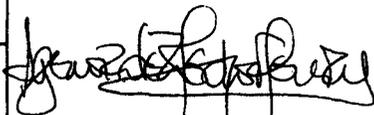


1. Publicação nº <i>INPE-2517-RPI/072</i>	2. Versão	3. Data <i>Ago., 1982</i>	5. Distribuição <input checked="" type="checkbox"/> Interna <input type="checkbox"/> Externa <input type="checkbox"/> Restrita
4. Origem <i>DSE/DDO</i>	Programa <i>ORBAT</i>		
6. Palavras chaves - selecionadas pelo(s) autor(es) <i>FILTRO DE KALMAN</i> <i>MÉTODOS DE FATORIZAÇÃO</i>			
7. C.D.U.: <i>681.5.015.44</i>			
8. Título <i>ANÁLISE NUMÉRICA PRELIMINAR DOS</i> <i>MÉTODOS DE ESTIMAÇÃO SEQUENCIAL</i>		10. Páginas: <i>36</i>	
		11. Última página: <i>29</i>	
		12. Revisada por	
9. Autoria <i>Hélio Koiti Kuga</i>		 Agenor de T. Fleury	
Assinatura responsável 		13. Autorizada por  Nelson de Jesus Parada Diretor	
14. Resumo/Notas <i>A técnica de estimação de estados via filtro de Kalman pode levar a resultados errôneos devido a erros numéricos e instabilidade das equações. Efetua-se, aqui, uma análise preliminar do filtro convencional, do filtro estabilizado, do filtro de raiz quadrada triangular superior e do filtro de fatorização UD, em termos de estabilidade numérica e consumo de tempo de processamento. Apresentam-se rotinas FORTRAN e conclusões preliminares sobre cada um dos algoritmos.</i>			
15. Observações			

ABSTRACT

Numerical errors and instability of the equations can lead to erroneous results in Kalman filtering. In this report the conventional filter, stabilized filter, upper triangular square root filter and UD factorization filter are analysed in terms of numerical stability and processing time costs. FORTRAN routines and preliminary conclusions about each algorithm are also presented.

SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS	v
1. <u>INTRODUÇÃO</u>	1
2. <u>FILTRO CONVENCIONAL DE KALMAN</u>	1
3. <u>COMENTÁRIOS E DISCUSSÕES</u>	5
3.1 - Comentários sobre os métodos de fatorização	9
4. <u>FUNDAMENTOS PRELIMINARES</u>	9
4.1 - Decomposição de Cholesky	10
5. <u>FATORIZAÇÃO DE RAIZ QUADRADA TRIANGULAR SUPERIOR</u>	16
6. <u>FATORIZAÇÃO UD</u>	19
7. <u>ANÁLISE DA EFICIÊNCIA DOS ALGORITMOS PARA ATUALIZAÇÃO DA COVA- RIÂNCIA</u>	22
8. <u>PROPAGAÇÃO DA COVARIÂNCIA</u>	25
9. <u>COMENTÁRIOS E CONCLUSÕES</u>	28
REFERÊNCIAS BIBLIOGRÁFICAS	29

LISTA DE FIGURAS

	<u>Pág.</u>
1 - Rotina para processamento de uma medida escalar usando-se o algoritmo convencional de Kalman.....	4
2 - Rotina para processamento de uma medida escalar, usando-se o algoritmo estabilizado de Kalman	8
3 - Decomposição de Cholesky $P = UU^T$	12
4 - Decomposição livre de Cholesky $P = UDU^T$	13
5 - Montagem de $P = UU^T$	14
6 - Montagem de $P = UDU^T$	15
7 - Rotina para processamento de medida escalar, usando-se o algoritmo de raiz quadrada triangular superior.....	18
8 - Rotina para processamento de medida escalar, usando-se o algoritmo UD	21

1 - INTRODUÇÃO

O constante avanço científico consagrou a técnica de estimação de estado via filtro de Kalman. Sendo, porém, uma solução de aplicação essencialmente numérica, o filtro pode levar a resultados errôneos, pois erros inerentes à máquina (computador) estão sempre presentes e levam, por vezes, o usuário à interpretação falsa dos resultados obtidos.

Neste relatório faz-se uma apresentação preliminar de quatro algoritmos do filtro de Kalman, classificados entre os mais utilizados atualmente. Descreve-se o filtro convencional de Kalman na forma apresentada por Kalman (1960), o filtro estabilizado de Kalman, ou forma de Joseph (Bucy and Joseph, 1968), o filtro de raiz quadrada (Bellantoni and Dodge, 1967; Andrews, 1968), e o algoritmo de fatorização UD (Bierman, 1977). Apresentam-se as listagens dos algoritmos em linguagem FORTRAN padrão, conforme a sequência do trabalho. Origens e ou detalhes particulares dos algoritmos são deixados em segundo plano por se pretender apenas uma descrição e utilização de tais algoritmos.

Apresenta-se, também, uma análise comparativa preliminar dos algoritmos, tendo em vista os aspectos numéricos de estabilidade e consumo de tempo de processamento.

2 - FILTRO CONVENCIONAL DE KALMAN

O filtro de Kalman (1960) é um algoritmo recursivo que permite a obtenção de estimativas do estado e da matriz de covariância dos erros no estado, de forma sequencial. A recursividade se expressa em duas etapas, nas quais se leva em conta primeiramente a dinâmica do sistema, e, em seguida, o processamento das medidas. As duas etapas que compõem um ciclo completo de estimação são denominadas:

- 1) Propagação ("time update") do estado e da matriz de covariância dos erros no estado.

2) Atualização ("measurement update") do estado e da matriz de co variância dos erros no estado.

Assim, seja o sistema dinâmico completo modelado por:

$$x_{k+1} = \phi_{k+1,k} x_k + \Gamma_k w_k, \quad (1)$$

$$z_k = H_k x_k + v_k, \quad (2)$$

onde :

x_k é o vetor do estado no instante k de dimensão n ,

$\phi_{k+1,k}$ é a matriz $n \times n$ de transição do estado no intervalo k a $k+1$,

Γ_k é uma matriz $n \times r$ contínua,

w_k é um vetor de dimensão r do ruído no estado,

z_k é o vetor de medidas de dimensão m ,

H_k é a matriz $m \times n$ das observações, e

v_k é o vetor de dimensão m do erro nas medidas.

As estatísticas dos ruídos são representadas por:

$$E[w_k] = 0 ; E[w_k w_\ell^T] = Q_k \delta_{k\ell}, \quad (3)$$

$$E[v_k] = 0 ; E[v_k v_\ell^T] = R_k \delta_{k\ell}. \quad (4)$$

As equações abaixo definem um ciclo de propagação e atua lização correspondentes ao filtro convencional de Kalman:

. Propagação

- do estado: $\bar{x}_{k+1} = \phi_{k+1,k} \bar{x}_k,$ (5)

- da covariância: $\bar{P}_{k+1} = \phi_{k+1,k} P_k \phi_{k+1,k}^T + \Gamma_k Q_k \Gamma_k^T$. (6)

. Ganho de Kalman

$$K_{k+1} = P_{k+1} H_{k+1}' (H_{k+1} P_{k+1} H_{k+1}' + R_{k+1})^{-1} \cdot \dots$$

. Atualização

- do estado: $\hat{x}_{k+1} = \bar{x}_{k+1} + K_{k+1} (z_{k+1} - H_{k+1} \bar{x}_{k+1})$, (8)

- da covariância: $P_{k+1} = \bar{P}_{k+1} - K_{k+1} H_{k+1} \bar{P}_{k+1}$. (9)

É prática comum processar as medidas sequencialmente, i.é., uma de cada vez, para a atualização do estado e da covariância, Equações 7, 8 e 9. Bierman (1977) mostra que o processamento sequencial não tira a generalidade do procedimento. Na verdade, para poder ser utilizado o processamento sequencial, admite-se que os ruídos de medida são não-correlacionados entre si ou, em outras palavras, a matriz de covariância correspondente, R_{k+1} , é diagonal. Porém tal fato não é restritivo, pois podem-se "branquear" as medidas, tornando R_{k+1} diagonal. Admitir-se-á ao longo do trabalho a utilização do processamento sequencial para a fase de atualização.

A Figura 1 mostra o algoritmo convencional de Kalman para processar uma medida escalar, na forma de sub-rotina FORTRAN.

```

SUBROUTINE KALSTD(NP,N,R,RK,H,X,P,K)
C
C ENTRADAS : NP...PRIMEIRO INDICE DA MATRIZ P NO PROGR.
C *****
C             PRINCIPAL
C             N ...NUMERO DE ELEMENTOS DO ESTADO
C             R ...RESIDUO DA OBSERVACAO
C             RK...VARIANCIA DO RUIDO DA OBSERVACAO
C             H ...VETOR DAS PARCIAIS DA OBSERVACAO
C             X ...VETOR DE ESTADO A PRIORI
C             P ...MATRIZ DE COVARIANCIA DO ERRO A PRIORI
C
C SAIDAS : X ... VETOR DE ESTADO ATUALIZADO
C *****
C             P ... COVARIANCIA ATUALIZADA
C             K ... VETOR REAL REPRESENTA O GANHO DE KALMAN
C
C OBSERVACOES : - ATUALIZACOES DE X E P VIA EQUACOES DO
C *****
C             FILTRO CONVENCIONAL DE KALMAN
C             - ESTADO COM NO MAXIMO 10 ELEMENTOS,CASO
C             CONTRARIO AUMENTAR A DIMENSAO DO
C             VETOR LOCAL AUXILIAR V ATE N
C
C HELIO,DIV. DE DINAMICA ORBITAL,INPE,07-81
C
C     REAL H(N),P(NP,N),X(N),K(N),V(10)
C
C     SIGMA = RK
C
C     DO 10 I=1,N
C       V(I) = 0.
C       DO 5 J=1,N
C 5     V(I) = V(I) + P(I,J) * H(J)
C 10    SIGMA = SIGMA + H(I) * V(I)
C
C     CALCULOU-SE  $V = P*HT$  E  $SIGMA = H*P*HT + RK$ 
C
C     SIGMA = 1./SIGMA
C
C     DO 20 I=1,N
C       K(I) = V(I) * SIGMA
C       X(I) = X(I) + K(I) * R
C       DO 20 J=I,N
C         P(I,J) = P(I,J) - K(I) * V(J)
C 20    P(J,I) = P(I,J)
C
C     CALCULOU-SE "P" PELO FILTRO CONVENCIONAL DE KALMAN
C
C RETURN
C END
```

Fig. 1 - Rotina para processamento de uma medida escalar usando-se o algoritmo convencional de Kalman.

3 - COMENTÁRIOS E DISCUSSÕES

Os vários algoritmos alternativos surgem em decorrência do filtro convencional de Kalman ser simples, porém não livre de erros quando se utiliza um computador digital para realizar os cálculos. Na verdade, a fonte de erros reside na forma algébrica da equação de atualização da covariância, Equação 9. Tal equação é mal condicionada e os valores absolutos dos elementos da matriz de covariância tendem, gradualmente, a se tornar em pequenos. A Equação 9, por sua estrutura algébrica, não garante simetria e alguns elementos da diagonal da covariância podem se tornar negativos. A covariância, então, perde a característica se midefinida positiva. Assim, o algoritmo convencional de atualização é se sensível aos erros de arredondamento/truncamento computacionais e a pre cisão numérica se degrada, de forma que os resultados deixam de ser sig nificativos. A questão é: como utilizar vantajosamente a simplicidade e a versatilidade do filtro de Kalman sem ser vitimado por imprecisão e instabilidade numérica? Algumas alternativas imediatas podem ser prop ostas:

- utilizar dupla precisão, se esta for disponível na máquina;
- utilizar método da força bruta, eliminando o foco de instabilidade; ou seja, impondo simetria e diagonais positivas na matriz de covariância dos erros;
- utilizar técnicas de estabilização (para melhor condicionamento numérico), tal como o filtro de raiz quadrada, que fornece estabilidade e precisão numérica praticamente equivalente ao filtro convencional com dupla precisão.

A consequência natural da imprecisão e instabilidade ca racteriza-se pelo fenômeno da divergência, no qual as estatísticas calculadas se tornam incoerentes com a estimativa real (Fitzgerald, 1971). Basicamente, as causas da divergência são:

- 1) Estatísticas a priori incorretas e parâmetros mal ou não-modelados.

- 2) Presença de não-linearidades (para problemas não-lineares).
- 3) Erros devidos a efeitos de truncamento e ou arredondamento computacionais.

O efeito dos erros numéricos geralmente se manifesta pelo aparecimento de matrizes de covariância dos erros que não retêm a não-negatividade. Portanto, uma análise do ponto de vista numérico sugere que, em princípio, os métodos devem melhorar a precisão e manter a não-negatividade e simetria das covariâncias calculadas. Algumas maneiras usualmente utilizadas para contornar tal efeito são:

- a) Calcular só a parte triangular superior (ou inferior) da covariância e então forçar a simetria, o que também reduz o montante de cálculo.
- b) Calcular a matriz de covariância completa e forçar a simetria, tomando-se a média dos elementos correspondentes fora da diagonal.
- c) Testar e impor os elementos da diagonal da covariância, se negativos.
- d) Substituir o filtro convencional pelo filtro estabilizado, na forma apresentada por Joseph (Bucy and Joseph, 1968), que é mais preciso numericamente. Isto implica substituir a equação de atualização da covariância, Equação 9, por:

$$P_{k+1} = (I - K_{k+1} H_{k+1}) \bar{P}_{k+1} (I - K_{k+1} H_{k+1})^T + K_{k+1} R_{k+1} K_{k+1}^T \quad (10)$$

A justificativa é devida ao fato de que um pequeno erro no cálculo do ganho não se propaga na Equação 10. Considerando-se erros de 1ª ordem no ganho tem-se $\delta P_{k+1} = -\delta K_{k+1} H_{k+1} \bar{P}_{k+1}$, quando se utiliza a Equação 9. Agora, considerando-se a Equação 10 tem-se $\delta P_{k+1} = 0$. A Figura 2 mostra a listagem da rotina FORTRAN correspondente ao filtro estabilizado de Kalman. Naturalmente, o montante de cálculos se eleva.

e) Usar ruídos e covariância dos ruídos maiores, que podem ser im
plementados de vários modos:

- Pode-se limitar os elementos diagonais, com os limites infe
riores escolhidos conforme a confiança do usuário na estimati
va, é uma maneira de adicionar ruído indiretamente.
- Pode-se sobrepesar as medidas mais recentes (Jazwinski, 1970).
- Pode-se adicionar um nível de ruído no estado para eliminar a
divergência. Esses ruídos são normalmente escolhidos por algum
critério a ser definido pelo usuário.

Na verdade, todas estas alternativas são de alguma forma artificiais, com exceção do item d, quando vistas sob o aspecto numéri
co. Entretanto, este trabalho ater-se-á somente à exposição da qualida
de numérica dos algoritmos, sem propor soluções como as descritas nos
itens a, b, c, e.

```
      SUBROUTINE KALSTB(NP,N,R,RK,H,X,P,K)
C
C  ENTRADAS : NP...PRIMEIRO INDICE DA MATRIZ P NO PROGR.
C  *****
C              PRINCIPAL
C              N....NUMERO DE ELEMENTOS DO ESTADO
C              R ...RESÍDUO DA OBSERVAÇÃO
C              RK...VARIANCIA DO RUIDO DA OBSERVAÇÃO
C              H ...VETOR DAS PARCIAIS DA OBSERVAÇÃO
C              X ...VETOR DE ESTADO A PRIORI
C              P ...MATRIZ DE COVARIANCIA DO ERRO A PRIORI
C
C  SAIDAS : X ... VETOR DE ESTADO ATUALIZADO
C  *****
C              P ... COVARIANCIA ATUALIZADA
C              K ... VETOR REAL REPRESENTA O GANHO DE KALMAN
C
C  OBSERVAÇÕES : - ATUALIZAÇÕES DE X E P VIA EQUAÇÕES DO
C  *****
C                  FILTRO ESTABILIZADO DE KALMAN
C                  - ESTADO COM NO MÁXIMO 10 ELEMENTOS;CASO
C                  CONTRÁRIO,AUMENTAR A DIMENSÃO DO
C                  VETOR LOCAL AUXILIAR V PARA N
C
C  HELIO,DIV. DE DINAMICA ORBITAL,INPE,07-81
C
C      REAL H(N),P(NP,N),X(N),K(N),V(10)
C
C      SIGMA = RK
C      DO 10 I=1,N
C      V(I) = 0.
C      DO 5 J=1,N
C  5  V(I) = V(I) + P(I,J) * H(J)
C 10  SIGMA = SIGMA + H(I) * V(I)
C
C      CALCULOU-SE  $V = P*HT$  E  $SIGMA = H*P*HT + RK$ 
C
C      SIGMA = 1./SIGMA
C      DO 20 I=1,N
C      K(I) = V(I) * SIGMA
C      X(I) = X(I) + K(I) * R
C      DO 20 J=I,N
C      P(I,J) = P(I,J) - K(I) * V(J)
C 20  P(J,I) = P(I,J)
C      DO 30 I=1,N
C      V(I) = 0.
C      DO 30 J=1,N
C 30  V(I) = V(I) + P(I,J) * H(J)
C      DO 40 J=1,N
C      DO 40 I=1,J
C      S = 0.5 * ( 2.*P(I,J) - V(I)*K(J) - V(J)*K(I) )
C      P(I,J) = S + RK * K(I) * K(J)
C 40  P(J,I) = P(I,J)
C      RETURN
C      END
```

Fig. 2 - Rotina para processamento de uma medida escalar, usando-se o algoritmo estabilizado de Kalman.

3.1 - COMENTÁRIOS SOBRE OS MÉTODOS DE FATORIZAÇÃO

Os métodos de fatorização são essencialmente baseados na fatorização da matriz de covariância dos erros. Tais métodos, que envolvem matrizes de raiz quadrada, têm propriedades numéricas muito superiores aos métodos normais. Potter introduziu a fatorização de raiz quadrada ($P = SS^T$, onde S é a matriz raiz quadrada de P), o que permite atualizar a raiz quadrada S para uma medida escalar, sem necessidade de aparecer a matriz de covariância explicitamente. O uso de raiz quadrada preserva simetria e a estabilidade numérica, isto foi comprovado quando utilizado intensivamente na missão Apolo.

Kaminski (1971) em sua tese afirma, por meio de exaustivos testes, que os algoritmos de raiz quadrada podem usar meia palavra (de comprimento em computador) para chegar aos resultados dos filtros convencionais (sem raiz quadrada).

Posteriormente, Agee and Turner (Bierman, 1977) propuseram a fatorização UD ($P = UDU^T$, onde U é a matriz triangular superior unitária e D a matriz diagonal), que é uma mecanização bastante estável de decomposição da covariância. Bierman and Thornton (1977) fizeram estudos comparativos sobre os vários filtros em relação aos de raiz quadrada, com particular ênfase para as qualidades do algoritmo UD.

O fato é que com o advento dos métodos de fatorização, rapidamente assimilados e aceitos, eles ganharam importância e estão sendo extensivamente empregados em várias áreas, tais como navegação inercial, determinação de trajetória, órbita e atitude, etc.

4 - FUNDAMENTOS PRELIMINARES

A fatorização da matriz de covariância é feita via decomposição de Cholesky, obtendo-se matrizes triangulares superiores ou inferiores. Os algoritmos de decomposição e respectivos programas FORTRAN são descritos sumariamente.

4.1 - DECOMPOSIÇÃO DE CHOLESKY

Se $P = SS^T$ com S matriz quadrada, então S é uma raiz quadrada de P . Deve-se lembrar que S não é única, pois se a matriz A é ortogonal, i.e., $AA^T = I$, então SA também é uma raiz quadrada de P . O algoritmo para decomposição triangular inferior é dado abaixo.

Se $P > 0$, então existe uma fatorização triangular inferior $P = LL^T$, onde L é uma matriz triangular inferior, e a decomposição com elementos diagonais positivos é dado pelo algoritmo abaixo.

Para $j = 1, \dots, n-1$

$$L(j,j) = P(j,j)^{1/2}$$

$$L(k,j) = P(k,j)/L(j,j), \quad k = j + 1, \dots, n$$

$$P(i,k) = P(i,k) - L(i,j) L(k,j) \begin{cases} k = j + 1, \dots, n \\ i = k, \dots, n \end{cases}$$

e finalmente $L(n,n) = P(n,n)^{1/2}$.

É interessante notar que a decomposição envolve n raízes quadradas escalares, que aparecem na diagonal e dividem as colunas. Tal fato sugere fatorar L e escrever $P = \bar{L}D\bar{L}^T$, com \bar{L} triangular inferior unitária (diagonal unitária) e D diagonal.

Se $P > 0$, então existe $P = \bar{L}D\bar{L}^T$, com D matriz diagonal de elementos d_1, \dots, d_n , dados pelo algoritmo abaixo.

Para $j = 1, \dots, n-1$,

$$d_j = P(j,j); \quad \bar{L}(j,j) = 1,$$

$$P(i,k) = P(i,k) - \bar{L}(i,j) P(k,j) \begin{cases} k = j + 1, \dots, n \\ i = k, \dots, n \end{cases},$$

$$\bar{L}(k,j) = P(k,j)/d_j, \quad k = j + 1, \dots, n.$$

e finalmente $d_n = P(n,n)$. Portanto, o algoritmo de decomposição $P = \bar{L}D\bar{L}^T$ não exige raiz quadrada escalar para seu cálculo, o que implica menor gasto de tempo computacional. Para a decomposição triangular superior; i.é., $P = UU^T$ ou $P = UDU^T$, onde U é triangular superior unitária, o algoritmo é semelhante. As Figuras 3 e 4 mostram as rotinas FORTRAN que fazem a decomposição $P = UU^T$ e $P = UDU^T$, respectivamente. As Figuras 5 e 6 mostram as rotinas FORTRAN que fazem a montagem $P = UU^T$ e $P = UDU^T$, respectivamente. As rotinas são auto-explicativas quanto à utilização.

```
      SUBROUTINE CHLSK1(ND,N,P,U)
C
C   DECOMPOSICAO DE CHOLESKY : P = U U'
C
C   ENTRADAS : ND ... PRIMEIRO INDICE DAS MATRIZES P E U
C               N ... ELEMENTOS DE TRABALHO DAS MATRIZES P E U
C               P ... MATRIZ SIMETRICA NXN
C
C   SAIDA : U ... MATRIZ NXN TRIANGULAR SUPERIOR
C
C   HELIO - DIVISAO DE DINAMICA ORBITAL - 09/10/81
C
C   DIMENSION P(ND,N),U(ND,N)
C
      J = N+1
10  J = J-1
      UJJ = SQRT(P(J,J))
      U(J,J) = UJJ
      ALFA = 1./UJJ
      DO 5 K=1,J-1
      BETA = ALFA * P(K,J)
      U(K,J) = BETA
      DO 5 I=1,K
5  P(I,K) = P(I,K) - BETA*U(I,J)
      IF(J.GE.3) GO TO 10
      U(1,1) = SQRT(P(1,1))
      RETURN
      END
```

Fig. 3 - Decomposição de Cholesky $P = UU^T$.

```
      SUBROUTINE CHLSK2(ND,N,P,U,D)
C
C  DECOMPOSICAO LIVRE DE CHOLESKY P=U*D*U'
C
C  ENTRADAS : ND ... PRIMEIRO INDICE DAS MATRIZES U E P
C              N ... ELEMENTOS DE TRABALHO DAS MATRIZES U E P
C              P ... MATRIZ NXN SIMETRICA
C
C  SAIDA : U ... MATRIZ TRIANGULAR SUPERIOR NXN
C          D ... VETOR NX1 REPRESENTANDO A MATRIZ D NXN
C              DIAGONAL
C
C  HELIO - DIVISAO DE DINAMICA ORBITAL - 09/10/81
C
C      DIMENSION P(ND,N),U(ND,N),D(N)
C
      J = N + 1
10  J = J - 1
      DJ = P(J,J)
      D(J) = DJ
      ALFA = 1./DJ
      DO 5 K=1,J-1
      BETA = P(K,J)
      U(K,J) = ALFA * BETA
      DO 5 I=1,K
5   P(I,K) = P(I,K) - BETA*U(I,J)
      IF(J.GE.3) GO TO 10
      D(1) = P(1,1)
      RETURN
      END
```

Fig. 4 - Decomposição livre de Cholesky $P = UDU^T$.

```
      SUBROUTINE CHLSK3(ND,N,U,P)
C
C  COMPOSICAO P = U * U'
C
C  ENTRADAS : ND ... PRIMEIRO INDICE DAS MATRIZES U E P
C              N  ... ELEMENTOS DE TRABALHO DAS MATRIZES U E P
C              U  ... MATRIZ NXN TRIANGULAR SUPERIOR
C
C  SAIDA : P ... MATRIZ NXN SIMETRICA
C
C  HELIU - DIVISAO DE DINAMICA ORBITAL - 09/10/81
C
      DIMENSION U(ND,N),P(ND,N)
C
      DO 5 I=1,N
      DO 5 J=1,N
      SOMA = 0.
      DO 10 K=J,N
10  SOMA = SOMA + U(I,K) * U(J,K)
      P(I,J) = SOMA
      5  P(J,I) = SOMA
      RETURN
      END
```

Fig. 5 - Montagem de $P = UU^T$.

```
      SUBROUTINE CHLSK4(ND,N,U,D,P)
C
C   COMPOSICAO P = U D U'
C
C   ENTRADAS : ND ... PRIMEIRO INDICE DAS MATRIZES U E P
C               C ... ELEMENTOS DE TRABALHO DAS MATRIZES U E P
C               U ... MATRIZ NXN TRIANGULAR SUPERIOR
C               D ... VETOR NX1 REPRESENTANDO UMA MATRIZ D NXN
C                   DIAGONAL
C
C   SAIDA : P ... MATRIZ NXN SIMETRICA
C
C   HELIO - DIVISAO DE DINAMICA ORBITAL - 09/10/81
C
C   DIMENSION U(ND,N),P(ND,N),D(N)
C
      DN = D(N)
      DO 5 I=1,N-1
        U(I,I) = 1.
        DO 10 J=I,N-1
          SOMA = U(I,J) * D(J)
          DO 15 K=J+1,N
            15 SOMA = SOMA + U(I,K)*U(J,K)*D(K)
          P(I,J) = SOMA
        10 P(J,I) = SOMA
        CIN = U(I,N) * DN
        P(I,N) = CIN
      5 P(N,I) = CIN
      P(N,N) = DN
      RETURN
      END
```

Fig. 6 - Montagem de $P = UDU^T$.

5 - FATORIZAÇÃO DE RAIZ QUADRADA TRIANGULAR SUPERIOR

O primeiro método que utilizou a fatorização de raiz quadrada foi proposto por Potter. Potter mecanizou o procedimento de maneira a atualizar a raiz quadrada, sem a necessidade da covariância aparecer explicitamente. O maior inconveniente, apesar das suas propriedades numéricas, é a elevada quantidade de cálculos em relação ao filtro convencional, devido ao fato da raiz quadrada S ser uma matriz cheia. Atualmente, prefere-se utilizar uma raiz quadrada triangular superior, que, como consequência, diminui o tempo de processamento e a memória de máquina requerida, e mantém as qualidades numéricas. Carlson (1973) foi quem primeiro propôs que S fosse uma matriz triangular superior; posteriormente, Bierman (1977) melhorou o algoritmo (em termos computacionais) que se descreve em seguida.

Dados \bar{x} e \bar{P} a priori, $z = Hx + v$ medida escalar e $E[v^2] = r$, com $\bar{P} = \bar{S}\bar{S}^T$ e $P = SS^T$ a covariância atualizada devido à medida z , obtêm-se S triangular superior atualizada e o ganho de Kalman K por:

$$f = \bar{S}^T H, \quad f^T = (f_1 : \dots : f_n),$$

$$\alpha_0 = r, \quad K_2^T = (\bar{S}_{11} f_1 : \overbrace{0 : \dots : 0}^{n-1}).$$

Para $j = 1, \dots, n$

$$\alpha_j = \alpha_{j-1} + f_j^2,$$

$$\beta_j = (\alpha_{j-1} / \alpha_j)^{1/2}, \quad \gamma_j = f_j / (\beta_j \alpha_j),$$

$$S_{jj} = \bar{S}_{jj} \beta_j,$$

$$S_{ij} = \bar{S}_{ij} \beta_j - \gamma_j K_j(i), \quad i = 1, \dots, j-1 \text{ para } j \neq 1,$$

$$K_{j+1}(i) = K_j(i) + f_j \bar{S}_{ij}, \quad i=1, \dots, j$$

e, finalmente, $K = K_{n+1}/\alpha_n$ é o ganho de Kalman. Obtém-se a estimativa atualizada pela relação $\hat{x} = \bar{x} + K(z - H\bar{x})$. A Figura 7 mostra a rotina FORTRAN do algoritmo de raiz quadrada triangular superior.

```
      SUBROUTINE SORTUP(ND,N,RES,R,H,X,S,B)
C
C  ENTRADAS : ND...PRIMEIRO INDICE DA MATRIZ S NO PROG.
C  *****
C              PRINCIPAL
C              N...NUMERO DE ELEMENTOS DO ESTADO
C              RES..RESIDUO DA OBSERVACAO
C              R...VARIANCIA DO RUIDO NA OBSERVACAO
C              H...VETOR DE PARCIAIS DA OBSERVACAO
C              X...VETOR DE ESTADO A PRIORI
C              S...RAIZ QUADRADA TRIANG. SUP. DA MATRIZ DE
C              COV. A PRIORI
C
C  SAIDAS : X...VETOR DE ESTADO ATUALIZADO
C  *****
C              S...RAIZ QUADRADA TRIANG. SUP. DA MATRIZ DE
C              COV. ATUALIZADA
C              B...VETOR AUXILIAR DE DIMENSAO N, GANHO DE
C              KALMAN PONDERADO
C
C  HELIO, DIV. DE DINAMICA ORBITAL, INPE, 07-81
C
      DIMENSION S(ND,N),X(N),H(N),B(N)
      J = N+1
10  J = J-1
      SIGMA = 0.
      DO 5 K=1,J
      5  SIGMA = SIGMA + S(K,J)*H(K)
      
      H(K) = SIGMA
      
      IF(J.GE.2) GO TO 10
C
      ALFA = R + H(1)**2
      B(1) = S(1,1) * H(1)
      S(1,1) = S(1,1) * SQRT(R/ALFA)
C
      DO 20 J=2,N
      B(J) = S(J,J) * H(J)
      GAMA = ALFA
      ALFA = ALFA + H(J)**2
      BETA = SQRT(GAMA/ALFA)
      GAMA = H(J) / (ALFA*BETA)
      S(J,J) = S(J,J) * BETA
      DO 20 K=1,J-1
      ESSE = S(K,J)
      S(K,J) = ESSE*BETA - GAMA*B(K)
20  B(K) = B(K) + H(J)*ESSE
      RES = RES / ALFA
      DO 30 J=1,N
30  X(J) = X(J) + B(J)*RES
      RETURN
      END
```

Fig. 7 - Rotina para processamento de medida escalar, usando-se o algoritmo de raiz quadrada triangular superior.

6 - FATORIZAÇÃO UD

A essência desta técnica consiste na fatorização da matriz de covariância P , de forma que:

$$P = UDU^T, \quad P^{1/2} = UD^{1/2} = U \text{diag}\{\sqrt{d_1}, \dots, \sqrt{d_n}\},$$

onde U é a matriz triangular superior unitária (diagonal unitária) e D , a diagonal de elementos d_1, \dots, d_n .

Os comentários seguintes caracterizam o algoritmo UD:

- É um algoritmo do tipo raiz quadrada, pois $UD^{1/2}$ é uma raiz quadrada da covariância.
- Tem características de precisão dos algoritmos de raiz quadrada.
- Não envolve cálculo de raiz quadrada escalar.
- Qualifica-se para o uso, em tempo real, que envolva mini ou micro computadores de capacidade limitada de operações.
- É quase tão eficiente, em rapidez e quantidade de cálculos, quanto o (instável) filtro de Kalman convencional.

Em particular, em relação ao algoritmo de Potter, a fatorização UD:

- Diminui a quantidade de cálculos computacionais.
- Requer menos memória de computador, pois U é uma matriz triangular, com D podendo ser armazenado na sua diagonal, enquanto no algoritmo de Potter, $P = SS^T$ com S matriz cheia.
- Evita o consumo de tempo no cálculo de raiz quadrada (pois não há este cálculo).

O algoritmo para processamento de uma medida \bar{z} fornecido em seguida:

Dado \bar{x} e \bar{P} a priori, $z = Hx + v$ medida escalar e $E[v^2] = r$, com $\bar{P} = \bar{U}\bar{D}\bar{U}^T$ e $P = UDU^T$ a covariância atualizada devido à medida z , obtêm-se U triangular superior unitário e D diagonal atualizados, e o ganho de Kalman K por:

$$f = \bar{U}^T H, \quad f^T = (f_1 : \dots : f_n),$$

$$V = \bar{D}f \quad \text{ou} \quad V_i = \bar{d}_i f_i; \quad i=1, \dots, n,$$

$$d_1 = \bar{d}_1 r / \alpha_1 \quad \text{com} \quad \alpha_1 = r + V_1 f_1; \quad K_2^T = (V_1 : \overbrace{0 : \dots : 0}^{n-1}).$$

Para $j = 2, \dots, n$,

$$\alpha_j = \alpha_{j-1} + V_j f_j,$$

$$d_j = \bar{d}_j \alpha_{j-1} / \alpha_j,$$

$$U_j = \bar{U}_j + \lambda_j K_j \quad \text{com} \quad \lambda_j = -f_j / \alpha_{j-1},$$

$$K_{j+1} = K_j + V_j \bar{U}_j,$$

onde $\bar{U} = [\bar{U}_1 : \dots : \bar{U}_n]$ e $U = [U_1 : \dots : U_n]$. O ganho \bar{K} é dado por $K = K_{n+1} / \alpha_n$ e a estimativa atualizada por $\hat{x} = \bar{x} + K(z - H\bar{x})$. A Figura 8 mostra uma rotina FORTRAN para o processamento de uma medida escalar através do algoritmo UD.

```
      SUBROUTINE UDUTUP(ND,N,RES,R,H,X,U,D,B)
C
C  ENTRADAS : ND...PRIMEIRO INDICE DA MATRIZ U NO PROG.
C  *****
C              PRINCIPAL
C
C              N...NUMERO DE ELEMENTOS DO ESTADO
C              RES...RESIDUO DA OBSERVACAO
C              R...VARIANCIA DO RUIDO DA OBSERVACAO
C              H...VETOR DAS PARCIAIS DA OBSERVACAO
C              X...VETOR DE ESTADO A PRIORI
C              U...FATOR U DA MATRIZ DE COV. A PRIORI
C              D...FATOR D DA MATRIZ DE COV. A PRIORI
C
C  SAIDAS : X...VETOR DE ESTADO ATUALIZADO
C  *****
C              J...FATOR U DA MATRIZ DE COV. ATUALIZADA
C              D...FATOR D DA MATRIZ DE COV. ATUALIZADA
C              B...MATRIZ AUXILIAR DIMENSAO N*GANHO DE KALMAN
C              PONDERADO
C
C  HELIO, DIV. DE DINAMICA ORBITAL, INPE, 07-81
C
      DIMENSION U(ND,N),D(N),X(N),H(N),B(N)
      J = N+1
10  J = J-1
      HJ = H(J)
      DO 5 K=1,J-1
      5  HJ = HJ + U(K,J)*H(K)
      H(J) = HJ
      B(J) = D(J)+HJ
      IF(J.GE.3) GO TO 10
      B(1) = D(1)+H(1)
C
      ALFA = R + B(1)*H(1)
      GAMA = 1./ALFA
      D(1) = R * GAMA * D(1)
C
      DO 20 J=2,N
      BETA = ALFA
      ALFA = ALFA + B(J)*H(J)
      ALBDA = - H(J) * GAMA
      GAMA = 1./ALFA
      D(J) = BETA * GAMA * D(J)
      DO 20 I=1,J-1
      BETA = U(I,J)
      U(1,J) = BETA + B(I)*ALBDA
20  B(I) = B(I) + B(J) * BETA
      RES = RES * GAMA
      DO 30 J=1,N
30  X(J) = X(J) + B(J)*RES
      RETURN
      END
```

Fig. 8 - Rotina para processamento de medida escalar, usando-se o algoritmo UD.

7 - ANÁLISE DA EFICIÊNCIA DOS ALGORITMOS PARA ATUALIZAÇÃO DA COVARIÂNCIA

A Tabela 1 mostra o número de operações requerido, por cada algoritmo, para atualizar a covariância e a estimativa (Bierman, 1977). A variável n representa a dimensão do estado, e m representa a quantidade de medidas a serem processadas.

TABELA 1

NÚMERO DE OPERAÇÕES PARA ATUALIZAÇÃO DA COVARIÂNCIA E ESTADO

algoritmo	nº adições	nº multiplicações	nº divisões	nº raiz quadrada
convencional	$(1,5n^2+3,5n)m$	$(1,5n^2+4,5n)m$	m	-
UD	$0,5n^2-0,5n+(1,5n^2+1,5n)m$	$n^2-n+(1,5n^2+5,5n)m$	nm	-
raiz quad. triang. sup.	$0,5n^2+0,5n+(1,5n^2+3,5n)m$	$0,5n^2+0,5n+(2n^2+5n)m$	$2nm$	nm
estabilizado	$(4,5n^2+5,5n)m$	$(4n^2+7,5n)m$	m	-

À primeira vista, é difícil uma análise preliminar conclusiva. Assim, propõe-se um critério que consiste em comparar os tempos de cálculo de cada operação algébrica em relação à operação mais simples, ou seja, a de adição. Os seguintes valores foram retirados dos manuais referentes ao computador BURROUGHS 6700:

$$\tau_+ = 6,3 \mu\text{seg} ,$$

$$\tau_x = 7,0 \mu\text{seg} ,$$

$$\tau_{\div} = 9,65 \mu\text{seg} ,$$

$$\tau_{\sqrt{\quad}} = 114,3 \mu\text{seg} ,$$

onde τ representa o tempo gasto para realizar a operação. Portanto:

$$\tau_+ = 1 \tau_+ ,$$

$$\tau_X = 1,11 \tau_+ ,$$

$$\tau_{\dot{+}} = 1,53 \tau_+ ,$$

$$\tau_{\sqrt{+}} = 18,14 \tau_+ .$$

Em consequência obtêm-se a Tabela 2, que compara os tempos gastos por cada algoritmo, ponderados em relação a τ_+ e relativos ao algoritmo convencional.

TABELA 2

COMPARAÇÃO DE OPERAÇÕES PONDERADAS POR τ_+ EM RELAÇÃO AO ALGORITMO CONVENCIONAL

algoritmo	Expressão relativa a τ_+	n = 10	n = 15	n = 20	n = 30	n = 50
convencional	$(3,17n^2 + 8,5n + 1,53)m$	403,53 m	842,28 m	1.439,53 m	3.109,53 m	8.351,53 m
UD	$1,61n^2 - 1,61n +$ $(3,17n^2 + 9,14n)m$	1,01 + 0,36/m	1,01 + 0,40/m	1,01 + 0,42/m	1,01 + 0,45/m	1,00 + 0,47/m
raiz quad. triang. sup.	$1,06n^2 + 1,06n +$ $(3,72n^2 + 30,26n)m$	1,67 + 0,29/m	1,53 + 0,30/m	1,45 + 0,31/m	1,45 + 0,32/m	1,29 + 0,32/m
estabilizado	$(8,94n^2 + 13,83n + 1,53)m$	2,56	2,64	2,68	2,72	2,76

A análise da Tabela 2 mostra que:

- Quando $n \rightarrow \infty$, aumenta a vantagem do algoritmo UD em relação aos outros algoritmos alternativos.
- Quando $m = 1$, o algoritmo UD consome ~40% a mais de tempo de cálculo em relação ao filtro padrão; os outros dois algoritmos (raiz quadrada e estabilizado) consomem muito mais.
- Quando $m > 4$, a fração em m no algoritmo UD torna-se pequena e sua desvantagem em relação ao convencional é de ~10%.
- Na maioria das situações o tempo gasto, segundo o critério adotado, segue a seguinte hierarquia crescente: convencional, UD, raiz quadrada triangular superior e estabilizado.

8 - PROPAGAÇÃO DA COVARIÂNCIA

Em vista dos fatos expostos, visualizam-se novas maneiras de implementar a fase de propagação do filtro de Kalman, Equações 5 e 6. Anteriormente, deu-se enfoque especial à atualização da covariância, visto que esta atualização representa o local de provável falha numérica do filtro. Com a utilização de métodos de raiz quadrada, a fase de propagação deve ser revista para incorporar essa nova formulação.

A propagação via *raiz quadrada triangular superior* pode ser feita de duas formas:

1º) Aplicação direta da Equação 6 de propagação da covariância. Para tanto, são necessárias a composição e a decomposição da matriz \bar{P} . Assim, dada a matriz de raiz quadrada atualizada S :

- calcula-se $P = SS^T$, que corresponde à covariância atualizada;

- aplica-se a equação normal de propagação, i.e.,

$$\bar{P} = \phi P \phi^T + \Gamma Q \Gamma^T ;$$

- aplica-se a decomposição de Cholesky para \bar{P} , i.é., \bar{S} é tal que

$$\bar{P} = \bar{S}\bar{S}^T,$$

e, então, \bar{S} será a raiz quadrada propagada.

2º) Aplicação da transformação de Householder. Dado $W = \phi S$, e $A = (\Gamma Q \Gamma^T)^{1/2}$, ou seja, $\Gamma Q \Gamma^T = AA^T$:

- constrói-se $\begin{bmatrix} W & : & A \\ \text{nxn} & & \text{nxn} \end{bmatrix}$;

- assim, $\bar{P} = [W:A] [W:A]^T$ é a covariância propagada;

- por meio da transformação de Householder acha-se a matriz B ortogonal de modo que $\bar{P} = [W:A] B ([W:A] B)^T$, que satisfaz $[\bar{S}:0] = [W:A] B$ com \bar{S} triangular superior. Então, \bar{S} será a raiz quadrada propagada (Kaminski, 1971).

Carlson (1973) propõe explorar as formas particulares da matriz de transição (se houver) para diminuir o montante de cálculos. Porém, como não se consegue uma formulação geral, a regra aplicada obedece a um dos dois esquemas anteriormente propostos.

Para o *algoritmo UD*, há também duas formas equivalentes:

1º) Aplicação direta da propagação da covariância (Equação 6). Das as matrizes fatores U e D atualizadas:

- calcula-se $P = UDU^T$, que é a covariância atualizada;

- aplica-se a equação de propagação, i.é.,

$$\bar{P} = \phi P \phi^T + \Gamma Q \Gamma^T;$$

- aplica-se a decomposição livre de Cholesky para obter os fatores propagados \bar{U} e \bar{D} , tal que $\bar{P} = \bar{U}\bar{D}\bar{U}^T$.

2º) Aplicação da ortogonalização ponderada de Gram-Schmidt. Dado U e D fatores atualizados e Q matriz de ruído $r \times r$,

- constrói-se $W = \begin{pmatrix} \phi U : \Gamma \\ \end{pmatrix}$ e D^* que é a matriz diagonal com elemen
 $n \times n$ $n \times r$
tos de D e Q , i.é., $D^* = \text{diag}(D, Q)$;

- assim, $\bar{P} = W D^* W^T$ corresponde à covariância propagada;

- aplica-se a ortogonalização ponderada de Gram-Schmidt (Thornton and Bierman, 1977) a $W D^* W^T$ para obter \bar{U} e \bar{D} : tal que,

$$\bar{U}\bar{D}\bar{U}^T = W D^* W^T.$$

Os fatores \bar{U} e \bar{D} assim obtidos correspondem à propagação da covariância.

Um outro tipo de abordagem consiste em integrar diretamente a equação de Riccati, sem ser necessária a obtenção explícita da matriz de covariância, quando o sistema for contínuo. A equação de Riccati é dada por:

$$\dot{\bar{P}} = F\bar{P}^T + \bar{P}F^T + GQG^T,$$

onde o sistema dinâmico é representado por

$$\dot{x} = Fx + Gw;$$

$$E[w(t)] = 0 ; E[w(t)w^T(\tau)] = Q(t) \delta(t-\tau) .$$

No caso do algoritmo de raiz quadrada triangular superior, Tapley and Choe (1976) propõem integrar a equação de Riccati em termos de \bar{S} . Para o algoritmo UD Tapley and Peters (1980) propõem a integração em termos de \bar{U} e \bar{D} .

De qualquer forma, o aparecimento dos vários algoritmos para propagação da covariância em termos da raiz quadrada não permite uma conclusão adequada sobre a melhor política a ser adotada. Entretanto, atualmente, a ortogonalização de Gram-Schmidt para propagação da covariância (Thornton and Bierman, 1977) tende a ser a técnica mais utilizada, não só por facilidade de implementação como também por sua precisão numérica. Além disso, não estando restrita a sistemas discretos, ela pode ser também aplicada a sistemas de dinâmica contínua.

9 - COMENTÁRIOS E CONCLUSÕES

Os algoritmos de raiz quadrada e fatorização UD mostram-se bastante superiores numericamente em relação ao filtro de Kalman convencional e à forma de Joseph, embora um pouco mais lentos que o convencional, conforme a análise efetuada. Em particular, o filtro UD mostra-se facilmente aplicável a situações onde seja preferível trabalhar principalmente com as quatro operações fundamentais (adição, subtração, multiplicação, divisão), devido às restrições de tempo de processamento ou à inexistência de operadores aritméticos mais complexos (raiz quadrada, logaritmos, exponenciais etc) na máquina utilizada. A análise preliminar dos métodos de propagação da covariância em termos de tais algoritmos não permite uma escolha conclusiva; acredita-se que ela deva ser especificada para cada problema em estudo. Todavia, deve-se lembrar que todas as técnicas pressupõem um ruído no estado que, de alguma forma, já é conhecido ou determinado a priori. Num filtro adaptativo como o de Jazwinski (1970), onde o ruído no estado é determinado "on-line", o esquema de propagação por raiz quadrada falha, e outra abordagem deve ser empregada. Assim, pode ser preferível aplicar a decomposição de Cholesky para, então, atualizar a covariância, pois deste modo não se perde a generalidade do procedimento e propaga-se a matriz de covariância pelos métodos usuais.

Finalmente, convém lembrar que as medidas são processadas sequencialmente, uma de cada vez, o que evita instabilidades geradas pela inversão matricial, requerida por outros métodos.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANDREWS, A. A square root formulation of the Kalman covariance equations. *AIAA Journal* (6):1165-1166, 1968.
- BELLANTONI, J.F.; DODGE, K.W. A square root formulation of the Kalman-Schmidt filter. *AIAA Journal* (5):1309-1314, 1967.
- BIERMAN, G.J. *Factorization methods for discrete sequential estimation*. Academic, New York, 1977.
- BIERMAN, G.J.; THORNTON, C.L. Numerical comparison of Kalman filter algorithms: orbit determination case study. *Automática* (13):23-35, 1977.
- BUCY, R.S.; JOSEPH, P.D. *Filtering for stochastic process with applications to guidance*. Wiley, New York, 1968.
- CARLSON, N.A. Fast triangular formulation of the square root filter. *AIAA Journal* (11):1259-1265, 1973.
- FITZGERALD, R.J. Divergence of the Kalman filter. *IEEE Transactions on automatic control* (16):736-747, 1971.
- JAZWINSKI, A.H. *Stochastic processes and filtering theory*. Academic, New York, 1970. (Mathematics in Science and Engineering, 64).
- KALMAN, R.E. A new approach to linear filtering and prediction problems. *Transactions of the ASME* (D82):35-45, 1960.
- KAMINSKI, P.G. *Square root filtering and smoothing for discrete process*. Ph.D. Dissertation. Stanford, CA, Stanford University, 1971.
- TAPLEY, B.D.; CHOE, C.Y. An algorithm for propagating the square root covariance matrix in triangular form. *IEEE Transactions on Automatic Control* (21):122-123, 1976.
- TAPLEY, B.D.; PETERS, J.G. Sequential estimation algorithm using a continuous UDU^T covariance factorization. *Journal of Guidance and Control* (3):326-331, 1980.
- THORNTON, C.L.; BIERMAN, G.J. Gram-Schmidt algorithms for covariance propagation. *International Journal of Control* (25):243-260, 1977.