AN ABSTRACT OF THE THESIS OF

VITHOON PONGPARIT for the MASTER OF SCIENCE
(Name) (Degree)

in Electrical and Electronics Engineering presented on June 23, 1970
(Major) (Date)

Title: ON THE RATIONAL APPROXIMATIONS TO THE IDEAL DELAY FUNCTION AND THEIR TIME-DOMAIN OPTIMIZATION

Abstract approved: Redacted for privacy

Song B. Park

In the first part of this thesis, two new classes of rational approximations to the ideal delay function based on the Khovanskii continued fraction expansion of e^x are studied in detail in both the frequency domain and the time domain and comparisons with other delay approximations (by Budak, Allemendou and Storch) are made.

In the second part of this thesis, time-domain optimizations of the delay function are performed, first, by taking the pole locations of the all-pole delay function as variables, and then considering the element values of the lossy LC ladder delay network as variables.

Finally, typical passive and active network realizations of the Khovanskii delay function are considered.

On the Rational Approximations to the Ideal Delay Function and Their Time-Domain Optimization

by

Vithoon Pongparit

A THESIS

submitted to

Oregon State University

in partial fulfillment of the requirements for the degree of

Master of Science

June 1971

APPROVED:

Redacted for privacy

Assistant Professor of Electronics and Electrical Engineering

in charge of major

Redacted for privacy

Head of Department of Electronics and Electrical Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented June 25, 1970

Typed by Barbara Eby for Vithoon Pongparit

ACKNOWLEDGEMENTS

The author wishes to express his sincere gratitude to Professor Song Bai Park who suggested the topic of this thesis and gave much assistance, advice and encouragement during the course of study.

The author also gratefully acknowledges the Oregon State University Computer Center for their provision of the computer time needed in this study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ON THE RATIONAL APPROXIMATIONS TO THE IDEAL DELAY FUNCTION AND THEIR TIME-DOMAIN OPTIMIZATION

I. INTRODUCTION

A delay network is a system in which the output is a replica of the input with a certain amount of time delay. The need for such time-delay networks often arises in applications where two or more signal paths are used to transmit information to a summing point and the relative time position or difference between the two signals is of interest. In order to correct for this time difference or to introduce some time delay in one of the signals, a delay network is inserted in the pertinent path of the signal.

Ideal delay characteristics cannot be obtained from a lumped-constant network, but may be obtained from a transmission line. This is because the ideal delay function e^{-Ts} (s = Laplace transform variable, T = delay time) is a transcendental function of s. For practical reasons, however, a delay network with a large amount of delay time is usually synthesized with lumped-constant elements. Since network functions of lumped-constant networks are rational functions of s, the design of lumped-constant delay networks requires, first of all, a physically realizable rational function which approximates the ideal delay function in the "best" way.

The main purpose of this paper is twofold, first, to make a detailed study of two new classes of rational approximations to an ideal delay function, and second, to optimize rational delay functions to achieve a minimum rise time-to-delay time ratio (hereafter called rise-to-delay ratio) in the step response within certain constraints.

Khovanskii describes, in his book [10], two kinds of continued fraction expansions of the exponential function e^x. As far as the author knows, these expansions have not been studied in the literature in connection with rational approximations to the ideal delay function. In this paper the physical realizability of these expansions by lumped-constant network elements will be examined, and their frequency- and time-domain characteristics will be investigated in detail. Comparisons with other approximating functions will also be made.

Time-domain optimization of delay functions is a relatively recent subject, and several authors have used different criteria on the optimum delay function [8, 19]. In this paper, employing the conventional definitions of the rise and delay times of a step response and starting with the Storch approximating functions for the all-pole functions, we will attempt to achieve a minimum rise-to-delay ratio in the step response within specified tolerances of overshoot and undershoot by adjusting the pole and zero positions of the function in the complex s-plane.

Although a delay network is usually synthesized by lossless elements, a more realistic approach to the pertinent problem would take into consideration the loss of the reactive elements in the realized network. Therefore, in this paper, an attempt will also be made to minimize the rise-to-delay ratio by adjusting the element values of lossy LC ladder delay networks. Fletcher-Powell's method is employed as the minimization technique in both optimizations.

In Chapter II, physical realizability of transfer functions of passive networks is presented. Frequency-domain and time-domain characteristics of the ideal delay function are also given together with some time-domain definitions.

In Chapter III, two classes of new rational approximations of the normalized ideal delay function e^{-s} based on Khovanskii's continued fraction expansion of e^{x} are studied. The physical realizability of these approximating functions (all-pass and non-all-pass functions) are examined, and their frequency- and time-domain characteristics are investigated in detail. Other rational approximations of e^{-s} by Storch, Budak and Allemendou are briefly described, and comparisons between these and Khovanskii approximations are made.

In Chapter IV, optimization of rational delay functions is considered. In the first part, real parts and imaginary parts of zeros and poles are used as variables to optimize the rise-to-delay ratio of a step response under the constraints of 2% and 5% overshoot. In the second part, element values of a lossy LC ladder network are used as variables.

In Chapter V, realization of Khovanskii functions by passive and active networks are illustrated by taking a third order function as an example.

II. APPROXIMATIONS AND SOME TIME-DOMAIN DEFINITIONS

In this chapter, physical realizability of transfer functions of passive networks is presented. Frequency- and time-domain characteristics of the ideal delay functions are also given together with some time-domain definitions.

In general, network synthesis procedure consists of two steps. First, a mathematical function must be determined which is physically realizable and approximates, within specified tolerances, the desired network characteristics; second, a design procedure must follow which starts with the above mathematical function and leads to network configurations together with element values. The first step is called the approximation problem, and the second step is called the realization problem.

Approximations by rational functions are especially important in network theory because driving-point and transfer functions of a linear lumped network are rational functions (ratio of two polynomials) of s, the complex frequency, with real coefficients.

2.1 Physical Realizability

As mentioned above, the approximating function must be physically realizable. The necessary and sufficient conditions for a function of s to be a transfer function of a lumped-constant network will be described in the following.

The transfer function of a lumped-constant network must be, first of all, a ratio of two polynomials of s with real coefficients. The coefficients of the denominator polynomial must be positive and have no missing terms, except for even or odd polynomials, while the coefficients of the numerator polynomial may be negative and/or may have missing terms. The degree of the numerator polynomial must be less than or equal to that of the denominator polynomial. Poles of the transfer function must be in the left half of the s-plane with any multiplicity, but they must be simple on the imaginary axis. On the other hand, zeros of the transfer function may be anywhere in the s-plane. Possible zero and pole locations of the transfer function are shown in Figure 2.1.

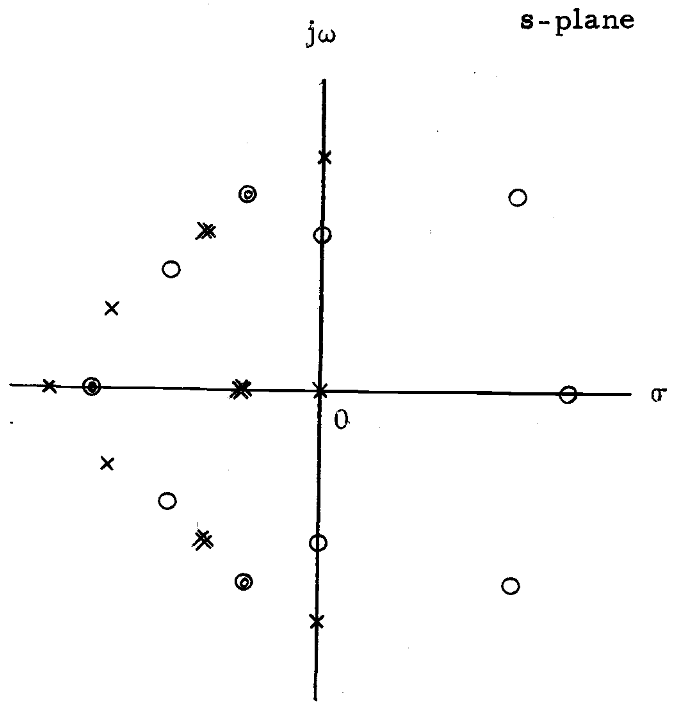


Figure 2.1. Pole and zero locations of a physically realizable network.

Once a transfer function $H(s)$ is chosen, the step response $r(t)$ is easily calculated by

$$r(t) = \mathcal{L}^{-1}\left[\frac{H(s)}{s}\right]$$

2.2 Time-delay Function

Suppose a system function is given by

$$H(s) = ke^{-sT} \qquad (2.1)$$

where k is a positive real constant. Then the frequency response of the system can be expressed as

$$H(j\omega) = ke^{-j\omega T} \qquad (2.2)$$

so that the amplitude response $A(\omega)$ is a constant k, and the phase response

$$\theta(\omega) = -\omega T \qquad (2.3)$$

is linear in ω. The response of such a system to an excitation denoted by the Laplace transform pair $[e(t), E(s)]$ is

$$R(s) = kE(s)e^{-sT} \qquad (2.4)$$

so that the inverse Laplace transform $r(t)$ can be written as

$$r(t) = ke(t-T) \qquad (2.5)$$

We see that the response $r(t)$ is simply the excitation delayed by a time T and multiplied by a constant. Thus no signal distortion results from transmission through the system described by Equation (2.1). We note further that the delay time T can be obtained by differentiating the phase response in Equation (2.3) with respect to ω; that is

$$\text{delay} = -\frac{d}{d\omega}\theta(\omega) = T \tag{2.6}$$

The amplitude, phase and delay characteristics of Equation (2.1) are given in Figure 2.2 (a), (b) and (c) respectively.

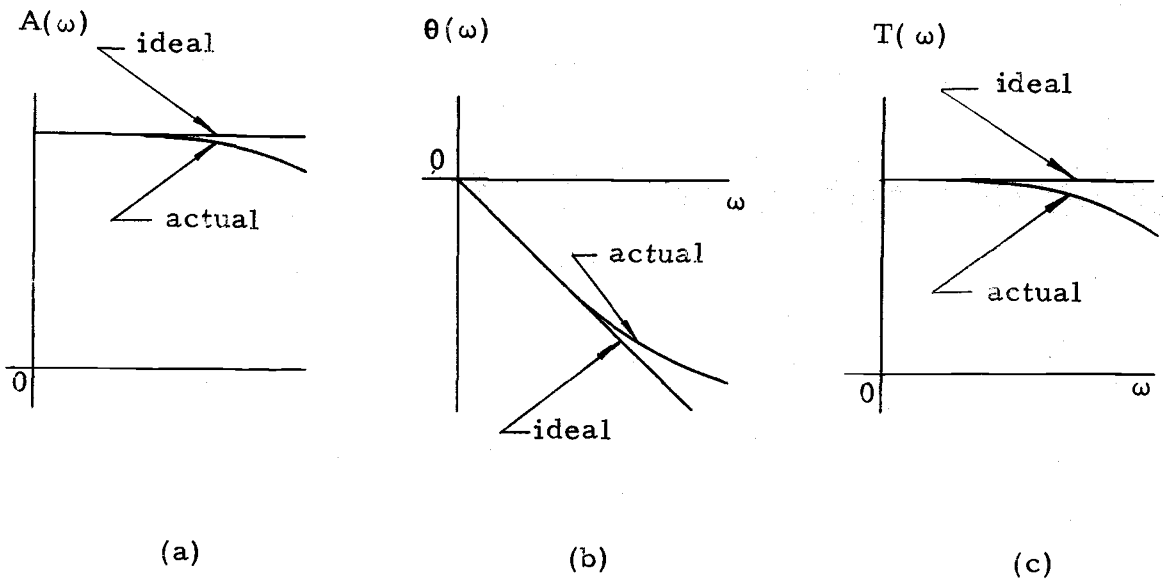


Figure 2.2. Amplitude, phase and delay of ideal and actual delay functions (a) Amplitude, (b) Phase (c) Delay.

A system with linear phase and constant amplitude is obviously desirable from a pulse transmission viewpoint. However, the system function H(s) in Equation (2.1) is realizable only by a lossless transmission line called a delay line. If we require that the delay network be made up of lumped-constant elements, then we must approximate Equation (2.1) by a rational function of s. This function must satisfy the constraints of a physically realizable network given in Section 2.1. Amplitude, phase and delay characteristics of actual delay networks are also given in Figure 2.2.

2.3 Some Time-domain Definitions

In time-domain synthesis, we are interested in determining a network whose time response to a given input is specified. Several definitions employed in this paper to describe a step response are given below:

2.3.1 Rise time T_r. The rise time of a step response is defined as the time required for the step response to rise from 10% to 90% of its final value.

2.3.2 Delay time T_d. This is a measure of the time lapse between the time a unit step input is applied and the time the step response reaches 50% of its final value.

2.3.3 Rise-to-delay ratio. This is the ratio between the two quantities described in Sections 2.3.1 and 2.3.2. This quantity is independent of any scaling in the time or frequency domain, and is usually the most significant quantity in measuring the quality of a delay network.

2.3.4 Overshoot. The overshoot of a step response is defined as the difference between the peak value above the final value and the final value of the step response, expressed as a percentage of the final value.

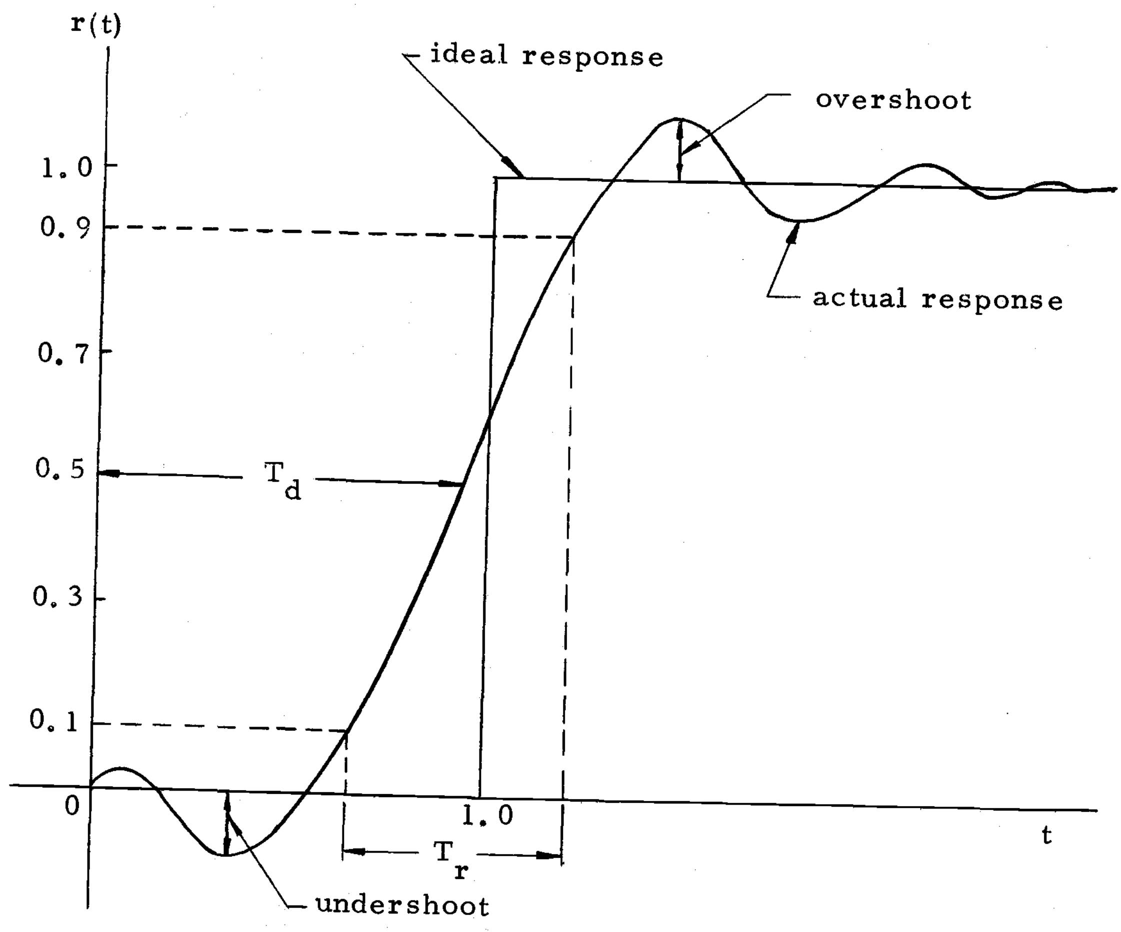


Figure 2.3. Time-delay characteristics of ideal and actual delay responses.

2.3.5 Undershoot. The undershoot of a step response is defined as the difference between the peak value below the initial value and the initial value of the step response, expressed as a percentage of its final value.

The definitions given above are illustrated in Figure 2.3. For an ideal delay network, the rise time is equal to zero (therefore its rise-to-delay ratio is equal to zero), without any overshoot or undershoot. However, actual delay networks have nonzero rise-to-delay ratios, with some overshoot and undershoot. Then within some specified tolerances of overshoot and undershoot, the qualities of these networks can be compared by their rise-to-delay ratios.

All calculations of frequency and time response were carried out by the digital computer CDC 3300 at OSU Computer Center. Brief descriptions of some main subroutines are given in Appendices A and B.

III. RATIONAL APPROXIMATIONS TO e^{-s}

Khovanskii, in his book [10], gives a convergent continued fraction expansion of e^{x}. Two series of rational functions result from this expansion truncated at a finite term. These functions have not been studied in the literature in connection with the rational approximations to the ideal delay function.

In this chapter, we will examine the physical realizability and other properties of these two new classes of rational approximations in both the frequency domain and the time domain. Comparisons with other rational approximations will also be made.

3.1 Khovanskii's Continued Fraction Expansion of e^{x}

The continued fraction expansion of e^{x} is given by Khovanskii [10] in the form

$$e^{x} \simeq \frac{1}{1\ -}\,\frac{x}{1\ +}\,\frac{x}{2\ -}\,\frac{x}{3\ +}\,\frac{x}{2\ -}\,\frac{x}{5\ +}\ \cdots\ \frac{x}{+\ 2\ -}\,\frac{x}{2n-1\ +}\ \cdots \tag{3.1}$$

which is convergent throughout the whole of the finite complex x-plane. With x replaced by $-s$, the normalized delay function is obtained:

$$e^{-s} \simeq \frac{1}{1\ +}\,\frac{s}{1\ -}\,\frac{s}{2\ +}\,\frac{s}{3\ -}\,\frac{s}{2\ +}\,\frac{s}{5\ -}\ \cdots\ \frac{s}{-\ 2\ +}\,\frac{s}{2n-1-}\ \cdots \tag{3.2}$$

Truncating Equation (3.2) at the i^{th} term, we get an i^{th} convergent

$$e^{-s} \simeq \frac{1}{1\ +}\,\frac{s}{1\ -}\,\frac{s}{2\ +}\,\frac{s}{3\ -}\,\frac{s}{2\ +}\,\frac{s}{5\ -}\ \cdots\ \frac{2}{+i-2\ -}\,\frac{s}{2} \quad \text{for } i = \text{odd} \tag{3.3}$$

$$e^{-s} \simeq \frac{1}{1+}\,\frac{s}{1-}\,\frac{s}{2+}\,\frac{s}{3-}\,\frac{s}{2+}\,\frac{s}{5-}\cdots\frac{s}{-2+}\,\frac{s}{i-1} \quad \text{for } i = \text{even} \tag{3.4}$$

which approximates e^{-s} in the vicinity of $s = 0$. The convergent of Equation (3.3) can be rewritten as

$$\begin{aligned} e^{-s} &\simeq \frac{P_1}{Q_3} = \frac{1}{1} \\ &\simeq \frac{P_3}{Q_3} = \frac{2-s}{2+s} \\ &\simeq \frac{P_5}{Q_5} = \frac{12-6s+s^2}{12+6s+s^2} \\ &\;\;\vdots \\ &\simeq \frac{P_i}{Q_i} \end{aligned} \tag{3.5}$$

where

$$Q_i(s) = s^k + \sum_{m=2}^{k+1} \frac{\left[\prod_{j=2}^{m} (k-j+2)(k+j-1)\right] s^{k-m+1}}{(m-1)!} \tag{3.6}$$

$$k = \frac{i-1}{2}, \quad i = \text{odd}$$

$$P_i(s) = Q_i(-s) \tag{3.7}$$

The convergent of Equation (3.4) can be rewritten as

$$e^{-s} \simeq \frac{P_2}{Q_2} = \frac{1}{1+s}$$

$$e^{-s} \simeq \frac{P_4}{Q_4} = \frac{6-2s}{6+4s+s^2}$$

$$\simeq \frac{P_6}{Q_6} = \frac{60-24s+3s^2}{60+36s+9s^2+s^3} \tag{3.8}$$

$$\vdots$$

$$\simeq \frac{P_i}{Q_i}$$

where

$$P_i(s) = 1 + \sum_{m=1}^{k} \frac{\prod_{j=1}^{m} [k-j+1](-s)^m}{\prod_{j=1}^{m} [2k-j+2]\, m!} \tag{3.9}$$

$$k = \frac{i-2}{2}, \quad i = \text{even}$$

$$Q_i(s) = 1 + \sum_{m=1}^{k+1} \frac{\prod_{j=1}^{m} [k-j+2]\, s^m}{\prod_{j=1}^{m} [(k-j+2)]\, m!} \tag{3.10}$$

The two series of rational approximations to e^{-s} given in Equations (3.5) and (3.8) are physically realizable as transfer functions of lumped-constant networks. We will prove this by showing that the denominators of these functions are Hurwitz, or they have left-half-plane poles only.

3.2 Physical Realizability of Kohvanskii Rational Approximations to e^{-s}

Let $Q_i(s)$ be the denominator polynomial of the Khovanskii function truncated at the i^{th} term, and let us split it into two parts:

$$Q_i(s) = E_i(s) + O_i(s) \tag{3.11}$$

where $E_i(s)$ and $O_i(s)$ are the even and odd parts. The necessary and sufficient conditon that $Q_i(s)$ be a Hurwitz polynomial is simply that all coefficients of s or $\frac{1}{s}$ in the continued fraction expansion of $\frac{E_i(s)}{O_i(s)}$ be positive. As an example, consider the Khovanskii function truncated at the 9th term:

$$Q_9(s) = 1680 + 840s + 180s^2 + 20s^3 + s^4$$

$$\frac{E_9(s)}{O_9(s)} = \frac{1}{20}s + \cfrac{1}{\frac{10}{69}s + \cfrac{1}{\frac{1587}{8680}s + \cfrac{1}{\frac{343}{966}s}}}$$

Since all coefficients are positive, $Q_9(s)$ is a Hurwitz polynomial. In the same manner we can prove that all denominators of the Khovanskii functions are Hurwitz. Hence they are all physically realizable transfer functions.

Incidentally, all zeros of the Khovanskii rational approximations are located in the right half of the complex s-plane.

3.3 Frequency- and Time-domain Characteristics of Khovanskii Rational Approximations

The series of functions expressed in Equation (3.5) are all-pass functions, while the series of functions expressed in Equation (3.8) are not. In this section, we will study the frequency- and time-domain characteristics of these functions in detail.

3.3.1 All-pass Functions

The coefficients of the denominators of this series of functions are given in Table 3.1. All coefficients are integer, but some of the coefficients greater than 8 digits are given in the E-field. The coefficients of the numerators are the same as those of the denominators, except the signs in the odd terms. The poles of these functions are given in Table 3.2 up to the 6th decimal point.

In this series of functions, zeros and poles are symmetrical with respect to the imaginary axis of the complex s-plane. The magnitude of this kind of function is unity for all frequencies and its phase angle is twice that of $P_n(j\omega)$. The delay characteristics of all-pass functions are shown in Figure 3.1; it is seen that the delay characteristics improve as n increases. The step response is given in Figure 3.2. In this case, since the degree of the numerator and the denominator are the same, the step response does not start at the origin; instead, it starts at +1 or -1. Hence this kind of function is not suitable for

Table 3. 1. Denominator coefficients of Khovanskii all-pass coefficients.

n	Denominator Coefficients Starting with Constant Term								
2	12	6	1						
3	120	60	12	1					
4	1680	840	180	20	1				
5	30240	15120	3360	420	30	1			
6	665280	332640	75600	10080	840	42	1		
7	17297280	8648640	1995840	277200	25200	1512	56	1	
8	5. 1891840E08	2. 5945920E08	60560480	8648640	831600	55440	2520	72	1

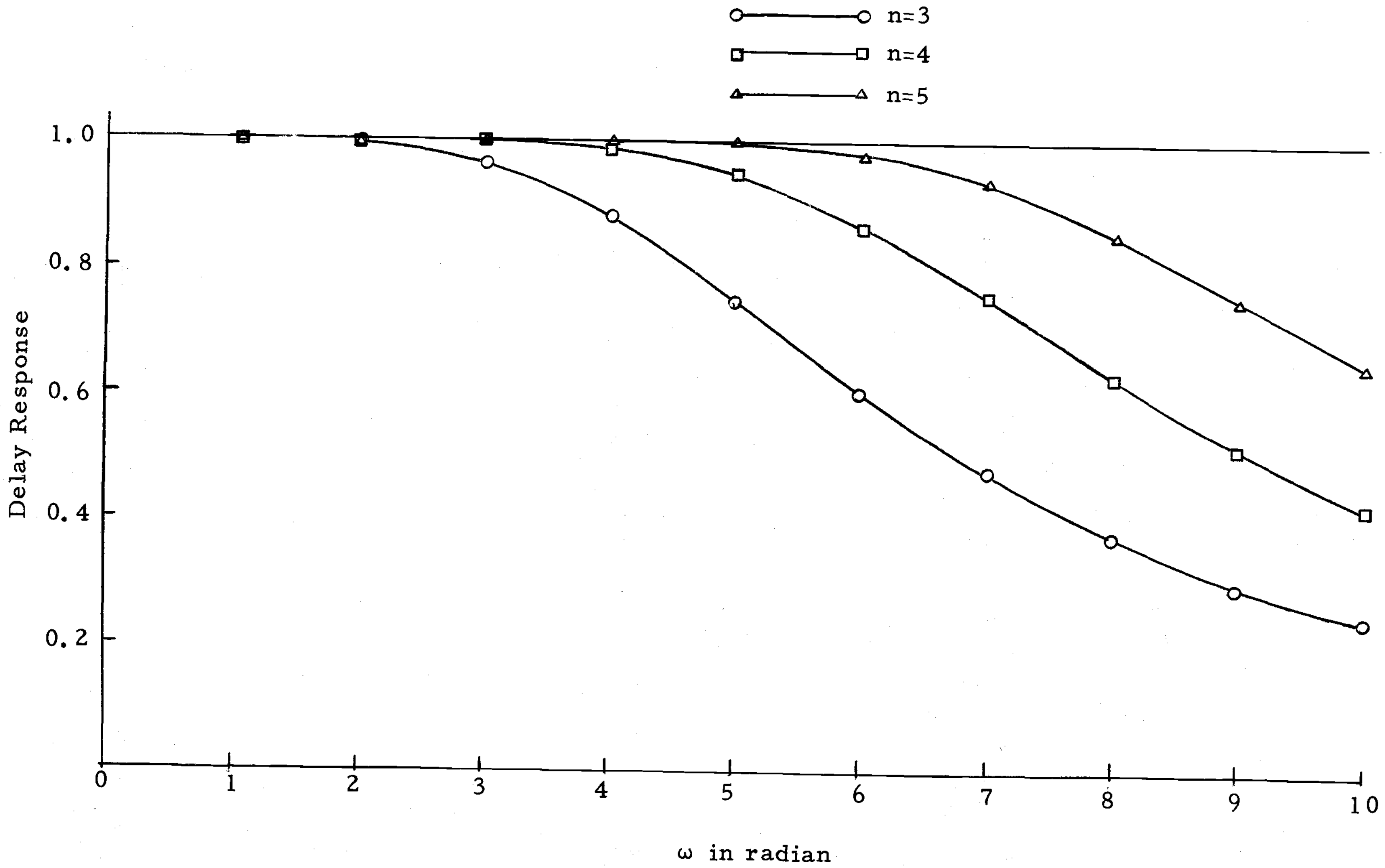


Figure 3.1. The delay characteristics of Khovanskii all-pass functions.

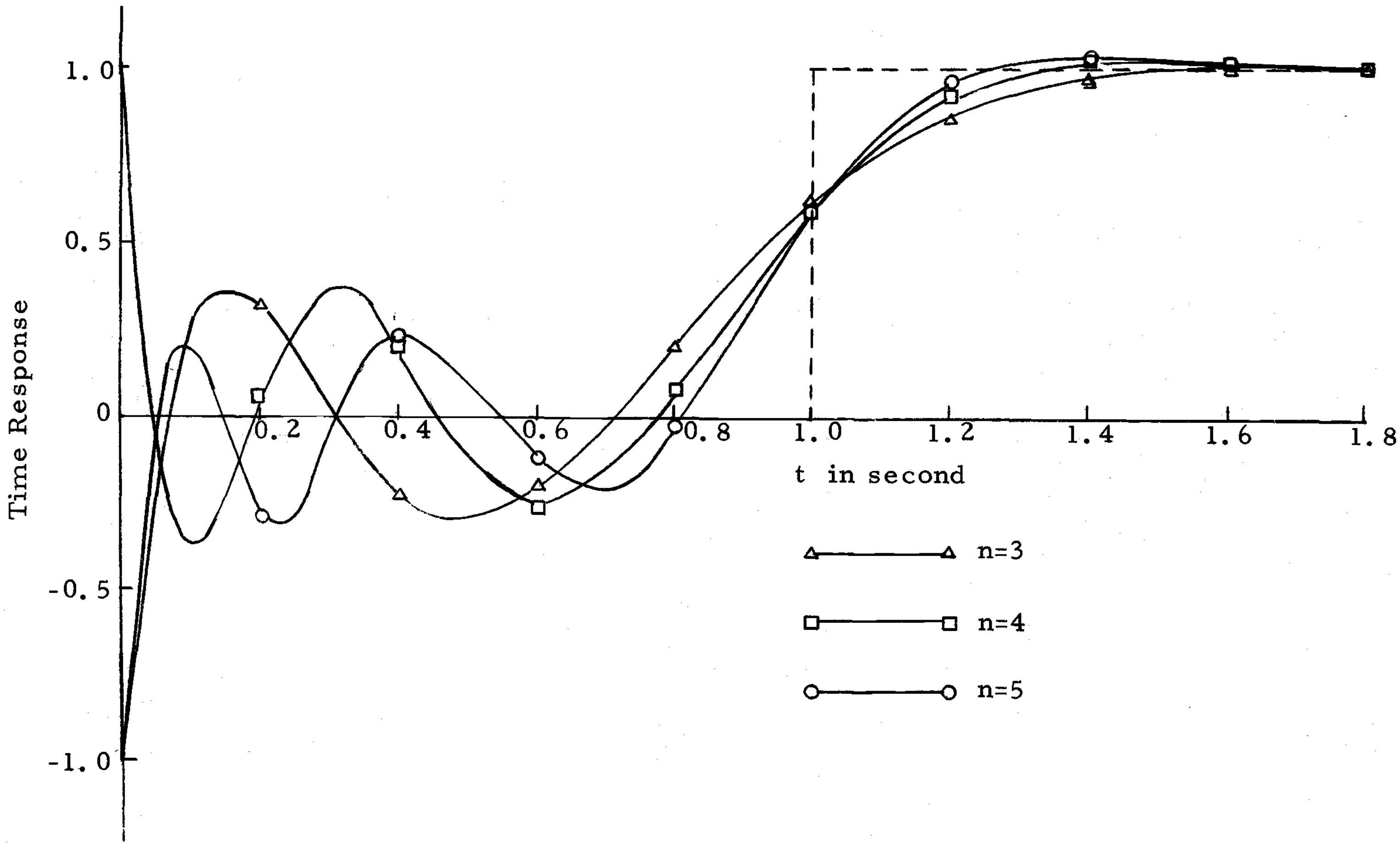


Figure 3. 2. The step responses of Khovanskii all-pass functions.

Table 3.2. Pole locations of Khovanskii all-pass functions.

n	Pole Location -x	Pole Location ±jy	n	Pole Location -x	Pole Location ±jy
1	2.000000		6	8.496719	1.735019
				7.471417	5.252545
2	3.000000	1.732051		5.031864	8.985346
3	4.644371		7	9.943574	
	3.677815	3.508762		9.516581	3.478572
				8.140278	7.034348
4	5.792421	1.734468		5.371354	10.841388
	4.207579	5.314836			
			8	11.850754	3.555235
5	7.293477			9.739363	3.892317
	6.703913	3.485323		8.740017	9.149211
	4.649349	7.142046		5.669866	12.662265

pulse transmission, although the amplitude characteristics are ideal. The correlation between the frequency response and the step response is not apparent for this sort of function.

3.3.2 Non-all-pass Functions

The functions of this series are even convergents of the Khovanskii function. Their numerator and denominator coefficients are calculated up to degree 12 by Equations (3.9) and (3.10), and are given in Tables 3.3 and 3.4. The zeros and poles of this series of functions up to degree 12 are given in Table 3.5. This class of functions has either one negative real pole or one positive real zero; all other zeros and poles are complex.

Table 3-3. Numerator coefficients of Khovanskii non-all-pass functions.

n	Numerator Coefficients Staring with Constant Term
2	6 -2
3	60 -24 3
4	840 -360 60 -4
5	15120 -6720 1260 -120 5
6	332640 -151200 30240 -3360 210 -6
7	8648640 -3991680 831600 -100800 7560 -336 7
8	2.5945920E08 -1.2108096E08 25945920 -3326400 277200 -15120 504 -8
9	8.8216128E09 -4.1513472E09 9.0810720E08 -1.2108096E08 10810800 -665280 27720 -720 9
10	3.3522128E11 -1.5878903E11 3.5286451E10 -4.8432384E09 4.5405360E08 -30270240 1441440 -47520 990 -10
11	1.4079294E13 -6.7044257E12 1.5084957E12 -2.1171870E11 2.0583763E10 -1.4529715E09 75675600 -2882880 77220 -1320 11
12	6.4764752E14 -3.0974446E14 7.0396470E13 -1.0056638E13 1.0056638E12 -7.4101547E10 4.1167526E09 -1.7297280E08 5405400 -120120 1716 -12

Table 3-4. Denominator coefficients of Khovanskii non-all-pass functions.

n	Denominator Coefficients Starting with Constant Terms
2	6 4 1
3	60 36 9 1
4	840 480 120 16 1
5	15120 8400 2100 300 25 1
6	332640 181440 45360 6720 630 36 1
7	8648640 4656960 1164240 176400 17640 1176 49 1
8	2.5945920E08 1.3837824E08 34594560 5322400 554400 40320 2016 64 1
9	8.8216128E09 4.6702656E09 1.1675664E09 1.8162144E08 19459440 1496880 83160 3240 81 1
10	3.3522128E11 1.7643225E11 4.4108064E10 6.9189120E09 7.5675600E08 60540480 3603600 158400 4950 100 1
11	1.4079294E13 7.3748683E12 1.8437170E12 2.9111322E11 3.2345913E10 2.6637811E09 1.6648632E08 7927920 283140 7260 121 1
12	6.4764752E14 3.3790305E14 8.4475764E13 1.3408851E13 1.5084957E12 1.2703122E11 8.2335052E09 4.1513472E08 16216200 480480 10296 144 1

Table 3-5. Zero and pole locations of Khovanskii non-all-pass function.

	Zero Locations		Pole Locations	
n	+x	±jy	+x	±jy
1			-1.000000	-
2	3.000000	-	-2.000000	±1.414214
3	4.000000	±2.000000	-3.637834	-
			-2.681083	±3.050430
4	5.648486	-	-4.787193	±1.567476
	4.675757	±3.913490	-3.212807	±4.773087
5	6.796057	±1.886649	-6.286705	-
	5.203941	±5.805857	-5.700953	±3.210266
			-3.655694	±6.543737
6	8.298523	-	-7.490638	±1.621502
	7.706097	±3.740053	-6.470515	±4.900121
	5.644642	±7.693546	-4.038848	±8.345600
7	9.501455	±1.841500	-8.936833	-
	8.472096	±5.582570	-8.511835	±3.281014
	6.026449	±9.582063	-7.141055	±6.623046
			-4.378694	±10.169693
8	10.949006	-	-10.169446	±1.649202
	10.820395	±3.665410	-9.406371	±4.969217
	9.139631	±7.422677	-7.738688	±8.370879
	6.365471	±11.473438	-4.685495	±12.010579
9	12.180999	±1.817155	-11.587351	-
	11.412402	±5.482757	-11.253270	±3.321341
	9.734818	±9.264046	-10.206883	±6.680141
	6.671780	±13.368387	-8.280042	±10.138360
			-4.966129	±13.864686
10	13.599697	-	-12.837676	±1.666062
	13.263326	±3.623432	-12.226132	±5.012721
	12.210485	±7.298670	-10.934303	±8.409673
	10.274200	±11.108354	-8.776435	±11.921854
	6.952141	±15.267079	-5.225453	±15.729529
11	14.849583	±1.802069	-14.238485	-
	14.234390	±5.425223	-13.962282	±3.347519
	12.935682	±9.115793	-13.112562	±6.720463
	10.769007	±12.956338	-11.602918	±10.154856
	7.211335	±17.169447	-9.235964	±13.718720
			-5.467034	±17.603299
12	16.250690	-	-15.499798	±1.679273
	15.973326	±3.596240	-14.990690	±5.041570
	15.118827	±7.226502	-13.927779	±8.442864
	13.602377	±10.935240	-12.223616	±11.913327
	11.227248	±14.808344	-9.664521	±15.526981
	7.452885	±19.075318	-5.693584	±19.484631

Table 3.6. Time-response characteristics of Khovanskii non-all-pass functions.

n	10% Time Response	90% Time Response	10-90% Rise Time	50% Delay Time	Rise-to-Delay Ratio	Overshoot %	Undershoot %
2	0.548	1.483	0.935	0.891	1.049	1.3	-17.9
3	0.709	1.293	0.584	0.945	0.618	2.4	-17.6
4	0.788	1.209	0.422	0.966	0.437	3.1	-16.9
5	0.834	1.163	0.329	0.976	0.337	3.6	-16.1
6	0.863	1.133	0.270	0.982	0.275	4.0	-15.5
7	0.884	1.112	0.228	0.986	0.208	4.3	-15.0
8	0.900	1.097	0.197	0.989	0.199	4.6	-14.6
9	0.912	1.086	0.174	0.991	0.175	4.9	-14.2
10	0.921	1.076	0.155	0.992	0.156	5.1	-13.9
11	0.929	1.069	0.140	0.993	0.141	5.2	-13.6
12	0.935	1.063	0.128	0.994	0.129	5.4	-13.4

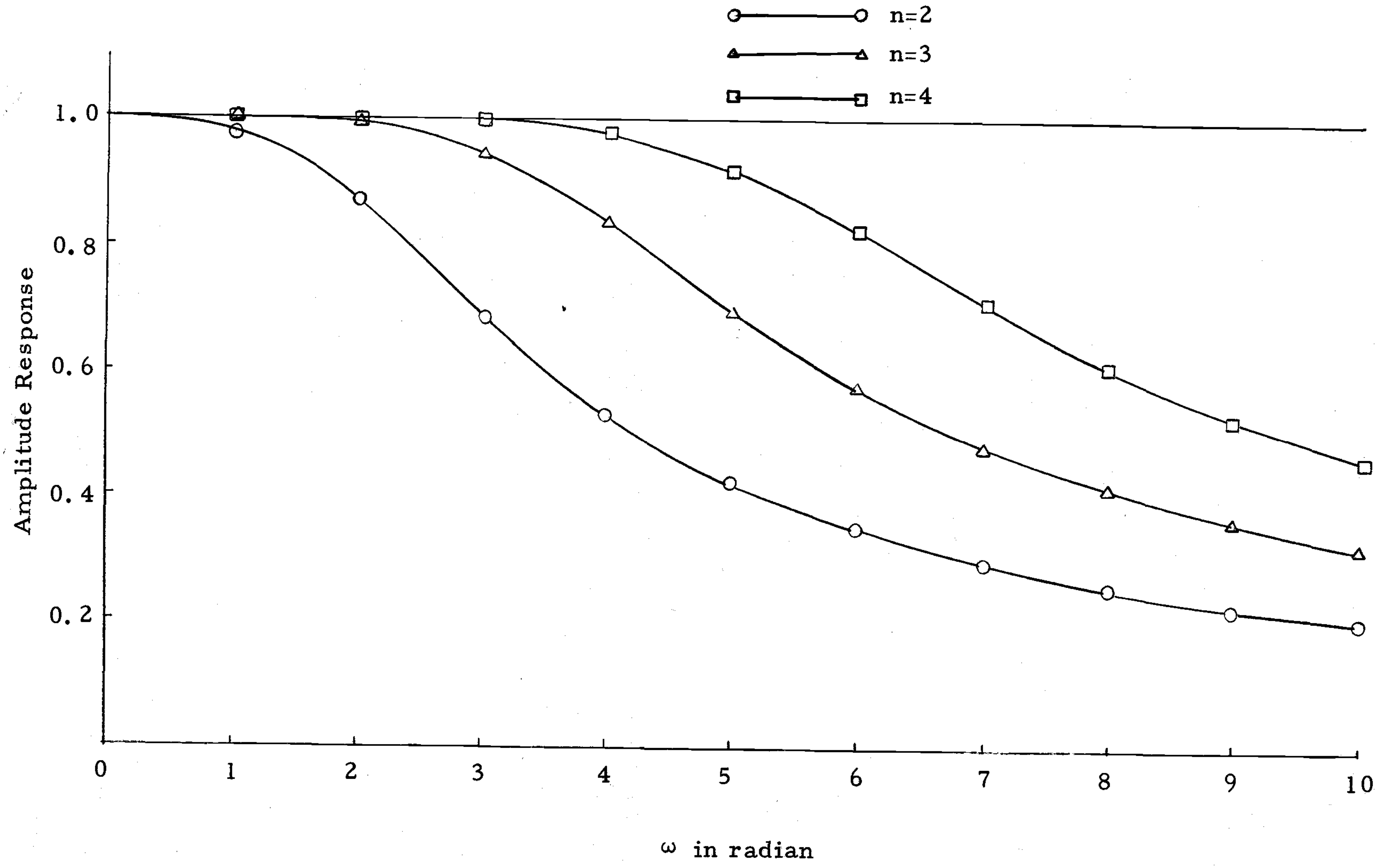


Figure 3.3. The amplitude characteristics of Kovanskii non-all-pass functions.

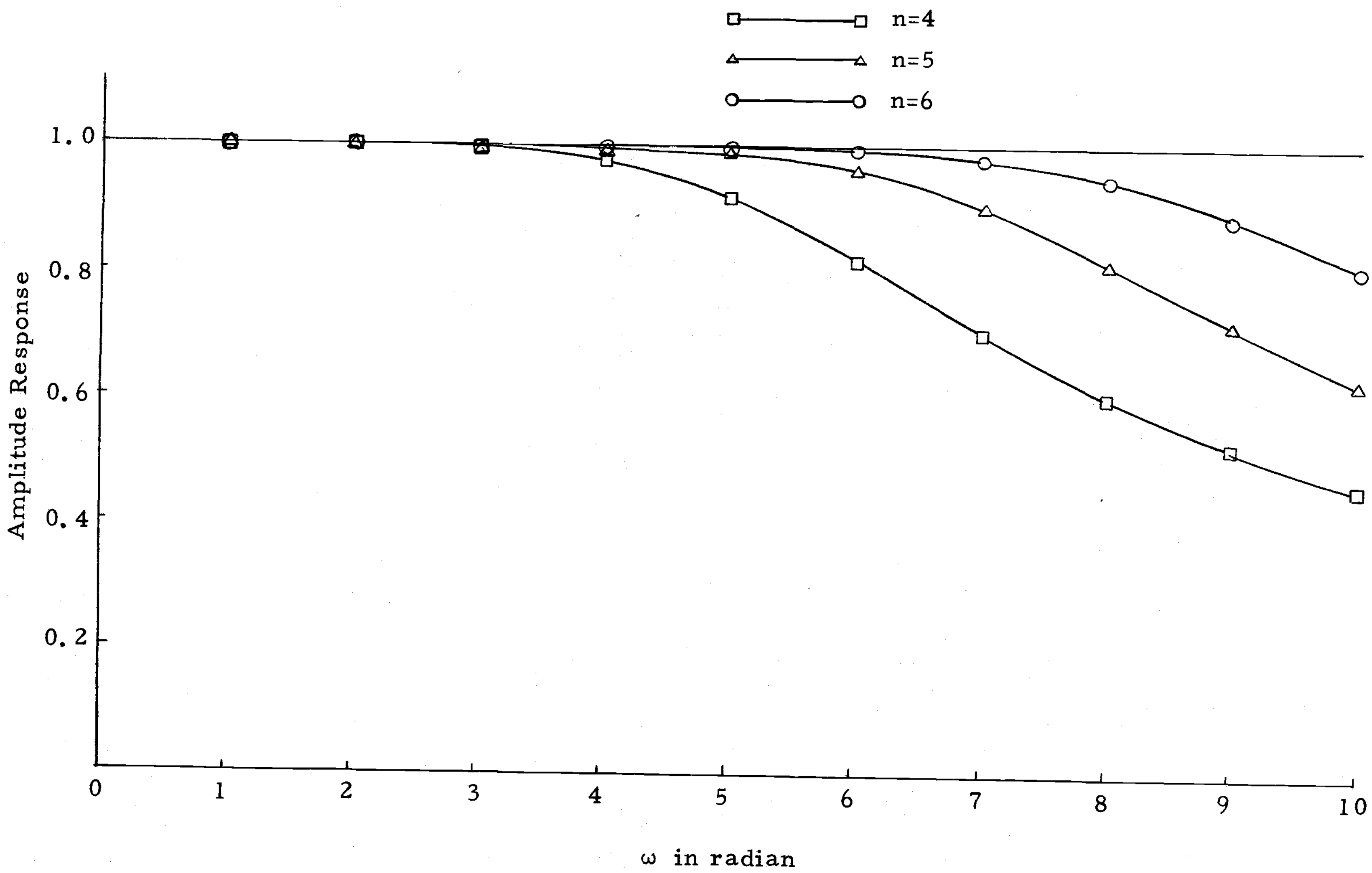


Figure 3.4. The amplitude characteristics of Khovanskii non-all-pass functions.

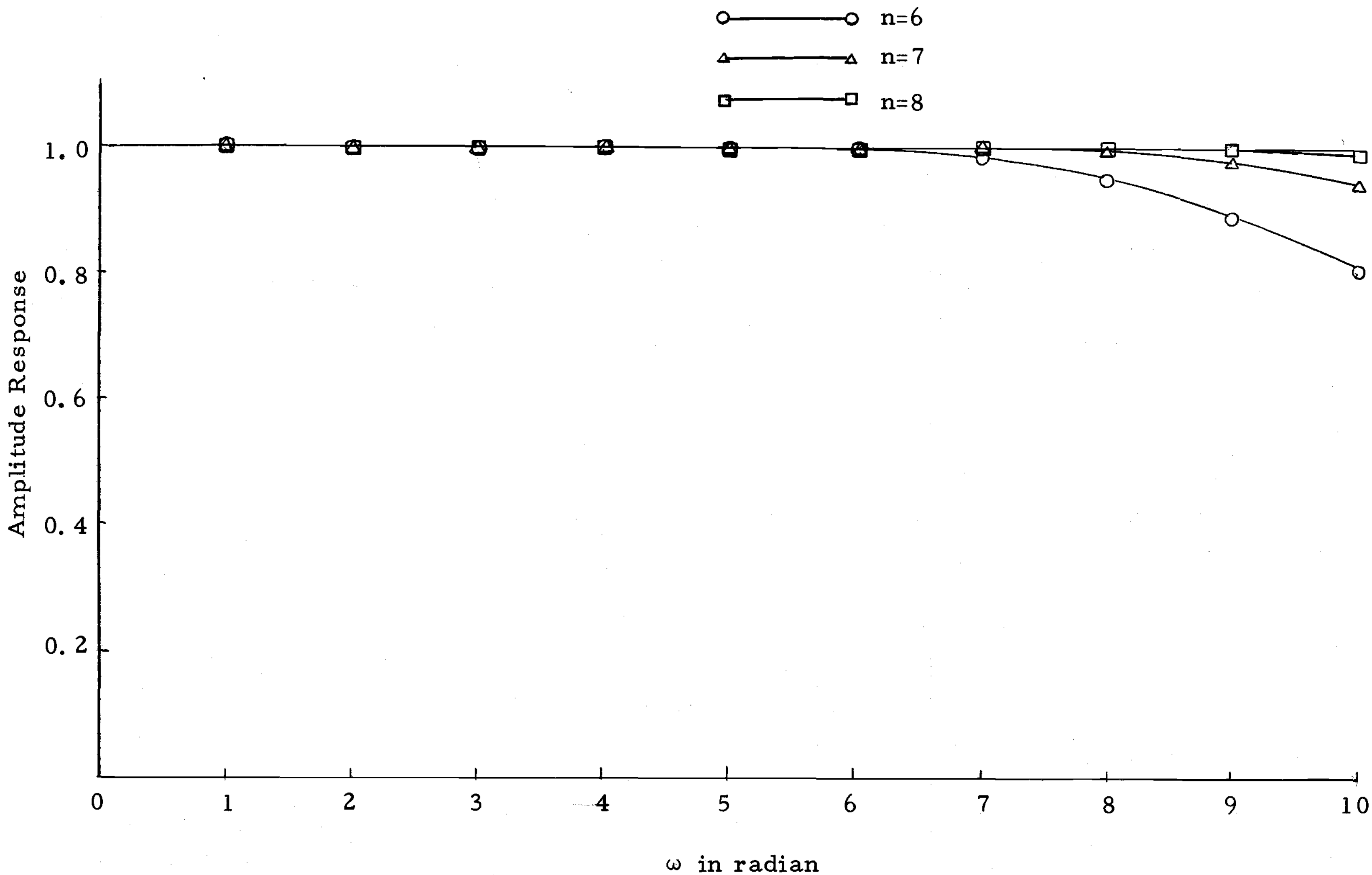


Figure 3.5. The amplitude characteristics of Khovanskii non-all-pass functions.

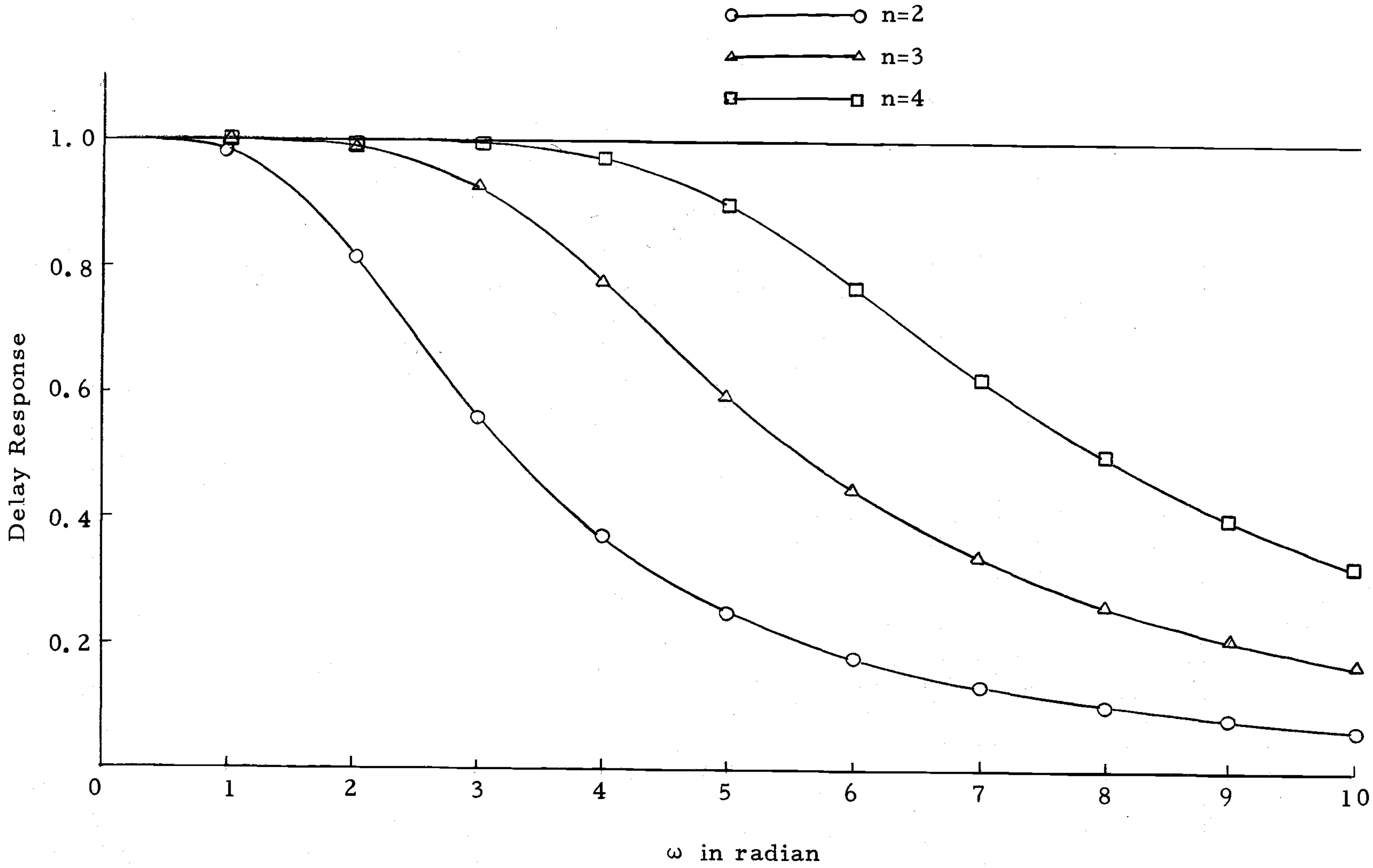


Figure 3.6. The delay characteristics of Khovanskii non-all-pass functions.

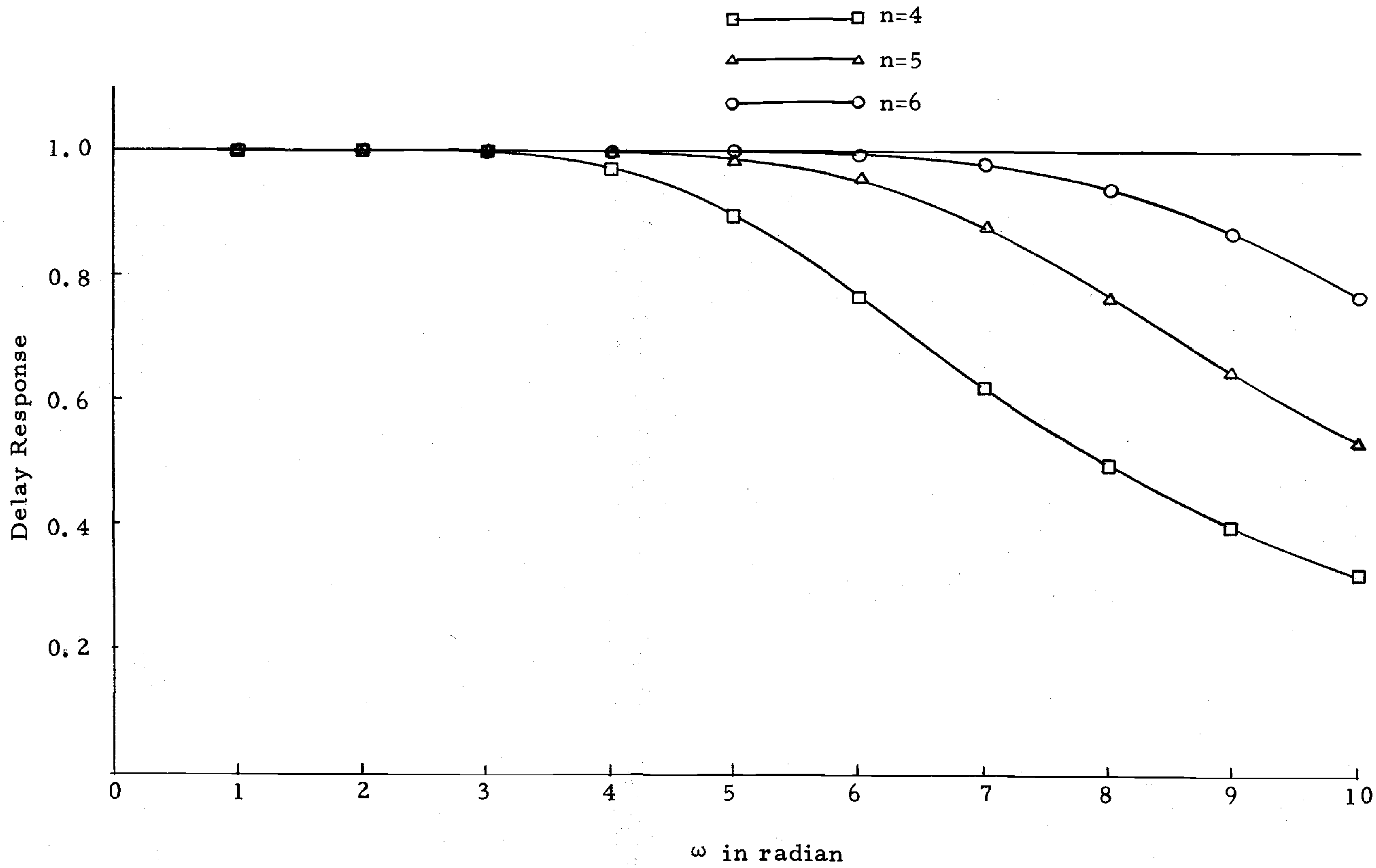


Figure 3. 7. The delay characteristics of Khovanskii non-all-pass functions.

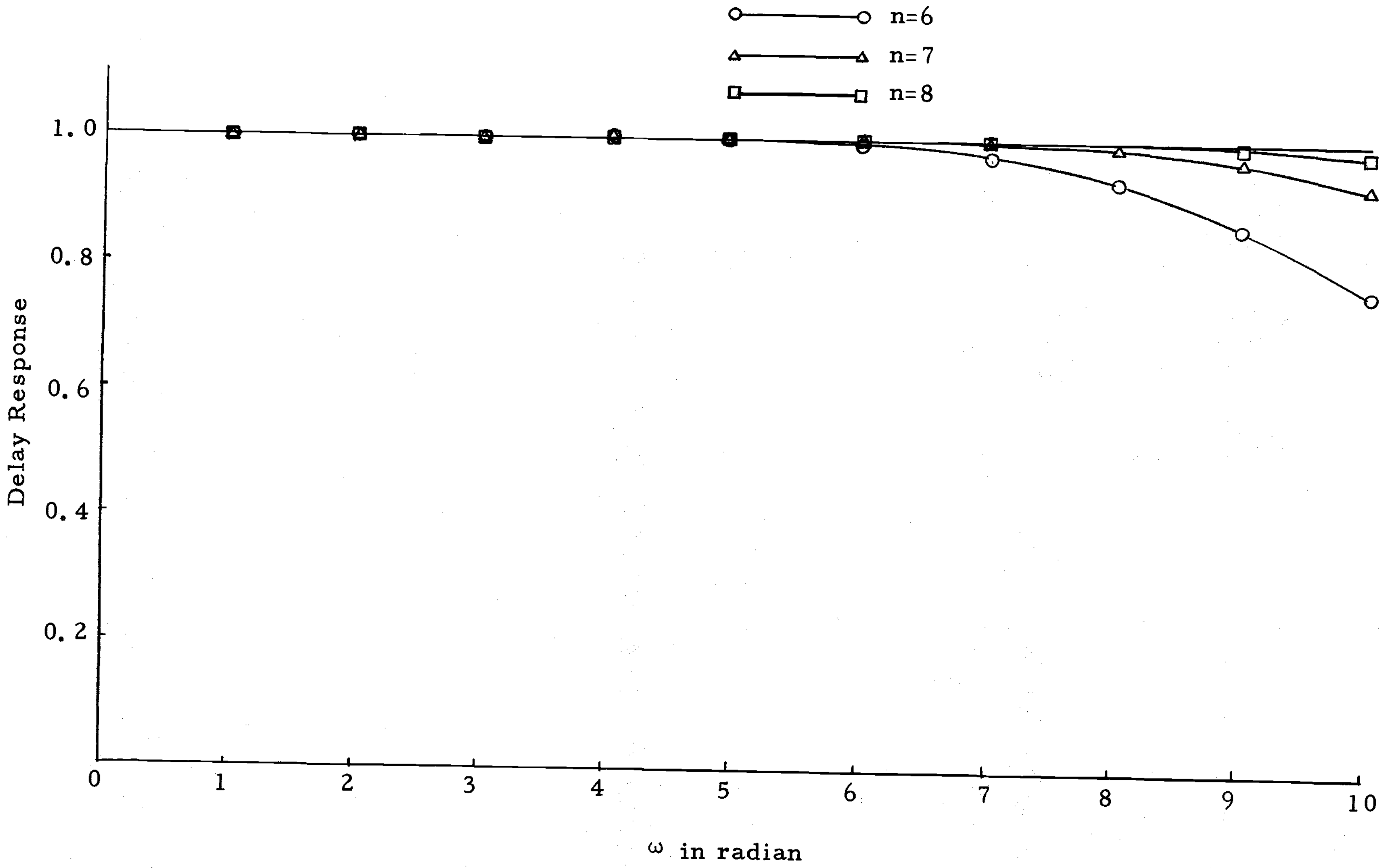


Figure 3. 8. The delay characteristics of Khovanskii non-all-pass functions.

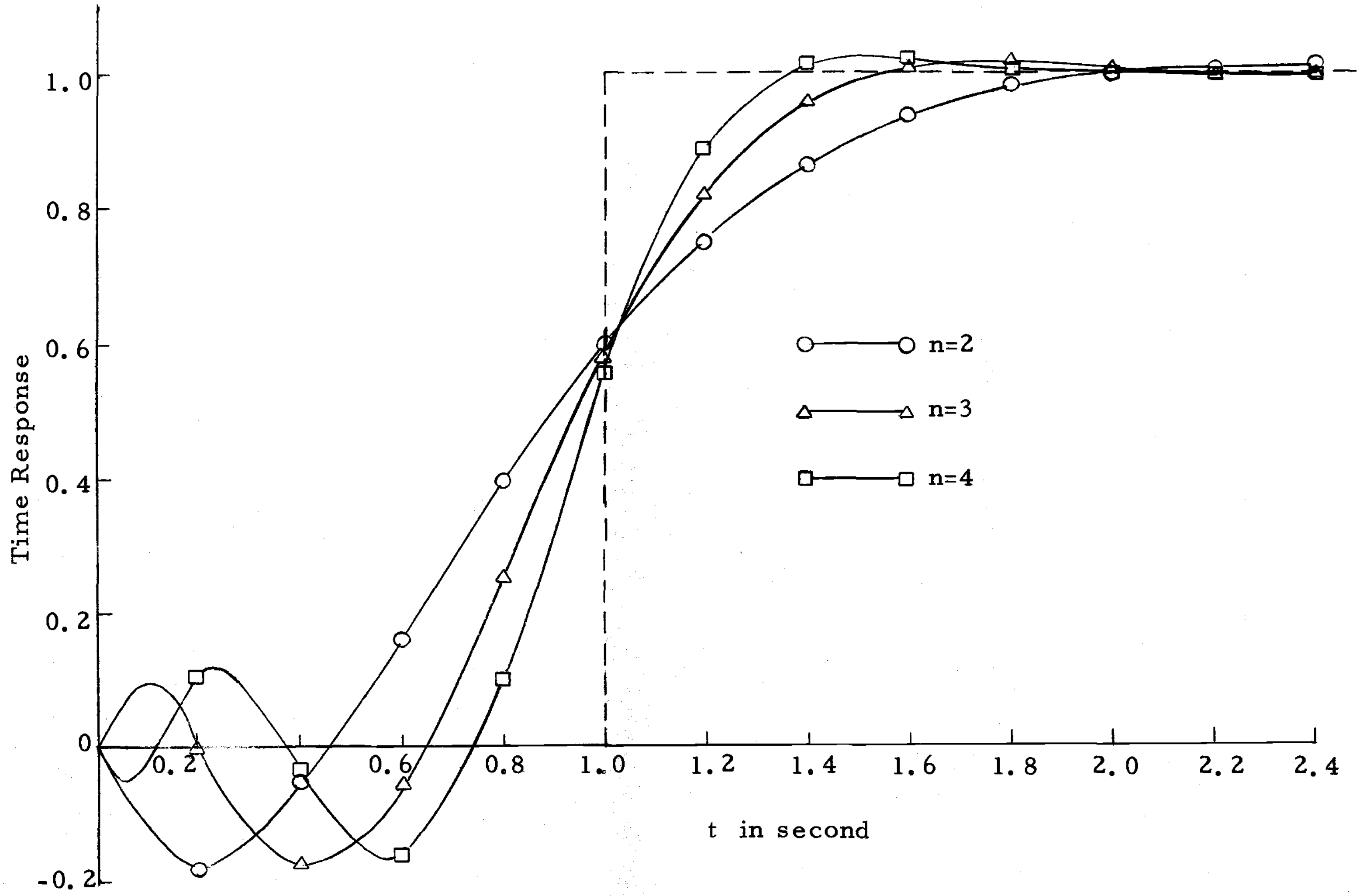


Figure 3.9. The step responses of Khovanskii non-all-pass functions.

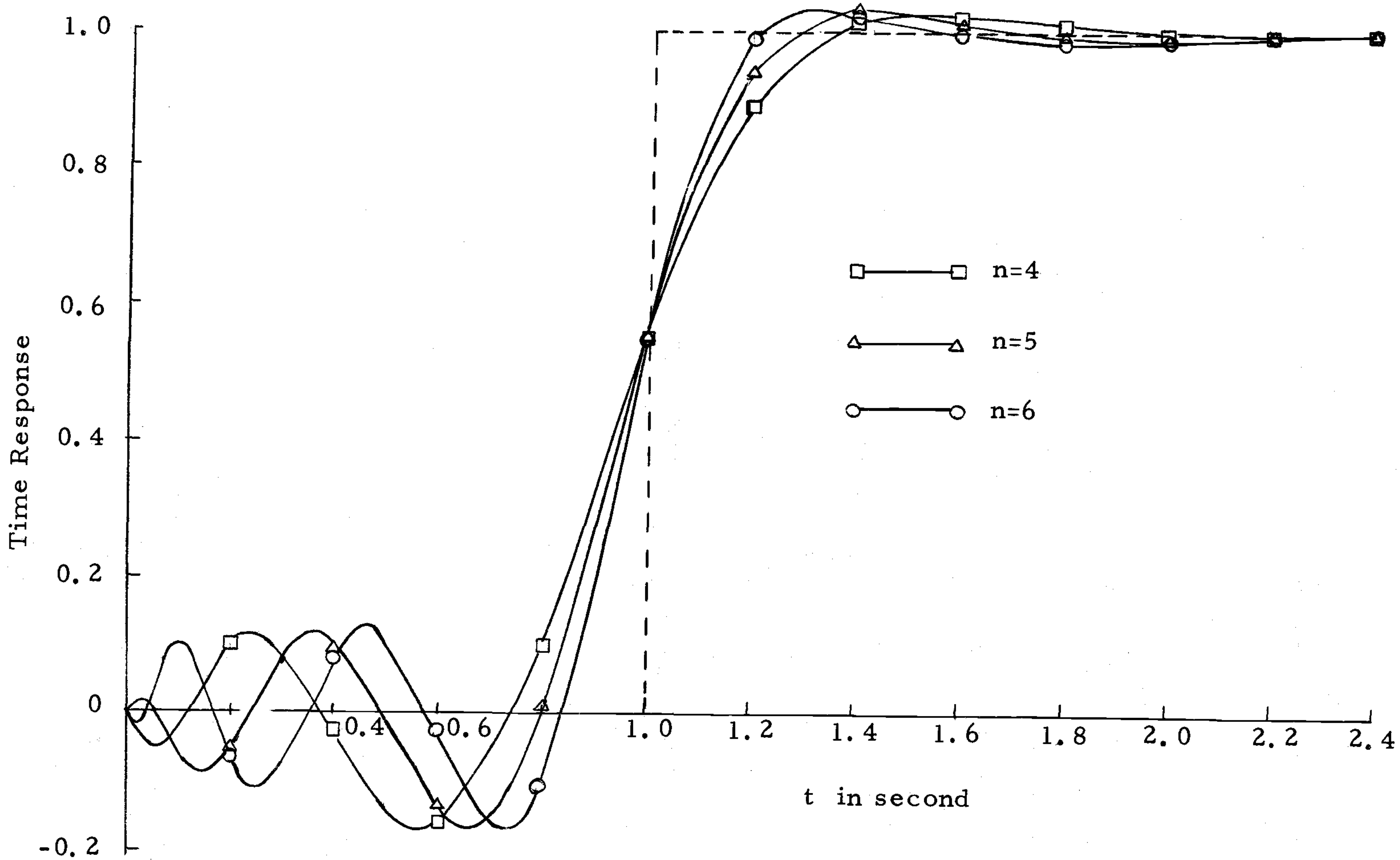


Figure 3.10. The step responses of Khovanskii non-all-pass functions.

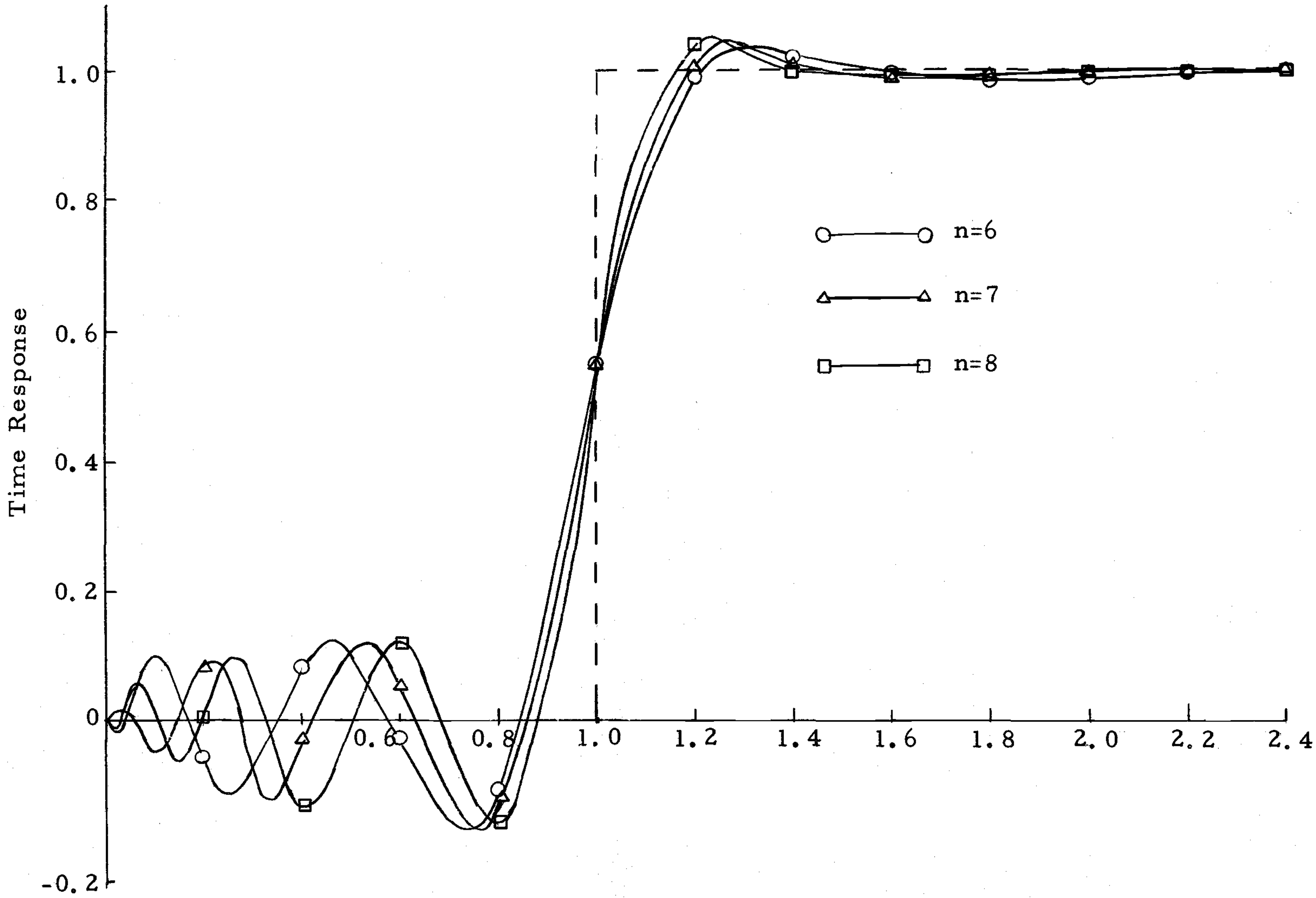


Figure 3.11. The step responses of Khovanskii non-all-pass functions.

The amplitude and delay characteristics of this series of functions from $n = 2$ to $n = 8$ are given in Figure 3.3 to Figure 3.5 and in Figure 3.6 to Figure 3.8, respectively. From these curves, it is seen that both characteristics improve as the order of the function increases.

The step responses are shown in Figure 3.9 to Figure 3.11. They show overshoot and undershoot; they oscillate one cycle after rising and $(n-1)$ half cycles before rising (n=the degree of the function). As the degree of the function increases the undershoot is slightly decreased while the overshoot is slightly increased. The rise-to-delay ratio improves as the degree of the function increases. The 10% to 90% rise time, 50% delay time, rise-to-delay ratio, undershoot, and overshoot up to degree 12 are given in Table 3.6.

Other delay functions are briefly reviewed in the following.

3.4 Rational Approximation of e^{-s} by Storch

Storch's method [14] of approximating the ideal delay function by a rational function starts with rewriting e^{-s} as

$$T(s) = e^{-s} = \frac{1}{\cosh(s) + \sinh(s)} = \frac{1/\sinh(s)}{\coth(s)+1} \tag{3.12}$$

Expanding coth(s) into a continued fraction, we get

$$\coth(s) = \frac{1}{s} + \cfrac{1}{\frac{3}{s} + \cfrac{1}{\frac{5}{s} + \cfrac{1}{\frac{7}{s} + \cfrac{1}{\frac{9}{s} + \ddots}}}} \tag{3.13}$$

If the continued fraction is truncated in n terms, then $T_n(s)$ can be written as

$$T_n(s) = \frac{k_0}{B_n(s)} \tag{3.14}$$

where k_0 is chosen such that $T_n(0) = 1$ and $B_n(s)$ is a Bessel polynomial of order n defined by the formula:

$$B_n(s) = \sum_{k=0}^{n} \frac{(2n-k)!\, s^k}{2^{n-k} k!\,(n-k)!} \tag{3.15}$$

The coefficients of higher degree Bessel polynomials are given in [18, p. 500], and an extensive table of roots of Bessel polynomials is given in [13], which shows that all roots are in the left half plane.

3.5 Rational Approximation of e^{-s} by Budak

In this approximation [3], a parameter k is introduced to split e^{-s} into two parts such that

$$e^{-s} \doteq \frac{e^{-ks}}{e^{-(k-1)s}}, \qquad 0 \leq k \leq 1 \tag{3.16}$$

and then e^{-ks} and $e^{-(k-1)s}$ are approximated independently by Storch functions mentioned in the previous section. Thus the resulting approximation of e^{-s} will have Bessel polynomials both in the numerator and in the denominator. The poles of the $e^{-(k-1)s}$ approximant will be the zeros of the final approximant, while the poles of the e^{-ks} approximant will be its poles. For realizability, the degree of the $e^{-(k-1)s}$ approximant should be less than the degree of the e^{-ks} approximant.

The amplitude, phase and time response characteristics are given in [3]. As k decreases, the frequency-domain characteristics improve, while the step response shows more overshoot and undershoot. For comparison with Khovanskii approximations, the step responses for $n = 3$ with $k = 0.6$ and for $n = 5$ with $k = 0.55$ are plotted in Figures 3.16 and 3.17, respectively; these have about the same undershoot as the corresponding Khovanskii non-all-pass function.

3.6 Rational Approximation of e^{-s} by Allemendou

This approximation [1] is obtained by the equation

$$e^{-s} \simeq \frac{f(s^2)}{B_n(s)} \tag{3.17}$$

where $B_n(s)$ is a Bessel polynomial of degree n defined by Equation

(3.15) and

$$f(\omega^2) = B_0\Big[1 + \frac{\omega^2}{2(2n-1)} + \frac{\omega^4}{2^2 2!(2n-1)(2n-3)} + \cdots + \frac{\omega^{2r}}{2^r r!(2n-1)(2n-3)\cdots(2n-2r+1)}\Big] \tag{3.18}$$

where $B_0 = 1.3.5\cdots(2n-1)$ and $\omega^2 = -s^2$.

When n is an odd integer, the degree of the numerator may not exceed $n-1$ and when n is an even integer, the degree of the numerator may not exceed n. Therefore, for the same polynomial $B_n(s)$ of degree n, any one of the following polynomials may be chosen as the polynomial $f(s^2)$:

$$\begin{aligned} f_{n,0}(\omega^2) &= B_0 \\ f_{n,1}(\omega^2) &= B_0\Big[1 + \frac{\omega^2}{2(2n-1)}\Big] \\ f_{n,2}(\omega^2) &= B_0\Big[1 + \frac{\omega^2}{2(2n-1)} + \frac{\omega^4}{8(2n-1)(2n-3)}\Big] \\ &\vdots \end{aligned} \tag{3.19}$$

The best approximation of the delay function in the case of an odd n is given by

$$e^{-s} \simeq \frac{f_{n,(n-1)/2}(s^2)}{B_n(s)} \tag{3.20}$$

and for an even n is given by

$$e^{-s} \approx \frac{f_{n,n/2}(s^2)}{B_n(s)} \tag{3.21}$$

3.7 Comparisons of the Four Functions

In this section, we will compare the frequency- and time-domain characteristics of the Khovanskii non-all-pass function, the Storch function, the Budak function and the Allemendou function. The amplitude response and the delay response for orders 3 and 5, respectively, are given in Figures 3.12, 3.13, and Figures 3.14, 3.15. The step responses of the various functions are given in Figures 3.16 and 3.17, and numerical values about their step responses are given in Tables 3.7 and 3.8

The following observations are made:

1. Both of the amplitude and delay characteristics of the Khovanskii non-all-pass function are superior to those of the Allemendou function or of the Storch function.

2. For the same amount of undershoot and for the same order of the function, the Khovanskii non-all-pass function shows a better rise-to-delay ratio than the Budak function, but has slightly more overshoot, while in the frequency domain, the former gives better amplitude and delay characteristics at low frequencies. Incidentally, the frequency characteristics of the Budak function improve, as the

parameter k decreases; the step response, however, deteriorates increasingly at the same time.

3. Although the Storch function has no undershoot and very little overshoot, the rise-to-delay ratio is the worst.

One may conclude that, as a whole, the Khovanskii non-all-pass function gives the best characteristics in the frequency and time domains within the tolerances of overshoot and undershoot obtained in these functions.

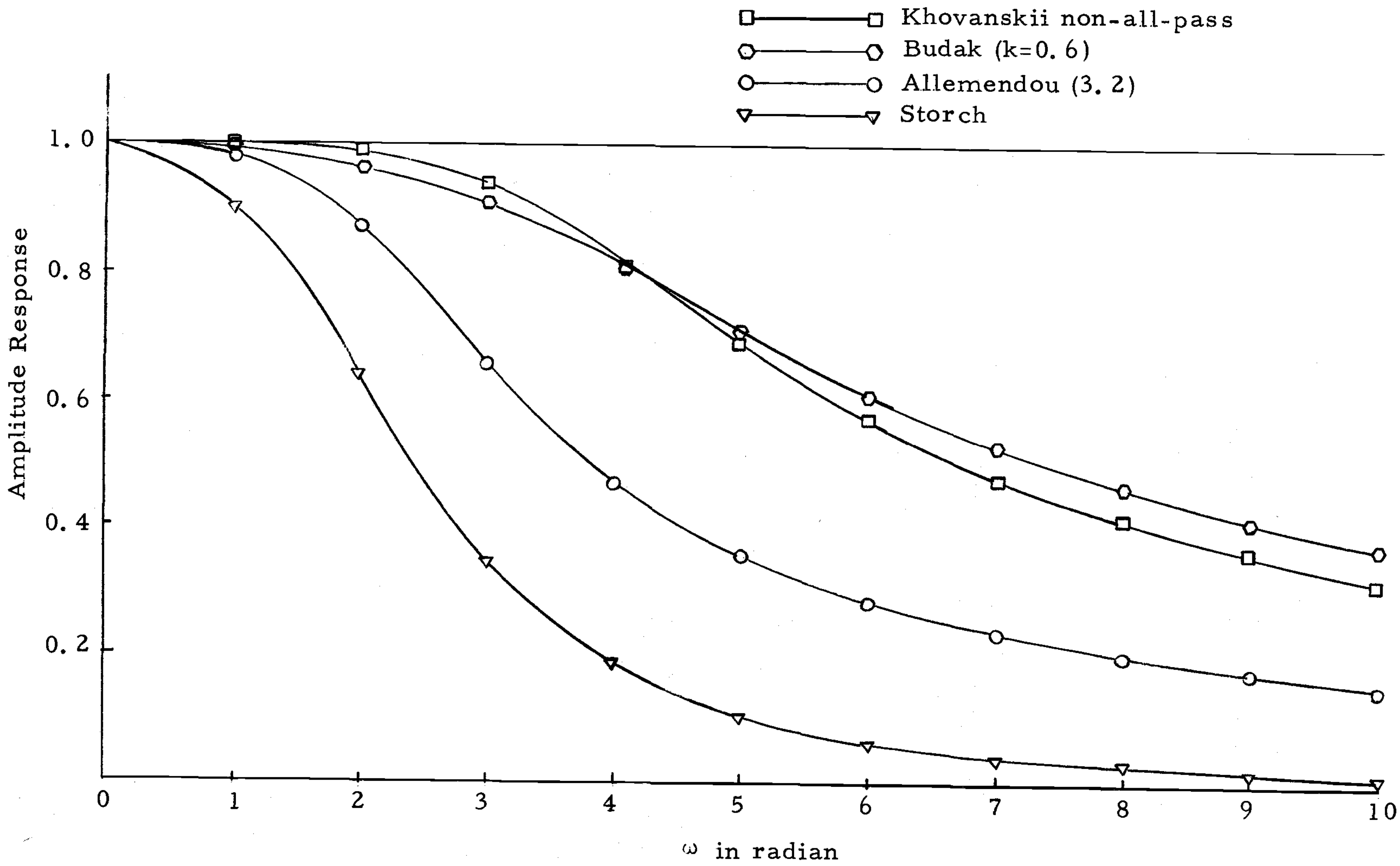


Figure 3. 12. The amplitude characteristics of the four functions of order 3.

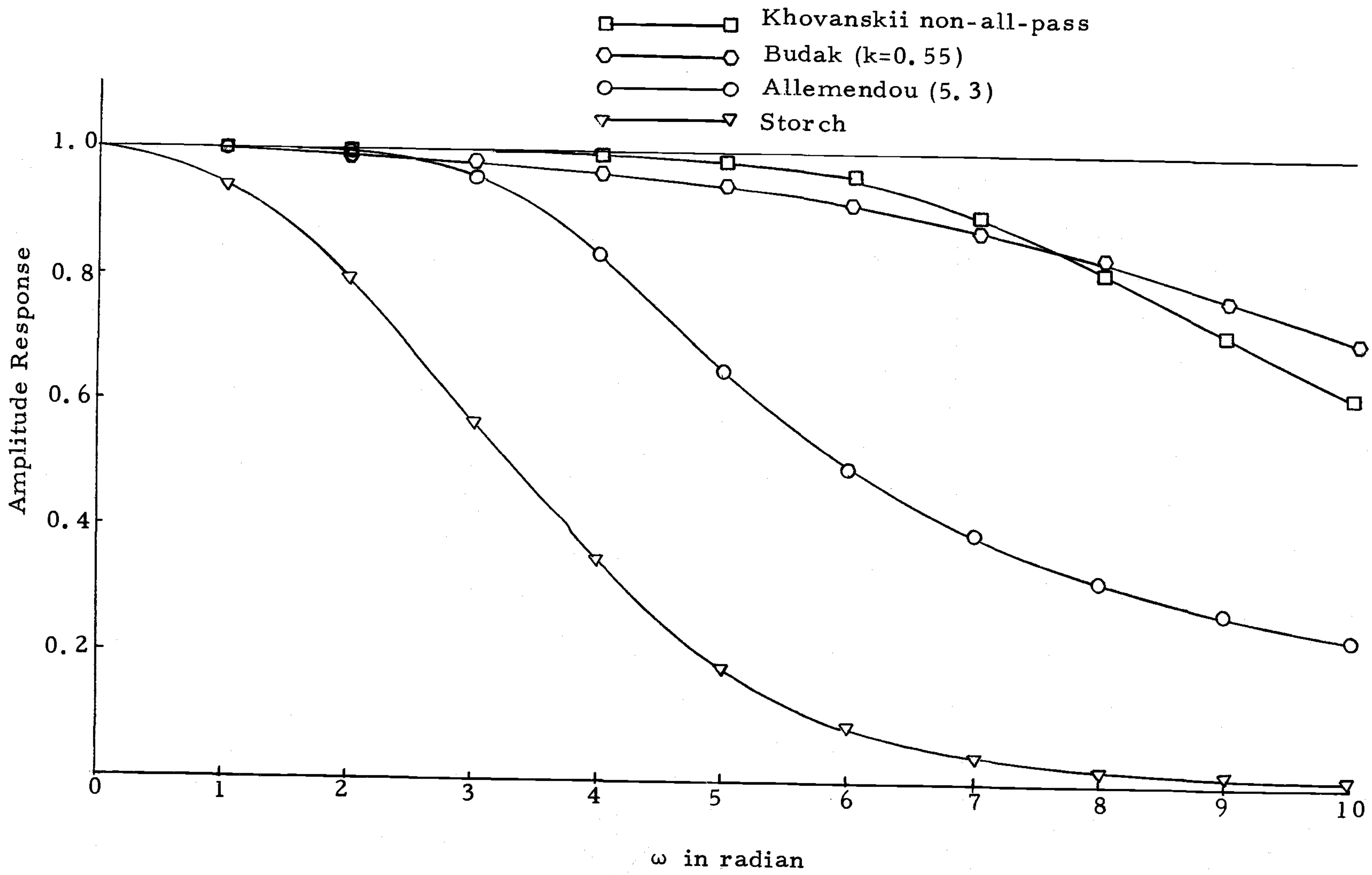


Figure 3. 13. The amplitude characteristics of the four functions of order 5.

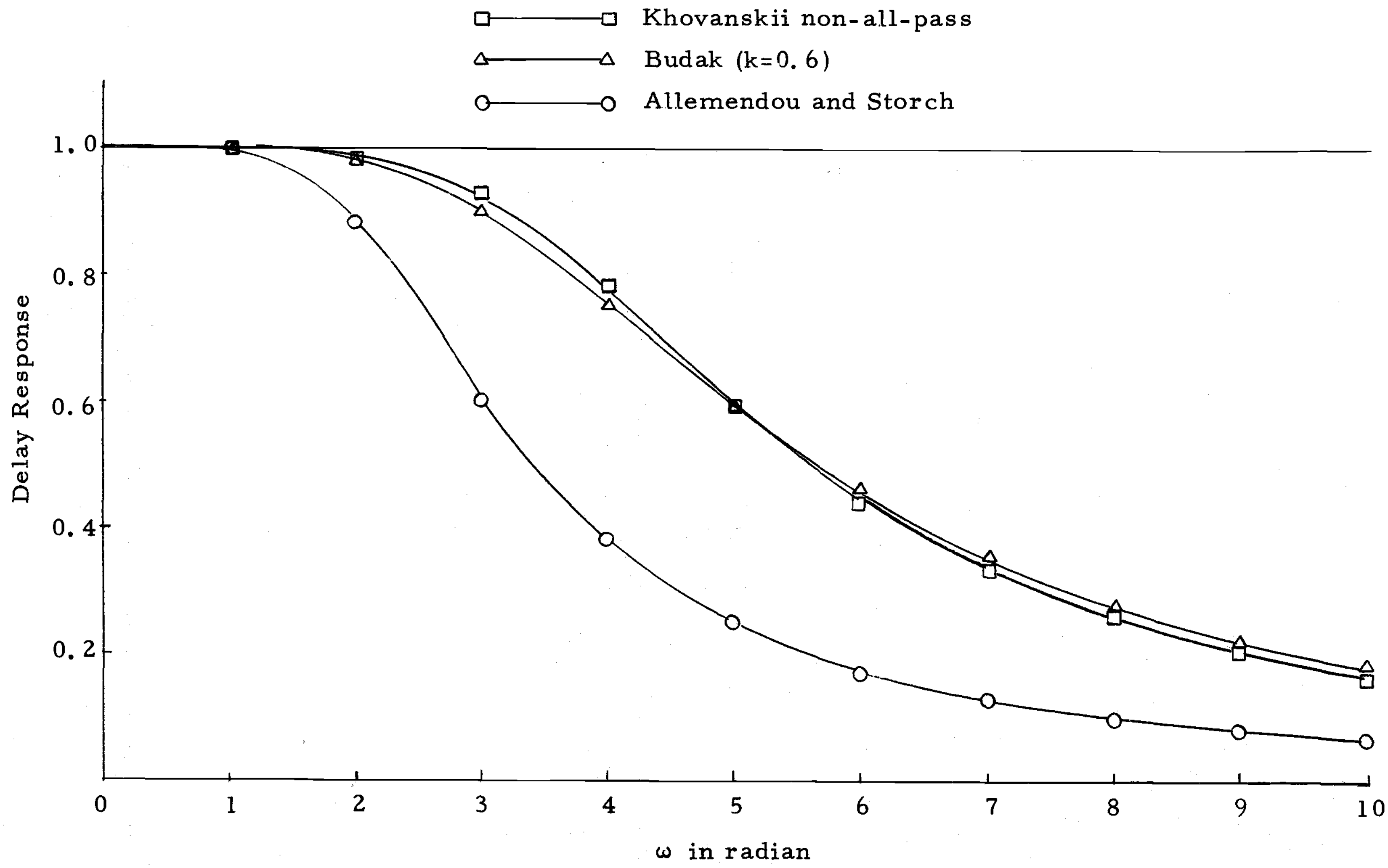


Figure 3.14. The delay characteristics of the four functions of order 3.

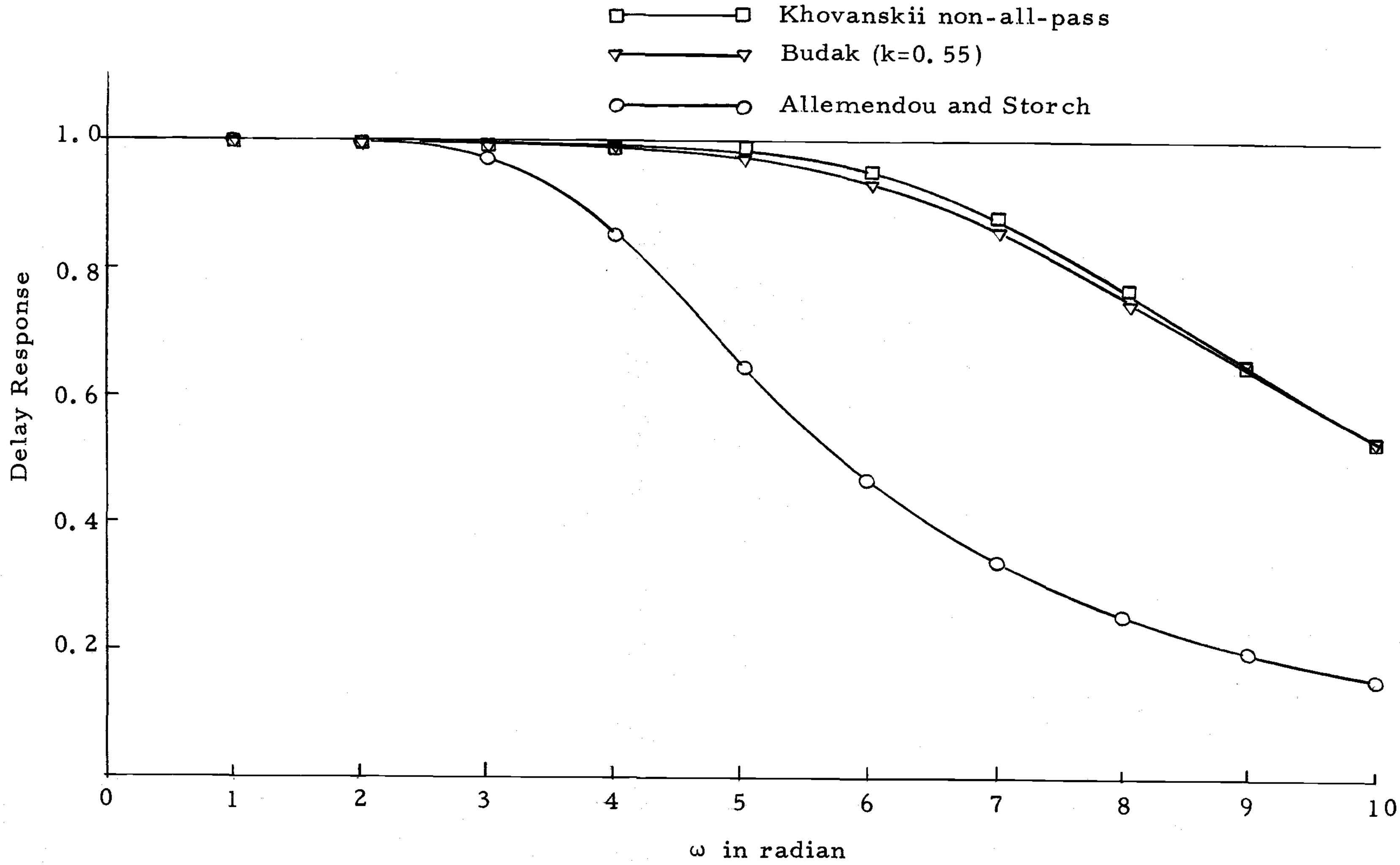


Figure 3. 15. The delay characteristics of the four functions of order 5.

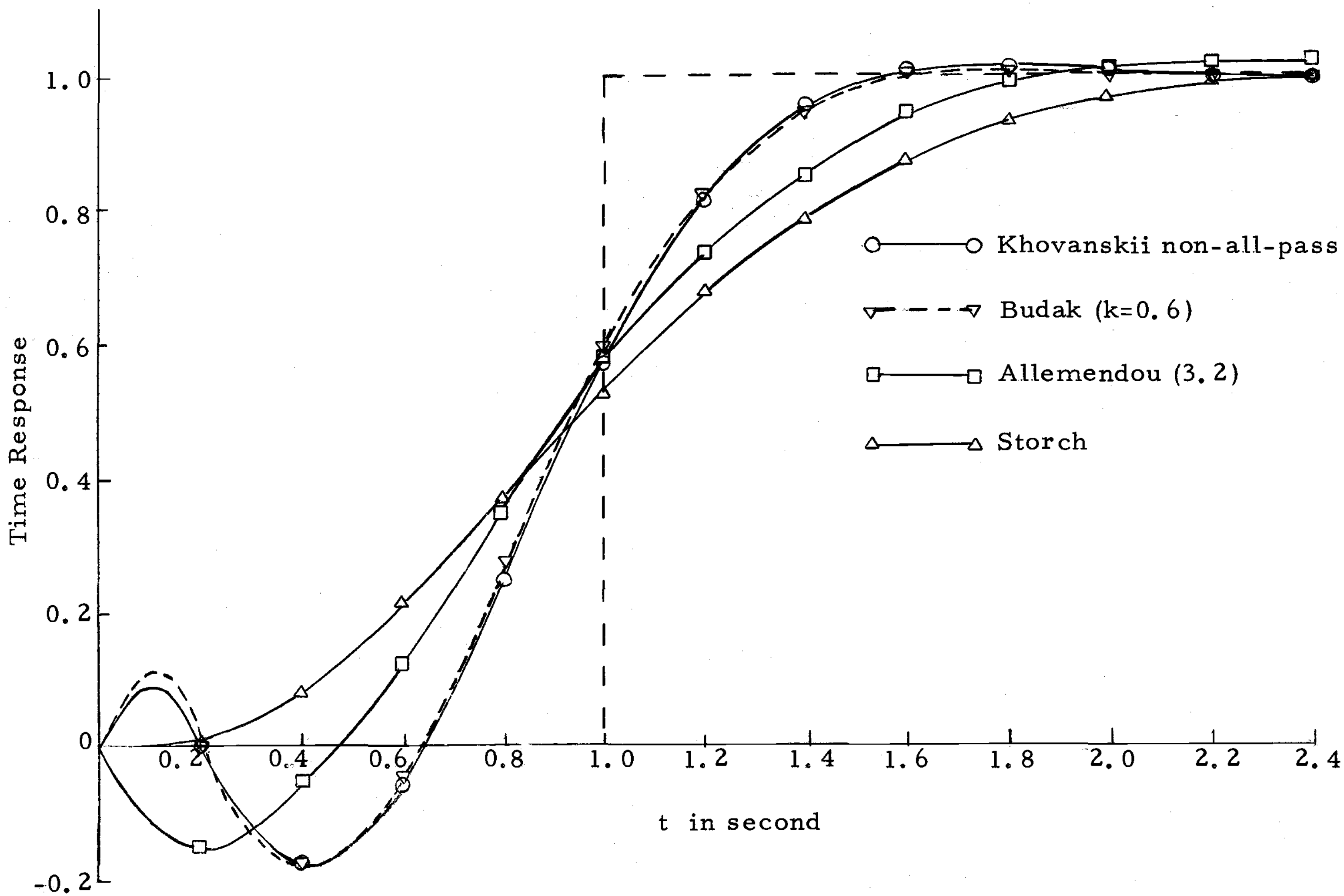


Figure 3.16. The step responses of the four functions of order 3.

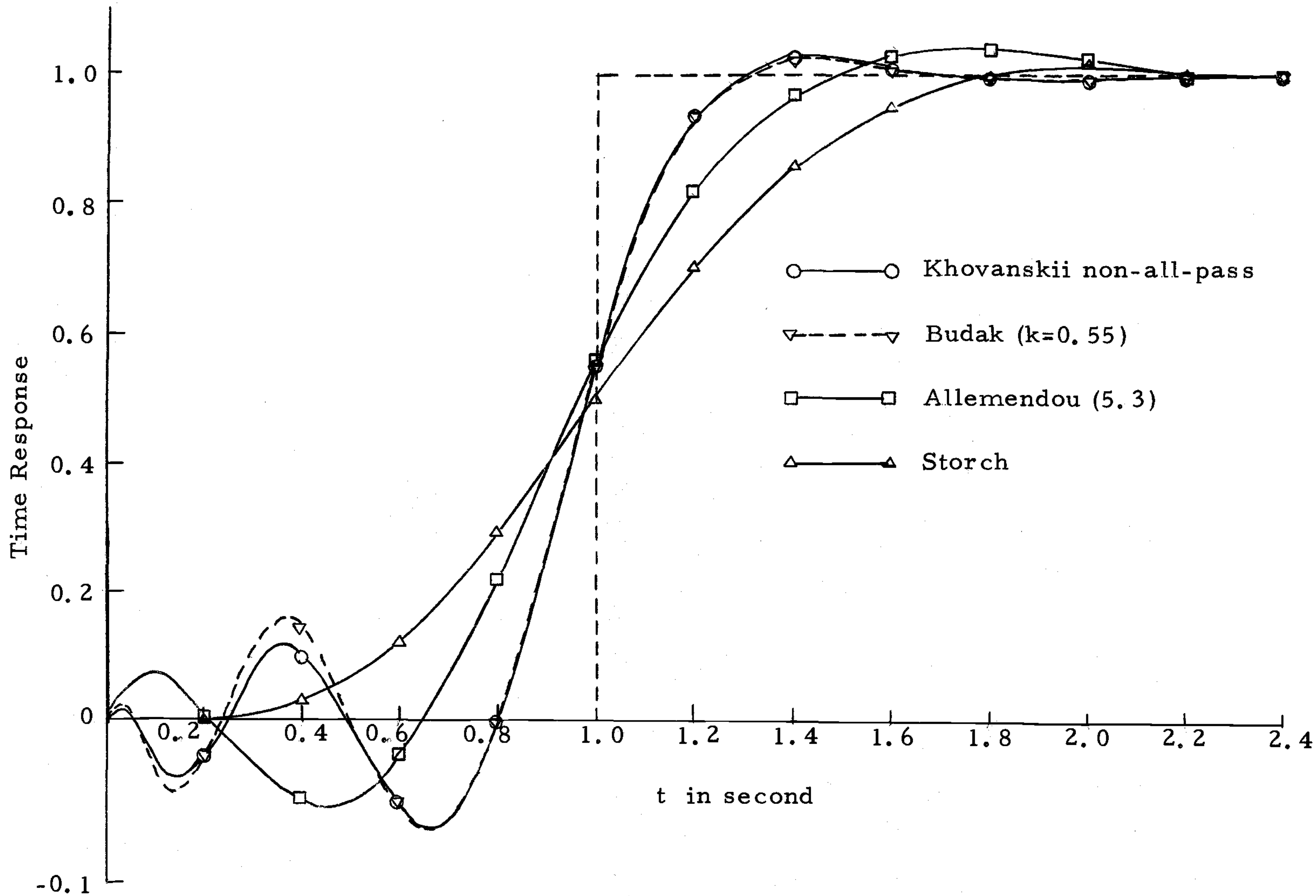


Figure 3.17. The step responses of the four functions of order 5.

Table 3.7. Time domain characteristics of four functions for n = 3.

Function	10% Response	90% Response	10%-90% Rise Time	50% Delay Time	Rise-to-Delay Ratio	Overshoot %	Undershoot %
Khovanskii (non-all-pass)	0.709	1.293	0.584	0.945	0.618	2.4	-17.6
Storch	0.424	1.666	1.242	0.957	1.298	0.9	--
Budak (k = 0.6)	0.697	1.310	0.613	0.935	0.655	1.1	-18.2
Allemendou	0.563	1.472	0.908	0.921	0.985	2.3	-14.6

Table 3.8. Time domain characteristics of four functions for n = 5.

Function	10% Response	90% Response	10%-90% Rise Time	50% Delay Time	Rise-to-Delay Ratio	Overshoot %	Undershoot %
Khovanskii (non-all-pass)	0.834	1.163	0.329	0.976	0.337	3.6	-16.1
Storch	0.562	1.469	0.907	0.989	0.917	0.8	--
Budak (0.55)	0.828	1.171	0.343	0.971	0.353	2.1	-16.8
Allemendou	0.722	1.285	0.563	0.963	0.585	3.5	-13.9

IV. OPTIMIZATION OF A DELAY FUNCTION

Although all the approximating functions considered in the previous chapters are derived systematically, they may not show the "best" responses in the frequency or time domain. In other words, there is still some room to adjust the coefficients or poles and zeros of the delay functions in order to improve the responses. The so-called network optimization is concerned with achieving the most desirable network characteristics by adjusting some parameters of the network functions or network elements under certain constraints.

Optimization can be performed either in the frequency domain or in the time domain, or in both domains. Since it is not easy to correlate the frequency response with the time response, and we are, in most cases, interested in the time response for the applications of delay networks, optimization of the step response rather than of the frequency response will be considered in this paper.

First, optimization of the all-pole rational delay function by adjusting the pole locations will be considered in order to obtain the minimum rise-to-delay ratio under specified tolerances of overshoot and undershoot. Then, optimization of a lossy delay ladder network by adjusting the element values will be considered.

Before going into the specific optimization, we will describe some basic techniques of optimization procedure.

4.1 Error Criterion

In any optimization problem, the error criteria must be, first of all, defined. The error between the current response and the desired response can be defined in many ways depending on the specific problem.

In this paper the error of the step response will be defined as the sum of the rise-to-delay ratio and the magnitude of the response over and under some specified tolerances of overshoot and undershoot:

$$F = \frac{T_r}{T_d} + w_1 \sum_{i=1}^{j} E_{o_i} + w_2 \sum_{i=1}^{k} E_{u_i} \tag{4.1}$$

where E_{o_i} and E_{u_i} are the overshoot and undershoot errors, respectively, exceeding the specified tolerances, and evaluated at discrete time points as shown in Figure 4.1; w_1 and w_2 are weighing functions.

The error defined in Equation (4.1) is a function of the poles and zeros of the network function, or a function of the element values of the actual network. We want to minimize this error by adjusting the independent variables.

The cut and try method should not be employed for finding the minimum error condition, since it is time-consuming and the true minimum point usually cannot be found in this way. The most widely used

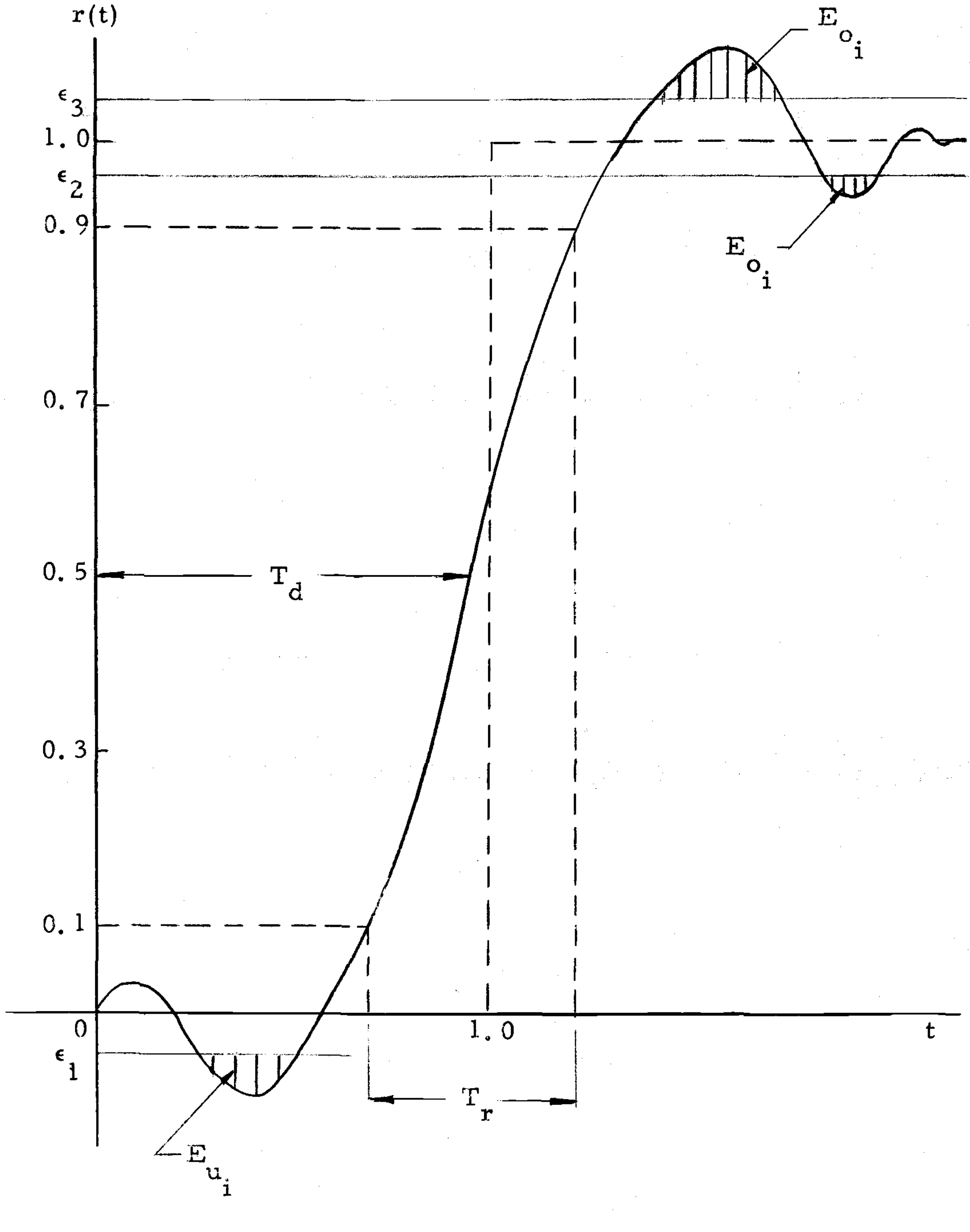


Figure 4.1. A delay network with overshoot and undershoot restrictions.

minimization method is the gradient method, in which the parameter adjustment follows the negative gradient of the error function.

4.2 Minimization Procedure

The Fletcher-Powell method of minimization [6] used in this paper is a variation of the gradient method. Only first derivatives of the function to be minimized are required, yet the technique has "second order" convergence; that is, it minimizes a positive-definite quadratic form of n variables in n iterations. The method has found wide acceptance and is generally regarded as the most powerful general procedure known at the present time for finding the local minimum in the unconstrained minimization problems.

Central to the method is a symmetric, positive-definite matrix $\underline{H}_i$, which is updated at each iteration, and which determines the current direction of motion $\underline{S}_i$ by multiplying the current gradient vector. An iteration is described as follows:

$\underline{H}_i$ = any positive definite matrix

$F(\underline{x}_i)$ = the error function to be minimized

$\underline{x}_i$ = the set of adjusted variables at i^{th} iteration

$$\underline{S}_i = - \underline{H}_i \underline{\nabla} F(\underline{x}_i) \tag{4.2}$$

Choose α_i so as to minimize $F(\underline{x}_i + \alpha_i \underline{S}_i)$. Denoting the minimizing value of α_i as $\hat{\alpha}_i$, define

$$\underline{\sigma}_i = \hat{\alpha}_i \underline{S}_i \tag{4.3}$$

$$\underline{d}_i = \underline{\nabla} F(x_{i+1}) - \underline{\nabla} F(\underline{x}_i) \tag{4.4}$$

$$\underline{A}_i = \frac{\underline{\sigma}_i \underline{\sigma}_i^T}{\underline{\sigma}_i^T \underline{d}_i} \tag{4.5}$$

$$\underline{B}_i = \frac{\underline{H}_i \underline{d}_i \underline{d}_i^T \underline{H}_i}{\underline{d}_i^T \underline{H}_i \underline{d}_i} \tag{4.6}$$

Note that the numerators of $\underline{A}_i$ and $\underline{B}_i$ are both matrices while the denominators are scalars. Now let

$$\underline{H}_{i+1} = \underline{H}_i + \underline{A}_i + \underline{B}_i \tag{4.7}$$

Fletcher and Powell have proved the following:

1. The matrix $\underline{H}_i$ is positive-definite for all i. As a consequence of this fact, the method will always converge, since

$$\frac{d}{d\alpha_i} F(\underline{x}_i + \alpha_i \underline{s}_i)\Big|_{\alpha_i = 0} = -\underline{\nabla} F^T(\underline{x}_i)\underline{H}_i \underline{\nabla} F(\underline{x}_i) < 0 \tag{4.8}$$

That is, the function is initially decreasing along the direction $\underline{S}_i$, so that the function can be decreased at each iteration by minimizing down $\underline{S}_i$.

2. When this method is applied to the quadratic

$$F(\underline{x}) = \underline{C}^T \underline{x} + \underline{x}^T \underline{A}\, \underline{x} \tag{4.9}$$

where $\underline{A}$ is a real symmetric matrix and $\underline{C}$ is a column vector, we get the following results:

a. The minimum is reached in n steps.

b. The matrix $\underline{H}_i$ converges to the inverse of the matrix of second partial derivatives of the quadratic after n iterations, that is

$$\underline{H}_n = \underline{A}^{-1} \tag{4.10}$$

Subroutine MIN in Appendix B implements this technique.

4.3 One-Dimension Minimization

Each iteration of the Fletcher-Powell method requires minimization of the one dimensional function $F(\underline{x}_i + \alpha_i \underline{S}_i)$, that is, the calculation of $\hat{\alpha}_i$. The method of successive quadratic polynomial fitting [5, p. 190] is used in this paper.

The computer program for this technique is given in Subroutine QUAD in Appendix B.

4.4 Optimization of a Rational Delay Function

In the first optimization problem, the real parts and imaginary parts of zeros and poles of a rational function are used as independent variables to minimize the error defined in Equation (4.1) under 2% and 5% overshooting tolerances.

The Storch delay functions are used as the starting functions. The flow chart of the main program is given in Figure 4.2. After 20

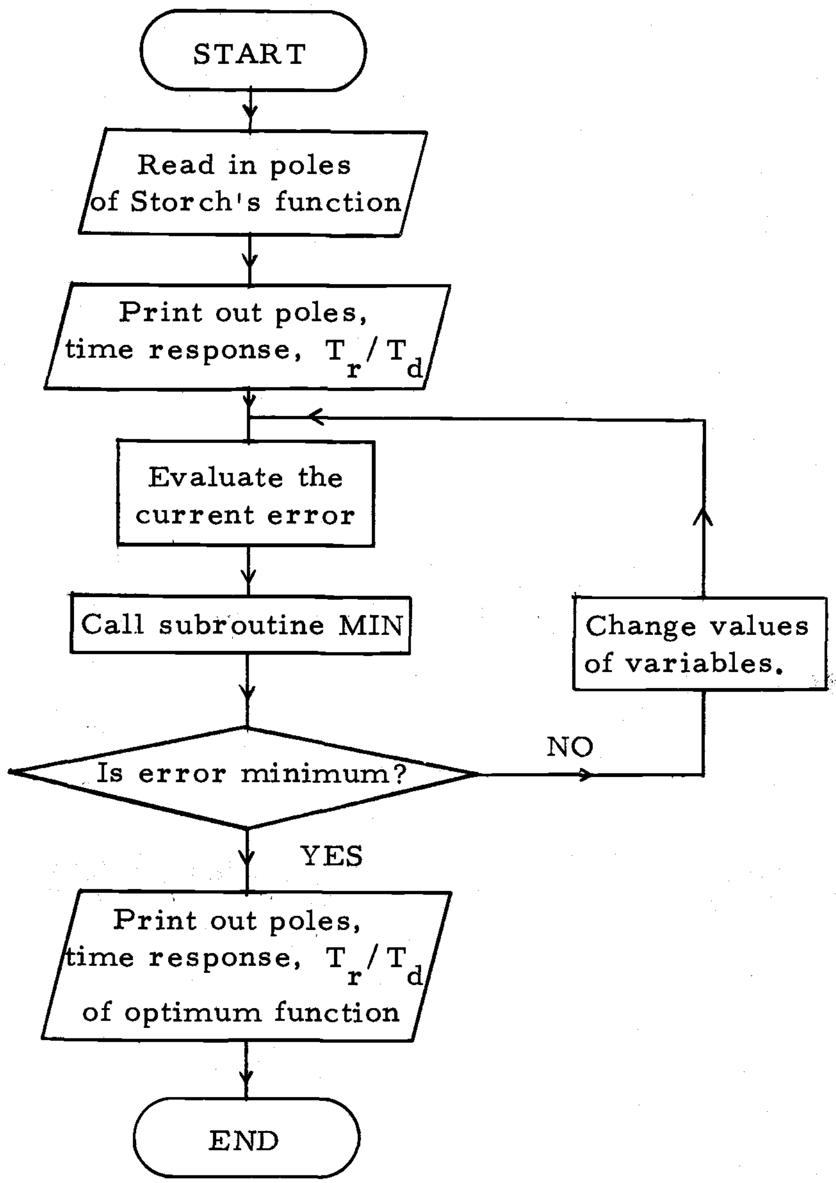


Figure 4.2. Flow chart of optimization with poles as variables.

Table 4.1. Overshoot and rise-to-delay ratio of Storch and optimum functions.

n	% Overshoot Restriction	Overshoot %	$\frac{T_r}{T_d}$	% Improvement of $\frac{T_r}{T_d}$	Remarks
3		0.9	1.298		Starting function
3	2	2.7	1.196	7.8	Optimum (2%)
3	5	5.4	1.124	13.4	Optimum (5%)
4		0.8	1.063		
4	2	2.1	0.931	12.5	
4	5	4.9	0.862	19.0	
5		0.8	0.917		
5	2	2.2	0.871	5.0	
5	5	5.2	0.759	17.2	
6		0.6	0.818		
6	2	2.7	0.727	11.1	
6	5	5.0	0.708	13.6	
7		0.4	0.743		
7	2	3.2	0.652	12.3	
7	5	5.4	0.635	14.5	
8		0.3	0.687		
8	2	2.5	0.640	6.8	
8	5	5.0	0.604	12.0	
9		0.2	0.641		
9	2	2.1	0.595	7.2	
9	5	5.0	0.562	12.3	
10		0.1	0.604		
10	2	2.3	0.557	7.6	
10	5	5.3	0.524	13.2	

Table 4.2. Pole locations of Storch and optimum functions.

			OPTIMUM FUNCTIONS			
	Storch Functions		2% Overshoot Restriction		5% Overshoot Restriction	
n	-x	± jy	-x	± jy	-x	± jy
3	2.322185		2.373398		2.443109	
	1.838907	1.754381	1.582264	1.952957	1.390272	2.071093
4	2.896211	0.867234	2.676115	0.850819	2.614168	0.899916
	2.103789	2.657418	1.415178	3.533409	1.036038	3.752518
5	3.646739		3.705123		3.584171	
	3.351956	1.742661	3.351830	1.898311	3.369780	2.057806
	2.324674	3.571023	2.066494	3.616314	1.042549	5.070504
6	4.248359	0.867510	4.188812	0.878212	4.514530	1.073753
	3.735708	2.626272	3.680436	2.902979	3.589436	3.020803
	2.515932	4.492673	1.461224	5.078451	1.939724	4.436298
7	4.971787		4.968211		5.171281	
	4.758291	1.739286	4.793421	1.844859	4.950218	2.127506
	4.070139	3.517174	3.940829	3.917871	3.759559	3.943848
	2.685677	5.420694	1.571265	5.803998	2.029693	5.249393
8	5.587886	0.867614	5.793161	0.978181	5.913013	1.048294
	5.204841	2.616175	5.231987	2.875669	5.243229	3.050436
	4.368289	4.414443	4.129868	4.615801	3.953359	4.785195
	2.838984	6.353911	2.486916	6.201951	2.137711	6.159643
9	6.297019		6.428657		6.532973	
	6.129368	1.737848	6.306272	1.949209	6.445862	2.115967
	5.604422	3.498157	5.553113	3.803687	5.512169	4.043085
	4.638440	5.317272	4.323672	5.496789	4.077652	5.633756
	2.979261	7.291464	2.606321	7.079707	2.323906	6.910807
10	6.922045	0.867665	7.241259	0.987214	7.493606	1.081724
	6.615291	2.611568	6.779793	2.920500	6.909877	3.164777
	5.967528	4.384947	5.872524	4.740152	5.797460	5.021191
	4.886220	6.224985	4.539904	6.409217	4.265926	6.555431
	3.108916	8.232699	2.746967	8.025722	2.459750	7.862887

to 60 iterations, depending on the order of the functions, the optimum points are reached. The optimum results are given in Tables 4.1 and 4.2. In Table 4.1, the first row for each value of n gives the response characteristics of the starting function (the Storch function), while the second and third rows give the response characteristics of optimum functions with 2% and 5% overshoot tolerances, respectively. For 2% overshoot tolerance, the rise-to-delay ratios improve by 5 to 12.5%, and for 5% overshoot tolerance they improve by 10.3 to 19%. No undershoots are seen in the optimum functions, as in the initial functions. In Table 4.2, the pole locations of the Storch and optimum all-pole delay functions are given. We see that all poles of all optimum functions are located in the left half plane; hence the optimum transfer functions are still realizable.

In conclusion, we can improve the rise-to-delay ratio significantly by allowing a small amount of overshoot.

4.5 Optimization of Lossy Delay Network

Although the transfer functions obtained in the last section are optimum, the actually realized network may not show the same optimum time response. In the realization ideal inductors and capacitors are usually assumed, whereas practical elements always have some loss. Therefore, the more realistic network synthesis must take into consideration the loss associated with reactive elements. This loss can be accounted for in an approximate manner by combining a resistance in series with each inductance and one in parallel with each capacitance. If these resistances are variables of each inductance and capacitance, then

$$Z_L = Ls + R_L = Ls + F_L L \tag{4.11}$$

$$Z_C = \frac{1}{Cs + \frac{1}{R_C}} = \frac{1}{Cs + F_C C} \tag{4.12}$$

where $F_L = R_L/L$ and $F_C = 1/R_C C$ are the dissipation constants of inductance and capacitance, respectively.

In this section, a ladder delay network which realizes the Storch transfer function with ideal elements is used as the initial network in the optimization. The losses of reactive elements are then incorporated in the way described above, as shown in Figure 4.4. Optimization is carried out to minimize the error defined by Equation (4.1), this time by the use of reactive element values as independent variables.

Figure 4.3 shows the flow chart of this optimization procedure and Tables 4.3 and 4.4 give the optimum results for a network of degree 5. We see that as the loss increases the rise-to-delay ratio worsens with a decreased final value of the step response.

Incidentally, although the loss factor is assumed to be the same for all of the same kind of reactive elements, it may take different values for different elements, for which cases the program can still be used

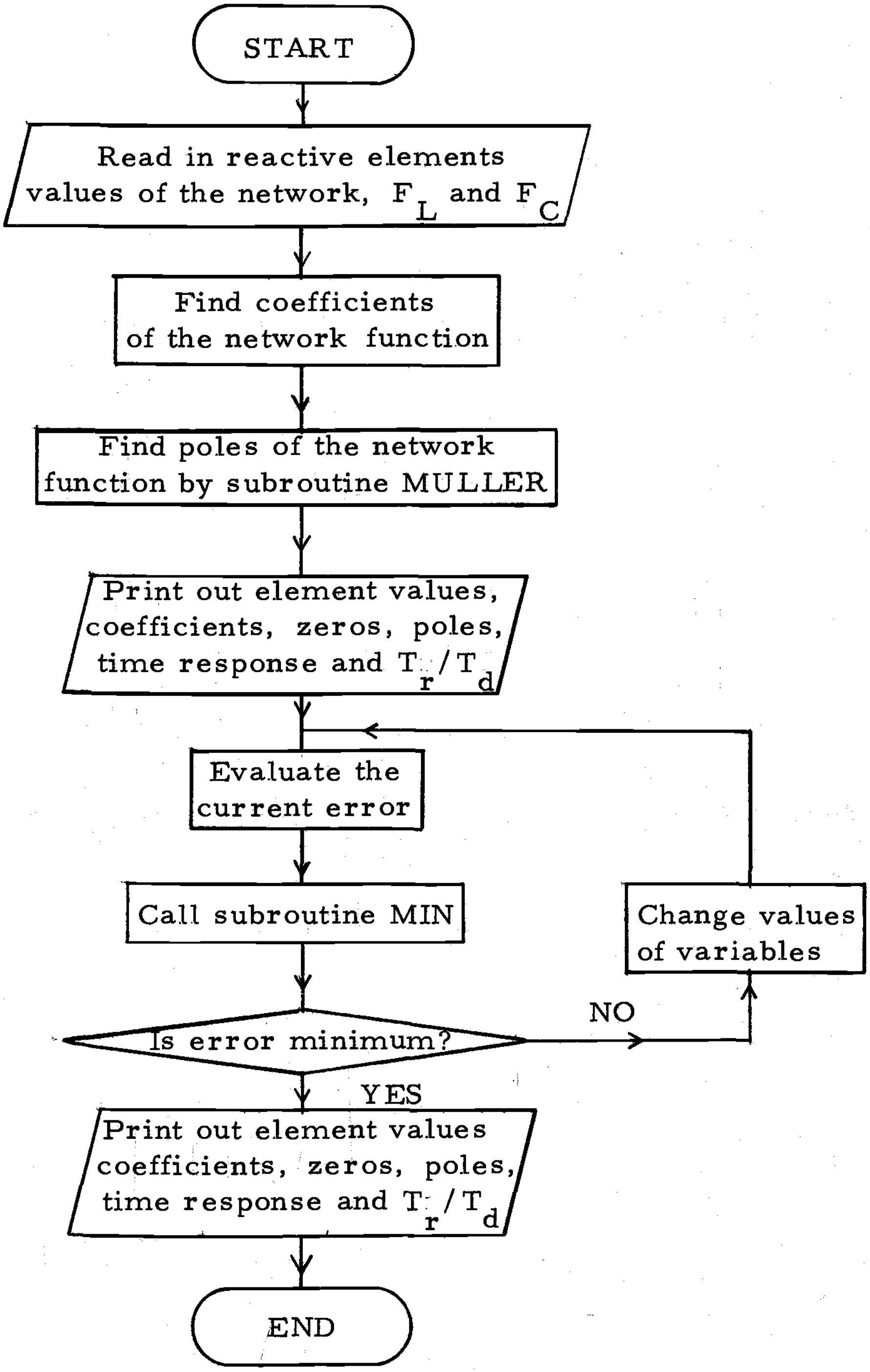


Figure 4. 3. Flow chart of the optimization with element values of a delay network as variables.

with only a slight modification.

It is noted that the optimization with poles of the transfer function as variables (Section 4.5) and the optimization with element values as variables in the lossless ladder network ($F_L = F_C = 0$) give different results; the latter method gives a better result, as we can see by comparing rows 8 and 9 of Table 4.1 and rows 2 and 3 of Table 4.4. This distinction may not be surprising if one considers the different set of variables used in the two methods, and the possibility of many local minimum points in each optimization.

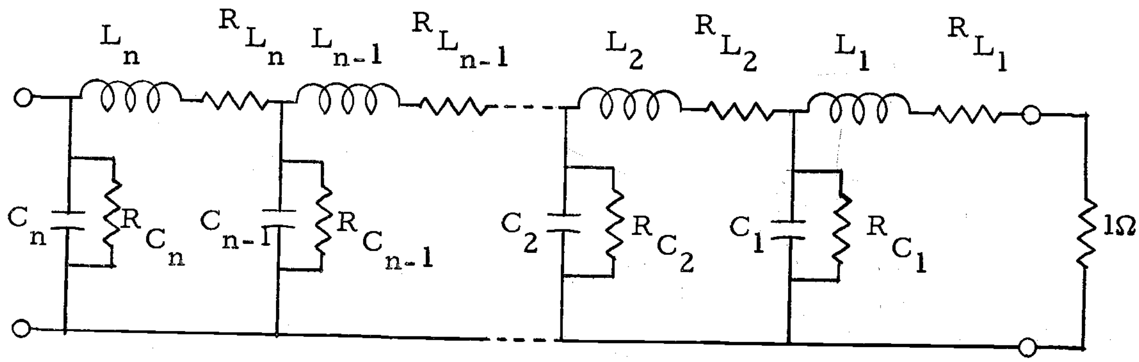


Figure 4.4. A lossy ladder delay network.

Table 4.3. Element values of Storch-lossy network of order 5 with different values of F_L, F_C and percent overshoot restriction.

L_1	C_1	L_2	C_2	L_3	C_3	F_L, F_C, % Overshoot Restriction
0	0.066667	0.194805	0.310256	0.421499	0.623077	(Storch)
0	0.405882	0.371800	0.379261	0.381645	0.548498	0, 0, 2
0	0.535838	0.360475	0.597835	0.397635	0.494072	0, 0, 5
0	0.432922	0.353481	0.369806	0.355133	0.531612	0.1, 0.1, 2
0	0.273152	0.339383	0.350415	0.292874	0.399082	0.1, 0.1, 5
0	0.362544	0.546126	0.569121	0.341084	0.640922	0.1, 0.2, 2
0	0.291874	0.625670	0.655940	0.405818	0.609130	0.1, 0.2, 5
0	0.388103	0.245695	0.387371	0.224473	0.467748	1.0, 1.0, 2
0	0.877547	0.217286	0.771390	0.233909	0.405358	1.0, 1.0, 5

Table 4.4. Time-domain characteristics of Storch lossy network of order 5 with different values of F_L and F_C.

F_L	F_C	% Overshoot Restriction	Overshoot %	Final Value	$\frac{T_r}{T_d}$	Remarks
0	0		0.8	1.0	0.917	(Storch)
0	0	2	1.9	1.0	0.709	
0	0	5	3.8	1.0	0.667	
0.1	0.1		0.7	0.905	0.927	
0.1	0.1	2	1.9	0.880	0.717	
0.1	0.1	5	5.0	0.904	0.688	
0.1	0.2		2.5	0.828	0.900	
0.1	0.2	2	2.0	0.750	0.718	
0.1	0.2	5	4.9	0.750	0.705	
1.0	1.0		0.2	0.389	0.996	
1.0	1.0	2	2.0	0.366	0.761	
1.0	1.0	5	5.2	0.252	0.728	

V. REALIZATION

In this chapter, we will consider the realization of the Khovanskii transfer function by passive and active networks.

5.1 Passive Network Realization

Since the all-pass and non-all-pass functions can be realized in a similar way, we will consider only the latter class of functions.

Because of the restriction of zero locations of the Khovanskii non-all-pass function in the right half plane, most of the methods of passive transfer function synthesis fail. Darlington's methods (for the single or double-terminated network), Cauer's parallel partial fraction network realization, zero-shifting technique and bridged-T network realization are all inadequate. However, constant-R symmetrical lattice realization is possible for the Khovanskii non-all-pass function of any order.

The order 3 non-all-pass function will be taken up for illustration.

Given the transfer impedance function

$$Z_{12}(s) = \frac{3s^2 - 24s + 60}{s^3 + 9s^2 + 36s + 60} ,$$

we want to realize it by a symmetrical lattice network terminated by 1 ohm resistance [17, p. 346]. The two arm impedances are calculated as

$$z_a = \frac{1 - Z_{12}}{1 + Z_{12}}$$

$$= \frac{s^3 + 6s^2 + 60s}{s^3 + 12s^2 + 12s + 120}$$

$$= \cfrac{1}{\cfrac{2}{s} + \cfrac{1}{\cfrac{6}{s} + \cfrac{1}{\cfrac{10}{s} + 1}}}$$

$$z_b = \frac{1}{z_a} = \cfrac{2}{s} + \cfrac{1}{\cfrac{6}{s} + \cfrac{1}{\cfrac{10}{s} + 1}}$$

Then the network of Figure 5.1 is obtained

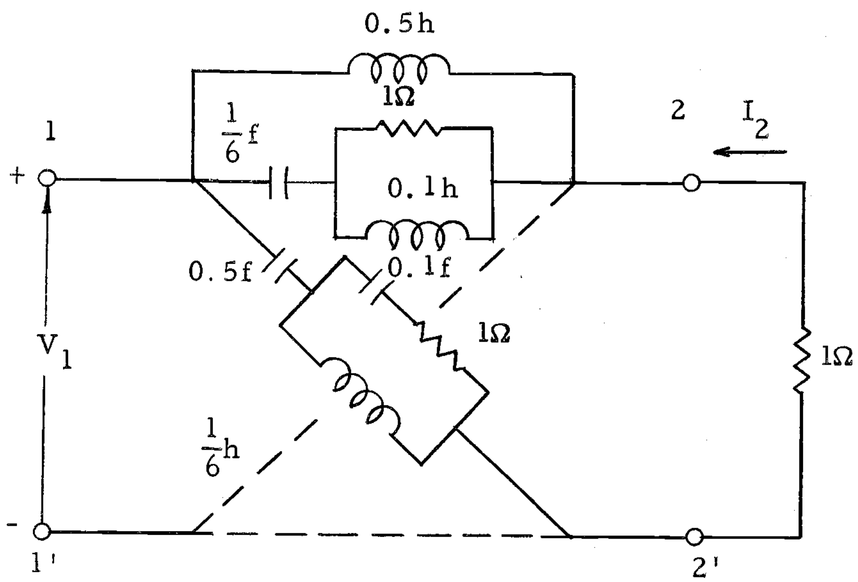


Figure 5.1. Realization of Khovanskii non-all-pass transfer impedance function as a constant-R symmetrical lattice.

The lattice network in Figure 5.1 can be converted into an unbalanced network [9, p. 254] as shown in Figure 5.2.

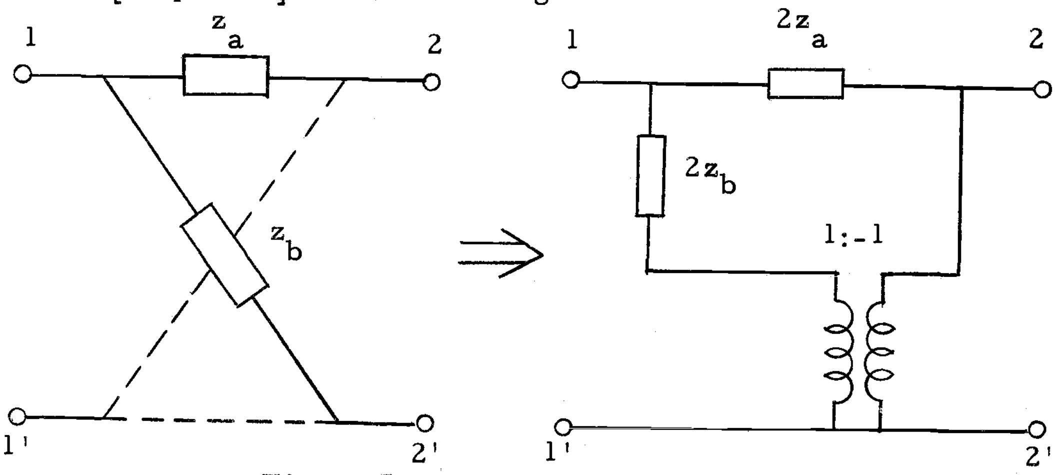


Figure 5.2. Unbalancing a lattice.

5.2 Active Network Realization

Among the many methods of transfer function synthesis by an active network, the RC-operational-amplifier circuit is advantageous in many respects. High-quality operational amplifiers are readily available commercially and a minimum number of capacitors are needed, which is a desirable feature in an integrated circuit. In this method, the given transfer function is factored into first and second order functions, each factor is simulated by an RC-operational-amplifier circuit, and the resulting sections are cascaded. Then, the impedance level of each section can be changed independently of other sections, and there is almost no interaction among sections. The RC-operational-amplifier circuit is relatively insensitive to the change of the parameters of active elements.

The Khovanskii non-all-pass function can be realized as a voltage transfer function by the use of RC-operational-amplifier circuits. A third order function is used as an illustration. The given voltage transfer function is first factored as follows:

$$T(s) = \frac{V_2}{V_1} = \frac{3s^2 - 24s + 60}{s^3 + 9s^2 + 36s + 60}$$

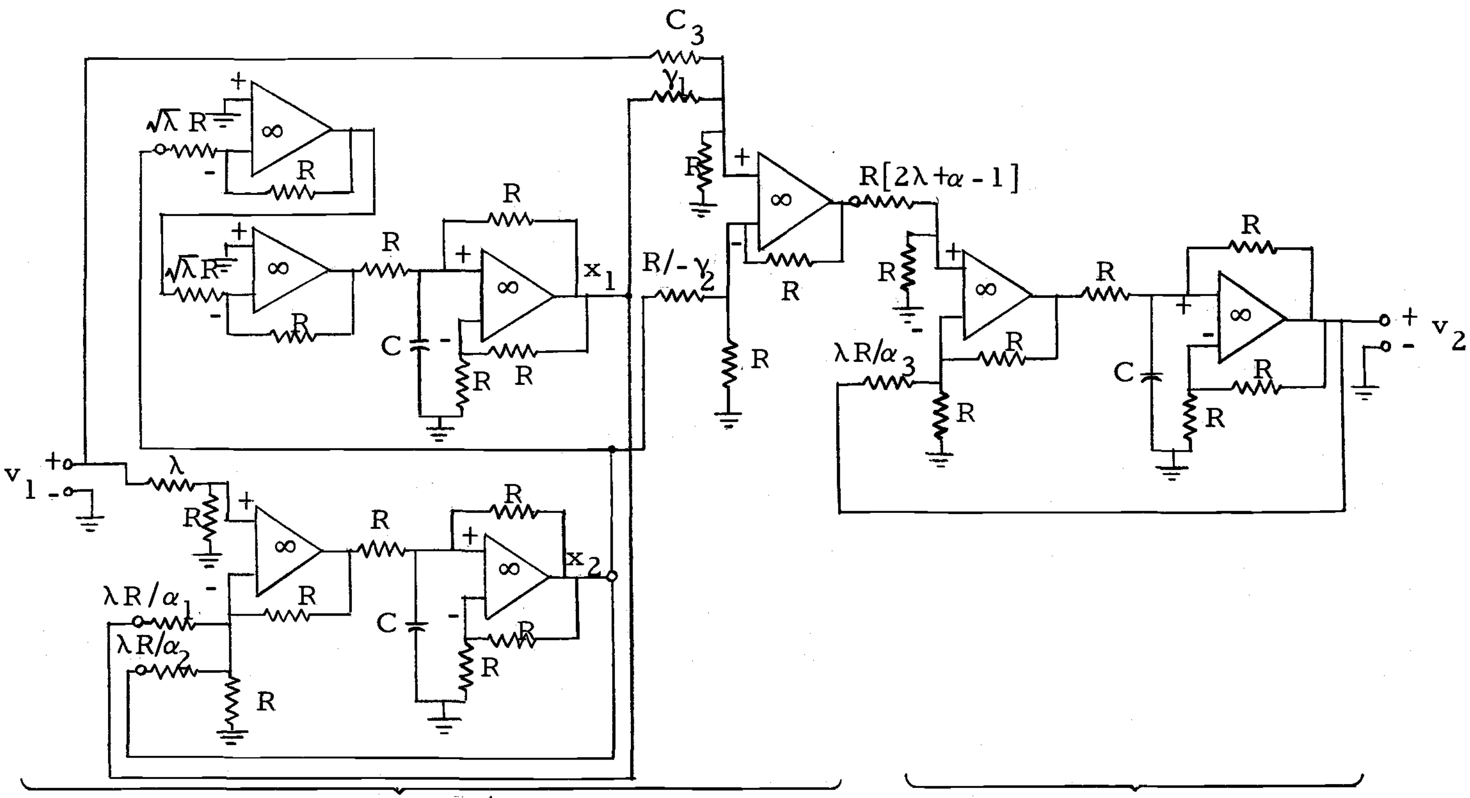
$$= \frac{3(s^2 - 8s + 20)}{s^2 + 5.362166s + 16.493329} \cdot \frac{1}{s + 3.637834}$$

$$= \left(3 + \frac{-40.086498s + 10.520013}{s^2 + 5.362166s + 16.493329}\right) \cdot \left(\frac{1}{s + 3.637834}\right)$$

$$= \left(c_3 + \frac{\gamma_2 s + \gamma_1}{s^2 + \alpha_2 s + \alpha_1}\right) \cdot \left(\frac{1}{s + \alpha_3}\right)$$

$$= T_1(s)\, T_2(s)$$

The synthesis method for the first and second order transfer functions by RC-operational-amplifier circuits are given in many text books on RC-active-network synthesis. In particular, the state-variable method [12] has the advantage of low sensitivity to the parameter variations. In this paper $T_1(s)$ and $T_2(s)$ are realized by the state-variable method. The complete network is shown in Figure 5. 3; any load impedance can be connected at the output without changing the overall voltage transfer function.



$$T_1(s) = C_3 + \frac{\gamma_2 s + \gamma_1}{s^2 + \alpha_2 s + \alpha_1}$$

$$T_2(s) = \frac{1}{s + \alpha_3}$$

Figure 5. 3. Active network realization of Khovanskii-non-all-pass function of order 3.

VI. CONCLUSION

In the first part of this paper, two new classes of rational approximations to the ideal delay function based on the Khovanskii continued fraction expansion of e^x were studied in detail in both the frequency domain and the time domain. Comparisons of the Khovanskii non-all-pass approximation with other approximations (by Budak, Allemendou and Storch) were made. The results show that, as a whole, the Khovanskii non-all-pass function gives the best frequency- and time-domain characteristics within the same tolerances of overshoot and undershoot in the step response.

The coefficients of the Khovanskii approximations, their poles and zeros and the response characteristics of the step response are all tabulated for convenient reference.

In the second part of this paper, time-domain optimizations of the delay function were performed by first taking the pole locations of the Storch all-pole delay function as variables and then considering the element values of the lossy LC ladder network as variables. The result shows that considerable improvement can be made in the rise-to-delay ratio within the specified overshoot, by either optimizing the pole locations or the lossy element values.

All the pertinent response characteristics of the optimum results are tabulated for $n = 3$ to 10 in the first case, which may have practical values for the designers of delay networks. Although optimization for the second case was performed for $n = 5$ only, the computer program can be used for any n.

Finally, typical realizations of the Khovanskii non-all-pass function were given in both passive and active networks.

BIBLIOGRAPHY

1. Allemendou, P. Low-pass filters approximating - in modulus and phase - the exponential function. IEEE Transactions on Circuit Theory 13:298-301. 1966.

2. Balabanian, N. Network synthesis. Englewood Cliffs, Prentice-Hall, 1958. 433 p.

3. Budak, A. A maximally flat phase and controllable magnitude approximation. IEEE Transactions on Circuit Theory 12:279. 1965.

4. Calahan, D. A. Computer design of linear frequency selective networks. Proceedings of the IEEE 53:1701-1706. 1965.

5. ________________ Computer-aided network design. New York, McGraw-Hill, 1968. 297 p.

6. Fletcher, R. and M. J. D. Powell. A rapid convergent descent method for minimization. The British Computer Journal 6:163-168. 1963.

7. Guillemin, E. A. Synthesis of passive networks. New York, John Wiley, 1957. 741 p.

8. Jess, J. and H. W. Schüssler. A class of pulse-forming networks. IEEE transactions on Circuit Theory 12:296-299. 1965.

9. Karni, S. Network theory: analysis and synthesis. Boston, Allyn and Bacon, 1966. 483 p.

10. Khovanskii, A. N. The application of continued fractions and their generalizations to problems in approximation theory, tr. by Peter Wynn. Groningen, Noordhoff, 1963, 212 p.

11. Muller, D. E. A method for solving algebraic equations using an automatic computer. Mathematical Tables and Other Aids in Computation 10:208-215. 1956.

12. Newcomb, R. W. Active integrated circuit synthesis. Englewood Cliffs, Prentice-Hall, 1968. p. 68-79.

13. Orchard, H. J. The roots of maximally flat delay polynomials. IEEE Transactions on Circuit Theory 12:452-454. 1965.

14. Storch, L. An application of modern network synthesis to the design of constant-time-delay networks with low-Q elements. IRE Convention Record 2:105-117. 1954

15. Su, K. L. The nominal delay and rise time of lumped delay networks. IEEE Transactions on Circuit Theory 16:574-577. 1969.

16. Thomson, W. E. Delay networks having maximally-flat frequency characteristics. Proceedings of IEE 96:487-490. 1949.

17. Van Valkenburg, M. E. Modern network synthesis. New York, John Wiley, 1960. 498 p.

18. Weinberg, L. Network analysis and synthesis. New York, McGraw-Hill, 1962. 692 p.

19. ________________ and T. C. G. Wagner. Time domain analysis of a class of delay network. Ninth Midwest Symposium on Circuit Theory 13:1-15. 1966.

APPENDIX A

Brief descriptions of some main subroutines are given in this appendix, and their listings are given in Appendix B.

A.1 Subroutine MULLER

This subroutine finds the poles and zeros of a network function by the Muller's method [11] of finding roots of a polynomial. The algorithm of this method is as follows:

Let

$$f_x = f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = 0$$

where the coefficients $a_0, a_1, \cdots, a_n$ are real numbers.

1. Let x_i, x_{i-1}, x_{i-2} be three approximations to a root z_1 of $f(x)$. To start choose $x_0 = -1$, $x_1 = 1$ and $x_2 = 0$, if better approximations are not known. Compute f_i, f_{i-1}, f_{i-2}.

2. Compute in order

$$h = x_i - x_{i-1}$$

$$\lambda_i = \frac{h}{x_{i-1} - x_{i-2}}$$

$$\delta_i = 1 + \lambda_i$$

3. Compute $g_i = f_{i-2}\lambda_i^2 - f_{i-1}\delta_i^2 + f_i(\lambda_i + \delta_i)$

4. Compute

$$\lambda_{i+1} = \frac{-2f_i\delta_i}{g_i \pm [g_i^2 - 4f_i\delta_i\lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i]^{1/2}}$$

choosing the sign so that the denominator will always have the largest magnitude.

5. Then $x_{i+1} = x_i + h\lambda_{i+1}$ is the next approximation.

6. Compute $f(x_{i+1}) = f_{i+1}$.

7. Repeat steps 1 to 6 until convergence based on either of the following criteria is satisfied:

(a) $\dfrac{|x_i - x_{i-1}|}{|x_i|} < \epsilon$

for prescribed ϵ, η

(b) $|f(x_i)| < \eta$

When one root z_1 has been found, then the degree of the polynomial is reduced by dividing the original polynomial by $(x-z_1)$ and the succeeding roots are obtained from the reduced polynomial.

A. 2 Subroutine FREQ

This subroutine calculates amplitude and delay characteristics of a network function for a given frequency by Equations (A. 1) and (A. 2) below:

$$F(s) = \frac{P}{Q} = \frac{m_1 + n_1}{m_2 + n_2}$$

$$= \frac{M + N}{m_2^2 - n_2^2}$$

where m_1, n_1 and m_2, n_2 are even and odd parts of the numerator and the denominator, respectively; and

$$M = m_1 m_2 - n_1 n_2$$

$$N = n_1 m_2 - m_1 n_2$$

Therefore

$$F(j\omega) = \left. \frac{M + N}{m_2^2 - n_2^2} \right|_{s = j\omega} \tag{A. 1}$$

Phase angle of $F(j\omega)$ is given by

$$-j \tan\theta = \left. \frac{N}{M} \right|_{s = j\omega} \quad \text{or} \quad -\theta = \tan^{-1}\left.\left(\frac{N}{jM}\right)\right|_{s = j\omega}$$

The delay time T_d is obtained by taking the derivative of $-\theta$ with respect to ω

$$T_d = -\frac{d\theta}{d\omega}$$

$$= \frac{M^2}{M^2 - N^2} \cdot \frac{d}{d\omega} \left.\left(\frac{N}{M}\right)\right|_{s = j\omega} \tag{A. 2}$$

$$= \left. \frac{MN' - NM'}{M^2 - N^2} \right|_{s = j\omega}$$

A. 3 Subroutine TIME1

This subroutine calculates, by Equation (A. 3), the step response for a given transfer function whose poles and zeros are already obtained by Subroutine MULLER.

Assuming that the transfer function has no multiple poles, let us express it as:

$$H(s) = \frac{P(s)}{Q(s)}$$

$$= \frac{(s-z_1)(s-z_2) \cdots (s-z_m)}{(s-p_1)(s-p_2) \cdots (s-p_n)}$$

Then the step response is obtained by taking the inverse Laplace transform of

$$R(s) = \frac{H(s)}{s}$$

as

$$r(t) = \sum_{j=0}^{n} k_j e^{p_j t} \qquad \text{(A. 3)}$$

where p_j and k_j are either real or complex. The residues k_j's are calculated by Subroutine RESDUE.

APPENDIX B

Listings of main subroutines used in this paper:

```
      SUBROUTINE MULLER (ZRO,N8,Z,KEY)
      DIMENSION ZRO(14), COE(14), Z(14,2)
      N1=N8-1
      IF (N1)1000,1000,1001
1000  WRITE (61,1119)
1119  FORMAT (//,20X,4HNONE)
      RETURN
1001  DO 1003 J=1, N8
      K=N8-J+1
1003  COE(J)=ZRO(K)
      N4=0
      I=N1+1
  19  IF (COE(I)) 9,7,9
   7  N4=N4+1
      Z(N4,1)=0.
      Z(N4,2)=0.
      I=I-1
      IF(N4-N1)19,37,19
   9  CONTINUE
  10  AXR=0.8
      AXI=0.
      L=1
      N3=1
      ALP1R=AXR
      ALP1I=AXI
      M=1
      GO TO 99
  11  BET1R=TEMR
      BET1I=TEMI
      AXR=0.85
      ALP2R=AXR
      ALP2I=AXI
      M=2
      GO TO 99
  12  BET2R=TEMR
      BET2I=TEMI
      AXR=0.9
      ALP3R=AXR
      ALP3I=AXI
      M=3
      GO TO 99
  13  BET3R=TEMR
      BET3I=TEMI
  14  TE1=ALP1R-ALP3R
      TE2=ALP1I-ALP3I
      TE5=ALP3R-ALP2R
      TE6=ALP3I-ALP2I
```

APPENDIX B (continued)

```
      TEM=TE5*TE5+TE6*TE6
      TE3=(TE1*TE5+TE2*TE6)/TEM
      TE4=(TE2*TE5-TE1*TE6)/TEM
      TE7=TE3+1.
      TE9=TE3*TE3-TE4*TE4
      TE10=2.*TE3*TE4
      DE15=TE7*BET3R-TE4*BET3I
      DE16=TE7*BET3I+TE4*BET3R
      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
      TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
      TE7=TE9-1.
      TE1=TE9*BET2R-TE10*BET2I
      TE2=TE9*BET2I+TE10*BET2R
      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
      TE15=DE15*TE3-DE16*TE4
      TE16=DE15*TE4+DE16*TE3
      TE1=TE13*TE13-TE14*TE14-4.*(TE11*TE15-TE12*TE16)
      TE2=2.*TE13*TE14-4.*(TE12*TE15+TE11*TE16)
      TEM=SQRTF(TE1*TE1+TE2*TE2)
      IF(TE1)113,113,112
113   TE4-SQRIF(.5*(TEM-TE1))
      TE3=.5*TE2/TE4
      GO TO 111
112   TE3=SQRTF(.5*(TEM+TE1))
      IF(TE2) 110,200,200
110   TE3=-TE3
200   TE4=.5*TE2/TE3
111   TE7=TE13+TE3
      TE8=TE14+TE4
      TE9=TE13-TE3
      TE10=TE14-TE4
      TE1=2.*TE15
      TE2=2.*TE16
      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205
204   TE7=TE9
      TE8=TE10
205   TEM=TE7*TE7+TE8*TE8
      TE3=(TE1*TE7+TE2*TE8)/TEM
      TE4=(TE2*TE7-TE1*TE8)/TEM
      AXR=ALP3R+TE3*TE5-TE4*TE6
      AXI=ALP3I+TE3*TE6+TE4*TE5
      ALP4R=AXR
      ALP4I=AXI
      M=4
      GO TO 99
15    N6=1
      IF(ABSF(HELL)+ABSF(BELL)-1.E-10)18,18,16
16    TE7=ABSF(ALP3R-AXR)+ABSF(ALP3I-AXI)
      IF(TE7/(ABSF(AXR)+ABSF(AXI))-1.E-5)18,18,17
```

APPENDIX B (continued)

```
17      N3=N3+1
        ALP1R=ALP2R
        ALP1I=ALP2I
        ALP2R=ALP3R
        ALP2I=ALP3I
        ALP3R=ALP4R
        ALP3I=ALP4I
        BET1R=BET2R
        BET1I=BET2I
        BET2R=BET3R
        BET2I=BET3I
        BET3R=TEMR
        BET3I=TEMI
        IF(N3-40)14,18,18
18      N4=N4+1
        Z(N4,1)=ALP4R
        Z(N4,2)=ALP4I
        N3=0
        IF(N4-N1)30,37,37
37      IF(KEY) 998,998,999
999     WRITE (61,555)
555     FORMAT (//,7X,9HREAL PARK,9X,9HIMAG PART, 13X,9HREAL PART, 9X,
       19HIMAG PART,, 13X,9HREAL PART, 9X,9HIMAG PART)
        WRITE (61,666) (Z(NT,1),Z(NT,2), NT=1,N1)
666     FORMAT (F19.6,F18.6,F22.6,F18.6,F22.6,F18.6)
998     RETURN
30      IF(ABSF (Z(N4,2))-1.E-4)10,10,31
  31    GO TO (32,10),L
  32    AXR=ALP1R
        AXI=-ALP1R
        ALP1I=-ALP1I
        M=5
        GO TO 99
  33    BET1R=TEMR
        BET1I=TEMI
        AXR=ALP2R
        AXI=-ALP2I
        ALP2I=-ALP2I
        M=6
        GO TO 99
  34    BET2R=TEMR
        BET2I=TEMI
        AXR=ALP3R
        AXI=-ALP3I
        ALP3I=-ALP3I
        L=2
        M=3
99      TEMR=COE(1)
        TEMI=0.0
        DO 100 I=1,N1
        TE1= TEMR*AXR-TEMI*AXI
        TEMI=TEMI*AXR+TEMR*AXI
```

APPENDIX B (continued)

```
100 TEMR=TE1+COE(I+1)
    HELL=TEMR
    BELL=TEMI
    IF(N4)102, 103, 102
102 DO101I=1, N4
    TEM1=AXR-Z(I, 1)
    TEM2=AXI-Z(I, 2)
    TE1=TEM1*TEM1+TEM2*TEM2
    TE2=(TEMR*TEM1+TEMI*TEM2)/TE1
    TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1
101 TEMR=TE2
103 GO TO (11, 12, 13, 15, 33, 34), M
    END
    SUBROUTINE FREQ (NA, NB, A, B, G, W)
    DIMENSION A(14), AE(7), AO(7), AE1(7), AO1(7), B(14)
   1, BE(7), BO(7), BE1(7), BO1(7), G(4)
    CALL PARTS (NA, A, M1, AE, N1, AO, M11, AE1, N11, AO1)
    CALL PARTS (NB, B, M2, BE, N2, BO, M12, BE1, N12, BO1)
    EVN=SUM (M1, AE, W)
    ODDN=W*SUM (N1, AO, W)
    EV1N=W*SUM (M11, AE1, W)
    ODD1N=SUM (N11, AO1, W)
    EVD=SUM  (M2, BE, W)
    ODDD=W*SUM (N2, BO, W)
    EV1D=W*SUM (M12, BE1, W)
    ODD1D=SUM (N12, BO1, W)
    TOP=EVN*EVN+ODDN*ODDN
    BOTTOM=EVD*EVD+ODDD*ODDD
    AMPL=SQRT (TOP/BOTTOM)
    DELAY=(EVD*ODD1D+ODDD*EV1D)/BOTTOM-(EVN*ODD1N+ODDN*EV1N)/TOP
    G(1)=W
    G(2)=AMPL
    G(3)=DELAY
    RETURN
    END
    SUBROUTINE PARTS (NA, A, MK, AE, NK, AO, M1K, AE1, N1K, AO1)
    DIMENSION A(14), AE(7), AO(7), AO(7), AE1(7), AO1(7)
    I=1
    MK=1
    NK=0
    MIK=0
    AE(1)=A(1)
    IF(NA-I)3, 3, 1
1   I=I+1
    NK=NK+1
    AO(NK)=A(I)
    DUMMY=I-1
    AO1(NK)=DUMMY*A(I)
    IF(NA-I)3, 3, 2
2   M1K=MK
```

APPENDIX B (continued)

```
      MK=MK+1
      I=I+1
      AE(MK)=A(I)
      DUMMY=I-1
      AE1(M1K)=DUMMY*A(I)
      IF(NA-I)3,3,1
    3 N1K=NK
      RETURN
      END
      FUNCTION  SUM (N,A,W)
      DIMENSION A(14)
      SUM=0.0
      IF(N)3,3,2
    2 SUM=A(N)
      IF(N-1)3,3,4
    4 X=-W*W
      NM1=N-1
      DO 5 I=1,NM1
      K=N-I
    5 SUM=SUM *X+A(K)
    3 RETURN
      END

      SUBROUTINE TIME 1(NZ,NP,Z,P,KEY,G1,G2,LA,AA,T)
      DIMENSION Z(14,2),P(14,2),G1(100),G2(100),RRES(14),RPOLE(14)
     1,CRES(7),RCPOLE(7),QCPOLE(7),THETA(7),DEG(7)
      IF(KEY-1)200,200,201
C     FIRST TIME THRU, ADD POLE AT S=0
  200 KEY=2
      NP=NP+1
      P(NP,1)=0.
      P(NP,2)=0.
      DO 11 J=1,NP
      PR=P(J,1)
      PI=P(J,2)
      JJ=J+1
      IF(JJ-NP)117,117,11
  117 DO 8 K=JJ,NP
C     FIND CONJUGATE POLE
      IF (ABSF (PR-P(K,1))+ABSF (PI+P(K,2))-.00001)9,9,8
    9 P(K,1)=P(J+1,1)
      P(K,2)=P(J+1,2)
      P(J+1,1)=P(J,1)
      P(J+1,2)=-ABSF(P(J,2))
      P(J,2)=-P(J+1,2)
    8 CONTINUE
   11 CONTINUE
      NREAL=0
      NCOMP=0
      DO 21 J=1,NP
      KK=J
```

APPENDIX B (continued)

```
      PR=P(J,1)
      PI=P(J,2)
C     IF CONJUGATE POLE WITH NEG. IMAG. PART, IGNORE
      IF (PI+.00001)21,22,22
 22   CALL RESDUE (Z, P,NZ,NP,RPSAVE,QPSAVE,PR,PI,KK,AA)
C     REAL OR COMPLEX POLE
      IF(ABSF(PI)-.00001)23,23,24
 23   NREAL=NREAL+1
      RRES(NREAL)=RPSAVE
      RPOLE(NREAL)=PR
      GO TO 21
 24   NCOMP=NCOMP+1
      CRES(NCOMP)=2.*SQRTF(RPSAVE**2+QPSAVE**2)
      RCPOLE(NCOMP)=PR
      QCPOLE(NCOMP)=PI
      THETA(NCOMP)=ATANF(QPSAVE/RPSAVE)
      IF(RPSAVE)26,27,27
 26   THETA(NCOMP)=THETA(NCOMP)+3.14159
 27   DEG(NCOMP)=57.295779*THETA(NCOMP)
 21   CONTINUE
C     CALCULATE OUTPUT
 201  LA=LA+1
      G1(LA)=T
      G2(LA)=0.
      IF(NREAL)204,204,203
 203  DO 205 J=1,NREAL
 205  G2(LA)=G2(LA)+RRES(J)*EXPF(RPOLE(J)*T)
 204  IF(NCOMP)206,206,207
 207  DO 208 J=1,NCOMP
 208  G2(LA)=G2(LA)+CRES(J)*EXPF(RCPOLE(J)*T)*COSF(QCPOLE(J)*T+THETA(J))
 206  RETURN
      END

      SUBROUTINE RESDUE (Z,P,NZ,NP,R2,Q2,PR,PI,KK,AA)
      DIMENSION Z(14,2),P(14,2)
      Q2=0.
      R2=AA
      DO 1 K=1,NP
      IF(KK-K)3,2,3
 2    RD=1.
      QD=0.
      GO TO 13
 3    RD=PR-P(K,1)
      QD=PI-P(K,2)
 13   IF(K-NZ)5,5,4
 4    R1=1.
      Q1=0.
      GO TO 15
 5    R1=PR-Z(X,1)
      Q1=PI-Z(K,2)
```

APPENDIX B (continued)

```
15  A=R1*R2-Q1*Q2
    Q2=R1*Q2+R2*Q1
    R2=A
    A=RD**2+QD**2
    B=(QD*Q2+RD*R2)/A
    Q2=(-QD*R2+RD*Q2)/A
1   R2=B
    RETURN
    END

    SUBROUTINE MIN(F,G,X,N,ERROR,JUMP,NITER, ALFSAV, ALFMUL)
    DIMENSION G(14),X(14),GSAVE(14),S(14),SIG(14),DG(14),
   1A(14,14),B(14,14),C(14,14),XSAVE(14),H(14,14)
    IF(JUMP)20,1,20
1   KEY=1
    JUMP=1
    RETURN
20  GO TO (18, 17, 36, 36, 19),KEY
18  FSAVE=F
    ITER=NITER
    DO 14 J=1,N
    XSAVE(J)=X(J)
14  GSAVE(J)=G(J)
    IMAX=1
    KEY=2
    F1=FSAVE
    F2=FSAVE
    X1=0.
    X2=0.
    IF(JUMP)58,58,56
56  DO 12 J=1,N
    DO 13 K=1,N
13  H(J,K)=0.
12  H(J,J)=1.
    JUMP=-1
58  CALL MATMUL(H,N,N,G,1,S,-1.,14,1,14)
    CALL MATMUL (S,1,N,G,1, DEM,1.,1,1,1)
    IF(DEM) 5,56,56
5   ALF=ALFSAV
    ALFSAV=0.
    GO TO 15
17  IF(F) 44,44,31
44  ALFSAV=ALFSAV-ALF
    ALF=ALF/ALFMUL**4
    GO TO 15
31  IF((F-F2)/F-.00001) 28,28,3
3   GO TO (4,30),IMAX
4   ALFSAV=ALFSAV/(ALFMUL)**2
    GO TO 5
28  F1=F2
    F2=F
```

APPENDIX B (continued)

```
      X1=X2
      X2=ALFSAV
      IMAX=2
      ALF=ALFMUL*ALF
      GO TO 15
   30 TEMP=X2
      X2=ALFSAV
      ALFSAV=TEMP
      TEMP=F2
      F2=F
      F=TEMP
      PRINT 250
  250 FORMAT (1X,1HQ)
   36 CALL QUAD(X1,X2,ALFSAV,ITER,KEY,F,F1,F2,ERROR)
      IF(KEY-5)27,99,99
   99 JUMP=1
      GO TO 27
   15 ALFSAV=ALFSAV+ALF
   27 DO 21 J=1,N
   21 X(J)=XSAVE(J)+ALFSAV*S(J)
      IF(JUMP-1)26,81,26
   81 PRINT 24
   24 FORMAT (1X,1HG)
   26 RETURN
   19 IF(ABSF(F/FSAVE)-ERROR)11,10,10
   10 JUMP=0
      RETURN
   11 DO 7 J=1,N
      SIG(J)=ALFSAV*S(J)
    7 DG(J)=G(J)-GSAVE(J)
      CALL MATMUL(SIG,1,N,DG,1,DEM,1.,1,1,1)
      CALL MATMUL (SIG,N,1,SIG,N,A,DEM,1,1,14)
      CALL MATMUL (DG,1,N,H,N,B,1.,1,14,14)
      CALL MATMUL (B,1,N,DG,1,DEM,1.,14,1,1)
      CALL MATMUL (DG,N,1,B,N,C,1.,1,14,14)
      CALL MATMUL (H,N,N,C,N,B,DEM,14,14,14)
      DO 8 J=1,N
      DO 8 K=1,N
    8 H(J,K)=H(J,K)+A(J,K)-B(J,K)
      JUMP=-1
      GO TO 18
      END

      SUBROUTINE MATMUL (A,N,M,B,LL,C,DIV,NROW1,NROW2,NROW3)
      DIMENSION A(1),B(1),C(1)
      DO 2 L=1,LL
      DO 2 J=1,N
      JJ=J+NROW3*(L-1)
      C(JJ)=0.
      DO 1 K=1,M
      JK=J+(K-1)*NROW1
```

```
      KL=K+(L-1)*NROW2
    1 C(JJ)=C(JJ)+A(JK)*B(KL)
    2 C(JJ)=C(JJ)/DIV
      RETURN
      END

      SUBROUTINE QUAD(XA,XC,XMIN,ITER,KEY1,F,FA,FC,ERROR)
C     THIS SUBROUTINE FINDS THE ONE DIMENSIONAL MIN
C     POINT BY QUADRATIC FITTING.
      KEY=KEY1-1
      GO TO (100,300,100,100,100,100),KEY
  100 XAS=XA**2
      XCS=XC**2
      XB=XMIN
      XBS=XB**2
      FB=F
      J=0
   21 KEY1=KEY1+1
    2 IF(KEY1-4)4,23,23
   23 XMIN=XB
      KEY1=5
      F=FB
      RETURN
    4 XD=.5*((XBS-XCS)*FA+(XCS-XAS)*FB+(XAS-XBS)*FC)
      XD=XD/((XB-XC)*FA+(XC-XA)*FB+(XA-XB)*FC)
      XDS=XD**2
      XMIN=XD
      FS=FB
      RETURN
  300 FD=F
      IF(FD-FB)3,3,19
   19 IF(XD-XB)90,91,91
   90 CALL ARR(XA,XAS,FA,XD,XDS,FD)
      GO TO 11
   91 CALLARR(XC,XCS,FC,XD,XDS,FD)
      GO TO 11
    3 IF(XD-XB)9,9,8
    8 CALL ARR(XA,XAS,FA,XB,XBS,FB)
      GO TO 11
    9 CALL ARR(XC,XCS,FC,XB,XBS,FB)
   11 IF(FD-FB)10,18,18
   10 CALL ARR (XB,XBS,FB,XD,XDS,FD)
   18 J=J+1
      IF(J-ITER)32,21,2
   32 IF((ABS(FS-F))/F-.1*(1.-ERROR))33,33,4
   33 IF(J-1)4,4,21
      END
      SUBROUTINE ARR(A1,A2,A3, B1,B2,B3)
      A1=B1
      A2=B2
      A3=B3
      RETURN
      END
```