

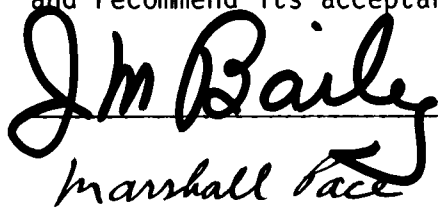
To the Graduate Council:

I am submitting herewith a thesis written by Cheng-Chiian Liu entitled "Algebraic Methods of Analysis and Synthesis of Dynamical Systems." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.



James C. Hung, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



Vice Chancellor
Graduate Studies and Research

Thesis
80
.L58
cop.2

ALGEBRAIC METHODS OF ANALYSIS
AND SYNTHESIS OF DYNAMICAL SYSTEMS

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Cheng-Chiian Liu

March 1980

111625

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to his advisor, Dr. J. C. Hung, for the guidance and encouragements he received during his study at The University of Tennessee, Knoxville. Thanks are due to Mr. Pu-Yuan Chou for his many inspiring discussions, and to Mr. Stephen T. Hung for his careful review of the manuscripts.

The author also wishes to acknowledge the support of the government of the Republic of China for his study.

Finally, the author is grateful to his family, particularly, his wife, for their patience and understanding of his long periods of absence from home during his study abroad.

ABSTRACT

This thesis presents the algebraic techniques for the analysis of dynamical systems. Methods for both "interval-analysis" and "point-analysis" are included. The main concern of the thesis is the introduction of the "block-pulse" method. Applications of the method to the solution of differential equations, to the synthesis of optimal control, and to the realization of Kalman filtering are described in detail.

TABLE OF CONTENTS

CHAPTER		PAGE
1	INTRODUCTION	1
2	BLOCK-PULSE ANALYSIS	4
	2-1 Block-Pulse Functions	4
	2-2 Block-Pulse Series Representation of a Function	6
	2-3 Relationship Between Block-Pulse Functions to Walsh and Haar Functions	10
	2-4 Block-Pulse Integration Technique	12
3	FIVE ADDITIONAL METHODS OF ANALYSIS	17
	3-1 Modified Block-Pulse Method	17
	3-2 Trapezoidal Rule	17
	3-3 Modified Trapezoidal Rule	19
	3-4 Runge-Kutta Method	20
	3-5 Euler's Method.	20
4	SOLUTION TO DIFFERENTIAL EQUATIONS	21
	4-1 Linear Time-Invariant Differential Equation	21
	4-2 Linear Time-Varying Differential Equation	28
5	BLOCK-PULSE METHOD FOR OPTIMAL CONTROL	37
	5-1 Problem Formulation	37
	5-2 Block-Pulse Solution to Transition Matrix	39
	5-3 Example	40
6	APPLICATION TO KALMAN FILTERING.	47
	6-1 Kalman Filter	47
	6-2 Solution of Riccati Equation by Transformation	48
	6-3 Direct Solution to Riccati Equation	55

CHAPTER	PAGE
7 CONCLUSION	60
REFERENCES	62
APPENDIXES	64
Appendix 1	65
Appendix 2	68
VITA	74

LIST OF FIGURES

FIGURE		PAGE
2-1	Block-Pulse Functions and Block-Pulse Matrix	5
2-2	Integration of Block-Pulse Functions and Integration Matrix.	7
2-3	Walsh Functions and Walsh Matrix	11
2-4	Haar Functions and Haar Matrix.	13
3-1	Algorithm of Modified Block-Pulse Method	18
4-1	Comparison of Interval-Analysis for Time- Invariant Equation.	26
4-2	Comparison of Point-Analysis for Time- Invariant Equation.	27
4-3	Comparison of Interval-Analysis for Time- Varying Equation	33
4-4	Comparison of Point-Analysis for Time-Varying Equation	34
5-1	Closed-Loop Implementation of the Optimal Control . . .	42
5-2	Optimal Feedback Gains K_k	43
5-3	Comparison of the Optimal Feedback Signal u^*	45
5-4	Comparison of Closed-Loop System Responses.	46
6-1	Comparison of Error Covariance Matrix Values	56

CHAPTER 1

INTRODUCTION

Recently, several methods for the analysis and synthesis of dynamic systems by means of piece-wise constant functions have been proposed. There are two primary merits of using piece-wise constant functions. First, it allows the solution of differential equation by algebraic means, and second, the hardware for generating piece-wise signal is often simpler than that for generating other shapes of signal. Chen and Hsiao proposed the use of Walsh functions, piece-wise constant in nature, for system analysis [1]. Following the similar concept, Sannuti proposed the use of block-pulse functions, also piece-wise constant in nature, for the same purpose [2]. As pointed out by Sannuti, there is a one-to-one relationship between an orthogonal set of Walsh functions and an orthogonal set of block-pulse functions. Therefore, the analysis using either set will give the same result. In fact, any signal represented by a set of Walsh functions may be converted to block-pulse representation using a matrix transformation. The converse, however, is not always possible because the one-to-one correspondence is from all Walsh sets "into" all block-pulse sets. For system synthesis applications, block-pulse functions are easier to be generated by hardware than their Walsh function counterparts, a merit for the former. The same argument applies to the relationship between a set of Haar functions and block-pulse functions.

Piece-wise constant functions are used for "interval analysis" where, over a time interval, the value of the function is represented by a constant. Recognizing that Sannuti's block-pulse function

technique is strongly connected to the trapezoidal integration rule, Shieh, Yeung and McInnis devised a method to obtain a "point-analysis" where the value of the function at certain time instants are evaluated. Their method makes use of the result of the block-pulse analysis and a technique which amounts to a reverse use of the trapezoidal rule. It gives much more accurate analysis results at specific time instants.

One cannot help but wonder how these methods compare to the classical methods used by numerical analysts [4,5]. For example, how does the combination of block-pulse analysis and the inverse-trapezoidal rule compare to Runge-Kutta integration method or the trapezoidal integration method for point-analysis? Also, how does the block-pulse analysis compare to other interval-analysis methods? This paper presents the result of a study aimed at answering these questions.

Analysis using block-pulse functions is the main concern of this thesis. In Chapter 2, block-pulse functions are defined, and techniques of using block-pulse functions for analysis are presented. The relationship of block-pulse functions to Walsh functions and Haar functions are also given.

Five additional methods of analysis which are related to the block-pulse analysis are discussed in Chapter 3.

There are three areas where the block-pulse method will be applied; namely, the solution to differential equations, synthesis of optimal control, and the realization of continuous-time Kalman filter. The solution of differential equations will be described in Chapter 4 by the block-pulse method and the five other methods. Comparisons of analysis accuracy and computation efforts will be made. Chapter 5 applies the block-pulse method to the synthesis of linear optimal

control. Chapter 6 presents the use of block-pulse method in Kalman filtering. Examples are given in various chapters for illustration and comparison. Chapter 7 concludes the thesis. Two computer programs are included in the Appendix.

CHAPTER 2

BLOCK-PULSE ANALYSIS

2-1 Block-Pulse Functions

An orthogonal set of "block-pulse functions" is one of many orthogonal sets of piece-wise constant functions. An m -element set of orthogonal block-pulse functions can be defined over a normalized time interval $[0,1]$. For each integer i , $1 \leq i \leq m$, the function ϕ_i is given by

$$\phi_i(t) = \begin{cases} 1, & \frac{i-1}{m} \leq t \leq \frac{i}{m} \\ 0, & \text{otherwise} \end{cases} \quad (2-1)$$

Figure 2-1 depicts a set of block-pulse functions for $m = 4$.

Any set of block-pulse functions can conveniently be represented by a $m \times m$ matrix Φ called "block-pulse matrix" as follows:

$$\Phi = \frac{1}{m} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (2-2)$$

where Φ for $m = 4$ is shown in Figure 2-1, also.

Define a vector block-pulse function to be an $m \times 1$ vector $\underline{\phi}(t)$ whose elements are $\phi_i(t)$, $i = 1$ to m . Note that m is therefore the dimension of the space of block-pulse functions. The time integral of $\underline{\phi}(t)$, denoted by $\underline{p}(t)$, can be approximated by a matrix multiplication as

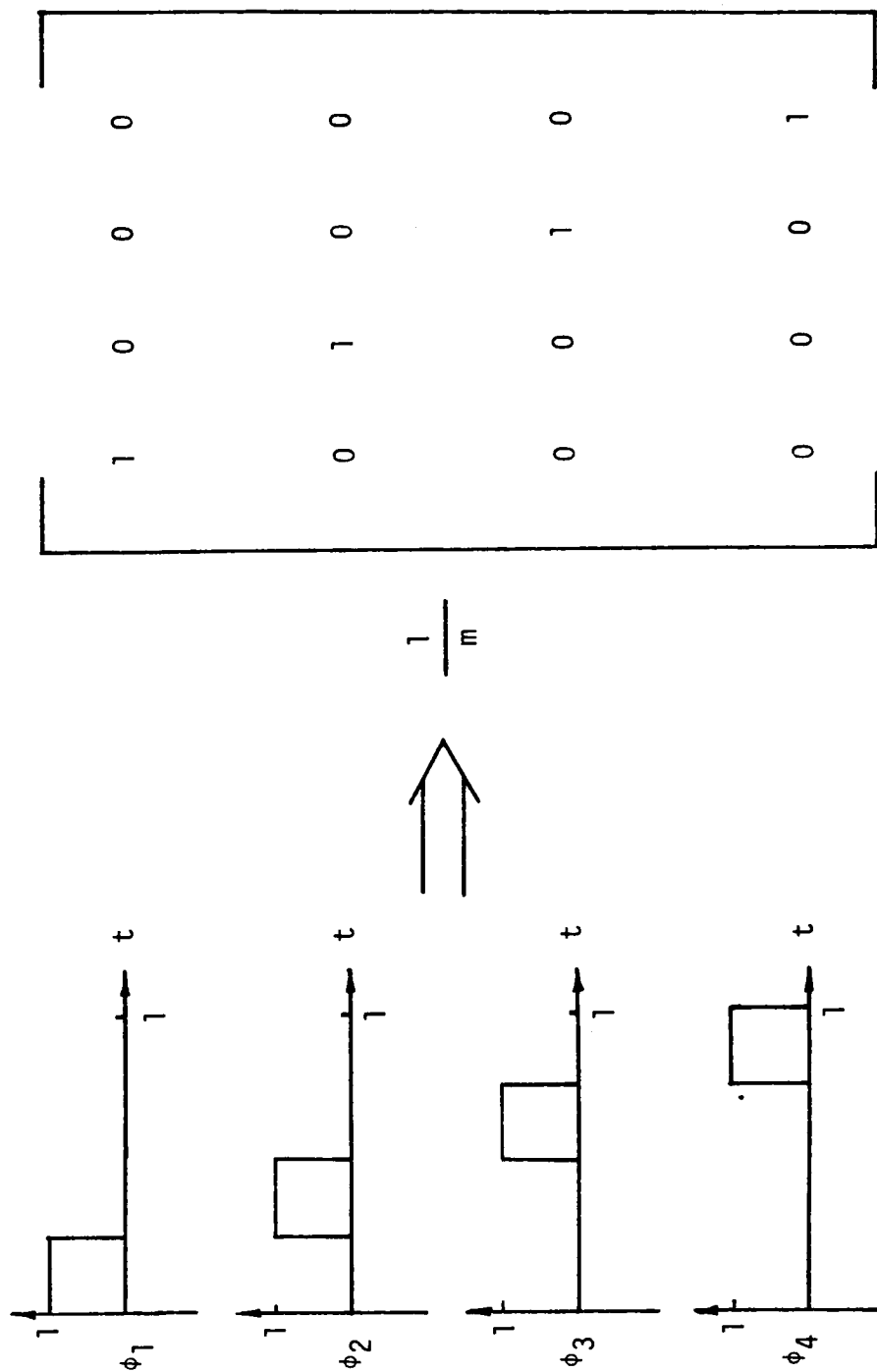


Figure 2-1. Block-Pulse Functions and Block-Pulse Matrix ($m=4$)

$$\underline{p}(t) = \int_0^t \underline{\phi}(t') dt' = B \underline{\phi}(t) \quad (2-3)$$

where

$$B = \frac{1}{m} \begin{bmatrix} \frac{1}{2} & 1 & 1 & \cdots & 1 \\ 0 & \frac{1}{2} & 1 & \cdots & 1 \\ 0 & 0 & \frac{1}{2} & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & \frac{1}{2} \end{bmatrix} \quad (2-4)$$

The set of $P_i(t)$, $i = 1, 2, \dots, m$, and the integration matrix B , for $m = 4$, are shown in Figure 2-2.

2-2 Block-Pulse Series Representation of a Function

A scalar function $f(t)$ which is integrable in $[0,1]$ can be approximated as

$$f(t) = \sum_{i=1}^m f_i \phi_i(t) \quad (2-5)$$

where values of f_i are chosen to minimize the integral square error of the approximation. The integral square error is

$$e = \int_0^1 \left[f(t) - \sum_{i=1}^m f_i \phi_i(t) \right]^2 dt \quad (2-6)$$

Setting $\frac{\partial e}{\partial f_j} = 0$ and solving for f_j , gives

$$f_j = \frac{\int_0^1 f(t) \phi_j(t) dt}{\int_0^1 \phi_j^2(t) dt} \quad (2-7)$$

= mean of $f(t)$ over the j th time interval $\frac{1}{m}$.

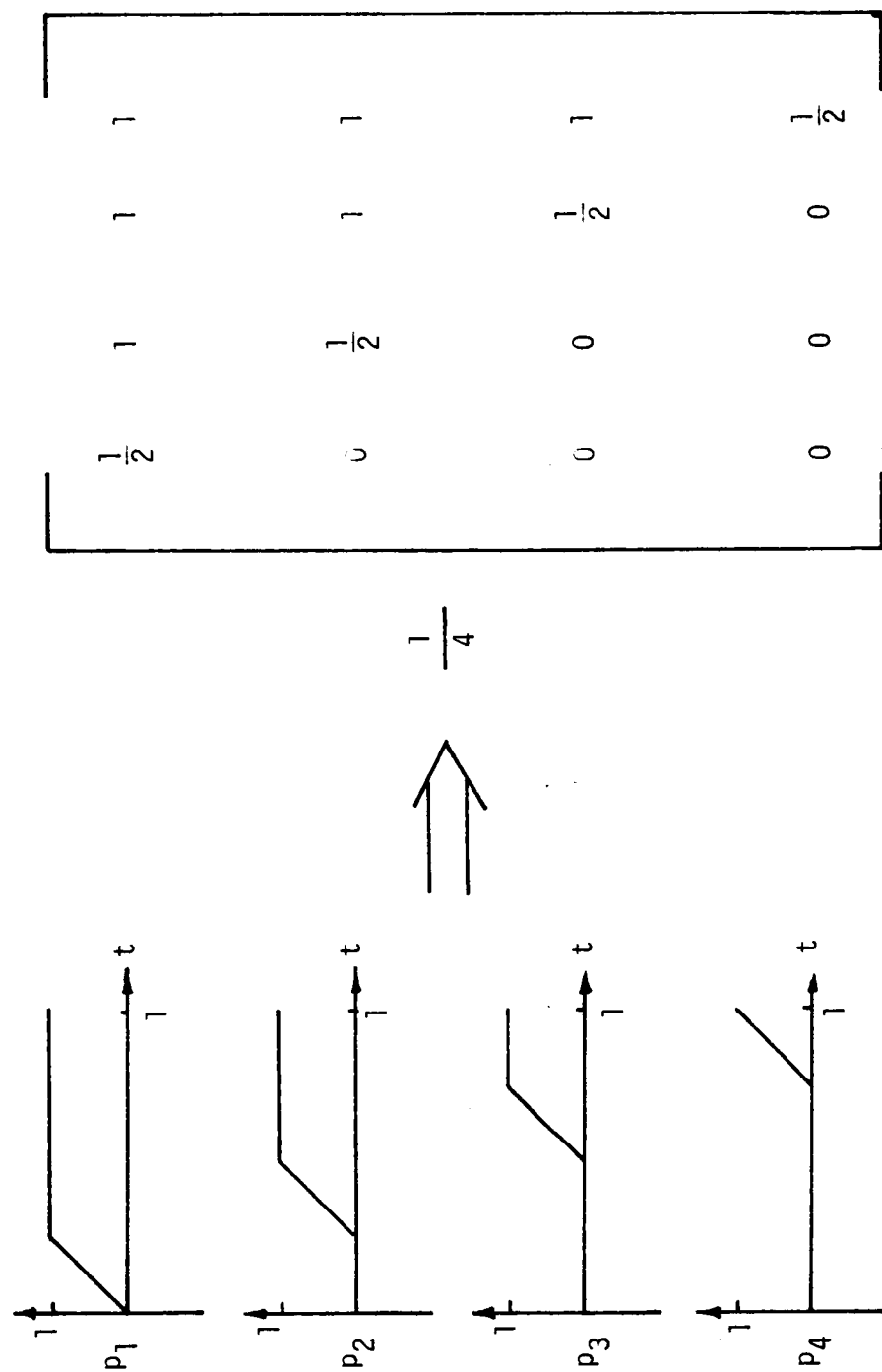


Figure 2-2. Integration of Block-Pulse Functions and Integration Matrix ($m=4$)

Equation (2-5) can be expressed in the matrix form as

$$f(t) = \underline{f}^T \underline{\phi} \quad (2-8)$$

where

$$\underline{f}^T = [f_1 f_2 \cdots f_m] \quad (2-9)$$

Let $F(t)$ be a matrix and let

$$F_i = m \int_{\frac{i-1}{m}}^{\frac{i}{m}} F(t) dt \quad (2-10)$$

Define the following special matrix notation

$$[[F_1 \ F_2 \ \cdots \ F_m]] \underline{\phi} = [\phi_1 F_1 \ \phi_2 F_2 \ \cdots \ \phi_m F_m] \quad (2-11)$$

where F_i are matrices and ϕ_i are scalars. The notation $[[\]]$ means that during the matrix multiplication each F_i is treated as a single element. This special notation is necessary because the expression

$$[F_1 \ F_2 \ \cdots \ F_m] \underline{\phi}$$

is not correct when each F_i itself is a matrix, because the dimensions of $[F_1 \ F_2 \ \cdots \ F_m]$ and $\underline{\phi}$ are not consistent.

The block-pulse representation of $F(t)$ is

$$\begin{aligned} F(t) &= \sum_{i=1}^m F_i \phi_i \\ &= [[F_1 \ \cdots \ F_m]] \underline{\phi} \\ &= \underline{\phi}^T \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix} \end{aligned} \quad (2-12)$$

Block-pulse representation of $A(t)$ $B(t)$ is given by

$$\begin{aligned}
 A(t) B(t) &= \begin{bmatrix} A_1 & A_2 & \cdots & A_m \end{bmatrix} \underline{\phi} \underline{\phi}^T \begin{bmatrix} B_1 \\ B_1 \\ \vdots \\ B_m \end{bmatrix} \\
 &= \begin{bmatrix} A_1 & A_2 & \cdots & A_m \end{bmatrix} \begin{bmatrix} \phi_1 & 0 & \cdots & 0 \\ 0 & \phi_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \phi_m \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_m \end{bmatrix} \\
 &= \begin{bmatrix} A_1 & A_2 & \cdots & A_m \end{bmatrix} \begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & B_m \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_m \end{bmatrix} \\
 &= \begin{bmatrix} A_1 B_1 & A_2 B_2 & \cdots & A_m B_m \end{bmatrix} \underline{\phi} \quad (2-13)
 \end{aligned}$$

where A_i and B_i are related to $A(t)$ and $B(t)$ as F_i is to $F(t)$ as shown in Equation (2-12).

Similarly, for three $A(t)$, $G(t)$, and $B(t)$, one has

$$A(t)G(t)B(t) = \begin{bmatrix} A_1 G_1 B_1 & A_2 G_2 B_2 & \cdots & A_m G_m B_m \end{bmatrix} \underline{\phi} \quad (2-14)$$

Very often the integrals given by Equations (2-7) and (2-10) can be approximated, respectively, by the following equations:

$$f_i = \frac{f(\frac{i}{m}) + f(\frac{i-1}{m})}{2} \quad (2-15)$$

$$F_i = \frac{F(\frac{i}{m}) + F(\frac{i-1}{m})}{2} \quad (2-16)$$

For any arbitrary time interval $[0, T]$, Equation (2-7) should be time scaled by replacing one by $\frac{m}{T}$.

2-3 Relationship Between Block-Pulse Functions to Walsh and Haar Functions

Relationship to Walsh Functions. Any set of Walsh functions can be related to a set of block-pulse functions by a nonsingular linear transformation. Consider the Walsh-ordered Walsh function set for $m = 4$, as shown in Figure 2-3. The associated Walsh matrix is:

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (2-17)$$

This matrix is related to a 4×4 block-pulse matrix Φ through the nonsingular transformation. That is,

$$W = T_W \Phi \quad (2-18)$$

where

$$T_W = mW$$

Walsh series representation of a function is similar, in concept, to that of block-pulse series representation. In the case of Walsh series, each coefficient of the series is a function of all coordinates in the orthogonal data space, where as in the block-pulse series each coefficient is a function of only one coordinate in the data space.

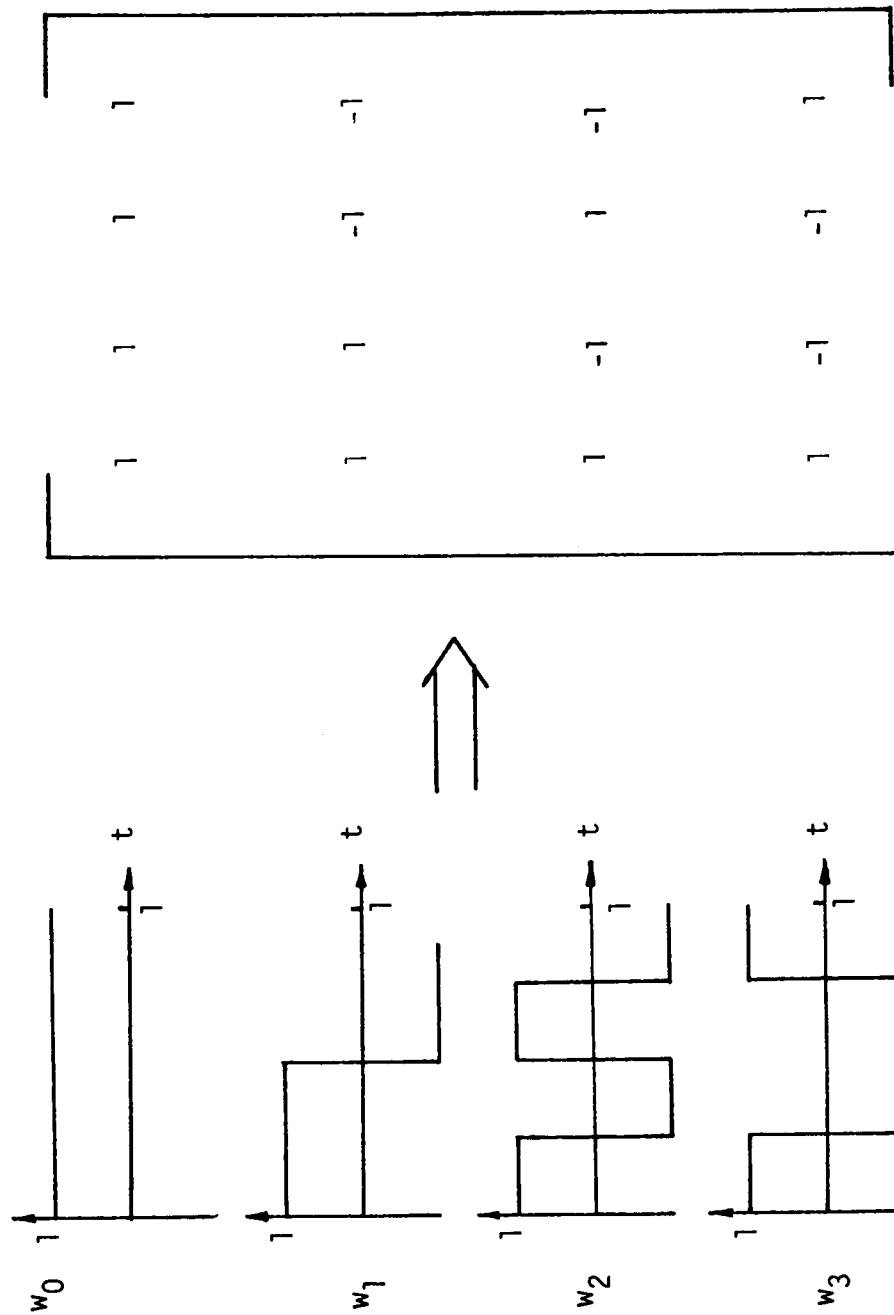


Figure 2-3. Walsh Functions and Walsh Matrix ($m=4$)

Relationship to Haar Functions. Any set of Haar functions can be related to a set of block-pulse functions by a nonsingular linear transformation. Consider Haar functions set for $m = 4$, as shown in Figure 2-4, the associated Haar matrix is:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \quad (2-19)$$

This matrix is related to a 4×4 block-pulse matrix ϕ through a nonsingular transformation

$$H = T_h \phi \quad (2-20)$$

where

$$T_h = mH$$

2-4 Block-Pulse Integration Technique

Block-pulse representation leads to a simple algebraic method for the solution of a set of differential equations. The method is applicable to linear as well as nonlinear differential equations, time-invariant as well as time-varying. In this section only linear cases will be discussed.

Time-Invariant Differential Equation. Consider a time-invariant vector differential equation

$$\dot{\underline{x}} = A \underline{x} + B u, \quad \underline{x}(0) = \underline{x}_0 \quad (2-21)$$

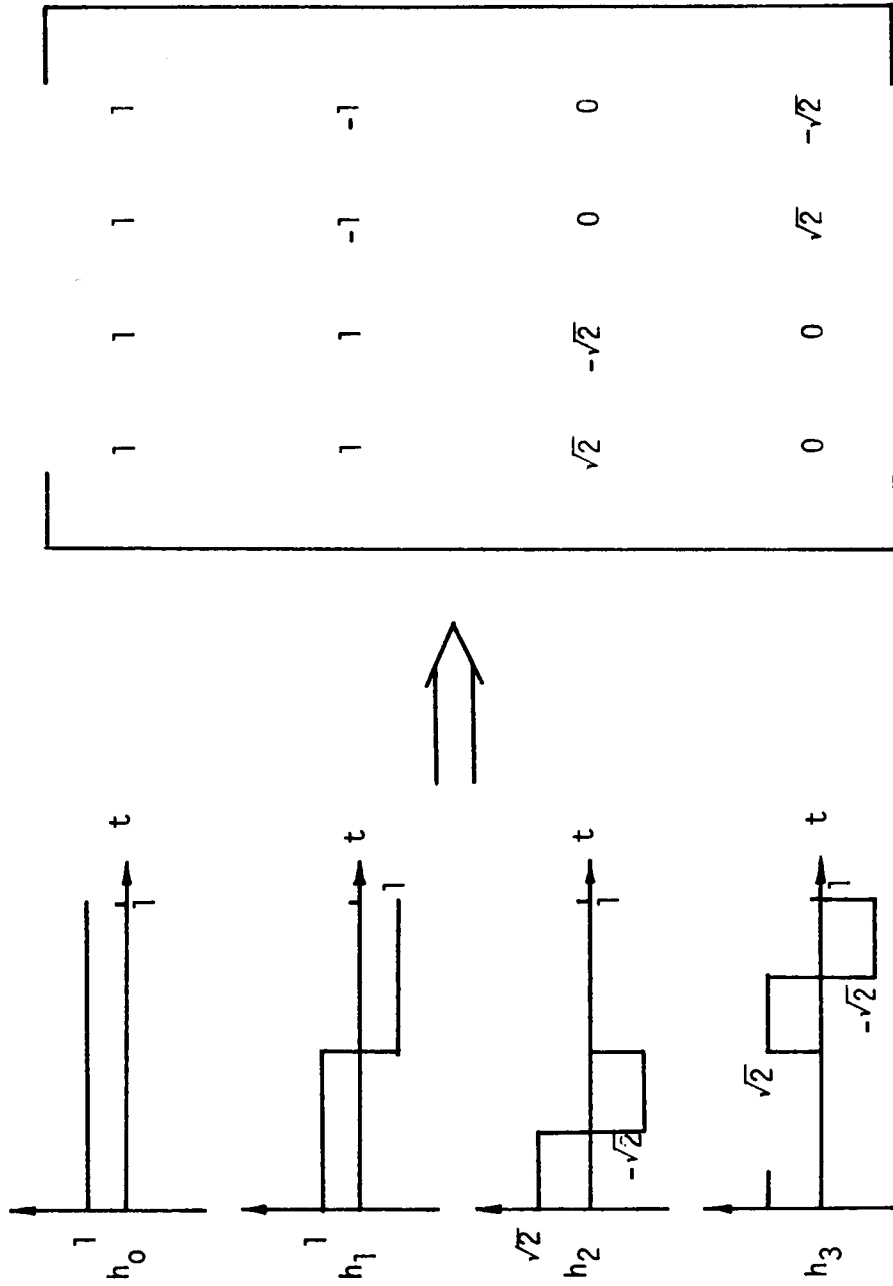


Figure 2-4. Haar Functions and Haar Matrix ($m=4$)

For a chosen m dimensional set of block-pulse functions, resolving $\dot{\underline{x}}$, $\underline{x}$, $\underline{Bu}$, $\underline{x}_0$ into their respective block-pulse series approximations,

$$\left. \begin{aligned} \dot{\underline{x}} &= C\underline{\phi} \\ \underline{x} &= D\underline{\phi} \\ \underline{Bu} &= E\underline{\phi} \\ \underline{x}_0 &= X_0\underline{\phi} \end{aligned} \right\} \quad (2-22)$$

where C , D , E and X_0 are $m \times m$ matrices. Since

$$\int_0^t \dot{\underline{x}} dt' = \underline{x} - \underline{x}_0$$

the use of Equations (2-3) and (2-22) in this expression will give

$$C\underline{P}\underline{\phi} = D\underline{\phi} - X_0\underline{\phi}$$

Equating coefficient matrices at both sides yields

$$C\underline{P} = D - X_0 \quad (2-23)$$

Applying Equation (2-22) to Equation (2-21) gives

$$C = A\underline{D} + E \quad (2-24)$$

The term $C\underline{P}$ in Equation (2-23) can be expressed in terms of C . Then, by using a column vector comparison technique, C can be eliminated from Equations (2-23) and (2-24). Letting $\underline{d}_k$ and $\underline{e}_k$ be column vectors of D and E , we have

$$\left. \begin{aligned} \underline{d}_1 &= Q\underline{x}_0 + \frac{1}{2m} Q \underline{e}_1 \\ \underline{d}_k &= R\underline{d}_{k-1} + \frac{1}{2m} Q(\underline{e}_k + \underline{e}_{k-1}) \end{aligned} \right\} \quad (2-25)$$

$$\left. \begin{aligned} Q &= [I - \frac{1}{2m} A]^{-1} \\ R &= Q [I + \frac{1}{2m} A] \end{aligned} \right\} \quad (2-26)$$

Thus the solution of Equation (2-12) is given by

$$\underline{x}(t) = \sum_{k=1}^m \phi_k(t) \underline{d}_k \quad (2-27)$$

which is piece-wise constant due to the nature of $\phi_k(t)$.

Time-Varying Differential Equation. For a time-varying differential equation

$$\dot{\underline{x}} = A(t) \underline{x} + B(t)u, \quad \underline{x}(0) = \underline{x}_0 \quad (2-28)$$

The matrix functions $A(t)$ and $B(t)$ are first resolved into their block-pulse function representations and then are multiplied to the block-pulse representations of $\underline{x}$ and u , respectively. However, direct use of this approach results in unwieldy expressions. Recall that each block-pulse coefficient, for a given function, represents the average value of that function in the elementary interval concerned. All one needs, therefore, are the average values of $A(t)$ and $B(t)$ in each computation interval. Thus the solution of Equation (2-28) can be obtained by modifying Equations (2-25) and (2-26) as follows:

$$\left. \begin{aligned} \underline{d}_1 &= Q_1 \underline{x}_0 + \frac{1}{2m} Q_1 \bar{\underline{e}}_1 \\ \underline{d}_k &= R_k \underline{d}_{k-1} + \frac{1}{2m} Q_k (\bar{\underline{e}}_k + \bar{\underline{e}}_{k-1}) \end{aligned} \right\} \quad (2-29)$$

where

$$\left. \begin{aligned} Q_k &= [I - \frac{1}{2m} \bar{A}_k]^{-1} \\ R_k &= Q_k [I + \frac{1}{2m} \bar{A}_k] \end{aligned} \right\} \quad (2-30)$$

$$\left. \begin{aligned} \bar{A}_k &= \frac{1}{2} (A_k + A_{k-1}) \\ \bar{e}_k &= \frac{1}{2} (e_k + e_{k-1}) \end{aligned} \right\} \quad (2-31)$$

Note that $\bar{A}_k$ and $\bar{e}_k$ are the average values of $A(t)$ and $B(t)u(t)$ in the k -th time interval, respectively. The solution of Equation (2-28) is:

$$\underline{x}(t) = \sum_{k=1}^m \phi_k(t) \underline{d}_k \quad (2-32)$$

which is also piece-wise constant.

CHAPTER 3

FIVE ADDITIONAL METHODS OF ANALYSIS

3-1 Modified Block-Pulse Method

The block-pulse method is an interval-analysis technique which is strongly connected to the trapezoidal integration rule. In fact, the quantity $\underline{d}_k$ determined by using Equation (2-25) or (2-29) gives the average value of $\underline{x}_{k-1}$ and $\underline{x}_k$ as shown in Figure 3-1, where $\underline{x}_j$ denotes the value of $\underline{x}(t)$ at $t = \frac{j}{m}$. That is,

$$\underline{d}_k = \frac{\underline{x}_{k-1} + \underline{x}_k}{2} \quad (3-1)$$

Therefore, if $\underline{x}_{k-1}$ and $\underline{d}_k$ are known, $\underline{x}_k$ can be determined from

$$\underline{x}_k = 2\underline{d}_k - \underline{x}_{k-1} \quad (3-2)$$

Equation (3-2) will be called the "inverse trapezoidal rule."

This rule obtains a point-analysis from the result of block-pulse analyzed interval-analysis. The complete algorithm for this point analysis method for time-invariant differential Equation (2-21) therefore includes Equations (2-25) and (3-2). Similarly, the complete algorithm for time-varying, differential Equation (2-28) includes Equations (2-29) and (3-2). This method, which involves the use of block-pulse analysis and the inverse trapezoidal rule, will be called the "modified block-pulse method."

3-2 Trapezoidal Rule

The trapezoidal rule of integration is a well known technique in numerical analysis [4,5]. Applying the rule to Equation (2-28) over the time interval t_{k-1} to t_k , gives

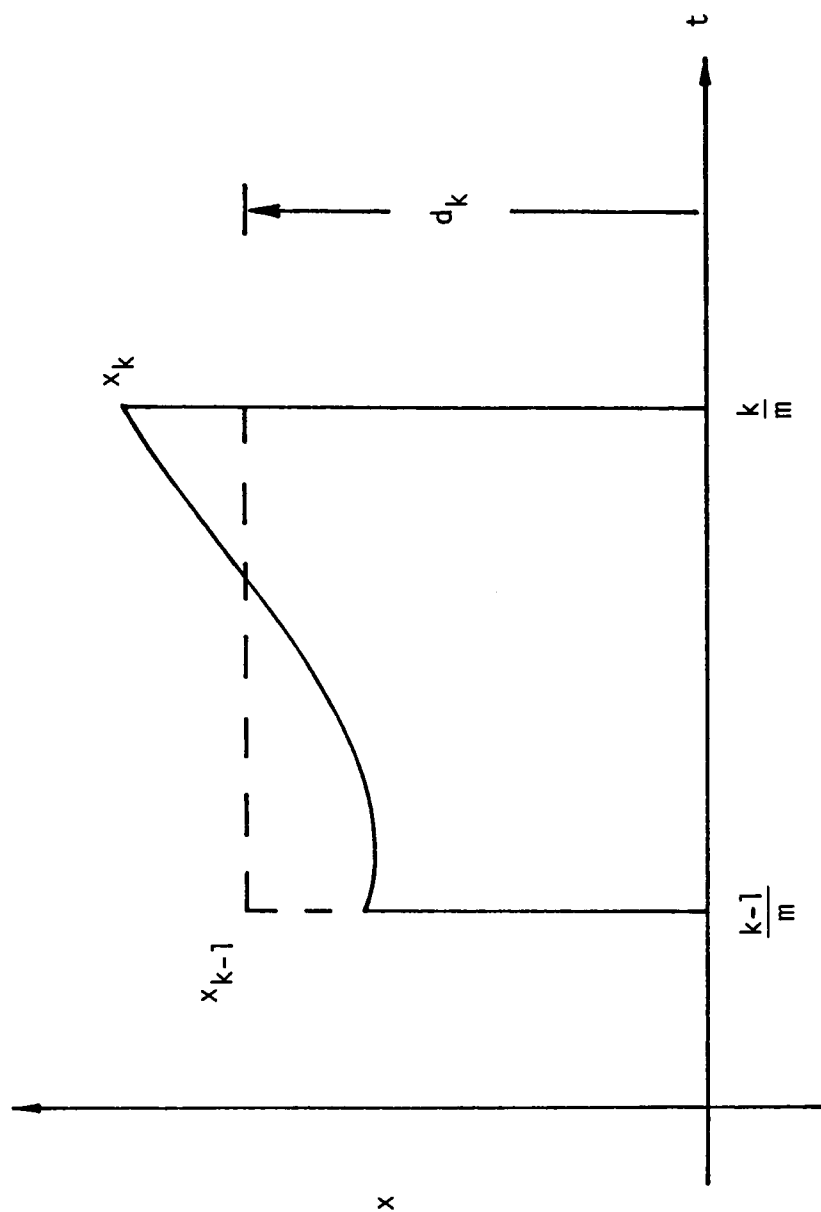


Figure 3-1. Algorithm of Modified Block-Pulse Method

$$\begin{aligned}
 \underline{x}_k &= \underline{x}_{k-1} + \int_{\frac{k-1}{m}}^{\frac{k}{m}} [A(t)\underline{x} + B(t)u] dt \\
 &= \underline{x}_{k-1} + \frac{A_k \underline{x}_k + A_{k-1} \underline{x}_{k-1}}{2m} + \frac{B_k u_k + B_{k-1} u_{k-1}}{2m}
 \end{aligned}$$

After rearranging, one gets

$$\underline{x}_k = Q_k R_k \underline{x}_{k-1} + Q_k F_k \quad (3-3)$$

where

$$\left. \begin{aligned}
 Q_k &= [I - \frac{1}{2m} A_k]^{-1} \\
 R_k &= [I + \frac{1}{2m} A_{k-1}] \\
 F_k &= \frac{1}{2m} [B_k u_k + B_{k-1} u_{k-1}]
 \end{aligned} \right\} \quad (3-4)$$

Equations (3-3) and (3-4) give the point-analysis based on trapezoidal rule.

3-3 Modified Trapezoidal Rule

An interval-analysis can be obtained from the result of trapezoidal rule by taking the midpoint value established from two consecutive data points given by the trapezoidal rule. That is,

$$\bar{\underline{x}}_k = \frac{\underline{x}_{k+1} + \underline{x}_k}{2} \quad (3-5)$$

where $\bar{\underline{x}}_k$ are the midpoint values. This approach will be called the "modified trapezoidal rule." The complete algorithm includes Equations (3-3), (3-4), and (3-5).

3-4 Runge-Kutta Method

Runge-Kutta method is well known for point-analysis. It is described here for the purpose of comparison. The fourth order Runge-Kutta algorithm is as follows:

$$x_{k+1} = x_k + \frac{1}{6} (q_1 + 2q_2 + 2q_3 + q_4) \quad (3-6)$$

where

$$\left. \begin{aligned} q_1 &= hf(t_k, x_k) \\ q_2 &= hf\left(t_k + \frac{h}{2}, x_k + \frac{q_1}{2}\right) \\ q_3 &= hf\left(t_k + \frac{h}{2}, x_k + \frac{q_2}{2}\right) \\ q_4 &= hf(t_k + h, x_k + q_3) \end{aligned} \right\} \quad (3-7)$$

3-5 Euler's Method

Euler's integration algorithm is also a well known point-analysis method. It is included here for the sake of comparison, though it is known to be not very accurate. The solution for Equation (2-28) by this method is given by

$$x_k = \left[I + \frac{1}{m} A_k \right] x_{k-1} + \frac{1}{m} B_k u_{k-1} \quad (3-8)$$

CHAPTER 4

SOLUTION TO DIFFERENTIAL EQUATIONS

The analysis methods described in Chapters 2 and 3 will be applied to the solution of differential equations. Although these methods are applicable to both linear and nonlinear differential equations, only linear cases will be discussed.

4-1 Linear Time-Invariant Differential Equation

Consider the first order, time-invariant vector differential equation

$$\dot{\underline{x}}(t) = A\underline{x}(t) + Bu(t) \quad , \quad \underline{x}(0) = \underline{x}_0 \quad (4-1)$$

where

$$\left. \begin{aligned} \underline{x} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} & \underline{x}(0) &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ A &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} & B &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \right\} \quad (4-2)$$

and $u(t)$ is a unit-step input function.

The exact solution for Equation(4-1) is

$$\left. \begin{aligned} x_1(t) &= e^{-t} - \frac{1}{2}e^{-2t} + \frac{1}{2} \\ x_2(t) &= e^{-2t} - e^{-t} \end{aligned} \right\} \quad (4-3)$$

For numerical computation the total time interval $[0,1]$ is divided into 4 sub-intervals.

Block-Pulse Method. By Equations (2-25), (2-26), and (2-27),

$$\left. \begin{aligned} \underline{d}_1 &= Q \underline{x}(0) + \frac{1}{2m} Q \underline{e}_1 \\ \underline{d}_k &= R \underline{d}_{k-1} + \frac{1}{2m} Q (\underline{e}_k + \underline{e}_{k-1}), \quad k = 2 \text{ to } 4 \end{aligned} \right\} \quad (4-4)$$

where

$$\left. \begin{aligned} Q &= [I - \frac{A}{2m}]^{-1} = \begin{bmatrix} 0.978 & 0.089 \\ -.178 & 0.711 \end{bmatrix} \\ R &= Q[I + \frac{A}{2m}] = \begin{bmatrix} 0.956 & 0.178 \\ -.356 & 0.422 \end{bmatrix} \\ \underline{e}_k &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad k = 1 \text{ to } 4 \end{aligned} \right\} \quad (4-5)$$

$$\underline{x}(t) = \sum_{k=1}^4 \phi_k(t) \underline{d}_k \quad (4-6)$$

The result is given below.

$\underline{t}$	$\underline{x}_1$	$\underline{x}_2$
$0 \leq t < \frac{1}{4}$	0.989	-.089
$\frac{1}{4} \leq t < \frac{1}{2}$	0.952	-.212
$\frac{1}{2} \leq t < \frac{3}{4}$	0.894	-.251
$\frac{3}{4} \leq t \leq 1$	0.832	-.246

Modified Block-Pulse Method. By Equation (3-2), the point-analysis value can be obtained from $\underline{d}_k$ of Equation (4-4) and the given $\underline{x}(0)$,

$$\underline{x}_k = 2 \underline{d}_k - \underline{x}_{k-1} \quad , \quad k = 1 \text{ to } 4 \quad (4-7)$$

The result is shown below.

<u>t</u>	<u>x₁</u>	<u>x₂</u>
0.00	1.000	0.000
0.25	0.978	-.178
0.50	0.926	-.246
0.75	0.862	-.256
1.00	0.802	-.236

Trapezoidal Rule. By Equations (3-3) and (3-4) the values of $\underline{x}_k$ are given by

$$\underline{x}_k = QR \underline{x}_{k-1} + Q\underline{F}_k \quad , \quad k = 1 \text{ to } 4 \quad (4-8)$$

where

$$\left. \begin{aligned} Q &= [I - \frac{1}{2m} A_k]^{-1} = \begin{bmatrix} 0.978 & 0.089 \\ -.178 & 0.711 \end{bmatrix} \\ R &= [I + \frac{1}{2m} A_{k-1}] = \begin{bmatrix} 1.000 & 0.125 \\ -.250 & 0.625 \end{bmatrix} \\ \underline{F}_k &= \frac{1}{2m} [B_k u_k + B_{k-1} u_{k-1}] = \begin{bmatrix} 0.000 \\ 0.250 \end{bmatrix} \end{aligned} \right\} \quad (4-9)$$

The numerical result is as follows:

<u>t</u>	<u>x₁</u>	<u>x₂</u>
0.00	1.000	0.000
0.25	0.967	-.267
0.50	0.899	-.279
0.75	0.831	-.260
1.00	0.770	-.228

Modified Trapezoidal Rule. By Equation (3-5) an interval-analysis is obtained from the result of Equation (4-8). That is

$$\bar{x}_k = \frac{1}{2} (x_k + x_{k-1}) , \quad k = 1 \text{ to } 4 \quad (4-10)$$

The numerical result of each sub-time interval is given below:

<u>t</u>	<u>x₁</u>	<u>x₂</u>
$0 \leq t < \frac{1}{4}$	0.984	-.134
$\frac{1}{4} \leq t < \frac{1}{2}$	0.933	-.273
$\frac{1}{2} \leq t < \frac{3}{4}$	0.865	-.270
$\frac{3}{4} \leq t \leq 1$	0.801	-.244

Runge-Kutta Method and Euler's Method. The result using Runge-Kutta method is listed below.

<u>t</u>	<u>x₁</u>	<u>x₂</u>
0.00	1.000	0.000
0.25	0.975	-.172
0.50	0.922	-.238
0.75	0.861	-.249
1.00	0.800	-.232

The result by Euler's method is as follows:

<u>t</u>	<u>x₁</u>	<u>x₂</u>
0.00	1.000	0.000
0.25	1.000	-.250
1.50	0.937	-.313
1.75	0.857	-.297
1.00	0.785	-.254

Comparison. Figure 4-1 compares the results of block-pulse method and modified trapezoidal method. Both are for interval-analysis. The exact solution is also shown for the convenience of comparison. Similarly, Figure 4-2 compares the results of modified block-pulse method, trapezoidal rule, Runge-Kutta, and Euler's methods, and compares them to the exact solution.

It is clear that the block-pulse method is superior to the modified trapezoidal method for interval-analysis. It is also clear that the Runge-Kutta method is the best among the four point-analysis methods discussed. The second best is the modified block-pulse method, followed by the trapezoidal method. Euler's method is the least accurate.

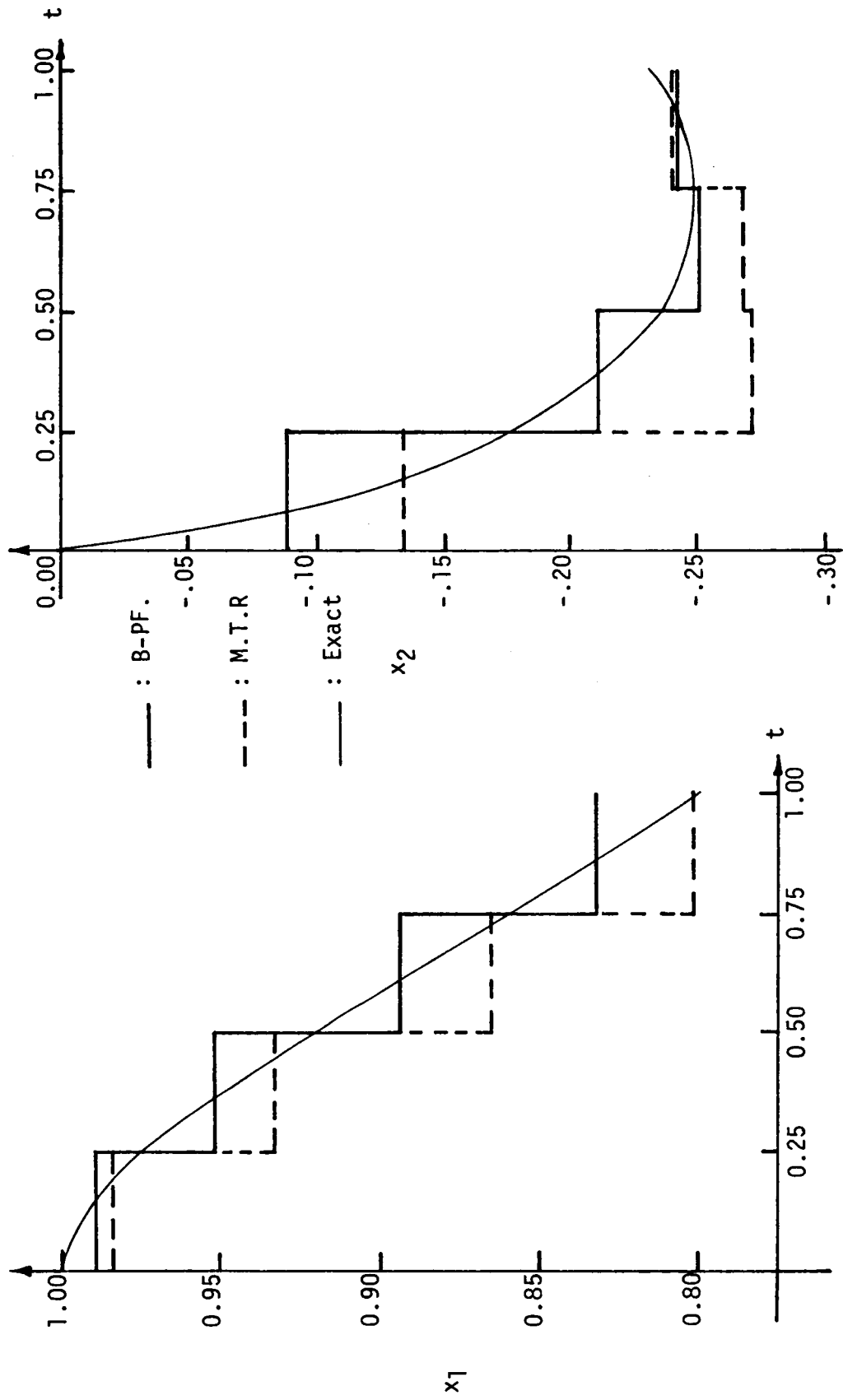


Figure 4-1. Comparison of Interval-Analysis for Time-Invariant Equation

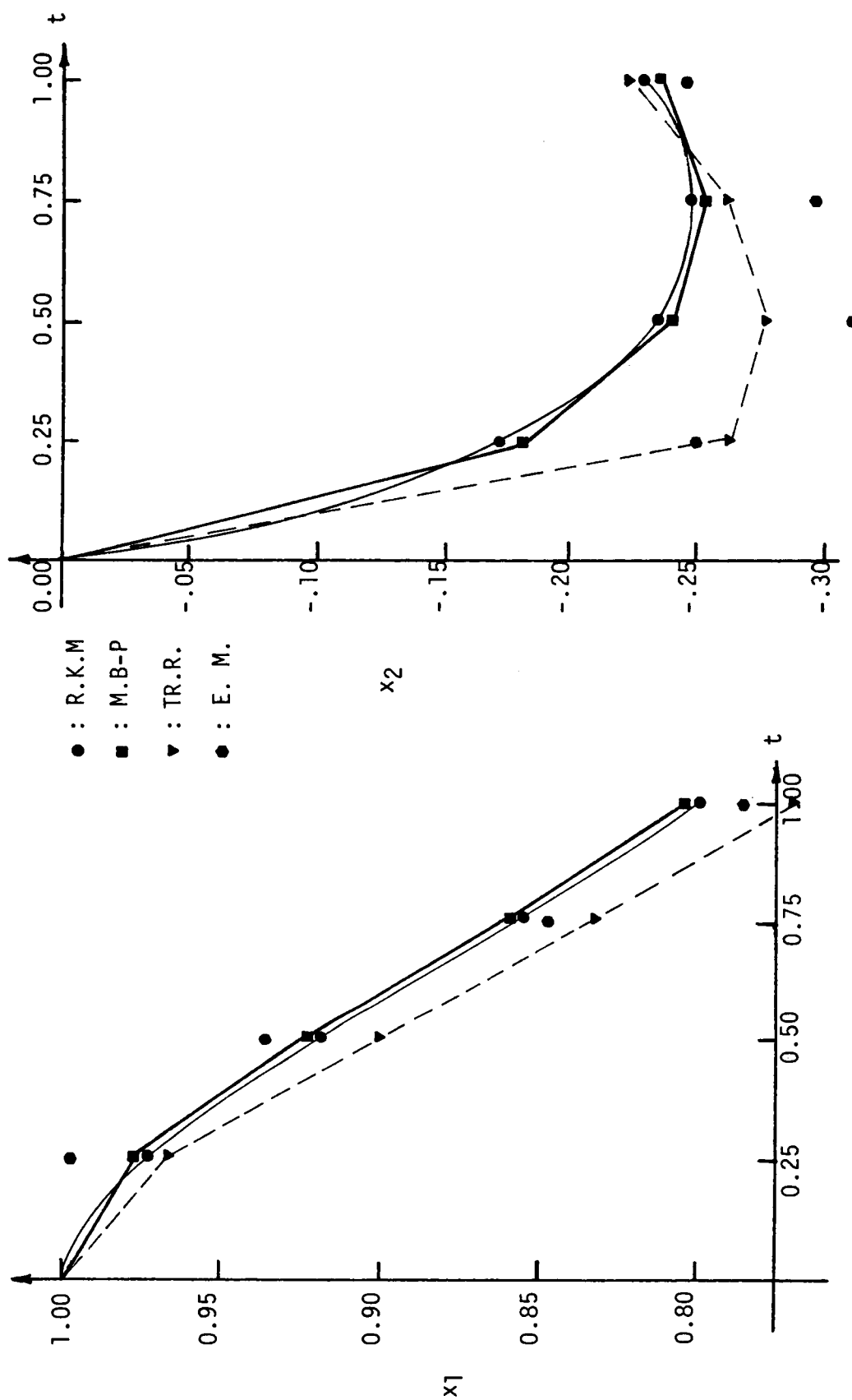


Figure 4-2. Comparison of Point-Analysis for Time-Invariant Equation

It will be shown later that all methods involve about an equal amount of computation effort. Therefore, solution accuracy is a good measure of the effectiveness of the method.

4-2 Linear Time-Varying Differential Equation

Consider a first order, time-varying vector differential equation

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -25 & 5t \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \quad (4-11)$$

with

$$\underline{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

where $u(t)$ is a unit-step input function.

For numerical computation the total time interval $[0,1]$ is divided into 4 sub-intervals. The exact point-analysis solution for Equation (4-11) is given below.

<u>t</u>	<u>x₁</u>	<u>x₂</u>
0.00	0.000	0.000
0.25	0.026	0.168
0.50	0.059	0.078
0.75	0.060	-.058
1.00	0.043	-.059

Block-Pulse Method. By Equations (2-29), (2-30), and (2-31),

$$\left. \begin{aligned} d_1 &= Q_1 \underline{x}_0 + \frac{1}{2m} Q_1 \bar{e}_1 \\ d_k &= R_k d_{k-1} + \frac{1}{2m} Q_k (\bar{e}_k + \bar{e}_{k-1}), \quad k = 2 \text{ to } 4 \end{aligned} \right\} \quad (4-12)$$

where

$$\begin{aligned} Q_1 &= \begin{bmatrix} 0.734 & 0.085 \\ -2.128 & 0.681 \end{bmatrix} \\ Q_2 &= \begin{bmatrix} 0.760 & 0.077 \\ -1.923 & 0.615 \end{bmatrix} & R_2 &= \begin{bmatrix} 0.519 & 0.154 \\ -3.846 & 0.231 \end{bmatrix} \\ Q_3 &= \begin{bmatrix} 0.781 & 0.070 \\ -1.754 & 0.561 \end{bmatrix} & R_3 &= \begin{bmatrix} 0.561 & 0.140 \\ -3.509 & 0.123 \end{bmatrix} \\ Q_4 &= \begin{bmatrix} 0.798 & 0.065 \\ -1.613 & 0.516 \end{bmatrix} & R_4 &= \begin{bmatrix} 0.597 & 0.129 \\ -3.226 & 0.032 \end{bmatrix} \\ e_k &= \begin{bmatrix} 0.000 \\ 1.000 \end{bmatrix}, \quad k = 1 \text{ to } 4 \end{aligned}$$

and

$$\underline{x}(t) = \sum_{k=1}^4 \phi_k(t) \underline{d}_k \quad (4-13)$$

The result is:

$\underline{t}$	$\underline{x}_1$	$\underline{x}_2$
$0 \leq t < \frac{1}{4}$	0.011	0.085
$\frac{1}{4} \leq t < \frac{1}{2}$	0.039	0.141
$\frac{1}{2} \leq t < \frac{3}{4}$	0.061	0.034
$\frac{3}{4} \leq t \leq 1$	0.057	-.063

Modified Block-Pulse Method. By Equation (3-2), the point-analysis value can be obtained from $\underline{d}_k$ of Equation (4-12) and the given $\underline{x}(0)$,

$$\underline{x}_k = 2 \underline{d}_k - \underline{x}_{k-1} \quad , \quad k = 1 \text{ to } 4 \quad (4-14)$$

The result is:

$\underline{t}$	$\underline{x}_1$	$\underline{x}_2$
0.00	0.000	0.000
0.25	0.021	0.170
0.50	0.057	0.111
0.75	0.065	-.044
1.00	0.049	-.082

Trapezoidal Rule. By Equations (3-3) and (3-4) the values of $\underline{x}_k$ are given by

$$\underline{x}_k = Q_k R_k \underline{x}_{k+1} + Q_k F_k \quad , \quad k = 1 \text{ to } 4 \quad (4-15)$$

where

$$\begin{aligned} Q_1 &= \begin{bmatrix} 0.747 & 0.081 \\ -2.020 & 0.646 \end{bmatrix} & R_1 &= \begin{bmatrix} 1.000 & 0.125 \\ -3.125 & 1.000 \end{bmatrix} \\ Q_2 &= \begin{bmatrix} 0.771 & 0.073 \\ -1.835 & 0.587 \end{bmatrix} & R_2 &= \begin{bmatrix} 1.000 & 0.125 \\ -3.125 & 0.844 \end{bmatrix} \\ Q_3 &= \begin{bmatrix} 0.790 & 0.067 \\ -1.681 & 0.538 \end{bmatrix} & R_3 &= \begin{bmatrix} 1.000 & 0.125 \\ -3.125 & 0.688 \end{bmatrix} \end{aligned}$$

$$Q_4 = \begin{bmatrix} 0.806 & 0.062 \\ -1.550 & 0.496 \end{bmatrix} \quad R_4 = \begin{bmatrix} 1.000 & 0.125 \\ -3.125 & 0.531 \end{bmatrix}$$

$$F_k = \begin{bmatrix} 0.000 \\ 0.250 \end{bmatrix}, \quad k = 1 \text{ to } 4$$

The point-analysis result is:

<u>t</u>	<u>x₁</u>	<u>x₂</u>
0.00	0.000	0.000
0.25	0.020	0.162
0.50	0.055	0.116
0.75	0.065	-.031
1.00	0.051	-.081

Modified Trapezoidal Rule. By Equation (3-5) an interval-analysis is obtained from the result of Equation (4-15). That is

$$\bar{x}_k = \frac{1}{2} (x_k + x_{k-1}), \quad k = 1 \text{ to } 4 \quad (4-16)$$

The interval-analysis result so obtained is:

<u>t</u>	<u>x₁</u>	<u>x₂</u>
$0 \leq t < \frac{1}{4}$	0.010	0.081
$\frac{1}{4} \leq t < \frac{1}{2}$	0.038	0.139
$\frac{1}{2} \leq t < \frac{3}{4}$	0.060	0.074
$\frac{3}{4} \leq t \leq 1$	0.058	-.056

Runge-Kutta Method. The point-analysis result using Runge-Kutta method is:

<u>t</u>	<u>x₁</u>	<u>x₂</u>
0.00	0.000	0.000
0.25	0.026	0.168
0.50	0.059	0.078
0.75	0.060	-.058
1.00	0.043	-.059

Comparison of Accuracy. Figure 4-3 compares the results of block-pulse method and modified trapezoidal method; both are for interval-analysis. The exact solution is also shown for the convenience of comparison. Similarly, Figure 4-4 compares the results of modified block-pulse method, trapezoidal rule, and Runge-Kutta methods, and compares them to the exact solution.

It is again clear that the block-pulse method is superior to the modified trapezoidal method for interval-analysis. It is also clear that the Runge-Kutta method is the best among the three point-analysis methods discussed. The second best is the modified block-pulse method, followed by the trapezoidal method.

Comparison of Computational Efforts. In the case of interval-analysis, block-pulse method is obviously the best, both in accuracy and in computation. But in the case of point-analysis, both the modified block-pulse method and the Runge-Kutta method give about the same result. A tally of the number of algebraic operations needed for each method is made. The result is shown in Table 4-1. Notice that

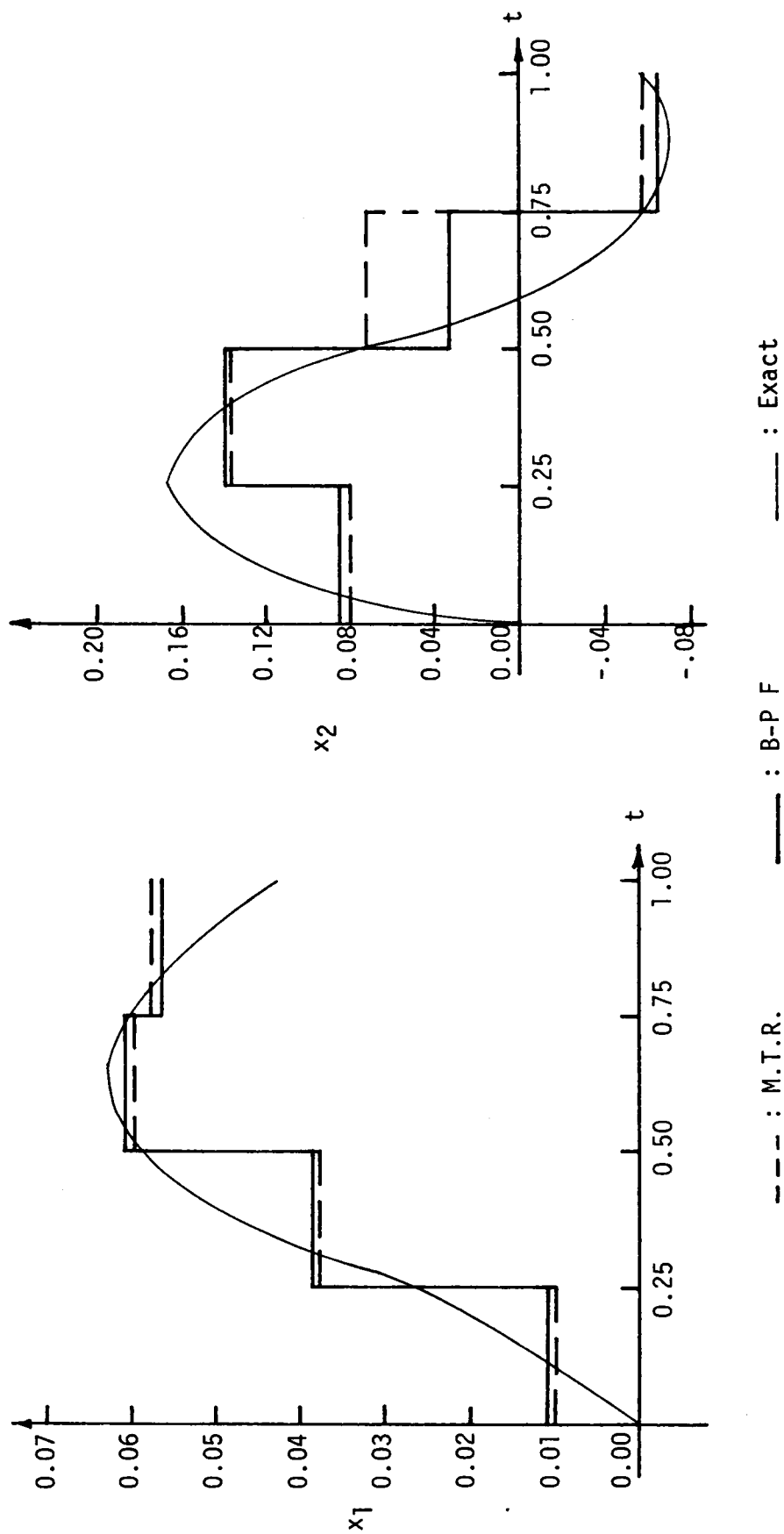


Figure 4-3. Comparison of Interval-Analysis for Time-Varying Equation

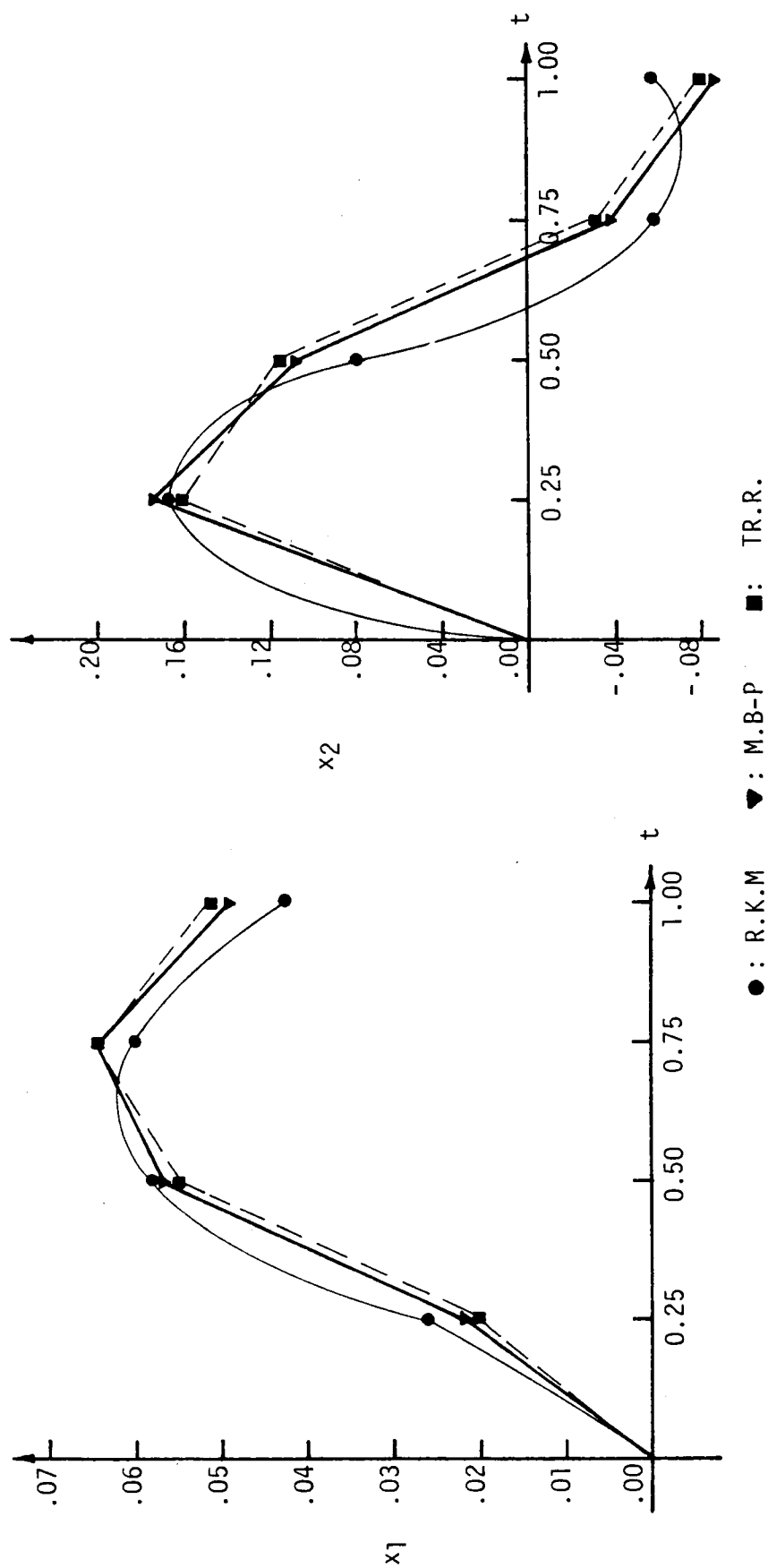


Figure 4-4. Comparison of Point-Analysis for Time-Varying Equation

TABLE 4-1. COMPARISON OF MODIFIED BLOCK-PULSE
METHOD AND RUNGE-KUTTA METHOD

Operation	Number of Operations in	
	Modified Block-Pulse Rule	Runge-Kutta Method
Addition	196	258
Subtraction	108	42
Multiplication	166	166
Division	24	8

both methods require about the same computational efforts, with the Runge-Kutta method requiring a smaller number of divisions.

CHAPTER 5

BLOCK-PULSE METHOD FOR OPTIMAL CONTROL

It is known that the optimal control of a linear system with respect to quadratic performance criteria over a finite interval of time often involves the determination of time-varying feedback gains. This chapter describes the use of block-pulse method to obtain the gains for a control system. The block-pulse approach not only offers a computational advantage in getting the approximate optimal control, but also allows simple hardware generation of the control signal.

5-1 Problem Formulation [6]

Consider a control system represented by the state equation

$$\dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}u \quad (5-1)$$

It is desired that the transients of the system state $\underline{x}$ subside within a finite time interval of t_f second. To achieve this end, the quadratic cost function

$$J = \int_0^{t_f} (\underline{x}^T \underline{W} \underline{x} + u^T \underline{V} u) dt \quad (5-2)$$

is minimized using the optimal control given by

$$u^* = \underline{V}^{-1} \underline{B}^T \underline{p}(t) \quad (5-3)$$

where $\underline{p}(t)$ satisfies the following 2nth order state equation

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{p}} \end{bmatrix} = \underline{M} \begin{bmatrix} \underline{x} \\ \underline{p} \end{bmatrix} \quad (5-4)$$

and

$$M = \begin{bmatrix} A & BV^{-1}B^T \\ W & -A^T \end{bmatrix} \quad (5-5)$$

with the boundary conditions

$$\left. \begin{aligned} \underline{x}(0) &= \underline{x}_0 \\ \underline{p}(t_f) &= \underline{0} \end{aligned} \right\} \quad (5-6)$$

Equation (5-4) is a state equation without input excitation. It is convenient to change the independent variable by defining

$$\tau = t_f - t \quad (5-7)$$

Then Equation (5-4) becomes

$$\begin{bmatrix} \dot{\underline{x}}(\tau) \\ \dot{\underline{p}}(\tau) \end{bmatrix} = -M \begin{bmatrix} \underline{x}(\tau) \\ \underline{p}(\tau) \end{bmatrix} \quad (5-8)$$

The transition matrix of Equation (5-8) is:

$$\Phi(\tau) = e^{-M\tau} = \begin{bmatrix} \Phi_{11}(\tau) & \Phi_{12}(\tau) \\ \Phi_{21}(\tau) & \Phi_{22}(\tau) \end{bmatrix} \quad (5-9)$$

where $\Phi(\tau)$ is shown partitioned into four square $n \times n$ matrices. Since $\underline{p}(\tau = 0) = \underline{0}$, the solution of Equation (5-8) can be written as:

$$\underline{x}(\tau) = \Phi_{11}(\tau)\underline{x}(\tau = 0) \quad (5-10)$$

$$\underline{p}(\tau) = \Phi_{21}(\tau)\underline{x}(\tau = 0) \quad (5-11)$$

From Equation (5-10), we have

$$\underline{x}(\tau = 0) = \phi_{11}^{-1}(\tau) \underline{x}(\tau) \quad (5-12)$$

Substituting Equation (5-12) into Equation (5-11), yields

$$\underline{p}(\tau) = \phi_{21}(\tau) \phi_{11}^{-1}(\tau) \underline{x}(\tau) \quad (5-13)$$

Then, the optimal control is reduced to

$$\begin{aligned} u^*(t) &= R^{-1} B^T \phi_{21}(t_f - t) \phi_{11}^{-1}(t_f - t) \underline{x}(t_f - t) \\ &= \underline{k}(t_f - t) \underline{x}(t_f - t) \end{aligned} \quad (5-14)$$

where

$$\underline{k}(t_f - t) = R^{-1} B^T \phi_{21}(t_f - t) \phi_{11}^{-1}(t_f - t) \quad (5-15)$$

is the optimal feedback gain vector.

The above discussion shows that the solution to the optimal control problem reduces to the determination of the transition matrix $\phi(t)$ for Equation (5-8). Block-pulse method will be used to obtain the piece-wise constant $\phi(t)$.

5-2 Block-Pulse Solution to Transition Matrix

Using the block-pulse representation,

$$\begin{bmatrix} \underline{x}(\tau) \\ \underline{p}(\tau) \end{bmatrix} \approx \sum_{i=1}^m \underline{d}_i \phi_i(\tau) \quad (5-16)$$

where $\underline{d}_i$ is an n -vector.

From Equations (2-25) and (2-26) of Chapter 2, for a system without input, the block-pulse method give

$$\underline{d}_i = R^{i-1} Q \begin{bmatrix} \underline{x}(0) \\ \underline{p}(0) \end{bmatrix} \quad (5-17)$$

where

$$\left. \begin{aligned} Q &= [I - \frac{1}{2m} M]^{-1} \\ R &= Q[I + \frac{1}{2m} M] \end{aligned} \right\} \quad (5-18)$$

Equation (5-17) shows that the block-pulse approximation to the transition matrix $\phi(\tau)$ is:

$$\phi(\tau) \approx R^{i-1} Q, \quad \frac{i-1}{m} \leq \tau < \frac{i}{m}, \quad i = 1 \text{ to } m \quad (5-19)$$

Recalling τ by $t_f - t$ and taking $t_f = 1$ yields

$$\begin{aligned} \phi(t_f - t) &= \phi(1 - t) \\ &= R^{i-1} Q, \quad 1 - \frac{i}{m} < t \leq 1 - \frac{i-1}{m}, \quad i = 1 \text{ to } m \end{aligned} \quad (5-20)$$

Let ϕ_k be the block-pulse value of ϕ in the k th sub-interval, Equation (5-20) can be written as:

$$\phi_k = R^{m-k} Q, \quad k = 1 \text{ to } m \quad (5-21)$$

where $k = m$ is the final value of $\phi(t)$ in the time interval $[0,1]$.

5-3 Example

Consider a second order time-invariant system

$$\dot{\underline{x}} = A\underline{x} + B\underline{u}, \quad \underline{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (5-22)$$

with

$$\left. \begin{aligned} A &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, & \underline{B} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ Q &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & R &= 1 \end{aligned} \right\} \quad (5-23)$$

The associate $-M$ matrix is

$$-M = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 2 & 3 & 0 & 1 \\ 1 & 0 & 0 & -2 \\ 0 & 1 & 1 & -3 \end{bmatrix} \quad (5-24)$$

Let $t_f = 1$ second and $m = 5$.

The computational steps involve the following:

- 1) Obtain the block-pulse transition matrix ϕ_k using Equation (5-21).
- 2) Partition the ϕ matrix according to Equation (5-9), and then compute the optimal feedback gain

$$\underline{K}_k = R^{-1} \underline{B}^T \Phi_{21k} \Phi_{11k}^{-1} \quad (5-25)$$

- 3) Compute the feedback control signal u_k^* from

$$u_k^* = \underline{K}_k \underline{x}_k \quad (5-26)$$

where $\underline{x}_k$ is the value of the system state $\underline{x}$ at the beginning of the k th sub-interval.

A block-diagram of the closed-loop system is shown in Figure 5-1.

The optimal feedback gains $\underline{K}_k$ are given in Figure 5-2. The optimal

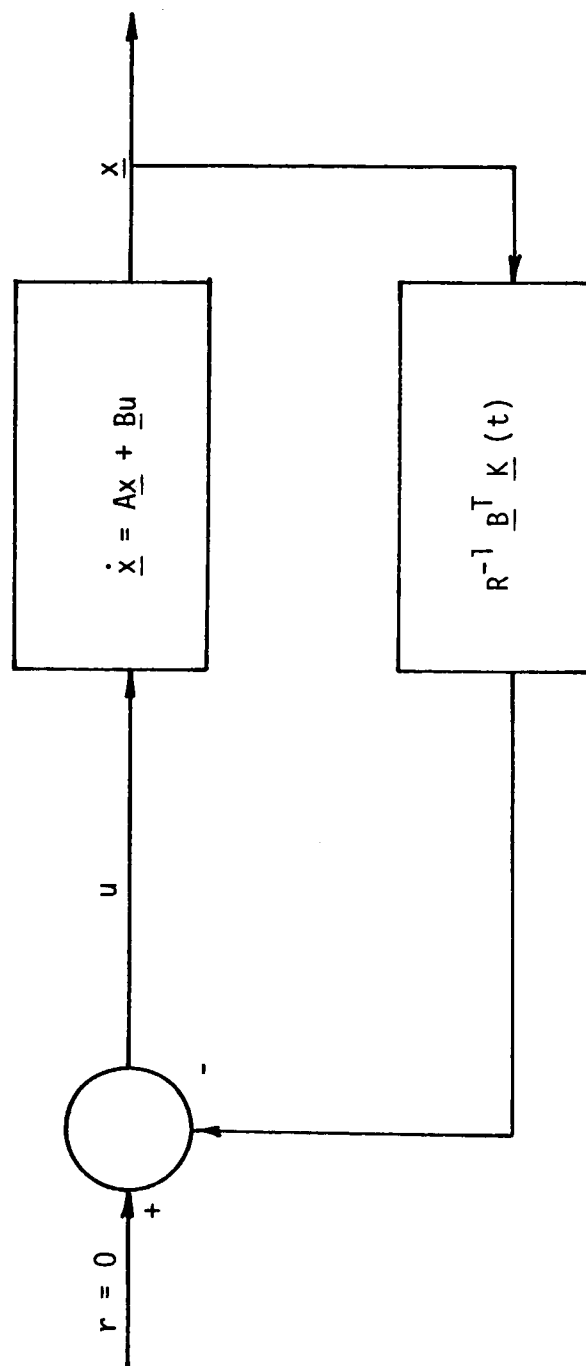


Figure 5-1. Closed-Loop Implementation of the Optimal Control

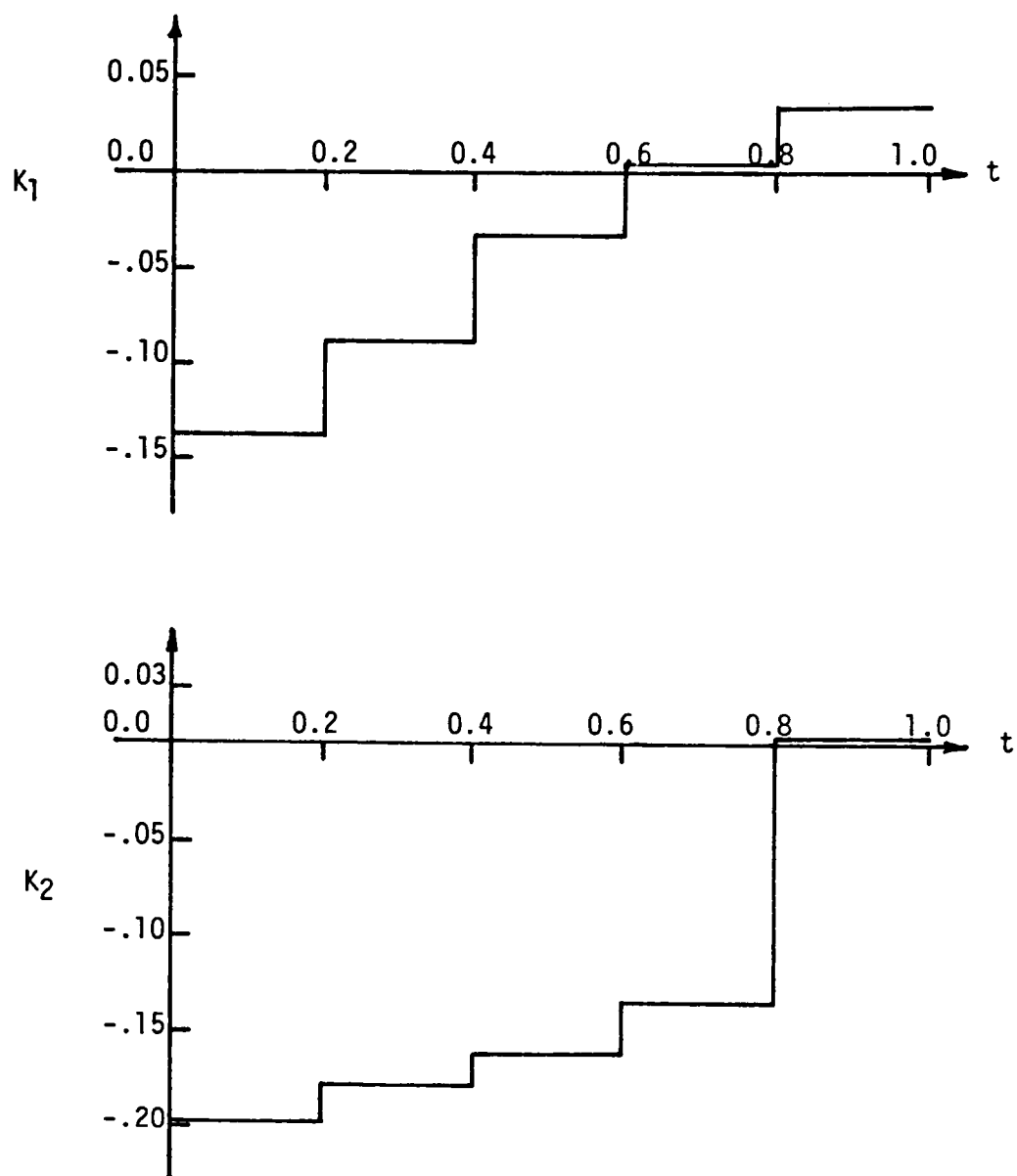


Figure 5-2. Optimal Feedback Gains $\underline{K}_k$

feedback signal u^* is depicted in Figure 5-3. The response of the system under the block-pulse optimal control is shown in Figure 5-4. The exact continuous control is also shown in the figure for comparison.

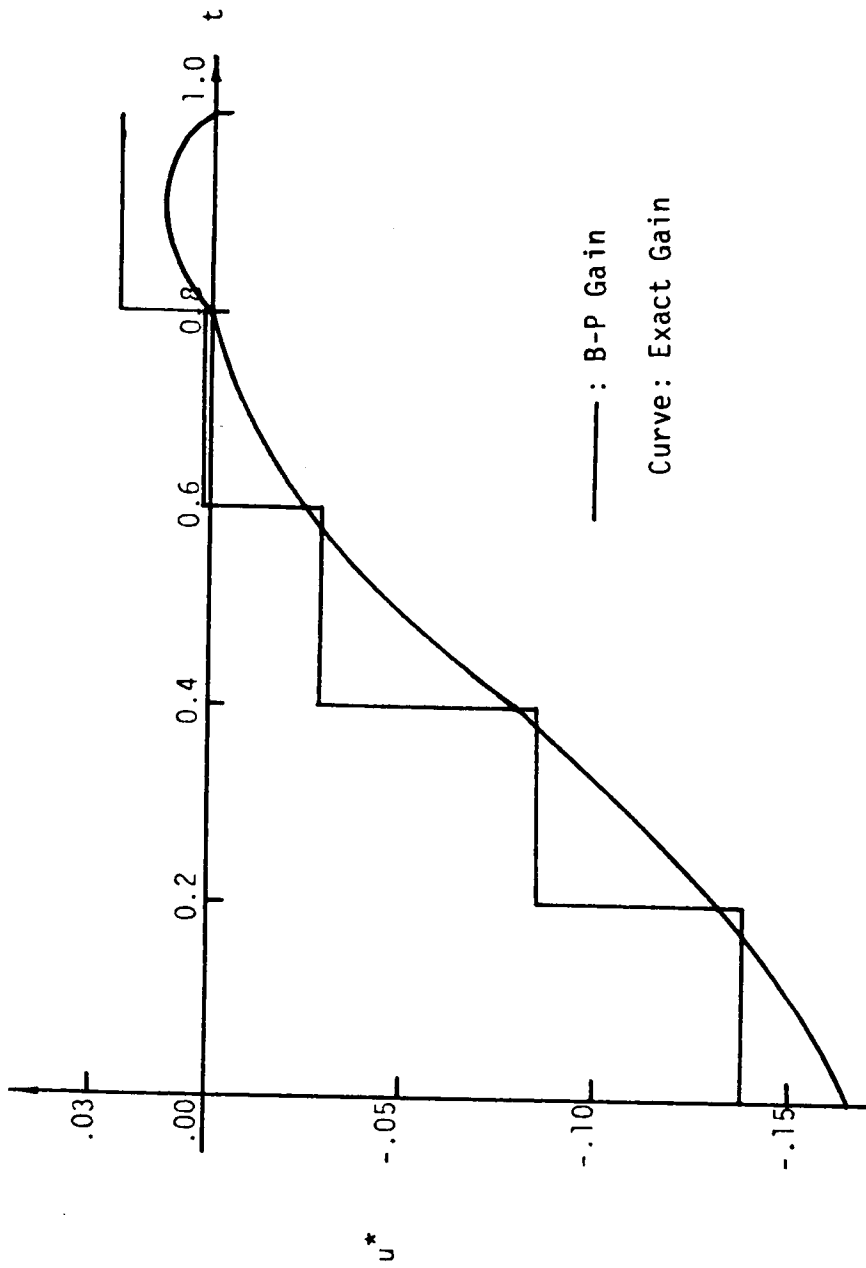


Figure 5-3. Comparison of the Optimal Feedback Signal u^*

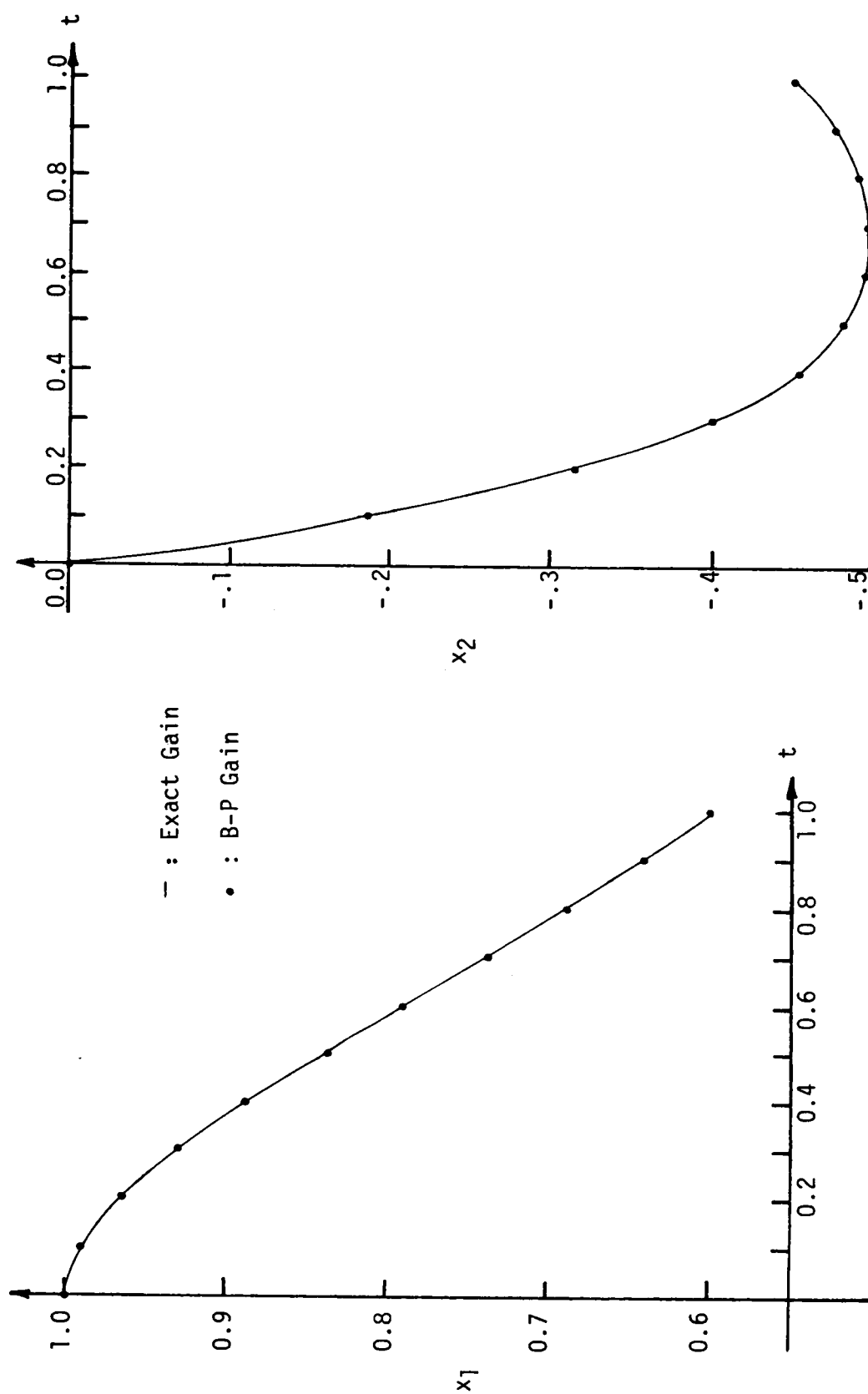


Figure 5-4. Comparison of Closed-Loop System Responses

CHAPTER 6

APPLICATION TO KALMAN FILTERING

6-1 Kalman Filter [7]

Kalman filtering is a scheme for estimating the state vector $\underline{x}(t)$ of a process where measurements $\underline{z}(t)$ are contaminated by random noise. Consider a process represented by the following state equation,

$$\dot{\underline{x}}(t) = F(t) \underline{x}(t) + \underline{w}(t) \quad (6-1)$$

and a measurement represented by

$$\underline{z}(t) = H(t) \underline{x}(t) + \underline{v}(t) \quad (6-2)$$

where $\underline{x}$ and $\underline{w}$ are n -vectors. For an $n \times n$ matrix, $\underline{z}$ and $\underline{v}$ are m -vectors, and H an $m \times n$ matrix. Vectors $\underline{w}$ and $\underline{v}$ represent Gaussian white noise with the following statistics:

$$E[\underline{w}(t)] = E[\underline{v}(t)] = 0 \quad (6-3)$$

$$\left. \begin{aligned} [w(t)w^T(\tau)] &= W(t) \delta(t - \tau) \\ [v(t)v^T(\tau)] &= V(t) \delta(t - \tau) \\ [w(t)v^T(\tau)] &= 0 \end{aligned} \right\} \quad (6-4)$$

The Kalman Filter is accomplished by the solution of the following three equations:

$$\dot{\hat{\underline{x}}}(t) = F(t)\hat{\underline{x}}(t) + K(t)[\underline{z}(t) - H(t) \hat{\underline{x}}(t)], \hat{\underline{x}}(0) = \underline{x}_0 \quad (6-5)$$

$$K(t) = P(t) H^T(t) R^{-1}(t) \quad (6-6)$$

$$\left. \begin{aligned} \dot{P}(t) &= F(t) P(t) + P(t) F^T(t) + W(t) \\ &- P(t) H^T(t) V^{-1}(t) H(t) P(t), P(0) = P_0 \end{aligned} \right\} \quad (6-7)$$

Matrix $P(t)$ in Equation (6-7) is called the error covariance matrix. Equation (6-5) is a linear differential equation whose solution can easily be computed. Equation (6-6) involves only matrix algebra. Equation (6-7) is a nonlinear matrix differential equation called the Riccati equation. The computer solution of the Riccati equation is the major effort in operating Kalman filter.

Direct application of block-pulse method to the Riccati equation leads to a set of nonlinear matrix algebraic equations whose solution is difficult. However, by introducing a set of transformations, the nonlinear Riccati equation can be replaced by two linear matrix equations whose solution, by block-pulse method, is more readily obtained. Both approaches will be discussed, with the transformation approach being presented first.

6-2 Solution of Riccati Equation by Transformation

For the nonlinear Equation (6-7) the transformations

$$\underline{p} = P\underline{y} \quad (6-8)$$

and

$$\dot{\underline{y}} = -F^T \underline{y} + H^T V^{-1} H P \underline{y} \quad (6-9)$$

result in a system of linear differential equations. Using the above, compute

$$\begin{aligned}
\dot{\underline{p}} &= \dot{P}\underline{y} + P\dot{\underline{y}} \\
&= FP\underline{y} + PF^T \underline{y} + GWG^T \underline{y} - PH^T V^{-1} HP\underline{y} \\
&\quad - PF^T \underline{y} + PH^T V^{-1} HP\underline{y}
\end{aligned}$$

Therefore

$$\dot{\underline{p}} = F\underline{p} + GWG^T \underline{y} \quad (6-10)$$

and the equivalent 2n-th order linear system equations are:

$$\begin{bmatrix} \dot{\underline{y}} \\ \dot{\underline{p}} \end{bmatrix} = \underbrace{\begin{bmatrix} -F^T & H^T V^{-1} H \\ GWG^T & F \end{bmatrix}}_M \begin{bmatrix} \underline{y} \\ \underline{p} \end{bmatrix} \quad (6-11)$$

where the system matrix has been denoted by M. By denoting the transition matrix for this linear system as ϕ , one has

$$\begin{bmatrix} \underline{y}(\tau) \\ \underline{p}(\tau) \end{bmatrix} = \underbrace{\begin{bmatrix} \phi_{yy}(\tau) & \phi_{yp}(\tau) \\ \phi_{py}(\tau) & \phi_{pp}(\tau) \end{bmatrix}}_{\phi(\tau)} \begin{bmatrix} \underline{y}(0) \\ \underline{p}(0) \end{bmatrix} \quad (6-12)$$

where $\phi(\tau)$ is shown partitioned into square nxn matrices. Writing out the expressions for $\underline{y}(\tau)$ and $\underline{p}(\tau)$ and employing Equation (6-8) yields

$$P(\tau) = [\phi_{py}(\tau) + \phi_{pp}(\tau) P(0)][\phi_{yy}(\tau) + \phi_{yp}(\tau) P(0)]^{-1} \quad (6-13)$$

Thus, the original nonlinear matrix Riccati equation has been converted to an equivalent linear matrix equation. Once $\phi(\tau)$ is computed, P may be generated as a function of time by repeated use of Equation (6-13).

This transformation approach offers a substantial advantage for time-invariant systems, that is, when the matrices F , G , V , and W are constant. Under this condition the associated transition matrix $\phi(\tau)$ can easily be computed. Although the technique is also applicable to time-varying systems, the effort required to compute the associated transition matrix $\phi(t, \tau)$ discourages the use of the method.

The next step is to obtain the transition matrix ϕ of Equation (6-12) by block-pulse method. For solving the Riccati Equation by the block-pulse functions approach, the variables $\underline{y}$, $\underline{p}$ in Equation (6-12) will be expanded into their block-pulse representations as

$$\begin{bmatrix} \underline{y}(t) \\ \underline{p}(t) \end{bmatrix} \approx \sum_{i=1}^m \underline{d}_i \phi_i(t) \quad (6-14)$$

where $\underline{d}_i$ are $2n$ -vectors.

Applying the result of Chapter 2, Equations (2-25) and (2-26) the block-pulse solution to Equation (6-12) is given by

$$\underline{d}_i = R \underline{d}_{i-1} = R^{i-1} Q \begin{bmatrix} \underline{y}(0) \\ \underline{p}(0) \end{bmatrix} \quad (6-15)$$

where

$$\left. \begin{aligned} Q &= \left[I - \frac{M}{2m} \right]^{-1} \\ R &= Q \left[I + \frac{M}{2m} \right] \end{aligned} \right\} \quad (6-16)$$

Note that

$$\underline{d}_i = \begin{bmatrix} \underline{y}(t) \\ \underline{p}(t) \end{bmatrix}, \quad \frac{i-1}{m} \leq t < \frac{i}{m}, \quad i = 1 \text{ to } m \quad (6-17)$$

Therefore, the transition matrix is approximated by

$$\Phi(t) \approx R^{i-1} Q, \frac{i-1}{m} \leq t < \frac{i}{m}, i = 1 \text{ to } m \quad (6-18)$$

Equation (6-18) shows that the block-pulse solution to $\Phi(t)$ requires only the computation of R and Q given by Equation (6-17).

Example. For a certain class of integrating gyroscopes, all units have an unknown constant drift rate, ϵ , once thermally stabilized. The gyroscopes are used to stabilize a single-axis test table. Continuous measurement is made of the table angle, θ , which is a direct measure of the integrated gyroscope drift rate. The measurement contains an error, e , which is described by an exponential autocorrelation function model of standard deviation $\sigma(\text{sec})$ and short correlation time T (sec). Kalman filtering will be used to estimate the gyro drift rate ϵ .

The dynamics of the test table, neglecting servo errors, is

$$\dot{\theta} = \epsilon t + \theta_0 \quad (6-19)$$

and the measurement is described by

$$Z = \theta + e \quad (6-20)$$

with

$$\dot{e} = -\frac{1}{T} e + w \quad (6-21)$$

where w is a white noise of zero mean and spectral density

$$r_w = 2\frac{\sigma^2}{T} (\text{sec}^2/\text{sec}) \quad (6-22)$$

Since ϵ is the quantity to be estimated, and since $\theta(t)$ represents the table dynamics, ϵ and θ are two state variables. Then Equation (6-19) is equivalent to the following set of equations:

$$\left. \begin{aligned} \dot{\theta} &= \epsilon, & \theta(0) &= \theta_0 \\ \dot{\epsilon} &= 0 \end{aligned} \right\} \quad (6-23)$$

There are two ways to apply the Kalman filter. One is to augment the two differential equations of (6-23) with (6-21), resulting in a third order system. Another way is to approximate the relatively high frequency correlated noise by a white noise process and thus delete Equation (6-21), resulting in a second order system. In this case the spectral density of the noise (now directly representing e) is $r_e = 2 \sigma^2 T (\text{sec}^2 \text{ sec})$. The latter approach will be used.

Expressing Equations (6-23) and (6-20) in matrix form, gives, respectively,

$$\begin{bmatrix} \dot{\theta} \\ \dot{\epsilon} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \epsilon \end{bmatrix} \quad (6-24)$$

and

$$Z = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \epsilon \end{bmatrix} + e \quad (6-25)$$

where the statistics of e is given by

$$\left. \begin{aligned} E[e] &= 0 \\ E[e(t)e(\tau)] &= r \delta(t - \tau), \quad r = 2\sigma^2 T \end{aligned} \right\} \quad (6-26)$$

In terms of quantities in Equation (6-11),

$$\left. \begin{aligned} F &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ H &= [1 \quad 0] \\ G &= 0 \\ R &= r_e = 2 \sigma^2 T \\ Q &= 0 \end{aligned} \right\} \quad (6-27)$$

Using Equation (6-24) in Equation (6-11), yields

$$M = \begin{bmatrix} 0 & 0 & \frac{1}{r_e} & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6-28)$$

The use of block-pulse method requires a given final time t_f . This does not restrict the original generality of the Kalman filter. Let

$$\left. \begin{aligned} t_f &= 1 \\ m &= 4 \\ r_e &= 1 \end{aligned} \right\} \quad (6-29)$$

Equation (6-18) gives the block-pulse transition matrix as shown in Table 6-1. Finally, using Equation (6-13),

TABLE 6-1. BLOCK-PULSE REPRESENTATION OF TRANSITION MATRIX

$$\Phi = \begin{bmatrix} \phi_{yy}(\tau) & | & \phi_{yp}(\tau) \\ \hline \phi_{py}(\tau) & | & \phi_{pp}(\tau) \end{bmatrix}$$

$$\Phi_1 = \begin{bmatrix} 1.000 & 0.000 & 0.125 & 0.016 \\ -0.125 & 1.000 & -0.016 & -0.002 \\ 0.000 & 0.000 & 1.000 & 0.125 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

$$\Phi_2 = \begin{bmatrix} 1.000 & 0.000 & 0.375 & 0.078 \\ -0.375 & 1.000 & -0.078 & -0.014 \\ 0.000 & 0.000 & 1.000 & 0.375 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

$$\Phi_3 = \begin{bmatrix} 1.000 & 0.000 & 0.750 & 0.297 \\ -0.750 & 1.000 & -0.297 & -0.086 \\ 0.000 & 0.000 & 1.000 & 0.750 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

$$\Phi_4 = \begin{bmatrix} 1.000 & 0.000 & 1.125 & 0.656 \\ -1.125 & 1.000 & -0.656 & -0.270 \\ 0.000 & 0.000 & 1.000 & 1.125 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

$$\underline{P}(\tau) = [\phi_{py}(\tau) + \phi_{pp}(\tau) P(o)][\phi_{yy}(\tau) + \phi_{yp}(\tau) P(o)]^{-1}$$

$$, i = 1 \text{ to } 4 \quad (6-30)$$

Values of P_i are shown in Table 6-2. A comparison of the block-pulse values of P to that of the exact P is shown in Figure 6-1.

6-3 Direct Solution to Riccati Equation

Very often the exact solution of the Riccati equation is not necessary for a practical use of Kalman filter. Instead, a piece wise constant approximate solution is adequate. Block-pulse method is ideal for this application.

Integrating Equation (6-7) gives

$$P(t) - P(o) = \int_0^t F(t') P(t') dt' + \int_0^t P(t') dt'$$

$$+ \int_0^t Q(t') dt' - \int_0^t P(t') H^T(t') R(t') H(t') P(t') dt'$$

$$(6-31)$$

By applying the block-pulse representation technique shown in Equations (2-12), (2-13), and (2-14) and the block-pulse integration formula (2-3), the terms in Equation (6-31) become

$$P(t) = \begin{bmatrix} P_1 & P_2 & \cdots & P_m \end{bmatrix} \underline{\phi} \quad (6-32)$$

$$\int_0^t F(t') P(t') dt' = \begin{bmatrix} F_1 P_1 & F_2 P_2 & \cdots & F_m P_m \end{bmatrix} B \underline{\phi} \quad (6-33)$$

$$\int_0^t P(t') F^T(t') dt' = \begin{bmatrix} P_1 F_1^T & P_2 F_2^T & \cdots & P_m F_m^T \end{bmatrix} B^T \underline{\phi} \quad (6-34)$$

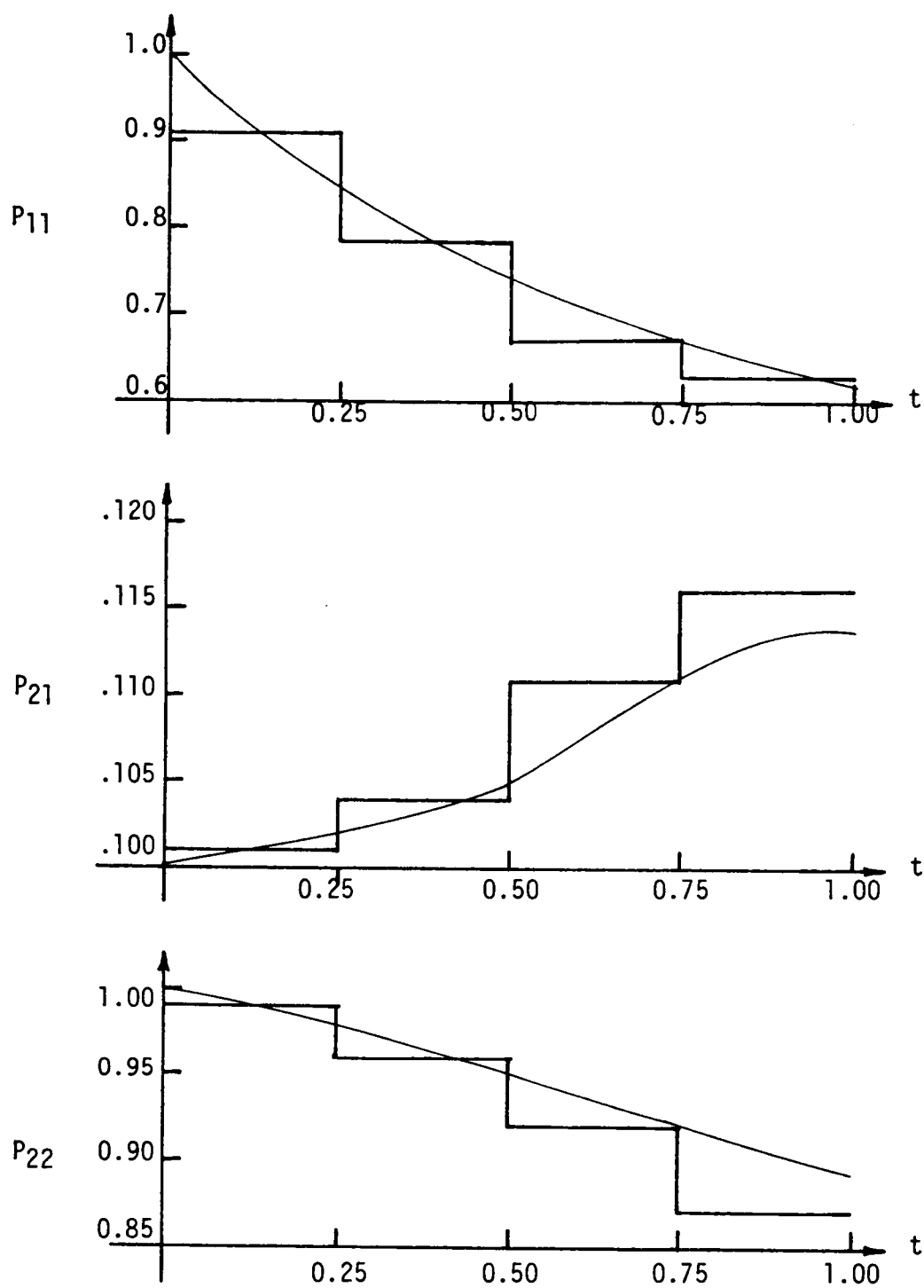


Figure 6-1. Comparison of Error Covariance Matrix Values

TABLE 6-2. BLOCK-PULSE VALUES AND EXACT VALUES OF P

Block-Pulse Values		Exact Values	
$P_1 =$	$\begin{bmatrix} 0.911 & 0.100 \\ 0.101 & 0.099 \end{bmatrix}$	,	$\begin{bmatrix} 0.841 & 0.102 \\ 0.102 & 0.098 \end{bmatrix}$
$P_2 =$	$\begin{bmatrix} 0.784 & 0.103 \\ 0.104 & 0.096 \end{bmatrix}$	,	$\begin{bmatrix} 0.738 & 0.105 \\ 0.105 & 0.095 \end{bmatrix}$
$P_3 =$	$\begin{bmatrix} 0.669 & 0.109 \\ 0.111 & 0.092 \end{bmatrix}$	,	$\begin{bmatrix} 0.669 & 0.111 \\ 0.111 & 0.092 \end{bmatrix}$
$P_4 =$	$\begin{bmatrix} 0.630 & 0.116 \\ 0.118 & 0.087 \end{bmatrix}$	,	$\begin{bmatrix} 0.621 & 0.114 \\ 0.114 & 0.089 \end{bmatrix}$

$$\int_0^t Q(t') dt' = [Q_1 Q_2 \cdot \cdot \cdot Q_m] B \underline{\phi} \quad (6-35)$$

$$\begin{aligned} & \int_0^t P(t') H^T(t') R(t)^{-1} H(t') P(t') dt' \\ &= [P_1 H_1^T R_1^{-1} H_1 P_1 \quad P_2 H_2^T R_2^{-1} H_2 P_2 \quad \cdot \cdot \cdot \quad P_m H_m^T R_m^{-1} H_m P_m] \end{aligned} \quad (6-36)$$

Because of the nature of each $\phi_i(t)$ the constant matrix $P(o)$ can be written as:

$$P(o) = \underbrace{[P(o) \quad P(o) \quad \cdot \cdot \cdot \quad P(o)]}_{m \text{ matrices}} \underline{\phi} \quad (6-37)$$

Substituting Equations (6-32) to (6-37) into Equation (6-31) and comparing the coefficient matrices of $\underline{\phi}$, yields

$$\begin{aligned} & [P_1 P_2 \cdot \cdot \cdot P_m] - [P_o P_o \cdot \cdot \cdot P_o] \\ &= [F_1 P_1 \quad F_2 P_2 \cdot \cdot \cdot F_m P_m] B \\ &+ [P_1 F_1^T \quad P_2 F_2^T \cdot \cdot \cdot P_m F_m^T] B + [Q_1 Q_2 \cdot \cdot \cdot Q_m] B \\ &+ [P_1 H_1^T R_1^{-1} H_1 P_1 \quad P_2 H_2^T R_2^{-1} H_2 P_2 \cdot \cdot \cdot P_m H_m^T R_m^{-1} H_m P_m] B \end{aligned} \quad (6-38)$$

In the above equation, B is an upper triangular matrix given by Equation (2-4). Multiplying B into the respective matrix, adding corresponding matrix elements together, and equating the corresponding matrix elements on both sides of Equation (6-38), the following set of equations are obtained.

$$\begin{aligned}
P_1 - \frac{1}{2m} [F_1 P_1 + P_1 F_1^T - P_1 H_1^T R_1^{-1} H_1 P_1] &= \frac{1}{2m} Q_1 + P_0 \\
P_2 - \frac{1}{2m} [F_2 P_2 + P_2 F_2^T - P_2 H_2^T R_2^{-1} H_2 P_2] &= \frac{1}{2m} Q_2 + 2P_1 - P_0 \\
P_3 - \frac{1}{2m} [F_3 P_3 + P_3 F_3^T - P_3 H_3^T R_3^{-1} H_3 P_3] &= \frac{1}{2m} Q_3 + 2P_2 - 2P_1 + P_0 \\
&\vdots \\
P_m - \frac{1}{2m} [F_m P_m + P_m F_m^T - P_m H_m^T R_m^{-1} H_m P_m] \\
&= \frac{1}{2m} Q_m + 2P_{m-1} - 2P_{m-2} + \dots + (-1)^{m-1} P_0 \quad (6-39)
\end{aligned}$$

The matrix brackets [] are used in this equation, rather than the brackets [], because there is no dimension consistency problem.

Equation (6-39) is the block-pulse solution to the Riccati differential Equation (6-7). The solution now involves only algebraic manipulations of matrices.

It should be mentioned Walsh functions can also be used to obtain the piecewise-constant solution. However, it is much more complex in computation as compared to the block-pulse method.

As discussed earlier, for time-invariant system the transformation approach presented in Section (6-2) is more convenient than the solution of Equation (6-39). For time-varying system, however, this statement may not be true.

CHAPTER 7

CONCLUSION

In this thesis a study was made on the analysis and synthesis of dynamical systems by means of algebraic methods. Six methods were discussed: two for interval-analysis and four for point-analysis. They were:

Interval-analysis:	Block-pulse method
	Modified trapezoidal rule
Point-analysis:	Modified block-pulse method
	Trapezoidal rule
	Runge-Kutta method
	Euler's method

It was demonstrated that the block-pulse method is better for interval-analysis and that the Runge-Kutta method is the best for point-analysis. The comparison was made on the exactness of the analyses and on the needed computation efforts.

The equivalences of the block-pulse functions to Walsh and Haar functions were explored. The results showed that there is a one-to-one transformation from a set of Walsh or Haar functions "into" a set of block-pulse functions.

The block-pulse functions are superior to Walsh and Haar functions in at least three respects. First, the "into" nature of the transformation implied that the class of all block-pulse function sets is larger, and therefore more general, than the class of all Walsh or Haar function sets. Second, the algebra involved in the block-pulse method is much simpler than that in either of the Walsh or Haar methods. Finally, the

hardware realization of block-pulse signal is also much simpler than that for Walsh and Haar signals.

Applications of the block-pulse method to optimal control and optimal estimation were presented. Block-pulse method provided a good piece-wise constant approximation for the continuous-time time-varying feedback gain of an optimal control system. The same technique could also be applied to obtain the piece-wise constant approximation for a time-varying gain of the Kalman filter.

It should be mentioned that all functions discussed in this thesis are not new [8]. The main contributions in this thesis included the expansion of the algebraic techniques for the block-pulse method, comparison of various methods by means of computer simulations, and advancing the use of block-pulse method for control and estimation.

REFERENCES

REFERENCES

1. Chen, C. F., and Hsiao, C. H., "A State-Space Approach to Walsh Series Solution of Linear Systems," *International J. System Science*, Vol. 6, No. 9, 1975, 833-858.
2. Sannuti, P. "Analysis and Synthesis of Dynamic System via Block-Pulse Functions," *Proceedings of the Institute of Electrical Engineers*, Vol. 124, No. 6, June 1977, 569-571.
3. Shieh, L. S., Yeung, C. K., and McInnis, B. C., "Solution of State-Space Equation via Block-Pulse Functions," *International J. of Control*, Vol. 28, No. 3, 1978, 383-392.
4. Ralston, A., A First Course in Numerical Analysis, McGraw-Hill, 1965.
5. Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill, 1965.
6. Anderson, B. D. O., and Moore, J. B. Linear Optimal Control, Prentice-Hall, 1971.
7. Gelb, A., Editor, Applied Optimal Estimation, The M. I. T. Press, 1977.
8. Harmuth, H. F., "Applications of Walsh Functions in Communications," IEEE Spectrum, Volume 6, No. 11, 1969, 82-91.

APPENDIXES

APPENDIX 1

```

C      TIME VARYING EXAMPLE VIA BLOCK-PULSE FUNCTIONS
C      AND INVERSE-TRAPEZOIDAL RULE
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(2,1),QX(2,1),X0(2,1),F(6,2),IPVT(4)
      DIMENSION XB(6,2),RI(2,2),XX0(2,1),EX(2,1)
      DIMENSION A0(2,2),A1(2,2),R(2,2),S(2,2),Q(2,2)
      DIMENSION UNI(2,2),E(2,1),P(2,2),B(2,1),WORK(4)
      READ (5,10) ((UNI(I,J),J=1,2),I=1,2)
      READ (5,10) ((A0(I,J),J=1,2),I=1,2)
      READ (5,10) (X0(I,1),I=1,2)
      READ (5,10) (E(I,1),I=1,2)
100  FORMAT (5X,2F10.3)
      55  FORMAT (30X,1HR,2X,2F10.3)
      54  FORMAT (10X,1HQ,2X,2F10.3)
      10  FORMAT (2F10.4)
      M=4
      MM=2*M
      T=0.25
      T2=T/2.
      A0(2,2)=-5.*T2
C      R = (I-A(K)/2M)
      DO 56 I=1,2
      DO 56 J=1,2
      56  R(I,J)=UNI(I,J)-A0(I,J)/MM
C      MATRIX RI = MATRIX R INVERSE
      CALL MINV(2,2,2,R,RI,COND,WORK,IPVT)
      WRITE (6,54) ((RI(I,J),J=1,2),I=1,2)
      CALL MMUL(2,2,2,2,2,1,RI,X0,XX0)
      CALL MMUL(2,2,2,2,2,1,RI,E,EX)
C      BLOCK-PULSE VALUE AT M=1
      DO 33 I=1,2
      33  XB(1,I)=XX0(I,1)+EX(I,1)/2.
      DO 1 I=1,2
      1  X(I,1)=XB(1,I)
C      FIND THE OTHER TIME INTERVAL'S BLOCK-PULSE VALUES
      DO 1000 L=2,4
      T1=(L-1)*T+T2
      A0(2,2)=-5.*T1
C      R = (I+A(K)/2M)
      DO 50 I=1,2
      DO 50 J=1,2
      50  R(I,J)=A0(I,J)/MM+UNI(I,J)
      A1(2,2)=-5.*T1
C      S = (I-A(K+1)/2M)
      DO 80 I=1,2
      DO 80 J=1,2
      80  S(I,J)=UNI(I,J)-A0(I,J)/MM
C      MATRIX P = MATRIX S INVERSE
      CALL MINV(2,2,2,S,P,COND,WORK,IPVT)
      WRITE (6,54) ((P(I,J),J=1,2),I=1,2)
      CALL MMUL(2,2,2,2,2,2,P,R,Q)
      WRITE (6,55) ((Q(I,J),J=1,2),I=1,2)

```

```

      CALL MMUL (2,2,2,2,2,1,Q,X,QX)
      DO 2 I=1,2
2     XB(L,I)=QX(I,1)
      CALL MMUL (2,2,2,2,2,1,P,E,B)
C     XB ARRAY IS THE COMPLETE SET OF BLOCK-PULSE
C     VALUES FOR EACH TIME INTERVAL
      DO 150 I=1,2
150   XB(L,I)=XB(L,I)+B(I,1)
      DO 3 I=1,2
3     X(I,1)=XB(L,I)
1000  CONTINUE
      WRITE (6,100) ((XB(I,J),J=1,2),I=1,4)
C     F ARRAY IS THE COMPLETE SET OF THE VALUES
C     VIA THE INVERSE-TRAPEZOIDAL RULE
      DO 111 J=1,2
111   F(1,J)=2.*XB(1,J)-0.0
      DO 500 NN=2,6
      DO 500 J=1,2
500   F(NN,J)=2.*XB(NN,J)-F(NN-1,J)
      WRITE (6,100) ((F(I,J),J=1,2),I=1,4)
      STOP
      END

```

APPENDIX 2

```

C      BLOCK-PULSE FUNTIONS GAIN OF THE SYSTEM ,
C      SIMUTATIONS WITH RUNGE-KUTIA METHOD
C      THEN , FIND INVERSE-TRAPEZOIDAL GAINS
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION G2(1,2),G02(1,2),E1IV(2,2),G0(1,2)
      DIMENSION X(2),XDOT(2),G( 5,2),XA(2),QQ(2,4)
      DIMENSION E1(2,2),E2(2,2),E1IV(2,2),BT(1,2),F(5,2)
      DIMENSION G0(1,2),Q1(4,4),WORK(4),IPVT(4)
      DIMENSION Q0(4,4),QOL(4,4),G1(1,2),E1(2,2),E2I(2,2)
      DIMENSION A(4,4),UNI(4,4),R(4,4),S(4,4),P(4,4),Q(4,4)
      DIMENSION BA(4,4),TA(4,4),Y(4,4),BTB(1,2)
      DIMENSION AA(2,2),BB(2,2),AAI(2,2),FF(1,2)
      READ (5,10) ((A(I,J),J=1,4),I=1,4)
      READ (5,10) ((UNI(I,J),J=1,4),I=1,4)
      READ (5,10) ((Q1(I,J),J=1,4),I=1,4)
      READ (5,1) (BT(1,I),I=1,2)
      1  FORMAT (2F10.4)
      10 FORMAT (4F10.4)
      11 FORMAT (1H0,4(2X,D13.6))
      12 FORMAT (1H0,2(2X,D13.6))
      13 FORMAT (55X,2(2X,D13.6))
      CALL MMUL(4,4,4,4,4,A,BA)
      CALL MMUL(4,4,4,4,4,A,BA,TA)
      DO 222 I=1,4
      DO 222 J=1,4
      222 Y(I,J)=.368*UNI(I,J)+.991*A(I,J)+1.164*BA(I,J)+.277*TA(I,J)
      DO 200 I=1,2
      DO 200 J=1,2
      AA(I,J)=Y(I,J)
      II=I+2
      200 BB(I,J)=Y(II,J)
      CALL MINV(2,2,AA,AAI,COND,WORK,IPVT)
      CALL MMUL(1,2,1,1,2,2,BT,BB,BTB)
      CALL MMUL(1,2,1,1,2,2,BTB,AAI,FF)
      M=5

```

```

C
M2=M*2.
A=A/2M
DO 15 I=1,4
DO 15 J=1,4
15 A(I,J)=A(I,J)/M2
DO 16 I=1,4
DO 16 J=1,4
16 S(I,J)=0.00
S=(I-A)
DO 20 I=1,4
DO 20 J=1,4
20 S(I,J)=UNI(I,J)-A(I,J)
C
MATRIX P = INVERSE OF MATRIX S
C
P=(I-A) INVERSE
CALL MINV(4,4,4,S,P,COND,WORK,IPVT)
DO 25 I=1,2
DO 25 J=1,2

25 E11(I,J)=0.00
DO 35 I=1,2
DO 35 J=1,2
35 E21(I,J)=0.00
E21 ARE ELEMENTS OF X'SITION MATRIX P
DO 30 J=1,2
DO 30 K=3,4
30 E21(K-3,J)=P(K,J)
C
E11 ARE ELEMENTS OF X'SITION MATRIX P
DO 40 I=1,2
DO 40 J=1,2
40 E11(I,J)=P(I,J)
CALL MINV(2,2,2,E11,E11V,COND,WORK,IPVT)
G01=BT*N21
C
CALL MMUL (1,2,1,1,2,2,8I,E21,G01)
GAIN G1 = G01*N11V
C
CALL MMUL (1,2,1,1,2,2,G01,E11V,G1)
WRITE (6,12) (G1(I,J),J=1,2)

```

```

DO 458 J=1,2
458 G(5,J)=G1(1,J)
C
R=(I+A)
DO 50 I=1,4
DO 50 J=1,4
50 R(I,J)=UNI(I,J)+A(I,J)
C
Q=P*R
CALL MMUL(4,4,4,4,4,4,P,R,Q)
QO=P*Q
CALL MMUL(4,4,4,4,4,4,P,Q,QO)
Q1=UNIT MATRIX
QOL=Q1*QO
KK=4
DO 1000 L=1,4
CALL MMUL(4,4,4,4,4,4,Q1,QO,QOL)
C
E2I ARE ELEMENTS OF X'SITION MATRIX QOL
DO 90 J=1,2
DO 90 K=3,4
90 E2I(K-2,J)=QOL(K,J)
C
E1I ARE ELEMENTS OF X'SITION MATRIX QOL
DO 100 I=1,2
DO 100 J=1,2
100 E1I(I,J)=QOL(I,J)
C
DO 60 I=1,4
DO 60 J=1,4
60 Q1(I,J)=0.DO
DO 70 I=1,4
DO 70 J=1,4
70 Q1(I,J)=QOL(I,J)
C
E1I ARE ELEMENTS OF X'SITION MATRIX P+QOL
E1IV = E1I INVERSE
CALL MINV(2,2,2,E1I,E1IV,COND,WORK,IPVT)
CALL MMUL(1,2,1,1,2,2,BT,E2I,G02)
CALL MMUL(1,2,1,1,2,2,G02,E1IV,G2)
WRITE(6,12) (G2(1,J),J=1,2)

```

```

DO 19 JJJ=1,2
  19 G(KK, JJJ)=G2(1, JJJ)
  KK=KK+1
1000 CONTINUE

C
  WRITE (6,12) ((G(I, J), J=1, 2), I=1, 5)
  RUNGE-KUTTA METHOD SIMULATION
  N=2
  TI=0.
  TF=2.
  H=.001
  T=TI
  X(1)=1.
  X(2)=0.
  WRITE (6, 71)
  71 FORMAT (1H0, //, 3X, 1H1, 13X, 2HX1, 11X, 2HX2)
  201 WRITE (6, 201) T, X(1), X(2)
  201 FORMAT (3E13.4)
  110 I=1
  L=1
  TB=T
  DO 112 K=1, N
    112 XA(K)=X(K)
    GO TO 101
  114 DO 116 K=1, N
    116 QQ(K, L)=H*XDOT(K)
    L=2
    TB=T+H/2.
    DO 122 K=1, N
      122 XA(K)=X(K)+QQ(K, L-1)/2.
    GO TO 101
  124 DO 126 K=1, N
    126 QQ(K, L)=H*XDOT(K)
    L=3
    TB=T+H/2.

```

```

DO132 K=1,N
132 XA(K)=X(K)+QQ(K,L-1)/2.
GO TO 101
134 DO136 K=1,N
136 QQ(K,L)=H*XDOT(K)
L=4
TB=T+H
DO142 K=1,N
142 XA(K)=X(K)+QQ(K,L-1)/2.
GO TO 101
144 DO150 K=1,N
146 QQ(K,L)=H*XDOT(K)
150 X(K)=X(K)+(1./6.)*(QQ(K,1)+2.*QQ(K,2)+2.*QQ(K,3)+QQ(K,4))
II=II+1
IF (II.EQ. 1002) GO TO 66
JJ=(II-1)/200+1
T=T+H
MM=T*1000.D0+.5D0
NN=MM-(MM/100)*100
IF (NN.EQ. 0) GO TO 666
GO TO 110
666 WRITE (6,201) T,X(1),X(2)
GO TO 110
101 XDOT(1)=XA(2)
XDOT(2)=(G(JJ,1)-2.)*XA(1)+(G(JJ,2)-3.)*XA(2)
GO TO (114,124,134,144),L
C F( 5,2) IS THE DIMENSION OF INVERSE TRAPEZOIDAL
C RULE'S VALUE OF THE OPTIMAL GAIN OF THE SYSTEM
66 DO 111 J=1,2
111 F(1,J)=2.*G(1,J)-FF(1,J)
DO 500 NN=2,5
DO 500 J=1,2
500 F(NN,J)=2.*G(NN,J)-F(NN-1,J)
WRITE (6,13) ((F(I,J),J=1,2),I=1,5)
STOP
END

```

VITA

Cheng-Chiian Liu was born in Taiwan Province, Republic of China on June 26, 1945. He attended Lung-Tan Public Elementary School and Chung-Li Provincial High School. He entered the Chinese Naval Academy on September 1, 1963, received a Bachelor of Science degree, and was commissioned as an Ensign, Chinese Navy, on September 1, 1967. During the next ten years he served in the Navy as a line officer and later as a research staff member at Chung-Shan Institute of Science and Technology.

On September 1, 1977, Mr. Liu received a government scholarship from the Republic of China for graduate study at The University of Tennessee, Knoxville. He received a Master of Science degree from The University of Tennessee, Knoxville, with a major in Electrical Engineering on March 18, 1980. Mr. Liu is currently a Commander in the Chinese Navy.