A Note on Continued Fraction Inversion by Routh's Algorithm

S. Vittal Rao and S. S. Lamba

Abstract—This correspondence presents an algorithm for finding continued fraction inversion by Routh's algorithm. A recursive relation is given to implement the algorithm in a very simple way. This approach requires less computation when compared with the existing techniques. The method is demonstrated by means of a numerical example.

Introduction

Chen and Shieh [1] have developed a method for reducing the order of linear dynamical systems. In their method of model simplification a given higher order transfer function is expanded into a continued fraction expansion and simplification is achieved by ignoring some of the coefficients. After the inverse procedure has been performed on the truncated continued fraction, the simplified model is obtained. Hence, the simplification process requires an algorithm for continued fraction inversion. The continued fraction inversion is also required in network analysis.

Derivation of the transfer function, directly from the continued fraction representation, is tedious and involves many multiplications. It is desirable to have a computerized algorithm to implement the continued fraction inversion. Chen and Shieh [4] have developed an algorithm for this purpose, through a linear transformation between the second cauer form in state space coordinates and phase-variable coordinates, and by utilizing the Routh's algorithm. Their approach involves the evaluation of a number of determinants. Shieh, Schneider, and Rilliarns [3] have presented an alternate approach for obtaining the Routh array involution. Their approach involves the evaluation of a number of determinants.

This technical note presents a new and simple algorithm for developing the entire Routh array and the corresponding transfer function from the known continued fraction coefficients. It is believed that the method proposed in this note requires less computations than the existing methods [2]-[4].

Continued Fraction Inversion

Let the truncated transfer function be represented by

\[ T(s) = \frac{1}{H_1 + \frac{1}{H_3 + \frac{1}{H_5 + \cdots}}}. \]

where \( H_l \) through \( H_{2n} \) are called "partial coefficients." The problem is to find the transfer function corresponding to the representation (1).

The continued fraction of (1) can be represented as

\[ T(s) = \frac{A_{2n} + A_{1+2p} + A_{1+4p} + \cdots + A_{n-1} s^{-1}}{A_{n} + A_{n+1} s + A_{n+2} s^2 + \cdots + A_{2n-1} s^{2n-2} + s^n}. \]  

(2)

The constants \( A_{j-1}, A_{j} \) and \( A_{j-1}, A_{j} \) are to be evaluated from the partial coefficients \( H_l \). It is easy to see that for the type of continued fraction form shown in (1), the coefficient of \( s^0 \) in (2) is always equal to unity and that the order of the numerator cannot be greater than \( n - 1 \).

The first and second rows of Routh table (3) are written by copying the coefficients of the denominator and numerator of (2), respectively. The subsequent rows are developed by using Routh algorithm.

Routh Table:

| \( A_n \) | \( A_{n-2} \) | \( A_{n-3} \) | \( \ldots \) | \( A_{n-1} \) | \( 1 \) |
| \( A_{n+1} \) | \( A_{n-1} \) | \( A_{n-3} \) | \( \ldots \) | \( A_{n-2} \) | \( 0 \) |
| \( A_{n+2} \) | \( A_{n-2} \) | \( \ldots \) | \( 0 \) |
| \( d_{n} \) | \( A_{n} \) | \( A_{n-2} \) | \( \cdots \) | \( 1 \) |
| \( A_{n+1} \) | \( s_{n-1} \) | \( A_{n-1} \) | \( \cdots \) | \( 0 \) |

(3)

If the last two elements of the first two rows of the Routh table are as shown then the last element of the first column is equal to unity. That is

\[ A_1 = 1. \]

(4)

The partial coefficients are related to the first column of Routh table by the following equations [5]:

\[ H_{p} = \frac{A_{2n-1} s^{2n-1}}{A_{2n-1} s^{2n-1}} \quad p = 2n, 2n - 1, \ldots, 1, \quad H_0 \neq 0. \]

(5)

Once we know the partial coefficients \( H_{-1}, H_{-2} \), the first column of (3) can be evaluated with the help of (4) and (5), starting from the bottom of the table.

The recursive relationship between the other elements of the Routh table is given by

\[ A_{j-k+1} = A_{j-k} + \frac{A_{j-k} A_{j-k+1}}{A_{j-k+1}} \]

(6)

for

\[ j = 2n, 2n - 1, \ldots, 3 \]

\[ k = 1, 2, \ldots, n - 1 \]

with

\[ A_{0} = 0, \quad A_{j-k} = 0, \quad A_{j-k} = 0, \quad A_{n-2} = 0, \quad A_{n-1} = 0, \quad A_{n} = 0, \quad A_{j-k+1} = 0, \]

(7a)

(7b)

With the help of (6) and (i), all the elements of the Routh table other than first column are evaluated. The transfer function corresponding to the continued fraction expansion (1) can be written from the first ~no rows of the Routh table (3). A numerical example is included in the next section to illustrate the algorithm.
A continued fraction is given as follows [2]:

\[
T(s) = \frac{1}{1 + \frac{1}{\frac{1}{s} + \frac{1}{\frac{1}{\frac{1}{2} + \frac{1}{\frac{1}{\frac{1}{3} + \frac{1}{\frac{1}{\frac{1}{4} + \frac{1}{\frac{1}{\frac{1}{5}}}}}}}}}}}}.
\]

From (4) and (5)

\[
A_0 = 1, A_1 = 5, A_2 = 1, A_4 = 1^3, A_{12} = 30, A_{32} = 120, \text{ and } X_n = 120.
\]

Now using recursive relationship (6), the remaining elements of the table (3) are evaluated.

For \( j = 6; k = 1,2 \), \( A_6^* = 8; A_{6+2} = 0 \).

For \( j = 7; k = 1,2 \), \( A_7 = 31; AB = 1 \).

For \( j = 8; k = 1,2 \), \( A_8^* = 99; A_{8+2} = 12 \).

From (7)

\[
A_{12} = 0 = A_{10} A_{12} = 1, A_{14} = 1.
\]

By inspection the corresponding transfer function is

\[
T(s) = \frac{120 + 99s + 12s^2}{130 + 129s + 339 + s^2}.
\]

CONCLTIPON

It has been shown that a simple backward recursive relationship exists between the elements of the Routh table and with the help of this relationship all the elements of Routh table can be evaluated if the partial coefficients are known. The simplicity of the algorithm column is unity.

It is believed that this algorithm is simpler than the existing methods.

REFERENCES


