ISE-TR-86-61

NUMERICAL ALGORITHM FOR SOLVING A POLYNOMIAL EQUATION IN H_∞ OPTIMIZATION PROBLEM

by

Masami SAEKI

and

Eddie KORNEGOOR

December 15, 1986

INSTITUTE
OF
INFORMATION SCIENCES AND ELECTRONICS

UNIVERSITY OF TSUKUBA

Numerical algorithm for solving a polynomial equation in H_∞ optimization problem.

Masami Saeki[†] & Eddie Kornegoor[††]

†) Institute of Information Sciences and Electronics
University of Tsukuba
Sakuramura, Niiharigun, Ibaraki 305, JAPAN

††) Department of Applied Mathematics
Twente University of Technology
7500 AE Enschede, The Netherlands

Abstract:

A new numerical algorithm is proposed for solving a polynomial equation arising in the H_∞ optimization problem. The solution is obtained by solving eigenvalue problems iteratively. The algorithm is not computationally demanding and is easy to implement. Numerical examples show that the solution can be obtained after only a few iterations.

Keywords:

Numerical algorithm, H_∞ optimization, polynomial equation, eigenvalue problem, single-input, single-output, discrete time system.

1. Introduction

At the present time, the numerical algorithms used to solve the H_∞ optimization problem include a bisection method for the Hankel norm approach in [1] and the polynomial approach in [2]. Thus, more efficient numerical solution procedures are sought.

In the polynomial approach, the H_∞ optimization problem is reduced to the solution of a certain polynomial equation. The analysis for a single-input, single-output discrete time system was reported in [3]. The solution equation was found by applying a LQG (linear quadratic guassian) optimal control method. Furthermore, the derived solution equation was shown to strictly correspond to that obtained for the continuous time case, [2]. In [3], a numerical algorithm was proposed for solving the polynomial equation. The route followed was to first obtain an approximate solution by solving an eigenvalue problem and then to obtain the exact solution by a Newton Raphson method.

The purpose of this paper is to propose a new algorithm to obtain the exact solution by extending the idea of the approximation used in the first step of [3].

<u>Notation</u>

R^{nxm}	space of nxm real matrices
$P(z^{-1})$	space of polynomials of z^{-1} with real coefficients
deg(A)	degree of $A \in P(z^{-1})$; the largest integer i for which $a_i \neq 0$ where $A(z^{-1}) = a_0 + a_1 z^{-1} + \ldots + a_n z^{-n}$
Toep(A)	Toeplitz matrix of $A \in P(z^{-1})$;

$$\mathrm{Toep}(A) = \begin{bmatrix} a_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_n & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & a_0 \\ \vdots & & \ddots & \vdots \\ \vdots & & & a_n \\ \vdots & & & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

where $A(z^{-1}) = a_0+a_1z^{-1}+\ldots+a_nz^{-n}$

* adjoint of a rational function of z^{-1}; $S^*(z^{-1}) = S(z)$

$A\in P(z^{-1})$ is called strictly Hurwitz, if all the zeros of A are inside the unit circle.

+, 0, - superscripts +, 0, - denote that the zeros of the polynomial are inside, on, and outside the unit circle, respectively.

2. H_∞ optimization problem and the polynomial equation

Consider the system shown in Fig.1 where the transfer functions of the plant and the controller are given by:

$$W(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} \text{ and } C_o(z^{-1}) = \frac{C_{on}(z^{-1})}{C_{od}(z^{-1})} \tag{1}$$

respectively. $A, B, C_{od}, C_{on} \in P(z^{-1})$. The plant is assumed to be free of unstable hidden modes. The factorizations of A and B are introduced:

$$A=A^+A^oA^- \text{ and } B=B^+B^oB^- \tag{2}$$

Consider the H_∞ - norm type cost function:

$$J_\infty = \sup_{|z|=1} (Q(z^{-1})\, S^*(z^{-1})\, S(z^{-1}) + R(z^{-1})\, T^*(z^{-1})\, T(z^{-1})) \tag{3}$$

The weights Q and R are:

$$Q = \frac{B_q^*B_q}{A_q^*A_q} \quad \text{and} \quad R = \frac{B_r^*B_r}{A_r^*A_r} \tag{4}$$

where A_q, B_q, A_r, $B_r \in P(z^{-1})$, and A_q and A_r are strictly Hurwitz. The return differnece S and the complementary return difference T are given by $S = (1+C_oW)^{-1}$ and $T = C_oW(1+C_oW)^{-1}$, respectively. This cost function corresponds to that of the continuous time case of [2]. The H_∞ optimization problem is to obtain a controller which minimizes J_∞ under the constraint that the closed-loop system is stable.

This problem is reduced to that of solving a suite of polynomial equations with respect to $\lambda \in R$ and $G, H, F^-, A_\sigma \in P(z^{-1})$ namely [3]:

$$A_r B^- G + A_q A^- H - F_s^- = 0 \tag{5}$$

$$B_r^* B_r A_q^* B^- Z^{-n} G - B_q^* B_q A_r^* A^- Z^{-n} H + A_\sigma F^- = 0 \tag{6}$$

$$A_\sigma^* A_\sigma = D_e^* D_e \lambda^2 - B_q^* B_q B_r^* B_r \tag{7}$$

where the spectral factor $D_e \in P(z^{-1})$ is defined by

$$D_e^* D_e = A_r^* A_r B_q^* B_q + A_q^* A_q B_r^* B_r \tag{8}$$

and D_e is assumed to be strictly Hurwitz and $n \triangleq \deg(D_e)$. The unknown polynomials G, H, F^-, A_σ satisfy $\deg(F^-) = \deg(D_e A^- B^-)-1$, $\deg(G) \leqq \deg(A^-)+\max(\deg(A_q), \deg(B_q))-1$, $\deg(H) \leqq \deg(B^-)+\max(\deg(A_r), \deg(B_r))-1$, and $\deg(A_\sigma) = n$. The polynomial $F_s^- \in P(z^{-1})$ is defined by $F_s^- \triangleq (F^-)^* z^{-m}$ where $m \triangleq \deg(F^-) = \deg(D_e A^- B^-) - 1$. The polynomials A_σ and F_s^- need to be strictly Hurwitz.

The optimal controller is given by:

$$C_o = (B^o B^+ A_q H)^{-1} (A^o A^+ A_r G) \tag{9}$$

and the minimal value of J_∞ is λ^2. For details such as the existence of a solution, see [3]. In the later sections it is assumed that the equations (4), (5), and (6) are well-defined and with an appropriate solution.

3. Numerical solution procedures [3]

3.1 Algorithm 1

If the value of λ is given, A_σ is uniquely determined by (7) except for the sign. Then (5) and (6) can be considered as a linear polynomial equation with respect to G, H, and F^-. By expanding (5) and (6) in powers of z^{-1} and equating the coefficients of like powers, the following linear algebraic equations obtain:

$$\Gamma_{11} g + \Gamma_{12} h + \Gamma_{13} f = 0 \tag{10}$$

$$\Gamma_{21} g + \Gamma_{22} h + \Gamma_{23} f = 0 \tag{11}$$

where $g \in R^{n_g}$, $h \in R^{n_h}$, $f \in R^{n_f}$, and $n_g \triangleq \deg(A^-)+\max(\deg(A_q), \deg(B_q))$, $n_h \triangleq \deg(B^-)+\max(\deg(A_r), \deg(B_r))$, $n_f \triangleq \deg(D_e A^- B^-)=m+1$. The elements of g, h, and f are the coefficients of the polynomials G, H, F^-, respectively. For example f is defined as $f \triangleq (f_0, f_1, \ldots, f_m)^T$ where $F^- = f_0 + f_1 z^{-1} + \ldots + f_m z^{-m}$. The matrices Γ_{ij} (i=1, 2; j=1, 2, 3) are given by $\Gamma_{11} = \text{Toep}(A_r B^-) \in R^{n_1 x n_g}$,

$\Gamma_{12} = \text{Toep}(A_q A^-) \in R^{n_1 x n_h}$, $\Gamma_{21} = \text{Toep}(B_r^* B_r A_q^* B^- z^{-n}) \in R^{n_2 x n_g}$,

$\Gamma_{22} = \text{Toep}(-B_q^* B_q A_r^* A^- z^{-n}) \in R^{n_2 x n_f}$, $\Gamma_{23} = \text{Toep}(A_\sigma) \in R^{n_2 x n_h}$

where $n_1 \triangleq m+1$ and $n_2 \triangleq m+1+\deg(D_e)$, and $\Gamma_{13} \in R^{n_f x n_f}$ is a matrix whose (i, n_f-i+1) th element is -1 for i=1, ..., n_f and other elements are zero.

By using $\Gamma_{13}^2 = I$, the equations (10) and (11) reduce to

$$\tilde{\Gamma} \begin{bmatrix} g \\ h \end{bmatrix} = \{[\Gamma_{21} \mid \Gamma_{22}] - \Gamma_{23}\Gamma_{13}[\Gamma_{11} \mid \Gamma_{12}]\} \begin{bmatrix} g \\ h \end{bmatrix} = 0 \tag{12}$$

Since $\tilde{\Gamma}$ is a $[m+1+\deg(D_e)] \times (m+1)$ matrix, (12) has a non-zero solution if and only if, rank $\tilde{\Gamma} \leqq m$. The optimality of a solution can be considered as the solvability of (12) with a non-zero solution. Let the singular values of $\tilde{\Gamma}$ be $\theta_1 \geqq \theta_2 \geqq \cdots \geqq \theta_{m+1}$. Then the reciprocal of the condition number of $\tilde{\Gamma}$, i.e. $E=\theta_{m+1}/\theta_1$, is adopted as the optimality index. When the index E is zero, rank $\tilde{\Gamma} \leqq m$.

If E is zero at several values of λ, the largest absolute value of λ gives the optimal λ. These results are used to construct the following algorithm.

Algorithm 1:

Step 1) Calculate the singular values of $\tilde{\Gamma}$ for given values of λ and obtain the index E.

Step 2) The largest absolute value of λ for which E is zero gives the optimal λ denoted λ^o. The solution g and h of (12) for λ^o gives

The graph (E, λ) usefully indicates the global characteristic of (12). The drawback of this algorithm is that the spectral factorization and singular value decomposition necessary to compute E for each λ, makes the algorithm computationally demanding.

3.2 Algorithm 2

A more efficient algorithm is given in the following.

The optimal λ can be estimated by solving an eigenvalue problem. From (7) A_σ is approximated as:

$$A_\sigma \cong D_e \lambda \tag{13}$$

This is a good approximation for large values of λ's.

Then, (12) can be approximated as:

$$\hat{\Gamma}\, x = (\Gamma_a - \lambda\, \Gamma_b) x = 0 \tag{14}$$

where $\Gamma_a \triangleq [\Gamma_{21} \vdots \Gamma_{22}]$, $\Gamma_b \triangleq \hat{\Gamma}_{23}\Gamma_{13}[\Gamma_{11} \vdots \Gamma_{12}]$, $\hat{\Gamma}_{23}$ = Toep $(D_{\ddot{e}}) \in R^{n_2 x n_h}$ and $x=(g^T, h^T)^T$. If $\deg(D_e) > 0$, $\hat{\Gamma}$ is an rectangular matrix. In this case the approximate equation (14) is usually overdetermined and there exists no non-zero solution x. Therefore, instead of (14), consider the generalised eigenvalue problem:

$$\Gamma_b^T \; \Gamma_a\, x \; = \lambda(\Gamma_b^{\;T}\; \Gamma_b)\; x \tag{15}$$

This is a necessary condition for (14). If A_r and A_q are coprime, the matrix $\Gamma_b^T \Gamma_b$ is invertible and the problem becomes the usual eigenvalue problem:

$$(\Gamma_b^T \Gamma_b)^{-1} \Gamma_b^T \Gamma_a x = \lambda x \tag{16}$$

Numerical examples show that the largest absolute value of λ gives a good approximation of the optimal λ and the corresponding eigenvector gives a good approximation of the solution of (14). This solution can be used as the starting point of a Newton Raphson method to solve the nonlinear equation:

$$B_q^* B_q A^{-*} A^- H^* H + B_r^* B_r B^{-*} B^- G^* G = \lambda^2 (A_r B^- G + A_q A^- H)^* (A_r B^- G + A_q A^- H) \quad (17)$$

as proposed in [2].

These results are used to construct the following algorithm.

Algorithm 2:

Step 1) solve the eigenvalue problem (15) or (16) and obtain an approximate solution of (12).

Step 2) Solve (17) by a Newton Raphson method with the approximate solution as the starting point.

4. Main Result

In step 2 of Algorithm 2, the linearity of the equations (5) and (6) with respect to G, H, and F^- is not utilized as the result of solving the quadratic order equation (17) instead of (5), (6), and (7). Since the linear equation is a sufficient condition for the quadratic equation, it is expected that an algorithm based on the linear equation (5), (6), and (7) will be better than the Newton Raphson method based on (17). In this section a new algorithm to obtain the exact solution is given by extending the idea of the approximation used in step 1 of Algorithm 2 and utilizing the linearity.

Since A_σ is the function of λ from (7), denote A_σ as $A_\sigma(\lambda)$. The equation (13) is a first order approximation of $A_\sigma(\lambda)$ at $\lambda = \infty$. A first order approximation of $A_\sigma(\lambda)$ at a finite value of $\lambda = \lambda_o$ is

$$A_\sigma(\lambda) \cong A_\sigma(\lambda_o) + (\lambda - \lambda_o) \frac{dA_\sigma}{d\lambda}\bigg|_{\lambda=\lambda_o} \tag{18}$$

where $A_\sigma(\lambda_o)$ is the spectral factor of $D_e^* D_e \lambda^2 - B_q^* B_q B_r^* B_r$ and $dA_\sigma/d\lambda$ at $\lambda = \lambda_o$ is given as the solution of the polynomial equation:

$$\left(\frac{dA_\sigma^*}{d\lambda}\right) A_\sigma(\lambda_o) + A_\sigma^*(\lambda_o)\left(\frac{d}{d\lambda} A_\sigma\right) = 2\lambda_o D_e^* D_e \tag{19}$$

where $dA_\sigma/d\lambda$, $A_\sigma(\lambda_o)$, $D_e \in P(z^{-1})$. Denote these polynomials as

$$A_\sigma(\lambda_o) = a_o + a_1 z^{-1} + \ldots + a_n z^{-n} \tag{20}$$

$$\frac{d}{d\lambda} A_\sigma = v_o + v_1 z^{-1} + \ldots + v_n z^{-n} \tag{21}$$

$$D_e^* D_e = d_n z^{-n} + \ldots + d_1 z^{-1} + d_o + d_1 z^1 + \ldots + d_n z^n \tag{22}$$

Then the equation (19) is equivalent to the linear algebraic equation:

$$\Sigma v = 2 \lambda_o d \tag{23}$$

where $v = (v_o, v_1, \ldots, v_n)^T$, $d = (d_o, d_1, \ldots, d_n)^T$,

$\Sigma = \Sigma_1 + \Sigma_2$

$$\Sigma_1 = \begin{bmatrix} 0 & & & a_0 \\ & & \cdot & a_1 \\ & \cdot & \cdot & \vdots \\ a_o & a_1 & \cdots & a_n \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} a_n & & & 0 \\ a_{n-1} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_0 & \cdots & a_{n-1} & a_n \end{bmatrix}$$

The coefficients of $dA_\sigma/d\lambda$ at $\lambda=\lambda_o$ are given by the solution of (23).

Depending upon the approximation (18), the matrix Γ_{23} of (12) is approximated as:

$$\Gamma_{23} \cong \Theta_1 + (\lambda - \lambda_0)\Theta_2 \tag{25}$$

where $\Theta_1 = \text{Toep}(A_\sigma(\lambda_0)) \in R^{n_2 x n_h}$ and $\Theta_2 = \text{Toep}(dA_\sigma/d\lambda|_{\lambda=\lambda_0}) \in R^{n_2 x n_h}$. By the substitution of (25) into (12), the approximation of (12) becomes

$$(\Gamma_c - \lambda\Gamma_d)x = 0 \tag{26}$$

where $\Gamma_c = [\Gamma_{21} \vdots \Gamma_{22}] - (\Theta_1 - \lambda_0\Theta_2)\Gamma_{13}[\Gamma_{11} \vdots \Gamma_{12}]$ and $\Gamma_d = \Theta_2\Gamma_{13}[\Gamma_{11} \vdots \Gamma_{12}]$.

A better approximation of the solution of (12) will be obtained by solving the generalized eigenvalue problem:

$$\Gamma_d^T\Gamma_c\, x = \lambda\Gamma_d^T\Gamma_d\, x \tag{27}$$

These proposals are formulated as a new algorithm:

Algorithm 3:

Step 1) Set k=0 and λ_0.

Step 2) Obtain $A_\sigma(\lambda_k)$ by the spectral factorization of $D_e^* D_e \lambda^2 - B_q^* B_q B_r^* B_r$.

Step 3) Obtain $dA_\sigma/d\lambda$ at $\lambda=\lambda_k$ by solving (23).

Step 4) Solve (27) and obtain λ. Set k=k+1 and $\lambda_k = \lambda$. If $|\lambda_k - \lambda_{k-1}|$ is sufficiently small, the optimal λ is obtained. Othewise go to Step 2.

The optimality of λ obtained by Algorithm 3 can be explained as follows. Equation (26) can be expressed as:

$$([\Gamma_{21} \vdots \Gamma_{22}] - \Theta_1\Gamma_{13}[\Gamma_{11} \vdots \Gamma_{12}])\, x = (\lambda-\lambda_0)\Theta_2\Gamma_{13}[\Gamma_{11} \vdots \Gamma_{12}]x \tag{28}$$

and (27) is obtained by multiplying (26) to the right of Γ_d^T. If $|\lambda-\lambda_0|=0$, clearly the equation (28) agrees with the original equation (12). Therefore when Algorithm 3 converges, the equation (27) becomes a necessary condition for the original equation (12) to be satisfied.

Algorithm 3 needs only the value of λ as the starting point, whereas a Newton Raphson method of Algorithm 2 needs not only the value of λ but also G and H as the starting point. Numerical examples show that Algorithm 3 converges for a wide interval of λ which includes $\lambda=\infty$ in most cases. Therefore the starting point of λ can be obtained by applying Step 1 of Algorithm 2. If Algorithm 3 does not converge for $\lambda=\infty$, a finite λ_0 can be set in Step 1 alternatively. If Algorithm 2 does not converge, Algorithm 2 does not give another starting point and fails to obtain the solution.

5. Numerical Example

Consider the case:

$A^- = (1+2z^{-1})(1-2z^{-1})$, $B^- = z^{-1}(1-4z^{-1})$, $A_q=1$, $B_q=1$

$A_r = 4-z^{-1}$, $B_r = (1+2z^{-1})(1-2z^{-1})$

$\hat{\Gamma}$ is an 8 x 6 matrix and the generalized eigenvalue problem need to be solved for a 6 x 6 matrix. The sequence of λ_k are given in Table 1. Other examples also show that λ_0 may be a good approximation and λ_k will converge only after a few iterations. The Euclidean norm of the following vector is adopted as the index of the accuracy of the solution λ, g, and h. This vector is the difference between the left and right handside of (11) after substituting g, h and f into (11) where f is obtained from (10). In this example this norm becomes $\underline{0.2 \times 10^{-14}}$. The relation between λ and $a_0(\lambda)$, the coefficient of Z^0 of $A_\sigma(\lambda)$, is illustrated in Fig. 2, which shows that (14) is a good approximation for a wide interval of λ around λ_0. The other coefficients also show this tendency. The relation between the optimality index E and λ is illustrated in Fig. 3.

6. Conclusion

An algorithm to solve the polynomial equation in H_∞ optimization problem is proposed. The solution can be obtained by solving generalized eigenvalue problems iteratively. This algorithm is automatic and does not need trial and error of the bisection method which is used in [1], [2]. Numerical examples show that it will converge after only a few iterations. Therefore, the algorithm is much less computationally demanding than these methods.

The algorithm is easy to implement. The matrix used in the eigenvalue problem is directly related to the coefficients of A_r, A_q, A_r, B_q of the weights of the performance index and the unstable factors A^- and B^- of the plant. It is necessary to solve the eigenvalue problem and a linear equation and to factorize one polynomial in each iteration.

From the above numerical advantages, the algorithm can be also used as the online tuning algorithm.

Acknowledgement:

This paper was written while the authors visited the Industrial Control Unit of Department of Electronic and Electrical Engineering. University of Strathclyde, Glasgow, Scotland, U.K.

References:

[1] B.A. Francis and J.C. Doyle, Linear Control Theory with an H_∞ Optimality Criterion, Systems and Control Group Report 8501, Dept. of Electrical Engineering, University of Tronto, October, 1985 (to appear SIAM J. Control).

[2] H. Kwakernaak, Minimax frequency domain performance and robustness optimization of linear feedback systems, IEEE Trans. Automat Contr AC-30 (1985) pp. 994-1004.

[3] M. Saeki, Polynomial approach to H_∞ optimal control problem for a discrete time system, Report ICU/103/1986, Industrial Control Unit, University of Strathclyde. submitted for publication.

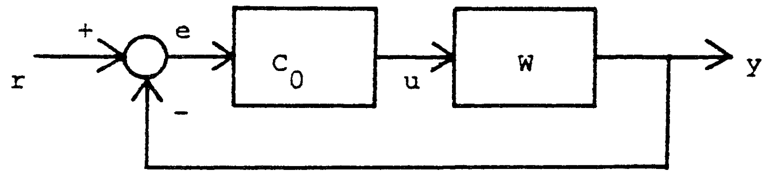

Fig.1 Feedback system

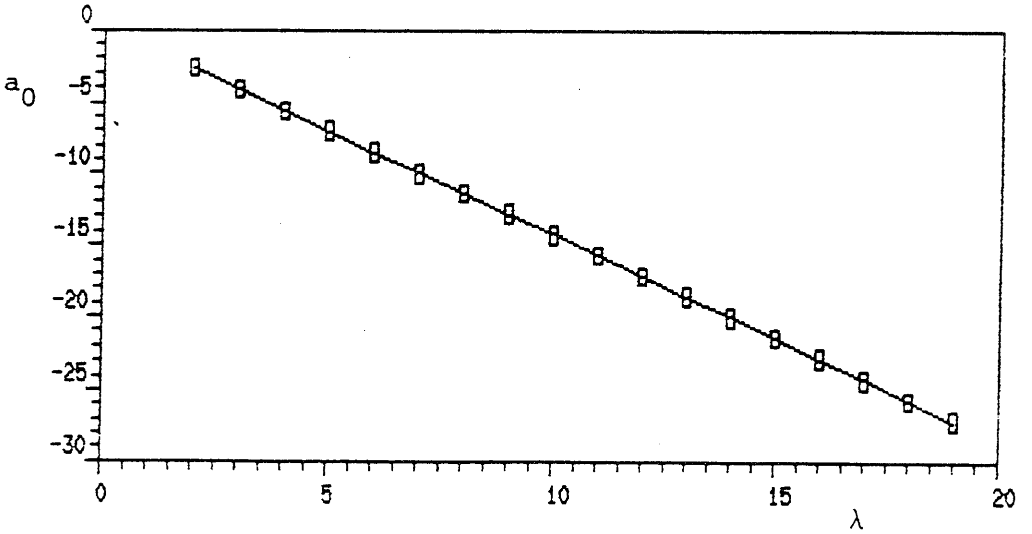

Fig.2 The relation between the coefficient of A_σ and λ

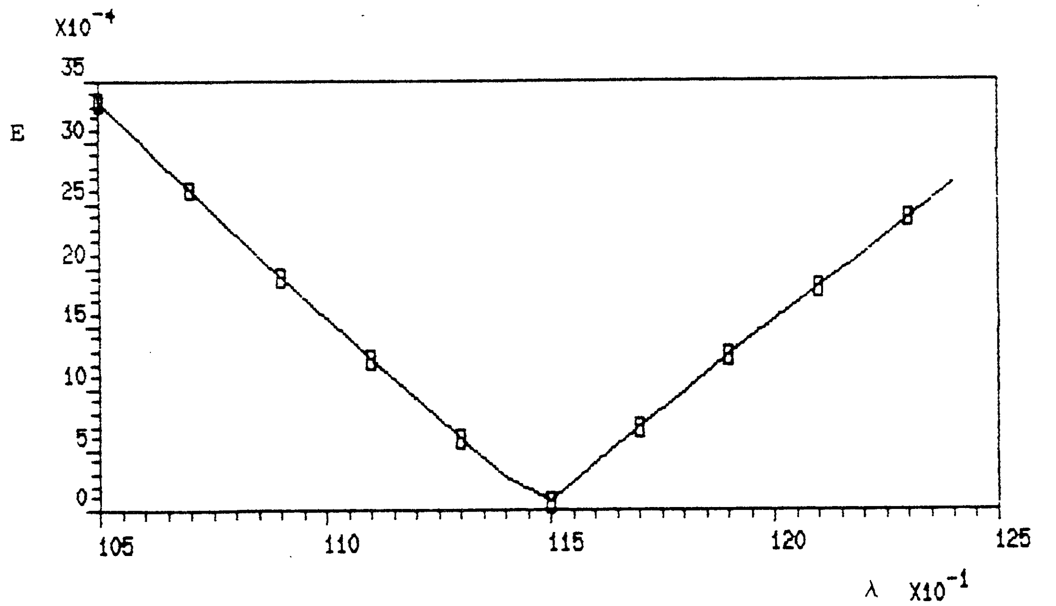

Fig.3 The relation between the optimality index E and λ

λ_0	-11. 459 147 543 952 44
λ_1	-11. 482 920 866 057 22
λ_2	-11. 482 920 968 751 89
λ_3	-11. 482 920 968 751 89

Table 1 The sequence of λ_k

INSTITUTE OF INFORMATION SCIENCES AND ELECTRONICS
UNIVERSITY OF TSUKUBA
SAKURA-MURA, NIIHARI-GUN, IBARAKI 305 JAPAN

REPORT DOCUMENTATION PAGE

REPORT NUMBER
ISE-TR-86-61

TITLE

NUMERICAL ALGORITHM FOR SOLVING A POLYNOMIAL EQUATION IN H_∞ OPTIMIZATION PROBLEM

AUTHOR(S)

Masami Saeki and Eddie Kornegoor

REPORT DATE
December 15, 1986

NUMBER OF PAGES
12

MAIN CATEGORY
OPTIMAL CONTROL THEORY

CR CATEGORIES

KEY WORDS
Numerical algorithm, H_∞ optimization, polynomial equation, Eigen value problem, discrete time system

ABSTRACT

A new numerical algorithm is proposed for solving a polynomial equation arising in the H_∞ optimization problem. The solution is obtained by solving eigenvalue problems iteratively. The algorithm is not computationally demanding and is easy to implement. Numerical examples show that the solution can be obtained after only a few iterations.

SUPPLEMENTARY NOTES