A Spectral Factorization Algorithm

Åström, Karl Johan

1973

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

A Spectral Factorization Algorithm

by

K.J. Åström

Abstract

Let B be a polynomial with zeros inside and outside the unit circ let B^* denote the reciprocial polynomial. The spectral factoriza problem is to find a polynomial C with zeros inside the unit circ such that $BB^* = CC^*$. Some numerical algorithms to solve this prob are discussed.

Contents

1. INTRODUCTION

Let B be a polynomial

$$B(z) = z^n + b_1 z^{n-1} + \ldots + b_n$$

and B^* its reciprocial i.e.

$$B^*(z) = b_n z^n + b_{n-1} z^{n-1} + \ldots + 1$$

The problem is to find a polynomial C with all its zeros inside the unit circle such that

$$C(z)C^*(z) = B(z)B^*(z)$$

2. PRELIMINARIES

Consider the stochastic system

$$z(t+1) = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & & 0 \end{bmatrix} z(t) + \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_{n-1} \\ b_n \end{bmatrix} e(t)$$

$$y(t) = z_1(t) + e(t)$$

where $\{e(t), t=0, \pm 1, \pm 2, \dots\}$ is discrete time white noise with E
$Ee^2(t) = 1$. The output $\{y(t), t=0, \pm 1, \pm 2, \dots\}$ has the spectral dens

$$\phi_y = \frac{1}{2\pi} B(z)B^*(z^{\prime})$$

It follows from optimal filtering theory that the process $\{y(t)\}$ als
can be represented as

$$\hat{z}(t+1) = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \hat{z}(t) + \begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_{n-1} \\ c_n \end{bmatrix} \varepsilon(t)$$

$$y(t) = \hat{z}_1(t) + \varepsilon(t)$$

where

$$c = [\Phi P \theta^T + b][\theta P \theta^T + 1]^{-1}$$

and

The output y of (6) also can be written as

$$y(t) = \varepsilon(t) + c_1\varepsilon(t-1) + \ldots + c_n\varepsilon(t-n) \qquad (9$$

The spectral density of the process is thus

$$\phi_y = \frac{1}{2\pi} C(z)C^*(z)\cdot k \qquad ($$

where

$$K(1+c_1^2+c_2^2+\ldots+c_n^2) = 1 + b_1^2+\ldots+b_n^2 = 1 + p_{11} \qquad ($$

The equation (8) has obviously the solution P=0 which corresponds to c=b. If the polynomial B has all its zero inside the unit circle thi is also the only non-negative solution. However, if B has zeros outs the unit cirlce there are also other solutions. In particular the po tive definite solution corresponds to a stable C.

3. THE ALGORITHM

The algorithm is a staight-forward iteration of the Riccati equatio (8) where the initial condition is chosen in such a way that P is p sitive definite, say P(0) = I (identity matrix).

Since Φ and θ have very special forms several simplifications can b made in the algorithm. Introduce

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & & & \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}, \quad \Phi = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & & 0 \end{bmatrix}.$$

$$\theta = [1 \quad 0 \; . \; . \; . \; 0]$$

Then

$$\theta P \theta^T + 1 = 1 + p_{11}$$

$$\Phi P \theta^T = \begin{bmatrix} p_{21} \\ p_{31} \\ \vdots \\ p_{n1} \\ 0 \end{bmatrix}$$

$$\Phi P \Phi^T = \begin{bmatrix} p_{22} & p_{23} & \cdots & p_{2n} & 0 \\ p_{32} & p_{33} & \cdots & p_{3n} & 0 \\ \vdots & & & & \\ p_{n2} & p_{n3} & \cdots & p_{nn} & 0 \\ 0 & 0 & & 0 & 0 \end{bmatrix}$$

Hence

$$p_{ij}(t+1) = p_{i+1,j+1}(t) + b_i b_j - \frac{(p_{i+1,1}(t)+b_i)(p_{j+1,1}(t)+b_j)}{1+p_{11}}$$

where

$$p_{ij}(t) = 0 \qquad \text{if} \quad i > n \quad \text{or} \quad j > n \qquad \forall\, t$$

and

$$c = \begin{bmatrix} p_{21} + b_1 \\ p_{31} + b_2 \\ \\ p_{n1} + b_{n-1} \\ b_n \end{bmatrix} \cdot \frac{1}{1+p_{11}}$$

It is clear from the construction of the algorithm that its conver
will depend strongly on the properties of the polynomial B. The al
will converge exponentially where the exponent is determined by th
of BB^* within the unit circle. The convergence thus will be very
if B has zeros close to the unit circle.

4. INTERPRETATION

It is of interest to investigate the algorithm in some more detail. First observe that (14) can be written as

$$P(t+1) = \Phi P(t)\Phi + bb^T - (1+p_{11}(t))C(t)C^T(t)$$

Since $\Phi^n = 0$, we get

$$P(t+n) = \sum_{i=0}^{n-1} \Phi^i[bb^T - (1+p_{11}(t+n-i-1))C(t+n-i-1)C^T(t+n-i-1)](\Phi^T)^i$$

Furthermore

$$\Phi b = \begin{bmatrix} b_2 \\ b_3 \\ \vdots \\ b_{n-} \\ b_n \\ 0 \end{bmatrix},\ \Phi^2 b = \begin{bmatrix} b_3 \\ b_4 \\ \vdots \\ b_n \\ 0 \\ 0 \end{bmatrix}, \ldots,\ \Phi^{n-1} b = \begin{bmatrix} b_n \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Similar equations also hold for $\Phi^i c$. Hence

$$(\Phi^i c)_k = \begin{cases} c_{i+k} & \text{if} \quad i+k \le n \\ 0 & \text{if} \quad i+k > n \end{cases}$$

Writing the matrix equation (17) componentwise we get

$$p_{11}(t+n) = \sum_{i=0}^{n-1} b_{i+1}^2 - \sum_{i=0}^{n-1} [1+p_{11}(t+n-i-1)]c_{i+1}^2(t+n-i-1) =$$

$$p_{21}(t+n) = \sum_{i=0}^{n-2} b_{i+} \; b_{i+2} - \sum_{i=0}^{n-2} [1+p_{11}(t+n-i-1)] \; c_{i+1}(t+n-i-1)c_{i+2}(t$$

$$= \sum_{i=1}^{n-1} b_i b_{i+1} - \sum_{i=1}^{n-1} \left[1+p_{11}(t+n-i)\right] \; c_i(t+n-i)c_{i+1}(t+n-i)$$

.

.

.

$$p_{n1}(t+n) = b_1 b_n - [1+p_{11}(t+n-1)]c_n(t+n-1)c_1(t+n-1)$$

Hence

$$p_{11}(t+n) = \sum_{i=1}^{n} [b_i^2 - \sigma^2(t+n-1)c_i^2(t+n-i)]$$

$$p_{21}(t+n) = \sum_{i=1}^{n-1} [b_i b_{i+1} - \sigma^2(t+n-i)c_i(t+n-i)c_{i+1}(t+n-i)]$$

.

.

.

$$p_{n1}(t+n) = b_1 b_n - \sigma^2(t+n-1)c_1(t+n-1)c_n(t+n-1)$$

where

$$\sigma^2(t) = [1+p_{11}(t)]$$

Furthermore introduce

$$c_i(t) = \frac{p_{i+11}(t)+b_i}{1+p_{11}(t)} = \frac{p_{i+11}(t)+b_i}{\sigma^2(t)}$$

The above equations then can be written as

$$\sigma^2(t+n) = 1 + \sum_{i=1}^{n} [b_i^2 - \sigma^2(t+n-i)c_i^2(t+n-i)] =$$

$$= \sum_{i=0}^{n} b_i^2 - \sum_{i=1}^{n} \sigma^2(t+n-i)c_i^2(t+n-i)$$

where

$$b_o = 1$$

Furthermore

$$c_i(t+n) = \frac{P_{i+1}(t+n)+b_i}{\sigma^2(t+n)} = \frac{P_{i+1}(t+n)+b_o b_i}{\sigma^2(t+n)} =$$

$$= [\sum_{j=0}^{n-i} b_j b_{j+i} - \sum_{j=1}^{n-i} \sigma^2(t+n-j)c_j(t+n-j)d_{j+i}(t+n-j)]/\sigma^2(t+n)$$

The algorithm thus can be written as follows

$$\sigma^2(t+n) = \sum_{i=0}^{n} b_i^2 - \sum_{i=1}^{n} \sigma^2(t+n-i)c_i^2(t+n-i)$$

$$c_1(t+n) = [\sum_{i=0}^{n-1} b_i b_{i+1} - \sum_{j=1}^{n-1} \sigma^2(t+n-j)c_j(t+n-j)c_{j+1}(t+n-j)]/\sigma^2(t+n)$$

$$c_{n-1}(t+n) = [b_o b_{n-1} + b_1 b_n - \sigma^2(t+n-1)c_1(t+n-1)c_n(t+n-1)]/\sigma^2(t+n)$$

$$c_n(t+n) = b_o b_n/\sigma^2(t+n)$$

The algorithm discussed thus can be interpreted as a modified subs
algorithm for solving the algebraic equation

$$\sigma^2(1+c_1^2 + c_2^2+\ldots+ c_n^2) = b_o^2 + b_1^2+\ldots+ b_n^2$$

$$\sigma^2(c_1 + c_1c_2+\ldots+c_{n-1}c_n) = b_o b_1 + b_1 b_2+\ldots+b_n b_{n-1}$$

$$\sigma^2(c_{n-1} + c_1 c_n) = b_o b_{n-1} + b_1 b_n$$

$$\sigma^2 \cdot c_n = b_o b_n$$

Notice, however, that the special algorithm implies that the solut
will converge to a solution where all the zeros of the C-polynomia
are inside the unit circle. An alternative thus would be the follo
algorithm

$$\sigma^2(t+1) = \sum_{i=0}^{n} b_1^2 - \sum_{i=1}^{n} \sigma^2(t)c_i^2(t)$$

$$c_1(t+1) = [\sum_{i=0}^{n-1} b_i b_{i+1} - \sum_{i=1}^{n-1} \sigma^2(t)c_i(t)c_{i+1}]/\sigma^2(t+1)$$

$\cdot$
$\cdot$
$\cdot$

This algorithm does not require as much storage as (12). However, i is not immediate clear that it will always converge to the correct solution.

An Alternative

Another possibility to do the spectral factorization would be to solve the nonlinear equations by a Newton-Raphson algorithm. This gives

$$c(t+1) = c(t) - [f'(c(t))]^{-1} f(c(t))$$

where

$$f(c) = \begin{bmatrix} c_o^2+c_1^2+\ldots+c_n^2 \\ c_oc_1+\ldots+c_{n-1}c_n \\ \vdots \\ c_oc_n \end{bmatrix} = \begin{bmatrix} c_o & c_1 \ldots c_n \\ 0 & c_o \ldots c_{n-1} \\ \vdots & \\ 0 & 0 \ldots c_o \end{bmatrix} \begin{bmatrix} c_o \\ c_1 \\ \\ c_n \end{bmatrix}$$

$$f'(c) = \begin{bmatrix} c_o & c_1 & \ldots & c_n \\ 0 & c_o & \ldots & c_{n-1} \\ \vdots & & & \\ 0 & 0 & & c_o \end{bmatrix} + \begin{bmatrix} c_o & c_1 & \ldots & c_{n-1} & c_n \\ c_1 & c_2 & \ldots & c_n & 0 \\ \vdots & & & & \\ c_n & 0 & \ldots & 0 & 0 \end{bmatrix}$$

This algorithm most likely will converge faster than the other algorithms. However, it is again not obvious that it will converge to the correct result.

5. IMPLEMENTATION

The algorithm has been implemented in FORTRAN on PDP 15/35 as a program called SPFZN (Spectral FactoriZatioN). A test program TSPFZN and an interactive users program USPFZN also are available. These programs are listed below. A printout of the test routine also is provided.

PRINTOUT FROM TEST PROGRAM TSPFZN

```
TEST OF SPFZN
THE ORIGINAL POLYNOMIAL IS
 1.00000000 -2.50000000 -1.00000000  5.00000000 -1.00999999 -2.47500002  0.98999999
THE FACTORED POLYNOMIAL IS
 2.20061284 -2.19992346 -3.43050772  3.97954106  0.80510892 -1.79964948  0.44987468
THE EXACT VALUE OF FACTORED POLYNOMIAL IS
 2.19999999 -2.19999999 -3.43000001  3.98000002  0.80500001 -1.80000001  0.45000000
NUMBER OF ITERATIONS =  100
COMPUTING TIME          8.0 SEC
P-MATRIX
 3.84269691 -2.34117979 -6.54921961  3.75742924  2.78173304 -1.48533177
-2.34117979  2.43286926  2.70506734 -2.80412722 -0.53823970  0.55312569
-6.54921949  2.70506752 13.20107508 -5.94561279 -6.57609677  3.15991336
 3.75742924 -2.80412722 -5.94561267  4.03701556  2.30756092 -1.36199445
 2.78173304 -0.53823970 -6.57609689  2.30756092  3.66526520 -1.64079985
-1.48533177  0.55312569  3.15991336 -1.36199445 -1.64079985  0.77775559
R-MATRIX
-0.00102502  0.00091904  0.00075185 -0.00083500  0.00013208  0.00002334
 0.00091869 -0.00030309 -0.00119376  0.00042093  0.00033820 -0.00013497
 0.00075197 -0.00119364 -0.00003147  0.00093842 -0.00055242  0.00009680
-0.00083494  0.00042105  0.00093853 -0.00047457 -0.00017893  0.00009057
 0.00013208  0.00033820 -0.00055254 -0.00017893  0.00038302 -0.00010294
 0.00002337 -0.00013497  0.00009674  0.00009057 -0.00010294  0.00002444
```

LISTING OF SUBROUTINE SPFZN

```
      SUBROUTINE SPFZN(B,C,C0,N,EPS,IND)
C     THIS SUBROUTINE DETERMINES A POLYNOMIAL
C        C(Z)= C0*(Z**N+C(1)*Z**(N-1)+...+C(N))
C     WITH ZEROES INSIDE THE UNIT CIRCLE SUCH THAT
C        C(Z)*C"(Z)=B(Z)*B"(Z)
C     WHERE
C        B(Z)=Z**N+B(1)*Z**(N-1)+...+B(N)
C
C     AND B" DENOTES THE RECIPROCIAL POLYNOMIAL
C
C     REFERENCE K.J. ASTROM "A SPECTRAL FACTORIZATION ALGORITHM"
C
C     AUTHOR K.J. ASTROM 1971-12-29
C
C        B  - VECTOR CONTAINING COEFFICIENTS OF POLYNOMIAL
C             TO BE FACTORED. IT IS ASSUMED THAT LEADING
C             COEFFICIENT OF B IS 1.
C        C  - VECTOR CONTAINING COEFFICIENTS OF THE FACTORED
C             POLYNOMIAL AS GIVEN ABOVE. NOTICE THAT
C             THE COEFFICIENTS ARE NORMALIZED BY C0
C        C0 - COEFFICIENT OF LEADING TERM OF POLYNOMIAL C
C        N  - DEGREE OF POLYNOMIALS B AND C (MAX 15).
C        EPS- TEST QUANTITY TO STOP ITERATION OF RICCATI EQUATION
C        IND- INDICATOR RETURNED AS 1 IF THE ITERATION
C             DOES NOT CONVERGE RETURNED AS -NLOOP OTHERWISE
C
C     SUBROUTINES REQUIRED
C        NORM
C
      DIMENSION B(1),C(1)
      COMMON/SLASK/P(16,16),R(16,16)
C
      IS=16
      NLOP=1000
      NP1=N+1
      DO 10 I=1,NP1
      DO 10 J=1,NP1
      R(I,J)=0.0
   10 P(I,J)=0.0
      DO 11 I=1,N
   11 P(I,I)=1.
      NLOOP=0
C
C     MAIN LOOP COMPUTE SOLUTION OF RICCATI EQUATION
C
   20 RI=1./(1.+P(1,1))
      NLOOP=NLOOP+1
      DO 21 I=1,N
      DO 21 J=1,N
   21 R(I,J)=P(I,J)
      DO 22 I=1,N
      DO 22 J=1,N
   22 P(I,J)=B(I)*B(J)+R(I+1,J+1)-(R(I+1,1)+B(I))*(R(J+1,1)+B(J))*RI
      DO 25 I=1,N
      DO 25 J=1,N
   25 R(I,J)=P(I,J)-R(I,J)
C
C     TEST FOR STEADY STATE
C
      CALL NORM(R,N,IS,RNORM)
      CALL NORM(P,N,IS,PNORM)
      IF(RNORM-EPS*PNORM) 24,24,23
   24 IND=-NLOOP
      GO TO 29
   23 IF(NLOOP-NLOP) 20,20,28
   28 IND=1
C
C     COMPUTE C
C
   29 T=1.+P(1,1)
      RI=1./T
      DO 30 I=1,N
   30 C(I)=(P(I+1,1)+B(I))*RI
      C0=SQRT(T)
```

LISTING OF TESTPROGRAM TSPFZN

```
C         THIS IS A TEST PROGRAM FOR SPFZN
          DIMENSION B(16),C(16),D(16)
          COMMON/SLASK/P(16,16),R(16,16)
C
          WRITE (6,100)
100       FORMAT (14H TEST OF SPFZN)
          N=6
          NP1=N+1
          B0=1.
          B(1)=-2.5
          B(2)=-1.
          B(3)=5.
          B(4)=-1.01
          B(5)=-2.475
          B(6)=.99
C
          D(1)=2.2
          D(2)=-2.2
          D(3)=-3.43
          D(4)=3.98
          D(5)=0.805
          D(6)=-1.8
          D(7)=0.45
C
          EPS=1.E-3
          DO 50 I=1,3
          EPS=EPS/10.
          IOF=0
          CALL TIME10(IM,IS,IS10,IOF)
          CALL SPFZN (B,C,C0,N,EPS,IND)
          IOF=1
          IS1=60*IM+IS
          TIM=FLOAT(IS1)+0.1*FLOAT(IS10)
          WRITE (6,103)
103       FORMAT (27H THE ORIGINAL POLYNOMIAL IS)
          WRITE (6,104) B0,(B(I),I=1,N)
104       FORMAT (10F12.8)
          WRITE (6,105)
105       FORMAT (27H THE FACTORED POLYNOMIAL IS)
          DO 20 I=1,N
20        C(I)=C(I)*C0
          WRITE (6,104) C0,(C(I),I=1,N)
          NLOP=-IND
106       FORMAT (23H NUMBER OF ITERATIONS =,I5)
          WRITE (6,108)
          WRITE (6,104) (D(I),I=1,NP1)
          WRITE (6,106) NLOP
          WRITE (6,102) TIM
102       FORMAT (15H COMPUTING TIME,F12.1,4H SEC)
108       FORMAT (42H THE EXACT VALUE OF FACTORED POLYNOMIAL IS)
          WRITE (6,109)
109       FORMAT (9H P-MATRIX)
          DO 30 I=1,N
30        WRITE (6,104) (P(I,J),J=1,N)
          WRITE (6,110)
110       FORMAT (9H R-MATRIX)
          DO 32 I=1,N
32        WRITE (6,104) (R(I,J),J=1,N)
50        CONTINUE
          END
```

LISTING OF INTERACTIVE USER PROGRAM USPFZN

```
C        THIS IS A  PROGRAM FOR USE OF SPFZN
C        THE PROGRAM REQUESTS THE POLYNOMIAL B FROM THE TTY
C        IT THEN CALLS SPFZN AND COMPUTES THE FACTORED POLYNOMIAL
         DIMENSION B(16),C(16)
         COMMON /SLASK/P(16,16),R(16,16)
         DATA A/1HY/
         B0=1.
    1    WRITE (9,100)
  100    FORMAT (12H0TYPE DEGREE)
         ICNTRL=1
         N=RTTFF(ICNTRL)
         WRITE (9,101)
  101    FORMAT (16H TYPE COEFF OF B)
         ICNTRL=1
         DO 10 I=1,N
   10    B(I)=RTIFF(ICNTRL)
         ICNTRL=1
         WRITE (9,102)
  102    FORMAT (9H TYPE EPS)
         ICNTRL=1
         EPS=RTTFF(ICNTRL)
         IOF=0
         CALL TIME10(IM,IS,IS10,IOF)
         CALL SPFZN (B,C,C0,N,EPS,IND)
         IOF=1
         IS1=60*IM+IS
         TIM=FLOAT(IS1)+0.1*FLOAT(IS10)
         WRITE (9,103)
  103    FORMAT (27H THE ORIGINAL POLYNOMIAL IS)
         WRITE (9,104) B0,(B(I),I=1,N)
  104    FORMAT (9F8.4)
         WRITE (9,105)
  105    FORMAT (27H THE FACTORED POLYNOMIAL IS)
         DO 20 I=1,N
   20    C(I)=C(I)*C0
         WRITE (9,104) C0,(C(I),I=1,N)
         NLOP=-IND
         WRITE (9,106) NLOP
  106    FORMAT (23H NUMBER OF ITERATIONS =,I5)
         WRITE (9,120) TIM
  120    FORMAT (15H COMPUTING TIME,F12.1,4H SEC)
         WRITE (9,107)
  107    FORMAT (44H DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O))
         READ (8,110) ANS
  110    FORMAT (A5)
         IF(ANS.NE.A) GO TO 1
   30    DO 32 I=1,N
   32    WRITE (9,108) (P(I,J),J=1,N)
  108    FORMAT (6E12.4)
         WRITE (9,109)
  109    FORMAT (35H DO YOU WANT R-MATRIX Y(ES) OR N(O))
         READ (8,110) ANS
         IF (ANS.NE.A) GO TO 1
         DO 34 I=1,N
   34    WRITE (9,108) (R(I,J),J=1,N)
         GO TO 1
         END
```

6. TEST EXAMPLES

To construct test examples it is observed that if

$$B(z) = \Pi(z-\alpha_i)\pi(z-\beta_i)$$

where α_i are the zeros inside the unit circle and β_i the zeros outs
the unit circle. Then

$$B^*(z) = \Pi(\alpha_i z - 1)\pi(\beta_i z - 1)$$

and we find

$$C(z) = \Pi(z - \alpha_i)\pi(\beta_i z - 1)$$

In particular we find that if the polynomial B has all its zeros outside the unit circle then

$$B(z) = b_o z^n + b_1 z^{n-1} + \ldots + b_n$$

implies that

$$C(z) = b_n z^n + b_1 z^{n-1} + \ldots + b_o$$

The first set of examples is designed to check the accuracy of the
algorithm. The following examples were selected.

Example 1

$$B(z) = (z-2)(z-0.5) = z^2 - 2.5z + 1$$

$$C(z) = (z-0.5)(2z-1) = 2z^2 - 2z + 0.5$$

Example 2

$$C(z) = (z - \frac{9}{10})(\frac{11}{10} z - 1) = \frac{11}{10} z^2 - \frac{199}{100} z + \frac{9}{10}$$

Example 3

$$B(z) = (z - \frac{99}{100})(z - \frac{101}{100}) = z^2 - 22 + 0.9999$$

$$C(z) = (z - \frac{99}{100})(\frac{101}{100} z - 1) = \frac{101}{100} z^2 - \frac{19999}{10000} z + \frac{99}{100}$$

Example 4

$$B(z) = (z + 1)^2 = z^2 + 2z + 1$$

$$C(z) = (z + 1)^2 = z^2 + 2z + 1$$

Example 5

$$B(z) = (z - 1)^2 = z^2 - 2z + 1$$

$$C(z) = (z - 1)^2 = z^2 - 2z + 1$$

Example 6

$$B(z) = (z + 1)^3 = z^3 + 3z^2 + 3z + 1$$

$$C(z) = (z + 1)^3 = z^3 + 3z^2 + 3z + 1$$

Example 7

$$B(z) = z^{15} + 1.01$$

$$C(z) = 1.012^{15} + 1$$

Example 8

$$B(z) = (z^5+1.01)(z^5+0.99) = z^{10} + 22^5 + 0.9999$$

$$C(z) = (z^5+0.99)(1.01z^5+1) = 1.01z^{10} + 1.9999z^5 + 0.99$$

Example 9

$$B(z) = (z^3+2)(z^3+0.5) = z^6 + 2.5z^3 + 1$$

$$C(z) = (2z^3+1)(z^3+0.5) = 2z^6 + 2.5z^3 + 0.5$$

Example 10

$$B(z) = (z^2-0.9)(z^2-1.1)(2-0.5)(z-2)$$

$$= z^6-2.5z^5-z^4+5z^3 - 1.01z^2 - 2.415z + 0.99$$

$$C(z) = (z^2 - 0.9)(1.1z^2 - 1)(z - 0.5)(2z - 1)$$

$$= 2.2z^6 - 2.2z^5 - 3.43z^4 + 3.98z^3 - 0.805z^2 - 1.8z + 0.45$$

The examples were calculated using the program USPFZN. The results are given below.

```
TYPE DEGREE
#
2 EXAMPLE 1
TYPE COEFF OF B
#-2.5 1.
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000 -2.5000  1.0000
THE FACTORED POLYNOMIAL IS
 2.0000 -2.2000  0.5000
NUMBER OF ITERATIONS =    11
COMPUTING TIME           0.1 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
Y
 0.3000E+01 -0.1500E+01
-0.1500E+01  0.7500E+00
DO YOU WANT R-MATRIX Y(ES) OR N(O)
Y
-0.1243E-03  0.5633E-04
 0.5633E-04 -0.2554E-04

TYPE DEGREE
#
2 EXAMPLE 2
TYPE COEFF OF B
#-2. 0.99
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000 -2.0000  0.9900
THE FACTORED POLYNOMIAL IS
 1.1001 -1.9900  0.9000
NUMBER OF ITERATIONS =    61
COMPUTING TIME           0.6 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
2 EXAMPLE 3
TYPE COEFF OF B
#-2. 0.9999
TYPD EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000 -2.0000  0.9999
THE FACTORED POLYNOMIAL IS
 1.0105 -1.9999  0.9895
NUMBER OF ITERATIONS =   397
COMPUTING TIME           3.9 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N
```

```
TYPE DEGREE
#
2 EXAMPLE 3 WITH SMALLER EPS=1.E-8
TYPE COEFF OF B
#-2. 0.9999
TYPE EPS
#1.E-7
THE ORIGINAL POLYNOMIAL IS
 1.0000 -2.0000  0.9999
THE FACTORED POLYNOMIAL IS
 1.0110 -1.9999  0.9890
NUMBER OF ITERATIONS =    -1
COMPUTING TIME            9.9 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
Y
 0.2216E-01 -0.2191E-01
-0.2191E-01  0.2167E-01
DO YOU WANT R-MATRIX Y(ES) OR N(O)
Y
 0.8904E-05 -0.8689E-05
-0.8731E-05  0.8507E-05

TYPE DEGREE
#
2 EXAMPLE 4
TYPE COEFF OF B
#2. 1.
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  2.0000  1.0000
THE FACTORED POLYNOMIAL IS
 1.0058  2.0000  0.9943
NUMBER OF ITERATIONS =  460
COMPUTING TIME            4.5 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
2 EXAMPLE 4 WITH SMALLER EPS=1.E-7
TYPE COEFF OF B
#2. 1.
TYPE EPS
#1.E-7
THE ORIGINAL POLYNOMIAL IS
 1.0000  2.0000  1.0000
THE FACTORED POLYNOMIAL IS
 1.0064  2.0000  0.9937
NUMBER OF ITERATIONS =    -1
COMPUTING TIME            9.9 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
```

```
TYPE DEGREE
#
2 EXAMPLE 5
TYPE COEFF OF B
#-2. 1.
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000 -2.0000  1.0000
THE FACTORED POLYNOMIAL IS
 1.0058 -2.0000  0.9943
NUMBER OF ITERATIONS =  460
COMPUTING TIME           4.5 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
3 EXAMPLE 6
TYPE COEFF OF B
#3. 3. 1.
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  3.0000  3.0000  1.0000
THE FACTORED POLYNOMIAL IS
 1.0536  3.0509  2.9464  0.9491
NUMBER OF ITERATIONS =  724
COMPUTING TIME          15.2 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
15 EXAMPLE 7
TYPE COEFF OF B
#0 0 0 0 0 0 0 0 0 0 0 0 0 1.0100000000
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0100
THE FACTORED POLYNOMIAL IS
 1.0136  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.9965
NUMBER OF ITERATIONS =    -1
COMPUTING TIME         511.1 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N
```

```
TYPE DEGREE
#
10 EXAMPLE 8
TYPE COEFF OF B
#0 0 0 0 2. 0 0 0 0 0.9999
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  0.0000  0.0000  2.0000  0.0000  0.0000  0.
 0.0000  0.9999
THE FACTORED POLYNOMIAL IS
 1.0129  0.0000  0.0000  0.0000  0.0000  1.9999  0.0000  0.0000  0.
 0.0000  0.9871
NUMBER OF ITERATIONS =    -1
COMPUTING TIME          227.4 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
6 EXAMPLE 9
TYPE COEFF OF B
#0 0 2.5 0 0 1.
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  2.5000  0.0000  0.0000  1.0000
THE FACTORED POLYNOMIAL IS
 2.0000  0.0000  0.0000  2.0000  0.0000  0.0000  0.5000
NUMBER OF ITERATIONS =    31
COMPUTING TIME          2.5 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
6 EXAMPLE 10
TYPE COEFF OF B
#-2.5 -1. 5. -1.01 -2.475 0.99
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000 -2.5000 -1.0000  5.0000 -1.0100 -2.4750  0.9900
THE FACTORED POLYNOMIAL IS
 2.2006 -2.2000 -3.4305  3.9796  0.8051 -1.7997  0.4499
NUMBER OF ITERATIONS =  100
COMPUTING TIME          8.0 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N
```

The next set of examples were selected in order to demonstrate how the computing time is influenced by the location of the zeros of the B-polynomial. Second order polynomials of the form

$$B(z) = z^2 + a^2 \qquad a > 1$$

were selected. The following values of a were used a = 2 (Example 11 a = 1.5 (Example 12), a = 1.1 (Example 13), a = 1.01 (Example 14). The results are given in Table . Summarizing the number of iteratic required and the computing time we get

Table 1. Illustrates the dependence of the convergence rate on the location of the zeros of the B - polynomial

a	Number of iterations	Computing time
2	11	< 0.1 s
1.5	15	< 0.1 s
1.1	41	0.4 s
1.01	299	3.0 s

```
TYPE DEGREE
#
2 EXAMPLE 11
TYPE COEFF OF B
#0 4
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  4.0000
THE FACTORED POLYNOMIAL IS
 4.0000  0.0000  1.0000
NUMBER OF ITERATIONS =    11
COMPUTING TIME            0.1 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
2 EXAMPLE 12
TYPE COEFF OF B
#0 2.25
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  2.2500
THE FACTORED POLYNOMIAL IS
 2.2500  0.0000  1.0000
NUMBER OF ITERATIONS =    15
COMPUTING TIME            0.1 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
2 EXAMPLE 13
TYPE COEFF OF B
#0 1.21
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  1.2100
THE FACTORED POLYNOMIAL IS
 1.2101  0.0000  1.0000
NUMBER OF ITERATIONS =    41
COMPUTING TIME            0.4 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
2 EXAMPLE 14
TYPE COEFF OF B
#0 1.0201
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  1.0201
```

The next set of examples (Example 15 - Example 18) are designed in order to show how the computing time depends on the size of the problem. The following polynomials were used

$$B(z) = z^n - 2^n$$

The results are tabulated below.

Table 2. Illustrates how the computing time depends on the order of the B - polynomial.

Order of polynomial	Computing time	Example
3	0.2 s	15
4	0.4 s	16
5	0.9 s	17
6	1.5 s	18

```
TYPE DEGREE
#
3 EXAMPLE 15
TYPE COEFF OF B
#0 0 8
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  8.0000
THE FACTORED POLYNOMIAL IS
 8.0000  0.0000  0.0000  1.0000
NUMBER OF ITERATIONS =    13
COMPUTING TIME               0.2 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES)  OR  N(O)
N

TYPE DEGREE
#
4 EXAMPLE 16
TYPE COEFF OF B
#0 0 0 16
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  0.0000 16.0000
THE FACTORED POLYNOMIAL IS
15.9999  0.0000  0.0000  0.0000  1.0000
NUMBER OF ITERATIONS =    13
COMPUTING TIME               0.4 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES)  OR  N(O)
N

TYPE DEGREE
#
5 EXAMPLE 17
TYPE COEFF OF B
#0 0 0 0 32
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  0.0000  0.0000 32.0000
THE FACTORED POLYNOMIAL IS
32.0000  0.0000  0.0000  0.0000  0.0000  1.0000
NUMBER OF ITERATIONS =    16
COMPUTING TIME               0.9 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES)  OR  N(O)
N

TYPE DEGREE
#
6 EXAMPLE 18
TYPE COEFF OF B
#0 0.0 0 0 64
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
                          0.0000  0.0000  0.0000 64.0000
```

The examples 19 - 23 also are constructed in order to show how computing time depends on the order of the system. The following polynomials are used

$$B(z) = z^{2n} - 2^n$$

The results are tabulated below

Table 3. Illustrates how the computing time depends on the order of the polynomial.

Order of polynomial	Computing time	Example
2	0.1	19
4	6.7	20
6	2.1	21
8	3.6	22
10	7.0	23

```
TYPE DEGREE
#
2 EXAMPLE 19
TYPE COEFF OF B
#0 2.
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  2.0000
THE FACTORED POLYNOMIAL IS
 1.9999  0.0000  1.0000
NUMBER OF ITERATIONS =    15
COMPUTING TIME            0.1 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
4 EXAMPLE 20
TYPE COEFF OF B
#0 0 0 4.
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  0.0000  4.0000
THE FACTORED POLYNOMIAL IS
 4.0000  0.0000  0.0000  0.0000  1.0000
NUMBER OF ITERATIONS =    21
COMPUTING TIME            0.7 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
6 EXAMPLE 21
TYPE COEFF OF B
#0 0 0 0 0 8
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  8.0000
THE FACTORED POLYNOMIAL IS
 8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000
NUMBER OF ITERATIONS =    25
COMPUTING TIME            2.1 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N
```

```
TYPE DEGREE
#
8 EXAMPLE 22
TYPE COEFF OF B
#0 0 0 0 0 0 0K\ 16.
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000 1
THE FACTORED POLYNOMIAL IS
15.9999  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
NUMBER OF ITERATIONS =    25
COMPUTING TIME           3.6 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N

TYPE DEGREE
#
10 EXAMPLE 23
TYPE COEFF OF B
#0 0 0 0 0 0 0 0 0 32.
TYPE EPS
#1.E-4
THE ORIGINAL POLYNOMIAL IS
 1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0
 0.0000 32.0000
THE FACTORED POLYNOMIAL IS
32.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0
 0.0000  1.0000
NUMBER OF ITERATIONS =    31
COMPUTING TIME           7.0 SEC
DO YOU WANT COVARIANCE MATRIX Y(ES) OR N(O)
N
```