NAG Toolbox for MATLAB

g13bj

1 Purpose

g13bj 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 g13bh 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 Syntax

```
[para, xxy, rmsxy, mrx, fva, fsd, sttf, nsttf, ifail] = g13bj(mr, mt,
para, kfc, nev, nfv, xxy, kzef, rmsxy, mrx, parx, isttf, 'nser', nser,
'npara', npara)
```

3 Description

g13bj has two stages. The first stage is essentially the same as a call to the model estimation function g13be, 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 of the document for g13be, and the state set for forecasting is constituted.

The second stage is essentially the same as a call to the forecasting function g13bh. The same information is required, and the same information is returned.

Use of g13bj 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. g13bj returns the state set for use in producing further forecasts using g13bh, or for updating the state set using g13bg.

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **mr(7) – int32 array**

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

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

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

Constraints:

$p, d, q, P, D, Q, s \geq 0$;
$p + q + P + Q > 0$;
$s \neq 1$;

if $s = 0$, $P + D + Q = 0$;
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: **mt(4,nser) – int32 array**

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: $\mathbf{mt}(4, i) = 1$, 2 or 3, for $i = 1, 2, \ldots,$ **nser** $- 1$.

3: **para(npara) – double array**

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.

4: **kfc – int32 scalar**

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: **kfc** $= 0$ or 1.

5: **nev – int32 scalar**

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

6: **nfv – int32 scalar**

the number of forecast values of the output series required.

Constraint: **nfv** > 0.

7: **xxy(ldxxy,nser) – double array**

ldxxy, the first dimension of the array, must be at least (**nev** + **nfv**).

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.

8: **kzef – int32 scalar**

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.

9: **rmsxy**(**nser**) **– double array**

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.

10: **mrx**(**7,nser**) **– int32 array**

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.

11: **parx**(**ldparx,nser**) **– double array**

ldparx, the first dimension of the array, must be at least nce, where nce is the maximum number of parameters in any of the input series ARIMA models. If there are no input series, then 1.

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.

12: **isttf – int32 scalar**

Constraint: **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 $r_i > 1$.

5.2 Optional Input Parameters

1: **nser – int32 scalar**

Default: The dimension of the arrays **mt**, **mrx**, **xxy**, **rmsxy**, **parx**. (An error is raised if these dimensions are not equal.)

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

2: **npara – int32 scalar**

Default: The dimension of the array **para**.

the exact number of ϕ, θ, Φ, Θ, ω, δ, c parameters, so that **npara** $= p + q + P + Q +$ **nser** $+ \sum(p_i + q_i)$, the summation being over all the input series. (c must be included whether its value was previously estimated or was set fixed.)

5.3 Input Parameters Omitted from the MATLAB Interface

ldxxy, ldparx, wa, iwa, mwa, imwa

5.4 Output Parameters

1: **para**(**npara**) **– double array**

The parameter values may be updated using an additional iteration in the estimation process.

2: **xxy**(**ldxxy,nser**) **– double array**

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** $\neq 0$ 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.

3: **rmsxy**(**nser**) – **double array**

rmsxy(**nser**) contains the estimated residual variance of the output noise ARIMA model which is calculated from the supplied series. Otherwise **rmsxy** is unchanged.

4: **mrx**(**7**,**nser**) – **int32 array**

Unchanged, except for column **nser** which is used as workspace.

5: **fva**(**nfv**) – **double array**

The required forecast values for the output series. (Note that these are also output in column **nser** of **xxy** if **kzef** $= 0$.)

6: **fsd**(**nfv**) – **double array**

The standard errors for each of the forecast values.

7: **sttf**(**isttf**) – **double array**

The **nsttf** values of the state set based on the first **nev** sets of (past) values of the input and output series.

8: **nsttf** – **int32 scalar**

The number of values in the state set array **sttf**.

9: **ifail** – **int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **kfc** < 0,
or **kfc** > 1,
or **ldxxy** $<$ (**nev** $+$ **nfv**),
or **nfv** ≤ 0.

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 **mt**$(4, i)$, does not equal 1, 2 or 3.

ifail $= 3$

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

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 f04as, 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 g13bj 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

```
mr  =  [int32(1);  int32(0);  int32(0);  int32(0);  int32(0);  int32(1);
int32(4)];
mt = [int32(0), int32(0), int32(0), int32(0), int32(1), int32(0);
      int32(0), int32(0), int32(0), int32(0), int32(0), int32(0);
      int32(0), int32(0), int32(0), int32(0), int32(1), int32(0);
      int32(1), int32(1), int32(1), int32(1), int32(3), int32(0)];
para   =   [0.495;   0.238;   -0.367;   -3.876;   4.516;   2.474;   8.629;
0.6879999999999999; -82.858];
kfc = int32(1);
nev = int32(40);
nfv = int32(8);
xxy = [1, 1, 0, 0, 8.074999999999999, 105; 1, 0, 1, 0, 7.819, 119; 1, 0,
...
    0, 1, 7.366, 119; 1, -1, -1, -1, 8.113, 109; 2, 1, 0, 0, 7.38, 117;
2, ...
    0, 1, 0, 7.134, 135; 2, 0, 0, 1, 7.222, 126; 2, -1, -1, -1, 7.768,
112; 3, ...
    1, 0, 0, 7.386, 116; 3, 0, 1, 0, 6.965, 122; 3, 0, 0, 1, 6.478, 115;
3, ...
    -1, -1, -1, 8.105, 115; 4, 1, 0, 0, 8.06, 122; 4, 0, 1, 0, 7.684,
138; 4, ...
    0, 0, 1, 7.58, 135; 4, -1, -1, -1, 7.093, 125; 5, 1, 0, 0, 6.129,
115; 5, ...
    0, 1, 0, 6.026, 108; 5, 0, 0, 1, 6.679, 100; 5, -1, -1, -1, 7.414,
96; 6, ...
```

```
    1, 0, 0, 7.112, 107; 6, 0, 1, 0, 7.762, 115; 6, 0, 0, 1, 7.645, 123;
6, ...
    -1, -1, -1, 8.638999999999999, 122; 7, 1, 0, 0, 7.667, 128; 7, 0, 1,
0, ...
     8.08, 136; 7, 0, 0, 1, 6.678, 140; 7, -1, -1, -1, 6.739, 122; 8, 1,
...
    0, 0, 5.569, 102; 8, 0, 1, 0, 5.049, 103; 8, 0, 0, 1, 5.642, 89; 8, -
1, -1, ...
    -1, 6.808, 77; 9, 1, 0, 0, 6.636, 89; 9, 0, 1, 0, 8.241, 94; 9, 0, 0,
...
     1, 7.968, 104; 9, -1, -1, -1, 8.044, 108; 10, 1, 0, 0, 7.791, 119;
10, 0, ...
     1, 0, 7.024, 126; 10, 0, 0, 1, 6.102, 119; 10, -1, -1, -1, 6.053,
103; 11, ...
    1, 0, 0, 5.941, 0; 11, 0, 1, 0, 5.386, 0; 11, 0, 0, 1, 5.811, 0; 11,
...
    -1, -1, -1, 6.716, 0; 12, 1, 0, 0, 6.923, 0; 12, 0, 1, 0, 6.939, 0;
12, ...
    0, 0, 1, 6.705, 0; 12, -1, -1, -1, 6.914, 0; 0, 0, 0, 0, 0, 0; 0, 0,
0, 0, 0, 0];
kzef = int32(1);
rmsxy = [0; 0; 0; 0; 0.172; 0];
mrx  =  [int32(0),  int32(0),  int32(0),  int32(0),  int32(2),  int32(0);
int32(0), ...
    int32(0), int32(0), int32(0), int32(0), int32(0); int32(0), int32(0),
...
    int32(0), int32(0), int32(2), int32(0); int32(0), int32(0), int32(0),
int32(0), ...
    int32(0), int32(0); int32(0), int32(0), int32(0), int32(0), int32(1),
...
    int32(0); int32(0), int32(0), int32(0), int32(0), int32(1), int32(0);
int32(0), ...
     int32(0), int32(0), int32(0), int32(4), int32(0)];
parx = [0, 0, 0, 0, 1.6743, 0; 0, 0, 0, 0, -0.9505, 0; 0, 0, 0, 0,
1.4605, ...
     0; 0, 0, 0, 0, -0.4862, 0; 0, 0, 0, 0, 0.8993, 0];
isttf = int32(20);
[paraOut, xxyOut, rmsxyOut, mrxOut, fva, fsd, sttf, nsttf, ifail] = ...
     g13bj(mr, mt, para, kfc, nev, nfv, xxy, kzef, rmsxy, mrx, parx,
isttf)
```

```
paraOut =
    0.4950
    0.2380
   -0.3391
   -3.8886
    4.5139
    2.4789
    8.6290
    0.6880
  -82.8580
xxyOut =
     array elided
rmsxyOut =
         0
         0
         0
         0
    0.1720
   20.7599
mrxOut =
           0           0           0           0           2           1
           0           0           0           0           0           0
           0           0           0           0           2           0
           0           0           0           0           0           0
           0           0           0           0           1           0
           0           0           0           0           1           1
           0           0           0           0           4           4
fva =
   93.3977
```

```
   96.9577
   86.0463
   77.5887
   82.1393
   96.2755
   98.3451
   93.5774
fsd =
    4.5563
    6.2172
    7.0933
    7.3489
    7.3941
    7.5823
    8.1445
    8.8536
sttf =
    6.0530
  193.8741
    2.0790
   -2.8580
   -3.5906
   -2.5203
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
nsttf =
           6
ifail =
           0
```