統計数理研究所彙報
第31巻 第1号 (1983)

# 最適制御系設計のための SQUARE ROOT ALGORITHM

統計数理研究所 北 川 源四郎

(1983年4月 受付)

## 1. 序

現実の制御対象は完全には把握しきれないさまざまな不確定要因を含んでいる．制御対象に加わる未知の外乱の影響や制御系の構造自体の時間的変化などがその代表的な例といえる．こうした複雑な現実の制御対象に対しては，理論的なモデルを構成し，それに基づいて最適制御系を設計するという正統的な方法では適用範囲が極めて限定される．このような複雑なシステムの制御のために赤池と中川 [2] は統計的最適制御系設計の実用的な方法を与えた．この方法は，多次元定常時系列の自己回帰表現に基礎をおくもので，制御システムの状態空間表現を導き，二次評価基準のもとでの最適制御系を求めるものである．この方法はセメントキルンの制御で実用化され，その後，船舶の自動操舵 (Ohtsu, Horigome and Kitagawa [8]) および火力発電所の制御 (Nakamura and Akaike [7]) に適用されてその有効性が再確認された．マイクロプロセッサーの普及にともない今後その適用範囲はさらに拡大されるものと考えられる．

プログラムパッケージ TIMSAC には多変量自己回帰モデルの同定から最適制御ゲインの計算までのプログラムが完備されている．最適制御ゲインはプログラム OPTDES により求めることができる．ただし，制御性の悪い系の場合には実際の制御ゲインの計算において四捨五入による誤差の累積に十分注意する必要があるとされている (赤池，中川 [2])．マイコン等で単精度演算を行なうときには特に注意が必要である．

同様の安定性の問題は filtering においても良く知られている．初期分散の大きなシステムにおいて精度の高い観測が得られると桁落ちによって有効桁数が著しく減少したり共分散行列の対称性が失われたりして最悪の場合には発散することが知られている．このような現象を防ぐためにいわゆる square root algorithm が開発されている (Kaminski et. al. [6], Dyer et. al. [4])．これらの algorithm は条件の悪い最小二乗問題を精度よく解くための Householder 変換法 (Golub [5]) に基づくもので計算精度の向上が実現されるのはもちろんであるが，同時に，共分散行列 (あるいはその逆行列) の平方根の更新を行なうので，常に行列の対称性および半正定値性が保障されるという利点がある．

Filtering の問題と最適制御の問題は双対な関係にある．したがって，filtering と同様に最適制御ゲインの計算においても square root algorithm を利用することができる．この小稿の目的は，最適制御系設計のための square root algorithm の FORTRAN サブルーチンを与えることである．Square root algorithm は通常，既存のアルゴリズムの共分散行列を LU 分解することによって導びかれているが，本稿では動的計画法 (Bellman [3]) の手法により直接導く方法を示した．同様の導出法が filter および smoother の構成にも適用できるものと考えられる．

## 2. Square Root Algorithm

次のような制御システムが与えられているものとする.

$$\begin{aligned} x(n) &= F(n)x(n-1)+G(n)u(n)+v(n) \\ y(n) &= H(n)x(n)+w(n) \end{aligned} \tag{1}$$

ここで, $x(n)$ は $k$ 次元状態ベクトル, $u(n)$ は $m$ 次元制御入力, $v(n)$ は $k$ 次元プロセスノイズ, $w(n)$ は $l$ 次元観測ノイズである. $y(n)$ は直接制御系の評価に利用される変数を表わす. $F$, $G$ および $H$ は, それぞれ, $k\times k$, $k\times m$ および $l\times k$ 行列となる. また, $v(n)$ は平均0, 共分散行列 $V(n)$ の正規白色過程とする. このとき, 我々の目的は, 与えられた初期値 $x(0)$ に対し二次形式評価基準

$$I_N=\min_{\{u(n)\}} \mathrm{E}\left[\sum_{n=1}^{N}\{y(n)'A(n)y(n)+u(n)'B(n)u(n)\}\right] \tag{2}$$

を最小とする制御入力 $\{u(1),\cdots,u(N)\}$ を求めることである. ただし, $A(n)$ は $l\times l$ 半正定値行列, $B(n)$ は $m\times m$ 正定値行列, $t$ は転置を表わすものとする.

まず上三角行列 $U(n)$ および $V(n)$ をそれぞれ $A(n)$ と $B(n)$ の Choleski 分解

$$\begin{aligned} A(n) &= U(n)'U(n) \\ B(n) &= V(n)'V(n) \end{aligned}$$

によって定義する. このとき (2) は

$$I_N=\min_{\{u(n)\}} \mathrm{E}\left[\sum_{n=1}^{N}\{\|U(n)y(n)\|_l^2+\|V(n)u(n)\|_m^2\}\right]$$

と表現できる. ただし, $\|\cdot\|_k$ は $k$ 次元ベクトルのユークリッドノルムを表わす. ここで, 動的計画法の手法に従って $f_n(x)$ を

$$f_n(x)=\min_{\substack{\{u(n),\cdots,u(N)\}\\ x(n-1)=x}} \mathrm{E}\left[\sum_{i=n}^{N}\{\|U(i)y(i)\|_l^2+\|V(i)u(i)\|_m^2\}\right]$$

と定義すると最適性の原理によって

$$f_n(x)=\min_{\substack{u(n)\\ x(n-1)=x}} \mathrm{E}\{\|U(n)y(n)\|_l^2+\|V(n)u(n)\|_m^2+f_{n+1}(x(n))\} \tag{3}$$

が成りたつ. 次にこの関係を利用して, 最適制御問題の解を導く.

**[1]** まず $f_n(x)$ が

$$f_n(x)=\min_{\substack{u(n)\\ x(n-1)=x}} \mathrm{E}\{\|R(n)x(n)\|_k^2+\|V(n)u(n)\|_m^2\} \tag{4}$$

という形に表現できるものとする. このとき右辺の $\mathrm{E}\{\ \}$ の式は

$$\begin{aligned} &\mathrm{E}\{\|R(n)x(n)\|_k^2+\|V(n)u(n)\|_m^2\} \\ &\quad=\mathrm{E}\{\|R(n)(F(n)x+G(n)u(n)+v(n))\|_k^2+\|V(n)u(n)\|_m^2\} \\ &\quad=\|R(n)(F(n)x+G(n)u(n))\|_k^2+\|V(n)u(n)\|_m^2+\mathrm{E}\{\|R(n)v(n)\|_k^2\} \end{aligned}$$

$$= \left\| \left[\begin{array}{c|c} V(n) & 0 \\ \hline R(n)G(n) & R(n)F(n) \end{array}\right] \left[\begin{array}{c} u(n) \\ \hline x \end{array}\right] \right\|^2_{m+k} + \mathrm{E}\{\|R(n)v(n)\|_k^2\}$$

となる．以下，簡単のために時刻を表わす $n$, $n-1$ は省略することがある．ここで，$(m+k)\times(m+k)$ 行列で規定される Householder 変換によって

$$Q\left[\begin{array}{c|c} V & 0 \\ \hline RG & RF \end{array}\right] = \left[\begin{array}{c|c} S_{11} & S_{12} \\ \hline 0 & S \end{array}\right] \tag{5}$$

ただし，$S_{11}$ および $S$ はそれぞれ $m\times m$, $k\times k$ 上三角行列，となることから

$$f_n(x) = \min_u \{\|S_{11}u+S_{12}x\|_m^2 + \|Sx\|_k^2 + \mathrm{E}\|Rv\|_k^2\} \tag{6}$$

となる．右辺 { } 内の第2項および第3項は制御入力 $u$ には依存しないことを考慮すると，右辺の最小値は $u(n)$ が

$$S_{11}(n)u(n) + S_{12}(n)x(n-1) = 0$$

を満たすときに達成される．したがって

$$f_n(x) = \|Sx\|_k^2 + \mathrm{E}\|Rv\|_k^2 \tag{7}$$

となり，また最適制御入力は

$$u^*(n) = -S_{11}(n)^{-1}S_{12}(n)x(n-1) \tag{8}$$

で与えられる．((6) の右辺第3項は $x$ にも $u$ にも依存しない定数だから無視してもよい.)

**[2]** $f_n(x)$ が $\|S(n)x\|_k^2$ という形で与えられるものと仮定する．このとき，(3) により

$$f_{n-1}(x) = \min_{\substack{u(n-1) \\ x(n-2)=x}} \mathrm{E}\{\|U(n-1)y(n-1)\|_l^2 + \|V(n-1)u(n-1)\|_m^2 + \|S(n)x(n-1)\|_k^2\}$$

$$= \min \mathrm{E}\{\left\| \left[\begin{array}{c} S(n) \\ \hline U(n-1)\ H(n-1) \end{array}\right] x(n-1) \right\|^2_{k+l} + \|V(n-1)u(n-1)\|_m^2\}$$

となる．ここで

$$Q'\left[\begin{array}{c} S(n) \\ \hline U(n-1)\ H(n-1) \end{array}\right] = \left[\begin{array}{c} R(n-1) \\ \hline 0 \end{array}\right] \tag{9}$$

なる Householder 変換によって三角化を行なうと

$$f_{n-1}(x) = \min_{\substack{u(n-1) \\ x(n-2)=x}} \mathrm{E}\{\|R(n-1)x(n-1)\|_k^2 + \|V(n-1)u(n-1)\|_m^2 + \|U(n-1)w(n-1)\|_l^2\} \tag{10}$$

となる．右辺第3項は，状態ベクトル，制御入力のいずれにも依存しない定数なので無視することができる．

**[3]** $n=N$ において $H(N)'A(N)H(N)$ の Choleski 分解によって上三角行列 $R(N)$ を定義すると，$f_N(x)$ は確かに (4) の形になっている．したがって，最適制御入力 $u(N)^*$ は (8) で求められ，$f_N(x)$ は (7) で与えられる．次に **[2]** によって $f_{N-1}(x)$ も同様に (4) の形をしている事がわかる．したがって．これを繰返して，$f_n(x)$ を達成する入力 $u(n)^*$ が次々と求め

られる．動的計画法によれば，このようにして求められた $\{u(1)^*, \cdots, u(N)^*\}$ が当初の最適制御問題 (2) の解となる．

以上をまとめて次の square root algorithm が得られる．

**[ALGORITHM]**

**初期化**

$$R(N)^tR(N) = (H(N)^tA(N)H(N))$$

**$n = N, \cdots, 1$ に対して**

$$\left[\begin{array}{c:c} V(n) & 0 \\ \hdashline R(n)G(n) & R(n)F(n) \end{array}\right] \xrightarrow{\text{Householder 変換}} \left[\begin{array}{c:c} S_{11}(n) & S_{12}(n) \\ \hdashline 0 & S(n) \end{array}\right]$$

$$\left[\begin{array}{c} S(n) \\ \hdashline U(n-1)\ H(n-1) \end{array}\right] \xrightarrow{\text{Householder 変換}} \left[\begin{array}{c} R(n-1) \\ \hdashline 0 \end{array}\right]$$

$$u(n)^* = -S_{11}(n)S_{12}(n)x(n-1)$$

## 3. 従来の方法との関係

赤池，中川 [2] では次のアルゴリズムが使われている．

**初期化**

$$P(N) = H^tAH$$

**$n = N-1, \cdots, 1$ に対して**

$$T(n) = P(n+1) - P(n+1)G(B+G^tP(n+1)G)^{-1}G^tP(n+1) \tag{11}$$

$$P(n) = F^tT(n)F + H^tAH \tag{12}$$

$$u^* = -(B+G^tP(n+1)G)^{-1}G^tP(n+1)Fx \tag{13}$$

このアルゴリズムと第2節で導いたアルゴリズムの関係を調べてみよう．

まず，$S(n)^tS(n) = T(n)$，$R(n)^tR(n) = P(n)$ と書くことにする．直交変換 $Q$ に対して，$X^tQ^tQX = X^tX$ が成り立つことを考慮すると (5) から

$$S_{11}{}^tS_{11} = V^tV + G^tR^tRG = B + G^tPG \tag{14}$$

$$S_{11}{}^tS_{12} = G^tR^tRF = G^tPF \tag{15}$$

$$S^tS + S_{12}{}^tS_{12} = F^tR^tRF = F^tPF \tag{16}$$

が成り立つ．したがって

$$T(n) = S(n)^tS(n) = F(n)^tP(n)F(n) - S_{12}(n)^tS_{12}(n) \tag{17}$$

となり，(14) および (15) より

$$\begin{aligned} S_{12}(n)^tS_{12}(n) &= F(n)^tP(n)G(n)S_{11}(n)^{-1}S_{11}(n)^{-t}G(n)^tP(n)F(n) \\ &= F^tPG(B+GPG^t)^{-1}G^tPF \end{aligned}$$

が得られる．したがって (17) より

$$T = F^tPF - F^tPG(B+GPG^t)^{-1}G^tPF$$

が成りたつことがわかる．同様にして (9) より

$$S^tS+H^tV^tVH=R^tR$$

が得られ

$$P=T+H^tAH$$

が導かれる．また (8) と (15) より

$$\begin{aligned}u(n)^* &=-S_{11}^{-1}(n)\,S_{12}(n)\,x(n-1)\\ &=-S_{11}^{-1}(n)\,S_{11}^{-t}(n)\,G(n)^tP(n)\,F(n)\,x(n-1)\\ &=-(B(n)+G(n)^tP(n)G(n))^{-1}G(n)^tP(n)\,F(n)\,x(n-1)\end{aligned}$$

が得られる．

こうして，上述の square root algorithm は (11), (12) および (13) で与えられる計算法と理論的には同等であることがわかる．

## 4. 計算プログラム

ここでは，一般の状態空間表現 (1) が与えられたとき最適制御ゲインを求めるサブルーチンを示す．HUSHLD および LTINV はそれぞれ Householder 変換および対称行列の Choleski 分解のためのサブルーチンで TIMSAC-78 (Akaike et. al. [1]) にそのソースリストが与えられている．制御系のモデルが自己回帰表現で与えられている場合には行列 $F$, $G$, $H$ の特殊な構造を利用してより効率のよいプログラムを作ることができる．最後にこのサブルーチンの利用例として簡単な driver program を示した．

SUBROUTINE OPTDES (F, G, H, U, V, R, S, K, L, M, IT, MJ1, MJ2, GAIN)

入力:

| | | |
|---|---|---|
| F: | 倍精度実数アレイ (MJ2, MJ2) | K×K 遷移行列 |
| G: | 倍精度実数アレイ (MJ2, MJ1) | K×M 応答行列 |
| H: | 倍精度実数アレイ (MJ1, MJ2) | L×K 観測行列 |
| U: | 倍精度実数アレイ (MJ1, MJ1) | 被制御変数の重み行列 (L×L) |
| V: | 倍精度実数アレイ (MJ1, MJ1) | 制御入力の重み行列 (M×M) |
| K: | 整数 | 状態空間の次元 |
| L: | 整数 | 被制御変数の次元 |
| M: | 整数 | 制御入力の次元 |
| IT: | 整数 | 評価する区間の長さ |
| MJ1: | 整数 | 行列の大きさ |
| MJ2: | 整数 | 行列の大きさ |

作業領域:

| | | |
|---|---|---|
| R: | 倍精度実数アレイ (MJ2, MJ2) | |
| S: | 倍精度実数アレイ (MJ2, MJ2) | |

出力:

| | | |
|---|---|---|
| GAIN: | 倍精度実数アレイ (MJ1, MJ2) | M×K 最適制御ゲイン行列 |

```
00010       SUBROUTINE  OPTDES( F,G,H,U,V,R,S,K,L,M,IT,MJ1,MJ2,GAIN )
00020 C
00030 C  ...  OPTIMAL CONTROLLER DESIGN  ...
00040 C
00050 C     THIS PROGRAM COMPUTES OPTIMAL CONTROLLER GAIN MATRIX
00060 C     BY A SQUARE ROOT ALGORITHM.  THE BASIC MODEL IS
00070 C
00080 C          X(N) = F(N)X(N-1) + G(N)U(N) + V(N)
00090 C          Y(N) = H(N)X(N)
00100 C
00110 C     THE CRITERION FUNCTION IS
00120 C
00130 C          E( SUM( Y(N)'Q(N)Y(N) + U(N)'R(N)U(N) ) )
00140 C
00150 C     WHERE
00160 C          SUM:      SUMMATION OVER N=1,...,IT
00170 C          Q(N):     L*L WEIGHTING MATRIX
00180 C          R(N):     M*M WEIGHTING MATRIX
00190 C
00200 C
00210 C     THE FOLLOWING SUBROUTINES ARE DIRECTLY CALLED BY THIS SUBROUTINE:
00220 C          HUSHLD
00230 C          LTINV
00240 C          INVERS
00250 C
00260 C     INPUTS:
00270 C        F:     K*K TRANSITION MATRIX
00280 C        G:     K*M RESPONSE MATRIX
00290 C        H:     L*K OBSERVATION MATRIX
00300 C        U:     WEIGHTING MATRIX FOR CONTROLLED VARIABLES (L*L)
00310 C        V:     WEIGHTING MATRIX FOR CONTROL INPUTS (M*M)
00320 C        K:     DIMENSION OF THE STATE VECTOR
00330 C        L:     DIMENSION OF THE CONTROLLED VARIABLE
00340 C        M:     DIMENSION OF THE CONTROL INPUT
00350 C        IT:    TIME SPAN FOR DYNAMIC PROGRAMMING
00360 C        MJ1:   SIZE OF THE ARRAIES
00370 C        MJ2:   SIZE OF THE ARRAIES
00380 C     WORKING SPACE:
00390 C        R:     (M+K)*(M+K) MATRIX
00400 C        S:     (L+K)*K MATRIX
00410 C     OUTPUT:
00420 C        GAIN: STATIONARY OPTIMAL CONTROL GAIN MATRIX (M*K)
00430 C
00440       IMPLICIT REAL*8(A-H,O-Z)
00450       DIMENSION R(MJ2,MJ2), S(MJ2,MJ2), U(MJ1,MJ1), V(MJ1,MJ1)
00460       DIMENSION F(MJ2,MJ2), G(MJ2,MJ1), H(MJ1,MJ2), GAIN(MJ1,MJ2)
00470       DIMENSION  T(10,10), D(100), Q(10,100)
00480 C
00490       WRITE(6,11)  K, M, L, IT
00500       WRITE(6,5)
00510       DO 10 I=1,K
00520    10 WRITE(6,4)  (F(I,J),J=1,K)
00530       WRITE(6,6)
00540       DO 20 I=1,K
00550    20 WRITE(6,4)  (G(I,J),J=1,M)
00560       WRITE(6,7)
00570       DO 30 I=1,L
00580    30 WRITE(6,4)  (H(I,J),J=1,K)
00590       WRITE(6,8)
00600       DO 40 I=1,L
00610    40 WRITE(6,4)  (U(I,J),J=1,L)
00620       WRITE(6,9)
00630       DO 50 I=1,M
00640    50 WRITE(6,4)  (V(I,J),J=1,M)
```

```
00650 C
00660 C  ...  INITIALIZATION  ...
00670 C
00680        CALL  LTINV(U,L,MJ1)
00690        CALL  LTINV(V,M,MJ1)
00700 C
00710        DO 60 I=1,K
00720        DO 60 J=1,K
00730     60 R(I,J) = 0.0D0
00740        DO 80 I=1,L
00750        DO 80 J=1,K
00760        SUM = 0.0D0
00770        DO 70 JJ=1,L
00780     70 SUM = SUM + U(I,JJ)*H(JJ,J)
00790        R(I,J) = SUM
00800     80 Q(I,J) = SUM
00810 C
00820 C  ...  ITERATION  ...
00830 C
00840        DO 200 II=1,IT
00850 C
00860        DO 110 I=1,M
00870        DO 110 J=1,K
00880    110 S(I,M+J) = 0.0D0
00890        DO 120 I=1,M
00900        DO 120 J=I,M
00910    120 S(I,J) = V(I,J)
00920        DO 140 I=1,K
00930        DO 140 J=1,M
00940        SUM = 0.0D0
00950        DO 130 JJ=1,K
00960    130 SUM = SUM + R(I,JJ)*G(JJ,J)
00970    140 S(M+I,J) =SUM
00980        DO 160 I=1,K
00990        DO 160 J=1,K
01000        SUM = 0.0D0
01010        DO 150 JJ=1,K
01020    150 SUM = SUM + R(I,JJ)*F(JJ,J)
01030    160 S(M+I,M+J) = SUM
01040 C
01050        CALL  HUSHLD(S,D,MJ2,M+K,M+K)
01060 C
01070        DO 170 I=1,K
01080        DO 170 J=1,K
01090    170 R(I,J) = S(M+I,M+J)
01100        DO 180 I=1,L
01110        DO 180 J=1,K
01120    180 R(K+I,J) = Q(I,J)
01130 C
01140        CALL  HUSHLD(R,D,MJ2,K+L,K)
01150 C
01160 C 200 CONTINUE
01170 C
01180 C  ...  GAIN MATRIX  ...
01190 C
01200        DO 210 I=1,M
01210        DO 210 J=1,M
01220    210 T(I,J) = S(I,J)
01230        CALL  INVERS(T,M,MJ1)
01240        DO 230 I=1,M
01250        DO 230 J=1,K
01260        SUM = 0.0D0
01270        DO 220 JJ=I,M
01280    220 SUM = SUM + T(JJ,I)*S(JJ,M+J)
01290    230 GAIN(I,J) = -SUM
```

```
01300 C
01310       I = IT - II + 1
01320       WRITE(6,3)  I
01330       DO 240 I=1,M
01340   240 WRITE(6,4)  (GAIN(I,J),J=1,K)
01350   200 CONTINUE
01360 C
01370       STOP
01380     1 FORMAT( 16I5 )
01390     2 FORMAT( 8F10.5 )
01400     3 FORMAT( 1H ,'N  =',I4 )
01410     4 FORMAT( 1H ,10X,10F12.7 )
01420     5 FORMAT( 1H0,'*****  F  *****' )
01430     6 FORMAT( 1H0,'*****  G  *****' )
01440     7 FORMAT( 1H0,'*****  H  *****' )
01450     8 FORMAT( 1H0,'*****  Q  *****' )
01460     9 FORMAT( 1H0,'*****  R  *****' )
01470    11 FORMAT( 1H0,'<<<<<  OPTIMAL CONTROL DESIGN BY A SQUARE ROOT ALGORI
01480      *THM  >>>>>',/,3X,'K =',I2,5X,'M =',I2,5X,'L =',I2,5X,'IT =',I4 )
01490       END
```

```
00010       PROGRAM  MAIN
00020 C
00030 C  ...  THIS IS A DRIVER PROGRAM FOR THE SUBROUTINE OPTDES  ...
00040 C
00050       IMPLICIT REAL*8(A-H,O-Z)
00060       DIMENSION  F(100,100), G(100,10), H(10,100), GAIN(10,100)
00070       DIMENSION  RR(100,100), SS(100,100), Q(10,10), R(1,10)
00080 C
00090       MJ1 = 10
00100       MJ2 = 100
00110 C
00120       READ(5,1)  K,M,L,IT
00130       DO 10 I=1,K
00140    10 READ(5,2)  (F(I,J),J=1,K)
00150       DO 20 I=1,K
00160    20 READ(5,2)  (G(I,J),J=1,M)
00170       DO 30 I=1,L
00180    30 READ(5,2)  (H(I,J),J=1,K)
00190       DO 40 I=1,L
00200    40 READ(5,2)  (Q(I,J),J=1,L)
00210       DO 50 I=1,M
00220    50 READ(5,2)  (R(I,J),J=1,M)
00230 C
00240       CALL  OPTDES( F,G,H,Q,R,RR,SS,K,L,M,IT,MJ1,MJ2,GAIN )
00250 C
00260       STOP
00270     1 FORMAT( 16I5 )
00280     2 FORMAT( 8F10.5 )
00290       E N D
```

## 参 考 文 献

[1] Akaike, H., Kitagawa, G., Arahata, E. and Tada, F. (1979). TIMSAC-78, *Computer Science Monographs*, **11**.

[2] 赤池弘次, 中川東一郎 (1972). ダイナミックシステムの統計的解析と制御, コンピューターサイエンスライブラリー, サイエンス社.

[3] Bellman, R. (1957). *Dynamic Programming*, Princeton University Press, New Jersey.

[4] Dyer, P. and McReynolds, S. (1969). Extension of square-root filtering to include process noise, *J. Optimiz. Theory Appl.*, **3**, 444-459.

[5] Golub, G.H. (1965). Numerical methods for solving linear least squares problems, *Num. Math.*, **7**, 206-216.

[6] Kaminski, P.G., Bryson, A.E. and Schmidt, S.F. (1971). Discrete square root filtering: a survey of current techniques, *IEEE Trans. Automat. Contr.*, **16**, 727-735.

[7] Nakamura, H. and Akaike, H. (1981). Statistical identification for optimal control of supercritical power plants, *Automatica*, **17**, 143-155.

[8] Ohtsu, K., Horigome, M. and Kitagawa, G. (1979). A new ship's auto pilot design through a stochastic model, *Automatica*, **15**, 255-268.

# A Square Root Algorithm for Optimal Controller Design

Genshiro Kitagawa
(The Institute of Statistical Mathematics)

A FORTRAN subroutine is given for the solution of the stochastic optimal control problem:

Find the optimal control inputs $u(1), \cdots, u(N)$ that minimize

$$I_N = \min_{\{u(n)\}} \mathrm{E}\left[\sum_{n=1}^{N} \{y(n)^t A(n) y(n) + u(n)^t B(n) u(n)\}\right]$$

given the state space model

$$x(n) = F(n)x(n-1) + G(n)u(n) + v(n)$$
$$y(n) = H(n)x(n) + w(n)$$

with the initial state vector $x(0)$.

The program is based on a square root algorithm that is derived by using dynamic programming technique. The algorithm is:

*Initialization*

$$R(N)^t R(N) = (H(N)^t A(N) H(N))$$

*For* $n=N-1$ *through* 1

$$\left[\begin{array}{c|c} V(n) & 0 \\ \hline R(n)G(n) & R(n)F(n) \end{array}\right] \xrightarrow{\text{Householder Transf.}} \left[\begin{array}{c|c} S_{11}(n) & S_{12}(n) \\ \hline 0 & S(n) \end{array}\right]$$

$$\left[\begin{array}{c} S(n) \\ \hline U(n-1)H(n-1) \end{array}\right] \xrightarrow{\text{Householder Transf.}} \left[\begin{array}{c} R(n-1) \\ \hline 0 \end{array}\right]$$

$$u(n)^* = -S_{11}(n)S_{12}(n)x(n-1)$$

Here $U(n)$ and $V(n)$ are the Choleski decomposition of $A(n)$ and $B(n)$, respectively. The relation between the presented algorithm and the conventional one is also shown.

Key words: Householder transformation, optimal controller, square root algorithm, state space representation, time series model.