REDESIGN OF THE STABILIZED PITCH CONTROL SYSTEM OF A SEMI-ACTIV-ETC(U)

UNCLASSIFIED
Report DAAD 40-78-C-0017

REDESIGN OF THE STABILIZED PITCH CONTROL SYSTEM OF A SEMI-ACTIVE TERMINAL HOMING MISSILE SYSTEM

AD A089171

Leang-San Shieh
Department of Electrical Engineering
University of Houston
Houston, Texas 77004

20 April 1979

Final Report for period 8 December 1977 - 31 January 1979

Prepared for
U. S. ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND
Redstone Arsenal, Ala. 35809
REDESIGN OF THE STABILIZED PITCH CONTROL SYSTEM OF A SEMI-ACTIVE TERMINAL HOMING MISSILE SYSTEM.

Leang-San Shieh
Department of Electrical Engineering
University of Houston
Houston, Texas 77004

Final Report, 8 Dec 77 - 31 Jan 79

Prepared for
U. S. ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND
Redstone Arsenal, Ala. 35809
Redesign of the stabilized pitch control system of a semi-active terminal homing missile system

Final Report for period 8 December 1977 - 31 January 1979
20. Abstract

A high-order stabilization filter was formerly designed to stabilize an unstable pitch control system of a terminal homing missile system. In this report, a new dominant-data matching method is presented to redesign the high order stabilization filter. Using this new method several reduced order filters are obtained. As a result, the cost of implementation is reduced and the reliability is increased. An algebraic method is also applied to redesign the stabilization filter so that the performance of the redesigned pitch control system is improved. In addition, the proposed dominant-data matching method can be applied to determine a reduced order model of a high order system. Unlike the reduced order models obtained by most existing model reduction methods, the reduced order model mentioned above has the exact assigned frequency-domain specifications of the original system. The dominant-data matching method can also be applied to identify any practical system.
ABSTRACT

A high-order stabilization filter was formerly designed to stabilize an unstable pitch control system of a terminal homing missile system. In this report, a new dominant-data matching method is presented to redesign the high order stabilization filter. Using this new method several reduced order filters are obtained. As a result, the cost of implementation is reduced and the reliability is increased. An algebraic method is also applied to redesign the stabilization filter so that the performance of the redesigned pitch control system is improved. In addition, the proposed dominant-data matching method can be applied to determine a reduced order model of a high order system. Unlike the reduced order models obtained by most existing model reduction methods, the reduced order model mentioned above has the exact assigned frequency-domain specifications of the original system. The dominant-data matching method can also be applied to identify any practical system.
TABLE OF CONTENTS

TABLE OF CONTENTS ........................................ iv
LIST OF FIGURES .............................................. v

CHAPTER
I. INTRODUCTION ............................................ 1
II. THE DOMINANT-DATA MATCHING METHOD ................. 10
III. THE INITIAL GUESS ..................................... 17
IV. SIMPLIFICATION OF THE EXISTING STABILIZATION FILTER . 41
V. REDESIGN OF THE STABILIZATION FILTER BY AN ALGEBRAIC METHOD ........................................ 53
VI. CONCLUSION ............................................. 70
REFERENCES .................................................. 72

APPENDIX
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>DESCRIPTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 1</td>
<td>The Block Diagram of the Existing Control System</td>
<td>1</td>
</tr>
<tr>
<td>Fig. 2</td>
<td>The Nyquist Plots of Various Open-Loop Transfer Functions</td>
<td>5</td>
</tr>
<tr>
<td>Fig. 3</td>
<td>Time Responses of Original and Third Order Reduced Models</td>
<td>40</td>
</tr>
<tr>
<td>Fig. 4</td>
<td>Time Responses of Various Models</td>
<td>49</td>
</tr>
<tr>
<td>Fig. 5-1</td>
<td>The Block Diagram of a Redesigned System with Fixed Configuration Compensators</td>
<td>57</td>
</tr>
<tr>
<td>Fig. 5-2</td>
<td>The Modified Block Diagram of the Redesigned System</td>
<td>57</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

This report deals with the simplification and realization of a stabilization filter designed to stabilize the pitch control system of an unstable semi-active terminal homing missile system [1]. The block diagram of the existing stabilized system is shown in Fig. 1.

![Block Diagram of the Existing Control System](image)

Figure 1. The Block Diagram of the Existing Control System

The overall transfer function of the existing system shown in Fig. 1 is given by

\[ T_e(s) = \frac{F_{stab}(s)T_{act}(s)T_{miss}(s)}{1 + F_{stab}(s)T_{act}(s)T_{miss}(s)H_g(s)} \]

\[ \Delta = \frac{F_{stab}(s)G_0(s)}{1 + F_{stab}(s)G_0(s)H_g(s)} \]

\[ \Delta = \frac{G_e(s)}{1 + G_e(s)H_g(s)} \]  \( (1.1) \)

where
The transfer function of the actuator and the air frame dynamics of the missile system.

\[ G_0(s) = \text{The open loop transfer function of the original pitch control system iff} \quad F_{\text{stab}}(s) = 1 \text{ and } H_g(s) = 1. \]

\[ = \begin{bmatrix} T_{\text{act}}(s) \\ T_{\text{miss}}(s) \end{bmatrix} \]

\[ = \begin{bmatrix} 26937.9(s+65)(s+1500) \\ (s+87.9+j95.5)(s+112.5)(s+1385) \end{bmatrix} \begin{bmatrix} 12.04(s+0.1933) \\ s(s-2.921)(s+3.175) \end{bmatrix} \]

\[ = \frac{324332.316(s+0.1933)(s+65)(s+1500)}{s(s-2.921)(s+3.175)(s+87.9+j95.5)(s+112.5)(s+1385)} \] (1.3)

\[ G_e(s) = F_{\text{stab}}(s) G_0(s) H_g(s) \]

= The open loop transfer function of the existing stabilized system.

\[ H_g(s) = \text{Transfer function of the gyro.} \]

\[ = 1, \text{ as the rate gyro is not present in the system.} \]

After substituting \( H_g(s) = 1 \) and Eqn. (1.2) and (1.3) into Eqn. (1.1) it becomes
From Fig. 1 as well as from Eqn. (1.2) it can be noticed that the existing stabilization filter $F_{\text{stab}}(s)$ is a fourth-order series compensator with two pairs of complex poles. $F_{\text{stab}}(s)$ is not a positive real function and hence cannot be synthesized with passive elements. The objective of
this report is to develop computer-aided design methods for redesigning
the stabilization filter in a simpler form so that the cost of imple-
mentation of the compensator can be reduced and at the same time the
performance of the redesigned pitch control system remains the same as
that of the existing pitch control system.

Nyquist plots of $G_e(s)$ and $G_0(s)$ are shown in Fig. 2. The
dominant frequency-response data of $G_e(s)$ are given below:

i) The real and imaginary parts of $G_e(s)$ at $s = j\omega = jo$ are
$$\text{Re} \left[ G_e(j\omega) \right] = -2.103817 \quad \text{and} \quad \text{Im} \left[ G_e(j\omega) \right] = \infty$$
or $$T_e(j\omega) = 1$$

ii) The gain margin $G_{em}$ of this system $G_e(j\omega)$ is
$$G_{em} = \left| \frac{1}{G_e(j\omega_{em})} \right| = \left| \frac{1}{\text{Re}[G_e(j\omega_{em})]} \right| = \left| \frac{1}{-1.5} \right|$$

where the phase-crossover frequency $\omega_{em}$ is given by
$$\omega_{em} = 1.9 \text{ rad/sec} \quad \text{such that} \quad \angle G_e(j\omega_{em}) = -180^\circ$$

The equivalent real and imaginary parts of $G_e(j\omega_{em})$ at $\omega_{em} = 1.9$
rad/sec. are
$$\text{Re}[G_e(j\omega_{em})] = -1.507944 \quad (2.4)$$
$$\text{Im}[G_e(j\omega_{em})] = -0.006490205 \quad (2.5)$$

iii) The phase margin $\Phi_{em}$ of the system $G_e(j\omega)$ is
\[ \phi_{em} = 180^\circ + \angle G_e(j\omega_{ec}) = 5.7787^\circ \]  
(2.6)

where the gain cross-over frequency \( \omega_{ec} \) is given by \( \omega_{ec} = 3.2 \)
rad/sec so that

\[ |G_e(j\omega_{ec})| = 1 \]  
(2.7)

The equivalent real and imaginary parts of \( G_e(j\omega) \) at \( \omega = \omega_{ec} = 3.2 \) rad/sec. are

\[ \text{Re}[G_e(j\omega_{ec})] = -0.9939143 \]  
(2.8)

\[ \text{Im}[G_e(j\omega_{ec})] = -0.09547478 \]  
(2.9)

The frequency response data at \( \omega = 0 \) in (2.1) indirectly indicates the steady-state value of the unit step response of the system \( T_e(s) \). The data at \( \omega = \omega_{en} \) and \( \omega = \omega_{ec} \) in Eqn. (2) represent two control specifications [2]: gain margin and phase margin. These control specifications characterize the relative stability and the transient response of the existing stabilized system. The dominant frequency response data of \( G_0(s) \), \( F_{stab}(s) \) etc. are listed below:

i) The real and imaginary parts of \( G_0(j\omega) \) at \( \omega = 0 \) are

\[ \text{Re}[G_0(j0)] = -1.304841 \quad \text{and} \quad \text{Im}[G_0(j0)] = 0 \]  
(3.1)

ii) The phase margin \( \phi_{0m} \) of the original system \( G_0(j\omega) \) is
\( \phi_{0m} = 180^\circ + \angle G_0(j\omega_{0c}) = -5.58^\circ \) \hspace{1cm} (3.2)

where the gain crossover frequency \( \omega_{0c} \) is given by

\( \omega_{0c} = 1.6 \text{ rad/sec. so that } |G_0(j\omega_{0c})| = 1. \) \hspace{1cm} (3.3)

Other frequency response data at \( \omega_{\text{e}\pi} = 1.9 \text{ rad/sec.} \) and \( \omega_{\text{ec}} = 3.2 \text{ rad/sec.} \) are

iii) \( \text{Re}[G_0(j\omega_{\text{e}\pi})] = -0.9370766 \) \hspace{1cm} (3.4)
\( \text{Im}[G_0(j\omega_{\text{e}\pi})] = 0.06716120 \)

iv) \( \text{Re}[G_0(j\omega_{\text{ec}})] = -0.6181657 \) \hspace{1cm} (3.5)
\( \text{Im}[G_0(j\omega_{\text{ec}})] = 0.01949691 \)

The dominant frequency response data of the stabilization filter \( F_{\text{stab}}(s) \) are

i) \( \text{Re}[F_{\text{stab}}(j0)] = 1.6 \) \hspace{1cm} \text{and} \hspace{1cm} \text{Im}[F_{\text{stab}}(j0)] = 0 \hspace{1cm} (4.1)

ii) \( \text{Re}[F_{\text{stab}}(j\omega_{\text{e}\pi})] = 1.600492 \) \hspace{0.5cm} \text{and} \hspace{0.5cm} \text{Im}[F_{\text{stab}}(j\omega_{\text{e}\pi})] = 0.1216316
\hspace{3cm} \text{at } \omega_{\text{e}\pi} = 1.9 \text{ rad/sec.} \hspace{1cm} (4.2)

or

\( |F_{\text{stab}}(j\omega_{\text{e}\pi})| = 1.605107127 \hspace{0.5cm} \text{and} \hspace{0.5cm} \angle F_{\text{stab}}(j\omega_{\text{e}\pi}) = 4.345918198^\circ \)
\hspace{3cm} \text{at } \omega_{\text{e}\pi} = 1.9 \text{ rad/sec.} \hspace{1cm} (4.3)

iii) \( \text{Re}[F_{\text{stab}}(j\omega_{\text{ec}})] = 1.601402 \) \hspace{1cm} \text{and} \hspace{1cm} \text{Im}[F_{\text{stab}}(j\omega_{\text{ec}})] = 0.2049554
\hspace{3cm} \text{at } \omega_{\text{ec}} = 3.2 \text{ rad/sec.} \hspace{1cm} (4.4)
or

\[ |F_{\text{stab}}(j\omega)| = 1.614464333 \text{ and } \angle F_{\text{stab}}(j\omega) = 7.293349493^\circ \]

at \( \omega_{\text{ec}} = 3.2 \text{ rad/sec.} \) \( (4.5) \)

Now, analyzing the data we have from Eqn. (1.3) and (1.4), it is clear that \( G_0(s) \) and \( G_e(s) \) are non-minimum phase functions and they are unstable because of the pole \( s = 2.821 \) which is in the right half plane of the s-plane. Referring to the Nyquist plots in Fig. 2, and according to Nyquist stability criterion the original missile system (without \( F_{\text{stab}}(s) \)) is unstable whereas the existing stabilized system is asymptotically stable. However, due to the small positive phase margin given in Eqn. (2.5), the time response of the existing stabilized system is quite oscillatory.

To redesign the pitch control system or the stabilization filter so that the cost of implementation is reduced and the flight control performance of the missile system is improved, two computer-aided methods have been developed. These two proposed methods will be discussed in this report step by step. In Chapter II a transfer function (called a standard transfer function \( T_r(s) \)) is obtained by using a dominant-data matching method. \( T_r(s) \) matches the assigned specifications given in Eqn. (2). Therefore, the standard transfer function \( T_r(s) \) mentioned above is a reduced-order model of the existing stabilized system \( T_e(s) \) in Eqn. (1.5). The unit step response curves of \( T_e(s) \) and \( T_r(s) \) will be compared. This comparison will also verify that the data in Eqn. (2) are dominant ones.

To solve the nonlinear equations obtained in Chapter II four
different methods of finding initial guesses are discussed in Chapter III.

In Chapter IV two reduced order models of the stabilization filter $F_{\text{stab}}(s)$ are obtained. One of these two is obtained by using the dominant-data matching method and the other by using a similar approach to fit a low order model that satisfies the specifications shown in Eqn. (4).

Chapter V consists of two parts, in the first part the dominant-data matching method is used to obtain an unstable reduced order model of the original high-order unstable system $G_0(s)$ shown in Eqn. (1.3). This is done just to simplify the design procedure. In the second part of Chapter V the algebraic method of Shieh [3] and Chen [4] is applied to redesign the pitch control system. This is done by designing a series filter in the feed forward path and a parallel filter in the feedback path. Thus, the advantages of feedback control structure have been fully utilized.
CHAPTER II

THE DOMINANT-DATA MATCHING METHOD

The design goals of a control system are often characterized by a set of control specifications. These specifications can be classified as i) time-domain specifications such as rise time, setting time, ii) frequency domain specifications such as phase margin, gain margin and iii) complex domain specifications such as damping ratio, and natural angular frequency. Empirical rules that link the specifications in the time, frequency, and complex domains have been developed by Truxal [5], Toro and Parker [6], Axelby [7] and Seshadri et al. [8]. From these results, it is observed that most time-domain specifications and complex-domain specifications can be approximately converted to frequency-domain specifications. Some of these frequency-domain specifications are phase margin ($\phi_m$), gain margin ($G_m$), maximum value of the closed-loop frequency response ($M_p$), phase crossover frequency ($\omega_n$), gain-crossover frequency ($\omega_c$), peak value frequency ($\omega_p$), the bandwidth ($\omega_b$), and the velocity error constant ($K_v$). Other important frequency response data are:

1. The real part and imaginary part of the closed-loop function $T(s)$ as well as the corresponding open-loop function $G(s)$ at $s = j\omega = j0$,

2. the real part of the open-loop transfer function $G(j\omega)$ at the phase crossover frequency $\omega_n$ which has been used to define the gain margin ($G_m$),
the corner frequencies in the Bode plot of $G(j\omega)$ in the regions of $\omega = \omega_c^1$ where $20 \log |G(j\omega_c^1)| = +15$ db and $\omega = \omega_c^2$ where $20 \log |G(j\omega_c^2)| = -15$ db.

Chen [9] has shown empirically that the open-loop poles and zeros of a system can be approximated by retaining the Bode plot in the regions of the $\pm 15$ db boundaries.

The data at $\omega = 0$ often indicate the final value and the type of the system. More specifically, the value of $T(\infty)$ or real part of $G(\infty)$ indicates the final value of the system, while the imaginary part of $G(\infty)$ indicates the type of the system. For example, for a type '0' system, the imaginary part of $G(\infty)$ is 0, and for any system other than type '0', for example, say type '1', it is infinity.

Depending on the problem one has, one can use any one or a combination of the time domain, frequency domain and complex-domain specifications. However, in this case the original pitch control system that is available is a high order transfer function with large coefficients Eqn. (1.5). As a result the time response curve and the corresponding time domain specifications of this practical system $T_e(s)$ are difficult to obtain. On the other hand, with the help of a digital computer the frequency response curve and hence the frequency domain specifications of $T_e(s)$ can be easily determined. Therefore, a frequency domain approach or a dominant data matching method is proposed to construct a transfer function $T_r(s)$, a reduced order model of $T_e(s)$, and to redesign the pitch control system. Several methods have been already proposed [10, 11,12] to obtain reduced order models by considering frequency domain specifications. However, the only reduced order models that satisfy the
assigned specifications exactly are the ones obtained by the proposed method.

From the rules of thumb it is observed that the gain margin, the phase margin, the gain cross-over frequency and the phase cross-over frequency are the most important frequency response data. These data are called the dominant frequency response data. Another important frequency response data is the steady state value of a closed-loop system, which is indirectly given by the value of the real part of the open loop transfer function \( G(j\omega) \) at \( \omega = 0 \). These dominant frequency response data may be considered as the design goal. Let us assume that the desired reduced order model of \( T_e(s) \) which may be called the standard model of \( T_e(s) \) is

\[
T_e(s) = \frac{b_0 + b_1 s + b_2 s^2}{a_0 + a_1 s + a_2 s^2 + a_3 s^3}
\]

It is required that \( T_r(s) \) satisfies all the conditions given in Eqn. (2).

From the conditions in (2.1), it may be observed that the system \( T_e(s) \) is a type 1 system. Therefore \( b_0 = a_0 \). Also, to simplify the equation we let \( a_3 = 1 \). Thus, we have

\[
T_r(s) = \frac{a_0 + b_1 s + b_2 s^2}{a_0 + a_1 s + a_2 s^2 + s^3} = \frac{G_r(s)}{1+G_r(s)} \tag{5.1}
\]

where the open-loop transfer function \( G_r(s) \) is given by

\[
G_r(s) = \frac{a_0 + b_1 s + b_2 s^2}{s[(a_1 - b_1) + (a_2 - b_2)s + s^2]} = \frac{K[1+c_1 s+c_2 s^2]}{s[l+d_1 s+d_2 s^2]} \tag{5.2}
\]
where \( K = \frac{a_0}{a_1-b_1}, \quad c_1 = \frac{b_1}{a_0}, \quad c_2 = \frac{b_2}{a_0}, \quad d_1 = \frac{a_2-b_2}{a_1-b_1}, \quad d_2 = \frac{1}{a_1-b_1} \)

The unknown coefficients \( a_i \) and \( b_i \) are to be determined by using the conditions in Eqn. (2). Following the basic definitions and substituting the assigned data in Eqn. (2) yields a set of nonlinear equations \( f_i(a_0, a_1, a_2, b_1, b_2) = 0 \) for \( i = 1, 2, \ldots, 5 \) as follows:

\[
\begin{align*}
1) \quad G_r(j\omega) &= \frac{K[1+j\omega c_1+(j\omega)^2 c_2]}{j\omega[1+j\omega d_1+(j\omega)^2 d_2]} \\
&= \frac{K}{j\omega} \left[1+j\omega(c_1-d_1)+(j\omega)^2(\quad)\right] + \ldots \\
&= \frac{K}{j\omega} [1+j\omega(c_1-d_1)] \\
&= K(c_1-d_1) - j\frac{K}{\omega} \\
\end{align*}
\]

\[
\lim_{\omega \to 0} G_r(j\omega) = K(c_1-d_1) - j\omega
\]

\[
\text{Re}_{\omega \to 0} [G_r(j\omega)] = K(c_1-d_1) = \frac{a_0}{a_1-b_1} \left(\frac{b_1}{a_0} - \frac{a_2-b_2}{a_1-b_1}\right)
\]

(6.0)

Eqn. (2.1) gives \( \text{Re}[G_r(j0)] = -2.1 \)

or \[
\frac{a_0}{a_1-b_1} \left(\frac{b_1}{a_0} - \frac{a_2-b_2}{a_1-b_1}\right) = -2.1
\]

or \[
\frac{b_1}{a_1-b_1} - \frac{a_0(a_2-b_2)}{(a_1-b_1)^2} = -2.1
\]

or \[
f_1(a_0, a_1, a_2, b_1, b_2) = b_1(a_1-b_1) - a_0(a_2-b_2) + 2.1(a_1-b_1)^2 = 0
\]

(6.1)
ii) The data in (2.2), or $\text{Re}[G_r(j\omega_{\pi})] = -1.5$, at $\omega_{\pi} = 1.9$ rad/sec gives

$$
\text{Re}[G_r(j\omega)] \bigg|_{\omega = \omega_{\pi}} = \frac{(a_0 - \omega^2 b_2) + j\omega b_1}{-\omega^2 (a_2 - b_2) + j\omega (a_1 - b_1 - \omega^2)}
$$

or

$$
\frac{-\omega_{\pi}^2 (a_0 - \omega_{\pi}^2 b_2)(a_2 - b_2) + \omega_{\pi}^2 b_1 (a_1 - b_1 - \omega_{\pi}^2)}{4 \omega_{\pi}^2 (a_2 - b_2)^2 + \omega_{\pi}^2 (a_1 - b_1 - \omega_{\pi}^2)^2} = 1.5
$$

or

$$
f_2(a_0, a_1, a_2, b_1, b_2) = (a_2 - b_2)(a_0 - 3.61b_2^2) - b_1(a_1 - b_1 - 3.61)
$$

$$= -1.5[3.61(a_2 - b_2)^2 + (a_1 - b_1 - 3.61)^2] = 0 \quad (6.2)
$$

iii) The condition in (2.3), or $\text{Re}[G_r(j\omega_{\pi})] = -180^\circ$ where $\omega_{\pi} = 1.9$ rad/sec, gives

$$
tan^{-1} \frac{\omega_{\pi} b_1}{a_0 - \omega_{\pi}^2 b_2} - 180^\circ + tan^{-1} \frac{\omega_{\pi} (a_1 - b_1 - \omega_{\pi}^2)}{\omega_{\pi} (a_2 - b_2)} = -180^\circ
$$

or

$$
tan^{-1} \frac{1.9b_1}{a_0 - 3.61b_2 + 1.9(a_2 - b_2)} = 0^\circ
$$

or

$$
f_3(a_0, a_1, a_2, b_1, b_2) = 3.61b_1(a_2 - b_2) + (a_0 - 3.61b_2)(a_1 - b_1 - 3.61) = 0 \quad (6.3)
$$

iv) The data in (2.6) or

$$
\phi_{\text{em}} = 180^\circ + \frac{\text{Re}(G_r(j\omega_{\text{ec}}))}{\omega_{\text{ec}}} = 5.7787^\circ, \text{ yields}
$$

$$
\omega_{\text{ec}} = 3.2 \text{ rad/sec}
$$
\[ 180^\circ + \tan^{-1} \frac{3.2b_1}{a_0-10.24b_2} = 180^\circ + \tan^{-1} \frac{a_1-b_1-10.24}{3.2(a_2-b_2)} = 5.7787^\circ \]

or
\[ \frac{3.2b_1}{a_0-10.24b_2} + \frac{a_1-b_1-10.24}{3.2(a_2-b_2)} = 5.7787^\circ \]

\[ \tan^{-1} \frac{3.2b_1(a_1-b_1-10.24)}{3.2(a_2-b_2)(a_0-10.24b_2)} = 5.7787^\circ \]

or
\[ \frac{10.24b_1(a_2-b_2)+(a_0-10.24b_2)(a_1-b_1-10.24)}{3.2(a_2-b_2)(a_0-10.24b_2)-3.2b_1(a_1-b_1-10.24)} = 0.10120072 \]

or
\[ f_4(a_0,a_1,a_2,b_1,b_2) = 10.24b_1(a_2-b_2)+(a_0-10.24b_2)(a_1-b_1-10.24) 
\]
\[ - 0.3238423014[(a_2-b_2)(a_0-10.24b_2)-b_1(a_1-b_1) - 10.24)] = 0 \quad (6.4) \]

v) The condition in (2.7) or
\[ |G_r(j\omega_{ec})| = 1 \text{ where } \omega_{ec} = 3.2 \text{ rad/sec, gives} \]
\[ \left| \frac{a_0-10.24b_2+j3.2b_1}{-10.24(a_2-b_2)+j3.2(a_1-b_1-10.24)} \right| = 1 \]

or
\[ f_5(a_0,a_1,a_2,b_1,b_2) = (a_0-10.24b_2)^2+10.24b_1^2-104.8576(a_2-b_2)^2 
\]
\[ - 10.24(a_1-b_1-10.24)^2 = 0 \quad (6.5) \]

Eqn. (6) is a set of high order simultaneous nonlinear algebraic equations which are very difficult to solve. The Newton-Raphson method that is available in most digital computers as a computer program package
(called the z system [15]) can be used to solve the nonlinear equations. However, it is well known that the Newton-Raphson method will only converge for a small range of starting values or the initial guesses. In order to improve the speed of convergence of the method four different methods of finding good initial guesses will be discussed in the next chapter.
CHAPTER III

THE INITIAL GUESS

In this report, the Newton-Raphson multidimensional method is suggested for solving nonlinear equations. However, as it is well known, high order nonlinear equations have many solutions and, depending on the starting values or the initial guesses, a solution may or may not be obtained. Therefore, the solution and the speed of convergence of a numerical method for solving nonlinear equations depend heavily on the initial guesses. In numerical mathematics, as well as in other areas of science, finding an appropriate initial guess for a set of nonlinear equations is itself a big problem to be solved. In this report, the following methods are proposed for good initial guesses. The applications of these methods depend on the type of problem one has.

(1) Initial guess by the synthesis method.

Suppose only the dominant frequency-response data in (2) are available and an approximate transfer function $T_r^*(s)$ of the desired $T_r(s)$ in (3.1) is required. The $T_r^*(s)$ is

$$T_r^*(s) = \frac{a_0^*b_1^*s + b_2^*s^2}{a_0^*s^3 + a_1^*s^2 + s^2}$$

(7)

where $a_1^*$ and $b_1^*$ are the initial guesses of the numerical method. In the synthesis method $T_r^*(s)$ in (7) is obtained as follows:

Step 1. In this step a second-order approximate transfer function $T_2^*(s)$ is obtained by using $\phi_m = 5.7^\circ$ and $\omega_c = 3.2$ rad/sec. in (2.6) and (2.7).
This $T_2^*(s)$ is

$$T_2^*(s) = \frac{\frac{2}{\omega_n}}{\frac{s^2 + 2\zeta \omega_n s + \omega_n^2}{s^2 + 2\zeta \omega_n s}} = \frac{G_2^*(s)}{1+G_2^*(s)}$$

(8.1)

where $\zeta$ = the damping ratio and $\omega_n$ = the natural angular frequency.

By following the basic definitions of $\omega_c$ and $\phi_m$ the following equations are obtained.

From (8.1)

$$G_2^*(s) = \frac{\frac{2}{\omega_n}}{s(s+2\zeta \omega_n)} = \frac{\frac{2}{\omega_n}}{s^2 + 2\zeta \omega_n s}$$

$$G_2^*(j \omega) = \frac{\omega_c^2}{-\omega_c^2 + j2\zeta \omega_c \omega} = \frac{\omega_c^2}{\sqrt{\omega_c^4 + 4\zeta^2 \omega_c^2 \omega^2}}$$

$$/\frac{-180^\circ + \tan^{-1} \frac{2\zeta \omega_c}{\omega}}{\omega}$$

By definition $|G_2^*(j \omega_c)| = 1$, where $\omega_c = 3.2$ rad/sec.

\[ \therefore \frac{\omega_c}{\sqrt{\omega_c^4 + 4\zeta^2 \omega_c^2 \omega^2}} = 1 \]

or, \[ \frac{\omega_c^4 - 40.96\zeta^2 \omega_c^2 \omega^2}{\omega^2} = 104.8576 = 0 \] (8.2)

Next, by definition

$$\phi_m = \angle G_2^*(j \omega_c) + 180^\circ = 5.7^\circ \text{ given}$$

\[ \therefore \ 5.7^\circ = -180^\circ + \tan^{-1} \ \frac{2\zeta \omega_c}{\omega} + 180^\circ \]

or, \[ \frac{2\zeta \omega_c}{3.2} = \tan \ 5.7^\circ \]
or \[ \omega_n = \frac{0.1597012}{\zeta} \] (8)

substituting (8.3) into (8.2) yields

\[ \zeta^4 = 0.0000061422 \]

the square root of which is

\[ \zeta^2 = \pm 0.0024783561 \]

considering the positive root only

\[ \zeta = 0.0497830911 \] we neglect the negative root

Substituting this in (8.3), yields

\[ \omega_n = 3.207940617 \text{ rad/sec.} \]

\[ \therefore T_2^*(s) = \frac{10.290883}{s^2 + 0.3194024s + 10.290883} \] (8.4)

The poles that can be considered as dominant poles of a system can be determined from the characteristic equation in (8.1). As such dominant poles are

\[ S_{1,2} = -\zeta n \pm j\omega_n \sqrt{1-\zeta^2} = -0.1598\pm j3.2039 \]
Step 2. In this step a third-order transfer function $T_3^*(s)$ is constructed by inserting a pole ($s = -p$) in it and modifying the term in the numerator of $T_2^*(s)$ so that the steady state value of the $T_3^*(s)$ is equal to unity, or

$$T_3^*(s) = \frac{p \omega_n^2}{(s^2+2\zeta \omega_n s+\omega_n^2)(s+p)} = \frac{10.290883p}{(s^2+0.3194024s+10.290883)(s+p)}$$

$$= \frac{G_3^*(s)}{1+G_3^*(s)} \quad \text{(8.5)}$$

The unknown constant $p$ is determined by using the condition in (2.2), or $\text{Re}[G_3^*(j\omega_{e\pi})] = -1.5$, where $\omega_{e\pi} = 1.9 \text{ rad/sec}$. Thus, from (8.5)

$$G_3^*(s) = \frac{10.290883p}{s^3+(p+0.3194024)s^2+(0.3194024p+10.290883)s} \quad \text{(8.6)}$$

let $s = j\omega$, then

$$G_3^*(j\omega) = \frac{10.290883p}{-\omega^2(p+0.3194024)+j\omega(0.3194024p+10.290883-\omega^2)} \quad \text{(8.7)}$$

$$\text{Re}[G_3^*(j\omega)] = \frac{-10.290883p \omega^2(p+0.3194024)}{\omega^4(p+0.3194024)^2+\omega^2(0.3194024p+10.290883-\omega^2)^2} \quad \text{(8.8)}$$

at $\omega = \omega_{e\pi} = 1.9 \text{ rad/sec}$, $\text{Re}[G_3^*(j\omega)] = -1.5$

$$\therefore \quad \frac{-10.290883(p+0.3194024)}{3.61(p+0.3194024)^2+(0.3194024p+6.680883)} = -1.5 \quad \text{(8.9)}$$

After simplification, (8.9) becomes
\[ p^2 - 1.391925823p - 14.29298735 = 0 \]

or \[ p = 4.540095027, \text{ we neglect the negative root.} \]

Thus (8.5) becomes

\[ T^*_3(s) = \frac{46.72158673}{46.72158673 + 11.74100025s + 4.859497427s^2 + s^3} \quad \text{(8.10)} \]

**Step 3.** In this step another third-order transfer function \( T_{**}^*(s) \) is constructed by inserting a zero in (8.10) as shown below.

\[ T_{**}^*(s) = \frac{46.72158673 + b_1^*s}{46.72158673 + 11.74100025s + 4.859497427s^2 + s^3} = \frac{G_{**}^*(s)}{1 + G_{**}^*(s)} \quad \text{(8.11)} \]

The unknown constant \( b_1^* \) can be determined by using the condition in (6.0) and (2.1), or \( \text{Re}[G(s)] = -2.1 \) as shown below. From (8.11), we get

\[ G_{**}^*(s) = \frac{b_1^*s + 46.72158673}{s^2 + 4.859497427s^2 + (11.74100025 - b_1)b_1s} \]

or

\[ G_{**}^*(s) = \frac{46.72158673[1 + \frac{b_1}{46.72158673}]}{s(11.74100025 - b_1)[1 + \frac{4.859497427}{11.74100025 - b_1}s + ...]} \]

According to Eq. (6.0)

\[ \lim_{\omega \to 0} \text{Re}[G_{**}^*(j\omega)] = \frac{46.72158673}{11.74100025 - b_1} \left( \frac{b_1}{46.72158673} - \frac{4.859497427}{11.74100025 - b_1} \right) \]

Given \( \text{Re}[G_{**}^*(j\omega)] = -2.1 \) in (2.1)
\[ 46.72158673 - \frac{b_1}{11.74100025 - b_1} \left( \frac{b_1}{46.72158673 - \frac{4.859497427}{11.74100025 - b_1}} \right) = -2.1 \]

or
\[ b_1^2 - 34.15563709b_1 + 56.76713817 = 0 \]

which gives \( b_1 = 32.4037687 \), since we are interested in the positive value only.

Substituting this into (8.11), we have

\[ T_3^*(s) = \frac{46.7216 + 32.4038s}{46.7216 + \frac{11.7410s + 4.8595s^2}{2 + s}} \]  \hspace{1cm} (8.12)

Equation (8.12) can be considered as an approximation of (7) by assuming \( b_2^* = 0 \). Thus the initial guesses in (7) are \( a_0^* = 46.7216 \), \( a_1^* = 11.7410 \), \( a_2^* = 4.8595 \), \( b_1^* = 32.4038 \), and \( b_2^* = 0 \). For solving Eq. (6.1)-(6.5) these constants are used as initial guesses for the Newton-Raphson method [15]. It is found that the numerical method converges at the 9th iteration with the error tolerance of \( 10^{-6} \). The solutions of (6.1)-(6.5) are
\begin{align*}
    a_0 &= 6.378070, \\
    a_1 &= 10.462220, \\
    a_2 &= 1.259008, \\
    b_1 &= 20.55667 \\
    b_2 &= 0.243466.
\end{align*}

Therefore, the desired transfer function \( T_r(s) \) is

\[ T_r(s) = \frac{6.378070 + 20.55667s + 0.243466s^2}{6.378070 + 10.462220s + 1.259008s^2 + s^3} \]  \hspace{1cm} (9)

The system represented by Eq. (9) has the exact frequency response data specified in (2).

(2) Initial guess by complex-curve fitting and continued fraction methods

In this part a simple method is presented to determine the ap-
proximate coefficients of a transfer function using the real parts and imaginary parts of the limited number of frequency-response data that are available. Using these data a low-order model is constructed. The low-order model is then expanded into a continued fraction of the second Cauer form to obtain a set of dominant quotients. Some non-dominant quotients are then inserted into the continued fraction to obtain an amplified-order model [16], which is the desired approximate transfer function \( T_r^*(s) \) for the use of the initial guess.

Consider the transfer function

\[
T_r(s) = \frac{b_0 + b_1 s + b_2 s^2 + \ldots + b_m s^m}{1 + a_1 s + a_2 s^2 + \ldots + a_n s^n}
\]  

where \( a_i \) and \( b_i \) are unknown coefficients to be determined. The problem of finding unknown coefficients of a transfer function as a ratio of two frequency-dependent polynomials has been investigated by Levy [17]. His method minimizes the sum of squares of the errors at arbitrary experimental points. However, for finding the unknown coefficients of a transfer function the method presented next is comparatively simple and straightforward.

Substituting \( s = j\omega_k \) into (10.1), we have

\[
T_r^*(j\omega_k) = \frac{b_0 + j\omega_k b_1 + (j\omega_k)^2 b_2 + \ldots + (j\omega_k)^m b_m}{1 + j\omega_k a_1 + (j\omega_k)^2 a_2 + \ldots + (j\omega_k)^n a_n}
\]

Separating the real parts and imaginary parts in the numerator and denominator of \( T_r^*(j\omega_k) \) we have
where \( R_k \) and \( I_k \) are the given values of the real and imaginary parts of the \( T_r^*(s) \) at the available frequencies \( \omega_k \). Multiplying both sides of (10.2) by the common denominator and separating the real and imaginary parts, we have

\[
T_r^*(j\omega_k) = \frac{(b_0 - b_2 \omega_k^2 + b_4 \omega_k^4 - b_6 \omega_k^6 + \ldots) + j(b_{-3} \omega_k^3 + b_{-5} \omega_k^5 - b_{-7} \omega_k^7 + \ldots)}{(1 - a_2 \omega_k^2 + a_4 \omega_k^4 - a_6 \omega_k^6 + \ldots) + j(a_{-3} \omega_k^3 + a_{-5} \omega_k^5 - a_{-7} \omega_k^7 + \ldots)}
\]

\[
= R(\omega_k) + jI(\omega_k)
\]

\[
= R_k + jI_k
\]

(10.2)

Equating the real and imaginary parts from both sides, yields

\[
b_0 - b_2 \omega_k^2 + b_4 \omega_k^4 - b_6 \omega_k^6 + \ldots = R_k - a_2 R_k \omega_k^2 + a_4 R_k \omega_k^4 - a_6 R_k \omega_k^6 + \ldots - a_1 I_k \omega_k^3 - a_3 I_k \omega_k^5 - a_5 I_k \omega_k^7 + \ldots
\]

(10.3)

\[
b_1 \omega_k + b_3 \omega_k^3 + b_5 \omega_k^5 - b_7 \omega_k^7 + \ldots = a_1 R_k \omega_k^2 + a_3 R_k \omega_k^4 + a_5 R_k \omega_k^6 + \ldots - a_2 I_k \omega_k^3 - a_4 I_k \omega_k^5 - a_6 I_k \omega_k^7 + \ldots
\]

(10.4)

Eq. (10.3) and (10.4) can be written as
\[ b_0 - b_2 \omega_k^2 + b_4 \omega_k^4 - b_6 \omega_k^6 + \ldots + a_1 I \omega_k + a_2 R \omega_k^2 + a_3 I \omega_k^3 - a_4 R \omega_k^3 + \ldots = R_k \]

(10.5)

\[ b_1 \omega_k - b_3 \omega_k^3 + b_5 \omega_k^5 - b_7 \omega_k^7 + \ldots - a_1 R \omega_k + a_2 I \omega_k^2 + a_3 R \omega_k^3 - a_4 I \omega_k^4 + \ldots = I_k \]

(10.6)

In matrix form, (10.5) becomes

\[
\begin{bmatrix}
1 & -\omega_1 & \omega_4 & -\omega_6 & I_1 \omega_1 & R_1 \omega_2 & -I_1 \omega_1 & -R_1 \omega_1 & \vdots & b_0 \\
1 & -\omega_2 & \omega_1 & -\omega_6 & I_2 \omega_2 & R_2 \omega_2 & -I_2 \omega_2 & -R_2 \omega_2 & \vdots & b_2 \\
1 & -\omega_3 & \omega_4 & -\omega_6 & I_3 \omega_3 & R_3 \omega_3 & -I_3 \omega_3 & -R_3 \omega_3 & \vdots & b_4 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & -\omega_1 & \omega_4 & -\omega_6 & I_n \omega_1 & R_n \omega_2 & -I_n \omega_2 & -R_n \omega_2 & \vdots & a_n \\
\end{bmatrix}
\begin{bmatrix}
b_0 \\
b_2 \\
b_4 \\
\vdots \\
a_n \\
\end{bmatrix}
= \begin{bmatrix}
R_1 \\
R_2 \\
R_3 \\
\vdots \\
R_n \\
\end{bmatrix}
\]

(10.7)

where \( x = n + \frac{m}{2} + 1 \), if \( m \) is even

\[ = n + \frac{m+1}{2} \], if \( m \) is odd

Substituting \( a_i \) obtained in (10.7) into (10.6), we have another matrix equation to solve for \( b_i \), \( i = 1, 3, 5, \ldots \)

\[
\begin{bmatrix}
\omega_1 & -\omega_1 & \omega_5 & -\omega_7 & \ldots & b_1 & \left((a_0 I \omega_1^0 + a_1 R \omega_1^1) - (a_2 I \omega_1^2 + a_3 R \omega_1^3) + \ldots\right)
\\
\omega_2 & -\omega_2 & \omega_5 & -\omega_7 & \ldots & b_3 & \left((a_0 I \omega_2^0 + a_1 R \omega_2^1) - (a_2 I \omega_2^2 + a_3 R \omega_2^3) + \ldots\right)
\\
\omega_3 & -\omega_3 & \omega_5 & -\omega_7 & \ldots & b_5 & \ldots
\\
\omega_4 & -\omega_4 & \omega_5 & -\omega_7 & \ldots & b_k & \left((a_0 I \omega_4^0 + a_1 R \omega_4^1) - (a_2 I \omega_4^2 + a_3 R \omega_4^3) + \ldots\right)
\end{bmatrix}
\begin{bmatrix}
b_1 \\
b_3 \\
b_5 \\
\vdots \\
b_k \\
\end{bmatrix}
= \begin{bmatrix}
\left((a_0 I \omega_1^0 + a_1 R \omega_1^1) - (a_2 I \omega_1^2 + a_3 R \omega_1^3) + \ldots\right)
\\
\left((a_0 I \omega_2^0 + a_1 R \omega_2^1) - (a_2 I \omega_2^2 + a_3 R \omega_2^3) + \ldots\right)
\\
\ldots
\\
\left((a_0 I \omega_4^0 + a_1 R \omega_4^1) - (a_2 I \omega_4^2 + a_3 R \omega_4^3) + \ldots\right)
\end{bmatrix}
\]

(10.8)
where \( \omega_k^0 = 1, a_0 = 1; K = m \) and \( y = \frac{m+1}{2} \) if \( m \) is odd; \( K = m-1 \) and \( y = \frac{m}{2} \) if \( m \) is even.

In this pitch control system, the available data is given in (2) from which the following data is obtained,

\[
\begin{align*}
\omega_1 &= \omega_0 = 0, & R_1 &= T_e(j0) = 1, & I_1 &= 0 \\
\omega_2 &= \omega_e = 1.9, & R_2 &= \text{Re}\left[\frac{G(j\omega_e)}{1+G(j\omega_e)}\right] = 2.968398, & I_2 &= \text{Im}\left[\frac{G(j\omega_e)}{1+G(j\omega_e)}\right] = 0.02515098 \\
\omega_3 &= \omega_c = 3.2, & R_3 &= \text{Re}\left[\frac{G(j\omega_c)}{1+G(j\omega_c)}\right] = 0.3350731, & I_3 &= \text{Im}\left[\frac{G(j\omega_c)}{1+G(j\omega_c)}\right] = -10.43159
\end{align*}
\]

Data is available only at three frequencies, therefore the approximate transfer \( T_2^*(s) \) is assumed to be

\[
T_2^*(s) = \frac{b_0 + b_1 s}{1 + a_1 s + a_2 s^2}
\]  

(12.1)

Substituting the data at \( \omega_1, \omega_2 \) and \( \omega_3 \) in (11) into (10.7) yields

\[
\begin{bmatrix}
1 & 0 & 0 \\
1 & -0.047786862 & 10.71591678 \\
1 & -33.381088 & 3.431148544
\end{bmatrix}
\begin{bmatrix}
b_0 \\
a_1 \\
a_2
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
2.968398 \\
0.3350731
\end{bmatrix}
\]

(12.2)

From (12.2), we get

\[
\begin{align*}
b_0 &= 1 \\
b_0 - 0.047786862a_1 + 10.71591678a_2 &= 2.968398 \\
b_0 - 33.381088a_1 + 3.431148544a_2 &= 0.3350731
\end{align*}
\]  

(12.3)  

(12.4)
Substituting $b_0 = 1$ into (12.3) and (12.4) yields

$$-0.047786862a_1 + 10.71591678a_2 = 1.968398$$
$$-33.381088a_1 + 3.431148544a_2 = -0.6649269$$

Solving these two equations, we get

$$a_1 = 0.0388179596$$
$$a_2 = 0.1838622891$$

Then substituting $a_1$ and the data at $\omega_2$ into (10.8) yields

$$3.2b_1 = 9.250106342$$
$$\therefore b_1 = 2.890658232$$

Substituting $a_1$ and $b_1$, into (12.1) gives

$$T_2^*(s) = \frac{1+2.890658232s}{1+0.0388179596s+0.1838622891s^2}$$

However, the desired approximate transfer function in (7) is a third-order function. Therefore $T_2^*(s)$ in (12.5) needs to be amplified. In this case this is done by using the continued fraction method [16] as follows.

$T_2^*(s)$ is first expanded into a continued fraction of the second Cauer form to obtain a set of dominant quotients. They are given as
Then the order of $T_2^*(s)$ is amplified by inserting two nondominant quotients $h_5 = 100$ and $h_6 = 0.1$, or

$$T_2^*(s) = \frac{1 + 2.890658232s}{1 + 0.0388179596s + 0.1838622891s^2}$$

$$= \frac{1}{h_1 + \frac{1}{h_2 + \frac{1}{h_3 + \frac{1}{h_4 + \frac{1}{h_5 + \frac{1}{h_6}}}}}}$$

(12.6)

Substituting

$$h_1 = 1$$
$$h_2 = -0.3506507744$$
$$h_3 = 0.9650474175$$
$$h_4 = 16.07251656$$
\[ h_5 = 100 \]
\[ h_6 = 0.1 \]

Into (12.6), it becomes

\[
T_2^*(s) = T_3^*(s) = T_r^*(s) = \frac{54.3885 + 162.6914s + 15.8219s^2}{54.3885 + 7.5839s + 10.2146s^2 + s^3}
\]  \hspace{1cm} (12.7)

In solving (6.1)-(6.5) if we use the coefficients in (12.7) as initial guesses; \( a_0^* = 54.3885, a_1^* = 7.5839, a_2^* = 10.2146, b_1^* = 162.6914 \) and \( b_2^* = 15.8219 \), we have the desired coefficients in (9) at the 15th iteration [15] with the error tolerance of \( 10^{-6} \). This proves once again that if the inserted positive quotients \( h_i \gg 1 \) and \( h_{i+1} < 1 \) (i is an odd number) the amplified order model is a good approximation of the original low-order model.

(3) Initial guess by continued fraction method [18]

Shieh [3] and Chen [10] have proposed a continued fraction method for model reductions. In this case their method is utilized to find initial guesses to solve Eq. (6.1)-(6.5). The numerator polynomial \( N(s) \) and the denominator polynomial \( D(s) \) in Eq. (1.5) are arranged in ascending order and expanded into the continued fraction of the second Cauer form by performing repeated long divisions as follows.

\[
T_e(s) = \frac{N(s)}{D(s)} = \frac{b_{10} + b_9 s + b_8 s^2 + \ldots + b_0 s^{10}}{a_{11} + a_{10} s + a_9 s^2 + \ldots + a_0 s^{11}} \text{ where } a_i, b_i \text{ are given in (1.5)}
\]

\[
= \frac{1}{h_1 + \frac{s}{h_2 + \frac{s}{h_3 + \frac{s}{h_4 + \frac{s}{h_5 + \ldots}}}}}
\]  \hspace{1cm} (13.0)
where \( h_1 = 1 \)
\[
\begin{align*}
h_2 &= -0.401749 \\
h_3 &= -0.475321 \\
h_4 &= 25.1998 \\
h_5 &= -0.0322195 \\
h_6 &= -24.1061 \\
h_7 &= \ldots \\
h_{22} &= \ldots
\end{align*}
\]

The reduced order models of \( T_e(s) \) can be obtained by retaining the first few dominant quotients, \( h_i = 1, 2, \ldots \). The number of quotients used depends on the order and form of the reduced model. This is explained below.

\[
T_e(s) = \frac{1}{h_1 + \frac{s}{h_2}} = \frac{h_2}{h_1 h_2^2 + s} \tag{13.1}
\]

\[
T_e(s) = \frac{1}{h_1 + \frac{s}{h_2}} = \frac{h_2 h_3 + s}{h_1 h_2 h_3 + (h_1 + h_3)s} \tag{13.2}
\]

\[
T_e(s) = \frac{1}{h_1 + \frac{s}{h_2}} = \frac{h_2 h_3 h_4 + (h_2 h_4 + h_3 h_4)s + s^2}{h_1 h_2 h_3 h_4 + (h_1 h_2 h_4 + h_1 h_3 h_4 + h_2 h_3 h_4)s + s^2} \tag{13.3}
\]
Substituting the $h_i$'s in (13.1) into (13.6) yields the third-order approximate model of $T_e(s)$ as follows:

\[
T_3^*(s) = \frac{3.7376 + 10.4692s + 0.6920s^2}{3.7376 + 10.1661s + 0.9488s^2 + s^3}
\]

Using the coefficients in (14.1) as the initial guesses: $a_0^* = 3.7376$, $a_1^* = 10.1661$, $a_2^* = 0.9488$, $b_1^* = 19.4692$ and $b_2^* = 0.6920$, the desired solution ($T_r(s)$ given in (9)) of the set of nonlinear equation (6.1)-(6.5) are obtained at the 8th iteration with the error tolerance of $10^{-6}$.

As it has just been shown in this particular case, the continued fraction method of finding the initial guess has worked out nicely.

However, this is not true always. For example, if the reduced order model by the continued fraction method turns out to be an unstable system,
the coefficients of such a reduced order model cannot be used to solve a set of nonlinear equations. Because, an unstable initial guess often leads to solutions which will give rise to an unstable system only. In such cases the following mixed method can be used to obtain a stable reduced-order model for approximation.

(4) Initial guess by using the mixed method.

In this section of the report two mixed methods are discussed. One has the advantages of both the continued fraction method [3,10] and the dominant pole method [19]. The other has the advantages of the continued fraction as well as the Routh table [11], from which the equivalent dominant-poles can be obtained. The reduced-order models obtained by the mixed method are stable and can be used as good initial guesses.

The relationship between the quotients \( h_i \) and the coefficients \( a_i \) and \( b_i \) in (13.0) can be expressed by the following matrix Eq. [3,4]:

\[
[b] = [H] [a]
\]  

(15)

where \( [a]^T = [a_{n-1}, a_{n-2}, \ldots, a_2, a_1, a_0] \), \( [b]^T = [b_{n-1}, b_{n-2}, \ldots, b_2, b_1, b_0] \), \( [H] = [H_2]^{-1} [H_1] \),

here \( T \) designates transpose of a matrix.
Consider the reduced-order model of the original system as

\[
T_r(s) = \frac{e_0 + e_1 s + \ldots + e_{r-1} s^{r-1}}{d_0 + d_1 s + \ldots + d_{r-1} s^{r-1} + d_r s^r}, \quad d_r = 1
\]  

(16)

The denominator polynomial in (10) is approximated by the product of the dominant poles of the original system \( T_e(s) \). Thus \( d_1 \) is known. Replacing \( a_i \) and \( b_i \) in (15) by \( d_i \) and \( e_i \) in (16), respectively, Eq. (15) can be solved for \( e_i \) in (16). The \( T_r(s) \) obtained has the dominant poles and the dominant quotients of \( T_e(s) \) and it is always stable, therefore, \( T_r(s) \) can be used as a good initial guess in solving (6.1)-(6.5).

In case the roots of \( D(s) \) in (13.0) are not available, the ap-
proximate equivalent dominant poles and the resulting coefficients $d_i$ can be determined from the Routh table as suggested by Hutton and Friedland[9]. The steps involved are explained below.

Assume $T(s) = \frac{b_n s^{n-1} + b_{n-2} s^{n-2} + \ldots + b_2 s + b_1 + b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_2 + a_1 + a_0} = \frac{n(s)}{d(s)}$ \hspace{1cm} (17.1)

is the original transfer function for which the reduced order model is needed.

**Step 1.** Construct a Routh array [20] using the coefficients $a_i$ of the $d(s)$ above and the Routh algorithm. The Routh array is shown below. To obtain a general algorithm $a_{i,j}$ is expressed double-subscripted notation, for example, $a_{1,1}$

\[
\begin{align*}
\gamma_1 &= \frac{a_{11}}{a_{21}} \\
\gamma_2 &= \frac{a_{21}}{a_{31}} \\
\gamma_3 &= \frac{a_{31}}{a_{41}} \\
\end{align*}
\]

\[
\begin{align*}
a_{11} &= \hat{a}_n \\
a_{12} &= \hat{a}_{n-2} \\
a_{13} &= \hat{a}_{n-4} \\
\vdots & \quad \ldots \\
a_{0} &= \\
\end{align*}
\]

\[
\begin{align*}
a_{21} &= \hat{a}_{n-1} \\
\gamma_1 &= a_{12} - \gamma_1 a_{22} \\
a_{23} &= \hat{a}_{n-5} \\
\gamma_2 &= a_{13} - \gamma_1 a_{23} \\
\gamma_3 &= \hat{a}_{n-9} \\
\end{align*}
\]

\[
\begin{align*}
a_{31} &= \hat{a}_{12} - \gamma_2 a_{22} \\
a_{32} &= \hat{a}_{23} - \gamma_1 a_{23} \\
\gamma_1 &= a_{13} - \gamma_2 a_{23} \\
a_{33} &= \hat{a}_{n-13} \\
\end{align*}
\]

\[
\begin{align*}
a_{41} &= \hat{a}_{22} - \gamma_2 a_{32} \\
a_{42} &= \hat{a}_{23} - \gamma_2 a_{33} \\
\gamma_2 &= a_{23} - \gamma_2 a_{33} \\
\end{align*}
\]

\[
\begin{align*}
\gamma_3 &= \hat{a}_{n-17} \\
\gamma_3 &= \hat{a}_{n-19} \\
\vdots & \quad \ldots \\
\end{align*}
\]
In general

\[ a_{i,j} = a_{i-2,j+1} - \gamma_{i-2} a_{i-1,j+1}; \quad i = 1, 2, \ldots, j = 3, 4, \ldots \]

\[ \gamma_{i-1} = a_{i-1,1}/a_{i+1,1} \]  

(17.3)

**Step 2.** In this step various approximate low-order polynomials \( d_i^*(s) \) are constructed from any two consecutive rows in the Routh array, for example, say from the last row and the next to the last row and so on. This is explained below.

The first order \((i = 1)\) approximate equation is

\[ d_1^*(s) = a_{n-1,1} s + a_{n+1,1} = a_{n,1} s + a_0 = 0 \]  

(17.4)

The second order \((i = 2)\) approximate equation is

\[ d_2^*(s) = a_{n-1,1} s^2 + a_{n,1} s + a_{n-2,1} = a_{n-1,1} s^2 + a_{n,1} s + a_0 = 0 \]  

(17.5)
The third order \((i = 3)\) approximate equation is

\[
d_3^*(s) = \frac{a_{n-2,1}s^3 + a_{n-1,1}s^2 + a_{n-2,2}s + a_n}{s^3 + 0.9523822967s^2 + 10.19241445s + 3.745517989} = 0
\]

and so on.

When the original system (17.1) is asymptotically stable, all \(\gamma_i\) are positive values and the approximate polynomials \(d_i^*(s)\) are the Hurwitz polynomials. The \(d_i^*(s)\) are normalized simply by dividing each coefficient in \(d_i^*(s)\) by the coefficient of the highest order term in \(s\). These normalized \(d_i^*(s)\) are the denominator polynomials of the reduced-order models \(T_i(s)\) of the original system. Then the numerator polynomials of \(T_i(s)\) are determined simply by substituting the coefficients of \(d_i^*(s)\) in place of \([a]\) in (15) and then solving the matrix equation (15) for \([b]\), which are the coefficients of the numerator polynomial of \(T_i(s)\).

The third-order reduced order model \(T_{3m}(s)\) of the original pitch control system in (1.5) obtained by using the mixed method is explained below.

At first, the Routh array of the pitch control system in (1.5) is obtained. From the Routh array the normalized approximate denominator \(d_3^*(s)\) is found.

\[
d_3^*(s) = s^3 + 0.9523822967s^2 + 10.19241445s + 3.745517989
\]

To determine \(n_3(s) = b_2s^2 + b_1s + b_0\), the coefficients of \(d_3^*(s)\) are substituted into (15) as shown below.
where the $h_i$'s are the quotients of $T_e(s)$ in (1.5), which are given in (13.1). Substituting the values of $h_1$, $h_2$ and $h_3$ from (13.1) into (17.9)
and then simplifying, we get

\[ b_0 = 3.7455 \]  \hspace{1cm} (18.1)

\[ b_0 - 0.401749 b_1 = -4.094787 \]

Substituting (18.1) yields

\[ b_1 = 19.5154 \]  \hspace{1cm} (18.2)

and

\[ 0.524679 b_1 + 0.19096 b_2 = 10.37427 \]

Substituting (18.2) yields

\[ b_2 = 0.7066 \]  \hspace{1cm} (18.3)

Therefore

\[ T_{3m}^*(s) = \frac{0.7066s^2 + 19.5154s + 3.7455}{s^3 + 0.9524s^2 + 10.1924s + 3.7455} \]

(19)

In solving the nonlinear Eqs. (6.1)-(6.5) if the coefficient of \( T_{3m}^*(s) \) in (19) are used as starting values: \( a_0^* = 3.7455 \), \( a_1^* = 10.1924 \), \( a_2^* = 0.9524 \), \( b_1^* = 19.5154 \) and \( b_2^* = 0.7066 \); the Newton-Raphson method [15] converges to the desired solution in (9) or

\[ T_r(s) = \frac{6.37807 + 20.55661s + 0.243466s^2}{6.37807 + 10.46222s + 1.259008s^2 + s^3} = \frac{G_r(s)}{1+G_r(s)} \]

at the 8th iteration with the error tolerance of \( 10^{-6} \).
From (9)

\[ G_r(s) = \text{The open-loop transfer function of the standard model } T_r(s). \]

\[ T_r(s) = \frac{6.37807+20.55661s+0.24346s^2}{s(-10.09439+1.015542s+s^2)} \]  \hspace{1cm} (20)

The Nyquist plot of \( G_r(s) \) is shown in Fig. 2 and the unit step responses of \( T_r(s), T_3(s), T_{3m}(s) \) and \( T_e(s) \) are compared in Fig. 3. All three reduced-order models \( T_r(s), T_3(s) \) and \( T_{3m}(s) \) give very satisfactory approximate time response curves. However, only the \( T_r(s) \) in (9), which uses the method of dominant frequency response data matching, has the exact dominant-frequency response data as the original system \( T_e(s) \) given in (2).
Figure 3. Time responses of original and third order reduced models.
CHAPTER IV

SIMPILIFICATION OF THE EXISTING STABILIZATION FILTER

As it appears from its name, the purpose of the stabilization filter is to stabilize the original unstable system. The transfer function of the existing stabilization filter $F_{stab}(s)$ is known and is given in (1.2). As it is mentioned in the introduction of this report, the objective of this report is to redesign the stabilization filter so that the cost of implementation can be reduced and at the same time the performance of the redesigned pitch control system is the same as that of the existing stabilized pitch control system.

In this chapter two different transfer functions are obtained for the stabilization filter. Both of these transfer functions are obtained by direct simplification of the available transfer function of $F_{stab}(s)$, and one of them is obtained by using the dominant data matching method of Chapter II.

The $F_{stab}(s)$ in (1.2) can be considered as the closed-loop transfer function of a control system as

$$F_{stab}(s) = \frac{N(s)}{D(s)} = \frac{G_{stab}(s)}{1 + G_{stab}(s)} = \frac{460800s^2 + 69120000s + 144 \times 10^7}{s^4 + 250s^3 + 76900s^2 + 72 \times 10^5 s + 9 \times 10^8} \quad (21.1)$$

where the open-loop transfer function $G_{stab}(s)$ is

$$G_{stab}(s) = \frac{460800s^2 + 69120000s + 144 \times 10^7}{s^4 + 250s^3 + 383900s^2 + 61920000s + 5.4 \times 10^8} \quad (21.2)$$

The dominant frequency-response data of this system are given below.
\[ G_{\text{stab}}(j0) = \frac{1}{0.375} \] (22.1)

\[ \text{Re}[G_{\text{stab}}(j\omega_s)] = -1.032833 \] (22.2)
\[ \text{Im}[G_{\text{stab}}(j\omega_s)] = 0.002017351 \] (22.3)

where \( \omega_s \) = The phase crossover frequency of the stabilization filter
\[ = 140 \text{ rad/sec}. \]

\[ \text{Re}[G_{\text{stab}}(j\omega_{sc})] = -1.002941 \] (22.4)
\[ \text{Im}[G_{\text{stab}}(j\omega_{sc})] = -0.03668759 \] (22.5)

where \( \omega_{sc} \) = The gain crossover frequency of the stabilization filter
\[ = 200 \text{ rad/sec}. \]

Suppose the reduced-order model \( F_{s1}(s) \) of the stabilization filter is

\[ F_{s1}(s) = \frac{b_0 + b_1 s}{a_0 + a_1 s + s^2} = \frac{G_{s1}(s)}{1 + G_{s1}(s)} \] (23.1)

where \( G_{s1}(s) \) = The open-loop transfer function of \( F_{s1}(s) \)
\[ = \frac{b_0 + b_1 s}{(a_0 - b_0) + (a_1 - b_1) s + s^2} \] (23.2)

The constants \( a_1 \) and \( b_1 \) are unknown constants to be determined. Using the specifications given in (22) and following the basic definitions of these specifications the unknown constants \( a_1 \) and \( b_1 \) are determined as
shown below.

For $F_{s_1}(s)$ in (23.1) to be a reduced order model of $F_{\text{stab}}(s)$, $G_{s_1}(s)$ must satisfy all the specifications of $G_{\text{stab}}(s)$ in (22). Applying the condition in (22.1) to the system $G_{s_1}(s)$ in (23.2) yields

$$
G_{s_1}(j0) = \frac{b_0}{a_0 - b_0} = \frac{1}{0.375}
$$

or, $b_0 = 1.6a_0$ \hspace{1cm} (24.1)

Substituting (24.1) into (23.1) and (23.2), respectively, we get

$$
F_{s_1}(s) = \frac{1.6a_0 + b_1 s}{a_0 + a_1 s + s^2}
$$

(24.2)

and

$$
G_{s_1}(s) = \frac{1.6a_0 + b_1 s}{-0.6a_0 + (a_1 - b_1)s + s^2}
$$

(24.3)

at $s = j0$

$$
G_{s_1}(j0) = \frac{1.6a_0 + j\omega b_1}{-(0.6a_0 + \omega^2) + j\omega(a_1 - b_1)}
$$

$$(1.6a_0 + j\omega b_1)(-(0.6a_0 + \omega^2) - j\omega(a_1 - b_1)) = \frac{(1.6a_0 + j\omega b_1)(-0.6a_0 + \omega^2 + j\omega(a_1 - b_1))}{(0.6a_0 + \omega^2)^2 + \omega^2(a_1 - b_1)^2}
$$

$$. \cdot \cdot\cdot \text{ Re}[G_{s_1}(j\omega)] = \frac{-1.6a_0(0.6a_0 + \omega^2) + \omega^2 b_1(a_1 - b_1)}{(0.6a_0 + \omega^2)^2 + \omega^2(a_1 - b_1)^2}
$$

(24.4)

$$
\text{ Im}[G_{s_1}(j\omega)] = \frac{-\omega b_1(0.6a_0 + \omega^2) - 1.6a_0(a_1 - b_1)}{(0.6a_0 + \omega^2)^2 + \omega^2(a_1 - b_1)^2}
$$

(24.5)

i) Specification in (22.2) yields
\[ \text{Re}[G_{s1}(j140)] = -1.032833 \]

Substituting (24.4) above gives

\[
-1.6a_0(0.6a_0+19600)+19600b_1(a_1-b_1)
\]
\[
\frac{1}{(0.6a_0+19600)^2+19600(a_1-b_1)^2} = -1.032833
\]

or

\[ f_1(a_0,a_1,b_1) = -1.6a_0(0.6a_0+19600)+19600b_1(a_1-b_1) \]
\[ + 1.032833[(0.6a_0+19600)^2+19600(a_1-b_1)^2] = 0 \]

(25.1)

ii) The data in (22.3) when applied to (24.5) yields

\[ \text{Im}[G_{s1}(j140)] = 0.002017351 \]

or

\[
-140b_1(0.6a_0+19600)-224a_0(a_1-b_1)
\]
\[
\frac{1}{(0.6a_0+19600)^2+19600(a_1-b_1)^2} = 0.002017351
\]

or

\[ f_2(a_0,a_1,b_1) = -140b_1(0.6a_0+19600)-224a_0(a_1-b_1) \]
\[ - 0.002017351[(0.6a_0+19600)^2+19600(a_1-b_1)^2] = 0 \]

(25.2)

iii) The data in (22.4) when applied to (24.4) gives

\[ \text{Re}[G_{s1}(j200)] = -1.002941 \]

or

\[
-1.6a_0(0.6a_0+40000)+40000b_1(a_1-b_1)
\]
\[
\frac{1}{(0.6a_0+40000)^2+40000(a_1-b_1)^2} = -1.002941
\]
or \[ f_j(a_0, a_1, b_1) = -1.6a_0(0.6a_0 + 40000) + 40000b_1(a_1 - b_1) \]
\[ + 1.002941[(0.6a_0 + 40000)^2 + 40000(a_1 - b_1)^2] = 0 \]
(25.3)

Equation (25) is a set of nonlinear equations. The unknown constants \( a_1 \) and \( b_1 \) in (23.1) are determined by solving (25). However, to solve equation (25) the proper initial guesses have to be determined first. As discussed in Chapter III, the initial guesses can be determined from the reduced-order model of the existing stabilization filter \( F_{stab}(s) \) in (1.2). Using the mixed method of the continued fraction approximation and the Routh approximation, a reduced-order model \( F_{rl}^*(s) \) is obtained as follows.

Assume \( F_{rl}^*(s) = \frac{b_0 + b_1 s}{a_0 + a_1 s + s^2} \) \( n^*(s) \) \( d^*(s) \)

Routh's stability criterion for \( F_{stab}(s) \) is

\[
\begin{array}{cccc}
s^4 & 1 & 76900 & 9 \times 10^8 \\
s^3 & 250 & 72 \times 10^5 \\
s^2 & 48100 & 9 \times 10^8 \\
s^1 & 2522245.322 \\
s^0 & 9 \times 10^8 \\
\end{array}
\]

As discussed in the Section 4 of Chapter III, the \( d^*(s) \) in (26.1) is approximated from the Routh criterion shown above. Thus

\[ d^*(s) = 48100s^2 + 2522245.322s + 9 \times 10^8 = 0 \]

After normalization \( d^*(s) \) becomes
\[ d^*(s) = s^2 + 52.4375s + 18711.01871 \]  

Therefore, now (26.1) becomes

\[ \frac{F^*_r(s)}{s^2 + 52.4375s + 18711.01871} = \frac{b_s^* + b_0^*}{b_s + b_0} \]  

The quotients \( h_i \) of \( F_{stab}(s) \) are obtained below:

\[
\begin{align*}
    h_1 &= 0.625 \\
    h_2 &= -40 \\
    h_3 &= -0.5933 \\
    \vdots & \quad \vdots
\end{align*}
\]

Using Eq. (15) \( b_0^* \) and \( b_1^* \) in (26.1) are determined as follows.

\[
\begin{align*}
    \begin{bmatrix} h_1 & 0 \\ 1 & h_1 h_2 \end{bmatrix} b_0^* &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 18711.01871 \\ 52.84375 \end{bmatrix} \\
    \begin{bmatrix} 0.625 & 0 \\ 1 & -25 \end{bmatrix} b_0^* &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 18711.01871 \\ 52.4375 \end{bmatrix}
\end{align*}
\]
or \[ b_0^* = \frac{18711.01871}{0.625} = 29937.62994 \] (26.4)

and \[ b_0^* - 25b_1^* = -2097.5 \]

or \[ b_1^* = \frac{-2097.5 - b_0^*}{-25} = \frac{2097.5 + 29937.62994}{25} \]

or \[ b_1^* = 1281.40525 \] (26.5)

Substituting (26.4) and (26.5) into (26.3) yields

\[ F_{r1}(s) = \frac{1281.40525s + 29937.62994}{s^2 + 52.4375s + 18711.01871} \] (26.6)

Using the coefficients of \( F_{r1}(s) \) in (26.6): \( a_0^* = 18711.01871 \), \( a_1^* = 52.4375 \), \( b_1^* = 29937.62994 \) as initial guesses, the nonlinear equations in (25) are solved by the Newton-Raphson method [15] and the following solutions are obtained at the 7th iteration with the error tolerance of \( 10^{-6} \):

\[ a_0 = 20917.459536 \]
\[ a_1 = 29.981293 \]
\[ b_1 = 957.260014 \]

Since \( b_0^* = 1.6a_0 \) as in (24.1) \( b_0 \) becomes

\[ b_0 = 33467.93525 \]

\[ \therefore F_{s1}(s) \] the desired low-order stabilization filter in (23.1) is
The unit step response of the existing stabilized pitch control system in Eq. (1.5) and the redesigned pitch control system using $F_{s1}(s)$ in (27) and the $G_0(s)$ in (1.3) are shown in Fig. 4. The result is fairly satisfactory.

An alternate approach for redesigning the stabilization filter by direct simplification of the existing stabilization filter is proposed as follows:

As it is mentioned at the beginning of this chapter, the function of the stabilization filter is to convert the dominant data at $\omega = 0$, $\omega_{eC} = 1.9$ rad/sec. and $\omega_{Ce} = 3.2$ rad/sec. of the original unstable system $C(s)$ in (3) to the assigned dominant data of $G_e(s)$ in (2). Taking advantage of this fact, we can directly apply the dominant-data matching method to fit a low-order stabilization filter that satisfies the specifications assigned in Eqn. (4). Let us assume that the desired low-order model of $F_{stab}(s)$ is

$$F_{s2}(s) = \frac{b_0 + b_1 s}{a_0 + a_1 s + s^2} \quad (28.1)$$

Applying the condition in (4.1) to $F_{s2}(s)$ in (28.1) yields

$$\text{Re}\{F_{s2}(j0)\} = \frac{b_0}{a_0} = 1.6$$

$$\therefore b_0 = 1.6a_0 \quad (28.2)$$

Substituting (28.2) into (28.1) gives
Figure 4. Time Responses of Various Models
\( F_{s2}(s) = \frac{1.6a_0+b_1 s}{a_0+a_1 s+s^2} \)  \hspace{1cm} (28.3)

At \( s = j\omega \)

\( F_{s2}(j\omega) = \frac{1.6a_0+j\omega b_1}{(a_0-\omega^2)+j\omega a_1} \)

\[ |F_{s2}(j\omega)| = \frac{\sqrt{2.56a_0^2+\omega^2 b_1^2}}{\sqrt{(a_0-\omega^2)^2+\omega^2 a_1^2}} \]  \hspace{1cm} (28.4)

\[ \phi_{F_{s2}}(j\omega) = \tan^{-1} \frac{\omega b_1}{1.6a_0} - \tan^{-1} \frac{\omega a_1}{a_0-\omega^2} \]

\[ = \tan^{-1} \frac{\omega b_1(a_0-\omega^2)-1.6\omega a_0 a_1}{1.6a_0(a_0-\omega^2)+\omega^2 a_1 b_1} \]  \hspace{1cm} (28.5)

At \( s = j\omega e^{\pi} = j1.9 \) the values of \( |F_{s2}(j\omega)| \) and \( \phi_{F_{s2}}(j\omega) \) in (28.4) and (28.5) respectively are matched to the corresponding values of \( |F_{stab}(j1.9)| \) together and \( \phi_{F_{stab}}(j1.9) \) in (4.3). Thus, we have

\[ |F_{s2}(j1.9)| = \frac{\sqrt{2.56a_0^2+3.61b_1^2}}{\sqrt{(a_0-3.61)^2+3.61a_1^2}} = 1.605107127 \]

or \( f_1(a_0,a_1,b_1) = 2.56a_0^2+3.61b_1^2-2.576368889[(a_0-3.61)^2 \]

\[ + 3.61a_1^2] = 0 \)  \hspace{1cm} (29.1)

and \( |F_{s2}(j1.9)| = \tan^{-1} \frac{1.9b_1(a_0-3.61)-3.04a_0 a_1}{1.6a_0(a_0-3.61)+3.61a_1 b_1} = 4.34591898 \)
or \[ f_2(a_0, a_1, b_1) = 1.9b_1(a_0-3.61)-3.04a_0a_1 \]
\[-0.0759963811[1.6a_0(a_0-3.61)+3.61a_1b_1]=0 \quad (29.2)\]

When \( s = j\omega_{ec} = j3.2 \) the value of \( F_{s2}(j\omega_{ec}) \) in (28.5) is compared with the value of \( F_{stab}(j\omega_{ec}) \) in (4.5). Thus, we have

\[
F_{s2}(j3.2) = \tan^{-1} \frac{3.2b_1(a_0-10.24)-5.12a_0a_1}{1.6a_0(a_0-10.24)+10.24a_1b_1} = 7.293349493^\circ
\]

or \[ f_3(a_0, a_1, b_1) = 3.2b_1(a_0-10.24)-5.12a_0a_1 \]
\[-0.1279849782[1.6a_0(a_0-10.24)+10.24a_1b_1]=0 \quad (29.3)\]

Using the initial guesses obtained in (26.6) the set of nonlinear equations in (29) is solved for the unknowns \( a_0, a_1 \text{ and } b_1 \) by using the Newton-Raphson method. The solutions are obtained at the 9th iteration with the error tolerance of \( 10^{-6} \). The solutions obtained are

\[
a_0 = 13301.999297 \\
a_1 = 3.318051 \\
b_1 = 856.628596
\]

Thus, the desired low-order model in (28.3) is

\[
F_{s2}(s) = \frac{856.628596s+21283.19886}{s^2+3.318051s+13301.999297}
\]

The unit-step response of the existing stabilized pitch control
system $T_c(s)$ in (1.5) and the redesigned system that uses the low-order
filter $F_{s2}(s)$ in (30) and $G_0(s)$ in (1.3) are shown in Fig. 4. The re-
sult is perfect. Comparing the unit-step response curves in Fig. 4, it
is clear that as far as the performance of the entire pitch control sys-
tem is concerned $F_{s2}(s)$ in (30) is a better filter than $F_{s1}(s)$ in (27).
This implies that the existing stabilization filter $F_{stab}(s)$ in (1.2)
might be overdesigned. Obviously, the implementation cost of the filter
$F_{s2}(s)$ is less than that of $F_{stab}(s)$ in (1.2).
CHAPTER V

REDESIGN OF THE STABILIZATION FILTER BY AN ALGEBRAIC METHOD

In Chapter IV of this report the original fourth-order stabilization filter $F_{stab}(s)$ has been simplified to two second-order filters, $F_{s1}(s)$ and $F_{s2}(s)$, using the dominant-data matching method discussed in Chapter II. It is noticed that all three stabilization filters, the original as well as its simplified models consist of complex poles. It is also observed that all three filters mentioned above are placed in the feed forward loop and as a result the system becomes very sensitive to external disturbances. If alternate filters can be designed and placed in both feed forward and feedback loops, i) the designed filters may turn out to be simple transfer functions with positive real roots and because of this it may be possible to synthesize the filters using passive elements, and ii) the performances of the designed system can be greatly improved. The fact that the fixed configuration of the compensators in the feedback loop enables the designed system to be insensitive to the parameter variations and modeling errors will reduce the effects of external disturbances and improve the stability of the system. Thus the redesigned feedback system has all the advantages [14] of feedback control systems.

In this chapter the algebraic method proposed by Shieh [3] and Chen [4] is extended and modified to redesign the pitch control system. The steps involved are summarized as follows.

**Step 1.** Assign the design goals using frequency-domain specifications and model a standard transfer function, known as the standard model,
using the dominant data matching method discussed in Chapter 11 of this thesis.

**Step 2.** Expand the standard model obtained in Step 1 into a standard fraction expansion of the second Cauer form by performing repeated long divisions as shown in (13.0) to obtain the dominant quotients. Using these quotients obtain the matrix \([H]\) in Equation (15).

**Step 3.** Assume the fixed configuration of compensators with unknown parameters and determine the overall transfer function of the system. Thus, the overall transfer function of the system will consist of the unknown parameters.

**Step 4.** Substitute the coefficients of the overall transfer function obtained in Step 3 into the vectors \([a]\) and \([b]\) in Equation (15) and expand the matrix equation (15) to obtain a set of equations.

**Step 5.** Solve the set of equations obtained in Step 4 to determine the unknown constants assigned in the compensators.

The designed system obtained by using the algebraic method has the exact dominant quotients of the standard model. In other words, the designed system is a good approximation of the standard model that has the exact assigned dominant data.

It is noticed that the original unstable system \(G_0(s)\) in (1.3) is a high order transfer function with large coefficients. Therefore, in order to simplify the procedure, before proceeding to design the Pitch control system by using the algebraic method, a reduced-order model of \(G_0(s)\) is determined by using the dominant-data matching method.
The unstable transfer function $G_0(s)$ in (1.3) can be decomposed into a stable function and an unstable portion as follows:

$$G_0(s) = \frac{1}{s(s-2.921)} T_0(s) \quad (31.1)$$

where the stable portion $T_0(s)$ is

$$T_0(s) = \frac{324332.316(s+0.1933)(s+65)(s+1500)}{(s+3.175)(s+87.9+j95.5)(s+112.5)(s+1385)} \quad (31.2)$$

The pole at the origin and the unstable pole at $s = 2.921$ are considered to be the dominant poles of the system. Therefore, they are retained in the simplified model $G_0^*(s)$ of $G_0(s)$, or

$$G_0(s) = G_0^*(s) = \frac{1}{s(s-2.921)} T_0^*(s) \quad (31.3)$$

where $T_0^*(s)$ is the reduced-order model of $T_0(s)$ obtained by using the dominant-data matching method. The frequency response data of $T_0(s)$ that are used as dominant data for the transfer function fitting are gain margin, phase margin, phase-crossover frequency, gain-crossover frequency, and the final value at $\omega = 0$. The $T_0^*(s)$ obtained is

$$T_0^*(s) = \frac{496.854897s^2+192897.961011s+37103.3375}{s^3+117.073733s^2+16552.300003s+50595.685093} \quad (31.4)$$

The $T_0^*(s)$ obtained is a low-order model of $T_0(s)$ with smaller coefficients. Thus, the design process can be greatly simplified.

Therefore $G_0^*(s) = \frac{496.854897s^2+192897.961011s+37103.3375}{s^5+114.152733s^4+16210.32763s^3+2246.41679s^2-147789.9961s}$ \quad (31.5)
Following the steps proposed at the beginning of this chapter the first step to design a system by the algebraic method is to determine the standard model. In this case the standard model $T_r(s)$ has been determined earlier in Chapter III and is given in (9). Writing $T_r(s)$ once again and expanding it in a continued fraction expansion yields

\[
T_r(s) = \frac{6.37807 + 20.55661s + 0.243466s^2}{6.37807 + 10.46222s + 1.259008s^2 + s^3}
\]

\[
= \frac{1}{h_1 + \frac{s}{h_2 + \frac{s}{h_3 + \frac{s}{h_4 + \frac{s}{h_5 + \frac{s}{h_6}}}}}}
\]

where

\[
\begin{align*}
h_1 &= 1 \\
h_2 &= -0.631845015 \\
h_3 &= -0.476189214 \\
h_4 &= 14.79958905 \\
h_5 &= -0.102867450 \\
h_6 &= -13.92427804 \\
\end{align*}
\]

In the next step a series compensator $G_1(s)$ and a parallel compensator $G_2(s)$ are assigned as shown in the block diagram of Fig. 5-1. $G_1(s)$
Fig. 5-1. The Block Diagram of a Redesigned System with Fixed Configuration Compensators

\[ G_1(s) = \frac{x_6s + x_7}{s + x_5} \]

\[ G_2(s) = \frac{x_3s^2 + x_4s + x_2}{s^2 + x_1s + x_2} \]

Fig. 5-2. The Modified Block Diagram of the Redesigned System

\[ R(s) \rightarrow R(s) \rightarrow G_1(s)G_2(s) \rightarrow G_0^*(s) \rightarrow \frac{1}{G_2(s)} \rightarrow Y(s) \]

Figure 5. The Block Diagrams of the Redesigned System Using Algebraic Method
and $G_2(s)$ are assumed with unknown parameters $x_i$, $i = 1, 2, \ldots, 7$ as

$$G_1(s) = \frac{x_6 s + x_7}{s + x_5} \quad (32.1)$$

and

$$G_2(s) = \frac{x_3 s^2 + x_4 s + x_2}{s^2 + x_1 s + x_2} \quad (32.2)$$

The overall transfer function $T_f(s)$ of the feedback system shown in Fig. 5-1 is

$$T_f(s) = \frac{b_0 + b_1 s + \ldots + b_7 s^7}{a_0 + a_1 s + \ldots + a_8 s^8} \quad (32.3)$$

where

$$a_0 = 37103.33375 x_2 x_7$$
$$a_1 = 192897.961011 x_2 x_7 + 37103.33375 (x_2 x_6 + x_4 x_7) - 147789.9961 x_5$$
$$a_2 = 496.854897 x_2 x_7 + 192897.961011 (x_2 x_6 + x_4 x_7)$$
$$+ 37103.33375 (x_4 x_6 + x_3 x_7) + 2246.41679 x_2 x_5$$
$$- 147789.9961 (x_2 + x_1 x_5)$$
$$a_3 = 496.854897 (x_2 x_6 + x_4 x_7) + 192897.961011 (x_4 x_6 + x_3 x_7)$$
$$+ 37103.33375 x_3 x_6 - 147789.9961 (x_1 + x_5)$$
$$+ 2246.41679 (x_2 + x_1 x_5) + 16210.32763 x_2 x_5$$
$$a_4 = 496.854897 (x_4 x_6 + x_3 x_7) + 192897.961011 x_3 x_6 - 147789.9961$$
$$+ 114.152733 (x_1 + x_5) + 16210.32763 (x_2 + x_1 x_5) + 114.152733 x_2 x_5$$
$$a_5 = 496.854897 x_3 x_6 + 2246.41679 + 16210.32763 (x_1 + x_5)$$
$$+ 114.152733 (x_2 + x_1 x_5) + x_2 x_5$$
$$a_6 = 16210.32763 + 114.152733 (x_1 + x_5) + x_2 x_5$$
\[ a_7 = 114.152733 + x_1^2 + x_5 \]
\[ a_8 = 1 \]
\[ b_0 = 37103.3375x_2x_7 \]
\[ b_1 = 192897.961011x_2x_7 + 37103.3375(x_2x_6 + x_1x_7) \]
\[ b_2 = 496.854897x_2x_7 + 192897.961011(x_2x_6 + x_1x_7) \]
\[ + 37103.3375(x_1x_6 + x_7) \]
\[ b_3 = 496.854897(x_2x_6 + x_1x_7) + 192897.961011(x_1x_6 + x_7) + 37103.3375x_6 \]
\[ b_4 = 496.854897(x_1x_6 + x_7) + 192897.961011x_6 \]
\[ b_5 = 496.854897x_6 \]
\[ b_6 = 0 \]
\[ b_7 = 0 \]

In order to match the seven unknown parameters, \( x_i \), \( i = 1, 2, \ldots, 7 \) in (32) for this type '1' system we need eight quotients \( h_i, i=1,2,\ldots,8 \) in (9). Therefore, the third order standard model in (9) with the quotients \( h_i \) given in (31.6) has to be amplified to a fourth-order system. This is done by inserting \( h_7 = 100 \) and \( h_8 = 0.1 \) as shown below.

\[
T_r(s) = \frac{6.37807 + 20.55661s + 0.243466s^2}{6.37807 + 10.46222s + 1.259008s^2 + s^3}
\]
\[ = \frac{1}{h_1 + \frac{s}{h_2} + \frac{s}{h_3} + \frac{s}{h_4} + \frac{s}{h_5} + \frac{s}{h_6}} \]
\[ T_a(s) = \frac{1}{63.78098007 + 211.8989926s + 22.87561717s^2 + 0.34346s^3} \]

It has been shown that \( T_a(s) \) in (33) is a good approximation of the original standard model \( T(s) \) in (9).

Substituting the \( h_i, i = 1,\ldots,6 \) in (31.6) including \( h_7 = 100 \) and \( h_8 = 0.1 \), the matrices \( [H_1] \) and \( [H_2] \) in (5) are obtained next.
\[
H_1 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.6318422 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0.30087727 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 14.167729 & 4.4528546 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1.1565286 & -0.45805621 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.24346000 & 20.55667 & 6.378098 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 23.189471 & 2055.2089 & 637.8098 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.34346 & 22.875617 & 211.89899 & 63.78098 & 0 \\
\end{bmatrix}
\]

\[
H_2 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.63184224 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.52380951 & 0.30087727 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 7.1203141 & 4.4528546 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.42094152 & -0.4315751 & -0.45805 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1.259011 & 10.462223 & 6.378098 & 0 & 0 & 0 \\
0 & 0 & 0 & 100.42094 & 125.6695 & 1045.7642 & 637.8098 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 11.301105 & 23.009176 & 110.95452 & 63.78098 \\
\end{bmatrix}
\]
Substituting the unknown constants \( a_i, i = 0, 1, \ldots, 7 \) and \( [H_1] \) and \( [H_2] \) obtained above into (15) yields a set of equations in terms of \( a_i \) and \( b_i \) as follows.

\[
[b] = [H_1][a] = [H_2]^{-1} [H_1] [a]
\]
or

\[
[H_2][b] = [H_1][a]
\]

or expanding the above matrix equation yields

\[
\begin{align*}
\mathbf{f}_1(a_i, b_i) &= a_0 - b_0 = 0 \\
\mathbf{f}_2(a_i, b_i) &= b_0 - 0.631842396(b_1 - a_1) = 0 \\
\mathbf{f}_3(a_i, b_i) &= a_1 + 0.30087727(a_2 - b_2) - 0.52380951b_1 = 0 \\
\mathbf{f}_4(a_i, b_i) &= b_1 + 7.1203141b_2 + 4.4528546(b_3 - a_3) - 14.167729a_2 = 0 \\
\mathbf{f}_5(a_i, b_i) &= a_2 - 1.1565286a_3 - 0.45805621(a_4 - b_4) - 0.42094152b_2 \\
&\quad + 0.4315751b_3 = 0 \\
\mathbf{f}_6(a_i, b_i) &= b_2 + 1.259011b_3 + 10.462223b_4 + 6.378098(b_5 - a_5) \\
&\quad - 0.24346000a_3 - 20.55667a_4 = 0 \\
\mathbf{f}_7(a_i, b_i) &= a_3 + 23.189471a_4 + 2055.2089a_5 + 637.8098(a_6 - b_6) \\
&\quad - 100.42094b_3 - 125.4695b_4 - 1045.7642b_5 = 0 \\
\mathbf{f}_8(a_i, b_i) &= b_3 + 11.301105b_4 + 23.009176b_5 + 110.95452b_6 \\
&\quad + 63.78098(b_7 - a_7) - 0.34346a_4 - 22.875617a_5 - 211.89899a_6 = 0
\end{align*}
\]

where \( i = 0, 1, \ldots, 7 \).
Now, substituting the values of $a_i$ and $b_i$ in terms of $x_i$, $i = 1, 2, ..., 7$ from (32.4) yields a set of nonlinear equations shown below. It is noticed that as $a_0 = b_0$ the equation $f_1(a_i, b_i) = a_0 - b_0 = 0$ gives no information. The rest of the equations are

$$f_1(x_1, ..., x_7) = x_2 x_7 + 0.6318422396 [x_7 (x_4 - x_1) - 3.98319992 x_2 x_5] = 0$$

(33.1)

$$f_2(x_1, ..., x_7) = x_7 (8.22822291 x_2 + 8.522553136 x_4 - 6.939879587 x_1$$

+ $x_3 - 1) + x_2 (1.582676549 x_6 - 13.17807554 x_3$$

+ $x_6 (x_4 - x_1) - 3.98319992 (x_2 + x_1 x_5) = 0$

(33.2)

$$f_3(x_1, ..., x_7) = x_2 (-12.71361621 x_6 - x_5) + x_7 (13.58355291 x_1$$

+ $1.820964317 x_2 - 26.29716913 x_4) + 10.79844539 (x_1 x_6 + x_7)$

- $13.31248704 (x_4 x_6 + x_3 x_7) + 6.327224282 (x_1 + x_5)$

+ $1.588477708 x_6 (1 - x_3) + 20.03527143 (x_2 + x_1 x_5) = 0$

(33.3)

$$f_4(x_1, ..., x_7) = x_7 (x_2 + 668.4670071 x_4 - 281.48094 x_1) + x_6 (386.9860673 x_2$$

+ $362.767005 - 456.258273 x_3) - 647.2403649 (x_4 x_6 + x_3 x_7)$

- $57.53603068 x_2 x_5 - 548.5188427 (x_2 + x_1 x_5)$

+ $235.861385 (x_1 x_6 + x_7) + 235.2945185 + 590.5096275 (x_1$$

+ x_5) = 0$

(33.4)

$$f_5(x_1, ..., x_7) = 2357.408023 (x_1 x_6 + x_7) + x_6 (1598.839931 x_2 + 17096.15228$$

- $32881.95043 x_3) + x_7 (4.16745091 x_2 + 1599.839931 x_1 - x_4)$

- $472.673532 (x_4 x_6 + x_3 x_7) + 24996.98242 - 9.39.0287936 (x_1$$

+ x_5) - 2765.323026 (x_2 + x_1 x_5) - 52.07771943 x_2 x_5 = 0$

(33.5)
Equation (33) is a set of high order nonlinear simultaneous equations which is very difficult to solve. However, with proper initial guesses the Newton-Raphson [15] method can be applied to solve it. Therefore, the problem lies in finding an appropriate set of initial guesses. In this case, the following method is suggested for estimating the initial guesses.

As mentioned earlier, the block diagram of the structure of the desired fixed configuration control system is shown in Fig. 5-1. Without affecting the overall transfer function of the system, this structure can be modified into a form as shown in Fig. 5-2. The overall transfer function of this structure is

\[
T_1(s) = T_2(s) \frac{1}{G_2(s)}
\]

where

\[
T_2(s) = \frac{G_1(s)G_2(s)G_0^*(s)}{1+G_1(s)G_2(s)G_0^*(s)}
\]
Where \( G_0^*(s) \), the approximate transfer function of \( G_0(s) \), is given in (31.5).

The purpose is to determine \( G_1(s) \) and \( G_2(s) \) such that the response of \( T_1(s) \) becomes close to that of the standard model \( T_r(s) \) in (9). Replacing the series compensator \( G_1(s)G_2(s) \) in Fig. (5-2) by the designed stabilization filter \( F_{s2}(s) \) in (30) the resulting transfer function \( T_1(s) \) in (34.1) is equated to the standard model \( T_r(s) \) in (9) as follows.

\[
T_1(s) \quad \text{force} \quad T_r(s)
\]

or

\[
T_2(s) \frac{1}{G_2(s)} = T_r(s)
\]

or

\[
G_2(s) = \frac{T_2(s)}{T_r(s)}
\]

or

\[
G_2^*(s) = G_2(s) = \frac{1}{T_r(s)} \left[ \frac{G_1(s)G_2(s)G_0^*(s)}{1+G_1(s)G_2(s)G_0^*(s)} \right]
\]

or

\[
G_2^*(s) = G_2(s) = \frac{1}{T_r(s)} \left[ \frac{F_{s2}(s)G_0^*(s)}{1+F_{s2}(s)G_0^*(s)} \right]
\]

\[
= \frac{1}{T_r(s)} \left[ \frac{789677630.6+4137269440s}{789677630.6+2171367017s+175816571.1s^2+425620.1128s^3+205208030.9s^2+215915050.5s^3+154492.684s^4+29891.09151s^5+117.470784s^6+s^7} \right]
\]

\[
(34.2)
\]
Substituting $T_r(s)$ in (9) into (34.2) and simplifying, the appropriate transfer function $G_2^*(s)$ of $G_2(s)$ is obtained.

$$G_2^*(s) = \frac{5.036619205 \times 10^9 + 3.46495752 \times 10^{10}s + 4.540060393 \times 10^{12}s^2}{5.036619205 \times 10^9 + 3.008227329 \times 10^{10}s + 4.613716606 \times 10^{12}s^2}$$

$$+ 7.840679235 \times 10^9s^3 + 4.363076841 \times 10^9s^4 + 1.763524302 \times 10^8s^5 + 6.124169121 \times 10^9s^3 + 4.498497844 \times 10^9s^4 + 8.512494768 \times 10^7s^5$$

$$+ 4.256201128 \times 10^5s^6 + 9.985459768 \times 10^5s^6 + 9.698650697 \times 10^3s^7 + 49.1568119s^8 + 0.243466s^9$$

$$= \frac{1}{h_1}$$

A set of dominant quotients $h_i$ of $G_2^*(s)$, given below, are determined by expanding (34.3) into a continued fraction of the second Cauer form

$$h_1 = 1$$
$$h_2 = -1.102755917$$
$$h_3 = -0.128794873$$
$$h_4 = 5.593229805$$
$$h_5 = 0.1338916858$$
$$h_6 = \ldots$$
$$\vdots$$
$$h_{18} = \ldots$$

Substituting the first five quotients $h_1, h_2, \ldots, h_5$ into (34.5) gives a second order approximate model $G_2^{**}(s)$ of the approximate parallel filter $G_2^*(s)$ in (34.3) as
\(G(s)\), the approximate model of \(G_2(s)\), is also an approximate model of \(G_2(s)\) in (32.2).

The approximate model \(G_1(s)\) of the series compensator \(G_1(s)\) in Fig. 5-1 can be obtained as follows

\[
G_1(s) = \frac{F_{s2}(s)}{G_{s2}(s)} = \frac{856.628596s^3 + 21834.46821s^2 + 13787.1076s + 0.994929057s + 4.040722745s^3}{0.994929057s + 4.040722745s^3 + 2252.284999 + 13237.10512s^2 + 9837.144919s + 1407.678125}
\]  

(35.1)

To obtain a set of dominant quotients Equation (35.1) is expanded into a continued fraction of the second Cauer form. Some of the quotients obtained are

\[h_1 = 0.625\]
\[h_2 = 1.845828612\]
\[h_3 = 0.0839039052\]
\[h_4 = \ldots\]
\[\vdots\]
\[h_8 = \ldots\]

(35.2)

The first three quotients \(h_1, h_2, h_3\) are substituted into (13.3), which gives \(G_{s2}(s)\), an approximate model of \(G_1(s)\) as well as of \(G_1(s)\) in (32.1). \(G_{s2}(s)\) obtained is
Comparing (32.2) with (34.5) and (32.1) with (35.2) we have a set of initial guesses to solve the set of high order nonlinear equations in (33). Thus, the set of initial guesses is

\[
\begin{align*}
  x_1^* &= 0.643533679 \\
  x_2^* &= 0.1058245527 \\
  x_3^* &= 0.994929057 \\
  x_4^* &= 0.7394973923 \\
  x_5^* &= 0.1306419803 \\
  x_6^* &= 1.410628426 \\
  x_7^* &= 0.2184671685
\end{align*}
\]  

Using these initial guesses the Newton-Raphson method [15] is applied to solve the nonlinear equations in (33). It is found that the Newton-Raphson method converges to the desired solutions, given below, at the 14th iteration with the error tolerance of $10^{-6}$. The solutions are

\[
\begin{align*}
  x_1 &= 0.503850 \\
  x_2 &= 0.059928 \\
  x_3 &= 1.051503 \\
  x_4 &= 0.580016
\end{align*}
\]
\[ x_5 = 4.831826 \\
x_6 = 1.885577 \\
x_7 = 6.744450 \]

Therefore, the desired compensators \( G_1(s) \) and \( G_2(s) \) are

\[
G_1(s) = \frac{1.885577s + 6.744450}{s + 4.831826} = \frac{1.885577(s + 3.57688)}{s + 4.831826} \quad (37.1)
\]

and

\[
G_2(s) = \frac{1.051503s^2 + 0.580016s + 0.059928}{s^2 + 0.503850s + 0.059928} = \frac{1.051503(s + 0.13769)(s + 0.41391)}{(s + 0.19244)(s + 0.311405)} \quad (37.2)
\]

The unit step response curves of the existing stabilized system \( T_e(s) \) in (1.5) and the redesigned system using the compensators \( G_1(s) \) and \( G_2(s) \) in (37), and \( G_0(s) \) in (1.3), are compared in Fig. 4. The result is satisfactory.

It is interesting to note that \( G_1(s) \) and \( G_2(s) \) in Eq. (37) are positive real functions with positive real poles and zeros, which makes it possible to realize the compensators \( G_1(s) \) and \( G_2(s) \) using passive elements, whereas, the existing stabilization filter \( F_{\text{stab}}(s) \) is a non-positive real function and it is realized by using active elements.
CHAPTER VI

CONCLUSION

The existing stabilized pitch control system has been redesigned by redesigning the existing stabilization filter. Two computer-oriented methods, a dominant data matching method and an algebraic method, have been presented to redesign the existing stabilization filter. Thus, various low-order stabilization filters have been obtained. As a result, the implementation cost of the missile system is reduced.

The proposed dominant-data matching method can be used for various purposes. For example, when the specifications or the design goals of a control system are given, the proposed method can be used to obtain a standard transfer function, which significantly simplifies the design process in the frequency domain. When a high-order transfer function is given, various low-order models can be obtained with the help of the dominant-data matching method. The method can be used in the problems of identification as well. The great advantage of this method is that the transfer functions obtained by using this method have the exact assigned frequency-domain specifications.

The algebraic method has been applied to achieve the advantages of the feedback control system so that the performances of the redesigned pitch control system can be greatly improved.

The application of the dominant-data matching method always gives rise to a set of nonlinear equations which can be solved if a set of proper initial guesses is known. In this connection, various methods
have been discussed for estimating a set of proper initial guesses.

Finally, it is important to mention that the proposed computer-
aided design methods can be used to design general control systems.
REFERENCES


A matrix in the block Schwarz form and the stability of matrix polynomials

LEANG-SAN SHIEH† and SHAILENDRA SACHET†

A matrix which consists of block elements is established in the block Schwarz form via a linear transformation. The transformation matrix constructed by Chen and Chu is modified and extended for transforming the block companion form to the block Schwarz form. A sufficient condition is derived for determining the stability of a multivariable system whose characteristics are expressed by a matrix polynomial. The matrix polynomial may or may not be symmetric.

1. Introduction

The properties and applications of the Schwarz matrix, which has scalar elements, has been investigated by various authors (Schwarz 1956, Parks 1963, Wall 1948, Anderson et al. 1976, Barnett and Storey 1970), and the transformation matrix, which relates various matrix forms and the Schwarz form, has also been established by numerous authors (Butchart 1965, Chen and Chu 1966, 1967, Barnett and Storey 1967, Loo 1968, Power 1969, Datta 1974, Sarma et al. 1968). Chen and Chu (1966, 1967) constructed a transformation matrix which links the Schwarz form and the companion form by using the Routh array elements (Routh 1877). However, all existing methods (Schwarz 1956, Parks 1963, Wall 1948, Anderson et al. 1976, Barnett and Storey 1967, 1970, Butchart 1965, Chen and Chu 1966, 1967, Loo 1968, Power 1969, Datta 1974, Sarma et al. 1968) deal only with the system matrix which has scalar elements but not block elements. In this paper a matrix which consists of block elements is constructed in the block Schwarz form and a linear transformation matrix which consists of block elements is established to transform the matrix in the block companion form (Shieh 1975) (the block phase variable form) to the block Schwarz form. A sufficient condition is then derived to determine the stability of a multivariable system whose characteristics are described by a matrix polynomial (Shieh 1975, Shieh et al. 1976). The matrix polynomial may or may not be symmetric.

2. A transformation for a matrix in the Schwarz block form

Consider a set of linear, time-invariant, ordinary differential equations in the differential matrix polynomial form

\[ \sum_{i=1}^{++} B_i D^{i-1} X(t) = [0], \quad B_{n+1} = I \quad (1\ a) \]

\[ D^{i-1} X(0) = [\alpha_{i-1}], \quad i = 1, 2, \ldots, n \quad (1\ b) \]

where \( X(t) \) is the \( m \)-dimensional state of the system. The \( B_i \) are \( m \times m \) real constant matrices and the differential operator \( D \) is \( D = d/dt \). The matrix \( I \)

Received 20 October 1976.
† Department of Electrical Engineering, University of Houston, Houston, Texas 77004, U.S.A.
is an identity matrix and $[0]$ is a null matrix. The corresponding state equation of eqn. (1) in the block companion form is

$$\begin{align*}
[\dot{x}] &= [B][x] \tag{2a} \\
[x(0)] &= [a] \tag{2b}
\end{align*}$$

where

$$[B] = \begin{bmatrix}
0 & I & 0 & 0 & 0 \\
0 & 0 & I & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & I \\
-B_1 & -B_2 & -B_3 & -B_4 & -B_n
\end{bmatrix}$$

The dimensions of the matrix $[B]$, each block element in the $[B]$, and the state vector $[x]$ are $(n \times m) \times (n \times m)$, $m \times m$, and $(n \times m) \times 1$, respectively. The $[B]$ is the matrix in the block companion form or the block phase variable form (Shieh 1975).

Equation (2) can be transformed into the block Schwarz form by using the following linear transformation $[K_1]$

$$[y] = [K_1][x] \tag{3}$$

and

$$[y] = [K_1][B][K_1]^{-1}[y] = [A][y] \tag{4}$$

where

$$[K_1] = \begin{bmatrix}
I & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
C_{n-1,1}^{-1} C_{n-1,2} & I & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & I & 0 & 0 & 0 & 0 \\
C_{n-3,1}^{-1} C_{n-3,2} & C_{61}^{-1} C_{42} & 0 & I & 0 & 0 & 0 \\
0 & 0 & C_{51}^{-1} C_{52} & 0 & I & 0 & 0 \\
C_{n-5,1}^{-1} C_{n-5,2} & C_{41}^{-1} C_{43} & 0 & C_{41}^{-1} C_{42} & 0 & I & 0 \\
0 & 0 & C_{31}^{-1} C_{32} & 0 & C_{31}^{-1} C_{33} & 0 & I \\
\vdots & \vdots & C_{21}^{-1} C_{24} & 0 & C_{21}^{-1} C_{23} & 0 & C_{21}^{-1} C_{22} & 0 & I
\end{bmatrix}$$
A matrix in the block Schwarz form

and

\[
[A] = \begin{bmatrix}
0 & I & 0 & 0 & 0 \\
-A_1 & 0 & I & 0 & 0 \\
0 & -A_2 & 0 & I & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & I \\
0 & 0 & 0 & 0 & -A_{n-1} - A_n
\end{bmatrix}
\]

The dimension of each block element in the matrix \([A]\) and the matrix \([K_1]\) is \(m \times m\). The \([A]\) is the matrix in the block Schwarz form. The linear transformation matrix \([K_1]\), which relates the coordinates \([x]\) and \([y]\) in eqns. (2) and (4), is constructed by following the approach proposed by Chen and Chu (1966). The block elements \(C_{i,j}\) having dimension \(m \times m\) in eqn. (3) can be obtained from the following matrix Routh algorithm and the matrix Routh array (Shieh and Gaudiano 1974, Shieh et al. 1976).

Before performing the matrix Routh array we define \(l = (n/2) + 1\) if \(n\) is an even number; otherwise, \(l = n + 1/2\), and the double subscripted block elements \(C_{1,j}\) and \(C_{2,j}\) become:

\[
\begin{align*}
C_{1,j} &= B_{n+3-2j}, \quad j = 1, 2, 3, \ldots, l \\
C_{2,j} &= B_{n+3-2j} - 2, \quad j = 1, 2, 3, \ldots, l \\
C_{11} &= I
\end{align*}
\]

(5 a)

The \(B_i\) are the \(m \times m\) real constant matrices shown in eqn. (1). The matrix Routh array and the matrix Routh algorithm are

\[
\begin{align*}
H_1 &= C_{11}C_{21}^{-1} \quad C_{21} \quad C_{22} \quad C_{23} \quad C_{24} \quad \ldots \\
H_2 &= C_{21}C_{31}^{-1} \quad C_{31} \quad C_{32} \quad C_{33} \quad \ldots \\
H_3 &= C_{31}C_{41}^{-1} \quad C_{41} \quad C_{42} \quad C_{43} \quad \ldots \\
H_4 &= C_{41}C_{51}^{-1} \quad C_{51} \quad C_{52} \quad \ldots \\
H_5 &= C_{51}C_{61}^{-1} \quad C_{61} \quad C_{62} \quad \ldots \\
H_6 &= C_{61}C_{71}^{-1} \quad C_{71} \quad \ldots \\
H_7 &= C_{71}C_{81}^{-1} \quad C_{81} \quad \ldots \\
H_8 &= C_{81}C_{91}^{-1} \quad C_{91} \quad \ldots \\
H_9 &= \ldots \\
H_n &= C_{n,1}C_{n+1,1}^{-1} \quad C_{n,1} \\
\end{align*}
\]

(5 b)

where

\[
H_l = C_{l-2,l-1} - H_{l-2}C_{l-3,l-1}, \quad j = 1, 2, \ldots, \quad i = 3, 4, \ldots
\]

\[
H_l = C_{l,1}C_{l+1,1}^{-1}, \quad i = 1, \ldots, n
\]

The matrices \(H_l\) in eqn. (5 b) are called the matrix quotients. Performing a new linear transformation

\[
[z] = [K_2][y]
\]

(6)
on eqn. (4) yields an alternative matrix \([F]\) in the block Schwarz form as follows:

\[
[z] = [K_4][K_2][z] = [F][z]
\]

(7 a)

where

\[
[K_2] = \begin{bmatrix}
    C_{n,1} & 0 & \cdots & 0 \\
    0 & C_{n-1,1} & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & C_{11}
\end{bmatrix}
\]

(7 b)

and

\[
[F] = \begin{bmatrix}
    0 & H_{n-1}^{-1} & 0 & \cdots & 0 & 0 & 0 \\
    -H_n^{-1} & 0 & H_{n-2}^{-1} & \cdots & 0 & 0 & 0 \\
    0 & -H_{n-1}^{-1} & 0 & \cdots & 0 & 0 & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
    0 & 0 & 0 & \cdots & H_3^{-1} & 0 & 0 \\
    0 & 0 & 0 & -H_4^{-1} & 0 & H_3^{-1} & 0 \\
    0 & 0 & 0 & \cdots & -H_5^{-1} & 0 & H_4^{-1} \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots \\
    0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots \\
    0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots \\
    0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix}
\]

(7 c)

The \([K_2]\) is a block diagonal matrix having the diagonal block elements obtained from the block elements \(C_{i,1}, i = 1, 2, \ldots\), which are in the first column of the matrix Routh array in eqn. (5 b), while the matrix \([F]\) is the required matrix in the block Schwarz form which can be constructed by using the matrix quotients \(H_i, i = 1, 2, \ldots\), obtained from the same matrix Routh array. A similar matrix (Schwarz and Friedland 1965), which was formulated in the Schwarz form but not in the block Schwarz form, was used to prove the stability of a linear system by Parks (1963).

The linear transformation matrix \([K]\), which links the coordinates \([x]\) in the block companion form and the coordinates \([z]\) in the block Schwarz form, is

\[
[z] = [K_4][K_1][x] = [K][x]
\]

(8 a)
A matrix in the block Schwarz form

where

\[
[K] = \begin{bmatrix}
H_n C_{n+1,1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \ddots & 0 & 0 & 0 & 0 & 0 \\
H_n C_{n+1,2} & H_n C_{n-1,2} & 0 & 0 & 0 & 0 & 0 \\
0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\
H_n C_{n-2,3} & H_n C_{n-3,3} & 0 & 0 & 0 & 0 & 0 \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
H_n C_{n-4,5} & H_n C_{n-5,5} & 0 & 0 & 0 & 0 & 0 \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & H_n C_{n-1,1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & H_n C_{n-1,2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & H_n C_{n-1,3} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & H_n C_{n-1,4} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & H_n C_{n-1,5}
\end{bmatrix}
\]

The matrix \([K]\) is a block triangular matrix. All the block elements in eqn. (8b) can be directly obtained from the matrix Routh array in eqn. (5b). For example, the block elements in the main diagonal, which are shown by the first dotted diagonal line, are obtained by the respective products of the matrix quotients \(H_i\) and block elements \(C_{i,1}\) in the first column of the matrix Routh array. The block elements of the first lower diagonal in the \([K]\) are null matrices, and the block elements of the second lower diagonal in the \([K]\), which are shown by the second dotted diagonal line, are determined by the respective products of the matrix quotients \(H_i\) and the block elements \(C_{i,2}\) in the second column of the same matrix Routh array, etc. The sizes of the matrices \([F]\) and \([K]\) are determined by the degree of the matrix polynomial and the order of the matrix coefficients in eqn. (1). For instance, when the degree of a matrix polynomial is \(n=4\) and the dimension of each matrix coefficient is \(m\), the corresponding \(4m \times 4m\) square matrices \([F]\) and \([K]\) are taken from the right-hand side lower corner of the matrices \([F]\) and \([K]\) in eqns. (7c) and (8b).

3. A sufficient condition for the stability of a matrix polynomial

In a single variable system the Routh criterion (Routh 1877) is applied to the characteristic polynomial of a linear system for determining the absolute stability. In other words, the scalar polynomial in the form of eqn. (1) is arranged in the Routh array in eqn. (5b), then the Routh criterion is applied. If the scalars \(C_{i,1}\) in the first column of the Routh array have no sign changes or all elements \(C_{i,1}, i = 1, 2, \ldots\), are positive real, then the system is asymptotically stable. Since the Routh algorithm and the Routh array have been successfully extended to the matrix Routh algorithm and the matrix Routh array (Shieh and Gaudiano 1974, Shieh et al. 1976), also a positive definite matrix (Bellman 1970) is commonly considered as a natural generalization of a positive number,
it is interesting to ask whether or not a multivariable system whose characteristic matrix polynomial shown in eqn. (1) is asymptotically stable if the block elements \( C_{ii}, i = 1, 2, \ldots, \) in the first column of the matrix Routh array in eqn. (5 b) are positive definite matrices. In other words, can we directly extend the Routh criterion (Routh 1877) to the matrix Routh criterion?

This paper will partially answer this question.

Because the stability of a system is invariant under the transpose operation of the system matrix, we consider the following system:

\[
[q] = [F^T][q] = [G][q] \tag{9}
\]

The matrix \([F]\) in eqn. (9) is defined in eqn. (7) and the transpose of the matrix \([F]\) is defined as \([G]\). If the matrix quotients \(H_i\) in eqn. (5 b) are positive-definite symmetric and real matrices, then we can consider the following quadratic equation (Kalman and Bertram 1960, Barnett 1971):

\[
v = [q^T][P][q] \tag{10 a}
\]

where

\[
[P] = \begin{bmatrix}
H_n & 0 & \cdots & 0 & 0 \\
0 & H_{n-1} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & H_2 & 0 \\
0 & 0 & \cdots & 0 & H_1
\end{bmatrix}
\]

The derivative function \(\dot{v}\) is

\[
\dot{v} = [q^T][PG + GTP][q] = -[q^T][Q][q] \tag{10 b}
\]

where

\[
[P][G] = \begin{bmatrix}
0 & -I & 0 & \cdots & 0 & 0 \\
I & 0 & -I & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & -I \\
0 & 0 & \cdots & I & -I
\end{bmatrix},
\]

\[
[Q] = \begin{bmatrix}
0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0
\end{bmatrix}
\]

The \(v\) function in eqn. (10 a) is in a positive-definite quadratic form and the \(\dot{v}\) function in eqn. (10 b) is in a negative-semidefinite form. Therefore the system in eqn. (9) or in eqn. (1) is asymptotically stable. From the above derivation we obtain the sufficient condition that a multivariable system, whose characteristic matrix polynomial has the form shown in eqn. (1), is asymptotically stable if the matrix quotients \(H_i\) in eqn. (5 b) are real symmetric positive-definite matrices. From eqns. (2 a) and (7 c) it can be observed that the \(B_n(=H_1^{-1}=C_{31})\) must be symmetric and positive-definite for the sufficient
A matrix in the block Schwarz form

Condition to apply. It is also noted that this sufficient condition deals only with \( H_i \) and not \( C_{i,j} \). This implies that, if a system is asymptotically stable, the block elements \( C_{i,j} \), \( i = 1, 2, \ldots \), in the first column of the matrix Routh array and the \( B_i \) in eqn. (1) are not necessarily symmetric and positive-definite matrices. In other words, a positive-definite matrix is not necessarily a natural generalization of a positive number, and the necessary and sufficient condition of the Routh criterion (Routh 1877) cannot be completely extended to the matrix Routh criterion for general cases.

On the other hand, the necessary conditions for asymptotic stability due to the Routh criterion (Routh 1877) can be partially extended to the case of matrix polynomials. The necessary conditions are described as follows:

(i) The determinant of \( B_1 \) in eqn. (1) is non-zero.
(ii) The determinants of \( B_{i+1} \) and \( B_1 \) in eqn. (1) have the same sign if the determinant of \( B_{i+1}(=C_{1i}) \) is non-zero.

These conditions can be easily verified from the basic laws of algebra. Thus, in this paper, we have partially extended the Routh criterion (Routh 1877) to the matrix Routh criterion for determining the asymptotic stability of a class of matrix polynomials.

Sometimes in applying the approach proposed in this paper difficulties may be encountered in calculating the matrix quotients \( H_i \) in eqn. (5b). This implies that if any \( C_{1,i} \) in eqn. (5b) is singular, then the \( H_i \) cannot be obtained to determine the stability of a matrix polynomial. This limitation can be remedied by multiplying the original matrix polynomial, defined as \( T(s) \), by a polynomial matrix defined as \( E(s) \), where the roots of the determinant \( E(s) \) have negative real parts. Then we apply the matrix Routh procedure to the modified matrix polynomial \( T(s)E(s) \) or \( E(s)T(s) \). It is noted that the stability of the original system is reserved because the stability of a system is invariant under this modification. An alternative way is to perform transformation \( s \rightarrow 1/s \) to the \( T(s) \) and then applying the matrix Routh procedure to the modified matrix polynomial defined as \( T(1/s) = T(s)|_{s \rightarrow 1/s} \). In other words, the \( C_{1,i} \) and \( C_{2,j} \) in eqn. (5a) are rewritten as follows:

\[
C_{1,i} = B_{j,i-1} \quad \text{for} \quad j = 1, 2, 3, \ldots \\
C_{2,j} = B_{2j} \quad \text{for} \quad j = 1, 2, 3, \ldots
\]

Again, the stability of the original system is invariant to this modification and the numerically ill-conditioned problem (i.e. \( C_{1,i} \) is singular) can be solved. Examples are established in this paper to show the procedure.

4. Illustrative examples
4.1. Example 1

Consider the following differential matrix polynomial:

\[
\sum_{i=1}^{n+1-5} B_i D^{i-1} X(t) = [0] \quad (11)
\]

or

\[
B_5 X(t) + B_4 X(t) + B_3 \ddot{X}(t) + B_2 \dot{X}(t) + B_1 X(t) = C_{11} X(t) + C_{21} X(t) + C_{12} \ddot{X}(t) + C_{22} \dot{X}(t) + C_{13} X(t) + C_{13} X(t) = [0] \quad (11)
\]
where

\[
C_{11} = B_5 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},
\]

\[
C_{12} = B_3 = \begin{pmatrix} -37.05 & -78.8 \\ 33 & 65 \end{pmatrix},
\]

\[
C_{13} = B_1 = \begin{pmatrix} -10.5 & -23 \\ -0.1 & -0.6 \end{pmatrix},
\]

\[
C_{21} = B_4 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix},
\]

\[
C_{22} = B_2 = \begin{pmatrix} -43.1 & -94.6 \\ -6.05 & -16.3 \end{pmatrix}.
\]

A matrix in the block Schwarz form and the linear transformation matrix which transforms the block companion form to the block Schwarz form are required to be constructed, and the stability of the system is of interest.

Arranging the matrices \(B_i\) in eqn. (11) in the matrix Routh array in eqn. (5b) results in the following:

\[
H_1 = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix},
\]

\[
H_2 = \begin{pmatrix} 4 & 1 \\ 1 & 0.5 \end{pmatrix},
\]

\[
H_3 = \begin{pmatrix} 1.125 & 0.25 \\ 0.25 & 0.5 \end{pmatrix},
\]

\[
H_4 = \begin{pmatrix} 0.1 & -0.5 \\ -0.5 & 7.5 \end{pmatrix}.
\]

The matrix quotients \(H_{ij}, i = 1, 2, ..., 4, \) in eqn. (12) are positive-definite symmetric real matrices; therefore the system is asymptotically stable. It is
A matrix in the block Schwarz form

noted that the block elements $C_{ij}$, $i = 1, \ldots, 5$ in the first column of the matrix Routh array in eqn. (12) are not all positive-definite symmetric real matrices. The state equation in the block companion form is

$$[\dot{z}] = [B][x]$$  \hspace{1cm} (13)$$

where

$$[B] = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
10.5 & 23 & 43.1 & 94.6 & 37.05 \\
0.1 & 0.6 & 6.05 & 16.3 & 33 & 65 & -2 & -1
\end{bmatrix}$$

and the state equation in the block Schwarz form is

$$[\dot{z}] = [F][x]$$  \hspace{1cm} (14 a)$$

where

$$[F] = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.5 & 2.25 & 0 \\
-15 & -1 & 0 & 0 & 0 \\
-1 & -0.2 & 0 & 0 & 0 \\
0 & 0 & -1 & 0.5 & 0 \\
0 & 0 & 0.5 & -2.25 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1.4 \\
0 & 0 & -0.5 & 1 & -2 & -1 \\
0 & 0 & 0 & 1 & 1 & -1 & -1
\end{bmatrix}$$

It is interesting to note that the characteristic equation

$$|sI - B| = |sI - F| = 0$$
and the roots which have negative real parts are, respectively:

\[ s^8 + 3s^7 + 28.95s^6 + 79.35s^5 + 206s^4 + 458.875s^3 + 221.05s^2 + 48.48s + 4 = 0 \]  \hspace{1cm} (14b)

and

\[
\begin{aligned}
-0.0239155 \pm j4.27199 \\
-0.0784809 \pm j2.95637 \\
-0.189163 \pm j0.165319 \\
-0.177194 \\
-2.23969 \\
\end{aligned}
\]  \hspace{1cm} (14c)

The linear transformation matrix between the \([x]\) and \([z]\) coordinates is

\[ [z] = [K][x] \]  \hspace{1cm} (15)

where

\[
[K] =
\begin{bmatrix}
-1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\
4.5 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \\
-42.1 & -92.6 & 0 & 0 & 2 & 1 & 0 & 0 \\
-10.55 & -23.3 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 31 & 62 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

4.2. Example 2

Consider the following transfer-function matrix \([T(s)]\) which is expressed by the product of two relatively prime polynomial matrices \([N(s)]\) and \([D(s)]^{-1}\) or

\[ [T(s)] = [N(s)][D(s)]^{-1} \]  \hspace{1cm} (16)

The characteristic matrix polynomial \([D(s)]\) is

\[
[D(s)] = B_3s^4 + B_2s^3 + B_3s^2 + B_4s + B_1 \\
= C_{11}s^4 + C_{21}s^3 + C_{12}s^2 + C_{22}s + C_{13}
\]
A matrix in the block Schwarz form

where

\[ C_{11} = B_5 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad C_{12} = B_2 = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}, \quad C_{13} = B_1 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \]

\[ C_{21} = B_4 = \begin{pmatrix} 2 & 3 \\ 3 & 7 \end{pmatrix}, \quad C_{22} = B_3 = \begin{pmatrix} 3 & 4 \\ 4 & 9 \end{pmatrix} \]

It is interesting to determine the stability of this system. Following eqn. (5b) yields the matrix Routh array as follows:

\[
\begin{align*}
C_{11} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & C_{12} &= \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}, & C_{13} &= \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \\
C_{21} &= \begin{pmatrix} 2 & 3 \\ 3 & 7 \end{pmatrix}, & C_{22} &= \begin{pmatrix} 3 & 4 \\ 4 & 9 \end{pmatrix}
\end{align*}
\]

\[
H_1 = \begin{pmatrix} 1.4 & -0.6 \\ -0.6 & 0.4 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 3 & 4 \\ 4 & 9 \end{pmatrix}, \quad H_3 = \begin{pmatrix} 2.2 & 2.2 \\ 2.2 & 2.2 \end{pmatrix}
\]

\[
H_4 = \begin{pmatrix} 2.6 & 5.4 \\ 5.4 & 4.6 \end{pmatrix}, \quad H_5 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}
\]

The matrix quotients \( H_i, i = 1, \ldots, 4 \), in the matrix Routh array are positive-definite symmetric real matrices. Therefore the system is stable. It is observed that the block element \( C_{31} \) in the first column of the matrix Routh array is not symmetric.
4.3. Example 3

Consider the stability and the structure of the matrix Routh array of the following matrix polynomial $T(s)$ are of interest:

$$
T(s) = B_4 s^3 + B_3 s^2 + B_2 s + B_1 = C_{11} s^3 + C_{12} s^2 + C_{13} s + C_{14} = 0
$$

\begin{align*}
\text{where} \\
C_{11} &= B_4 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \\
C_{12} &= B_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
C_{21} &= B_2 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \\
C_{22} &= B_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
\end{align*}

The determinant of $B_4 (= C_{11}) = -1$ and that of $B_1 (= C_{22}) = 1$. From the derived necessary conditions for asymptotic stability we conclude that the system is unstable because the determinants of $B_4$ and $B_1$ have different sign. Further study of the stability is not necessary. It might be interesting to see the corresponding characteristic equation of this system which can be written as follows:

$$
\det T(s) = -s^6 - 2s^5 + 3s^2 + 2s + 1 = 0
$$

(18 b)

Because the first and the last coefficients, which are the determinants of $B_4$ and $B_1$ respectively, have different sign, therefore the system is unstable. In order to study the structure of the matrix Routh array of this unstable system and the numerically ill-conditioned problem (i.e. $C_{ik}$ is singular) we apply the matrix Routh algorithm in eqn. (5) and use the $C_{ik}$ in eqn. (18 a). The matrix Routh array cannot be obtained because $C_{21}$ is singular. This is a numerically ill-conditioned case. Since the stability is invariant between the original system $T(s)$ and the modified system $T_1(s) = T(s)|_{s^{-1/3}}$, we can construct the matrix Routh array for the $T_1(s)$. The $T_1(s)$ can be written as follows:

$$
T_1(s) = T(s)|_{s^{-1/3}} = C_{22} s^3 + C_{12} s^2 + C_{21} s + C_{11} = C_{11} s^3 + C_{21} s^2 + C_{12} s + C_{22} = 0
$$

(18 c)

\begin{align*}
\text{where} \\
C_{11} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \\
C_{12} &= \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \\
C_{21} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \\
C_{22} &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
\end{align*}
A matrix in the block Schur form

The corresponding matrix Routh array is

\[
\begin{align*}
H_1 &= C_{31}C_{21}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
C_{11} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad C_{12} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \\
H_2 &= C_{21}C_{31}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
C_{21} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad C_{22} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
H_3 &= C_{21}C_{41}^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
C_{31} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad C_{32} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
\end{align*}
\]

(18d)

Although the \(C_{12}\) is singular, we can determine all the \(H_i\)'s. It is observed that the \(H_1\) and \(H_3\) are symmetric and positive definite matrices, while the \(H_2\) is a symmetric and non-positive definite matrix.

An alternative method can be described as follows. Let us construct a new matrix polynomial \(T'_3(s)\) by multiplying a matrix polynomial \(E(s) = (s + 1)I\) to the \(T(s)\) and then defining the matrix coefficients as \(C_{1, i}'\) and \(C_{2, i}'\):

\[
T'_3(s) = (s + 1)T(s) = C_{11}' s^4 + C_{12}' s^2 + C_{13}' s + 1 = 0 \quad (18e)
\]

where

\[
\begin{align*}
C_{11}' &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, & C_{12}' &= \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, & C_{13}' &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
C_{21}' &= \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, & C_{22}' &= \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}
\end{align*}
\]

If we wish to maintain the consistency of \(C_{11} = I\), we may interchange the rows in the \(T'_3(s)\) and define new matrix coefficients as \(C_{1, i}'\) and \(C_{2, i}'\):

\[
T'_3(s) = C_{11}' s^4 + C_{12}' s^3 + C_{13}' s^2 + C_{14}' s + 1 = 0 \quad (18f)
\]
where

\[
C_{11} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad C_{12} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, \quad C_{13} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
\]

\[
C_{21} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad C_{22} = \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}
\]

The corresponding matrix Routh array is

\[
C_{11} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad C_{12} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, \quad C_{13} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
\]

\[
H_1 = \frac{1}{2} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix},
\]

\[
H_2 = \frac{1}{8} \begin{pmatrix} 8 & 1 \\ 1 & 8 \end{pmatrix},
\]

\[
H_3 = \frac{1}{16} \begin{pmatrix} 17 & 32 \\ 32 & 17 \end{pmatrix},
\]

\[
H_4 = \frac{1}{6} \begin{pmatrix} 6 & -1 \\ -1 & 6 \end{pmatrix},
\]

\[
C_{41} = \frac{1}{6} \begin{pmatrix} -1 & 6 \\ 6 & -1 \end{pmatrix},
\]

\[
C_{42} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
\]  

No singular matrix appears in the matrix Routh array and all the $H_i$'s can be obtained. It is observed that only the $H_3$ is a symmetric but non-positive definite matrix.

From the above illustrations we conclude that if any ill-conditioned problem occurs in the calculation, then the above methods can be applied to solve the problem.

5. Conclusion

The transformation matrix established by Chen and Chu (1966) for transforming the companion form to the Schwarz form has been modified and
A matrix in the block Schwarz form

extended to transform the companion block form to the block Schwarz form. The new matrix in the block Schwarz form has been constructed by using the matrix quotients obtained from the matrix Routh array which is constructed from the characteristic matrix polynomial. When the matrix quotients in the matrix Routh array are positive-definite symmetric real matrices, the sufficient condition derived in this paper shows that the multivariable system is asymptotically stable. Also, a set of necessary conditions has been derived for the asymptotic stability. Thus, we have partially extended the Routh criterion (Routh 1877) to the matrix Routh criterion for a class of matrix polynomials. The direct extension of the necessary and sufficient condition of the Routh criterion (Routh 1877) to a general matrix polynomials need further studies.

Acknowledgments

The authors wish to express their gratitude for the valuable comments and suggestions of the reviewer.

This work was supported in part by U.S. Army Missile Command, Redstone Arsenal, Alabama, DAAK 40-78-C-0017.

References

TRANSFER FUNCTION FITTING FROM EXPERIMENTAL FREQUENCY-RESPONSE DATA

L. S. SHIEH and M. H. COHEN
Department of Electrical Engineering, University of Houston, Houston, TX 77004, U.S.A.

(Received 20 January 1978; received for publication 30 March 1978)

Abstract—A simple method is proposed that will fit the coefficients of a transfer function from the real and imaginary parts of experimental frequency-response data. An approximate logarithmic amplitude-frequency plot is used to formulate an irrational transfer function which then estimates the interpolation data and the degree of the final transfer function. The present method is applicable to either minimum or non-minimum phase system identification.

1. INTRODUCTION

The problem of finding unknown coefficients of a transfer function as a ratio of two frequency-dependent polynomials has been investigated by Levy[1], Kardashov and Karniushin[2], and Sanathanan and Koerner[3]. In general, they would evaluate the polynomial coefficients by minimizing the weighted sum of squares of the errors in magnitude at arbitrary experimental points. Ausman[4] proposed a graphical method to rapidly estimate the coefficients of a transfer function; however, that procedure is only applicable for a minimum phase system.

In this paper a simple method is presented to approximate the coefficients of a transfer function for minimum and non-minimum phase systems. The generalized Bode plot is used to formulate an irrational transfer function from which we obtain interpolation frequency-response data that will allow us to estimate the polynomial coefficients without minimizing the weighted sum of squares of the errors in magnitude at arbitrary points.

2. THE DERIVATION

Consider the transfer function

\[ G(s) = \frac{P_0 + P_1 s + P_2 s^2 + \cdots + P_m s^m}{1 + q_1 s + q_2 s^2 + \cdots + q_n s^n} \]  

(1)

where \( p_i \) and \( q_i \) are unknown coefficients to be determined. Substituting \( s = j \omega_k \) we have

\[ G(j \omega_k) = \frac{P_0 + p_1 j \omega_k + p_2 j \omega_k^2 + \cdots + p_m j \omega_k^m}{1 + q_1 j \omega_k + q_2 j \omega_k^2 + \cdots + q_n j \omega_k^n} \]

(2)

where \( R_k \) and \( I_k \) are the real and imaginary parts of the transfer function at the experimental frequencies \( \omega_k \). After we multiply both sides of eqn (2) by the common denominator, we separate the real and imaginary parts and then equate the respective real and imaginary parts. We now have

\[ p_0 - p_2 \omega_k^2 + p_4 \omega_k^4 - \cdots + q_1 I_\omega_k + q_3 R_\omega_k - q_5 I_\omega_k^3 - q_7 R_\omega_k^2 + \cdots = R_k \]  

(3)

and

\[ p_1 \omega_k - p_3 \omega_k^3 + p_5 \omega_k^5 - \cdots - q_1 R_\omega_k + q_3 I_\omega_k + q_5 R_\omega_k^3 - q_7 I_\omega_k^4 + \cdots = I_k \]  

(4)
REDESIGN OF THE STABILIZED PITCH CONTROL SYSTEM OF A SEMI-ACTIV-ETC(U)

UNCLASSIFIED

APR 79
L SHIEH
Combining eqns (3) and (4) results in

\[ p_0 + p_1 \omega_k - p_2 \omega_k^2 - p_3 \omega_k^3 + p_4 \omega_k^4 + p_5 \omega_k^5 - \cdots - q_1 (R_k - I_k) \omega_k + q_2 (R_k + I_k) \omega_k^2 + q_3 (R_k - I_k) \omega_k^3 - q_d (R_k + I_k) \omega_k^4 - \cdots = R_k + I_k \]  

The complete form of eqn (5) is

\[
\begin{bmatrix}
1 \omega_1 - \omega_1^2 & -\omega_1 \omega_1^3 & \omega_1^4 & \omega_1^5 & \cdots & T_1 \omega_1 - s_1 \omega_1^4 & -s_1 \omega_1^5 & \omega_1^6 & \cdots \\
1 \omega_2 - \omega_2^2 & -\omega_2 \omega_2^3 & \omega_2^4 & \omega_2^5 & \cdots & T_2 \omega_2 - s_2 \omega_2^4 & -s_2 \omega_2^5 & \omega_2^6 & \cdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
1 \omega_m - \omega_m^2 & -\omega_m \omega_m^3 & \omega_m^4 & \omega_m^5 & \cdots & T_m \omega_m - s_m \omega_m^4 & -s_m \omega_m^5 & \omega_m^6 & \cdots \\
1 \omega_1 - \omega_1^2 & -\omega_1 \omega_1^3 & \omega_1^4 & \omega_1^5 & \cdots & T_1 \omega_1 - s_1 \omega_1^4 & -s_1 \omega_1^5 & \omega_1^6 & \cdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
\end{bmatrix}
\begin{bmatrix}
p_0 \\
p_1 \\
p_2 \\
p_3 \\
p_4 \\
\vdots \\
p_m \\
q_1 \\
q_2 \\
q_3 \\
\vdots \\
q_n \\
\end{bmatrix}
= \begin{bmatrix}
R_1 + I_1 \\
R_2 + I_2 \\
R_3 + I_3 \\
R_4 + I_4 \\
R_5 + I_5 \\
\vdots \\
R_m + I_m \\
q_1 \\
q_2 \\
q_3 \\
\vdots \\
q_n \\
\end{bmatrix}
\]

where

\[
s_k = R_k + I_k; \quad k = 1, 2, \ldots \\
T_k = R_k - I_k; \quad k = 1, 2, \ldots \\
x = m + n + 1
\]

By substituting the selected \( x \) sets of frequency response data into eqn (6), we can solve for the required unknown coefficients \( p_i \) and \( q_i \).

### 3. Estimation of the Corner Frequency and the Order


The approximate transfer function is

\[
F(s) = \frac{k \left(1 + \frac{s}{a_1}\right)^{m_1} \left(1 + \frac{s}{a_2}\right)^{m_2} \cdots \left(1 + \frac{s}{a_n}\right)^{m_n}}{\left(1 + \frac{s}{b_1}\right)^{n_1} \left(1 + \frac{s}{b_2}\right)^{n_2} \cdots \left(1 + \frac{s}{b_1}\right)^{n_n}}
\]

where \( a_i \) and \( b_i \) are corner frequencies, and where \( m_i \) and \( n_i \) may be integer or fractional values. In general, eqn (7) is an irrational transfer function. Compared to an approximation made by other methods[4, 5], this present analysis is much better because the slopes may be precise fractional values. However, the worst errors caused by piecewise segment approximation occur at the corner frequencies \( a_i \) and \( b_i \); therefore, these corner frequencies provide the most important information of the frequency-response curve. If the interpolation data in eqn (6) include these important corner frequencies, a good transfer-function fitting is expected. In this paper the corner frequency-response data are chosen as main interpolation points for determining the unknown coefficients in eqn (6). The difference of the order of two polynomials in eqn (1) can be estimated from eqn (7). In other words

\[
n - m = \sum_{k=1}^{n} n_k - \sum_{k=1}^{m} m_k.
\]

Based on eqn (8), the numbers of the unknown coefficients and the interpolation points may be estimated.
4. ILLUSTRATIVE EXAMPLES

Example 1. Consider Levy's non-minimum phase example. The frequency-response data generated from the transfer function in eqn (9) is shown in Table 1 and the log-amplitude plot versus log-frequency is shown in Fig. 1.

\[ T(s) = \frac{1 - s}{1 + 0.1s + 0.01s^2}. \]  

(9)

The irrational transfer function approximated from the generalized Bode plot is

\[ T(s) = \frac{(1 + \frac{s}{0.5})^{0.54} (1 + \frac{s}{2})^{0.57} (1 + \frac{s}{40})^{0.15}}{(1 + \frac{s}{10})^{2.25}} \]  

(10)

where the corner frequencies are

\[ \omega_1 = \frac{\omega_j}{2} = 0.5, \quad \omega_3 = \omega_j = 10 \]
\[ \omega_2 = \frac{\omega_j}{2} = 2, \quad \omega_4 = \omega_j = 40 \]

Table 1.

| k | \( \omega \) | \( |T(j\omega)| \) | \( |G(j\omega)| \) | \( R_k \) | \( I_k \) | \( |G(j\omega)| \) | \( |G(j\omega)| \) | \( R_k \) | \( I_k \) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.1 | 1.0064 | -6.45 | 1.0000 | -0.1130 | 1.0013 | -6.26 | 0.9953 | -0.1092 |
| 2 | 0.2 | 1.0239 | -12.41 | 1.0000 | -0.2200 | 1.0160 | -12.41 | 0.9928 | -0.2183 |
| 3 | 0.5 | 1.1194 | -29.43 | 0.9753 | -0.5500 | 1.1142 | -29.34 | 0.9713 | -0.5459 |
| 4 | 0.7 | 1.2393 | -39.01 | 0.9630 | -0.7800 | 1.2171 | -38.91 | 0.9472 | -0.7644 |
| 5 | 1.0 | 1.4399 | -51.06 | 0.9050 | -1.1200 | 1.4125 | -50.66 | 0.8955 | -1.0924 |
| 6 | 2.0 | 2.2772 | -75.04 | 0.5880 | -2.2000 | 2.2631 | -75.15 | 0.5798 | -2.1875 |
| 7 | 4.0 | 4.375 | -102.0 | 0.9250 | -4.3400 | 4.3954 | -101.50 | 0.8770 | -4.3071 |
| 8 | 7.0 | 8.7501 | -135.9 | 0.9750 | -5.6000 | 8.0664 | -136.08 | 0.9472 | -5.6088 |
| 9 | 10.0 | 10.65 | -174.0 | 0.9050 | -1.0000 | 9.9115 | -174.67 | 0.8955 | -1.0924 |
| 10 | 20.0 | 5.5541 | -233.4 | 0.5880 | -3.3100 | 5.4612 | -233.4 | 0.5798 | -3.245 |
| 11 | 40.0 | 2.5451 | -253.5 | 0.9250 | -6.7200 | 2.5363 | -253.5 | 0.914 | -6.6088 |
| 12 | 70.0 | 1.4479 | -261.0 | 0.9750 | -2.2000 | 1.4205 | -261.0 | 0.954 | -2.1875 |
| 13 | 100 | 0.9994 | -263.5 | 0.9050 | -1.1300 | 0.9892 | -263.6 | 0.9832 |

Fig. 1. Bode plot shows magnitude/frequency response and piecewise segment approximations of \( F(s) = (1 - s)(1 + 0.1s + 0.01s^2) \).
The order of eqn (1) may be estimated from eqn (10), or
\[ m = 0.54 + 0.57 + 0.15 = 1 \]
\[ n = 2.25 - 2 \]
\[ n - m = 1. \]

Four frequency-response data \((\omega_1, \omega_2, \omega_3, \omega_4)\) are required in eqn (6) to fit the four unknown coefficients \(p_0, p_1, q, \) and \(q^2\). The identified transfer function is
\[
G(s) = \frac{0.99628 - 0.991402s + 0.10053s + 0.010072s^2}{s + 1 + 0.10053s + 0.010072s^2}. \tag{11}
\]

The corresponding frequency-response data of eqn (11) and that of eqn (9) are compared in Table I. The results are very satisfactory.

**Example 2.** A set of frequency-response data generated by the following transfer function is shown in Table 2 and the log-amplitude versus log-frequency plot is shown in Fig. 2.

### Table 2.

| \(k\) | \(\omega_l\) | \(|T(\omega_l)|\) | \(\angle T(\omega_l)\) | \(R_k\) | \(I_k\) | \(|G(\omega_l)|\) | \(\angle G(\omega_l)\) | \(R_k\) | \(I_k\) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.1 | 1.0002 | -0.28 | 1.0002 | -0.0048 | 1.0002 | -0.28 | 1.0002 | -0.0048 |
| 2 | 0.4 | 1.0029 | -1.10 | 1.0027 | -0.0193 | 1.0029 | -1.10 | 1.0028 | -0.0193 |
| 3 | 0.8 | 1.0117 | -2.13 | 1.0110 | -0.0375 | 1.0124 | -2.18 | 1.0116 | -0.0383 |
| 4 | 2.0 | 1.1113 | -5.68 | 1.1058 | -0.1101 | 1.1113 | -5.69 | 1.1058 | -0.1101 |
| 5 | 2.2 | 1.1470 | -6.61 | 1.1394 | -0.1321 | 1.1470 | -6.62 | 1.1394 | -0.1322 |
| 6 | 3.6 | 1.4936 | -33.8 | 1.2418 | -0.0048 | 1.4935 | -33.8 | 1.2416 | -0.0048 |
| 7 | 5.4 | 0.8425 | -57.8 | 0.4483 | -0.7132 | 0.8424 | -57.8 | 0.4484 | -0.7132 |
| 8 | 8.0 | 0.6123 | -59.1 | 0.3147 | -0.5353 | 0.6123 | -59.1 | 0.3147 | -0.5252 |
| 9 | 16 | 0.3730 | -69.5 | 0.1309 | -0.3493 | 0.3730 | -69.5 | 0.1309 | -0.3493 |
| 10 | 20 | 0.3091 | -72.9 | 0.0906 | -0.2955 | 0.3091 | -72.9 | 0.0908 | -0.2955 |
| 11 | 100 | 0.0662 | -86.3 | 0.0042 | -0.0661 | 0.0662 | -86.3 | 0.0042 | -0.0661 |
| 12 | 110 | 0.0602 | -86.7 | 0.0035 | -0.0601 | 0.0602 | -86.7 | 0.0035 | -0.0601 |

Fig. 2. Bode plot shows magnitude/frequency response and piecewise segment approximations of \(F(s) = (6.6378s^2 + 22.9999s + 111.27974)/(s^2 + 9.8027s + 28.3706s + 111.27974). \)
Transfer function fitting from experimental frequency-response data

\[ T(s) = \frac{6.6378918s^2 + 22.999878s + 111.27974}{s^3 + 9.882741s^2 + 28.37056s + 111.27974} \]  

(12)

The irrational transfer function approximated from the generalized Bode plot is

\[ T(s) = \frac{\left(1 + \frac{s}{0.8}\right)^{0.09}}{\left(1 + \frac{s}{2.2}\right)^{0.63}} \frac{\left(1 + \frac{s}{5.4}\right)^{1.02}}{\left(1 + \frac{s}{3.6}\right)^{2.92}} \frac{\left(1 + \frac{s}{16}\right)^{0.38}}{\left(1 + \frac{s}{100}\right)^{0.98}} \]  

(13)

The corner frequencies are

\[ \omega_1 = \omega_3 = 0.8, \quad \omega_3 = \omega_6 = 3.6, \quad \omega_5 = \omega_6 = 16 \]

\[ \omega_2 = \omega_4 = 2.2, \quad \omega_4 = \omega_7 = 5.4, \quad \omega_6 = \omega_7 = 100. \]  

(14)

The order of Eqn (1) is estimated as follows:

\[ m = 0.09 + 0.63 + 1.02 = 2 \]

\[ n = 2.42 + 0.28 + 0.04 = 3 \]

\[ n - m = 1. \]

At least six unknown coefficients are required to be identified. By substituting the corner frequencies into eqn (6), we have the identified transfer function

\[ G(s) = \frac{1.000029 + 0.206648s + 0.05966s^2}{1 + 0.254924s + 0.088779s^2 + 0.008965s^3}. \]  

(15)

The comparison of the frequency-response data of eqns (12) and (15) is shown in Table 2. These results are also satisfactory.

5. CONCLUSION

A simple method has been presented for fitting a transfer function from experimental frequency-response data. A logarithmic amplitude-frequency curve is first plotted from the available frequency-response data, then it is smoothed and approximated by piecewise segments with integer or fractional slopes. As a result, the most important interpolation data and the order of the transfer function may be obtained from the irrational transfer function. When the slope at two consecutive low frequencies, \( \omega_1 \) and \( \omega_2 \), is

\[ x(\text{slope}) = \frac{T(\omega_1)s - T(\omega_2)s}{20 \log \frac{\omega_1}{\omega_2}} \]

(In other words there exists \( x \) poles at the origin.), then the available frequency-response data should be multiplied by \( (j\omega)^x \) so that eqn (6) may be applied. The method presented in this paper is useful for digital computation and provides an additional tool for system identification.

A computer program, based on the approach discussed, has been written in the appendix.

Acknowledgement—This work was supported in part by U.S. Army Missile Command, Redstone Arsenal, Alabama, DAAK4-78-C-0017.

REFERENCES

calculated and printed on the last data card. The values of the selected subscript numbers (i.e., the available data are given in Table 6. D. E.

The corner frequencies (the most important data) occur at

The data on the next card is the number of the unknown constants in the numerator and denominator polynomials in eqn (16): m = 2 and n = 2.

The corner frequencies (the most important data) occur at \( XW_0 = 0.5, XW_1 = 2, XW_2 = 10, \) and \( XW_3 = 40; \) therefore, the values of the selected subscript numbers (i.e., \( ND \)) are \( ND_1 = 3, ND_2 = 6, ND_3 = 9, \) and \( ND_4 = 11. \) These data appear on the last data card.

The output of this program is \( p_0 = 0.99628, p_1 = -0.99140, q_1 = 0.10053 \) and \( q_2 = 0.01007. \) Also, the real parts, imaginary parts, magnitudes, and phase angles at available frequencies of the identified transfer function in eqn (16) are calculated and printed for comparison with the given data.

A listing of the computer program is as follows:

```
C A PROGRAM TO FIT TRANSFER FUNCTION USING FREQUENCY-RESPONSE DATA.
C
DOUBLE PRECISION * (50), XM (50), XI (50), XR (50), XW (50), B (30)
10 A (30, 30), C (30, 30), D (10, 10), XS, XW (50)
01 FORMAT (5X, 16H, X) , CXY
100 FORMAT (5X, 13H, NDT, NP)
001 FORMAT (16H, 15B )
002 FORMAT (16H, 15B )
003 FORMAT (5X, 16H, X)
004 FORMAT (5X, 16H, X)
005 FORMAT (5X, 16H, X)
```

APPENDIX

This program is used to fit a transfer function using frequency-response data. The details to prepare the input cards can be summarized as follows:

The first data card:
- **NDT**—number of available frequency-response data
- **NP**—number of different transfer function structures to be identified.

The second data card:
- A vector of the frequency values at which there is available data

The third data card:
- A vector of the values of the real parts of the available data at \( XW_i \)

The fourth data card:
- A vector of the values of the imaginary parts of the available data at \( XW_i \)

The fifth data card:
- \( m \)—the number of the unknown constants in the numerator polynomial of the transfer function to be identified.
- \( n \)—the number of the unknown constants in the denominator polynomial of the transfer function to be identified.

The sixth data card:
- A vector of the values of the imaginary parts of the available data at \( XW_i \)

The output of this program is

\[
T(s) = \frac{p_0 + q_0}{1 + q_1 s + q_2 s^2}
\]   (16)

The data on the first data card are \( NDT = 13 \) and \( NP = 1. \) The values of the frequencies, real parts, and imaginary parts of the available data are given in Table 1. Therefore, the data on the subsequent input cards are

\[
\begin{align*}
XW_0 &= 0.1, & XW_1 &= 0.2, \ldots, & XW_{10} &= 100 \\
XR_1 &= 1.0000, & XR_2 &= 1.0000, \ldots, & XR_{10} &= -0.1130 \\
XI_1 &= -0.1130, & XI_2 &= -0.2200, \ldots, & XI_{10} &= 0.9930
\end{align*}
\]

The data on the next card is the number of the unknowns in the numerator and denominator polynomials in eqn (16):

- \( m = 2 \) and \( n = 2. \)

Transfer function fitting from experimental frequency-response data

```
NAME
WEAO (5,501) (ND(J),J=1,MM)
WRITE (6,601) (ND(J),J=1,MM)
DO 10 J=1,MM
J=40(J)
1) READ (501) (HU(J),J=1,MM)
WRITE (6,602) HU(J)
K=1
XRI(J)=XR(J)
XI(J)=XI(J)
DO 20 K=1,MM
20 IF (K,EQ,1) GO TO 21
K=K+1
21 CONTINUE
DO 30 J=2,MM
A(K,J)=XS*(K)**(J-1)
LK=K+1
KEQ=2)
WRITE (501) (J,XP(J),XN(J),XI(J))
WRITE (b,606) J,XP(J),C(J)
 FORMAT (/2X,5F20.8,2X,F20.8,2X,F20.8,2X,F20.8,2X,F20.8)
CONTINUE
10 CONTINUE
CALL INVER (A,MM,G,0,DET,B,M)
WRITE (b,605) J,X(J),XI(J),XR(J),XI(J)
WRITE (b,606) J,XP(J),C(J)
 FORMAT (12)
END
SUBROUTINE INVER (A,M,M,DET,XC,XB)
DOUBLE PRECISION A(30,30),B(30,30),IPIVOT(30),INDEX(30,2)
IPIVOT(30),XC(30),XD(30),DET,T,S
EQUIVALENCE (INOM,JROW),(ICOL,JCOL)
FORMAT (12)
```

601 FORMAT ('/'(2X,#15.6))
   57 DET=M,
      DO 17 J=1,N
  17 IPVOT(J)=0
      DO 14 I=1,N
      T=0,
      DO 9 J=1,N
  9 IF(IPVOT(J)=1) 13,9,13
 11 DO 23 K=1,N
 13 IF(IPVOT(K)=1) 41,23,11
 43 IF (DABS(T)-DABS(A(J,K))) .GT. 23,23
 83 IR=J
   ICOL=K
   T=TA(J,K)
 23 CONTINUE
   4 CONTINUE
   IPVOT(ICOL)=IPVOT(ICOL)+1
   IF(IPVOT(ICOL)=1) 73,109,73
  73 DET=DET
      DO 12 L=1,N
      T=TA(L,J)
         A(INOW,L)=A(INOW,L)
 12 A(INOW,L)=T
      IF(M) 109,109,33
 13 DO 2 L=1,N
      B(INOW,L)=B(INOW,L)
  2 R(INOW,L)=T
 109 INDEXX(I,1)=IROW
      INDEXX(I,2)=ICOL
      PIVOT(I)=A(INOW,ICOL)
      DFT=0.0000000003
         A(INOW,ICOL)=1.
      ON 205 L=1,N
 205 A(INOW,ICOL)=A(INOW,ICOL)/PIVOT(I)
      IF(4) 347,347,66
 66 DO 52 L=1,N
 52 B(INOW,L)=B(INOW,L)/PIVOT(I)
 347 DO 134 L=1,N
      IF (LI=ICOL) 21,114,21
 71 T=TA(L,ICOL)
      A(L,ICOL)=0.
      DO 89 L=1,N
 89 A(LI,L)=A(LI,L)-A(ICAL,L)*T
      IF(4) 134,134,18
 10 DO 12 L=1,N
 52 B(LI,L)=B(LI,L)-R(INOW,L)*T
 134 CONTINUE
 135 CONTINUE
 222 DO 3 1 I=1,N
      LN=I-1.
      IF(*INDEXX(L,1)-INDEXX(L,2)) 19,3,19
 19 JNOW=INDEXX(L,1)
      JCOL=INDEXX(L,2)
      ON 549 K=1,N
      T=TA(K,JNOW)
         A(K,JNOW)=A(K,JNOW)
         A(K,JNOW)=T
 549 CONTINUE
 3 CONTINUE
      DO 40 K=1,N
 40 CONTINUE
      ON 20 K=1,N
 380. DO 10 J=1,N
  30 S=S*A(K,J)*XC(J)
  20 XD(K)=S
      #RITE (6,691) (XD(K),#1,N)
  81 CONTINUE
      RETURN
      END
Solution of state-space equations via block-pulse functions

L. S. SHIEH†, C. K. YEUNG† and B. C. McINNIS†

A recursive algorithm is developed for the piecewise-constant solution of dynamic equations via block-pulse functions \( \phi_j(t) \), where \( j = 1, 2, \ldots, m \). For \( 1 \leq j \leq m \) (where \( j \) and \( m \) are integers) and final time \( T \), each block-pulse function \( \phi_j(t) \) is defined by \( \phi_j(t) = 1 \) for \( (j-1)T/m \leq t < jT/m \) and \( \phi_j(t) = 0 \) otherwise. Compared with Walsh function approaches, the proposed method is simpler to compute, is more suitable for computer programming, and provides the same accuracy. Also, a discrete-time solution is derived for a zero-input state equation.

1. Introduction

Consider a linear time-invariant system described by the state equation

\[
\dot{x}(t) = Ax(t) + Bu(t) \tag{1a}
\]

and an initial vector

\[
x(0) = x_0 \tag{1b}
\]

where \( A \) is an \( n \times n \) system matrix, \( B \) is an \( n \times r \) constant matrix, \( x(t) \) is a state vector of \( n \) components, \( \dot{x}(t) \) is a rate vector, and \( u(t) \) is an \( r \)-component input vector. It is often difficult to evaluate the integration \( \int_0^t \dot{x}(t) \, dt \), which is the solution \( x(t) \) in (1), by a numerical method (Carnahan et al. 1969). One approach is to find a set of orthogonal functions \( \psi_i(t) \) for the approximate solution as follows:

\[
x(t) = x(0) + \int_0^t \dot{x}(t) \, dt \approx P \int_0^t \psi_i(t) \, dt \approx PQ \psi(t) = W \psi(t) \tag{2}
\]

where \( P \), \( Q \) and \( W \) are \( n \times m \), \( m \times m \) and \( n \times m \) weighting matrices, respectively, and \( \psi(t) \) is an \( m \times 1 \) vector with \( m \) orthogonal functions \( \psi_i(t) \), which are both suitable for approximation of \( \dot{x}(t) \) and easy to integrate numerically. Corrington (1973), Chen and Hsiao (1975), and Rao and Sivakumar (1975) chose Walsh functions as the \( \psi_i(t) \) for the approximate solution in (2) and reported that their piecewise-constant solution gives a satisfactory result. However, their computational methods (Chen and Hsiao 1975, Rao and Sivakumar 1975) either required the inversion of a large matrix or the inversion of many small matrices. This results in computing time and storage being wasted, and the truncation and round-off errors might be seriously accumulated. Recently, Chen et al. (1976) and Gopalsami and Deekshatulu (1976) introduced a set of 'block-pulse functions' for the solutions of distributed systems and identification problems. They pointed out that there is a one-to-one relationship between Walsh functions and block-pulse functions. For \( 1 \leq j \leq m \), where \( j \) and \( m \) are integers, the block-pulse function \( \phi_j(t) \) is

Received 6 April 1977.
† Department of Electrical Engineering, University of Houston, Houston, Texas 77004, U.S.A.
re-defined and extended in the interval $0 \leq t < T$ (rather than in the interval $0 \leq t < 1$ as in Chen et al. (1976), and Gopalsami and Deekshatulu (1976)) and by

$$
\phi_j(t) = \begin{cases} 
1 & \text{for } (j-1)T/m \leq t < jT/m \\
0 & \text{otherwise}
\end{cases}
(3a)
$$

$T$ is the final time, and $m$ is the number of subintervals between $t=0$ and $t=T$ as well as the number of block-pulse functions to be used. When $m$ block-pulse functions are used to approximate the integration of the original block-pulse functions, we have

$$
\int_0^t \phi(t) \, dt \approx \frac{T}{m} H \phi(t) = \frac{T}{m} \begin{bmatrix}
\frac{1}{m} & 1 & 1 \\
0 & \frac{1}{m} & 1 \\
\vdots & \vdots & \vdots \\
0 & 0 & \frac{1}{m}
\end{bmatrix}
\begin{bmatrix}
\phi_1(t) \\
\phi_2(t) \\
\vdots \\
\phi_m(t)
\end{bmatrix}
(3b)
$$

where $\phi(t)$ is an $m \times 1$ vector with $m$ block-pulse functions. The constant matrix $(T/m)H$, with the dimensions $m \times m$, is the operational matrix (Chen et al. 1976, Gopalsami and Deekshatulu 1976) for the block-pulse functions. Sannuti (1976) discussed the properties of the $\phi_j(t)$ and proposed a method for the solutions of linear and non-linear problems. From (3b) we observe that the matrix $H$ is an upper triangular matrix that consists of diagonal elements being $\frac{1}{m}$ and the other elements being 1. By taking advantage of this peculiar arrangement of $H$ and by choosing the block-pulse functions $\phi_j(t)$ as the $\psi(t)$ in (2), an alternative method is proposed in this paper to derive an effective algorithm for the piecewise-constant solution of the state equation in (1). The computation in our algorithm involves the inversion of only one matrix that has the same size as the system matrix. Compared with Walsh function approaches (Corrington 1973, Chen and Hsiao 1975, Rao and Sivakumar 1975) the proposed method is simpler to compute, is more suitable for computer programming, and provides the same accuracy.

2. Main result

Let $x_i(t)$ be the $i$th component of the state vector $x(t)$ that is the solution of the state equation in (1). The $x_i(t)$ can be expressed approximately as

$$
\sum_{j=1}^{m} C_{i,j} \phi_j(t),
$$

where $m$ is a large finite number, $\phi_j(t)$ are block-pulse functions, and $C_{i,j}$ are weighting constants to be determined. The state vector $x(t)$ can also be approximated as

$$
x(t) \approx C \phi(t)
(4a)
$$

where

$$
C = [C_1, C_2, ..., C_m]
(4b)
$$

and

$$
\phi(t) = [\phi_1(t), \phi_2(t), ..., \phi_m(t)]'
(4c)
$$

The prime designates the transpose, and the $n \times m$ matrix $C$ consists of $m$ column vectors $C_j$ to be determined. Our goal is to develop an effective
algorithm to determine \( C_j \) for every \( j \) so that the piecewise-constant solution in (4 a) can be obtained.

We will now derive the recursive algorithm. Let the rate vector \( \dot{x}(t) \) in (1) be approximated as

\[
\dot{x}(t) \approx D \phi(t)
\]

by using \( m \) block-pulse functions, where

\[
D = [d_1, d_2, \ldots, d_m]
\]  

(5 b)

The \( D \) is an \( n \times m \) constant matrix with \( m \) column vectors \( d_j \) of size \( n \times 1 \) to be determined. Integrating (5 a) and using the results of (3 b) and (4 a) yields

\[
x(t) \approx D \int_{0}^{t} \phi(t) \, dt + x(0) \approx \left[ \frac{T}{m} Dh + G \right] \phi(t) = C \phi(t)
\]  

(6 a)

where

\[
G = [x(0), x(0), \ldots, x(0)] = [g_1, g_2, \ldots, g_m]
\]  

(6 b)

and

\[
C = \frac{T}{m} Dh + G = [C_1, C_2, \ldots, C_m]
\]  

(6 c)

The \( g_i \) in (6 b) is the initial vector \( x(0) \), and the constant matrix \( (T/m)H \) is shown in (3 b). The accuracy of an approximate solution in (6 a) depends on the number of block-pulse functions and the time interval \( T/m \) used. The \( r \times 1 \) input vector \( u(t) \) in (1) can also be approximated as

\[
u(t) \approx L \phi(t)
\]  

(7 a)

using \( m \) block-pulse functions, where

\[
L = [L_1, L_2, \ldots, L_m]
\]  

(7 b)

The \( r \times m \) matrix \( L \) consists of \( m \) column vectors \( L_j \) to be determined. By applying the orthogonal property of the block-pulse functions to (7 a), we have

\[
L_j \approx \frac{m}{T} \int_{(j-1)T/m}^{jT/m} u(t) \, dt \approx \frac{1}{T} \left[ u((j-1)T/m) + u(jT/m) \right]
\]  

(7 c)

equals average value of \( u(t) \) over the interval \((j-1)T/m \leq t \leq jT/m\). The accuracy of the approximation in (7 c) depends on the time interval \( T/m \) used. Substituting (5 a), (6 a) and (7 a) into (1 a) yields

\[
D = \frac{T}{m} ADH + AG + BL = \frac{T}{m} ADH + K
\]  

(8 a)

where

\[
K = AG + BL = [k_1, k_2, \ldots, k_m]
\]  

(8 b)

The column vector \( k_j \) is an \( n \times 1 \) known vector. The unknown matrix \( D \) in (8 a) and (5 a) can be determined from the matrix equation (Chen and Hsiao 1975)

\[
\begin{bmatrix}
I_{nm} - A \otimes \frac{T}{m} H' \\
\end{bmatrix}
\begin{bmatrix}
d_1 \\
\vdots \\
d_m 
\end{bmatrix}
= \frac{T}{m}
\begin{bmatrix}
I_{nm} - A \otimes H' \\
\end{bmatrix}
\begin{bmatrix}
d_1 \\
\vdots \\
d_m 
\end{bmatrix}
= \begin{bmatrix}
k_1 \\
\vdots \\
k_m 
\end{bmatrix}
\]

(8 c)
or
\[
\begin{bmatrix}
A_1 & 0 & 0 & \ldots & 0 \\
-A & A_1 & 0 & \ldots & 0 \\
-A & -A & A_1 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-A & -A & -A & \ldots & A_1
\end{bmatrix}
\begin{bmatrix}
d_1 \\
d_2 \\
d_3 \\
\vdots \\
d_m
\end{bmatrix}
= \frac{m}{T}
\begin{bmatrix}
k_1 \\
k_2 \\
k_3 \\
\vdots \\
k_m
\end{bmatrix}
\tag{8 \, d}
\]

where
\[
A_1 = \frac{m}{T} I_n - \frac{1}{2} A
\tag{8 \, e}
\]

The \( I_{nm} \) in (8 \, e) is an \( nm \times nm \) identity matrix, and the \( \otimes \) in (8 \, e) represents the Kronecker product. Each \( n \times n \) block element 0 in (8 \, d) is a null matrix and \( I_n \) in (8 \, e) is an \( n \times n \) identity matrix. It is known that, as more orthogonal functions are used to approximate \( x(t) \), a better approximate solution is obtained. Therefore, \( m \) should be a large number and the matrix

\[
[(m|T)I_{nm} - A \otimes H']
\]

is large. The direct inversion of such a matrix for the solution of \( d_j \) in (8 \, e) is not an effective method as far as the computing time and storage are concerned. However, from the peculiar formulation of the square matrix in (8 \, d), we can derive an effective algorithm for solving \( d_j \) instead of inverting the matrix directly. This effective algorithm is derived in the following way. By pre-multiplying each block element on both sides of (8 \, d) by \( A_1^{-1} \) and by rearranging the new matrix equation, we have an alternative form of (8 \, d) as

\[
\begin{bmatrix}
d_2 \\
R_2 & 0 & 0 & \ldots & 0 \\
R_2 & R_2 & 0 & \ldots & 0 \\
R_2 & R_2 & R_2 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
R_2 & R_2 & R_2 & \ldots & R_2
\end{bmatrix}
\begin{bmatrix}
d_1 \\
d_2 \\
d_3 \\
\vdots \\
d_m
\end{bmatrix}
= \frac{m}{T}
\begin{bmatrix}
R_1k_1 \\
R_2k_2 \\
R_3k_3 \\
\vdots \\
R_mk_m
\end{bmatrix}
\tag{9 \, a}
\]

where
\[
R_1 = A_1^{-1} = \left(\frac{m}{T} I_n - \frac{1}{2} A\right)^{-1}
\]
\[
R_2 = A_1^{-1} A = R_1 A
\tag{9 \, b}
\]

\[
d_1 = \frac{m}{T} R_1 k_1
\]

Equation (9 \, a) can be solved readily for \( d_j \). After obtaining the matrices \( R_1 \) and \( R_2 \) and the vector \( d_1 \) in (9 \, b), we can immediately determine the vector \( d_2 \) from the first equation in (9 \, a). Then we can substitute \( d_2 \) into the second equation and solve for \( d_3 \), etc. Note that the \( m \) can be chosen so that \([(m|T)I_{nm} - (1/2) A]^{-1}\) exists.
The general algorithm is

\[\begin{align*}
  d_1 &= \frac{m}{T} R_1 k_1 \\
  d_j &= R_2 \left( \frac{1}{T} \sum_{i=1}^{j-1} d_i + \frac{m}{T} R_1 k_j - d_{j-1} \right) + \frac{m}{T} R_1 (k_j - k_{j-1}) \\
  &\quad \text{for } j = 2, 3, \ldots, m
\end{align*}\]

for \(j = 2, 3, \ldots, m\)

where

\[\begin{align*}
  R_1 &= \left( \frac{m}{T} I_n - \frac{1}{2} A \right)^{-1} = A_1^{-1} \\
  R_2 &= A_1^{-1} A = R_1 A
\end{align*}\]

Consequently from (6c) and (10) we have the required column vectors \(C_j\), or

\[\begin{align*}
  C_1 &= \frac{T}{2m} d_1 + g_1 \\
  C_j &= \frac{T}{m} \sum_{i=1}^{j-1} d_i + \frac{T}{2m} d_j + g_j = C_{j-1} + \frac{T}{2m} (d_{j-1} + d_j) \quad \text{for } j = 2, 3, \ldots, m
\end{align*}\]

Substituting (11) into (4) yields the required piecewise-constant solution of the state equation in (1). Note that the \(\phi_j(t)\) differs from zero only in the interval \((j-1)T/m \leq t < jT/m\); therefore, the \(j\)th column vector \(C_j\) is the required piecewise-constant solution in that interval. Another advantage of the proposed method is that \(C_j\) involves only the vectors \(d_i, k_i,\) and \(g_i,\) for \(i = 1 \ldots j,\) whereas the Walsh-function approaches (Corrington 1973, Chen and Hsiao 1975, Rao and Sivakumar 1975) require a whole matrix \(W\) and a whole vector \(\varphi(t)\) in (2).

If \(u(t) = 0\) in (1), (10) and (11) can be expressed by a set of difference equations

\[\begin{align*}
  d(1) &= \frac{m}{T} R_2 \varphi(0) \\
  d(j + 1) &= (I_n + R_2) d(j) \quad \text{for } j = 1, 2, \ldots, m - 1
\end{align*}\]

and

\[\begin{align*}
  c(1) &= \frac{1}{2} (2I_n + R_2) \varphi(0) \\
  c(j + 1) &= c(j) + \frac{T}{2m} (2I_n + R_2) d(j) \quad \text{for } j = 1, 2, \ldots, m - 1
\end{align*}\]

The solution of (12) is

\[d(j) = (I_n + R_2)^{j-1} d(1) = \frac{m}{T} (I_n + R_2)^{j-1} R_2 \varphi(0)\]

Substituting (14) into (13b) yields

\[\begin{align*}
  c(1) &= \frac{1}{2} (2I_n + R_2) \varphi(0) \\
  c(j + 1) &= c(j) + \frac{1}{2} (2I_n + R_2) (I_n + R_2)^{j-1} R_2 \varphi(0)
\end{align*}\]
The solution of (15) is
\[ c(j+1) = c(1) + \frac{1}{2}(2I_n + R_2) \sum_{i=0}^{j-1} (I_n + R_2)^i R_2 x(0) \]
\[ = \frac{1}{2}(2I_n + R_2) \left[ I_n + \sum_{i=0}^{j-1} (I_n + R_2)^i R_2 \right] x(0) \text{ for } j = 1, 2, \ldots \quad (16) \]

Since the trapezoidal rule (as shown in (7)) is used as a base for the numerical integration, or
\[ c(j+1) = \frac{x^*(j+1) + x^*(j)}{2} \quad (17a) \]
where \( x^*(j) \) is the discrete-time solution, therefore
\[ x^*(j+1) = -x^*(j) + 2c(j+1) \quad (17b) \]
Substituting (16) into (17b) we have the required discrete-time equations
\[ x^*(0) = x(0) \]
\[ x^*(1) = (I_n + R_2)x(0) \]
\[ x^*(j+1) = -x^*(j) + (2I_n + R_2) \left[ I_n + \sum_{i=0}^{j-1} (I_n + R_2)^i R_2 \right] x(0) \quad (18a) \]
The solution of (18a) is
\[ x^*(j) = (I_n + R_2)^j x(0) \text{ for } j = 0, 1, 2, \ldots \quad (18b) \]
where \( R_2 = \left( \frac{m}{T} I_n - \frac{1}{2} \Delta T \right)^{-1} A \), \( T \) = the final time, and the sampling period = \( \frac{T}{m} \).
Equation (18b) can be further analysed as
\[ x^*(j) = [I_n + R_2]^j x(0) = \Phi^*(j)x(0) \text{ for } j = 0, 1, 2, \ldots \quad (19a) \]
where
\[ \Phi^*(j) = \text{the transition matrix of a discrete-time system} \]
\[ = [I_n + R_2]^j \]
\[ = [I_n + (I_n - \frac{1}{2} \Delta T)^{-1} A \Delta T]^j \text{ for } j = 0, 1, 2, \ldots \text{, and } \Delta T = \frac{T}{m} \]
\[ = [(I_n - \frac{1}{2} \Delta T)^{-1}(I_n + \frac{1}{2} \Delta T)]^j \]
\[ = [I_n + A \Delta T + \frac{1}{2} (A \Delta T)^2 + \frac{1}{2^2} (A \Delta T)^3 + \frac{1}{2^3} (A \Delta T)^4 + \ldots]^j \]
\[ = \left[ I_n + A \Delta T + \frac{1}{2} (A \Delta T)^2 + \sum_{i=2}^{\infty} \frac{1}{2^{i-1}} (A \Delta T)^i \right]^j \quad (19b) \]
The exact solution of (1) (with \( u(t) = 0 \)) is
\[ x(t) = \exp(At)x(0) = \Phi(t)x(0) \quad (20a) \]
where

\[ \Phi(t) = \exp(At) = \text{the transition matrix of a continuous-time system} \]

\[ = [\exp(\Delta T)]^j \quad \text{for } j = 0, 1, 2, 3, \ldots, \text{ and } t = j\Delta T \]

\[ = \left[ I_n + A\Delta T + \frac{1}{2!} (A\Delta T)^2 + \frac{1}{3!} (A\Delta T)^3 + \frac{1}{4!} (A\Delta T)^4 + \ldots \right]^j \]

\[ = \left[ I_n + A\Delta T + \frac{1}{2!} (A\Delta T)^2 + \sum_{i=3}^{\infty} \frac{1}{i!} (A\Delta T)^i \right]^j \quad (20b) \]

Comparing \( \Phi^*(j) \) with \( \Phi(t) \) we observe that the first three terms of (19b) are equal to those of (20b), while other terms differ in weighting factors \( 1/2^{i-1} \) in (19b) and \( 1/i! = 1/i(i-1)(i-2) \ldots 1 \) in (20b). Therefore \( \Phi^*(j) \) is a good approximation of \( \Phi(t) \) if \( \Delta T \) is small. Also, we observe that \( \Phi^*(j) \) is a finite matrix, while \( \Phi(t) \) is an infinite series of matrices, therefore it is more convenient to evaluate \( \Phi^*(j) \) than \( \Phi(t) \).

It is believed that the derivation of the approximation of \( \Phi(t) \) in (20b) by \( \Phi^*(j) \) in (19b) is new. When \( u(t) \neq 0 \), the approximate discrete-time solution \( x^*(t) \) of \( x(t) \) in (1) can be obtained from (11) and (17b).

3. An illustrative example

Consider the dynamic equation

\[ \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ x(0) = x_0 \end{cases} \]

where

\[ A = \begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}, \quad x(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \]

and

\[ u(t) = \text{unit-step functions} \]

Figure 1. The block-pulse functions and their integrations.
The piecewise-constant solution of the state equation is

\[ x(t) \equiv C\phi(t) \quad (22) \]

The block-pulse functions \( \phi_j(t) \) and the integration of the \( \phi_j(t) \) are shown in Fig. 1. The \( C \) is an unknown matrix to be determined. The steps to determine \( C \) can be listed as follows:

**Step 1**
Choose \( T = 1 \) s and \( m = 4 \). This means that four block-pulse functions \( \phi_j(t) \), \( j = 1, \ldots, 4 \), are used in the interval \( 0 \leq t \leq 1 \), and the sampling period = \( T/m = 0.25 \) s.

**Step 2**
Construct \( G \) in (6b) and \( L \) in (7).

\[
G = [x(0), x(0), x(0), x(0)] = [g_1, g_2, g_3, g_4] = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}
\]

and

\[
L = [L_1, L_2, L_3, L_4] = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}
\]

**Step 3**
Calculate \( K \) in (8b).

\[
K = AG + BL = [k_1, k_2, k_3, k_4] = \begin{bmatrix} 5 & 5 & 5 & 5 \\ 1 & 1 & 1 & 1 \end{bmatrix}
\]

**Step 4**
Determine \( D \) in (10).

\[
D = [d_1, d_2, d_3, d_4]
\]

where

\[
R_1 = \left( \frac{m}{T} I - \frac{1}{4} A \right)^{-1} = \begin{bmatrix} 0.3077 & 0.0510 \\ 0.0769 & 0.1795 \end{bmatrix}
\]

\[
R_2 = R_1 A = \begin{bmatrix} 0.4616 & 0.4102 \\ 0.6154 & -0.5641 \end{bmatrix}
\]

\[
d_1 = \frac{m}{T} R_1 k_1 = \begin{bmatrix} 6.3592 \\ 2.2560 \end{bmatrix}
\]

\[
d_2 = d_1 + R_2 d_1 + \frac{m}{T} R_1 (k_2 - k_1) = \begin{bmatrix} 10.2200 \\ 4.8966 \end{bmatrix}
\]

\[
d_3 = d_2 + R_2 d_2 + \frac{m}{T} R_1 (k_3 - k_2) = \begin{bmatrix} 16.9468 \\ 8.4233 \end{bmatrix}
\]

\[
d_4 = d_3 + R_2 d_3 + \frac{m}{T} R_1 (k_4 - k_3) = \begin{bmatrix} 28.2240 \\ 14.0989 \end{bmatrix}
\]
Step 5
Evaluate the required \( C \) in (11).

\[
C = [C_1, C_2, C_3, C_4]
\]

where

\[
C_1 = \frac{T}{2m} d_1 + g_1 = \begin{bmatrix} 1.7949 \\ 1.2820 \end{bmatrix}
\]

\[
C_2 = C_1 + \frac{T}{2m} (d_2 + d_1) = \begin{bmatrix} 3.8773 \\ 2.1792 \end{bmatrix}
\]

\[
C_3 = C_2 + \frac{T}{2m} (d_3 + d_2) = \begin{bmatrix} 7.3038 \\ 3.8547 \end{bmatrix}
\]

\[
C_4 = C_3 + \frac{T}{2m} (d_4 + d_3) = \begin{bmatrix} 13.0125 \\ 6.6936 \end{bmatrix}
\]

The required piecewise-constant solution in (21) is

\[
x_1(t) \approx 1.7949\phi_1(t) + 3.8773\phi_2(t) + 7.3038\phi_3(t) + 13.0125\phi_4(t)
\]

\[
x_2(t) \approx 1.2820\phi_1(t) + 2.1792\phi_2(t) + 3.8547\phi_3(t) + 6.6936\phi_4(t)
\]

Figure 2. The exact solutions and the approximated solutions.
State-space equations

The exact solution of (21) is

\[ x_1(t) = \frac{1}{2} \exp(2t) - \frac{1}{2} \exp(-5t) \]
\[ x_2(t) = \frac{1}{2} \exp(2t) + \frac{1}{2} \exp(-5t) \]

The response curves of the exact solution and the approximated solution are shown in Fig. 2. The approximate discrete-time solution \( x^*(t) \) of \( x(t) \) in (21) can be obtained from the \( C \) in (22) and (17 b).

If \( u(t) = 0 \) in (21), the exact solution of (21) is

\[ x_1(t) = \frac{1}{2} \exp(2t) - \frac{1}{2} \exp(-5t) \]
\[ x_2(t) = \frac{1}{2} \exp(2t) + \frac{1}{2} \exp(-5t) \]

(23)

From (23) and (18) we can evaluate the exact solution \( x(t) \) and the approximated solution \( x^*(t) \) at samples \( j = 1, 2, 3, 4 \), and sampling period \( T/m = 0.25 \). The results are tabulated as follows:

<table>
<thead>
<tr>
<th>( j )</th>
<th>( t )</th>
<th>( x_1(t) )</th>
<th>( x_1^*(t) )</th>
<th>( x_2(t) )</th>
<th>( x_2^*(t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.25</td>
<td>1.843</td>
<td>1.872</td>
<td>1.065</td>
<td>1.051</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
<td>3.095</td>
<td>3.167</td>
<td>1.589</td>
<td>1.610</td>
</tr>
<tr>
<td>3</td>
<td>0.75</td>
<td>5.119</td>
<td>5.289</td>
<td>2.571</td>
<td>2.651</td>
</tr>
<tr>
<td>4</td>
<td>1.00</td>
<td>8.444</td>
<td>8.818</td>
<td>4.225</td>
<td>4.411</td>
</tr>
</tbody>
</table>

It is interesting to observe that the solution obtained by the four-point approximation is quite satisfactory.

ACKNOWLEDGMENT

This work was supported in part by the U.S. Army Missile Research and Development Command, under Grant DAAK 40–78–C–0017.

REFERENCES

STATE-EQUATION FITTING FROM FREQUENCY-RESPONSE DATA

L. S. SHIEH and M. H. COHEN
Electrical Engineering Department, University of Houston, Houston, Texas 77004

and

R. E. YATES and J. P. LEONARD
Technology, Terminal Homing, U.S. Army Missile Research and Engineering Laboratory, Redstone Arsenal, AL 35809, U.S.A.

(Received 17 March 1978; received for publication 20 July 1978)

Abstract—A method is given for optimally fitting parameter matrices of state equations from the real and imaginary parts of noise free frequency-response data of a multi-input, multi-output, linear dynamic system. It is assumed that all state variables are accessible for measurement. The obtained data may contain measurement errors.

1. INTRODUCTION
Several authors[1-3] have considered the application of frequency response concepts for identification of dynamic systems. The problems of predicting parametric error from frequency response measurements have also been investigated[3-5]. A method is presented here to determine the best estimate, in least mean square sense, of the parameter matrices of the multi-input, multi-output, linear, time-invariant dynamic system equations if all the state variables are accessible for measurement. The obtained data are noise free and contain measurement errors.

2. DERIVATION
The state equations of an asymptotically stable, completely controllable and observable, linear time-invariant system are given by:

\[
\begin{align*}
\dot{\mathbf{x}} &= \mathbf{A}x + \mathbf{B}u \\
\dot{y} &= \mathbf{C}x \\
\mathbf{x}(0) &= \mathbf{0}
\end{align*}
\]

where \( \mathbf{A} \) is a constant \( n \times n \) system matrix, \( \mathbf{x} \) is an \( n \times 1 \) state vector, \( \mathbf{B} \) is a constant \( n \times r \) input matrix, \( \mathbf{C} \) is a constant \( m \times n \) output matrix, \( u \) is an \( r \times 1 \) input vector, and \( \dot{y} \) is an \( m \times 1 \) output vector. Let us define,

\[
\begin{align*}
\hat{\mathbf{B}} &= [\hat{b}_1, \ldots, \hat{b}_r] \\
\hat{\mathbf{C}} &= \begin{bmatrix} \hat{C}_1^T \\ \vdots \\ \hat{C}_m^T \end{bmatrix} \\
\hat{\mathbf{U}} &= \begin{bmatrix} U_1 \\ \vdots \\ U_r \end{bmatrix}
\end{align*}
\]

where \( \hat{b}_i \) is an \( n \times 1 \) column vector and \( \hat{C}_i^T \) is an \( n \times 1 \) row vector.
The Laplace transformation of eqns (1) and (1a) yields,
\[(sI - \tilde{A}) \bar{X}(s) = \tilde{B} U(s)\]  
(3)
and
\[\dot{\bar{Y}}(s) = \bar{C} \bar{X}(s).\]  
(3a)

Successive choice of each of the scalars \(U_e(s)\) in eqn (2b) as an input while the remaining scalar components of \(U(s)\) are zero yields the following set of transfer functions from each of the scalar inputs to the state variables.
\[(sI - \tilde{A}) \hat{X}_e(s) = \tilde{B}_e\]  
(4)
where
\[\hat{T}_e(s) = \frac{1}{U_e(s)} \hat{X}_e(s) \quad \text{and} \quad e = 1, \ldots, r.\]  
(4a)

If the input functions of \(U(s)\) are sinusoidal functions with varying frequencies \(\omega_k\), we obtain the corresponding frequency response data \(\hat{T}_e(\omega_k)\) as follows:
\[\hat{T}_e(\omega_k) = \hat{P}_e(\omega_k) + j\hat{Q}_e(\omega_k), \quad e = 1, \ldots, r\]  
(5)
where \(\hat{P}_e(\omega_k)\) and \(\hat{Q}_e(\omega_k)\) are vectors of the real and the imaginary parts of \(\hat{T}_e(\omega_k)\). Multiplying the steady state portion of eqn (4a) by a normalization constant \(M_e\) (i.e. the magnitude of a sinusoidal input function) we have
\[\hat{X}_e(\omega_k) = M_e \hat{T}_e(\omega_k) = M_e \hat{P}_e(\omega_k) + jM_e \hat{Q}_e(\omega_k), \quad e = 1, \ldots, r\]  
(5a)
and
\[\hat{X}(\omega_k) = \sum_{e=1}^{r} \hat{X}_e(\omega_k)\]  
(5b)
Substituting \(s = j\omega_k\) and eqns (5) and (5a) into eqns (4) and (3a) yields
\[j\omega_k[sI - \tilde{A}] [\hat{P}_e(\omega_k) + j\hat{Q}_e(\omega_k)] = \tilde{B}_e\]  
(6)
\[\dot{\hat{Y}}_e(\omega_k) = \hat{C} [M_e \hat{P}_e(\omega_k) + jM_e \hat{Q}_e(\omega_k)] = \tilde{B}_e(\omega_k) + j\tilde{B}_e(\omega_k)\]  
(6a)
and
\[\dot{\hat{Y}}(\omega_k) = \sum_{e=1}^{r} \dot{\hat{Y}}_e(\omega_k)\]  
(6b)
where \(\tilde{B}_e(\omega_k)\) and \(\tilde{B}_e(\omega_k)\) are vectors of the real and imaginary parts of \(\dot{\hat{Y}}_e(\omega_k)\). After we separate the real and imaginary parts of eqns (6) and (6a) and equate the respective real and imaginary parts, we have
\[\hat{A}\hat{Q}_e(\omega_k) = \omega_k \hat{P}_e(\omega_k)\]  
(7)
\[\hat{A}\hat{P}_e(\omega_k) + \omega_k \hat{Q}_e(\omega_k) = -\tilde{B}_e\]  
(7a)
\[\hat{B}_e(\omega_k) = \hat{C} M_e \hat{P}_e(\omega_k)\]  
(7b)
and
\[\hat{H}_e(\omega_k) = \hat{C} M_e \hat{Q}_e(\omega_k)\]  
(7c)

The parameter matrices \(\hat{A}, \tilde{B}_e\) and \(\hat{C}\) can be obtained as follows:
State-equation fitting from frequency-response data

\[
\dot{\mathbf{A}} = [\omega_1 \mathbf{\hat{p}}_1(\omega_1), \omega_2 \mathbf{\hat{p}}_2(\omega_2), \ldots, \omega_n \mathbf{\hat{p}}_n(\omega_n)][\dot{\mathbf{q}}_1(\omega_1), \dot{\mathbf{q}}_2(\omega_2), \ldots, \dot{\mathbf{q}}_n(\omega_n)]^{-1}
\]  

(8)

\[
\dot{\mathbf{\delta}}_r = -[\dot{\Lambda} \mathbf{\hat{p}}_r(\omega_r) + \omega_r \dot{\mathbf{q}}_r(\omega_r)]
\]

(8a)

\[
\dot{\mathbf{C}} = [\dot{\mathbf{h}}_1(\omega_1), \dot{\mathbf{h}}_2(\omega_2), \ldots, \dot{\mathbf{h}}_n(\omega_n)][\dot{\mathbf{M}}\mathbf{\hat{p}}_1(\omega_1), \dot{\mathbf{M}}\mathbf{\hat{p}}_2(\omega_2), \ldots, \dot{\mathbf{M}}\mathbf{\hat{p}}_n(\omega_n)]^{-1}
\]

(8b)

The data in eqns (8)-(8b) can be chosen so that the matrix inversions exist.

3. EVALUATION OF OPTIMAL PARAMETER MATRICES

If the frequency response data are noise free and measurement error free, then there exist unique parameter matrices \( \dot{\mathbf{A}}, \dot{\mathbf{B}} \) and \( \dot{\mathbf{C}} \). However, in practice, there exist measurement errors even if the system is noise free. As a result, the evaluated parameter matrices have inaccuracies due to the errors. In this paper, optimal parameter matrices are evaluated from the measurement error contaminated data.

Consider \( i \) sets of parameter matrices \( \dot{\mathbf{A}}_i, \dot{\mathbf{B}}_i \) and \( \dot{\mathbf{C}}_i \), which are defined as \( \dot{\mathbf{A}}_i, \dot{\mathbf{B}}_i \) and \( \dot{\mathbf{C}}_i \), and which are evaluated from \( i \) sets of data using one control input or \( r \) control inputs. If many sets of experimental frequency response data can be obtained, then the optimal parameter matrices \( \dot{\mathbf{A}}, \dot{\mathbf{B}}, \) and \( \dot{\mathbf{C}} \) can be obtained from the matrix-mean values, or

\[
\dot{\mathbf{A}} = \frac{1}{k} \sum_{i=1}^{k} \dot{\mathbf{A}}_i
\]

(9)

\[
\dot{\mathbf{B}} = \frac{1}{k} \sum_{i=1}^{k} \dot{\mathbf{B}}_i
\]

(9a)

\[
\dot{\mathbf{C}} = \frac{1}{k} \sum_{i=1}^{k} \dot{\mathbf{C}}_i
\]

(9b)

However, to obtain many sets of frequency response data is often not practical and sometimes impossible. The following technique is proposed to obtain the optimal matrices with fewer sets of frequency response data. Suppose that the system matrices \( \dot{\mathbf{A}}_i, i = 1, \ldots, r \) can be evaluated by \( r \) sets of frequency response data which are obtained from the controllable system by any one input \( U_i \), or by \( r \) sets of inputs, then we construct the following matrix equation,

\[
\dot{\mathbf{E}} \dot{\mathbf{A}} = \dot{\mathbf{F}}
\]

(10)

where

\[
\dot{\mathbf{E}} = \begin{bmatrix}
\dot{\mathbf{A}}_1^{-1} \\
\dot{\mathbf{A}}_2^{-1} \\
\cdot \\
\cdot \\
\cdot \\
\dot{\mathbf{A}}_r^{-1}
\end{bmatrix}
\]

(10a)

and

\[
\dot{\mathbf{F}} = \begin{bmatrix}
\dot{I}_1 \\
\dot{I}_2 \\
\cdot \\
\cdot \\
\cdot \\
\dot{I}_r
\end{bmatrix}
\]

in which \( \dot{\mathbf{A}}_i^{-1} \) are \( n \times n \) inverse matrices of \( \dot{\mathbf{A}}_i \) obtained by the use of eqn (8) and \( \dot{I}_i \) are \( n \times n \) identity matrices. The desired optimal matrix \([6,7] \dot{\mathbf{A}} \) which minimizes the sum of squares of residuals \( \dot{S} = \dot{\mathbf{R}}^T \dot{\mathbf{R}} \), where \( \dot{\mathbf{R}} = \dot{\mathbf{F}} - \dot{\mathbf{E}} \dot{\mathbf{A}} \), is given by

\[
\dot{\mathbf{A}} = (\dot{\mathbf{E}}^T \dot{\mathbf{E}})^{-1} \dot{\mathbf{E}}^T \dot{\mathbf{F}}
\]

(11)

By a similar approach the optimal matrices \( \dot{\mathbf{B}} \) and \( \dot{\mathbf{C}} \) can be obtained as follows:
To obtain \( \hat{b} \), we construct the matrix equation

\[
\hat{G}_s\hat{H}_s = \hat{F}
\]

(12)

where

\[
\hat{G}_s = \begin{bmatrix}
\hat{G}_1^{-1} \\
\hat{G}_2^{-1} \\
\vdots \\
\hat{G}_r^{-1}
\end{bmatrix}
\]

(12a)

in which \( \hat{G}_i^{-1} \) are \( n \times n \) inverse matrices of \( \hat{G}_i \) and the elements at \( j \)th row and \( k \)th column in \( \hat{G}_i \) and \( \hat{H}_s \) are:

\[
\hat{G}_i(j, k) = \hat{b}_n(j, 1) \quad \text{if } j = k
\]

\[
= 0 \quad \text{if } j \neq k
\]

(12b)

\[
\hat{H}_s(j, k) = \hat{b}_s(j, 1) \quad \text{if } j = k
\]

\[
= 0 \quad \text{if } j \neq k
\]

(12c)

\( j = 1, \ldots, n \), \( i = 1, \ldots, r \)

\( k = 1, \ldots, n \), \( e = 1, \ldots, r \).

It is interesting to note the fact that \( \hat{G}_i(j, k) \) and \( \hat{H}_s(j, k) \) are diagonal which greatly reduces the practical problem of calculating \( \hat{H}_s \).

The optimal matrix \( \hat{H}_s \) is

\[
\hat{H}_s = (\hat{G}_s^T\hat{G}_s)^{-1}\hat{G}_s^T\hat{F}.
\]

(13)

The optimal vector \( \hat{b} \) can be obtained from eqn (12). To obtain the optimal row vector \( C^T \) in \( \hat{C} \) we use the following matrix equation:

\[
\hat{D}_s\hat{S}_s = \hat{F}
\]

(14)

where

\[
\hat{D}_s = \begin{bmatrix}
\hat{D}_1^{-1} \\
\hat{D}_2^{-1} \\
\vdots \\
\hat{D}_r^{-1}
\end{bmatrix}
\]

(14a)

in which \( \hat{D}_i^{-1} \) are \( n \times n \) inverse matrices of \( \hat{D}_i \) and the elements of the \( j \)th row and \( k \)th column in \( \hat{D}_i \) and \( \hat{S}_s \) are:

\[
\hat{D}_i(j, k) = \hat{C}_s^T(1, j) \quad \text{if } j = k
\]

\[
= 0 \quad \text{if } j \neq k
\]

(14b)

\[
\hat{S}_s(j, k) = \hat{C}_s^T(1, j) \quad \text{if } j = k
\]

\[
= 0 \quad \text{if } j \neq k
\]

(14c)

\( j = 1, \ldots, n \), \( i = 1, \ldots, r \), \( z = 1, \ldots, m \), \( k = 1, \ldots, n \). Here again the structure of \( \hat{D}_s(j, k) \) is quite favorable for performing the necessary inversions.

The optimal matrix \( \hat{S}_s \) can be obtained from

\[
\hat{S}_s = (\hat{D}_s^T\hat{D}_s)^{-1}\hat{D}_s^T\hat{F}.
\]

(15)
The optimal row vector $\hat{C}^T$ can be obtained from eqn (14c). After obtaining the optimal vectors $\hat{b}$ and $\tilde{C}^T$ we have the optimal input matrix and output matrix, or

$$\hat{B} = [\hat{b}_1, \ldots, \hat{b}_n]$$

and

$$\hat{C} = \begin{bmatrix} \hat{C}_1^T \\ \vdots \\ \hat{C}_n^T \end{bmatrix}$$

4. CONCLUSION

A method for the solution of the difficult problem of identifying a multi-input, multi-output, linear system from measurement error contaminated data has been presented. The resultant parameter matrices are optimal in the least mean square sense. The particular advantage of this technique is the ability to utilize a relatively limited amount of experimental data to obtain the systems dynamic equations. The identification process can be easily performed using digital computers.

Acknowledgements—The authors wish to acknowledge helpful discussion with Mr. Charles Lewis of the U.S. Army Missile Command. This work was supported in part by U.S. Army Missile Command Grant DAAR 40-78-C-0017.

REFERENCES


APPENDIX

I llustrative example. For a known dynamic system described by the following state equation,

$$\dot{X} = \hat{A}X + \hat{B}U$$

$$\dot{Y} = \hat{C}X$$

where

$$\hat{A} = \begin{bmatrix} -1 & -1 \\ 2 & -4 \end{bmatrix} \quad \hat{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \hat{C} = \begin{bmatrix} 1 & 0.5 \end{bmatrix}$$

the error contaminated frequency response of Table 1 was obtained.

Assuming a unity magnitude for the excitation function or $M$, and $M_e$ equal unity and by following eqns (8), (8a) and (8b), we have

$$\hat{A}_1 = \begin{bmatrix} -1.018087614 & -0.967404614 \\ 1.963874775 & -3.934809232 \end{bmatrix}$$

$$\hat{A}_2 = \begin{bmatrix} -0.993028846 & -1.015416667 \\ 2.00528842 & -4.00249999 \end{bmatrix}$$

Table 1. Frequency response data

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>$\hat{P}_1(s\omega)$</th>
<th>$\hat{Q}_1(s\omega)$</th>
<th>$\hat{P}_2(s\omega)$</th>
<th>$\hat{Q}_2(s\omega)$</th>
<th>$\hat{Y}_{11}(s\omega)$</th>
<th>$\hat{H}_{11}(s\omega)$</th>
<th>$\hat{Y}_{21}(s\omega)$</th>
<th>$\hat{H}_{21}(s\omega)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.658 -0.077 -0.821 0.104</td>
<td>0.82 -0.10 -0.91 0.13</td>
<td>0.326 -0.055 -0.158 0.060</td>
<td>0.65 -0.11 -0.32 0.12</td>
<td>0.269 -0.346 -0.288 0.442</td>
<td>0.29 -0.44 -0.20 0.51</td>
<td>0.038 -0.192 0.173 0.135</td>
<td>0.08 -0.38 0.35 0.27</td>
</tr>
<tr>
<td>2.0</td>
<td>0.658 -0.077 -0.821 0.104</td>
<td>0.82 -0.10 -0.91 0.13</td>
<td>0.326 -0.055 -0.158 0.060</td>
<td>0.65 -0.11 -0.32 0.12</td>
<td>0.269 -0.346 -0.288 0.442</td>
<td>0.29 -0.44 -0.20 0.51</td>
<td>0.038 -0.192 0.173 0.135</td>
<td>0.08 -0.38 0.35 0.27</td>
</tr>
</tbody>
</table>
for \( z = 1 \), we have
\[
\begin{bmatrix}
0.00220068 \\ 0.00005657
\end{bmatrix}
\]
and when \( z = 2 \), we have
\[
\begin{bmatrix}
0.00000071 \\ 0.00000102
\end{bmatrix}
\]

When \( z = 1 \), we have
\[
\begin{bmatrix}
1.00100654 \\ 0.47471686
\end{bmatrix}
\]
and for \( z = 2 \) we obtain
\[
\begin{bmatrix}
0.02201300 \\ 1.94943372
\end{bmatrix}
\]

Applying eqns (11), (13) and (15) we have the optimal parameter matrices
\[
\begin{bmatrix}
1.009 \\ 0.496
\end{bmatrix}
\]
An algebraic method to determine the common divisor, poles and transmission zeros of matrix transfer functions

L. S. SHIEH†, Y. J. WEI† and J. M. NAVARRO‡

A purely algebraic method which uses the matrix Routh algorithm and its reverse process of the algorithm is presented to decompose a matrix transfer function into a pair of right co-prime polynomial matrices or left co-prime polynomial matrices. The poles and transmission zeros of the matrix transfer function are determined from a pair of relatively prime polynomial matrices. Also, the common divisor of two matrix polynomials can be obtained by using the matrix Routh algorithm and the matrix Routh array.

1. Introduction

The properties and applications of poles and transmission zeros of a multi-variable system have been extensively studied in recent years by many researchers (Desoer and Schulman 1974, Kwakernaak and Sivan 1972, Rosenbrock 1970, Moore and Silverman 1972, Wolovich 1972, 1973, Davison and Wang 1974, Francis and Wonham 1975, Sinswat et al. 1976, Kouvaritakis and MacFarlane 1976, Wang and Desoer 1972). Desoer and Schulman (1974) defined the poles as real or complex numbers for which the responses of a circuit or system to a series of singular inputs are purely exponential. The transmission zeros are also defined as real or complex numbers for which the transmission of some particular signals is completely blocked. The role of poles in the analysis and synthesis of circuits and systems is well known, and in recent years the transmission zeros are found to be important in many aspects of feedback control theory (Desoer and Schulman 1974, Kwakernaak and Sivan 1972, Rosenbrock 1970, Moore and Silverman 1972, Wolovich 1972, 1973, Davison and Wang 1974, Francis and Wonham 1975, Sinswat et al. 1976, Kouvaritakis and MacFarlane 1976, Wang and Desoer 1972). Therefore, it is useful and desirable to have an effective method to determine the locations of these poles and transmission zeros. Several methods are available to locate the positions of these poles and zeros (Kwakernaak and Sivan 1972, Rosenbrock 1970, Moore and Silverman 1972, Wolovich 1972, 1973, Davison and Wang 1974, Francis and Wonham 1975, Sinswat et al. 1976, Kouvaritakis and MacFarlane 1976). However, most of the suggested approaches (Rosenbrock 1970, Moore and Silverman 1972, Wolovich 1972, 1973, Davison and Wang 1974, Francis and Wonham 1975, Sinswat et al. 1976, Kouvaritakis and MacFarlane 1976) are derived for the systems which are represented by state equations in the time domain. The major disadvantage of most time-domain approaches is that the computation may not be very attractive if the dynamic systems are
of high order. When a given multivariable system is described by a matrix transfer function that might have a high degree common divisor (the common factor) of the numerator and denominator matrix polynomials, the order of the corresponding state equations is in general very high. Therefore, most time-domain approaches may be difficult to apply. In this paper, a purely algebraic method is derived in the frequency domain for the determination of the poles and transmission zeros of a matrix transfer function. The matrix Routh algorithm and the reverse process of the algorithm (Shieh and Gaudiano 1974, Shieh 1975, Shieh et al. 1975) are used to decompose an \( n_0 \times n_i \) rational matrix transfer function \( T(s) \) into \( D_1(s)^{-1}N_1(s) \) and \( N_1(s)D_1(s)^{-1} \), where the polynomial matrices \( D_1(s) \) and \( N_1(s) \) with appropriate size are left co-prime and \( N_1(s) \) and \( D_1(s) \) right co-prime. When \( n_0 = n_i \), the poles (the transmission zeros) of the \( T(s) \) are determined from the zeros of the determinant \( D_1(s) \) or \( D_1(s)^{-1}N_1(s) \) when \( n_0 \neq n_i \), or the matrix Routh algorithm is of ill-conditioned case, the determinant of the rectangular polynomial matrices \( N_1(s) \) and \( N_1(s) \) cannot be obtained. An \( n_i \times n_0 \) matrix transfer function that might have a high degree common divisor is high, the dimension of the resultant matrix or the equivalent test matrix is very high. As a result, the effectiveness of their approaches is less.

2. The matrix Routh algorithm and the matrix Routh array

In a single variable system it is well known that the poles and zeros of a transfer function can be determined from the respective denominator and numerator polynomials that are relatively prime. The Routh algorithm and the Routh array (Fryer 1959) are often used to determine the common factor of the two polynomials in order to determine the pair of relatively prime polynomials. In this paper we extend the concept to a multivariable system that is described by a matrix transfer function. Let us define that \( R \) and \( C \) denote the field of real numbers and complex numbers, respectively, and \( R[s] \) and \( R(s) \) the sets of all polynomials and rational functions in the field of complex variables having real coefficients. We also define that \( R[s]^{n_0 \times n_i} \) and \( R(s)^{n_i \times n_0} \) are the sets of all \( n_0 \times n_i \) matrices with elements in \( R[s] \) and \( R(s) \), respectively.

Consider the following matrix transfer function \( T(s) \in R(s)^{n_i \times n_0} \) which is a product of a polynomial matrix \( A_1(s) \in R[s]^{n_0 \times n_i} \) and the inverse of another polynomial matrix \( A_2(s) \in R[s]^{n_i \times n_0} \), where \( q = \min (n_0, n_i) \):

\[
T(s) = A_1(s)A_2(s)^{-1} = [A_{11} + A_{12}s + \ldots + A_{1n_0}s^{n_0-1}] \\
\quad \times [A_{21} + A_{22}s + \ldots + A_{n_0,1}s^{n_i-1}]^{-1}
\]  

(1 a)
Poles and transmission zeros of matrix transfer functions

\[ T(s) = A_2(s)^{-1} A_3(s) = [A_{11} + A_{12}s + \ldots + A_{1,n+1}s^n]^{-1} \times [A_{21} + A_{22}s + \ldots + A_{2,n}s^n]^{-1} \]  

(1 b)

where

\[ A_2(s) = \sum_{i=1}^{n} A_{2,i}s^{i-1} \quad \text{and} \quad A_3(s) = \sum_{i=1}^{n+1} A_{3,i}s^{i-1} \]

The matrix coefficients in the \( A_2(s) \) and \( A_3(s) \) are expressed by the double subscript notation as \( A_{2,i} \in \mathbb{R}^{n \times n} \) and \( A_{3,i} \in \mathbb{R}^{n \times n} \) for the use of the matrix Routh algorithm. If the \( T(s) \) is expressed as follows:

\[ T(s) = \frac{1}{\Delta_0(s)} \Phi(s) \]  

(2)

then

\[ \Delta_0(s) = \sum_{i=1}^{n+1} a_is^{i-1}, \quad A_i(s) = \sum_{i=1}^{n+1} a_is^{i-1} = \sum_{i=1}^{n+1} A_{3,i}s^{i-1} \]

and

\[ \Phi(s) = \sum_{i=1}^{n} \Phi_is^{i-1} = \sum_{i=1}^{n} A_{2,i}s^{i-1} \]

where \( \Delta_0(s) \in \mathbb{R}[s] \) is a polynomial and \( I_n \in \mathbb{R}^{n \times n} \) is an identity matrix. By using the following matrix Routh algorithm and the reverse process of the algorithm, the \( T(s) \) can be factored into \( D_j(s)N_j(s) \) and \( N_j(s)D_j(s) \), where \( D_1(s), N_1(s), N_2(s), \ldots, N_n(s) \) are polynomial matrices of appropriate size. The matrix Routh algorithm (Shieh and Gaudiano 1974) and its reverse process (Shieh et al. 1975) of the algorithm for a multivariable system \( n_i = n_0 \) are expressed as follows:

\[
\begin{align*}
H_1 &= A_{11} \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1,n}A_{1,n+1} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2,n} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} \\ A_{n+1,1} & \cdots & \cdots & \cdots & \cdots \\ A_{2n,1} & A_{2n+1,1} & \cdots & \cdots & \cdots \\ \end{pmatrix} \\
H_2 &= A_{21}A_{22}^{-1} \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1,n}A_{1,n+1} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2,n} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} \\ A_{n+1,1} & \cdots & \cdots & \cdots & \cdots \\ A_{2n,1} & A_{2n+1,1} & \cdots & \cdots & \cdots \\ \end{pmatrix} \\
H_3 &= A_{31}A_{32}^{-1} \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1,n}A_{1,n+1} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2,n} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} \\ A_{n+1,1} & \cdots & \cdots & \cdots & \cdots \\ A_{2n,1} & A_{2n+1,1} & \cdots & \cdots & \cdots \\ \end{pmatrix} \\
H_{n-1} &= A_{n-1,1}A_{n-1,2}^{-1} \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1,n}A_{1,n+1} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2,n} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} \\ A_{n+1,1} & \cdots & \cdots & \cdots & \cdots \\ A_{2n,1} & A_{2n+1,1} & \cdots & \cdots & \cdots \\ \end{pmatrix} \\
H_n &= A_{n1}A_{n+1,1}^{-1} \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1,n}A_{1,n+1} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2,n} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} \\ A_{n+1,1} & \cdots & \cdots & \cdots & \cdots \\ A_{2n,1} & A_{2n+1,1} & \cdots & \cdots & \cdots \\ \end{pmatrix} \\
\end{align*}
\]

(3 a)

The \( H_i \) in eqn. (3 a) are the matrix quotients. The block elements of the first and second rows of eqn. (3 a) are the matrix coefficients of eqn. (1 a). The block elements of the subsequent rows are evaluated by the following matrix Routh algorithm:

\[
H_i = A_{i,i}A_{i+1,1}^{-1} \quad \text{for} \quad i = 1, 2, \ldots, 2k \quad \text{and} \quad k \leq n
\]

rank \( A_{i+1,1} = n_i = n_0 \)

(3 b)

\[
A_{i,j} = A_{i-j,j+1} - H_iA_{i-j-1,j+1} \quad \text{for} \quad j = 1, 2, \ldots; \quad i = 3, 4, \ldots
\]
When the two matrix polynomials $A_1(s)$ and $A_2(s)$ have no common factor, the matrix Routh array will terminate normally (i.e. we have $2n$ matrix quotients). When the two matrix polynomials have a common factor (the common divisor), the matrix Routh array in eqn. (3 a) will terminate prematurely, and the last non-vanishing row consists of the matrix coefficients of the common factor $B(s)$ in the original matrix polynomials $A_1(s)$ and $A_2(s)$. If we have $2k$ matrix quotients $H$, we can construct a pair of relatively prime matrix polynomials, $N_r(s)$ and $D_r(s)$, by using the reverse process of the matrix Routh algorithm in eqn. (3 b):

$$P_{2k+1,1} = I$$
$$P_{i,1} = H_i P_{i+1,1} \text{ for } i = 2k, 2k-1, \ldots, 2, 1$$
$$P_{i-2,j+1} = P_{i,j} + H_{i-2} P_{i-1,j+1} \text{ for } i = 2k+1, 2k, \ldots, 3; \quad j = 1, 2, \ldots, k$$

The $T(s)$ in eqn. (1) is

$$T(s) = A_2(s) A_1(s)^{-1} = N_r(s) B(s) [D_r(s) B(s)]^{-1} = N_r(s) D_r(s)^{-1}$$

$$= [P_{21} + P_{22}s + \ldots + P_{2,k}s^{k-1}] [P_{11} + P_{12}s + \ldots + P_{1,k+1}s^k]^{-1}$$

The procedure can be well illustrated by the following numerical example.

**Example 1**

Consider that the common divisor $B(s)$ and a pair of relatively prime matrix polynomials $N_r(s)$ and $D_r(s)$ of the following matrix transfer function $T(s)$ are required:

$$T(s) = A_2(s) A_1(s)^{-1} = N_r(s) B(s) [D_r(s) B(s)]^{-1} = N_r(s) D_r(s)^{-1}$$

where

$$A_1(s) = A_{11} + A_{12}s + A_{13}s^2 + A_{14}s^3$$

$$= \begin{pmatrix} 3 & 3 \\ 1 & -3 \end{pmatrix} + \begin{pmatrix} 7 & 1 \\ 1 & -5 \end{pmatrix} s + \begin{pmatrix} 5 & -3 \\ 0 & -3 \end{pmatrix} s^2 + \begin{pmatrix} 1 & -1 \\ 0 & -1 \end{pmatrix} s^3$$

$$A_2(s) = A_{21} + A_{22}s + A_{23}s^2$$

$$= \begin{pmatrix} 6 & -2 \\ -1 & -3 \end{pmatrix} + \begin{pmatrix} 6 & -6 \\ -1 & 0 \end{pmatrix} s + \begin{pmatrix} 2 & -4 \\ 0 & -1 \end{pmatrix} s^2$$

$n_0 = n_r = 2$ and $n = 3$

The matrix Routh array is
Poles and transmission zeros of matrix transfer functions

\[
\begin{align*}
A_{11} &= \begin{pmatrix} 5 & -3 \\ 0 & -3 \end{pmatrix}, & A_{14} &= \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\
A_{21} &= \begin{pmatrix} 2 \\ 0 \end{pmatrix}, & A_{24} &= \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\
A_{31} &= \begin{pmatrix} 7 & 1 \\ -1 & 5 \end{pmatrix}, & A_{34} &= \begin{pmatrix} 1 & -1 \\ 6 & 0 \end{pmatrix} \\
A_{41} &= \begin{pmatrix} 4 & -3 \\ -6 & 8 \end{pmatrix}, & A_{44} &= \begin{pmatrix} 1 & -1 \\ 6 & 0 \end{pmatrix}
\end{align*}
\]

\[
H_1 = \begin{pmatrix} 0.3 & -1.2 \\ 0.8 & 3 \end{pmatrix}, & H_2 = \begin{pmatrix} 1.5 & 0.25 \\ 0.7 & 7 \end{pmatrix}
\]

\[
H_3 = \begin{pmatrix} 0.25 & 0.45 \\ 0.7 & 7.2 \end{pmatrix}, & H_4 = \begin{pmatrix} 0.25 & 0.25 \\ 0.5 & 2.25 \end{pmatrix}
\]

\[
A_{11} = \begin{pmatrix} 3 & 3 \\ 1 & 3 \end{pmatrix}, & A_{14} = \begin{pmatrix} 1 & 1 \\ 3 & 3 \end{pmatrix}
\]

\[
A_{21} = \begin{pmatrix} 4 & 2.3 \\ 0 & 3.2 \end{pmatrix}, & A_{24} = \begin{pmatrix} 0.5 & 0.4375 \\ 0.25 & 0.53125 \end{pmatrix}
\]

\[
A_{31} = \begin{pmatrix} 0.5 & 0.5625 \\ 0.5 & 0.5625 \end{pmatrix}, & A_{34} = \begin{pmatrix} 1 & 0.5 \\ 1 & 0.5 \end{pmatrix}
\]

\[
A_{41} = \begin{pmatrix} 0.5 & 0.4375 \\ 0.25 & 0.53125 \end{pmatrix}, & A_{44} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}
\]
The matrix Routh array terminates prematurely because the only one block element $A_{45}$ in the sixth row is a null matrix; therefore, the common divisor $B(s)$ in $T(s)$ is

$$B(s) = A_{41} + A_{45}s = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} s \tag{6b}$$

By using the matrix quotients $H_1 \ldots H_4$ in eqn. (6a) and applying the algorithm in eqn. (3c) we have

$$N_1(s) = P_{11} + P_{22} = \begin{pmatrix} 2 & 4 \\ -2 & 1 \end{pmatrix} + \begin{pmatrix} 2 & 2 \\ 0 & 1 \end{pmatrix} s$$

and

$$D_1(s) = P_{11} + P_{12} + P_{13}s^2 = \begin{pmatrix} 3 & 0 \\ -1 & 2 \end{pmatrix} + \begin{pmatrix} 4 & 0 \\ -1 & 3 \end{pmatrix} s + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} s^2 \tag{6c}$$

In order to show that the $B(s)$ in eqn. (6b) is a common divisor of $A_1(s)$ and $A_4(s)$ in eqn. (5) we replace $A_{i,j}$ in eqn. (3b) by $P_{ij}$ in eqn. (6c) and apply the algorithm in eqn. (3b) to eqn. (6c). Thus we have the following alternative matrix Routh array that has the same matrix quotients $H_1$ as eqn. (6a):
\[ H_1 = \begin{pmatrix} 0.3 & -1.2 \\ 0.3 & 0.8 \end{pmatrix} \]
\[ H_2 = \begin{pmatrix} 1.5 & 1.9375 \\ -0.25 & 0.71875 \end{pmatrix} \]
\[ H_3 = \begin{pmatrix} 6.45 & 0.7 \\ -6.8 & 7.2 \end{pmatrix} \]
\[ H_4 = \begin{pmatrix} 0.5 & 0.0625 \\ 0.25 & 0.28125 \end{pmatrix} \]

\[ P_{11} = H_1 H_2 H_3 H_4 = \begin{pmatrix} 3 & 0 \\ -1 & 2 \end{pmatrix} \]
\[ P_{12} = H_1 H_2 + H_3 H_4 = \begin{pmatrix} 4 & 0 \\ -1 & 3 \end{pmatrix} \]
\[ P_{13} = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \]

\[ P_{22} = H_3 + H_4 = \begin{pmatrix} 2 & 2 \\ 0 & 1 \end{pmatrix} \]
\[ P_{31} = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \]

(6d) Variance and transmission zeroes of matrix transfer functions
The justification for the \( B(s) \) in eqn. (6 b), which is the common divisor of the matrix polynomials in eqn. (5), can be proved by the following induction method. Since the matrix Routh algorithm is developed from the repeated process of long division of two polynomial matrices, the reverse process of the algorithm can be applied to eqns. (6 a) and (6 d) to obtain the following identities:

\[
\begin{align*}
A_{31} + A_{32}s &= P_{31}(A_{31} + A_{32}s) = P_{31}B(s) = B(s) \\
A_{41} + A_{42}s &= H_4(A_{41} + A_{42}s) + sA_{41} \\
&= H_4B(s) \\
&= P_{41}B(s) \\
A_{51} + A_{52}s + A_{53}s^2 &= H_5(A_{51} + A_{52}s + A_{53}s^2) + s(A_{51} + A_{52}s) \\
&= H_5B(s) + sP_{51}B(s) \\
&= (P_{51} + P_{33}s)B(s) \\
A_{61} + A_{62}s + A_{63}s^2 + A_{64}s^3 &= H_6(A_{61} + A_{62}s + A_{63}s^2 + A_{64}s^3) + s(A_{61} + A_{62}s + A_{63}s^2) \\
&= H_6B(s) + s(P_{61} + P_{63}s)B(s) \\
&= (P_{61} + P_{33}s + P_{12}s^2)B(s)
\end{align*}
\]

From the last two equations in eqn. (6 e) we observe that

\[
\begin{align*}
A_2(s) &= N_r(s)B(s) \\
A_1(s) &= D_r(s)B(s)
\end{align*}
\]

Therefore \( B(s) \) is the common divisor of the two polynomial matrices \( A_1(s) \) and \( A_2(s) \).

When \( n_i \neq n_0 \) and rank \( A_{i,1} \neq q \) in eqn. (3 b), the matrix Routh algorithm in eqn. (3 b) cannot be directly applied. The matrix Routh algorithm and its reverse process of the algorithm in eqn. (3) are modified and discussed by the following case studies.

(1) \( T(s) = A_3(s)A_4(s)^{-1} \)

where

\[
A_2(s) = \sum_{i=1}^{n_0} A_{2,i}e^{s} \quad \text{and} \quad A_1(s) = \sum_{i=1}^{n_1} A_{1,i}e^{s}
\]

Case 1

\[
T(s) = A_2(s)A_1(s)^{-1} = N_r(s)B(s)[D_r(s)B(s)]^{-1} = N_r(s)D_r(s)^{-1}
\]

where

\[
N_r(s) = \sum_{i=1}^{k+1} P_{i,0}e^{s} \quad \text{and} \quad D_r(s) = \sum_{i=1}^{n_1} B_{i}e^{s}
\]
Poles and transmission zeros of matrix transfer functions

The $B(s)$ is the common right divisor of the $A_2(s)$ and $A_1(s)$. For the use of the matrix Routh algorithm, the matrix coefficients in the $N_i(s)$ and $D_i(s)$ are expressed by the double-subscript notation as $P_{k,i}$ and $P_{n,k}$ which can be obtained by the algorithms as follows.

The matrix Routh algorithm is

$$H_i = A_{i,1} A_{i+1,1}^{-1}, \quad i = 1, 2, ..., 2k \quad \text{and} \quad k \leq n$$

$$\text{rank } A_{i,1} = n_i$$

$$A_{i,j} = A_{i-2,j+1} - H_{i-2} A_{i-1,j+1}, \quad j = 1, 2, ..., i = 3, 4, ...$$

The constant matrices $H_i$ with appropriate size are called the matrix quotients. If $n_0 > n_i$, the pseudo-inverse of $A_{i+1,1} = A_{i+1,1}^{-1} = [A_{i+1,1} A_{i+2,1}]^{-1} A_{i+1,1}^T$ is the left inverse of $A_{i+1,1}$. The reverse process of the matrix Routh algorithm is

$$P_{2k+1} = I_{n_0}$$

$$P_{l,1} = H_l P_{l+1,1}, \quad l = 2k, 2k - 1, ..., 2, 1$$

$$P_{l-2,l+1} = P_{l,1} + H_{l-2} P_{l-1,l+1}, \quad i = 2k + 1, 2k, ..., 3, \quad j = 1, 2, ..., k$$

Case 2

$n_0 < n_i$, $T(s) \in R(s)^{n \times n}$, $A_2(s) \in R[s]^{n \times n}$, $A_1(s) \in R[s]^{n \times n}$

$$T(s) = A_2(s) A_1(s)^{-1} = D_i(s)^{-1} B(s) [N_i(s)^{-1} B(s)]^{-1} = D_i(s)^{-1} N_i(s)$$

where

$$N_i(s) \in R[s]^{n \times n}, N_i(s) = \sum_{i=1}^{k} Q_{i,k} s^{i-1}; \quad D_i(s) \in R[s]^{n \times n}$$

$$D_i(s) = \sum_{i=1}^{k+1} Q_{i,k+1} s^{i-1}, Q_{0,k+1} = I_{n_0}; \quad B(s) \in R[s]^{n \times n}, B(s) = \sum_{i=1}^{n-k+1} B_i s^{i-1}$$

For the use of the matrix Routh algorithm, the matrix coefficients in the $N_i(s)$ and $D_i(s)$ are expressed by the double-subscript notation as $Q_{k,i}$ and $Q_{n,k}$ which can be obtained by the algorithms as follows.

The matrix Routh algorithm in eqn. (8b) is applied to determine the matrix quotients $H_i$:

$$H_i = A_{i,1} A_{i+1,1}^{-1}, \quad i = 1, 2, ..., 2k \quad \text{and} \quad k \leq n$$

$$\text{rank } A_{i,1} = n_0$$

$$A_{i,j} = A_{i-2,j+1} - H_{i-2} A_{i-1,j+1}, \quad j = 1, 2, ..., i = 3, 4, ...$$

The new reverse algorithm is

$$Q_{2k+1} = I_{n_0}$$

$$Q_{l,1} = Q_{l+1,1} H_l, \quad l = 2k, 2k - 1, ..., 2, 1$$

$$Q_{l-2,l+1} = Q_{l,1} + Q_{l-3,l+1} H_{l-2}, \quad i = 2k + 1, 2k, ..., 3, \quad j = 1, 2, ..., k$$
\[ T(s) = A_1(s)^{-1}A_2(s) \]  
\[ A_1(s) = \sum_{i=1}^{n} A_{1,i}s^{i-1} \quad \text{and} \quad A_2(s) = \sum_{i=1}^{n+1} A_{2,i}s^{i-1}. \]

**Case 1**

\[ n_0 < n, \quad T(s) \in \mathcal{R}[s]^{n \times n}, A_1(s) \in \mathcal{R}[s]^{n \times n}, A_2(s) \in \mathcal{R}[s]^{n \times n} \]

\[ T(s) = A_1(s)^{-1}A_2(s) = [B(s)D_1(s)]^{-1}B(s)N_1(s) = D_1(s)^{-1}N_1(s) \]

where

\[ N_1(s) \in \mathcal{R}[s]^{n \times n}, \quad N_1(s) = \sum_{i=1}^{k} Q_{2,i}s^{i-1}; \quad D_1(s) \in \mathcal{R}[s]^{n \times n} \]

\[ D_1(s) = \sum_{i=1}^{k} Q_{1,i}s^{i-1}, \quad Q_{1,k+1} = I_{n_0}; \quad B(s) \in \mathcal{R}[s]^{n \times n}, \quad B(s) = \sum_{i=1}^{n-k+1} B_i s^{i-1} \]

The matrix coefficients in the \( D_1(s) \) and \( N_1(s) \) are expressed by the double-subscript notation as \( Q_{1,i} \) and \( Q_{2,i} \), which can be determined by the following algorithms.

The new matrix Routh algorithm is

\[ M_i = A_{i+1,1}^{-1}A_{i,1}, \quad i = 1, 2, ..., 2k \quad \text{and} \quad k \leq n \]

\[ \text{rank } A_{i,1} = n_0 \]

\[ A_{i,j} = A_{i-x,j+1} - A_{i-x,j}M_{i-j}, \quad j = 1, 2, ..., i = 3, 4, ... \]

The constant matrices \( M_i \) with appropriate size are called the matrix quotients. If \( n_0 < n, \quad A_{i+1,1}^{-1} = A_{i+1,1}^T[A_{i+1,1}M_{i+1,1}^T]^{-1} \) is the right inverse of the \( A_{i+1,1} \).

The reverse algorithm in (9 c) is applied to determine the \( Q_{1,i} \) and \( Q_{2,i} \).

\[ Q_{2k+1,1} = I_{n_0} \]

\[ Q_{l,1} = Q_{l+1,1}M_{i}, \quad l = 2k, 2k-1, ..., 2, 1 \]

\[ Q_{l-k,1} = Q_{l-i+1}M_{l-j}, \quad i = 2k+1, 2k, ..., 3, \quad j = 1, 2, ..., k \]

**Case 2**

\[ n_0 > n, \quad T(s) \in \mathcal{R}[s]^{n \times n}, A_1(s) \in \mathcal{R}[s]^{n \times n}, A_2(s) \in \mathcal{R}[s]^{n \times n} \]

\[ T(s) = A_1(s)^{-1}A_2(s) = [B(s)N_1(s)^{-1}]^{-1}B(s)D_1(s)^{-1} = N_1(s)D_1(s)^{-1} \]

where

\[ N_1(s) \in \mathcal{R}[s]^{n \times n}, \quad N_1(s) = \sum_{i=1}^{k} P_{1,i}s^{i-1}; \quad D_1(s) \in \mathcal{R}[s]^{n \times n} \]

\[ D_1(s) = \sum_{i=1}^{k} P_{1,i}s^{i-1}, \quad P_{1,k+1} = I_{n_0}; \quad B(s) \in \mathcal{R}[s]^{n \times n}, \quad B(s) = \sum_{i=1}^{n-k+1} B_i s^{i-1} \]

The matrix coefficients in the \( D_1(s) \) and \( N_1(s) \) are expressed by the double-subscript notation as \( P_{1,i} \) and \( P_{2,i} \), which can be obtained by the algorithms as follows.
The matrix Routh algorithm in eqn. (11 b) is applied to determine the matrix quotients $M_i$:

$$M_i = A_{i+1,1}^{-1} A_{i,1}, \quad i = 1, 2, ..., 2k \quad \text{and} \quad k \leq n$$

$$\text{rank } A_{i,1} = n_i$$

$$A_{i,j} = A_{i+k+1,j} - A_{i+k,j} M_{i-2} \quad j = 1, 2, ..., i = 3, 4, ...$$

The reverse algorithm in eqn. (8 c) is applied to determine the $P_{k,i}$ and $P_{k,i+1}$:

$$P_{2k+1} = I_n$$

$$P_{l,1} = M_l P_{l+1,1}, \quad l = 2k, 2k-1, ..., 2, 1$$

$$P_{l-k,j+1} = P_{l,j} + M_l P_{l-1,j+1}, \quad i = 2k+1, 2k, ..., 3, \quad j = 1, 2, ..., k$$

By using Gilbert’s theorem it has been shown (Shieh and Gaudiano 1975) that the dynamic state equations, which are constructed using $2k$ matrix quotients $H_i$ or $M_i$ that are obtained from the matrix Routh algorithms, are minimal realizations of the $T(s)$. The minimal dimension of the system matrix is $qk \times qk$, where $q = \min(n_i, n_b)$ and $k = \text{rank } T(s) = r_g$. The rank $T(s)$ can be determined from the Hankel matrix (Ho and Kalman 1966). By using the same $2k$ matrix quotients $H_i$ or $M_i$ and performing the reverse process of the matrix Routh algorithm, the monic polynomial matrices $D_i(s)$ and $D_j(s)$ are obtained and shown in eqns. (8), (9), (11) and (12). The highest power of $s$ in the $D_i(s)$ and $D_j(s)$ is $k$ and the matrix coefficients of $s^k$ (i.e. $P_{l,k+1}$ and $Q_{1,k+1}$) are identity matrices having size $q \times q$. Therefore

$$\det D_i(s) = \det D_j(s) = \sum_{i=1}^{k+1} d_i s^{i-1} \Delta d(s)$$

The highest power of $s$ in the monic polynomial $d(s)$ is $kq$, which is the rank of the $T(s)$. As a result, the $d(s)$ in eqn. (13) is the characteristic polynomial of the $T(s)$ and the polynomial matrices $D_i(s)$ and $N_j(s)$ are right co-prime and the $D_i(s)$ and $N_j(s)$ are left co-prime.

From the above discussion we also note that the necessary condition for the existence of the matrix Routh algorithm is that the ratio (denoted as $k$) of the rank $T(s)$ and the minimal dimension of the $T(s)$, $r_g/q = k$, is an integer. If the ratio $r_g/q$ is not an integer or it is an integer but the condition (rank $A_{i,1} = q$) in the matrix Routh algorithm in eqns. (8)–(12) is violated due to the ill-conditioned matrix $A_{i,1}$, then the original $T(s)$ should be modified. $T(s)$ is modified by adding another transfer-function matrix $T_0(s) = 1/g(s)K$ whose rank is of $(kq - r_g)$, where $k$ is the nearest integer and the scalar polynomial $g(s)$ is not a factor of the $\Delta_i(s)$ in eqn. (2). The $K$ is a constant matrix with appropriate dimension. The modified system $T'(s) \in R(s)^{n \times n}$ is

$$T'(s) = T(s) + \frac{1}{g(s)} K$$

where rank $[(1/g(s))K] = kq - r_g$ and rank $T'(s) = kq$. It is noted that the addition of $(1/g(s))K$ to the $T(s)$ does not affect the locations of the poles of the $T(s)$.
3. Determination of poles and transmission zeros

By using the matrix Routh algorithm the $T(s)$ is factored into $D_1(s)^{-1}N_1(s)$ and $N_2(s)D_2(s)^{-1}$, where $D_1(s)$ and $N_1(s)$ are left co-prime and $N_2(s)$ and $D_2(s)$ are right co-prime. When $n_1 = n_2 = q$, Desoer and Schulman (1974) have shown that the transmission zeros of the $T(s)$ are the zeros of the scalar polynomial $n(s)$, or

$$n(s) = \det N_1(s) = \det N_2(s) = 0 \quad (15)$$

where $N_1(s)$, $N_2(s)$, $D_1(s)$ and $D_2(s)$ are polynomial matrices; $N_1(s)$ and $N_2(s)$ are $q \times q$; $D_1(s)$ and $D_2(s)$ are $q \times q$. The poles of the $T(s)$ are the zeros of the following characteristic equation:

$$A(s) = \det D_1(s) = \det D_2(s) = 0 \quad (16)$$

When $r_q / q \neq k$ (an integer), the matrix Routh algorithm cannot be applied. The procedure shown in eqn. (14) can be applied to determine a pair of relatively prime polynomial matrices $D_1'(s)$ and $N_1'(s)$ or $N_2'(s)$ and $D_2'(s)$ as follows:

$$T_1'(s) = D_1'(s)^{-1}N_1'(s) = N_2'(s)D_2'(s)^{-1} \quad (17 a)$$

The poles of the $T_1'(s)$ can be determined from the following equations:

$$\det D_1'(s) = \det D_2'(s) = \{g(s)\}^{q-r_q}P(s) = 0 \quad (17 b)$$

where the $g(s)$ is the polynomial used in eqn. (14). The poles of the $T(s)$ are the zeros of $P(s) = 0$.

When $r_q / q = k$ (an integer) and $n_q \neq n_i$, the $N_1(s)$ and $N_2(s)$ obtained from the matrix Routh algorithm are not square polynomial matrices of size $n_q \times n_q$ and $n_i \times n_i$. Therefore the transmission zeros cannot be directly determined from eqn. (15). The transmission zeros of the $T(s)$ can be determined from the invariant zeros of the determinants of all $q \times q$ minors of the $N_1(s)$ or $N_2(s)$ in eqn. (15) where $q = \min (n_q, n_i)$. However, when the $n_i$ is much larger than the $n_q$ and vice versa, there exist many $q \times q$ minors which are expressed by polynomial matrices.

It is a cumbersome task to find the determinants of these $q \times q$ minors and to determine the roots of many polynomials. This difficulty can be overcome by applying Desoer and Schulman’s (1974) theorem. The transmission zeros of the $T(s)$ are obtained from the invariant poles of two generalized inverses of the modified $T(s)$. In this paper we present a procedure to obtain the generalized inverses of the modified $T(s)$ so that the transmission zeros of the $T(s)$ can be determined. The steps are shown as follows.

Step 1. Modify the $T(s)$ and formulate the generalized inverses $T_1^{*}(s)$, $i = 1, 2$ of the modified $T(s)$, or

$$T_i^{*}(s) = [m_i(s)T(s)]^{-1} = A_1(s)[m_i(s)A_2(s)]^{-1}, \quad n_q > n_i \quad (18 a)$$

$$= [m_i(s)T(s)]^{-1} = [m_i(s)A_2(s)]^{-1}A_1(s), \quad n_q < n_i \quad (18 b)$$

where $T_i^{*}(s) \in R(s)^{n_i \times n_q}$, $A_1(s) \in R(s)^{n_q \times n_q}$, $A_2(s) \in R(s)^{n_q \times n_q}$, $m_i(s) \in R(s)$. The monic polynomials $m_i(s) \in R(s)$, $i = 1, 2$, should not be the factors of the $\Delta_n(s)$ in eqn. (2) and of the $g(s)$ in eqn. (14), but are chosen in such a way that the power of the
Poles and transmission zeros of matrix transfer functions

961

s in the polynomial matrices $m_i(s)A_j(s)$ in eqn. (18) is 1° higher than that of the $A_j(s)$. This modification does not affect the transmission zeros of the $T(s)$ but it makes the matrix Routh algorithm applicable.

Step 2. Modify the $T_i(s)$ when $r_g/q \neq k$ (an integer).

The process of this modification is shown in eqn. (14). The additions of $(1/g(s))K$, $i = 1, 2$ to the $T_i(s)$, $i = 1, 2$, do not affect the locations of the poles of the $T_i(s)$ or the transmission zeros of the $T(s)$. After performing some matrix operations we have the generalized inverses of the modified $T(s)$ denoted as $T_i^+(s)$, $i = 1, 2$:

$$
T_i^+(s) = [g_i(s)A_i(s) + m_i(s)KA_i(s)][g_i(s)m_i(s)A_i(s)]^{-1}, \quad n_o \geq n_i \quad (19a)
$$

$$
= [g_i(s)m_i(s)K][g_i(s)A_i(s) + m_i(s)KA_i(s)], \quad n_o \leq n_i \quad (19b)
$$

Step 3. Determine two pairs of relatively prime polynomial matrices.

The algorithms in eqns. (9) and (12) can be applied to obtain two pairs of left co-prime polynomial matrices denoted as $D_i(s)$ and $N_i(s)$, $i = 1, 2$ or right co-prime polynomial matrices $N_i(s)$ and $D_i(s)$, $i = 1, 2$:

$$
T_i^+(s) = D_i(s)^{-1}N_i(s), \quad n_o \geq n_i \quad (20a)
$$

$$
= N_i(s)^{-1}D_i(s), \quad n_o \leq n_i \quad (20b)
$$

Step 4. Select the required transmission zeros.

The poles of the $T_i^+(s)$, $i = 1, 2$, are

$$
\det D_i(s) = (m_i(s))^n_k(s) = 0, \quad n_o \geq n_i \quad (21a)
$$

$$
\det D_i(s) = (m_i(s))^n_k(s) = 0, \quad n_o \leq n_i \quad (21b)
$$

The required transmission zeros of the $T(s)$ are the invariant poles of the $T_i^+(s)$, $i = 1, 2$. They are the zeros of polynomial $n(s)$ in eqn. (21), or

$$
n(s) = 0 \quad (21c)
$$

When the $T(s)$ is not a strictly proper rational matrix transfer function, eqn. (19) can also be used to determine the poles and transmission zeros of the $T(s)$.

Example 2

Consider that the poles and transmission zeros of the following matrix transfer function $T(s)$ are required to be determined:

$$
T(s) = A_2(s)A_1(s) \quad (22)
$$

where

$$
A_2(s) = \begin{bmatrix}
s^3 + 6s^2 + 11s + 12 & s^3 + 9s + 20 \\
s^3 + 4s - 5 & s^3 + 3s^2 - 7s + 15 \\
s^3 + 6s^2 + 13s + 10 & 3s^2 + 5s + 26
\end{bmatrix}
$$
and
\[A_1(s) = \begin{bmatrix} s^4 + 5s^3 + 11s^2 + 13s + 6 & s^3 + 8s^2 + 17s + 10 \\ s^3 + 2s^2 - s - 2 & s^4 + s^3 - s^2 + 5s + 6 \end{bmatrix}\]

\[n_o = 3 \quad \text{and} \quad n_i = 2\]

It is difficult to apply most time-domain approaches to this problem. By using the proposed algorithms in eqn. (8), the \(T(s)\) can be factored into a pair of relatively prime polynomial matrices \(N_i(s)\) and \(D_i(s)\):
\[T(s) = A_1(s)A_1(s)^{-1} = [N_i(s)B(s)]^{-1} = [D_i(s)B(s)]^{-1}\]

where
\[N_i(s) = \begin{bmatrix} s + 4 & 0 \\ 0 & s + 5 \\ s + 4 & 2 \end{bmatrix} \quad \text{and} \quad D_i(s) = \begin{bmatrix} s^2 + 3s + 2 & 0 \\ 0 & s^3 + 3s + 2 \end{bmatrix}\]

Following eqn. (16) we have the required poles of the \(T(s)\):
\[\Delta(s) = \det D_i(s) = (s + 1)^2(s + 2)^2 = 0\]  
\[\text{or} \quad s_1 = s_2 = -1 \quad \text{and} \quad s_3 = s_4 = -2\]

It is interesting to note that the common factor \(B(s)\) in eqn. (23) is
\[B(s) = \begin{bmatrix} s^2 + 2s + 3 & s + 5 \\ s - 1 & s^2 - 2s - 2 \end{bmatrix}\]

Since \(n_o > n_i\) and the determination of the transmission zeros of the \(T(s)\) are required, we construct the generalized inverses \(T_1^*(s)\) in eqn. (18) from the modified \(T(s)\) in eqn. (23), and apply the proposed algorithm in eqn. (9) to decompose the \(T_1^*(s)\) into two pairs of left co-prime polynomial matrices \(D_{ii}^*(s)\) and \(N_{ii}^*(s)\), \(i = 1, 2\) in eqn. (20 a) as follows:
\[T_1^*(s) = A_1(s)[m_1(s)A_1(s)]^{-1} = D_i(s)[m_1(s)N_i(s)]^{-1} = D_{ii}^*(s)^{-1}N_{ii}^*(s)\]

where \(m_1(s) = s^3 + s + 1\) is not a factor of the \(\Delta(s)\) in eqn. (24 a):
\[D_{ii}^*(s) = \begin{bmatrix} s^3 + 5s^2 + 5s + 4 & 1.1965s^3 + 2.1649s + 0.81525 \\ 0 & s^3 + 5.88073s^2 + 5.88073s + 4.88073 \end{bmatrix}\]

\[N_{ii}^*(s) = \begin{bmatrix} 0.4275s^3 + 1.329s + 0.981 & 0.9177s + 0.057 & 0.573s^2 + 1.671s + 1.02 \\ 0.0797s^3 + 0.1407s + 0.574 & s^3 + 3.04s + 1.72 & -0.08s^2 - 0.143s + 0.574 \end{bmatrix}\]

\[T_2^*(s) = A_1(s)[m_2(s)A_2(s)]^{-1} = D_i(s)[m_2(s)N_i(s)]^{-1} = D_{ii}^*(s)^{-1}N_{ii}^*(s)\]
where \( m_2(s) = s^2 - 10 \) is not a factor of the \( \Delta(s) \) in eqn. (24 a):

\[
D_{\text{irs}}(s) = \begin{bmatrix}
 s^2 + 4s^2 - 10s - 40 & -3.854672s^2 + 38.54672 \\
0 & s^2 + 0.4438694s^2 - 10s - 4.438694
\end{bmatrix}
\]

\[
N_{\text{irs}}(s) = \begin{bmatrix}
 2.24s^2 + 4.45s + 1.14 & -1.38s - 1.78 & -1.24s^2 - 1.45s + 0.59 \\
1.02s^2 - 0.36s - 0.04 & s^2 + 0.49s + 0.16 & -1.02s^2 + 0.36s + 0.038
\end{bmatrix}
\]

Following eqn. (21 a) yields

\[
\text{det } D_{\text{irs}}(s) = (s^2 + s + 1)(s^2 + 4+ 8.8073) = m_2(s)n(s)k_1(s) = 0 \tag{27 a}
\]

\[
\text{det } D_{\text{irs}}(s) = (s^2 + 10)(s^2 + 4+ 4.4386943) = m_2(s)n(s)k_2(s) = 0 \tag{27 b}
\]

The common divisor \( n(s) \) of the det \( D_{\text{irs}}^*(s) \) and det \( D_{\text{irs}}^*(s) \) is the common factor \( n(s)R(s) \) in eqns. (27 a) and (27 b). The transmission zeros of the \( T(s) \) which are the invariant poles of the \( T_1^*(s) \) in eqn. (26) are the zeros of the polynomial \( n(s) \), or

\[
n(s) = s + 4 = 0 \tag{28}
\]

The transmission zero is \( s = -4 \).

The computation involves only arithmetic operations of small-size matrices. Therefore, it is believed that the proposed method is computationally superior to most time-domain approaches if the system is given in the frequency domain.

4. Conclusion

A purely algebraic method has been presented for factorizing a rational matrix transfer function into a pair of relatively prime polynomial matrices and for determining the poles and transmission zeros of a multivariable system. Also, the common divisor of two matrix polynomials can be determined from the matrix Routh algorithm and the matrix Routh array. When a matrix transfer function that might have a high degree common divisor is given, the method proposed in this paper is computationally superior to most time-domain methods because the proposed algorithm only deals with arithmetic operations of small-size matrices. The matrix Routh algorithm has been extended for general cases \( n_i \neq n_0 \) and \( n_i = n_0 \).

Acknowledgments

The authors wish to thank a reviewer for the valuable comments and suggestions. This work was supported in part by U.S. Army Missile Command, Redstone Arsenal, Alabama, Grant DAAK 40–78–C–0017.

References


Poles and transmission zeros of matrix transfer functions


Some Sufficient and Some Necessary Conditions for the Stability of Multivariable Systems

Some sufficient and some necessary conditions for the stability of a class of multivariable systems represented by matrix polynomials are derived. A new linear block transformation is also established for transforming an observable block companion form to the block Schwarz form.

I Introduction

The accurate description of most practical systems, for example both a small semiactive terminal homing missile system [1] and an aircraft system [2], result in high order coupled multivariable differential equations. Linear representations of these systems are by a set of coupled high-order differential equations or a matrix differential equation. A primary concern in the design of these multivariable systems is the stability problem. One conventional approach is to formulate the system into a high dimensional state equation, then to determine the stability by either directly evaluating the roots of the scalar characteristic polynomial, indirectly applying the Routh criterion [3], or application of Jury's inner theory [4] on the characteristic polynomial. However, the determination of a characteristic polynomial for a large dimensional system is tedious. Moreover, if a system is modeled as a matrix differential equation, it is more natural to determine the stability directly from the matrix polynomial than indirectly from a scalar polynomial. Some approaches have been proposed to determine the stability of a multivariable system directly from the matrix polynomial. Papakostantinou [5] suggested a scheme for testing stability of polynomial matrices. In his work, a recursive algorithm was developed to compare the normalized largest eigenvalues with unity. However, the method requires the calculation of the eigenvalues of largest moduli for indirectly determining the stability of polynomial matrices. Recently, Shieh and Sacheti [6] partially extended the scalar Routh criterion [3] to the matrix case. In this work, it is shown that if a matrix polynomial $B(s) = I s^n + B_{n-1} s^{n-1} + \ldots + B_1 s + B_0$ is given, a matrix Routh array can be constructed by using the following matrix Routh algorithm:

$$C_{i,j} = B_{n-i-j} \quad j = 1, 2, 3, \ldots, l$$

where $i = \begin{cases} \frac{n+1}{2} n & \text{even} \\ \frac{n+1}{2} n & \text{odd} \end{cases}$

$$C_{n,0} = 1$$

$$C_{i,j} = C_{i-2,j+1} - H_{i-2,j} C_{i+1,j+1} \quad i = 1, 2, \ldots, j = 3, 4, \ldots$$

$$H_i = C_i (C_{i+1})^{-1} \quad i = 1, 2, \ldots, n$$

$$\det(C_{n+1,1}) \neq 0$$ (1)

A sufficient condition for stability of the $\det[B(s)]$ is that all the “matrix quotients” $H$ be real, symmetric, positive definite matrices. Note that this sufficient condition deals only with $H$, and not $C_{i,j}$ (the block elements in the first column of the matrix Routh array). Liapunov theory with the state equation in the controllable block companion (controllable phase-variable) form was used to derive their result.

In this paper, we develop two approaches for determining the stability of a class of multivariable systems. One approach uses the “matrix quotient” $M_i$ that are developed from an alternate matrix Routh algorithm and a state equation in the observable block companion form [7]. The other approach uses the block elements in the first column of the matrix Routh array. Two sufficient conditions and three necessary conditions are derived for the stability of matrix polynomials, thereby partially extending the scalar Routh criterion to the matrix Routh criterion.

II Sufficient Conditions

The objective of this paper is to establish the criteria for the stability of the following matrix differential equations.

$$\sum_{i=1}^{n} B_i \frac{d}{dt} z(t) = [0], B_{n+1} = I$$ (2a)

Transactions of the ASME
and

\[ D_{t+2}(0) = [\alpha_{i-1}] \quad i = 1, 2, 3, \ldots, n \]  

where \( x(t) \) is the \( m \)-dimensional state vector. \( H, I, \) and \( 0 \) are \( m \times m \) real constant matrix, identity matrix and null matrix, respectively. For the scalar case, it is well known that a system is asymptotically stable if and only if the Routh array elements in the first column are all positive. Shieh and Sacheti [6] partially extended the Routh criteria to the matrix case and derived the following sufficient and necessary conditions for the system in equation (2) from the observable block companion form.

Let us rewrite the system in equation (2) into the following observable block companion form:

\[ [\ddot{z}] = [B][\dot{z}] \]  
\[ [z(0)] = [\alpha] \]

where

\[ [B] = \begin{bmatrix} 0 & 0 & 0 & -B_1 \\ 1 & 0 & 0 & -B_2 \\ 0 & 1 & 0 & -B_3 \\ 0 & 0 & 1 & -B_n \end{bmatrix} \]

The dimensions of the matrix \([B]\), the block elements \(B_j\), and state vector \([\ddot{z}]\) are \((nm) \times (nm)\), \(m \times m\), and \((nm) \times 1\), respectively. Equation (3) can be transformed into the block Schwarz form by using the following linear transformation:

\[ [\dot{y}] = [K_1][y] \]

where

\[ [K_1] = \begin{bmatrix} 1 & D_{n-3}D_{n-3,1}^{-1} & D_{n-3,2}D_{n-3,2,1}^{-1} & \cdots & 0 & D_{n-3,4}D_{n-3,4,1}^{-1} \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & \ddots & \ddots \\ 0 & 0 & 0 & 0 & \cdots & \ddots \\ 0 & 0 & 0 & 0 & \cdots & \ddots \end{bmatrix} \]

and

\[ [A] = \begin{bmatrix} 0 & -A_1 & 0 & 0 & 0 \\ 0 & 1 & -A_2 & 0 & 0 \\ 0 & 0 & 1 & -A_3 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & -A_{n-1} \\ 0 & 0 & 0 & 1 & -A_n \end{bmatrix} \]

The dimension of each block element in \([A]\) and \([K_1]\) is \(m \times m\.

The construction of the matrix Routh array in equation (5a) is as follows. Arrange the matrix coefficients of the given matrix polynomial in equation (2a) in the first two rows of the array shown in equation (5b). A new matrix \(M_i\) is obtained by the matrix multiplication \(D_{i-1}D_0\) where \(D_0\) and \(D_n\) are the block elements in the first column of the array. The block elements in the third row are generated from the \(M_i\) and the block elements in the fourth row are generated from the block elements..
in the first two rows as follows: First, each block element in the second row is postmultiplied by $M_i$, then, subtract each resulting matrix from each block element in the first row; finally, shift each block element so obtained one column left and drop the zero-first block element to form the third row. The second and the obtained third row are then used as starting rows to generate the new matrix $M_3$ and the block elements in the fourth row. Repeating the processes to the $n + 1$ row yields the complete matrix Liou equation. When any $D_{n+1}$ matrices other than $D_0$ or $D_{n+1}$ in equation (5c) are singular, another set of $D_{n+1}$ can be chosen from the new matrix polynomial that is the product of the original matrix polynomial and an asymptotically stable matrix polynomial. Thus a new matrix Liou equation can be obtained and the stability of the original matrix polynomial is preserved because the stability of the original matrix polynomial is invariant under this transformation. Shieh and Sacheti [6] have shown that if $H_i = C_i C_{i+1}^{-1}$ for $i = 1, 2, \ldots, n$ in equation (1) are positive definite, then the system in equation (2) is stable.

Here, we show similar results when replacing $H_i$ by $M_i$. Note that a positive definite matrix means a matrix is real, symmetric and positive definite.

**Theorem 1.** If $\{ M_i \}$, $i = 1, 2, \ldots, n$ in equation (5) are positive definite, then the system in equation (2) is stable.

**Proof.** Performing the following new transformation

$$[y] = [K_1][z] \quad (6)$$

on equation (4) yields

$$[z] = [K_2][z] \quad (7a)$$

where

$$[K_2] = \left[ \begin{array}{ccccc} D_{n+1} & 0 & 0 & 0 & 0 \\ 0 & D_{n+1} & 0 & 0 & 0 \\ 0 & 0 & D_n & 0 & 0 \\ 0 & 0 & 0 & D_{n-1} & 0 \\ \end{array} \right] \quad (7b)$$

and

$$[F] = \left[ \begin{array}{cccccc} 0 & -M_n & 0 & 0 & 0 & 0 \\ -M_{n-1} & 0 & -M_{n-1} & 0 & 0 & 0 \\ 0 & M_{n-1} & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & -M_{r-1} & 0 \\ 0 & 0 & 0 & M_r & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{r-1} & 0 \\ \end{array} \right] \quad (7c)$$

It is noticed that, if each block element in the matrix $[F]$ in equation (7c) were a scalar, then the matrix $[F]$ would be a matrix of the Schwarz form [8]. Since the elements are blocks, the matrix $[F]$ in equation (7c) is a block Schwarz form matrix.

The linear transformation matrix $[K]$ between $x$ coordinates and $z$ coordinates is

$$[z] = [K][x] = [K_1][K_2][x] \quad (7d)$$

Now, consider the following quadratic equation:

$$V = [z]^T [F][z] \quad (8a)$$

where

$$P = \left[ \begin{array}{cccc} M_n & 0 & 0 & 0 \\ 0 & M_{n-1} & 0 & 0 \\ 0 & 0 & M_n & 0 \\ 0 & 0 & 0 & M_1 \\ \end{array} \right] \quad (8b)$$

and $T$ in equation (8a) designates transpose.

Since $[M_i]$ are positive definite which implies $P$ is positive definite, $V$ is positive definite. The derivative of $V$ is

$$\dot{V} = [\dot{z}]^T [P] [\dot{z}] = -[\dot{z}]^T [Q][\dot{z}] = -[\dot{z}]^T [R][\dot{z}] \quad (9a)$$

where

$$[Q] = \left[ \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2I \end{array} \right], \quad [R] = \left[ \begin{array}{cccc} 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array} \right] \quad (9b)$$

rank $[Q] = \text{rank} [R] = m$. From equations (8) and (9) we can see that $V$ is a Liapunov function. Hence, we conclude that the system in equation (2) is stable.

From the result obtained in Theorem 1, we establish another sufficient condition for the stability of the system in equation (2) by using the block elements $D_{n+1}$ in the matrix Routh array in equation (5) instead of the $M_i$ in equation (5).

**Theorem 2.** If $\{ D_i \}$, $i = 2, 4, 6, \ldots$, are positive definite, the eigenvalues of $\{ D_i \}$, $i = 1, 3, 5, \ldots$, are positive and real, and $\{ D_i D_{i+1} \}$, $i = 1, 3, 5, \ldots$, $\{ D_i D_{i+1} \}$, $i = 2, 4, 6, \ldots$, are Hermitian, the system (equation [2]) is stable.

In order to prove Theorem 2, we need the following lemma which is due to KyFan [9] [p. 137].

**Lemma 1.** Let $K_1$ be positive definite and $K_1$ such that $K_1 K_3$ is Hermitian. Then $K_1 K_3$ is positive definite if and only if the eigenvalues of $K_1$ are positive and real. In the following lemmas, we switch the conditions on $K_1$ and $K_3$ yielding the same result.
Lemma 3. If \( K_1 \) is positive definite and \( K_1K_4 \) is symmetric, then \( K_1^{-1}K_4 \) is symmetric.

Proof. Since \((K_1K_3)^T = K_3^TK_1^T = K_1K_3 \) which implies \( K_1^T = K_3^{-1}K_4 \). Hence \((K_1^T)^T = K_1^T(K_1^{-1})^T = K_1K_3^{-1}K_4^{-1}K_1 = K_1^{-1}K_4 \); i.e., \( K_1^{-1}K_4 \) is symmetric.

Proof of Theorem 2. By lemma 3, we know that \( D_{i+1}^{-1}D_{i-1} \) is symmetric for \( i = 1, 2, \ldots \). By lemma 2 or 3, we know that \( M_i = D_{i+1}^{-1}D_{i-1} \) is positive definite. Hence, the system in equation (2) is stable following the results of Theorem 1.

In order to show an application of Theorem 1 and Theorem 2, let us consider the following matrix characteristic equation:

\[ A \theta^2 + B \theta + C = 0 \quad (10a) \]

where

\[ A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \]

If we arrange the matrices \( A, B, \) and \( C \) in equation (10a) by following the matrix Routh algorithm of equation (1), we obtain

\[ C_1 = A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_4 = C = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \]

\[ H_1 = \frac{1}{8} \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} < 0 \]

\[ H_4 = \frac{1}{2} \begin{bmatrix} 6 & 0 \\ 2 & 3 \end{bmatrix} < 0 \]

\[ C_2 = C = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \]

In this case, no conclusion can be drawn from the sufficient condition established by Shieh and Sacheti [6]. However, if we arranged the matrices \( A, B, \) and \( C \) according to equation (5), we have

\[ D_1 = A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D_4 = C = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \]

\[ M_1 = \frac{1}{8} \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} < 0 \]

\[ D_2 = B = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \]

\[ M_1 = \frac{1}{2} \begin{bmatrix} 17 & 1 \\ 3 & 1 \end{bmatrix} < 0 \]

\[ D_3 = C = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \]

From Theorem 1, we see that the system is stable.

This example shows the application of Theorem 2. Let us consider the following matrix characteristic equation:

\[ A \theta^2 + B \theta + C_1 = 0 \quad (11a) \]

where

\[ D_1 = A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D_4 = B_1 = \begin{bmatrix} 25.77 & 13.7 \\ 13.7 & 7.3 \end{bmatrix} \]

In Bellman [9], p. 67, it is shown that, if \( A_1, B_1, \) and \( C_1 \) are positive definite, then the roots of \( \det \{A_1 \theta^2 + B_1 \theta + C_1\} = 0 \) have negative real parts. But in this example, no conclusion can be made from Bellman's results. However, we know that

\[ D_1 \cdot D_4 = A_1 \cdot B_1 = \begin{bmatrix} 25.77 & 13.7 \\ 13.7 & 7.3 \end{bmatrix} \]

and

\[ D_2 \cdot D_4 = C_1 \cdot B_1 = \begin{bmatrix} 3 & 1.63 \\ 1.63 & 0.9 \end{bmatrix} \]

which are symmetric, \( B_1 \) is positive definite, and the eigenvalues of \( A_1 \) and \( C_1 \) are positive and real. Therefore, from Theorem 2 we conclude that the system in equation (11a) is stable. Although only second order matrix polynomials with \( 2 \times 2 \) matrix coefficients are illustrated in the examples, the theory is valid for high order matrix polynomials.

### III Necessary Conditions

In this section we establish some necessary conditions for the stability of multivariable systems. The failure to satisfy the necessary conditions for stability is equivalent to the sufficient conditions for the instability of the same systems; i.e.,
If \(|M_i| \neq 0\) is negative definite, negative semi-definite, or indefinite, then the system in equation (2) is unstable.

**Proof.** Suppose the system is asymptotically stable and one of \([M_i]\) is negative definite, negative semi-definite, or indefinite. Since the stability is invariant under the linear transformation and the matrix \(F\) in equation (7) is a stable matrix. Let us consider the following equation:

\[
X \frac{dF}{dt} X = -Q
\]

where \(Q\) is a matrix defined in equation (8b). By Theorem 4 in Bellman [9] (p. 239) and the theorems in Anderson [10] and Barnett [11] (p. 80), we know that equation (12) has a unique solution. Since \(Q\) is positive semi-definite and rank \([Q] = \text{rank} \[R]\) = \(m\), we conclude that the solution \(X\) of equation (12) is also positive semi-definite. Furthermore, \(X\) is positive definite if the pair \([F, R]\) is observable. It is easy to verify that the matrix \(P\) which was defined in equation (8b) satisfies equation (12). Therefore \(X = P\) is positive semi-definite or positive definite. This implies that at least one of the \([M_i]\) is positive semi-definite and others positive definite or all positive definite. This contradicts our assumption that one of the \([M_i]\) is negative definite, negative semi-definite, or indefinite. Hence the system in equation (2) is unstable if one of the \([M_i]\) is negative definite, negative semi-definite, or indefinite.

To show an application of Theorem 3, consider the example [5].

\[
A_1 \frac{dy}{dt} + B_1 \frac{dy}{dt} + C_{1y} = 0
\]

where

\[
A_1 = \begin{bmatrix} 5 & 1 \\ 1 & 10 \end{bmatrix}, B_1 = \begin{bmatrix} 49 & 0 \\ 0 & 49 \end{bmatrix}, C_1 = \begin{bmatrix} -5 & -1 \\ -1 & 10 \end{bmatrix}
\]

Applying equation (5) yields the matrix Routh array and \(M_i\):

\[
M_i = \begin{bmatrix} 5 & 1 \\ 10 & 1 \end{bmatrix}
\]

\[
M_i = \begin{bmatrix} 49 & 1 \\ 10 & 1 \end{bmatrix}
\]

\[
M_i = \begin{bmatrix} -5 & 1 \\ 1 & -1 \end{bmatrix}
\]

\[
M_i = \begin{bmatrix} -5 & 1 \\ 1 & -1 \end{bmatrix}
\]

\[
D_i = A_i = \begin{bmatrix} 5 & 1 \\ 1 & 10 \end{bmatrix}, D_i = C_i = \begin{bmatrix} -5 & -1 \\ -1 & 10 \end{bmatrix}
\]

\[
D_i = B_i = \begin{bmatrix} 49 & 0 \\ 0 & 49 \end{bmatrix}
\]

\[
D_i = C_i = \begin{bmatrix} -5 & -1 \\ -1 & 10 \end{bmatrix}
\]

\[
M_i = \begin{bmatrix} 5 & 1 \\ 10 & 1 \end{bmatrix}
\]

\[
M_i = \begin{bmatrix} 49 & 1 \\ 10 & 1 \end{bmatrix}
\]

\[
M_i = \begin{bmatrix} -5 & 1 \\ 1 & -1 \end{bmatrix}
\]

\[
M_i = \begin{bmatrix} -5 & 1 \\ 1 & -1 \end{bmatrix}
\]

Hence, the system is unstable.

The next criteria is another necessary condition which we state as follows.

**Theorem 5.** If \(D_{ii} > 0\) and \(D_{i+} < 0\), or \(D_{ii} < 0\) and \(D_{i+} > 0\), and \(D_{ii} > 0\) and \(D_{i+} < 0\) are defined in equation (14), then the system in equation (2) is unstable.

**Proof.** Since the system in equation (2) has the matrix characteristic equation \([D(s)]\) in equation (14), then we expand the det\([D(s)]\). We find the constant term is equal to \(B_i = \text{det} D_{ii}\). If \(D_{ii} > 0\) and \(D_{i+} < 0\), this implies that the coefficient of the polynomial det\([D(s)]\) has a negative sign. We can then conclude that the det \([D(s)] = 0\) has a solution with a positive real part. Hence the system is unstable.

IV Conclusion

Some necessary and some sufficient conditions have been developed for the stability of a class of multivariable systems. A linear block transformation has been derived for transforming the coordinates of an observable block companion form to the coordinates of a block Schwarz form. The new method has partially extended the scalar Routh criterion to the matrix Routh criterion to a class of multivariable systems.

Acknowledgments

The authors wish to express their gratitude for the valuable remarks and suggestions of the reviewers. This work was supported in part by U. S. Army Missile Research and Development Command, DAAK 40-78-C-0017.

References


Transactions of the ASME