NAG Toolbox for MATLAB

g13ea

1 Purpose

g13ea performs a combined measurement and time update of one iteration of the time-varying Kalman filter using a square root covariance filter.

2 Syntax

```
[s, k, h, ifail] = g13ea(a, b, stq, q, c, r, s, 'n', n, 'm', m, 'l', l,
'tol', tol)
```

3 Description

The Kalman filter arises from the state space model given by:

$$X_{i+1} = A_i X_i + B_i W_i, \quad \mathrm{Var}(W_i) = Q_i$$

$$Y_i = C_i X_i + V_i, \qquad \mathrm{Var}(V_i) = R_i$$

where X_i is the state vector of length n at time i, Y_i is the observation vector of length m at time i, and W_i of length l and V_i of length m are the independent state noise and measurement noise respectively.

The estimate of X_i given observations Y_1 to Y_{i-1} is denoted by $\hat{X}_{i|i-1}$ with state covariance matrix $\mathrm{Var}(\hat{X}_{i|i-1}) = P_{i|i-1} = S_i S_i^{\mathrm{T}}$, while the estimate of X_i given observations Y_1 to Y_i is denoted by $\hat{X}_{i|i}$ with covariance matrix $\mathrm{Var}(\hat{X}_{i|i}) = P_{i|i}$. The update of the estimate, $\hat{X}_{i|i-1}$, from time i to time $(i+1)$, is computed in two stages. First, the measurement-update is given by

$$\hat{X}_{i|i} = \hat{X}_{i|i-1} + K_i\left[Y_i - C_i\hat{X}_{i|i-1}\right] \tag{1}$$

and

$$P_{i|i} = [I - K_i C_i]P_{i|i-1} \tag{2}$$

where $K_i = P_{i|i-1}C_i^{\mathrm{T}}\left[C_i P_{i|i-1} C_i^{\mathrm{T}} + R_i\right]^{-1}$ is the Kalman gain matrix. The second stage is the time-update for X which is given by

$$\hat{X}_{i+1|i} = A_i\hat{X}_{i|i} + D_i U_i \tag{3}$$

and

$$P_{i+1|i} = A_i P_{i|i} A_i^{\mathrm{T}} + B_i Q_i B_i^{\mathrm{T}} \tag{4}$$

where $D_i U_i$ represents any deterministic control used.

The square root covariance filter algorithm provides a stable method for computing the Kalman gain matrix and the state covariance matrix. The algorithm can be summarized as

$$\begin{pmatrix} R_i^{1/2} & C_i S_i & 0 \\ 0 & A_i S_i & B_i Q_i^{1/2} \end{pmatrix} U = \begin{pmatrix} H_i^{1/2} & 0 & 0 \\ G_i & S_{i+1} & 0 \end{pmatrix} \tag{5}$$

where U is an orthogonal transformation triangularizing the left-hand pre-array to produce the right-hand post-array. The relationship between the Kalman gain matrix, K_i, and G_i is given by

$$A_i K_i = G_i\left(H_i^{1/2}\right)^{-1}.$$

g13ea requires the input of the lower triangular Cholesky factors of the noise covariance matrices $R_i^{1/2}$ and, optionally, $Q_i^{1/2}$ and the lower triangular Cholesky factor of the current state covariance matrix, S_i, and returns the product of the matrices A_i and K_i, $A_i K_i$, the Cholesky factor of the updated state covariance matrix S_{i+1} and the matrix $H_i^{1/2}$ used in the computation of the likelihood for the model.

4 References

Vanbegin M, van Dooren P and Verhaegen M H G 1989 Algorithm 675: FORTRAN subroutines for computing the square root covariance filter and square root information filter in dense or Hessenberg forms *ACM Trans. Math. Software* **15** 243–256

Verhaegen M H G and van Dooren P 1986 Numerical aspects of different Kalman filter implementations *IEEE Trans. Auto. Contr.* **AC-31** 907–917

5 Parameters

5.1 Compulsory Input Parameters

1: **a(lds,n) – double array**

lds, the first dimension of the array, must be at least **n**.

The state transition matrix, A_i.

2: **b(lds,l) – double array**

lds, the first dimension of the array, must be at least **n**.

The noise coefficient matrix B_i.

3: **stq – logical scalar**

If **stq** $=$ **true**, the state noise covariance matrix Q_i is assumed to be the identity matrix. Otherwise the lower triangular Cholesky factor, $Q_i^{1/2}$, must be provided in **q**.

4: **q(ldq,$*$) – double array**

The first dimension, **ldq**, of the array **q** must satisfy

if **stq** $=$ **false**, **ldq** $\geq$ **l**;
ldq ≥ 1 otherwise.

The second dimension of the array must be at least **l** if **stq** $=$ **false** and at least 1 if **stq** $=$ **true**

If **stq** $=$ **false** **q** must contain the lower triangular Cholesky factor of the state noise covariance matrix, $Q_i^{1/2}$. Otherwise **q** is not referenced.

5: **c(ldm,n) – double array**

ldm, the first dimension of the array, must be at least **m**.

The measurement coefficient matrix, C_i.

6: **r(ldm,m) – double array**

ldm, the first dimension of the array, must be at least **m**.

The lower triangular Cholesky factor of the measurement noise covariance matrix $R_i^{1/2}$.

7: **s(lds,n) – double array**

lds, the first dimension of the array, must be at least **n**.

The lower triangular Cholesky factor of the state covariance matrix, S_i.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the arrays **a**, **c**, **s**. (An error is raised if these dimensions are not equal.)

n, the size of the state vector.

Constraint: $\mathbf{n} \geq 1$.

2: **m – int32 scalar**

Default: The dimension of the arrays **r**, **k**, **h**. (An error is raised if these dimensions are not equal.)

m, the size of the observation vector.

Constraint: $\mathbf{m} \geq 1$.

3: **l – int32 scalar**

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

l, the dimension of the state noise.

Constraint: $\mathbf{l} \geq 1$.

4: **tol – double scalar**

The tolerance used to test for the singularity of $H_i^{1/2}$. If $0.0 \leq \mathbf{tol} < m^2 \times$ ***machine precision***, then $m^2 \times$ ***machine precision*** is used instead. The inverse of the condition number of $H^{1/2}$ is estimated by a call to f07tg. If this estimate is less than **tol** then $H^{1/2}$ is assumed to be singular.

Suggested value: $\mathbf{tol} = 0.0$.

Default: 0.0

Constraint: $\mathbf{tol} \geq 0.0$.

5.3 Input Parameters Omitted from the MATLAB Interface

lds, ldq, ldm, iwk, wk

5.4 Output Parameters

1: **s(lds,n) – double array**

The lower triangular Cholesky factor of the state covariance matrix, S_{i+1}.

2: **k(lds,m) – double array**

The Kalman gain matrix, K_i, premultiplied by the state transition matrix, A_i, A_iK_i.

3: **h(ldm,m) – double array**

The lower triangular matrix $H_i^{1/2}$.

4: **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, **n** < 1,
or **m** < 1,
or **l** < 1,
or **lds** $<$ **n**,
or **ldm** $<$ **m**,
or **stq** $=$ **true** and **ldq** < 1,
or **stq** $=$ **false** and **ldq** $<$ **l**,
or **tol** < 0.0.

ifail $= 2$

The matrix $H_i^{1/2}$ is singular.

7 Accuracy

The use of the square root algorithm improves the stability of the computations as compared with the direct coding of the Kalman filter. The accuracy will depend on the model.

8 Further Comments

For models with time-invariant A, B and C, g13eb can be used.

If W_i and V_i are independent multivariate Normal variates then the log-likelihood for observations $i = 1, 2, \ldots, t$ is given by

$$l(\theta) = \kappa - \tfrac{1}{2}\sum_{i=1}^{t} ln(\det(H_i)) - \tfrac{1}{2}\sum_{i=1}^{t}\left(Y_i - C_iX_{i|i-1}\right)^{\mathrm{T}} H_i^{-1}\left(Y_i - C_iX_{i|i-1}\right)$$

where κ is a constant.

The Cholesky factors of the covariance matrices can be computed using f07fd.

Note that the model

$$X_{i+1} = A_iX_i + W_i, \quad \mathrm{Var}(W_i) = Q_i$$

$$Y_i = C_iX_i + V_i, \qquad \mathrm{Var}(V_i) = R_i$$

can be specified either with **b** set to the identity matrix and **stq** $=$ **false** and the matrix $Q^{1/2}$ input in **q** or with **stq** $=$ **true** and **b** set to $Q^{1/2}$.

The algorithm requires $\frac{7}{6}n^3 + n^2\left(\frac{5}{2}m + l\right) + n\left(\frac{1}{2}l^2 + m^2\right)$ operations and is backward stable (see Verhaegen and van Dooren 1986).

9 Example

```
a = [0.607, -0.033, 1, 0;
     0, 0.543, 0, 1;
     0, 0, 0, 0;
     0, 0, 0, 0];
b = [1, 0;
     0, 1;
     0.543, 0.125;
     0.134, 0.026];
stq = false;
```

```
q = [1.611831256676703, 0.5600000000000001;
     0.3474309098302363, 2.282387294675146];
c = [1, 0, 0, 0;
     0, 1, 0, 0];
r = [0, 0;
     0, 0];
s = [2.864751298105998, 2.0599, 1.4807, 0.3627;
     0.7190502021456042, 2.729004728246978, 0.9703000000000001, 0.2136;
     0.516868602513227, 0.2193640489823183, 0.7810417797724439, 0.2236;
            0.1266078490791838,  0.04491109863526066,  0.1898854854878202,
0.009846226280729009];
[sOut, k, h, ifail] = g13ea(a, b, stq, q, c, r, s)
```

```
sOut =
   -1.7911    2.0599    1.4807    0.3627
   -0.3955   -2.2825    0.9703    0.2136
   -0.8267   -0.2819    0.4030    0.2236
   -0.2025   -0.0585    0.0986   -0.0000
k =
    0.7672    0.0474
    0.0401    0.5595
         0         0
         0         0
h =
    2.8648         0
    0.7191    2.7290
ifail =
           0
```