d + s \times (P + D) \leq n;$$
$$p + d - q + s \times (P + D - Q) \leq n.$$

2:     NSER – INTEGER   *Input*

On entry: the number of input and output series. There may be any number of input series (including none), but only one output series.

3:     MT(4,NSER) – INTEGER array   *Input*

On entry: the transfer function model orders b, p and q of each of the input series. The data for input series i is held in column i. Row 1 holds the value b_i, row 2 holds the value q_i and row 3 holds the value p_i.

For a simple input, $b_i = q_i = p_i = 0$.

Row 4 holds the value r_i, where $r_i = 1$ for a simple input, and $r_i = 2$ or 3 for a transfer function input.

The choice $r_i = 3$ leads to estimation of the pre-period input effects as nuisance parameters, and $r_i = 2$ suppresses this estimation. This choice may affect the returned forecasts and the state set.

When $r_i = 1$, any nonzero contents of rows 1, 2 and 3 of column i are ignored.

Constraint: $\text{MT}(4, i) = 1$, 2 or 3, for $i = 1, 2, \ldots, \text{NSER} - 1$.

4:     PARA(NPARA) – REAL (KIND=nag_wp) array   *Input/Output*

On entry: estimates of the multi-input model parameters. These are in order, firstly the ARIMA model parameters: p values of ϕ parameters, q values of θ parameters, P values of Φ parameters, Q values of Θ parameters.

These are followed by the transfer function model parameter values $\omega_0, \omega_1, \ldots, \omega_{q_1}, \delta_1, \ldots, \delta_{p_1}$ for the first of any input series and similarly for each subsequent input series. The final component of PARA is the value of the constant c.

On exit: the parameter values may be updated using an additional iteration in the estimation process.

5:     NPARA – INTEGER   *Input*

On entry: the exact number of ϕ, θ, Φ, Θ, ω, δ, c parameters, so that $\text{NPARA} = p + q + P + Q + \text{NSER} + \sum(p + q)$, the summation being over all the input series. (c must be included whether its value was previously estimated or was set fixed.)

6:     KFC – INTEGER   *Input*

On entry: must be set to 1 if the constant was estimated when the model was fitted, and 0 if it was held at a fixed value. This only affects the degrees of freedom used in calculating the estimated residual variance.

Constraint: $\text{KFC} = 0$ or 1.

7:     NEV – INTEGER   *Input*

On entry: the number of original (undifferenced) values in each of the input and output time series.

8:      NFV – INTEGER   *Input*

        On entry: the number of forecast values of the output series required.

        Constraint: NFV > 0.

9:      XXY(LDXXY,NSER) – REAL (KIND=nag_wp) array   *Input/Output*

        On entry: the columns of XXY must contain in the first NEV places, the past values of each of the
        input and output series, in that order. In the next NFV places, the columns relating to the input
        series (i.e., columns 1 to NSER − 1) contain the future values of the input series which are
        necessary for construction of the forecasts of the output series y.

        On exit: if KZEF = 0 then XXY is unchanged except that the relevant NFV values in the column
        relating to the output series (column NSER) contain the forecast values (FVA), but if KZEF ≠ 0
        then the columns of XXY contain the corresponding values of the input component series z_t and the
        values of the output noise component n_t, in that order.

10:     LDXXY – INTEGER   *Input*

        On entry: the first dimension of the array XXY as declared in the (sub)program from which G13BJF
        is called.

        Constraint: LDXXY ≥ (NEV + NFV).

11:     KZEF – INTEGER   *Input*

        On entry: must be set to 0 if the relevant NFV values of the forecasts (FVA) are to be held in the
        output series column (NSER) of XXY (which is otherwise unchanged) on exit, and must not be set
        to 0 if the values of the input component series z_t and the values of the output noise component n_t
        are to overwrite the contents of XXY on exit.

12:     RMSXY(NSER) – REAL (KIND=nag_wp) array   *Input/Output*

        On entry: the first (NSER − 1) elements of RMSXY must contain the estimated residual variance of
        the input series ARIMA models. In the case of those inputs for which no ARIMA model is
        available or its effects are to be excluded in the calculation of forecast standard errors, the
        corresponding entry of RMSXY should be set to 0.

        On exit: RMSXY(NSER) contains the estimated residual variance of the output noise ARIMA
        model which is calculated from the supplied series. Otherwise RMSXY is unchanged.

13:     MRX(7,NSER) – INTEGER array   *Input/Output*

        On entry: the orders array for each of the input series ARIMA models. Thus, column i contains
        values of p, d, q, P, D, Q, s for input series i. In the case of those inputs for which no ARIMA
        model is available, the corresponding orders should be set to 0.

        On exit: unchanged, except for column NSER which is used as workspace.

14:     PARX(LDPARX,NSER) – REAL (KIND=nag_wp) array   *Input*

        On entry: values of the parameters (ϕ, θ, Φ, and Θ) for each of the input series ARIMA models.

        Thus column i contains MRX(1, i) values of ϕ, MRX(3, i) values of θ, MRX(4, i) values of Φ and
        MRX(6, i) values of Θ, in that order.

        Values in the columns relating to those input series for which no ARIMA model is available are
        ignored.

15:     LDPARX – INTEGER   *Input*

        On entry: the first dimension of the array PARX as declared in the (sub)program from which
        G13BJF is called.

        Constraint: LDPARX ≥ nce, where nce is the maximum number of parameters in any of the input
        series ARIMA models. If there are no input series, then LDPARX ≥ 1.

16:     FVA(NFV) – REAL (KIND=nag_wp) array   *Output*

   On exit: the required forecast values for the output series. (Note that these are also output in column NSER of XXY if KZEF = 0.)

17:     FSD(NFV) – REAL (KIND=nag_wp) array   *Output*

   On exit: the standard errors for each of the forecast values.

18:     STTF(ISTTF) – REAL (KIND=nag_wp) array   *Output*

   On exit: the NSTTF values of the state set based on the first NEV sets of (past) values of the input and output series.

19:     ISTTF – INTEGER   *Input*

   On entry: the dimension of the array STTF as declared in the (sub)program from which G13BJF is called.

   Constraint: $\text{ISTTF} \geq (P \times s) + d + (D \times s) + q + \max(p, Q \times s) + ncf$, where $ncf = \sum(b_i + q_i + p_i)$ and the summation is over all input series for which $r_i > 1$.

20:     NSTTF – INTEGER   *Output*

   On exit: the number of values in the state set array STTF.

21:     WA(IWA) – REAL (KIND=nag_wp) array   *Workspace*
22:     IWA – INTEGER   *Input*

   On entry: the dimension of the array WA as declared in the (sub)program from which G13BJF is called.

   It is not practical to outline a method for deriving the exact minimum permissible value of IWA, but the following gives a reasonably good approximation which tends to be on the conservative side.

   Note: there are three error indicators associated with IWA. These are IFAIL = 4, 5 or 6. The first of these probably indicates an abnormal entry value of NFV, while the second indicates that IWA is much too small and needs to be increased by a substantial amount. The last of these indicates that IWA is too small but returns the necessary minimum value in MWA(1).

   Let $q' = q + (Q \times s)$ and $d' = d + (D \times s)$, where the output noise ARIMA model orders are p, d, q, P, D, Q, s.

   Let there be l input series, where $l = \text{NSER} - 1$.

   Let $\qquad\qquad\qquad\qquad mx_i = \max(b_i + q_i, p_i),$
   if $\quad r_i = 3$, for
   $\qquad i = 1, 2, \ldots, l$, if
   $\qquad l > 0$
   $\qquad mx_i = 0, \qquad\qquad$ if $r_i \neq 3$, for $i = 1, 2, \ldots, l$, if $l > 0$

   where the transfer function model orders of input series i are given by b_i, q_i, p_i, r_i.

   Let $qx = \max(q', mx_1, mx_2, \ldots, mx_l)$

   Let $ncg = \text{NPARA} + qx + \sum_{i=1}^{l} mx_i$ and $nch = N + d + 6 \times qx$.

   Finally, let $nci = \text{NSER}$, and then increment nci by 1 every time any of the following conditions are satisfied. (The last two conditions should be applied separately to each input series, so that for example if we have two input series and $p_1 > 0$ and $p_2 > 0$, then nci is incremented by 2 in respect of these.)

   The conditions are:

   $\qquad p > 0$
   $\qquad q > 0$

$$P > 0$$
$$Q > 0$$
$$qx > 0$$
$$mx_i > 0, \text{ separately, for } i = 1, 2, \dots, l, \text{ if } l > 0$$
$$p_i > 0, \text{ separately, for } i = 1, 2, \dots, l, \text{ if } l > 0,$$

then $\text{IWA} > 2 \times (ncg)^2 + nch \times (nci + 4)$.

23: MWA(IMWA) – INTEGER array   *Workspace*
24: IMWA – INTEGER   *Input*

On entry: the dimension of the array MWA as declared in the (sub)program from which G13BJF is called.

Constraint: $\text{IMWA} \geq (16 \times \text{NSER}) + (7 \times ncg) + (3 \times \text{NPARA}) + 27$.

The derivation of ncg is described under IWA.

When IMWA is too small, as indicated by $\text{IFAIL} = 7$, the requisite minimum value of IMWA is returned in MWA(1).

25: IFAIL – INTEGER   *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: $\text{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6   Error Indicators and Warnings

If on entry $\text{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\text{IFAIL} = 1$

    On entry, KFC < 0,
    or   KFC > 1,
    or   LDXXY $< (\text{NEV} + \text{NFV})$,
    or   NFV ≤ 0.

$\text{IFAIL} = 2$

    On entry, LDPARX is too small or NPARA is inconsistent with the orders specified in arrays MR and MT; or one of the r_i, stored in $\text{MT}(4, i)$, does not equal 1, 2 or 3.

$\text{IFAIL} = 3$

    On entry or during execution, one or more sets of δ parameters do not satisfy the stationarity or invertibility test conditions.

$\text{IFAIL} = 4$

    On entry, IWA is too small for the final forecasting calculations. This is a highly unlikely error, and would probably indicate that NFV was abnormally large.

IFAIL = 5

On entry, IWA is too small by a very considerable margin. No information is supplied about the requisite minimum size.

IFAIL = 6

On entry, IWA is too small, but the requisite minimum size is returned in MWA(1).

IFAIL = 7

On entry, IMWA is too small, but the requisite minimum size is returned in MWA(1).

IFAIL = 8

This indicates a failure in F04ASF which is used to solve the equations giving the latest estimates of the parameters.

IFAIL = 9

This indicates a failure in the inversion of the second derivative matrix associated with parameter estimation.

IFAIL = 10

On entry or during execution, one or more sets of the ARIMA (ϕ, θ, Φ or Θ) parameters do not satisfy the stationarity or invertibility test conditions.

IFAIL = 11

On entry, ISTTF is too small.

7   Accuracy

The computations are believed to be stable.

8   Further Comments

The time taken by G13BJF is approximately proportional to the product of the length of each series and the square of the number of parameters in the multi-input model.

9   Example

The data in this example relates to 40 observations of an output time series and 5 input time series. The output series has one autoregressive (ϕ) parameter and one seasonal moving average (Θ) parameter, with initial values $\phi = 0.495$, $\Theta = 0.238$ and $c = -82.858$. The seasonal period is 4. This example differs from the example in G13BEF in that four of the input series are simple series and the fifth is defined by a transfer function with orders $b_5 = 1$, $q_5 = 0$, $p_5 = 1$, $r_5 = 3$, which allows for pre-observation period effects. The initial values for the transfer model are:

$$\omega_1 = -0.367, \qquad \omega_2 = -3.876, \qquad \omega_3 = 4.516, \qquad \omega_4 = 2.474 \qquad \omega_5 = 8.629, \qquad \delta_1 = 0.688.$$

A further 8 values of the input series are supplied, and it is assumed that the values for the fifth series have themselves been forecast from an ARIMA model with orders 2 0 2 0 1 1 4, in which $\phi_1 = 1.6743$, $\phi_2 = -0.9505$, $\theta_1 = 1.4605$, $\theta_2 = -0.4862$ and $\Theta_1 = 0.8993$, and for which the residual mean square is 0.1720.

The following are computed and printed out: the state set after initial processing of the original 40 sets of values, the estimated residual variance for the output noise series, the 8 forecast values and their standard errors, and the values of the components z_t and the output noise component n_t.

9.1   Program Text

```
      Program g13bjfe

!     G13BJF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g13bjf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: dp, i, ifail, imwa, isttf, iwa, kfc, &
                                          kzef, ldparx, ldxxy, mx, n, ncf,     &
                                          ncg, nch, nci, nev, nfv, nis, npara, &
                                          nparx, nser, nsttf, qp, qx, smx
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: fsd(:), fva(:), para(:), parx(:,:),  &
                                          rmsxy(:), sttf(:), wa(:), xxy(:,:)
      Integer                          :: mr(7)
      Integer, Allocatable             :: mrx(:,:), mt(:,:), mwa(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, sum
!     .. Executable Statements ..
      Write (nout,*) 'G13BJF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in problem size
      Read (nin,*) kfc, nev, nfv, nser, kzef

!     Number of input series
      nis = nser - 1

      Allocate (mt(4,nser))

!     Read in the orders for the output noise
      Read (nin,*) mr(1:7)

!     Read in transfer function
      Do i = 1, nis
        Read (nin,*) mt(1:4,i)
      End Do

!     Calculate NPARA
      npara = 0
      Do i = 1, nis
        npara = npara + mt(2,i) + mt(3,i)
      End Do
      npara = npara + mr(1) + mr(3) + mr(4) + mr(6) + nser

!     Calculate array sizes
      n = nev + nfv
      ldxxy = n
      ncf = 0
      Do i = 1, nis
        If (mt(4,i)>1) Then
          ncf = sum(mt(1:3,i))
        End If
      End Do
      isttf = mr(4)*mr(7) + mr(2) + mr(5)*mr(7) + mr(3) + &
        max(mr(1),mr(6)*mr(7)) + ncf
      qp = mr(3) + mr(6)*mr(7)
      dp = mr(2) + mr(5)*mr(7)
      smx = 0
      qx = qp
```

```
        nci = nser
        Do i = 1, nis
          If (mt(4,i)==3) Then
            mx = max(mt(1,i)+mt(2,i),mt(3,i))
            nci = nci + 1
          Else
            mx = 0
          End If
          If (mt(3,i)>0) Then
            nci = nci + 1
          End If
          smx = smx + mx
          qx = max(qx,mx)
        End Do
        ncg = npara + qx + smx
        nch = dp + 6*qx + nev
        If (qx>0) Then
          nci = nci + 1
        End If
        If (mr(1)>0) Then
          nci = nci + 1
        End If
        If (mr(3)>0) Then
          nci = nci + 1
        End If
        If (mr(4)>0) Then
          nci = nci + 1
        End If
        If (mr(6)>0) Then
          nci = nci + 1
        End If
        iwa = 2*(ncg**2) + nch*(nci+4)
        imwa = 16*nser + 7*ncg + 3*npara + 27
        Allocate (para(npara),xxy(ldxxy,nser),rmsxy(nser),mrx(7,nser),fva(nfv), &
          fsd(nfv),sttf(isttf),wa(iwa),mwa(imwa))

!       Read in multi-input model parameters
        Read (nin,*) para(1:npara)

!       Read in the observed values for the input and output series
        Read (nin,*)(xxy(i,1:nser),i=1,nev)

!       Read in the future values for the input series
        Read (nin,*)(xxy(nev+i,1:nis),i=1,nfv)

        If (nis>=1) Then
!         Read in residual variance of input series
          Read (nin,*) rmsxy(1:nis)

!         Read in orders for input series ARIMA where available
!         (i.e. where residual variance is not zero)
          ldparx = 0
          Do i = 1, nis
            If (rmsxy(i)/=0.0E0_nag_wp) Then
              Read (nin,*) mrx(1:7,i)
              nparx = mrx(1,i) + mrx(3,i) + mrx(4,i) + mrx(6,i)
              ldparx = max(ldparx,nparx)
            Else
              mrx(1:7,i) = 0
            End If
          End Do
        Else
!         No input series
          ldparx = 1
        End If

        Allocate (parx(ldparx,nser))

!       Read in parameters for each input series ARIMA
        If (nis>0) Then
          Do i = 1, nis
```

```
          If (rmsxy(i)/=0.0E0_nag_wp) Then
            nparx = mrx(1,i) + mrx(3,i) + mrx(4,i) + mrx(6,i)
            If (nparx>0) Then
              Read (nin,*) parx(1:nparx,i)
            End If
          End If
        End Do
      End If

      ifail = 0
      Call g13bjf(mr,nser,mt,para,npara,kfc,nev,nfv,xxy,ldxxy,kzef,rmsxy,mrx, &
        parx,ldparx,fva,fsd,sttf,isttf,nsttf,wa,iwa,mwa,imwa,ifail)

!     Display results
      Write (nout,99999) 'After processing', nev, ' sets of observations'
      Write (nout,99998) nsttf, ' values of the state set are derived'
      Write (nout,*)
      Write (nout,99997) sttf(1:nsttf)
      Write (nout,*)
      Write (nout,*) 'The residual mean square for the output'
      Write (nout,99996) 'series is also derived and its value is', &
        rmsxy(nser)
      Write (nout,*)
      Write (nout,*) 'The forecast values and their standard errors are'
      Write (nout,*)
      Write (nout,*) '    I       FVA        FSD'
      Write (nout,*)
      Write (nout,99995)(i,fva(i),fsd(i),i=1,nfv)
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',n,nser,xxy,ldxxy, &
        'The values of z(t) and n(t) are',ifail)
      Write (nout,99994) 'The first ', nis, &
        ' columns hold the z(t) and the last column the n(t)'

99999 Format (1X,A,I3,A)
99998 Format (1X,I3,A)
99997 Format (1X,6F10.4)
99996 Format (1X,A,F10.4)
99995 Format (1X,I4,F10.3,F10.4)
99994 Format (1X,A,I0,A)
    End Program g13bjfe
```

9.2 Program Data

```
G13BJF Example Program Data
 1   40    8    6    1                       :: KFC,NEV,NFV,NSER,KZEF
 1    0    0    0    0    1    4              :: MR
0  0  0  1                                   :: Transfer fun. series 1 MT(:,1)
0  0  0  1                                   :: Transfer fun. series 2 MT(:,2)
0  0  0  1                                   :: Transfer fun. series 3 MT(:,3)
0  0  0  1                                   :: Transfer fun. series 4 MT(:,4)
1  0  1  3                                   :: Transfer fun. series 5 MT(:,5)
0.4950  0.2380 -0.3670  -3.8760   4.5160
2.4740  8.6290  0.6880 -82.8580              :: End of PARA
   1.0     1.0     0.0     0.0    8.075 105.0
   1.0     0.0     1.0     0.0    7.819 119.0
   1.0     0.0     0.0     1.0    7.366 119.0
   1.0    -1.0    -1.0    -1.0    8.113 109.0
   2.0     1.0     0.0     0.0    7.380 117.0
   2.0     0.0     1.0     0.0    7.134 135.0
   2.0     0.0     0.0     1.0    7.222 126.0
   2.0    -1.0    -1.0    -1.0    7.768 112.0
   3.0     1.0     0.0     0.0    7.386 116.0
   3.0     0.0     1.0     0.0    6.965 122.0
   3.0     0.0     0.0     1.0    6.478 115.0
   3.0    -1.0    -1.0    -1.0    8.105 115.0
   4.0     1.0     0.0     0.0    8.060 122.0
   4.0     0.0     1.0     0.0    7.684 138.0
```

```
  4.0     0.0     0.0     1.0     7.580 135.0
  4.0    -1.0    -1.0    -1.0     7.093 125.0
  5.0     1.0     0.0     0.0     6.129 115.0
  5.0     0.0     1.0     0.0     6.026 108.0
  5.0     0.0     0.0     1.0     6.679 100.0
  5.0    -1.0    -1.0    -1.0     7.414  96.0
  6.0     1.0     0.0     0.0     7.112 107.0
  6.0     0.0     1.0     0.0     7.762 115.0
  6.0     0.0     0.0     1.0     7.645 123.0
  6.0    -1.0    -1.0    -1.0     8.639 122.0
  7.0     1.0     0.0     0.0     7.667 128.0
  7.0     0.0     1.0     0.0     8.080 136.0
  7.0     0.0     0.0     1.0     6.678 140.0
  7.0    -1.0    -1.0    -1.0     6.739 122.0
  8.0     1.0     0.0     0.0     5.569 102.0
  8.0     0.0     1.0     0.0     5.049 103.0
  8.0     0.0     0.0     1.0     5.642  89.0
  8.0    -1.0    -1.0    -1.0     6.808  77.0
  9.0     1.0     0.0     0.0     6.636  89.0
  9.0     0.0     1.0     0.0     8.241  94.0
  9.0     0.0     0.0     1.0     7.968 104.0
  9.0    -1.0    -1.0    -1.0     8.044 108.0
 10.0     1.0     0.0     0.0     7.791 119.0
 10.0     0.0     1.0     0.0     7.024 126.0
 10.0     0.0     0.0     1.0     6.102 119.0
 10.0    -1.0    -1.0    -1.0     6.053 103.0 :: XXY (observed values)
 11.0     1.0     0.0     0.0     5.941
 11.0     0.0     1.0     0.0     5.386
 11.0     0.0     0.0     1.0     5.811
 11.0    -1.0    -1.0    -1.0     6.716
 12.0     1.0     0.0     0.0     6.923
 12.0     0.0     1.0     0.0     6.939
 12.0     0.0     0.0     1.0     6.705
 12.0    -1.0    -1.0    -1.0     6.914              :: End of XXY (future values)
  0.0     0.0     0.0     0.0     0.1720             :: End of RMSXY
  2  0  2  0  1  1  4                                :: Orders for series 5, MRX(:,5)
1.6743  -0.9505  1.4605  -0.4862  0.8993            :: Params for series 5, PARX(:,5)
```

9.3 Program Results

```
G13BJF Example Program Results

After processing 40 sets of observations
  6 values of the state set are derived

    6.0530  193.8741    2.0790   -2.8580   -3.5906   -2.5203

The residual mean square for the output
series is also derived and its value is   20.7599

The forecast values and their standard errors are

   I      FVA       FSD

   1     93.398    4.5563
   2     96.958    6.2172
   3     86.046    7.0933
   4     77.589    7.3489
   5     82.139    7.3941
   6     96.276    7.5823
   7     98.345    8.1445
   8     93.577    8.8536

The values of z(t) and n(t) are
              1         2         3         4         5         6
   1     -0.3391   -3.8886    0.0000    0.0000  188.6028  -79.3751
   2     -0.3391   -0.0000    4.5139    0.0000  199.4379  -84.6127
   3     -0.3391   -0.0000    0.0000    2.4789  204.6834  -87.8232
   4     -0.3391    3.8886   -4.5139   -2.4789  204.3834  -91.9402
   5     -0.6782   -3.8886    0.0000    0.0000  210.6229  -89.0560
```

```
 6     -0.6782     -0.0000      4.5139      0.0000    208.5905    -77.4262
 7     -0.6782     -0.0000      0.0000      2.4789    205.0696    -80.8703
 8     -0.6782      3.8886     -4.5139     -2.4789    203.4065    -87.6242
 9     -1.0173     -3.8886      0.0000      0.0000    206.9738    -86.0678
10     -1.0173     -0.0000      4.5139      0.0000    206.1317    -87.6283
11     -1.0173     -0.0000      0.0000      2.4789    201.9196    -88.3812
12     -1.0173      3.8886     -4.5139     -2.4789    194.8194    -75.6979
13     -1.3564     -3.8886      0.0000      0.0000    203.9738    -76.7287
14     -1.3564     -0.0000      4.5139      0.0000    209.8837    -75.0412
15     -1.3564     -0.0000      0.0000      2.4789    210.7052    -76.8277
16     -1.3564      3.8886     -4.5139     -2.4789    210.3730    -80.9125
17     -1.6955     -3.8886      0.0000      0.0000    205.9421    -85.3580
18     -1.6955     -0.0000      4.5139      0.0000    194.5753    -89.3937
19     -1.6955     -0.0000      0.0000      2.4789    185.8662    -86.6496
20     -1.6955      3.8886     -4.5139     -2.4789    185.5090    -84.7094
21     -2.0346     -3.8886      0.0000      0.0000    191.6056    -78.6824
22     -2.0346     -0.0000      4.5139      0.0000    193.1941    -80.6734
23     -2.0346     -0.0000      0.0000      2.4789    199.8958    -77.3402
24     -2.0346      3.8886     -4.5139     -2.4789    203.4970    -76.3583
25     -2.3737     -3.8886      0.0000      0.0000    214.5519    -80.2896
26     -2.3737     -0.0000      4.5139      0.0000    213.7702    -79.9104
27     -2.3737     -0.0000      0.0000      2.4789    216.7963    -76.9015
28     -2.3737      3.8886     -4.5139     -2.4789    206.7803    -79.3024
29     -2.7128     -3.8886      0.0000      0.0000    200.4157    -91.8142
30     -2.7128     -0.0000      4.5139      0.0000    185.9409    -84.7420
31     -2.7128     -0.0000      0.0000      2.4789    171.4951    -82.2613
32     -2.7128      3.8886     -4.5139     -2.4789    166.6735    -83.8565
33     -3.0519     -3.8886      0.0000      0.0000    173.4176    -77.4771
34     -3.0519     -0.0000      4.5139      0.0000    176.5733    -84.0353
35     -3.0519     -0.0000      0.0000      2.4789    192.5940    -88.0211
36     -3.0519      3.8886     -4.5139     -2.4789    201.2606    -87.1045
37     -3.3910     -3.8886      0.0000      0.0000    207.8790    -81.5993
38     -3.3910     -0.0000      4.5139      0.0000    210.2493    -85.3721
39     -3.3910     -0.0000      0.0000      2.4789    205.2616    -85.3495
40     -3.3910      3.8886     -4.5139     -2.4789    193.8741    -84.3790
41     -3.7301     -3.8886      0.0000      0.0000    185.6167    -84.6003
42     -3.7301      0.0000      4.5139      0.0000    178.9692    -82.7953
43     -3.7301      0.0000      0.0000      2.4789    169.6066    -82.3091
44     -3.7301      3.8886     -4.5139     -2.4789    166.8325    -82.4095
45     -4.0692     -3.8886      0.0000      0.0000    172.7331    -82.6360
46     -4.0692      0.0000      4.5139      0.0000    178.5789    -82.7481
47     -4.0692      0.0000      0.0000      2.4789    182.7389    -82.8036
48     -4.0692      3.8886     -4.5139     -2.4789    183.5818    -82.8311
The first 5 columns hold the z(t) and the last column the n(t)
```
_____