Bull. Korean Fish. Tech. Soc. 19(1), 1983 漁業技術 19(1), 1983

Optimum Approximation of Linear Time-Invariant Systems by Low-Order Models

Sang-bong KIM*

A method is given for obtaining low-order models for a linear time-invariant system of high-order by minimizing a functional of the reduction error between the output response of the original system and the low-order model.

The method is based on the Åström's algorithm for the evaluation of complex integrals and the conjugate gradient method of Fletcher-Reeves. An example illustrating the application of this method is given for approximation of a 4-th order system to be used in the load frequency control of generator systems.

Introduction

The need for methods of optimum approximation of high-order systems by low-order models has been recognized for some time.

The incentiveness for obtaining a simplified system model arises when one is confronted with a state-space representation of a system with many state-variables but only a few output variables.

A complex dynamic system is frequently described by a high-order differential equation. When such many systems are interconnected, the resulted system size may be too large to be conveniently handled, even by a large computer.

A typical example of this situation occurs in the dynamic-stability studies of modern interconnected power system is defined as the stability under infinitesimal disturbances with the action of the resulted devices taken into account[1].

Under dynamic conditions, the system equations are linear, but the total number of differential equations that describe the system performance increase rapidly with the increase in the number of interconnected machines. Therefore, the necessity for techniques to produce low-order model equivalents of high-order systems is apparent.

In the recent years, a considerable attention has been given to the problem of approximating a linear time-invariant system of high-order by a model having a low-order. The time-domain methods of system simplification are usually based on the neglect of the non-dominant eigenvalues of the system or on the minimization of a functional of error between the output responses of the reduced model and original system[2]. On the other hand, the frequency-domain methods of system simplification usually consist of the determination of a transfer function whose frequency response is close to that of the system. A frequency-domain method of simplification based on the expansion of the system transfer function and ignoring some of the quotients is also available[2),3),4].

Sinha's method[5], in which an optimum low-order model is obtained with respect to any specified criterion, is based on the pattern-search algorithm of Hooke and Jeeves. But these methods require much computer time and the complicate matrix equations to be solved.

Therefore, the object of this paper is to develop a more general method, resulting in providing an optimum low-order model is specified, the proposed method is better suited for modelling

* Dongeui Technical Junior College

high-order systems in the points that this method requires less computer time than the case based on the criterion of optimization and does not need the complex matrix equations to be solved.

Problem statement

Considering that a linear time-invariant system of high-order is described by the n-th order differential equation,

$$\frac{d^n y(t)}{dt^n}+a_1\frac{d^{n-1}y(t)}{dt^{n-1}}+\cdots+a_{n-1}\frac{dy(t)}{dt}+a_n y(t)$$
$$=b_0\frac{d^n r(t)}{dt^n}+b_1\frac{d^{n-1}r(t)}{dt^{n-1}}+\cdots+b_{n-1}\frac{dr(t)}{dt}+b_n r(t) \quad \cdots\cdots(1)$$

where $y(t)$ is the output variable and $r(t)$ is the input.

Considering that a low-order model is descibed by the r-th order differential equation,

$$\frac{d^r y_r(t)}{dt^r}+a_1{}^r\frac{d^{r-1}y_r(t)}{dt^{r-1}}+\cdots+a^r{}_{r-1}\frac{dy_r(t)}{dt}+a_r{}^r y_r(t)$$
$$=b_0{}^r\frac{d^r r(t)}{dt^r}+b_1{}^r\frac{d^{r-1}r(t)}{dt^{r-1}}+\cdots+b^r{}_{r-1}\frac{dr(t)}{dt}+b_r{}^r r(t) \quad \cdots\cdots(2)$$

In the frequency domain, taking the Laplace transform on both sides of Eq. (1) and Eq. (2), we can write as rational fraction polynomials

$$G(s)=\frac{Y(s)}{R(s)} \text{ or } G(s)=\frac{B(s)}{A(s)} \quad \cdots\cdots(3)$$

where

$$A(s)=s^n+a_1s^{n-1}+\cdots+a_{n-1}s+a_n \quad \cdots\cdots(4)$$
$$B(s)=b_0s^n+b_1s^{n-1}+\cdots+b_{n-1}s+b_n \quad \cdots\cdots(5)$$

and

$$G_r(s)=\frac{Y_r(s)}{R(s)} \text{ or } G_r(s)=\frac{B_r(s)}{A_r(s)} \quad \cdots\cdots(6)$$

where

$$A_r(s)=s^r+a_1{}^r s^{r-1}+\cdots+a^r{}_{r-1}s+a_r{}^r \quad \cdots\cdots(7)$$
$$B_r(s)=b_0{}^r s^r+b_1{}^r s^{r-1}+\cdots+b^r{}_{r-1}s+b_r{}^r \quad \cdots\cdots(8)$$

The linear time-invariant system in Eq. (1) can be described as follows by the dynamic equations;

$$\boldsymbol{x}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\cdot\boldsymbol{r}(t) \quad \cdots\cdots(9)$$
$$\boldsymbol{y}(t)=\boldsymbol{H}\boldsymbol{x}(t) \quad \cdots\cdots(10)$$

where $\boldsymbol{x}(t)=n\times 1$ state vector
$\boldsymbol{y}(t)=m\times 1$ output vector
$\boldsymbol{A}=n\times n$ matrix
$\boldsymbol{H}=m\times n$ matrix
$\boldsymbol{B}=n\times p$ matrix
$\boldsymbol{r}(t)=p\times 1$ input vector

The low-order model in Eq. (2) can be described as follows by the r-th order dynamic equations;

$$\boldsymbol{x}_r(t)=\boldsymbol{A}_r\boldsymbol{x}(t)+\boldsymbol{B}_r\boldsymbol{r}(t) \quad \cdots\cdots(11)$$
$$\boldsymbol{y}_r(t)=\boldsymbol{H}_r\boldsymbol{x}_r(t) \quad \cdots\cdots(12)$$

where $\boldsymbol{x}_r(t)=r\times 1$ state vector
$\boldsymbol{y}_r(t)=m\times 1$ output vector
$\boldsymbol{A}_r=r\times r$ matrix
$\boldsymbol{B}_r=r\times p$ matrix
$\boldsymbol{H}_r=m\times r$ matrix

From the state-variable approaches, we can write Eq. (3) and Eq. (6) as

$$Y(s)=G(s)\cdot R(s)=\boldsymbol{H}(s\boldsymbol{I}_n-\boldsymbol{A})^{-1}\boldsymbol{B}R(s)$$
$$=\frac{\boldsymbol{H}[adj(s\boldsymbol{I}_n-\boldsymbol{A})]}{|s\boldsymbol{I}_n-\boldsymbol{A}|}\boldsymbol{B}R(s) \quad \cdots\cdots(13)$$

$$Y_r(s)=G_r(s)\cdot R(s)=\boldsymbol{H}_r(s\boldsymbol{I}_r-\boldsymbol{A}_r)^{-1}\boldsymbol{B}_rR(s)$$
$$=\frac{\boldsymbol{H}_r[adj(s\boldsymbol{I}_r-\boldsymbol{A}_r)]}{|s\boldsymbol{I}_r-\boldsymbol{A}_r|}\boldsymbol{B}_rR(s) \quad (14)$$

where

$\boldsymbol{I}_n$= the n-th order unit matrix
$\boldsymbol{I}_r$= the r-th order unit matrix.

The reduction error in Eq. (10) and Eq. (12) will be defined as

$$\boldsymbol{e}(t)=\boldsymbol{y}(t)-\boldsymbol{y}_r(t) \quad \cdots\cdots(15)$$

To synthetise the optimum low-order models, the reduction error of Eq. (15) must be minimized. Therefore, we consider that the cost function of the systems with deterministic inputs to be minimized is then

$$J=\int_0^\infty \boldsymbol{e}^2(t)dt \quad \cdots\cdots(16)$$

For several control inputs the general form of the cost function corresponding to Eq. (19) yields

$$J=\int_0^\infty[\boldsymbol{e}^T(t)Q\boldsymbol{e}(t)]dt$$
$$=\int_0^\infty||\boldsymbol{e}(t)||_Q^2\,dt \quad \cdots\cdots(17)$$

where Q is a real symmetric positive semi-definite $m\times m$ matrix.

Taking the Laplace Transform on both sides of Eq. (15), the reduction error is written as

$$E(s)=Y(s)-Y_r(s)=[G(s)-G_r(s)]R(s)$$
$$=[\boldsymbol{H}(s\boldsymbol{I}_n-\boldsymbol{A})^{-1}\boldsymbol{B}-\boldsymbol{H}_r(s\boldsymbol{I}_r-\boldsymbol{A}_r)^{-1}\boldsymbol{B}_r]R(s) \quad \cdots\cdots(18)$$

Using the Parseval's theorem in Eq. (17), it is rearranged

$$J=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}E^T(s)QE(-s)ds$$
$$=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}R^T(s)[G(s)-G_r(s)]^TQ[G(-s)$$
$$-G_r(-s)]R(-s)ds \quad \cdots\cdots(19)$$

The matrix Q is taken as

$$Q=diag[q_1,\ q_2,\ q_3,\cdots,\ q_n] \quad \cdots\cdots(20)$$

where is a square matrix with $a_{ij}=0$ for all $i\neq j$.

And, let's consider that the reduction error is

$$E(s)=[E_1(s),\ E_2(s),\ \cdots,\ E_n(s)] \quad \cdots\cdots(21)$$

Substituting Eq. (20), (21) into Eq. (19), we have

$$J=\sum_{i=1}^{m}q_i\left(\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}E_i(s)E_i(-s)ds\right) \quad \cdots\cdots(22)$$

From the relations of Eq. (3) and Eq. (6), the reduction error $E_i(s)$ can be written as rational polynomial

$$E_i(s)=B_i(s)/A_i(s) \quad \cdots\cdots(23)$$

where

$$A_i(s)=a_0{}^is^n+a_1{}^is^{n-1}+\cdots+a^i{}_{n-1}s+a_n{}^i \quad \cdots\cdots(24)$$
$$B_i(s)=b_1{}^is^{n-1}+\cdots+b^i{}_{n-1}s+b_n{}^i \quad \cdots\cdots(25)$$

Stbstituting Eq. (23) into Eq. (22), we have

$$J=\sum_{i=1}^{m}q_i\left(\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{B_i(s)\cdot B_i(-s)}{A_i(s)\cdot A_i(-s)}ds\right)$$
$$=\sum_{i=1}^{m}q_i\cdot J_i \quad \cdots\cdots(26)$$

where

$$J_i=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{B_i(s)\cdot B_i(-s)}{A_i(s)\cdot A_i(-s)}ds \quad \cdots\cdots(27)$$

If the polynomial $A(s)$ in Eq. (3) has all roots in the left half plane, then the cost function becomes

$$J=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}R(s)\left[\frac{B(s)}{A(s)}-\frac{B_r(s)}{A_r(s)}\right]\left[\frac{B(-s)}{A(-s)}\right.$$
$$\left.-\frac{B_r(-s)}{A_r(-s)}\right]R(-s)ds \quad \cdots\cdots(28)$$

Assumption of low-order model

Let's consider that the new state-variable $z_r(t)$ with respect to x_r is described by

$$z_r(t)=Tx_r(t) \quad \cdots\cdots(29)$$

where T is $r\times r$ nonsingular matrix with inverse matrix.

Rearranging Eq. (28), we have

$$x_r(t)=T^{-1}z_r(t) \quad \cdots\cdots(30)$$

where T^{-1}=inverse matrix of T.

Substituting Eq. (30) into Eq. (11) and (12), we obtain the following;

$$z_r(t)=TA_rT^{-1}z_r(t)+TB_rr(t) \quad \cdots\cdots(31)$$
$$y_r(t)=HT^{-1}z_r(t) \quad \cdots\cdots(32)$$

The transfer function of systems with dynamic equations (31) and (32) can be expressed as

$$G_T(s)=H_rT^{-1}(sI_r-TA_rT^{-1})^{-1}TB_r$$
$$=H_rT^{-1}[T(sI_r-A_r)T^{-1}]^{-1}TB_r$$
$$=H_r(sI_r-A_r)^{-1}B_r \quad \cdots\cdots(33)$$

Eq. (33) indicates that the same outputs can be obtained with respect to the same inputs, therefore Eq. (33) is equivalent with the transfer function of the low-order model given by Eq. (11) and (12).

If the controllability and observability are supposed, then this system has only a transfer function.

Problem solution

1. Cost function

To compute Eq. (26) we are led to the problem of evaluating the integral such as

$$I=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{B(s)\cdot B(-s)}{A(s)\cdot A(-s)}ds \quad \cdots\cdots(34)$$

where A and B are polynomials with real coefficients

$$A(s)=a_0s^n+a_1s^{n-1}+\cdots+a_{n-1}s+a_n \quad \cdots\cdots(35)$$
$$B(s)=b_1s^{n-1}+\cdots+b_{n-1}s+b_n \quad \cdots\cdots(36)$$

The integral Eq. (34) can of course be evaluated in a straight-forward manner using residue calculus. It turn out, however, that the general formulas are not practical to handle for systems of high-order. For this purpose we will present recursive formulas for the evaluation of the integral Eq. (34) which are convenient both for hand and machine calculations. A decomposition of the polynomial $A(s)$ of Eq. (24) into odd and even terms is

$$A(s)=\bar{A}(s)+\tilde{A}(s) \quad \cdots\cdots(37)$$

where

$$\bar{A}(s)=a_0s^n+a_2s^{n-2}+\cdots=\frac{1}{2}[A(s)+(-1)^nA(-s)] \quad \cdots\cdots(38)$$

$$\tilde{A}(s)=a_1s^{n-1}+a_3s^{n-3}+\cdots=\frac{1}{2}[A(s)-(-1)^nA(-s)] \quad \cdots\cdots(39)$$

Let's consider that the order of the polynomials $A_k(s)$ and $B_k(s)$ are lower than n-th order as the following;

$$A_k(s)=a_0^ks^k+a_1^ks^{k-1}+\cdots+a_k^k \quad \cdots\cdots(40)$$

$$B_k(s)=b_1^ks^{k-1}+b_2^ks^{k-2}+\cdots+b_k^k \quad \cdots\cdots(41)$$

which are defined recursively from the equations

$$A_k(s)=A_k(s)-\alpha_kS\tilde{A}_k(s) \quad \cdots\cdots(42)$$

$$B_k(s)=B_k(s)-\beta_k\tilde{A}_k(s) \quad \cdots\cdots(43)$$

where

$$\alpha_k=a_0^k/a^{1k} \quad \cdots\cdots(44)$$

$$\beta_k=b_1^k/a_1^k \quad \cdots\cdots(45)$$

and

$$A_n(s)=A(s) \quad \cdots\cdots(46)$$

$$B_n(s)=B(s) \quad \cdots\cdots(47)$$

The polynomials A_{k-1} and B_{k-1} can apparently only be defined if $a\neq 0$.

To compute the integral Eq. (34) we introduce

$$I_k=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{B_k(s)B_k(-s)}{A_k(s)A_k(-s)}ds \quad \cdots\cdots(48)$$

where the polynomials $A_k(s)$ and $B_k(s)$ are defined by Eq. (39) and (40).

If the polynomial $A(s)$ has all roots in the left half plane, then all the polynomials $A_k(s)$, $k=n-1, n-2, \cdots, 0$ have also roots in the left half plane, and all the coefficients a_i^n are positive.

Hence

$$I_k=I_{k-1}+\beta_k^2/2\alpha_k \quad k=1, 2, \cdots, n \quad \cdots\cdots(49)$$

$$I_0=0, \ I_n=I \quad \cdots\cdots(50)$$

where

$$a_i^{k-1}=\begin{cases} a^k{}_{i+1} & i: \text{even}, \ i=0, 1, 2, \cdots, k-1 \\ a^k{}_{i+1}-\alpha_ka^k{}_{i+2} & i: \text{odd}, \ \alpha_k=a_0^k/a_1^k \end{cases} \quad \cdots\cdots(51)$$

$$b_i^{k-1}=\begin{cases} b^k{}_{i+1} & i: \text{even}, \ i=1, 2, \cdots, k-1 \\ b^k{}_{i+1}-\beta_ka^k{}_{k+1} & i: \text{odd}, \ \beta_k=b_1^k/a_1^k \end{cases} \quad \cdots\cdots(52)$$

These are obtained by identifying coefficients of powers of S in Eq. (42) and (43).

Having obtained the values α_k and β_k, the value of the integral is then given by Eq. (49) as follows;

$$I=\sum_{k=1}^{n}\beta_k^2/2\alpha_k=\sum_{k=1}^{n}(b_1^k)^2/(2a_0^ka_1^k) \quad \cdots\cdots(53)$$

As the computations are defined recursively, it is now an easy matter to obtain a computer algorithm.

Therefore the cost function of Eq. (26) can be easily computed by the flow chart as shown in Fig. 1.

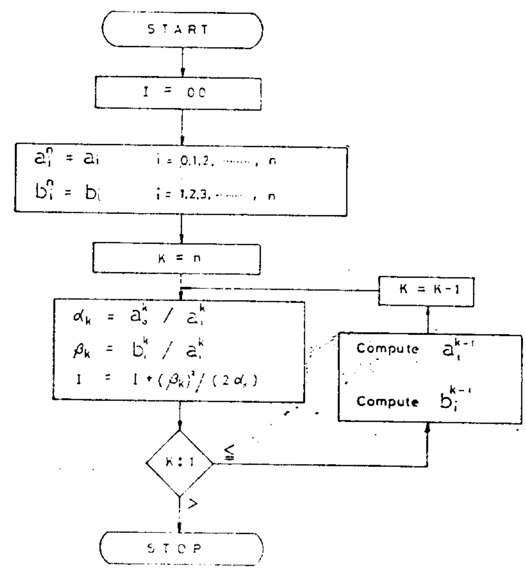

Fig. 1. Flow chart for Åström's algorithm.

If the polynomial $A_i(s)$ has zeros in the right half plane, as the cost function cannot be computed with above algorithm, let it part as the polynomial $\phi_i(s)$ having zeros in the left half plane, and the polynomial $\psi_i(s)$ having zeros in the right half plane. Then the polynomial $A(s)$ yields

$$A_i(s)=\phi_i(s)\psi_i(s) \quad \cdots\cdots(54)$$

The reduction error in Eq. (23) can be written as

$$E_i(s)=B_i(s)/[\phi_i(s)\psi_i(s)] \quad \cdots\cdots(55)$$

Eq. (55) can be rearranged as the following

$$B_i(s)=D_i(s)\phi_i(s)+R_i(s) \quad \cdots\cdots(56)$$

where $D_i(s)$ is the quotient of $B_i(s)/\phi_i(s)$ and $R_i(s)$ is the remainder.

As the condition taking the steady state error with zero, $R_i(s)$ in Eq. (56) must be zero as the following

$$R_i(s) \equiv 0 \qquad \cdots\cdots(57)$$

Now using the condition of Eq. (57), Eq. (56) becomes

$$E_i(s) = D_i(s)/\phi_i(s) \qquad \cdots\cdots(58)$$

As a result, the cost function J_i can be computed with the algorithm of Åström[7),8)], since $B_i(s)$ has only all zeros in the left half plane. Furthermore, the condition having all zeros of the denominator polynomial of $E_i(s)$ in the left half plane is equivalant to

$$\lim_{t\to\infty} e_i(t) = \lim_{s\to 0} sE_i(s) = 0 \qquad \cdots\cdots(59)$$

Therefore, the condition of Eq. (57) equals to the condition that the last values of the output of the original system and the low-order model have the same after a constant time.

2. Optimum method

The problem of approximating high-order systems by low-order models in an optimum manner. as fomulated in th the above, can only be solved for the specipic condition that the cost function of Eq. (26) must be minimized in the constrained condition of Eq. (57).

Since an analytical solution to the problem does not appear feasible, various search techniques may be considered. But in this paper the conjugate gradient method of Fletcher-Reeves[9),10),11)] was selected as a suitable method for this work.

The algorithm proceeds as follows:

1) A starting point is selected.

2) The direction of steepest descent is determined by determining the following direction vector components (normallized form) at the starting point,

$$P_i^{(k)} = \left[\frac{-\partial J}{\partial X_i} \Big/ \left\{\sum_{j=1}^{n}\left(\frac{\partial J}{\partial X_j}\right)^2\right\}^{+}\right]^{(k)}, \quad i=1,2,\cdots,n \qquad \cdots\cdots(60)$$

Where k=0 for the starting point.

3) A one dimensional search is then conducted along the direction of steepest descent utilizing the relation,

$$X_i(\text{new}) = X_i(\text{old}) + \delta_i P_i, \quad i=1,2,\cdots,n \qquad \cdots(61)$$

where δ_i is the distance moved in the P direction. When a minimum is obtained along the direction of steepest descent, a new "conjugate directon" search direction is evaluated at the new point with the following normalized components:

$$P_i^{(k)} = \frac{-\left(\frac{\partial J}{\partial X_i}\right)^{(k)} + \beta^{(k-1)} P_i^{(k-1)}}{\left[\sum_{j=1}^{n}\left(-\left(\frac{\partial J}{\partial X_j}\right)^{(k)} + \beta^{(k-1)} P_j^{(k-1)}\right)^2\right]^{1/2}}, \quad i=1,2,\cdots,n \qquad \cdots(62)$$

$$\beta^{(k-1)} = \frac{\sum_{i=1}^{n}\left[\left(\frac{\partial J}{\partial X_i}\right)^{(k)}\right]^2}{\sum_{i=1}^{n}\left[\left(\frac{\partial J}{\partial X_i}\right)^{(k-1)}\right]^2} \qquad \cdots\cdots(63)$$

4) A one dimensional search is the conducted in this direction. When a minimum is found, an overall convergence check is made. If convergence is achieved, the procedure terminates. If convergence is not achieved new "conjugate direction" vector components are evaluated per step 3) at the minimum point from the current one dimensional search.

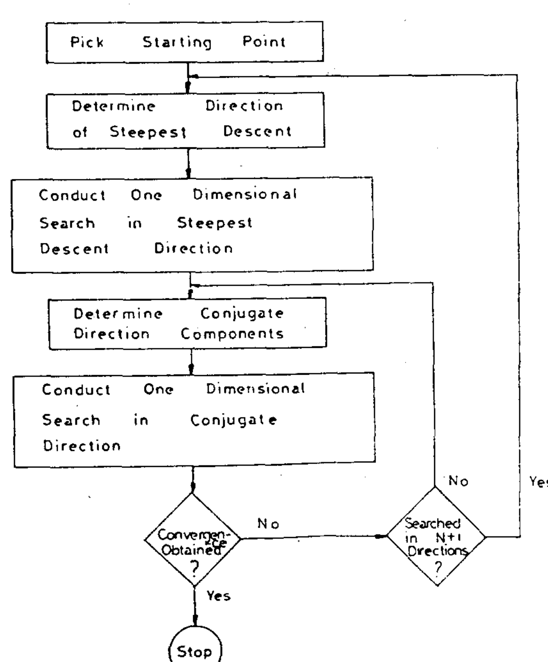

Fig. 2. Logic diagram of optimization algorithm

This process is continued until covergence is achieved or $N+1$ directions have been searc-hed. If a cycle of $N+1$ directions have been completed, a new cycle is started consisting of a steepest descent direction (step 2) and N "conjugate directions" (step3).

A logic diagram illustrating the above procedure is given in Fig. 2.

Since the partial derivatives of the cost function with respect to the parameters $X(1)$, $X(2), \cdots$, $X(n)$ of low-order model can not be analytically obtained, we can compute with the following approximation

$$\partial J[X(1), X(2), \cdots, X(n)]/\partial X(i)$$
$$\fallingdotseq [J(X(1), X(2), \cdots X(i)+\frac{e}{2}, X(i+1)\cdots)$$
$$-J(X(1), X(2), \cdots X(i)-\frac{e}{2}, X(i+1), \cdots X(n))]/e \quad \cdots\cdots(64)$$

where e is a minimum value that the difference approximation can be sufficiently obtained.

A flow chart of the program for obtaining the optimum parameters of low-order model is shown in Fig. 3. And in this paper, all computation was taken on Perkin-Elmer 3220.

Numerical examples

The method of system reduction outlined in above sections will now be applied to the simpl-ification of a 4-th order system to be used in the load frequency control of generator systems.

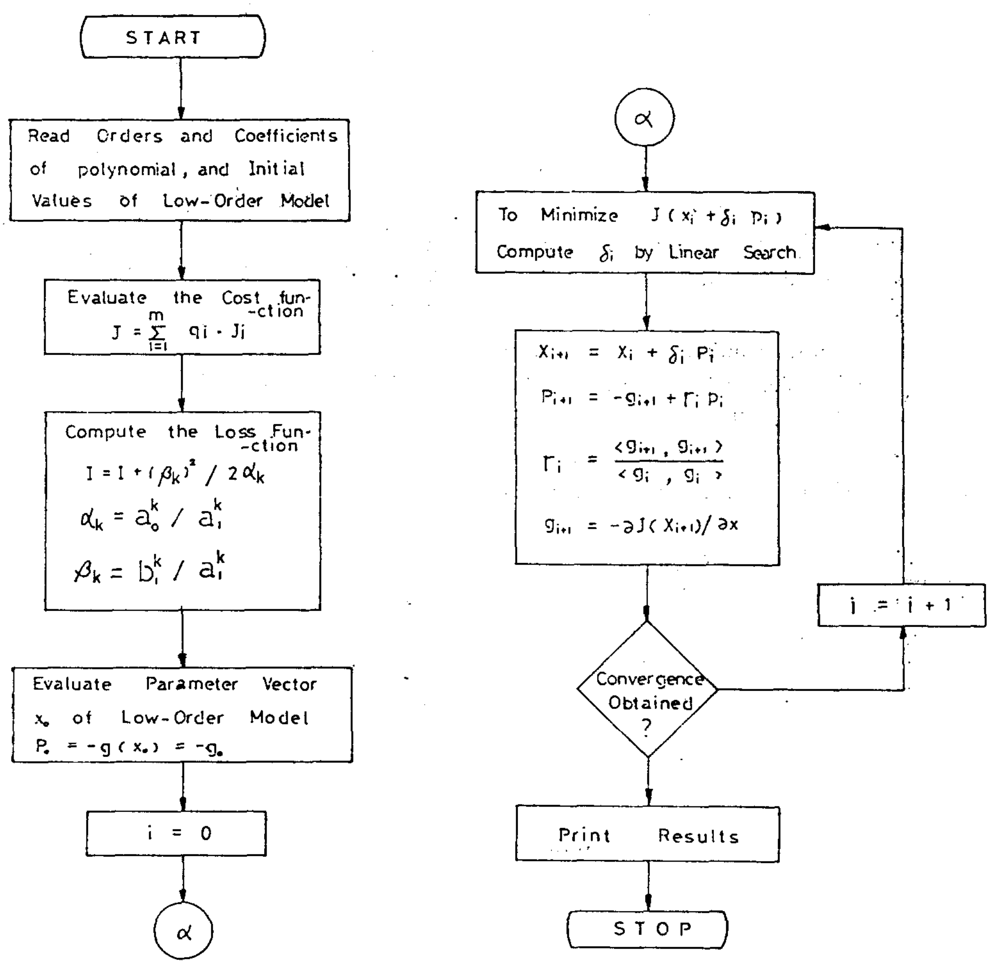

Fig. 3. Flow chart for computing optimum parameters of low-order model.

The output response of the original system is then compared with that of the low-order model obtained by this method.

The dynamic equation of the original system are

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2.7 & -6.7 & -5.7 & -4.3 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 8.0 \\ -31.3 \end{bmatrix} \quad \cdots(65)$$

$$y(t) = [1000] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} \quad \cdots\cdots(66)$$

For the two resposes of impulse and unit step inputs, the corresponding models to be reduced are given as follows;

(1) First order model

$$G(s) = 1/(a_0 s + 1) \quad \cdots\cdots(61)$$

where the parameter to be optimized is a.

(2) Second order model

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} r(t) \quad \cdots(68)$$

$$y(t) = [0\ 1] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \quad \cdots\cdots(69)$$

Where the parameters to be optimized are a_0, a_1, b_0, b_1.

The parameters of the low-order models and the cost functions obtained by this method are shown in the Table 1 and the output responses of the original system and low-order models obtained by Runge-Kutta's method[11] are shown in Fig. 4 and Fig. 5.

It is seen that as expected, the output responses of the original systems and the low-order model with respect to impulse and unit step inputs have approximatly the same values after a constant time.

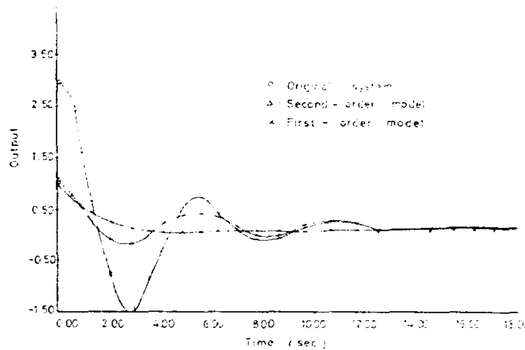

Fig. 4 Comparison of out-put responses for impulse input.

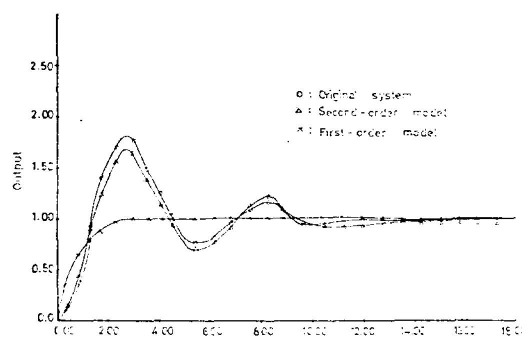

Fig. 5 Comparison of out-put responses for unit step input.

Table 1. Values of the parameters for low-order models and the peformance indexes for impulse and unit step inputs

Low-order models	Inputs	Parameters of low-order models				Performance indexes(J^0)
		a_0	a^1	b_0	b_1	
First-order model	Impulse	1.4230	—	—	—	1.71893
	Unit step	0.6401	—	—	—	1.39509
Second-order model	Impulse	1.5133	0.5423	1.212	-0.2831	0.01166
	Unit step	1.3170	0.3904	1.3156	0.0595	0.18399

Conclusions

A technigue for deriving a low-order equivalent model that approximates closely to the performance of a high-order linear time-invariant system has been proposed.

The Åström's algorithm and the steepest descent method have been used to determine models which are optimum in the sense that the square integral of the reduction error between the response of the original system and the low-order model, is minimized and the last response is well corresponded with the low-order models.

This method has been applied to determining first and second order models of 4-th order system to minimize the cost function by computer simulation.

This method appears most useful in the following advantages over previos method;

1. A funtional of the error between the original and the low-order model outputs is minimized.
2. The low-order model can be uniformly obtained for the inputs described by rational polynomial with respect to the Laplace operators.
3. Eigenvectors of the original system matrix A and the complex matrix equations are not necessary in being solved.
4. The algorithm obtaining the low-order model is very simply, and the memory capacity and the computing time taken for computation become less.

References

1. Davison, E.J .: "A method of simplifying linear dynamic systems", IEEE Trans., 1966, Ac-11, pp. 93-101.
2. Wilson, D,A.: "Optimum solution of model reduction problem", Proc. IEE, 1970, 117, (4), pp. 1161-1165.
3. Sanathanan, C,K, and Koerner, J.: "Transfer function synthesis as a ratio of two complex polynomials", IEEE Trans., 1963, Ac-8, pp. 56-58.
4. Chen, C.F., and Shieh, L.S.: "A noval approach to linear model simplification", JACC Preprinted, 1968, pp. 454-461.
5. Sinha, N.K., Bereznai, G.T.,: "Optimization of high-order systems by low-order models", INT. J. Control, 1971, 14-5, pp. 951-959.
6. Akin, J.E.: "A note on control system model simplification", INT. J. Control, 1971, 14-5, pp, 989-973.
7. Åström, K.J., Jury, E. I., and Agniel, R. G., "numerical method for the evaluation of complex integrals", IEEE Trans. Ac, 1970.
8. Åström, K. J.: "Introducten to stochastic control theory", Academic Press, 1970.
9. James, K., and Joe, H,M,: "Optimization techniques with FORTRAN", McGRAW-HILL Book Company, 1973.
10. Lawrene, H.: "Gradient optimization and nonlinear control", JOHN WILEY & SONS, 1976.
11. Conte, S.D. and Boor, C.D.: "Elementary numerical analysis an algorithmic approach", McGRAW-HILL Book Company, 1980.

線型時不變系의 低次元모델에 의한 最適近似化

김 상 봉

要 約

線型時不變系에 있어서, 高次의 原시스템과 近似化되는 低次元모델 사이의 出力應答誤差의 제곱의 積分을 評價函數로 取하여, 이 값을 最小로 하는 低次元모델의 파라미터를 求하는 하나의 近似化法을 提案하였다. 本 方法은 Åström의 알고리즘과 Fletcher-Reeves의 共役傾似法을 利用하여 低次元파라미터를 求한 것이다, 그 計算例로서는 發電機系統의 負荷周波數制御에 利用되고 있는 4次元시스템을 低次元 近似化시켜서, 그 應答들을 比較·考察하였다.