

SYSTEM IMPULSE RESPONSE FROM DIGITAL CROSSCORRELATION
OF ADDITIVE INPUT NOISE AND OUTPUT SIGNAL

A THESIS

Presented to

The Faculty of the Graduate Division

by

Olin Alvin Williams, Jr.

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

Georgia Institute of Technology

April, 1969

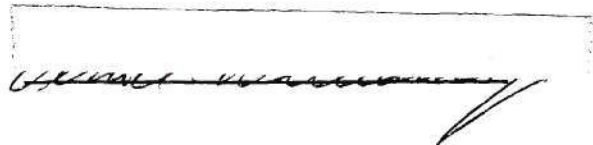
SYSTEM IMPULSE RESPONSE FROM DIGITAL CROSSCORRELATION
OF ADDITIVE INPUT NOISE AND OUTPUT SIGNAL

Approved:

Chairman

Date approved by Chairman: 5/1/69

In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

A handwritten signature in black ink, enclosed within a rectangular box. The signature is stylized and appears to be a cursive name.

7/25/68

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to Dr. John B. Peatman, my advisor, for his suggestion of this topic and especially for his guidance and encouragement during the course of this research. I would also like to thank Dr. J. H. Schlag and Dr. R. H. Pettit, the other members of my reading committee, for their help in carefully reviewing this manuscript. Appreciation is also extended to Dean Hansen and Dr. Trabant from whom the Institutional support was obtained for the purchase of logic boards and other special hardware. And, above all I would like to extend my deepest appreciation to Miss Georganne (Mickie) Pugh for her help in the preparation of this thesis, particularly in the typing.

TABLE OF CONTENTS

| | Page |
|---|------|
| ACKNOWLEDGMENTS | ii |
| LIST OF TABLES. | v |
| LIST OF ILLUSTRATIONS | vi |
| SUMMARY | viii |
| Chapter | |
| I. SYSTEM DYNAMIC RESPONSE MEASUREMENTS. | 1 |
| System Modeling and Identification | |
| Dynamic Characteristics | |
| Impulse Response Measurements - Correlation Method | |
| II. THEORY OF CROSSCORRELATION MEASUREMENTS | 5 |
| Introduction | |
| Correlation Functions | |
| Impulse Response Measurements by Correlation Method | |
| III. BINARY NOISE TEST SIGNALS | 15 |
| Introduction | |
| Discrete Interval Binary Noise | |
| Periodic Pseudo-Random Noise | |
| IV. HEURISTIC EXAMPLE OF CORRELATION METHOD. | 29 |
| Introduction | |
| Finite Pulse Response Approximation | |
| System Response to Periodic Binary Pseudo-Noise | |
| Digital Crosscorrelation of Input Noise and Output Response | |
| V. DESIGN AND CONSTRUCTION OF PROTOTYPE | 41 |
| Introduction | |
| Prototype Calculator Algorithm | |
| Basic Overall Design | |
| Logic Element Descriptions | |
| Details of Design and Construction | |

TABLE OF CONTENTS CONTINUED

| Chapter | Page |
|--|------|
| VI. EXPERIMENTAL TESTS AND RESULTS | 73 |
| Test Configurations | |
| Tests and Results | |
| Conclusions | |
| APPENDIX | 80 |
| BIBLIOGRAPHY | 83 |

LIST OF TABLES

| Table | Page |
|---|------|
| 1. Quantized Values of $n(t)$, $h(t)$, and $y(t)$ | 36 |
| 2. Values of Crosscorrelation | 37 |
| 3. Sign Plus Two's Complement Code | 42 |

LIST OF ILLUSTRATIONS

| Figure | Page |
|--|------|
| 1. A Causal, Time-Invariant Linear System. | 10 |
| 2. System With Signal Plus Noise Input | 13 |
| 3. A Discrete Interval Noise Signal. | 17 |
| 4. Noise Signals and Autocorrelation Functions . . . | 19 |
| 5. Graphical Representation of Convolution | 20 |
| 6. Shift Register of Degree n With Logical Feedback. | 22 |
| 7. Modulo 2 Adder. | 23 |
| 8. Maximum Length Shift Register Sequence Generator. | 24 |
| 9. Pseudo-Random Sequences, Waveform and Autocorrelation | 26 |
| 10. Periodic Pseudo-Random Binary Noise Autocorrelation Functions | 28 |
| 11. Impulse and Pulse Inputs to System and Associated Responses | 31 |
| 12. Pulse Input and Associated Output Response. . . . | 33 |
| 13. Pseudo-Random Noise Response by Superposition . . | 35 |
| 14. Digital Prototype and Test System | 43 |
| 15. Prototype Calculator Algorithm. | 46 |
| 16. Block Diagram of Prototype Design | 48 |
| 17. Word and Recirculating Loop Formats | 49 |
| 18. Logic Symbols | 53 |
| 19. General Purpose Logic Cards | 55 |
| 20. Special Purpose Logic Cards | 56 |

LIST OF ILLUSTRATIONS CONTINUED

| Figure | Page |
|--|------|
| 21. Delay Line and Input/Output Logic. | 57 |
| 22. Crystal Clock Waveforms. | 54 |
| 23. Delay Line Input and Output Waveforms. | 59 |
| 24. Logic Design of Prototype Subsystems | 61 |
| 25. Timing and Control Circuitry | 64 |
| 26. Timing Diagram | 65 |
| 27. Prototype Calculator | 71 |
| 28. Card Locations and Wiring of Prototype | 72 |
| 29. Test Waveforms | 74 |

SUMMARY

An important problem in the area of system modeling and identification is that of determining the dynamic characteristics of a given system while it is in operation. One useful measure of the dynamic behavior of a system is its impulse response. The impulse response of a linear system can be determined while the system is in operation by using a crosscorrelation technique. This method consists of crosscorrelating an appropriate additive input test signal with the system output. The objective of this thesis is to demonstrate the effectiveness of using a pseudo-random noise sequence as a test signal and digital techniques to perform the crosscorrelation operation. Pseudo-random noise sequences possess a number of desirable properties which significantly simplify the algorithm involved in performing the crosscorrelation calculation. A digital prototype using integrated circuit logic was constructed to implement this impulse response calculation process. The prototype calculator was tested and found to function as anticipated, thus demonstrating the effectiveness of a digital approach to the correlation method when pseudo-random noise is used as an input test signal.

CHAPTER I

SYSTEM DYNAMIC RESPONSE MEASUREMENTS

System Modeling and Identification

A model may be defined for engineering purposes as "a representation of the essential aspects of a planned or an existing system, which provides the information about that system in a usable form."¹ A model may be conceptual, physical, or mathematical, depending on the *essential aspects* of the system that is to be represented.

In the engineering field models of systems are used extensively. In the area of research, models are often used to give an interpretation of measured information or data. The designer may use models of components to create an overall system that meets specified design criteria. In adaptive control systems the actual operation of a system depends on certain knowledge about the system. First, suitable measurements are made to determine the parameters for an appropriate model. This knowledge of the system is then evaluated on the basis of predetermined criteria and adjustments are made according to some algorithm on controllable parameters of the system.²

There are a number of basic considerations involved in choosing a model for parameter measurements. One is

whether the dynamic or the static behavior of the system is to be represented. Another is whether or not the system to be modeled is linear or non-linear. Whether or not measurements may be made while a system is in normal operation must also be considered. From this point many more considerations are involved that are strongly object-oriented. For example, there might be a need to determine conditions under which a linear model could be used to represent a non-linear system. Other examples are incorporating *a priori* knowledge, approximating complex systems with simpler models, and judging the quality of the model.¹

Dynamic Characteristics

One important problem in the area of system modeling and identification is that of determining the dynamic characteristics of a given system while it is in operation. A useful measure of the dynamic behavior of a system is its impulse response $h(t)$. If the autocorrelation function of the input control signal (ψ_{ii}) and the crosscorrelation between input and output (ψ_{io}) are known, then the system function or impulse response, $h(t)$, can be found by a deconvolution process. That is, solving

$$\psi_{io}(\tau) = \int_{-\infty}^{\infty} h(t)\psi_{ii}(\tau - t)dt \quad (1-1)$$

for $h(t)$. In general, however, the process of deconvolution is not a straight forward task.³ By injecting specific test signals into the system along with the control signal and looking at the output, much information about the dynamic characteristics can be obtained.

Using various sinusoidal test signals with relatively small magnitudes, as compared to the control signal, and averaging the output over a suitable number of periods, the dynamic characteristics in the form of amplitude and phase information can be determined. In order to determine the impulse response of a system, a test signal consisting simply of a large impulse could be used. In this case the normal output of the system would be disrupted for a large part of the duration of the impulse response.

There are many systems in which it is undesirable to interrupt the normal operation or to have the impulse response present at the output. Also, a system may not be able to withstand a large impulse. One solution to these problems is to apply small impulses as a test signal and average the output. In this way, the impulse response can be obtained without drastically affecting the normal operation or output of the system. The time that it takes to obtain the impulse response depends on the relative magnitude of the test and control signals. For relatively small test impulses (or pulses of short duration with respect to the response time of the system) there will be a need to average

more output samples in order to obtain useful results. It may be the case, as it is in many dynamic systems, that the transfer function changes with time faster than the averaging time needed to obtain the impulse response. In these cases, a faster method is needed.

Impulse Response Measurements - Correlation Method

Using a crosscorrelation technique the impulse response of a system can be determined in one response time. This method consists of crosscorrelating an appropriate input test signal with the system output to obtain the impulse response. The test signal, which is added to the normal input of the system, can be any low level random signal with an *autocorrelation* function that approximates an impulse function.⁴

The objective of this thesis is thus to demonstrate the effectiveness of digital methods in obtaining a measurement of the dynamic characteristics (in the form of the impulse response) of a system using *pseudo-random noise* and the correlation technique.

CHAPTER II

THEORY OF CROSSCORRELATION MEASUREMENTS

Introduction

This chapter deals with the basic mathematical theory involved in the correlation measurement of system impulse responses. The first section includes the definitions and various useful properties of the autocorrelation and cross-correlation functions. This is followed by a brief development of the mathematical technique used in obtaining system impulse responses from a crosscorrelation of an input test signal and the system output response. Requirements of the test signal are also briefly outlined with a detailed discussion reserved for Chapter III. For an intuitive approach to the crosscorrelation technique, a heuristic example is presented in Chapter IV.

Correlation Functions

The autocorrelation function of a wide sense stationary random process is defined as

$$\psi(t_1, t_2) = \psi(\tau) = E[XY] \quad (2-1)$$

where X and Y are random variables at times t_1 and t_2

respectively and $\tau = t_2 - t_1$. $E[XY]$ is the expectation of the product XY . The autocorrelation function for any argument is simply the average of the product xy which are values of the function τ seconds apart. If the probability of any given product having the first term x between x and $x + dx$ and the second term y between y and $y + dy$ is given by $P_\tau(x,y)dxdy$, then the expected value $E[XY]$ is the summation, or integration, of all products multiplied by the respective probabilities. That is

$$\psi(\tau) = E[XY] = \iint_{-\infty}^{\infty} xy P_\tau(x,y) dx dy \quad (2-2)$$

If the random process is ergodic, the ensemble average may be replaced by a time average involving a single sample function $x_1(t)$ of the random process $[x(t)]$. Then,

$$E[XY] = \langle x_1(t_1) x_1(t_2) \rangle \quad (2-3)$$

And if $t_2 = t_1 + \tau$ then,

$$\psi_{xx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x_1(t_1) x_1(t_1 + \tau) dt_1 \quad (2-4)$$

or simply⁵

$$\psi_{xx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t)x(t + \tau)dt \quad (2-5)$$

Thus, the correlation function can be thought of in two ways. One is from the viewpoint of probability distributions describing a stationary time series. The other is in the light of a shifting, multiplying, and averaging process. Though both viewpoints are essential for analysis, the latter is usually strongly preferable for experimental measurements.⁶

Since it makes no difference whether the function is shifted forward or backward by τ seconds before multiplying and averaging, the autocorrelation can also be written as

$$\psi_{xx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t - \tau)x(t)dt \quad (2-6)$$

From these last two expressions it can be seen that the autocorrelation function is an even function of τ ; that is

$$\psi_{xx}(\tau) = \psi_{xx}(-\tau) \quad (2-7)$$

If the random variables X and Y are from different random processes, the term crosscorrelation is used to represent the expectation $E[XY]$.⁵ If both random processes are ergodic then the crosscorrelation function can be represented by a time average. That is,

$$\psi_{xy}(\tau) = \langle x(t)y(t + \tau) \rangle \quad (2-8)$$

The time average can be written in integral form as either of the following expressions:

$$\psi_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t)y(t + \tau) dt \quad (2-9)$$

or

$$\psi_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t - \tau)y(t) dt \quad (2-10)$$

This is due to the fact that a shift of $y(t)$ by τ seconds in one direction is equivalent to a shift of $x(t)$ by τ seconds in the opposite direction. This leads to the property that

$$\psi_{xy}(\tau) = \psi_{yx}(-\tau) \quad (2-11)$$

A form of the crosscorrelation function which will be referred to later is one in which the argument τ is replaced by a shift of $\tau' - a$.

$$\psi_{xy}(\tau) = \psi_{xy}(\tau' - a) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t - \tau')y(t - a) dt \quad (2-12)$$

A result that will be used later concerns the cross-correlation of a system input signal that is composed of the sum of two components with the output of the system.

Suppose

$$g(t) = x(t) + y(t) \quad (2-13)$$

Then the crosscorrelation of the functions $f(t)$ and $g(t)$ is given by

$$\psi_{fg}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(t - \tau) g(t) dt \quad (2-14)$$

or in terms of $x(t)$ and $y(t)$

$$\psi_{fg}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(t - \tau) [x(t) + y(t)] dt \quad (2-15)$$

Finally, writing the right side of the equation as two integrals the crosscorrelation becomes

$$\psi_{fg}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(t - \tau) x(t) dt + \quad (2-16)$$

$$\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(t - \tau) y(t) dt$$

or simply

$$\psi_{fg}(\tau) = \psi_{fx}(\tau) + \psi_{fy}(\tau) \quad (2-17)$$

Impulse Response Measurements by Correlation

The system transfer function or impulse response of a causal, time-invariant linear system can be measured by calculating the crosscorrelation of an input test signal and the system output. To show this, first consider a linear system, identified by its impulse response $h(t)$, with input $f_i(t)$ and corresponding output $f_o(t)$ as illustrated in Figure 1.

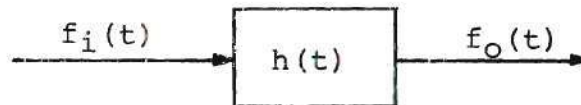


Figure 1. A Causal, Time-Invariant Linear System.

The crosscorrelation of the input and output $\psi_{io}(\tau)$ is given by

$$\psi_{io}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f_i(t - \tau) f_o(t) dt \quad (2-18)$$

The relationship between the input and the output of the

system is given by the superposition or convolution integral

$$f_o(t) = \int_0^{\infty} h(x) f_i(t - x) dx \quad (2-19)$$

The limits of integration are from 0 to ∞ rather than from $-\infty$ to ∞ due to the causality assumption for $h(x)$. After substituting this expression for $f_o(t)$ into the previous equation, $\psi_{io}(\tau)$ can be written as

$$\psi_{io}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f_i(t - \tau) \int_0^{\infty} h(x) f_i(t - x) dx dt \quad (2-20)$$

Interchanging the order of integration and averaging (which is valid where practical systems and signals are concerned) yields

$$\psi_{io}(\tau) = \int_0^{\infty} h(x) \left[\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f_i(t - \tau) f_i(t - x) dt \right] dx \quad (2-21)$$

The expression in the brackets is the autocorrelation of $f_i(t)$ with argument $\tau - x$.⁶ Thus

$$\psi_{io}(\tau) = \int_0^{\infty} h(x) \psi_{ii}(\tau - x) dx \quad (2-22)$$

A comparison of this equation with Equation (2-19) indicates

that $\psi_{iO}(\tau)$ would be the output response of the system if the input were $\psi_{ii}(\tau)$. If $\psi_{ii}(\tau)$ is an impulse function, then $\psi_{iO}(\tau)$ will be the impulse response of the system. That is, if

$$\psi_{ii}(\tau) = N_0 \delta(\tau) \quad (2-23)$$

then

$$\psi_{iO}(\tau) = N_0 h(\tau) \quad (2-24)$$

The problem then is to find a test signal such that its autocorrelation is an impulse.

White noise is defined as any random process whose spectral density is constant over all frequencies. It can be shown that if the spectral density of white noise is N_0 then its autocorrelation is the impulse function $N_0 \delta(t)$.⁵ Because of the infinite power requirements, white noise is not physically realizable. However, there are many random and pseudo-random noise signals that can be used to approximate the process. Various noise signals are discussed in the next chapter.

The impulse response can also be obtained using the above technique while a system is in normal operation. In this case the random noise test signal $n_i(t)$ is added to the normal output of the system $f_i(t)$. The impulse response

is then obtained from the crosscorrelation of the input noise test signal $n_i(t)$ and the output signal $y(t)$. To demonstrate this, consider the system in Figure 2.

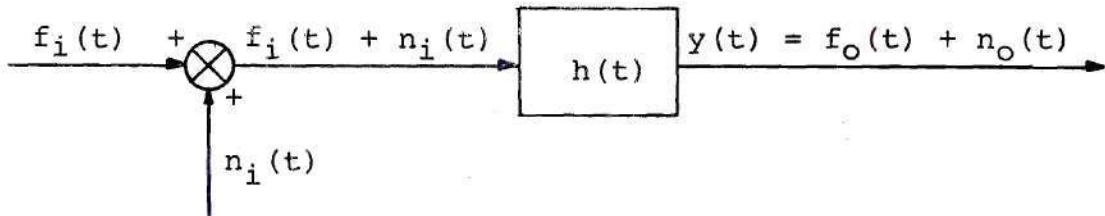


Figure 2. System With Signal Plus Noise Input.

The output $y(t)$ can be expressed as $f_o(t) + n_o(t)$ which are the output responses to inputs $f_i(t)$ and $n_i(t)$ respectively. The crosscorrelation of $n_i(t)$ and $y(t)$ can be written as

$$\psi_{n_i y}(\tau) = \psi_{n_i n_o}(\tau) + \psi_{n_i f_o}(\tau) \quad (2-25)$$

as shown earlier. If the input random noise and the normal system output $f_o(t)$ are uncorrelated, then

$$\psi_{n_i f_o}(\tau) = 0 \quad (2-26)$$

Thus

$$\psi_{n_i y}(\tau) = \psi_{n_i n_0}(\tau) = h(\tau) \quad (2-27)$$

the desired result.

CHAPTER III

BINARY NOISE TEST SIGNALS

Introduction

As discussed in the previous chapter, a test signal is needed with an autocorrelation function that approximates an impulse function as closely as possible. In almost any case the input test signal used may be any stationary random signal with great bandwidth.⁴ There are many advantages, however, in using a test signal that contains only two values, for example $\pm a$. This greatly simplifies the multiplication operation required in the computation of the cross-correlation function. If, furthermore, the test signal changes at only multiples of some basic clock period, then this simplifies the digital implementation of delaying and averaging, as required by the computation. Also, the maximum energy for a given peak value is obtained with the use of such binary signals. It is thus possible to obtain the greatest signal to noise ratio for a given degree of system disturbance.⁷ This advantage may be an important consideration in measuring the impulse response of a system if the measurement is to be made without drastically disturbing the normal operation of the system or driving it into saturation.

Although there are many types of binary noise, only

that of *discrete interval* binary noise will be discussed here. As an illustration of the probabilistic view, the autocorrelation of *truly random* discrete interval binary noise will be calculated. Also, *periodic pseudo-random* binary noise sequences and their randomness properties will be discussed and compared to the truly random noise signals. A pseudo-random signal is used in the prototype constructed to implement the correlation technique being presented here.

Discrete Interval Binary Noise

In the case of discrete interval binary noise the times of the transition from one state to the other are explicitly specified. The state in each interval is chosen independently of the state in any preceding interval. To determine the autocorrelation of this type of noise, consider the more general case of discrete interval noise. Figure 3 illustrates a discrete interval random noise signal. The signal is constant for a time interval T_1 then jumps to another value. There is equal probability of the signal being positive or negative. If A_n is the magnitude of the signal $f(t)$ in the interval $\{nT_1, (n+1)T_1\}$, the probability that A_n lies between x and $x + dx$ is given by

$$P(x < A_n < x + dx) = p(x)dx \quad (3-1)$$

The autocorrelation function can be computed from the

expectation of the product $e_1 e_2$.

$$\psi(t_1, t_2) = E[e_1 e_2] = \iint e_1 e_2 p(e_1, e_2; t_1, t_2) de_1 de_2 \quad (3-2)$$

Here $p(e_1, e_2; t_1, t_2) de_1 de_2$ is the probability of $f_1(t)$ lying between e_1 and $e_1 + de_1$ at time t_1 and τ seconds later at time t_2 between e_2 and $e_2 + de_2$. Since discrete interval noise is not a stationary process, the autocorrelation is a function of t_1 and τ rather than τ alone. One procedure often used to obtain a function of τ where nonstationary

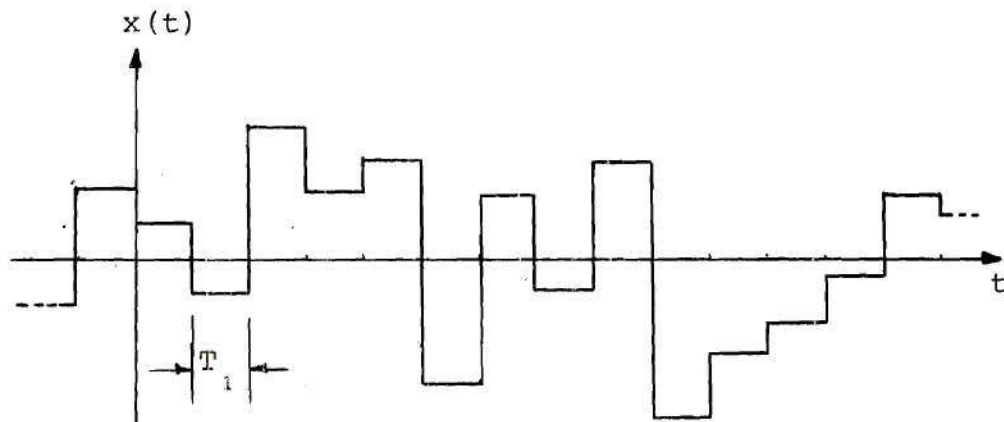


Figure 3. A Discrete Interval Noise Signal.

processes are concerned is to take a time average of the ensemble average. This leads to the result⁶

$$\left. \begin{aligned} \psi(\tau) &= \left(1 - \frac{|\tau|}{T_1}\right) \int_0^\infty x^2 p(x) dx & |\tau| \leq T_1 \\ \psi(\tau) &= 0 & |\tau| > T_1 \end{aligned} \right\} \quad (3-3)$$

If the signal has amplitudes of only $+\alpha$ and $-\alpha$, (discrete interval binary noise), then

$$p(x) = \frac{\delta(x - \alpha)}{2} + \frac{\delta(x + \alpha)}{2} \quad (3-4)$$

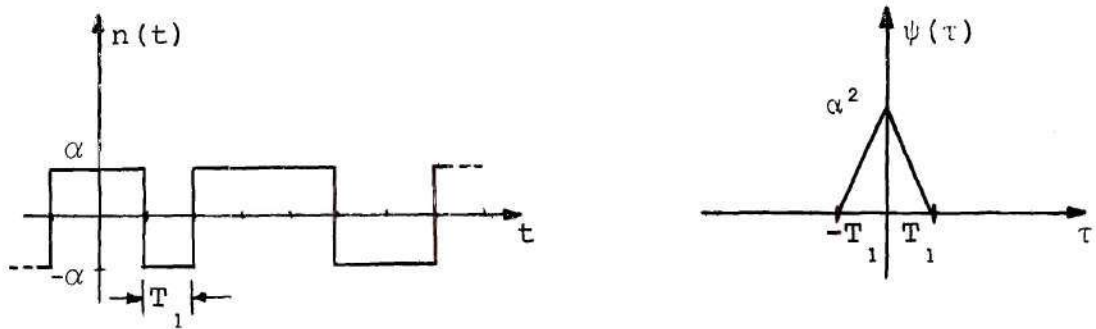
and the autocorrelation function can be written as

$$\left. \begin{aligned} \psi(\tau) &= \alpha^2 \left(1 - \frac{|\tau|}{T_1}\right) & |\tau| \leq T_1 \\ \psi(\tau) &= 0 & |\tau| > T_1 \end{aligned} \right\} \quad (3-5)$$

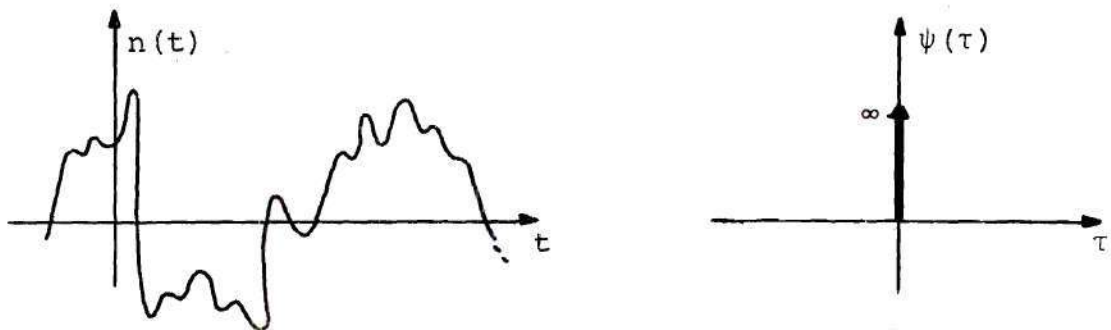
A sketch of discrete interval binary noise and its autocorrelation is shown in Figure 4(a). Figure 4(b) is a sketch of white noise and its autocorrelation function.⁴

Substitution of the expression for $\psi_{ii}(\tau)$, the autocorrelation of discrete interval binary noise, into Equation (2-22), gives

$$\psi_{iO}(\tau) = \alpha^2 \int_0^{\infty} h(x) \left(1 - \frac{|\tau - x|}{T_1}\right) dx \quad (3-6)$$



(a) Discrete Interval Binary Noise



(b) White Noise

Figure 4. Noise Signals and Autocorrelation Functions.

A graphical representation of this convolution is shown in Figure 5.

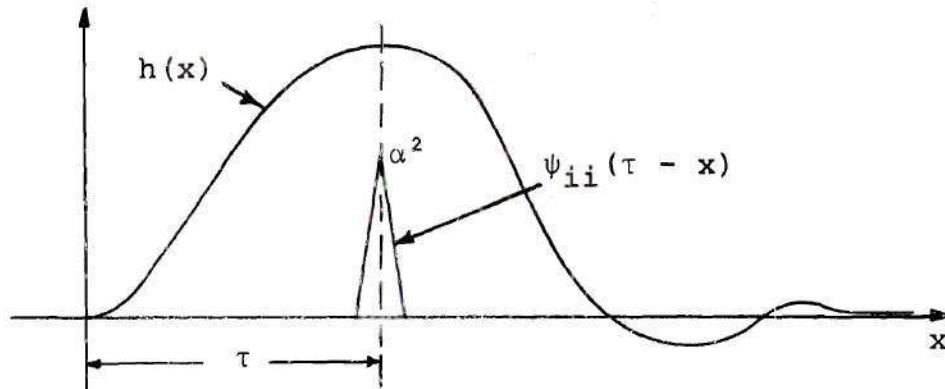


Figure 5. Graphical Representation of Convolution.

By substituting the general expression for the impulse response $h(t)$ for a linear system with single poles, Equation (3-7), into Equation (3-6) a comparison of the error can be made.

$$h(t) = \sum_{m=0}^M A_m e^{-a_m t} + \sum_{n=0}^N B_n e^{-a_n t} \sin(\omega_n t + \phi_n) \quad (3-7)$$

It can be shown that to minimize the error in $\psi_{i0}(\tau)$ the input noise signal must satisfy two conditions. (1) The duration T_1 of the smallest pulse must be much smaller than the time constant of the process. (2) The time T_1 must be much smaller than the oscillation time of the process.⁴

The final result for the impulse response is then given by

$$h(t) = \frac{\psi_{i0}(\tau)}{T_1 \psi_{ii}(0)} = \frac{\psi_{i0}(\tau)}{\alpha^2 T_1} \quad (3-8)$$

Periodic Pseudo-Random Noise

There are periodic binary sequences whose corresponding waveforms possess periodic autocorrelation functions of similar character as that of discrete interval binary noise. Such periodic sequences are not random but absolutely deterministic. They are known as *maximum-length shift register sequences*.

When a sequence is said to be random, information is inferred as to the process by which the sequence was generated rather than what the sequence actually looks like. However, statistical tests may be devised to determine how plausible the hypothesis is that a particular binary sequence was produced by a specified random process. Any sequence that passes a given set of tests for the plausibility of randomness can be referred to as a *pseudo-random sequence*. This is, of course, an *a posteriori* criterion which is neither necessary nor sufficient for true randomness, (which refers to the *a priori* circumstances of sequence generation).⁸

Binary sequences of ONE's and ZERO's can easily be generated through the use of a digital device known as a *shift register*. A shift register of degree n consists of n consecutive binary storage positions. The contents of each

position is shifted to the next position down the line in time to the count of a clock or other timing device. To prevent the shift register from emptying at the end of n clock pulses, a feedback term may be computed as a logical function of the contents of the n positions and fed back to the first position of the shift register. Figure 6 shows an example of such a device.

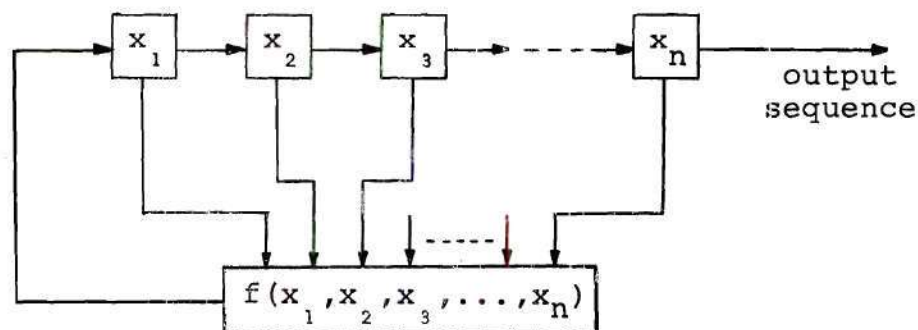


Figure 6. Shift Register of Degree n With Logical Feedback.

The output sequence is periodic with period p not exceeding $2^n - 1$. Those sequences with period equal to $2^n - 1$ are referred to as *maximum-length shift register sequences*.

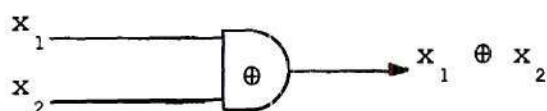
For every n there exists a maximum-length shift register sequence. They are all pseudo-random in the sense that they satisfy the following three randomness properties.⁸

- R-1. (The Balance Property) In each period the number of ONE's differs from the number of ZERO's by at most 1.

R-2. (The Run Property) Among the *runs* of ONE's and of ZERO's in each period, one-half the runs of each kind are of length one, one-fourth of each kind are of length two, one-eighth are of length three, and so on as long as these fractions give meaningful numbers of runs.

R-3. (The Correlation Property) If a period of the sequence is compared term by term with any cyclic shift of itself, the number of agreements differs from the number of disagreements by at most 1.

As an example of a maximum-length sequence generator, consider a shift register of degree 4 with feedback function equal to a *modulo 2* sum of the contents of the last two positions.



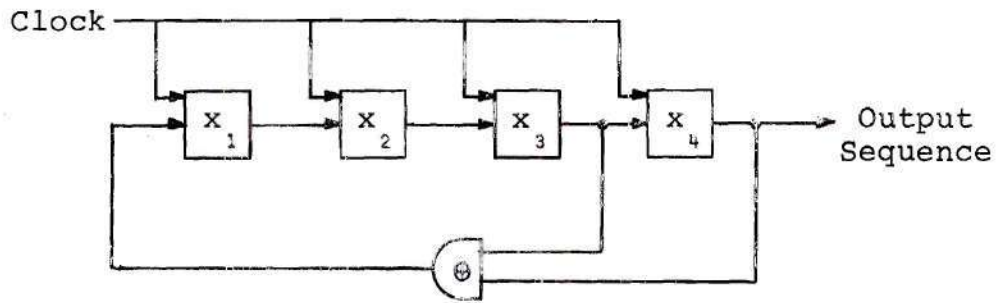
Symbol

| x_1 | x_2 | $x_1 \oplus x_2$ |
|-------|-------|------------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Truth Table

Figure 7. Modulo 2 Adder.

The symbol for a modulo 2 adder (exclusive OR function) and its truth table is shown in Figure 7. The sequence generator is shown in Figure 8(a).



(a)

| Clock Count | X ₁ | X ₂ | X ₃ | X ₄ | Generated Sequence |
|-------------|----------------|----------------|----------------|----------------|-------------------------------|
| 0 | 1 | 1 | 1 | 1 | |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 2 | 0 | 0 | 1 | 1 | 1 1 |
| 3 | 0 | 0 | 0 | 1 | 1 1 1 |
| 4 | 1 | 0 | 0 | 0 | 1 1 1 1 |
| 5 | 0 | 1 | 0 | 0 | 0 1 1 1 1 |
| 6 | 0 | 0 | 1 | 0 | 0 0 1 1 1 1 |
| 7 | 1 | 0 | 0 | 1 | 0 0 0 1 1 1 1 |
| 8 | 1 | 1 | 0 | 0 | 1 0 0 0 1 1 1 1 |
| 9 | 0 | 1 | 1 | 0 | 0 1 0 0 0 1 1 1 1 |
| 10 | 1 | 0 | 1 | 1 | 0 0 1 0 0 0 1 1 1 1 |
| 11 | 0 | 1 | 0 | 1 | 1 0 0 1 0 0 0 1 1 1 1 |
| 12 | 1 | 0 | 1 | 0 | 1 1 0 0 1 0 0 0 1 1 1 1 |
| 13 | 1 | 1 | 0 | 1 | 0 1 1 0 0 1 0 0 0 1 1 1 1 |
| 14 | 1 | 1 | 1 | 0 | 1 0 1 1 0 0 1 0 0 0 1 1 1 1 |
| 15 | 1 | 1 | 1 | 1 | 0 1 0 1 1 0 0 1 0 0 0 1 1 1 1 |

(b)

Figure 8. Maximum-Length Shift Register Sequence Generator.

Figure 8(b) indicates the states of x_1 , x_2 , x_3 and x_4 after each count and shows the sequence as it is being generated.

The generated sequence contains eight ONE's and seven

ZERO's which satisfies the randomness property R-1. Of the runs of ONE's in the sequence, there are two of length one, one of length two and one of length four. Thus, one-half are of length one and one-fourth are of length two, which satisfies the run property R-2. Similarly, one-half of the runs of ZERO's are of length one and one-fourth are of length two. Finally, the correlation property R-3 is satisfied in that by comparing term by term any cyclic shift of the sequence with itself, there are seven agreements and eight disagreements.

A periodic waveform corresponding to the sequence in the example and its autocorrelation function are shown in Figure 9. A ONE in the sequence corresponds to a waveform value of $+a$ for a period of time t_1 . Similarly a ZERO in the sequence corresponds to a waveform value of $-a$. The expression for the autocorrelation can be written as

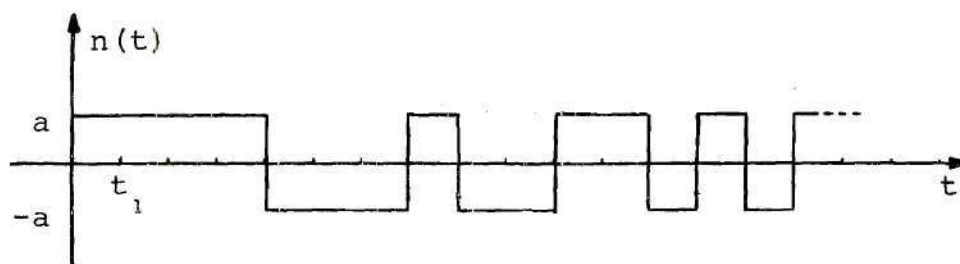
$$\left. \begin{aligned} \psi_{nn}(\tau) &= a^2 \left(1 - \frac{16}{15} \frac{|\tau|}{t_1} \right) & pT - t_1 \leq \tau \leq pT + t_1 \\ \psi_{nn}(\tau) &= \frac{1}{15} a^2 & \text{elsewhere} \end{aligned} \right\} \quad (3-9)$$

where T is the time period of the sequence, $t_1 = \frac{T}{15}$ and p is any integer.

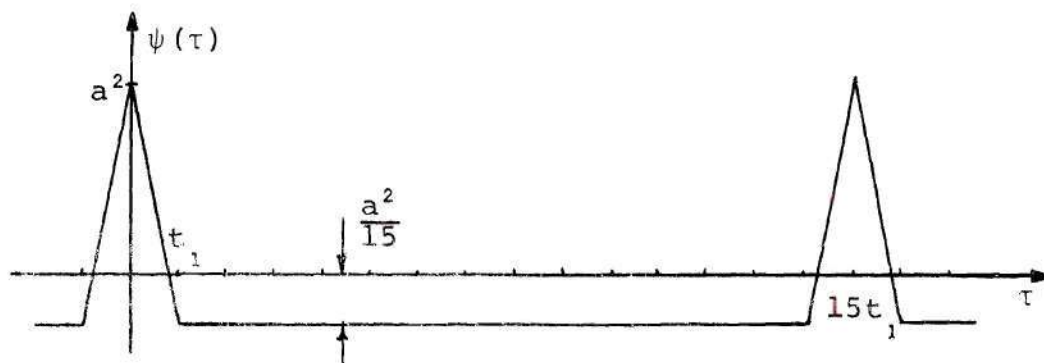
For the more general case of a maximum length sequence

1 1 1 1 0 0 0 1 0 0 1 1 0 1 0

(a) Sequence



(b) Waveform



(c) Autocorrelation

Figure 9. Pseudo-Random Sequence, Waveform and Autocorrelation.

with $T = (2^n - 1)t_1$, the autocorrelation can be written as⁴

$$\left. \begin{aligned} \psi_{nn}(\tau) &= a^2 \left[1 - \left(\frac{2^n}{2^n - 1} \right) \frac{|\tau|}{t_1} \right] & pT - t_1 \leq \tau \leq pT + t_1 \\ \psi_{nn}(\tau) &= \left(\frac{1}{2^n - 1} \right) a^2 & \text{elsewhere} \end{aligned} \right\} \quad (3-10)$$

In order to approximate the autocorrelation function of random discrete interval binary noise, $\psi_{nn}(\tau)$ must be zero outside the interval $pT - t_1 < \tau < pT + t_1$. This can be accomplished by choosing the waveform levels as $+b$ and $-c$ instead of $+a$ and $-a$ where $b + c = 2a$ and⁴

$$b = \frac{2}{2^n - 1} \left(2^{n-1} - 1 + \sqrt{2^{n-2}} \right) a \quad (3-11)$$

$$c = \frac{2}{2^n - 1} \left(2^{n-1} - \sqrt{2^{n-2}} \right) a \quad (3-12)$$

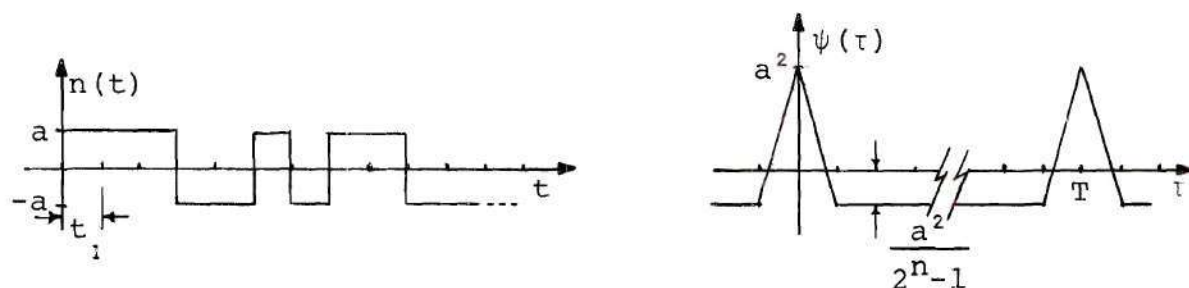
The autocorrelation under these conditions is given by

$$\left. \begin{aligned} \psi_{nn}(\tau) &= \frac{2^n}{2^n - 1} a^2 \left(1 - \frac{|\tau|}{t_1} \right) & \text{for} & & (3-13) \\ & & (pT - t_1) < \tau < (pT + t_1) \\ \psi_{nn}(\tau) &= 0 & \text{elsewhere} \end{aligned} \right\}$$

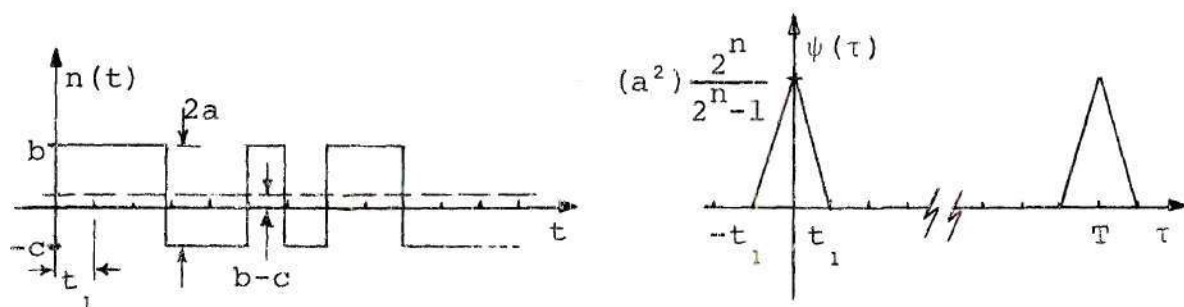
This effect of adding a DC component of

$$b - c = \frac{2}{2^n - 1} \left(2\sqrt{2^{n-2}} \right) a \quad (3-14)$$

to the periodic noise signal is illustrated in Figure 10.



(a) Without DC Component



(b) With DC Component

Figure 10. Periodic Pseudo-Random Binary Noise
Autocorrelation Functions.

CHAPTER IV

HEURISTIC EXAMPLE OF CORRELATION METHOD

Introduction

At this time an example will be presented to illustrate intuitively the correlation technique being presented. The system on which the impulse response measurement is to be made is assumed to be linear. Thus, the output response to a square pulse input will closely approximate the system impulse response as the width of the pulse is made smaller. The example will illustrate that the crosscorrelation process can be thought of in terms of an averaging of a large number of small narrow pulse responses. Because of the unique properties of binary pseudo-random noise, this averaging is done during one impulse response time (the time it takes for $h(t)$ to effectively die out).

In the example to follow, the output waveforms of the system will be converted to digital form. This is to illustrate the basic digital technique for performing the cross-correlation operation.

Finite Pulse Response Approximation

As mentioned previously, one method that can be used to obtain an approximation of the impulse response of a system is to apply a "narrow" pulse to the input. By narrow

it is meant that the time duration of the pulse be much smaller than the time constant of the process. By decreasing the pulse width and thus more closely approximating an ideal impulse, the output more closely approximates the impulse response of the system. A change in the amplitude of the pulse results only in a proportional change in the output approximation of $h(t)$. This is due to the linearity assumption of the system.

In Figure 11 three inputs to a linear system and their respective outputs are shown. The input $x_1(t)$ is a unit impulse and therefore the output $y_1(t)$ is the impulse response $h(t)$. Input $x_2(t)$ is a narrow pulse with unit area. As Δ approaches zero $x_2(t)$ approaches a unit impulse and $y_2(t)$ approaches the impulse response. Since $x_3(t)$ is identical to $x_2(t)$ except for a magnitude scaling, $y_3(t)$ is identical to $y_2(t)$ except for the same magnitude scaling.

If a system has noise or some other signal present at the input, an averaging technique can be used to improve the result of the pulse response. For example, consider an *on-line* process identification or evaluation. Here the normal operation of the system should be disturbed as little as possible. Relatively low level test pulses must then be used and the output response averaged over a correspondingly large number of samples. The primary disadvantage is the total time involved in obtaining the averaged response. The maximum output sample rate (or the maximum rate of input

pulses) will depend on the time constant of the system. That is to say the pulse response must die out or become negligible before another pulse is applied to the input.

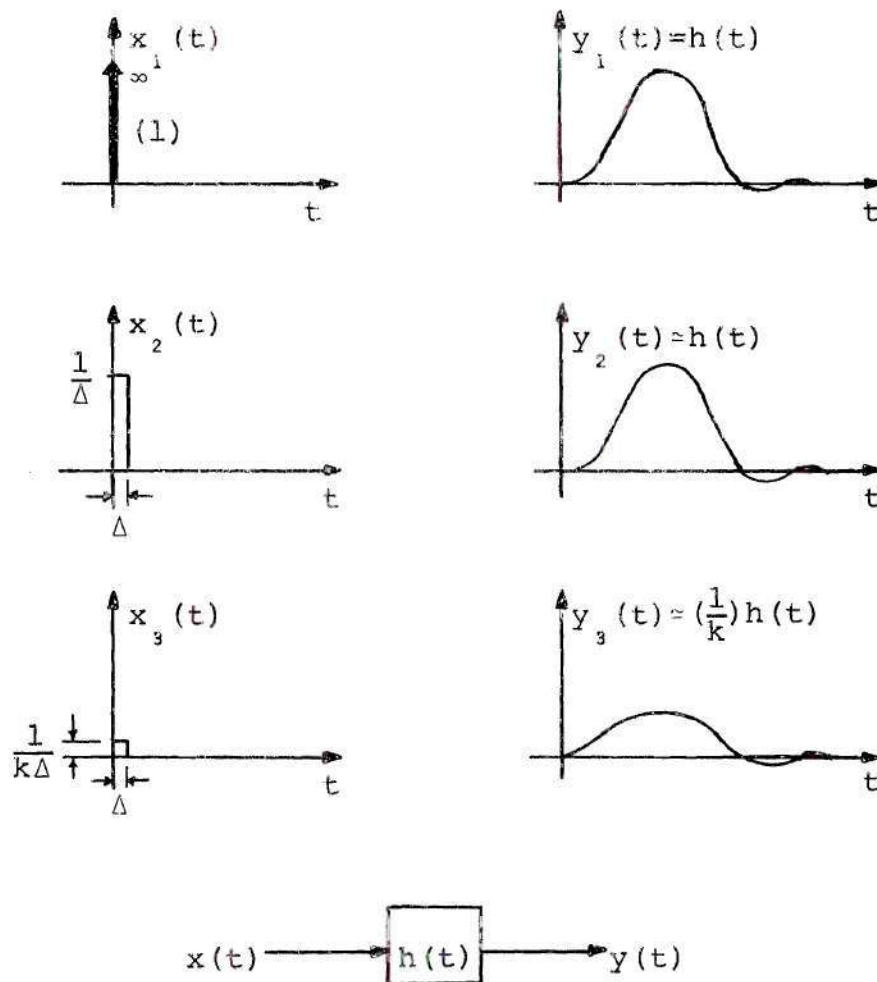


Figure 11. Impulse and Pulse Inputs
To System And Associated Responses.

System Response To Periodic Binary Pseudo-Noise

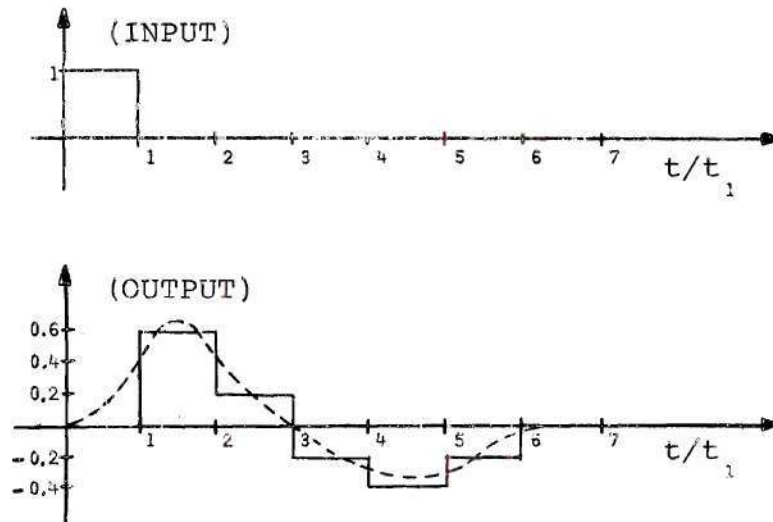
A periodic binary pseudo-random noise signal can be considered as the superposition of single pulses occurring at discrete time intervals. The output response of a linear system to such a binary noise signal can then be considered as the superposition of single pulse responses occurring at the same discrete time intervals.

To illustrate this first consider the quantization of the pulse response as shown in Figure 12(a).

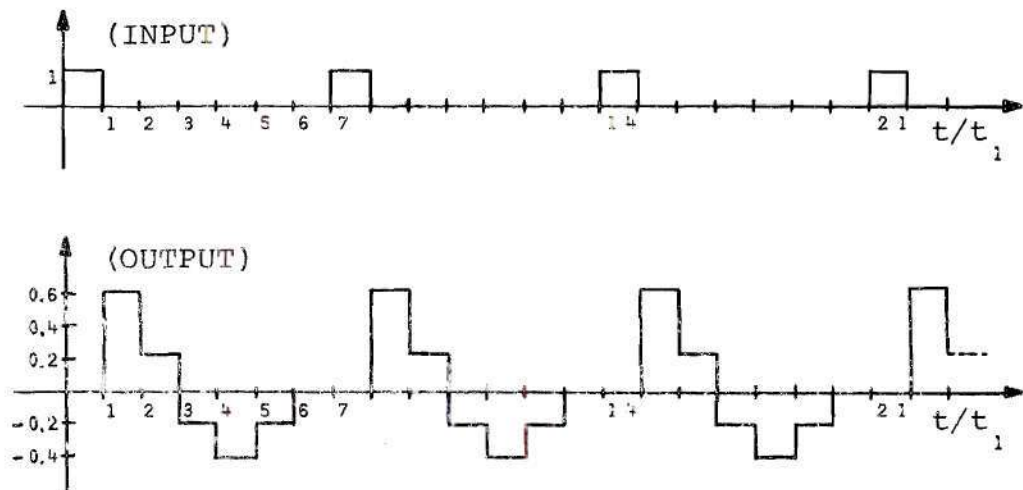
For digital calculations the system output must be converted from an analog waveform to a series of discrete binary coded magnitudes. The actual binary coding will be discussed in the next chapter. In the example the notation h_0 will denote the quantized value of $h(t)$ (the pulse response approximation of the actual impulse response) in the time interval $(0, t_1)$. Similarly, h_1 will represent the quantized value of $h(t)$ in the interval (t_1, t_2) , and so forth. Because the purpose of the example is to illustrate the technique involved, only a few amplitude quantization levels will be used in order to simplify the calculation.

The quantized output response to a series of periodic input pulses is shown in Figure 12(b). As long as the input pulses occur at intervals greater than $7t_1$, the series of output responses will be non-overlapping.

Now consider a period of seven, pseudo-random noise signal $n(t)$ as the system input. Figure 13 illustrates the



(a) Single Pulse Input and Output Response



(b) Periodic Pulse Input and Output Response

Figure 12. Pulse Input and Associated Output Response

use of the superposition principle to determine the output response $y(t)$. First, the input signal is considered to be the sum of seven component signals. Each component signal is a series of either positive or negative unit pulses of width t_1 and periodic with period $7t_1$. The corresponding output components will be periodic repetitions of a single pulse response. The quantized output waveform can thus be constructed by summing these output components. By referring to Figure 13 it can be seen that

$$y_0 = h_0 + h_6 + h_5 - h_4 - h_3 + h_2 - h_1 \quad (4-1)$$

and similarly

$$y_1 = h_1 + h_0 + h_6 - h_5 - h_4 + h_3 - h_2 \quad (4-2)$$

A convenient form of the relationship between the quantized values of $y(t)$ and $h(t)$ is given by the matrix equation (4-3).

The values of $n(t)$ and the quantized values of $h(t)$ and $y(t)$ calculated for the example are listed in Table 1.

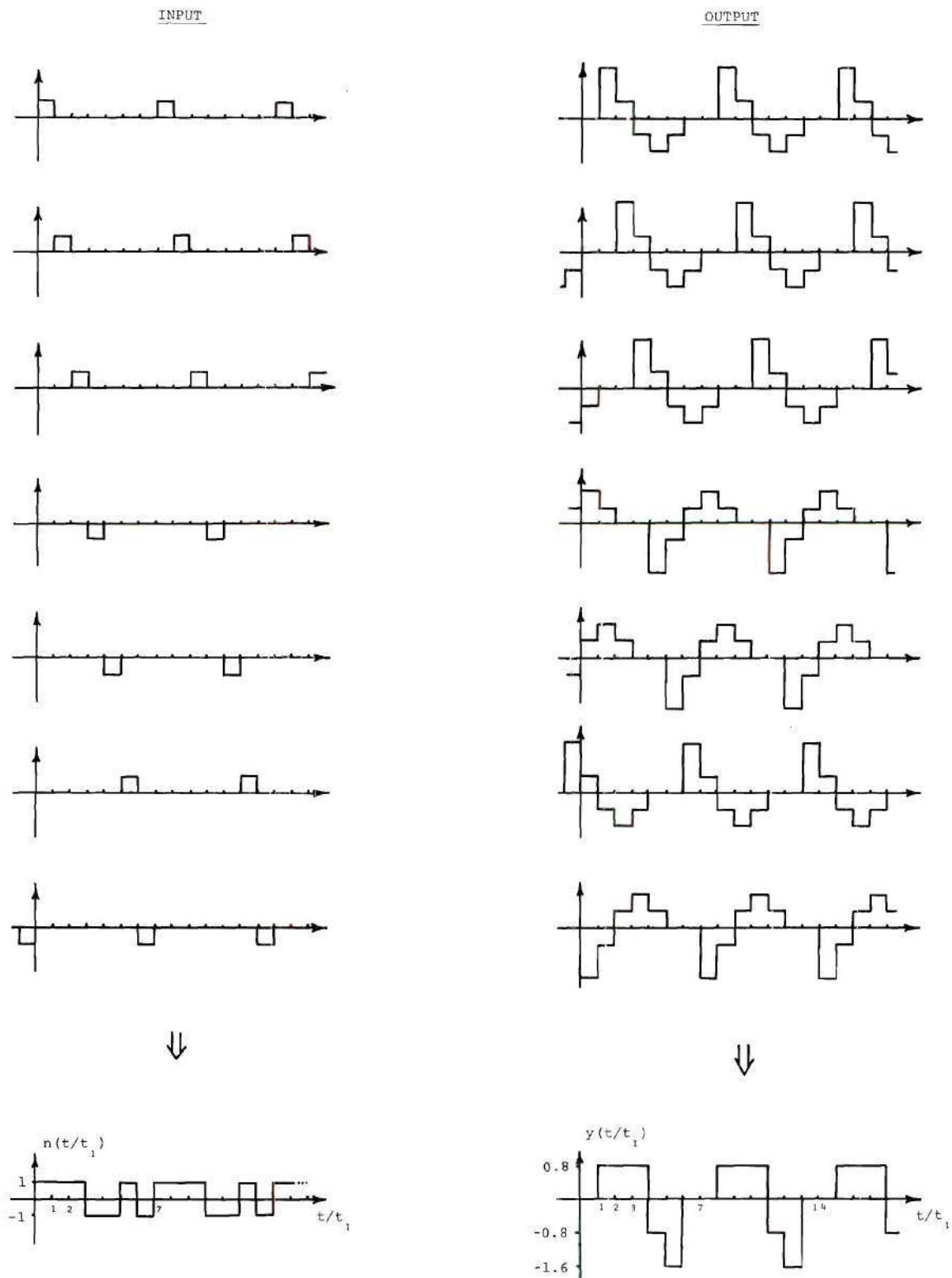


Figure 13. Pseudo-Random Noise Response By Superposition.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 \\ -1 & 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \end{bmatrix} \quad (4-3)$$

Table 1. Quantized Values of $n(t)$, $h(t)$, and $y(t)$

| $n(t)$ | $h(t)$ | $y(t)$ |
|------------|--------------|--------------|
| $n_0 = +1$ | $h_0 = 0$ | $y_0 = 0$ |
| $n_1 = +1$ | $h_1 = 0.6$ | $y_1 = 0.8$ |
| $n_2 = +1$ | $h_2 = 0.2$ | $y_2 = 0.8$ |
| $n_3 = -1$ | $h_3 = -0.2$ | $y_3 = 0.8$ |
| $n_4 = -1$ | $h_4 = -0.4$ | $y_4 = -0.8$ |
| $n_5 = +1$ | $h_5 = -0.2$ | $y_5 = -1.6$ |
| $n_6 = -1$ | $h_6 = 0$ | $y_6 = 0$ |

Digital Crosscorrelation of Input Noise and Output Response

Using the approach of crosscorrelation as a shifting and averaging process, $\psi_{ny}(\tau)$ can be calculated for values of $\tau = kt_1$, where k is any integer, from the equation

$$\psi_{ny}(kt_1) = \frac{1}{7t_1} \sum_{i=0}^6 n_i y_{i+k} \quad (4-4)$$

For example

$$\begin{aligned} \psi_{ny}(0) &= \frac{1}{7t_1} [0 + 0.8 + 0.8 - 0.8 + 0.8 - 1.6 - 0] \quad (4-5) \\ &= \frac{1}{7t_1} (0) = 0 \end{aligned}$$

Table 2 lists the values of $7t_1 \psi_{ny}(\tau)$ and $\frac{7t_1}{8} \psi_{ny}(\tau)$ for the example. From this table it can be seen that the final expression for the impulse response is then

$$h(\tau) = \frac{7t_1}{8} \psi_{ny}(\tau) \quad (4-6)$$

Table 2. Values of Crosscorrelation.

| τ | $7t_1 \psi_{ny}(\tau)$ | $\frac{7t_1}{8} \psi_{ny}(\tau)$ |
|--------|------------------------|----------------------------------|
| 0 | 0 | 0 |
| t_1 | 4.8 | 0.6 |
| $2t_1$ | 1.6 | 0.2 |
| $3t_1$ | -1.6 | -0.2 |
| $4t_1$ | -3.2 | -0.4 |
| $5t_1$ | -1.6 | -0.2 |
| $6t_1$ | 0 | 0 |

In order to see more clearly how the properties of pseudo-random noise signals enable the recovering of $h(t)$ from one period of the output, consider the matrix equation

$$(7t_1) \begin{bmatrix} \psi(0) \\ \psi(t_1) \\ \psi(2t_1) \\ \psi(3t_1) \\ \psi(4t_1) \\ \psi(5t_1) \\ \psi(6t_1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ -1 & -1 & 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} \quad (4-7)$$

which relates $\psi_{ny}(\tau)$ with $y(t)$. Substitution of Equation 4-3 into this expression yields

$$(7t_1) \begin{bmatrix} \psi(0) \\ \psi(t_1) \\ \psi(2t_1) \\ \psi(3t_1) \\ \psi(4t_1) \\ \psi(5t_1) \\ \psi(6t_1) \end{bmatrix} = \begin{bmatrix} 7 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 7 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 7 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 7 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 7 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 7 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 7 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \end{bmatrix} \quad (4-8)$$

The symmetry of the 7 by 7 matrix is a direct result of the autocorrelation property of the pseudo-random noise

signal with values of ± 1 . From this last equation a general expression for the autocorrelation can be written as

$$\psi(kt_1) = \frac{1}{7t_1} \left(8h_k - \sum_{i=0}^6 h_i \right) \quad (4-9)$$

The last term represents the area under the signal $h(t)$ which was zero in the example.

One approach in eliminating this constant error term is to choose the noise signal magnitudes as $+b$ and $-c$ according to Equations (3-13) and (3-14). In this case the peak to peak value of the noise signal remains $2a$.

For a signal with period $T = (2^n - 1)t_1$ (generated from a degree n shift register), equation (4-8) becomes

$$(T) \begin{bmatrix} (0) \\ (t_1) \\ (2t_1) \\ \vdots \end{bmatrix} = \{at_1(2^{n-1})\}^2 \begin{bmatrix} 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \end{bmatrix} \quad (4-10)$$

The general result for h_k is then given by

$$h_k = \left(\frac{2^n - 1}{2^{n-1}} \right)^2 \frac{\psi(kt_1)}{a^2 T} \quad (4-11)$$

Another method of eliminating the constant error term in Equation (4-9) is to calculate its value and remove it from each value of $\psi(kt_1)$. Equation (4-9) can be written in general as

$$\psi(kt_1) = \frac{1}{T} \left[2^n h_k - \sum_{i=0}^{2^n-2} h_i \right] \quad (4-12)$$

where $T = 2^n - 1$. Rearranging this equation yields

$$\frac{T \psi(kt_1)}{2^n} = h_k - \frac{1}{2^n} \sum_{i=0}^{2^n-2} h_i \quad (4-13)$$

The third term is the error to be calculated and is found to be

$$\frac{1}{2^n} \sum_{i=0}^{2^n-2} h_i = \frac{\sum_{i=0}^{2^n-2} \psi(it_1)}{2^n \left(\frac{2^n}{T} - T \right)} \quad (4-14)$$

The final result for h_k is then given by

$$h_k = \frac{T \psi(kt_1)}{2^n} - \frac{\sum_{i=0}^{2^n-2} \psi(it_1)}{2^n \left(\frac{2^n}{T} - T \right)} \quad (4-15)$$

CHAPTER V

DESIGN AND CONSTRUCTION OF PROTOTYPE

Introduction

In the previous chapters the background and theoretical basis for the correlation measurement of system impulse responses has been set forth. The next step is the design and construction of a working prototype.

The function of the prototype is to generate a pseudo-random noise signal to be added to the input of a test system and then to crosscorrelate the system output response with this noise signal. A six position shift register with mod 2 feedback is used to generate a pseudo-random noise signal. The period of this signal is thus $(2^6 - 1)t_1 = 63t_1$, where t_1 is the duration of the narrowest pulse. A scaling circuit is incorporated in the prototype to provide for several values of t_1 so that the impulse response can be spread out over one noise signal period. This enables the pulse width of the noise autocorrelation function to be made as narrow as possible with respect to the impulse response.

The crosscorrelation process is done digitally in the prototype. The analog output of the test system is sampled and converted to a series of binary numbers. An eleven bit *sign plus two's complement* binary coding is used in the A/D

(analog to digital) conversion and in the digital calculation of the crosscorrelation. Table 3 illustrates the eleven bit sign plus two's complement binary code.

Table 3. Sign Plus Two's Complement Code.

| <u>Binary</u> | | | | | | | | | | | <u>Decimal Equivalent</u> |
|---------------|------------------------------|---|---|---|---|---|---|---|---|---|-------------------------------|
| <u>Sign</u> | <u>Least Significant Bit</u> | | | | | | | | | | |
| ↓ | ↓ | | | | | | | | | → | |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1023 |
| | | | | | | | | | | | . |
| | | | | | | | | | | | . |
| | | | | | | | | | | | . |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | -1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | -2 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | -3 |
| | | | | | | | | | | | . |
| | | | | | | | | | | | . |
| | | | | | | | | | | | . |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1024 |

Finally, the impulse response is converted from digital to analog form for cathode-ray oscilloscope display. Figure 14 illustrates the basic process.

Before considering the actual logic design of the digital prototype, a description of the algorithm used to calculate the crosscorrelation, and hence the impulse response, will be discussed.

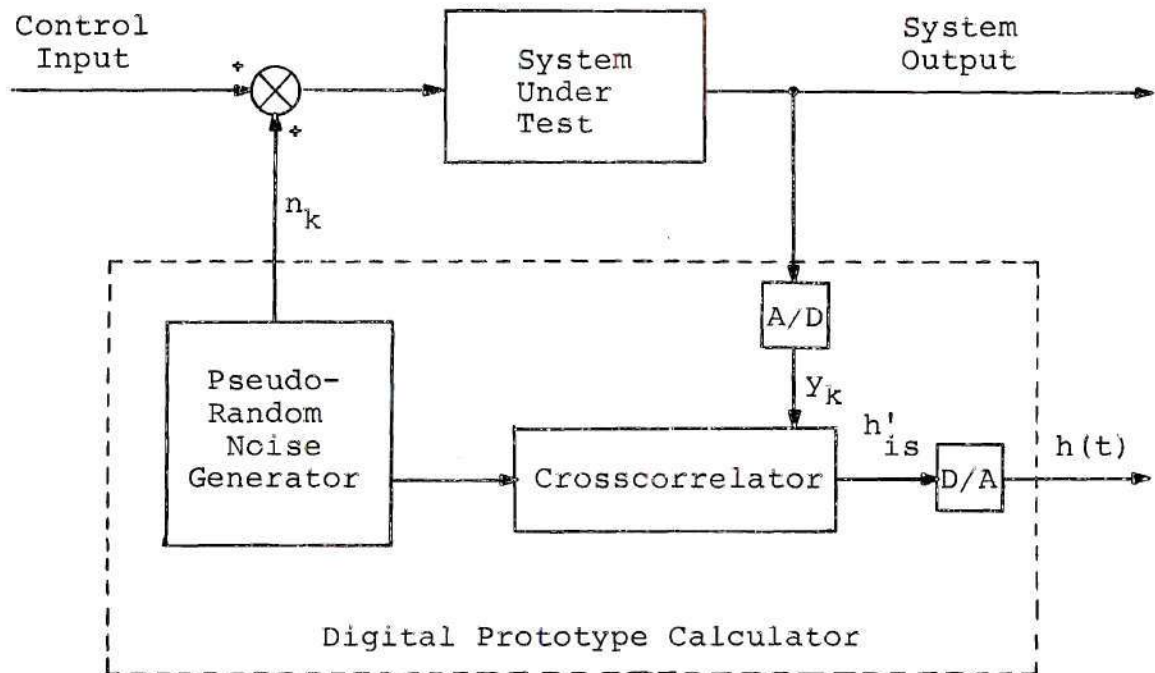


Figure 14. Digital Prototype and Test System.

Prototype Calculator Algorithm

The prototype calculator has one input and two outputs. Referring to Figure 14, the input y_k is the digital conversion of the k^{th} test system output sample. The output n_k is k^{th} value of the noise sequence being added to the test system input. And the output h'_{is} is the set of impulse response values from the previous calculation. Those values of h'_{is} are being stored for output display while the present calculation process is in progress.

The use of a noise test signal with values of ± 1 greatly reduces the complexity of the crosscorrelation computation

process. Under these conditions there is a constant error term produced as discussed in Chapter IV. This constant term could be calculated and removed through the use of additional logic circuitry. However, this type of constant error does not significantly affect the results of the technique being presented here. It is primarily for this reason that the prototype is designed to calculate and display only the values of $63t_1 \psi(kt_1)$ for k a positive integer. These values are proportional to the values of h_k except for the constant term.

The digital crosscorrelation can be written as

$$\psi(kt_1) = \frac{1}{63t_1} \sum_{i=0}^{62} n_{i-k} y_i \quad (5-1)$$

Since the noise sequence is periodic the values of n_{j+63r} is equivalent to n_j for r equal to any integer.

The algorithm for computing $63t_1 \psi(kt_1)$, which will be denoted by h'_k , can be more easily understood by considering the following set of equations.

$$\begin{aligned} h'_0 &= n_0 y_0 + n_1 y_1 + n_2 y_2 + \dots + n_{62} y_{62} \\ h'_1 &= n_{62} y_0 + n_0 y_1 + n_1 y_2 + \dots + n_{61} y_{62} \\ h'_2 &= n_{61} y_0 + n_{62} y_1 + n_0 y_2 + \dots + n_{60} y_{62} \\ &\vdots \\ h'_{62} &= n_1 y_0 + n_2 y_1 + n_3 y_2 + \dots + n_0 y_{62} \end{aligned} \quad (5-2)$$

A flowchart of the algorithm is shown in Figure 15.

After initialization the inputs and outputs are updated. The noise output is set to the value n_0 . The first sample of the test system output is made and there is no display of previously calculated h'_{1s} because there was no previous calculation. However, there will be a display during subsequent calculations. *These I/O values of y and n will not change until this point is again reached in the flowchart.* At this point, the first term of each of the values of h' is calculated (refer to Equation 5-2). That is, the first term to be calculated is $n_0 y_0$, then $n_{62} y_0$, followed by $n_{61} y_0$, and so forth until $n_1 y_0$ has been calculated. At this point $i = k + 1$, indicating that a component of h'_{62} has been calculated. The prototype output noise value to the system under test is updated to n_1 and a value of y_1 is taken from the test system output. Now the second component of each of the values of h' is calculated and added to the first term. After $n_2 y_1$ has been calculated and added to $n_1 y_0$, the I/O values of the prototype are again updated for the calculation of the third component of the h' values. It should be noted that all 63 values of the pseudo-random noise sequence are used each time the prototype calculates a component of the values of h' . However, as mentioned before, the values of the noise sequence that are added to the test system input are updated only after each of these component calculations. Since the values of the noise are ± 1 , the

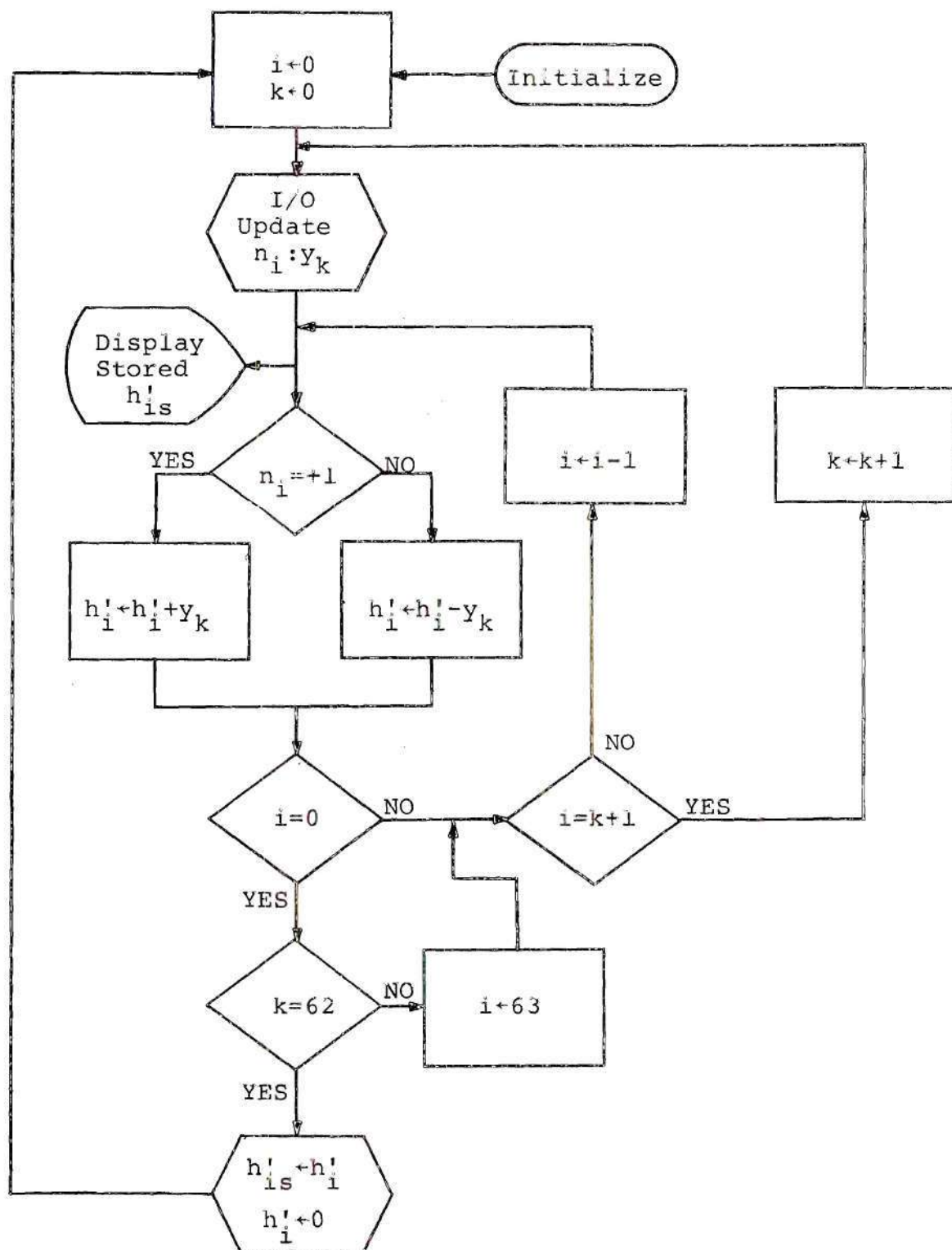


Figure 15. Prototype Calculator Algorithm.

calculation of a value of h' is simply a summing of the values of y with appropriate signs. When the last component of h'_{62} is calculated and added to the total of the other components, the calculation of a quantized impulse response has been completed. These 63 values of h' are then stored for display during the next calculation operation. The stored values are denoted by h'_{is} in the flow chart. At this point the calculation begins again.

Basic Overall Design

In this section a description of the basic overall design will be presented. A detailed discussion of the components and their operation follows in the next section.

Figure 16 illustrates the block diagram for the prototype design. Data is calculated and stored in a recirculating loop memory which consists of a two millisecond delay line and an eleven bit shift register denoted by register A in the diagram. Information in the loop consists of 63 word pairs. A word pair is made up of two 11 bit binary numbers as illustrated in Figure 17. The right most number in the pair is a value of the previously calculated impulse response. It is stored in this position and displayed during the calculation process. This calculation process takes 63 recirculations of the loop data and is being carried out in the 11 bit words immediately preceeding the stored words. During the 64th recirculation, the newly calculated values are shifted

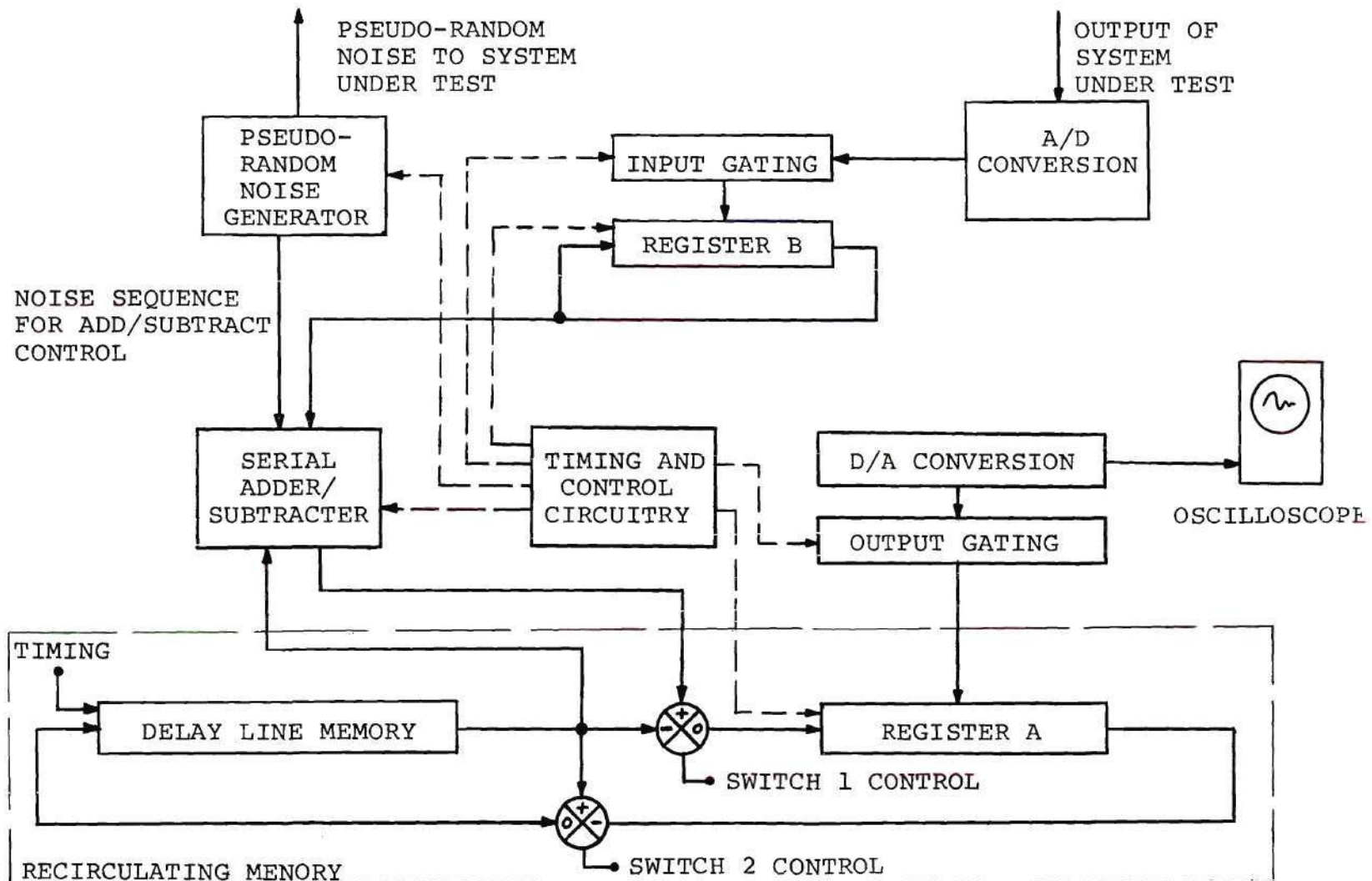
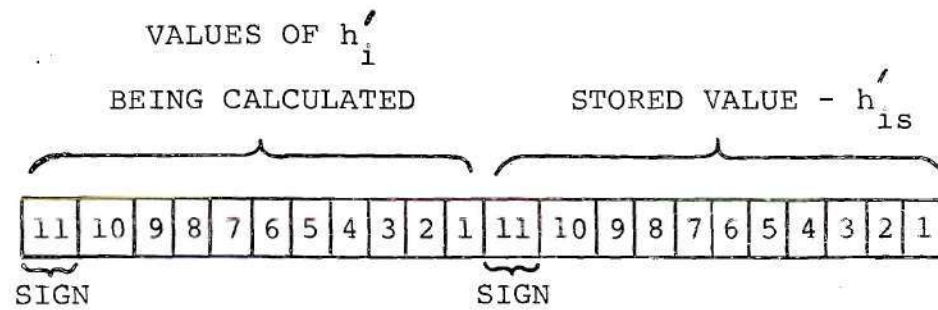


Figure 16. Block Diagram of Prototype Design.



WORD PAIR FORMAT

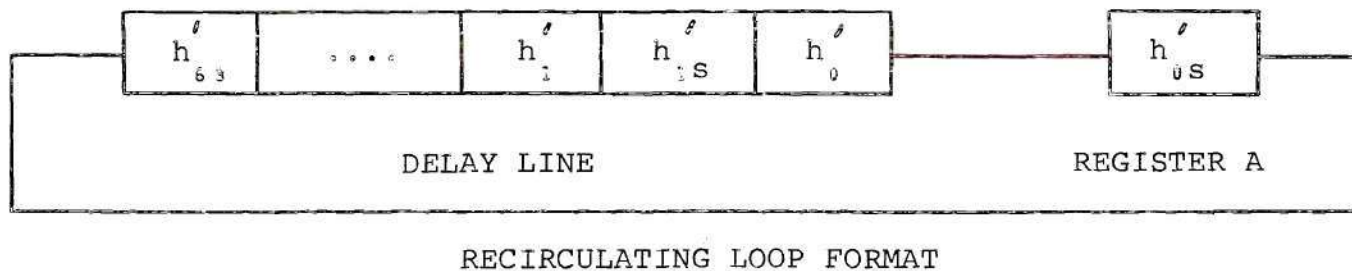


Figure 17. Word and Recirculating Loop Formats.

forward to replace the stored values and calculation of a new set is begun once again.

An impulse response curve is traced on the oscilloscope screen once each recirculation. Each time a previously calculated and stored word is in register A its value is transferred through the output gating to the D/A converter. This value is held by the gating circuitry until the next stored word is in register A, at which time the output to the oscilloscope is updated again.

For the calculation process the output value of the system under test is needed in digital form. The input gating circuit samples the A/D conversion of this value once each recirculation and sets it into the 11 bit shift register B. These samples are the values of y_0 , y_1 , y_2 and so forth of Equation (5-2). Register B forms an 11 bit recirculating loop, so that the values of y are available in serial form (one bit at a time, least significant bit first). When the calculation process begins, there is a value of zero set in the word space for each of the values of h' to be calculated. On the first recirculation of the calculation process the first value of y , that is y_0 , is added to or subtracted from the initial zero values of h' . This constitutes the first component of each of the values of h' that are being calculated. The add/subtract control is determined by the values of the pseudo-random noise sequence. On the second recirculation the second component of the h' values is added to the

total by appropriately adding or subtracting y_1 . The process is completed after the 63rd recirculation.

The purpose of switch 1 is to allow the stored words to recirculate unaltered during the calculation process. When the switch is in the (-) position, a stored word passes from the delay line through the switch into register A. When it is in the (+) position, a value of y is appropriately combined in the adder/subtractor with a word from the delay line and this value passes through the switch into register A. Switch 2, normally in the (-) position for the calculation and display mode, controls the shifting mode at the end of the calculation process. At the beginning of the shift mode, switch 1 changes to the (-) position and remains in this position for the entire recirculation. Switch 2 changes to the (+) position for the first 11 bit word, allowing the newly calculated h'_0 to bypass register A and enter the delay line in the position of the stored h'_{0s} . The word in register A is set to zero and the switch changes to the (-) position allowing the zero value to enter into the first calculation space. Switch 2 continues to alternate passing the newly calculated values of h' and values of zero into the delay line for the remainder of the recirculation.

Logic Element Descriptions

The prototype was constructed using Fairchild, Motorola and Systems Engineering Laboratories integrated circuit components of the following types:

1. J-K Flip-Flops (Set and Reset), Systems Engineering Laboratories, Type 8528 Micrologic Modules.

2. J-K Flip-Flops (DC reset only), Motorola RTL logic type MC 791 P.

3. Inverting Buffer Amplifiers, Systems Engineering Laboratories type 8501 Micrologic Modules.

4. Two-Nor gate, Systems Engineering Laboratories type 8527 Micrologic Modules.

5. Quad 2-Input Gates, Motorola RTL logic type MC 724 P.

6. Triple 3-Input Gates, Motorola RTL logic type MC 792 P.

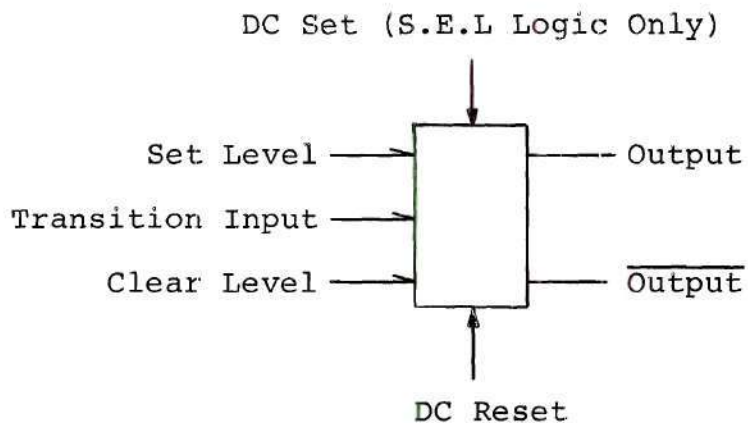
7. Quad 2-Input Expanders, Motorola RTL logic type MC 785 P.

8. Hex Inverters, Motorola RTL type MC 789 P.

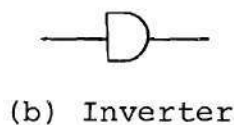
9. High-gain Operational Amplifiers, Fairchild type μ L709.

10. Dual high-speed Analog Comparators, Fairchild type μ L711.

The symbols used for the J-K flip-flops, NOR gates, buffers, and inverters are shown in Figure 18. Expanders are used to create multi-input gates. Much of the logic used in the prototype was in the form of general purpose printed circuit cards. Most of these cards were constructed using the Motorola 700 series integrated circuit logic. Photographs of these general purpose two-sided cards are shown in



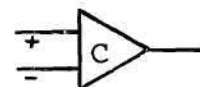
(a) Symbol for J-K Flip-flop



(b) Inverter



(e) 2-Input Gate



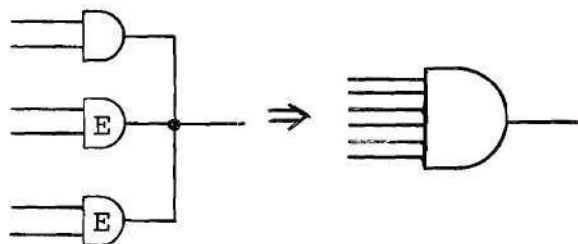
(g) Voltage Comparator



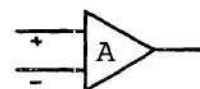
(c) Inverting Buffer



(f) 3-Input Gate



(d) Expanders Used to Form Multiple Input Gate



(h) Operational Amplifier

Figure 18. Logic Symbols.

Figure 19. Several special purpose cards were also constructed and are shown in Figures 20 and 21. Details of these cards are discussed in the following section.

Details of Design and Construction

This section includes a discussion of the subsystems that make up the prototype calculator. The function and design of these subsystems will be covered as well as their interconnections. Finally, the actual construction of the prototype will be discussed.

The circuit for the crystal clock oscillator that provides the basic timing pulses to control the overall system is shown in the appendix. The output waveforms of the clock are shown in Figure 22. Two DC pulses are provided for DC setting during a clock period. The clock frequency used is 687 KHz. A photograph of the clock card is shown in Figure 20.

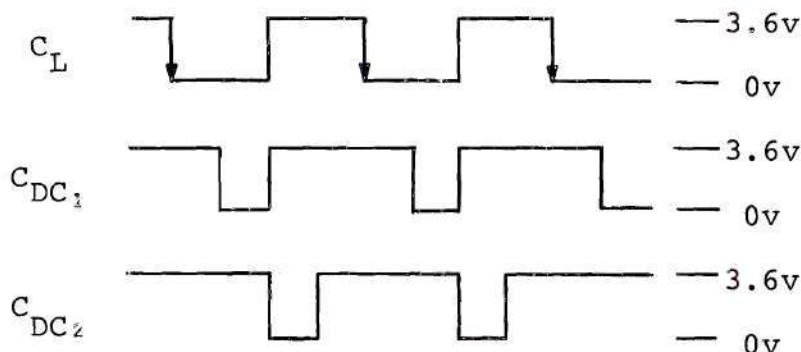
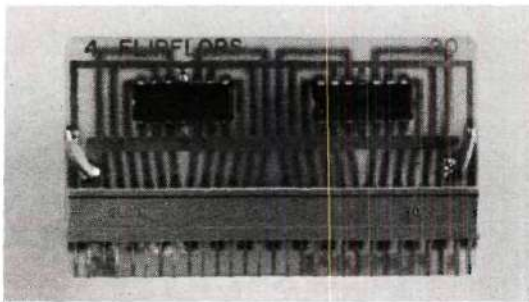
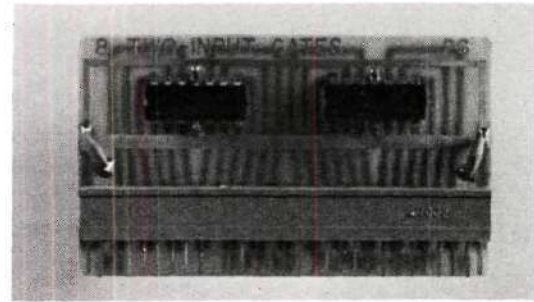


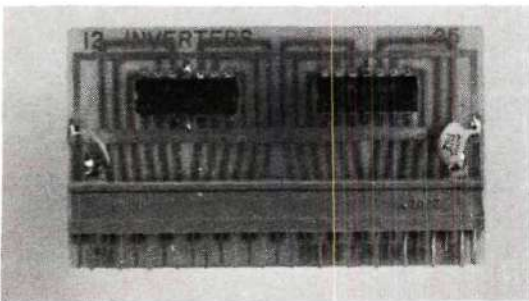
Figure 22. Crystal Clock Waveforms.



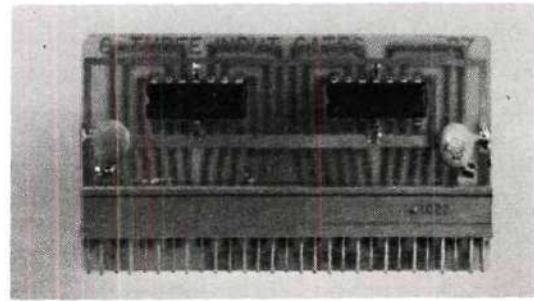
Four J-K Flip-Flops



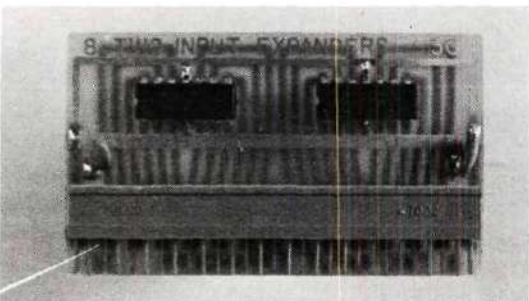
Eight - Two Input Gates



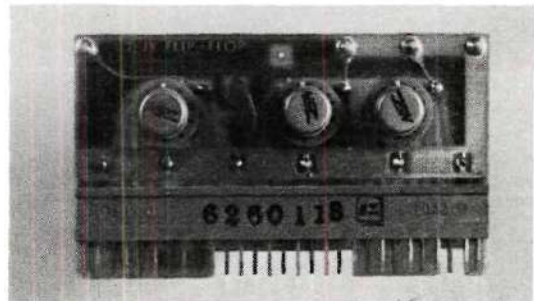
Twelve Inverters



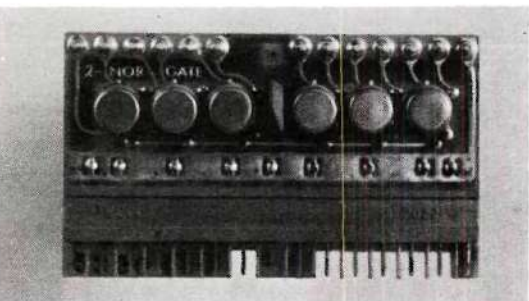
Six - Three Input Gates



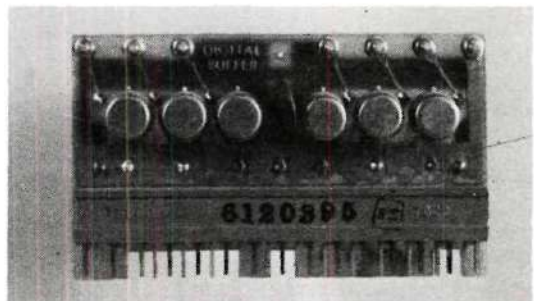
Eight - Two Input Expanders



Three J-K Flip-Flops

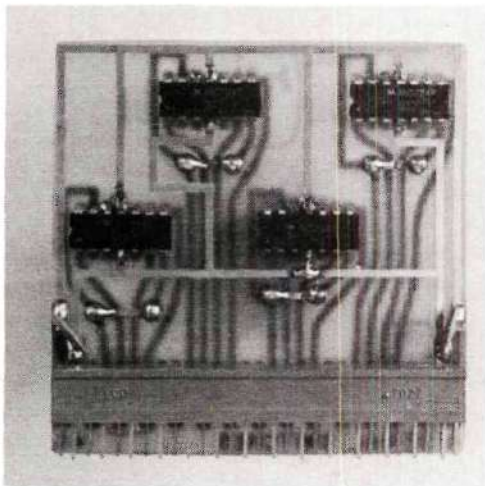


Two - NOR Gates

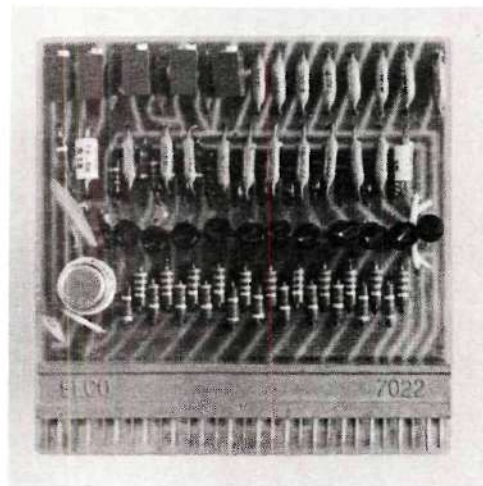


Six Buffers

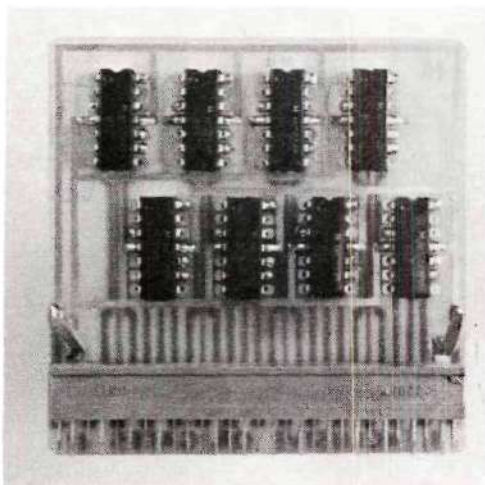
Figure 19. General Purpose Logic Cards.



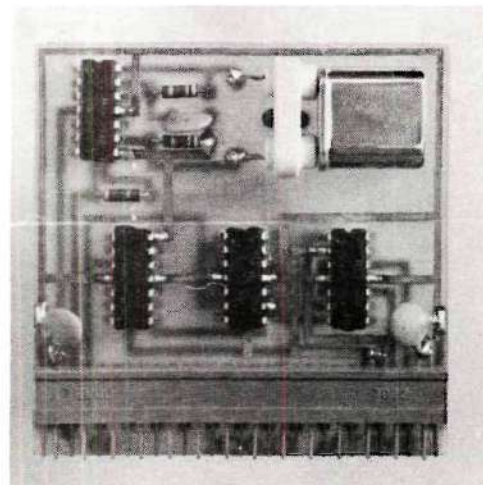
Serial Switches



Eleven Bit Ladder Decoder



Eight Exclusive OR Gates



Crystal Clock

Figure 20. Special Purpose Logic Cards.

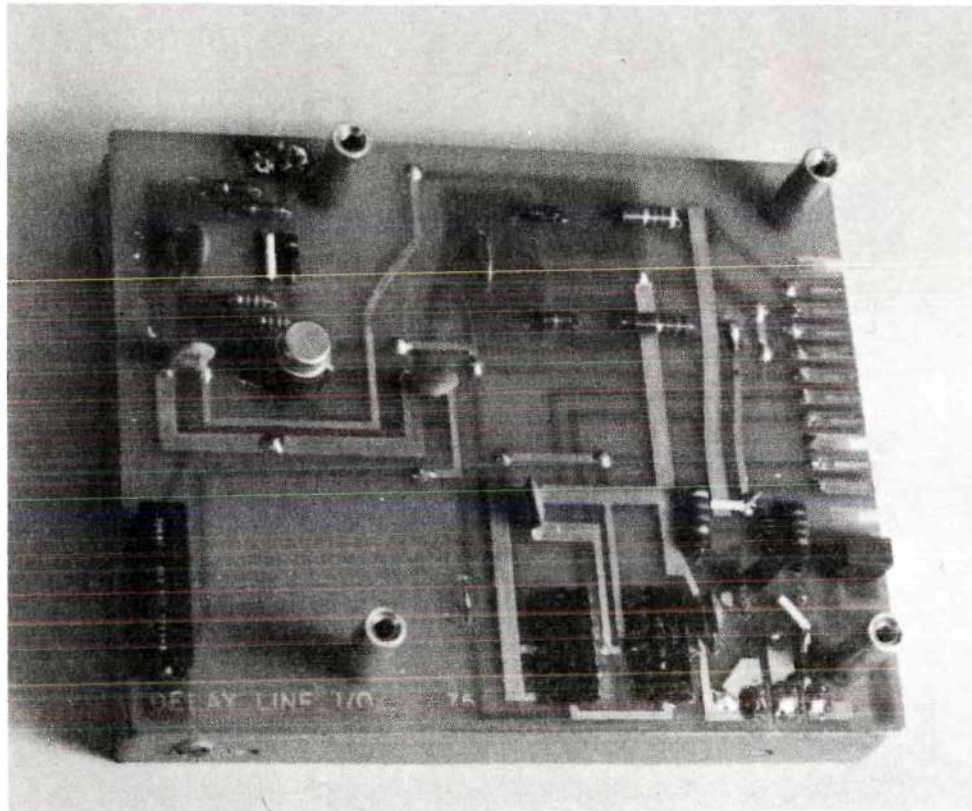


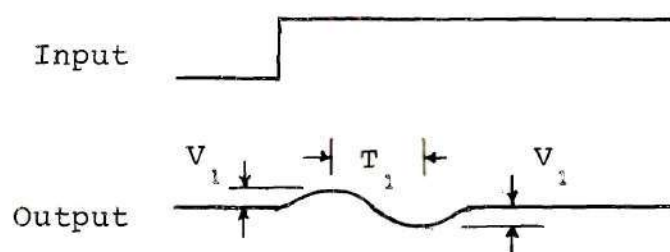
Figure 21. Delay Line and Input/Output Logic.

The memory section of the prototype consists of a 1386 bit recirculating loop (63 twenty-two bit word pairs = 1386 total bits). The data flows in the loop at the basic clock rate, that is 687×10^3 bits per second.

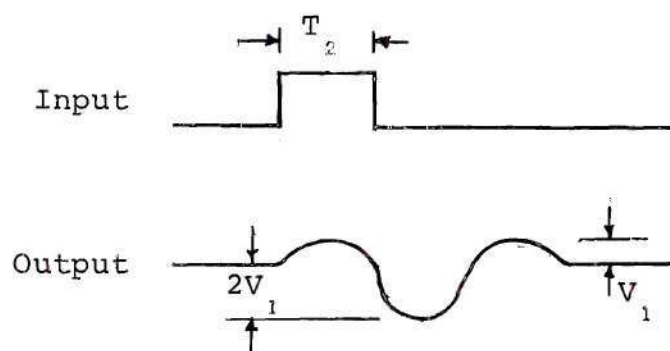
A 2 millisecond sonic magnetostrictive wire delay line is employed as the primary storage device of the loop. The function of the delay line is to delay the passage of electric pulses for some time period. At the clock rate of the loop the 2 millisecond Computer Devices Corporation delay line used in the prototype stores 1374 of the 1386 total information bits.

In a magnetostrictive delay line electrical current impulses are converted through an input transducer to strain waves which travel along the delay wire. An output transducer at the end of the delay wire generates a voltage proportional to these strain waves. Figure 23 shows the output waveforms of the delay line to a step and pulse voltage input. The width of the pulse T_2 should be approximately equal to T_1 of the step response. In the delay line used in the prototype the output values for a twelve volt step input were $T_1 \approx 0.5\mu$ seconds and $V_1 \approx 4$ millivolts. The input circuitry to the delay line consists of a transistor current switch with an adjustment to vary the input pulse width. This circuit is shown in the appendix. Detecting and synchronizing the output waveform is accomplished using a high speed voltage comparator and appropriate logic circuitry. The

output circuitry is also illustrated in the appendix. Figure 21 shows a photograph of the delay line with its input/output circuitry. The I/O circuitry on a printed circuit board is mounted directly on the delay line package itself. A ten pin amphenol connector is used to interface power and signal leads with the rest of the prototype circuitry.



(a) Voltage Step



(b) Voltage Pulse

Figure 23. Delay Line Input and Output Waveforms.

The recirculating loop is closed through an external eleven bit shift register (Register A), that is used for parallel access to information in the loop. There is also a one bit external shift register immediately following the

delay line output circuitry that is used to zero the contents of the delay line during the initialize mode. The shift registers are made using flip-flops as shown in Figure 24.

Two serial data flow switches are in the loop to control the calculating and shifting operations. The serial switch circuit is illustrated in the appendix and a photograph of the printed circuit card is shown in Figure 20. The switch control signals come from the timing and control circuitry to be discussed later.

The analog output waveform of the prototype comes from the output ladder decoder. The ladder decoder produces an output voltage proportional to the binary number applied to its input. The circuit diagram of the ladder decoder is shown in the appendix. When a value of h' (in the form of an eleven bit binary number) has been shifted into Register A, timing pulses load it into the latches of the output gating circuitry. First the reset pulse R_2 , strobed with C_{DC1} , resets the latches to zero. Then the pulse ST_2 , strobed with C_{DC2} , sets the latches in which there is a corresponding ONE in Register A. The value of h' is then held in the latches (and hence the input to the ladder decoder) for 22 clock pulses. At this time the timing pulses R_2 and ST_2 load a new value of h' into the output gating circuitry from Register A. In this manner a complete impulse response waveform of 63 values is displayed each recirculation.

The analog to digital converter of the prototype

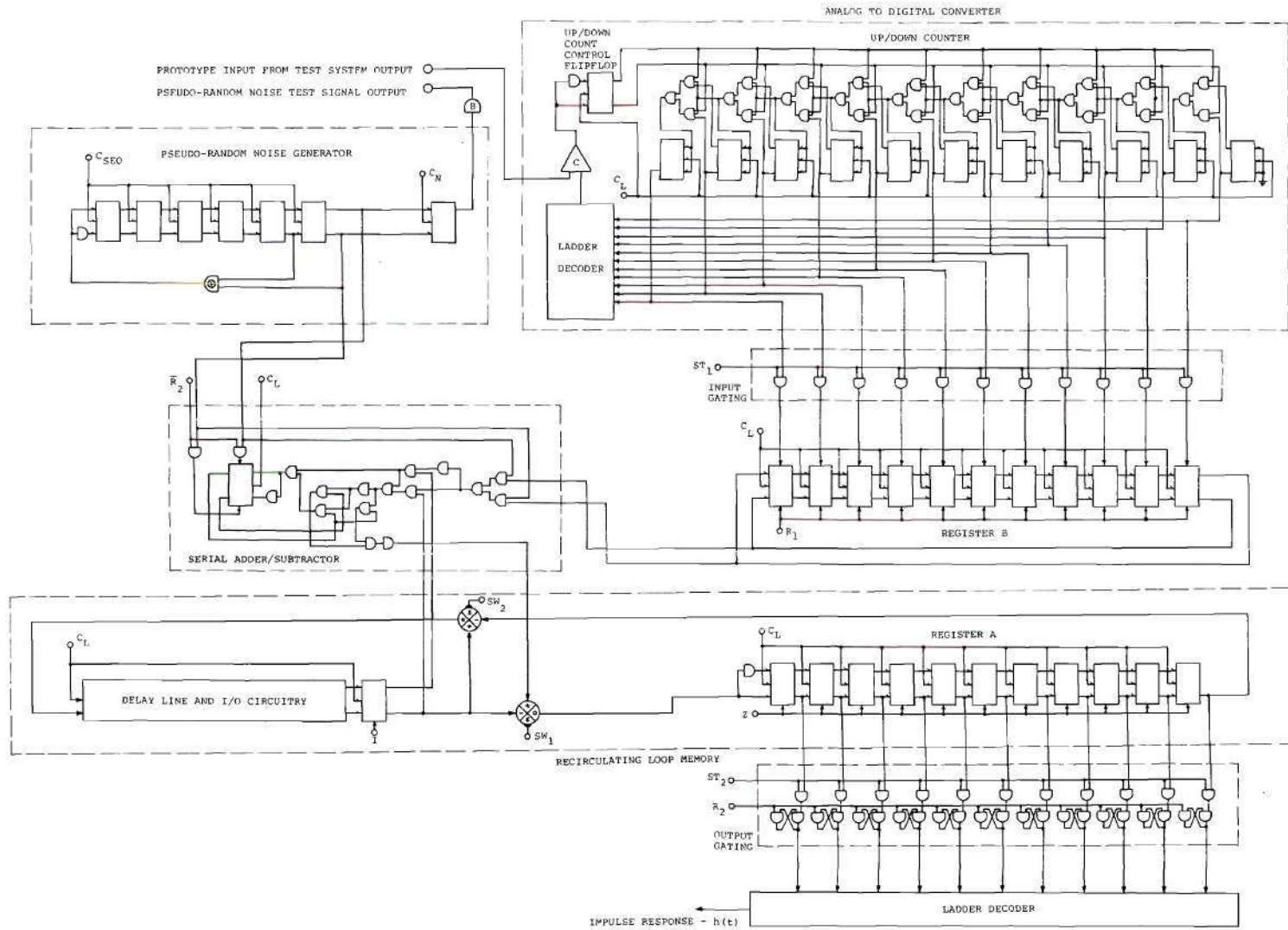


Figure 24. Logic Design of Prototype Subsystems.

consists of an up/down counter, a ladder decoder and a high speed voltage comparator. The binary number in the up/down counter is converted to an analog voltage by the ladder decoder. This voltage is compared to the analog input voltage by the high speed comparator. If the output of the ladder decoder (the analog equivalent of the binary number in the up/down counter) is larger than the input voltage, the comparator output applies a *down count* level to the up/down count control flip-flop. On the next C_L clock pulse this count control information is transferred to the counter circuitry and the counter counts down. Similarly, if the ladder decoder output is lower than the analog input, the count control flip-flop is set so that the counter will count up. The counter thus counts continuously either up or down following the analog voltage input at the basic clock rate C_L .

Once each recirculation the data of the 11 bit recirculating loop formed by Register B is updated from the A/D converter. First the reset pulse R_1 , strobed with C_{DC_1} , resets Register B to zero. Then the pulse ST_1 , strobed with C_{DC_2} , sets the flip-flops of Register B where there is a ONE in the corresponding bit position of the up/down counter. This sample of the test system's output, y , in binary form recirculates 126 times in Register B during one recirculation of the 1386 bit memory loop. Both loops are clocked at the same basic clock rate C_L . The values of y in digital form

are thus supplied to the adder/subtractor in serial form, least significant bit first.

The augend/minuend input to the serial adder/subtractor comes from the recirculating loop memory and the addend/subtrahend input comes from Register B as illustrated in Figure 24. The sum/difference output of the adder/subtractor is fed to the + input of serial data flow switch 1. Switch 1 changes the data flow every 11 clock pulses during the calculation process as discussed earlier. This allows the stored values of h' to pass unaltered from the delay line into Register A while the calculation of new values is being carried on in adjacent 11 bit words.

The add/subtract control comes from the maximum-length shift register sequence generator that produces the pseudo-random noise test signal. Every $22 C_L$ clock pulses the shift register is clocked once by the timing pulse C_{SEQ} except at the end of each recirculation of the memory loop. At that time the shifting transition is applied to the noise generator output flip-flop as C_N . Thus the noise sequence generator is clocked through 63 of its 64 states each recirculation of the memory loop and the output noise sequence changes once each recirculation.

The timing and control circuitry is illustrated in Figure 25 and Figure 26 shows a timing diagram of the major control signals.

There are four counters that are at the base of the

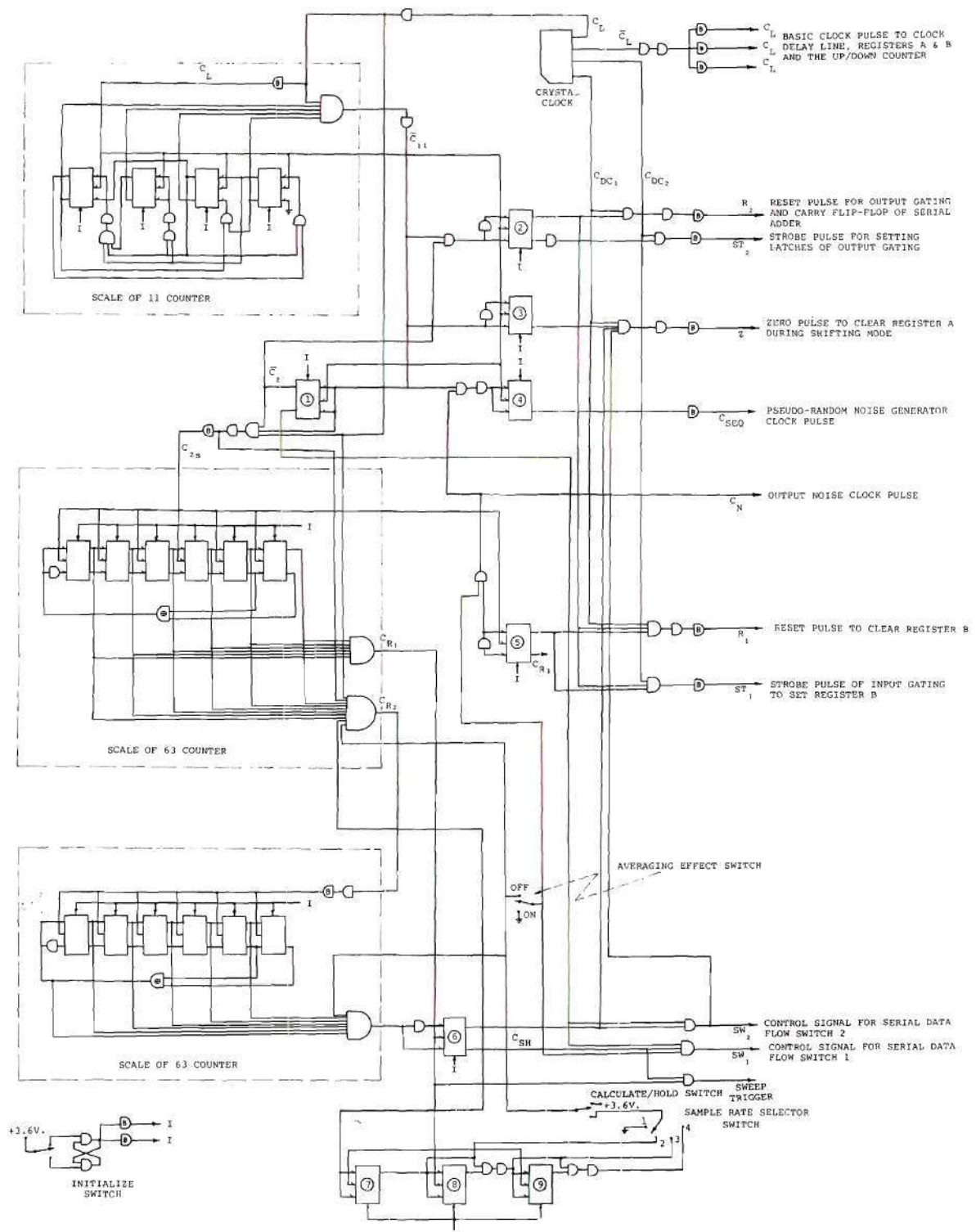


Figure 25. Timing and Control Circuitry.

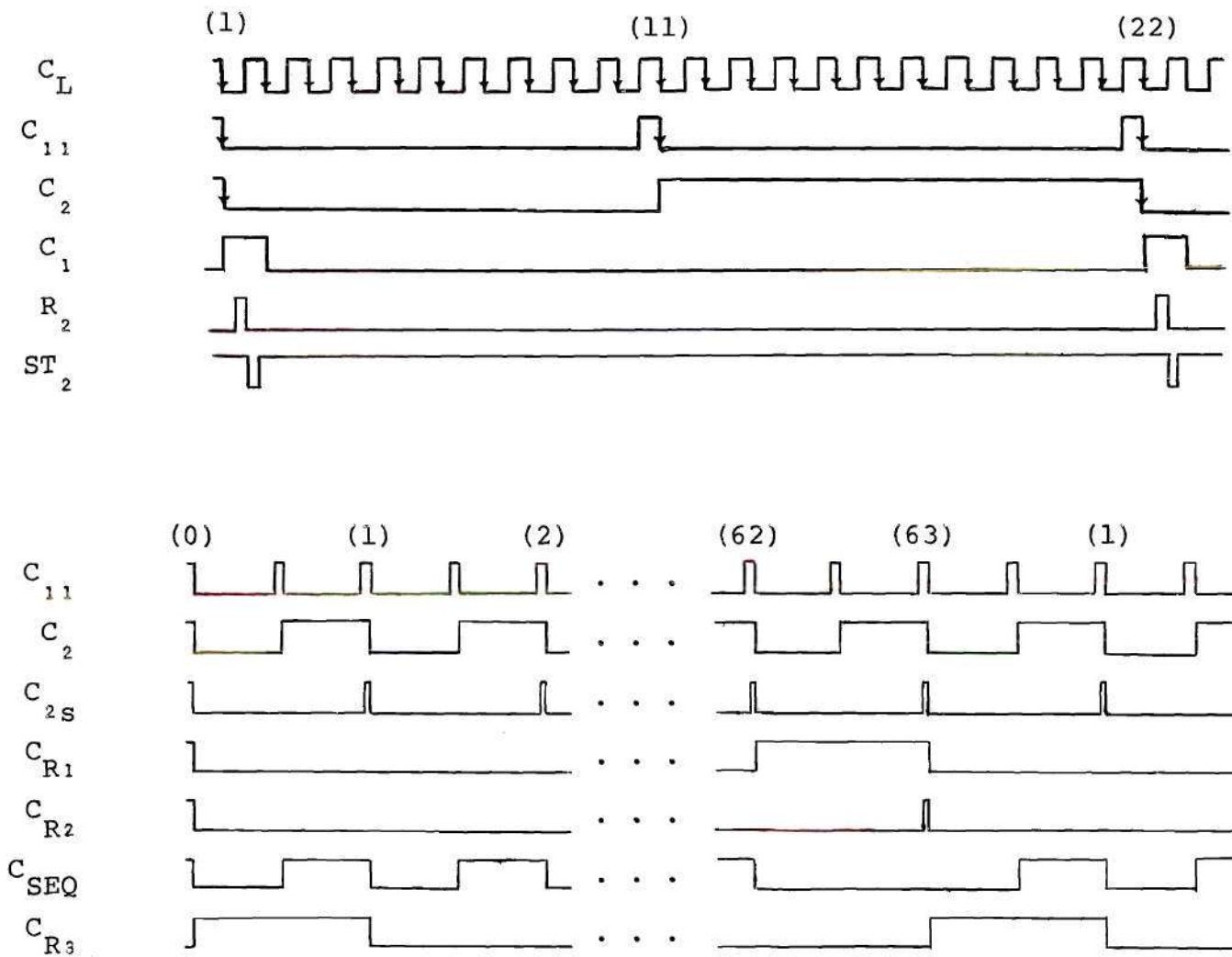


Figure 26. Timing Diagram.

timing circuitry. A scale of 11 counter produces a pulse every 11 C_L clock pulses to indicate each time a word passes a given point in the memory loop. These pulses are counted by a scale of two counter which indicates when a word pair has passed a given point in the loop. The signal from this counter is fed to a scale of 63 counter which produces a pulse each time information in the loop has made one recirculation. Finally, a second scale of 63 counter indicates the end of the crosscorrelation calculation (63 recirculations of the memory loop). It is from these counter signals that the various other timing and control signals are created.

The scale of 11 binary up counter clocked by C_L consists of four flip-flops. The set and reset levels of the flip-flops are generated as a function of the present state of the counter. The initialize pulse DC resets the four flip-flops to zero (the state 0000). On the first C_L clock pulse the right most flip-flop changes state, thus the first state of the counter is 0001. The levels of the flip-flops are set according to this state of the counter and on the second clock pulse the state changes to 0010. The counter thus counts up in a binary count sequence. The set and reset levels are determined in such a way that on the eleventh clock pulse the counter returns to the zero state (0000). It is this state that is detected by a five input gate, one input from each of the counter flip-flops and a strobe input \overline{C}_L . The strobe input is used to keep the negative transition

(+3.6 to zero volts) of the output pulse C_{11} synchronized with the negative transition of C_L .

The pulse \overline{C}_{11} is used to set the level of flip-flop 1 shown in Figure 25. This flip-flop, clocked by C_L , is the scale of two counter which indicates when a 22 bit word pair has passed a given point in the memory loop. Its output C_2 is shown in the timing diagram of Figure 26. By appropriately gating \overline{C}_{11} and \overline{C}_2 and using this as an input to a one bit shift register (flip-flop 2) clocked by C_L , the pulse C_1 is formed. The timing signals R_2 and ST_2 are then formed by strobing C_1 with C_{DC1} and C_{DC2} respectively. Inverting buffers are used to give the needed drive capability.

The pulse \overline{C}_2 , strobed with \overline{C}_{11} and \overline{C}_L to synchronize the negative transition with C_L , forms the pulse C_{2s} which is used to clock the first scale of 63 counter. This counter is actually a maximum length shift register sequence generator. By using a six bit shift register with mod 2 feedback of the last two bit positions, the counter can be made to go through 63 different states before repeating. The state set by the initializing pulse is 111111. This is also the state detected by the gates to form the output pulses C_{R1} and C_{R2} . C_{R1} is positive during the entire time the counter is in the detected state. The C_{R2} pulse is strobed with C_{2s} , \overline{C}_{11} and \overline{C}_L to synchronize its negative transition with C_L . There is also an inhibit signal input to the gate producing C_{R2} which comes from the sample rate

selector switch to be discussed later. The C_{R1} pulse is fed to a one bit shift register (flip-flop 5), clocked by C_{2s} , to produce the signal C_{R3} . The reset pulse R_1 used to clear Register B is then formed by ANDing C_{R3} , \bar{C}_1 and \bar{C}_{DC1} . Similarly, the strobe in pulse ST_1 used to set values of y in Register B is formed by ANDing \bar{C}_{R3} , \bar{C}_1 and \bar{C}_{DC2} .

The second scale of 63 counter (identical in construction to the previous scale of 63 counter) is clocked by the C_{R2} pulse. A gate detects the sixty-third state and this signal is shifted into a one bit shift register (flip-flop 6), clocked by C_{R1} . The output of this flip-flop is C_{SH} , the shift mode signal. This signal is +3.6 volts during the recirculation in which the old stored values of h' are replaced by the newly calculated values.

By ANDing \bar{C}_2 and \bar{C}_{SH} , the control signal SW_1 is formed. This signal controls the serial data flow switch 1 and is essentially the signal \bar{C}_2 inhibited during the shifting mode. Similarly, the signal SW_2 is formed by ANDing \bar{C}_2 and C_{SH} . This signal is identical to \bar{C}_2 during the shifting mode but zero otherwise.

The oscilloscope sweep trigger pulse is formed using \bar{C}_{R1} . ANDing \bar{C}_{SH} and \bar{C}_R is done to eliminate the trigger pulse before the shifting recirculation.

The pseudo-random noise generator clock pulse, C_{SEQ} , is created in the same manner as C_2 except that its negative transition is inhibited once each recirculation. (See Figure

26). The C_N negative transition occurs once each recirculation at the time the inhibited C_{SEQ} transition would occur. C_N is formed from an ANDing of $\overline{C_{R1}}$ and the signal from the sample rate selector. It is the C_N signal that is used to inhibit the C_{11} signal from setting the levels of flip-flop 4 once each recirculation.

The pulse Z, which clears Register A during the shifting mode, is formed by ANDing the output of flip-flop 3, $\overline{C_{DC1}}$, C_{SH} and $\overline{SW_2}$.

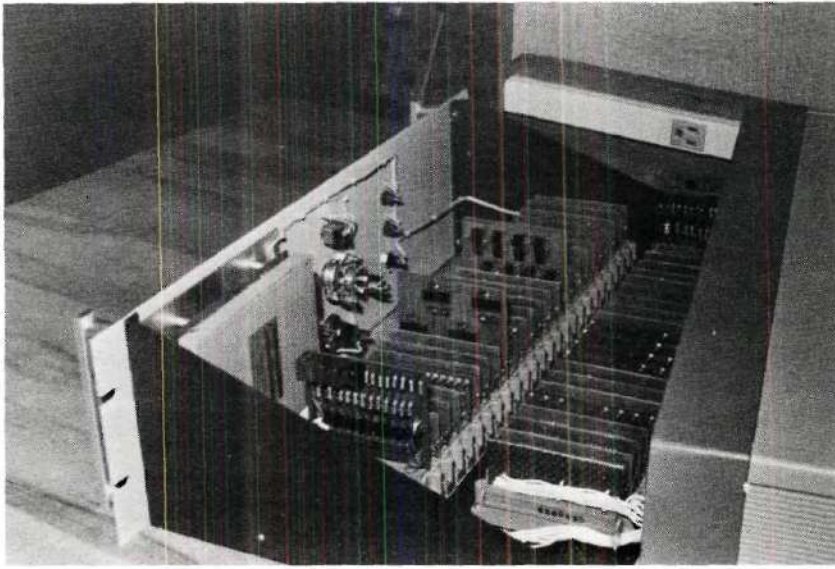
When the sample rate selector switch is in position 1 (grounded), the timing circuitry functions as described above and the prototype calculates an impulse response in 63 recirculations of the memory loop (0.127 seconds). When the switch is in position 2, the signal SR_2 inhibits the calculation process every other recirculation. Samples of y are taken every other recirculation and the rate at which the noise test signal is clocked is reduced by a factor of two. It therefore takes 126 recirculations of the memory loop to calculate a set of impulse response values (0.254 seconds). In position 3 the SR_3 signal allows the calculator process to take place only every fourth recirculation, thus producing a set of impulse response values every 252 recirculations (0.508 seconds). With the switch in the fourth position the calculation of a set of values takes 504 recirculations (1.017 seconds).

When the calculate/hold switch is in the calculate

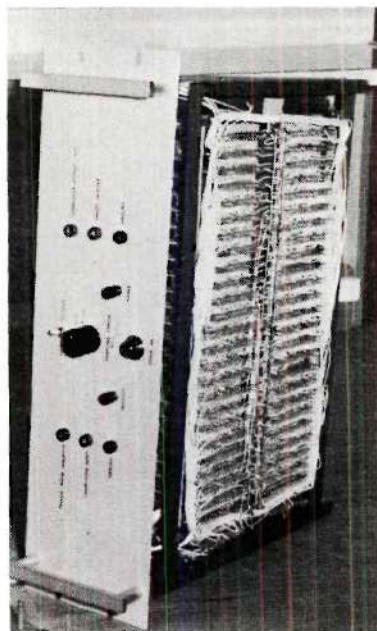
position, the sample rate selector switch controls the calculation rate. In the hold position, the shifting mode is ultimately inhibited, and the stored values in the memory loop are not changed. The impulse response waveform is held on the oscilloscope screen until the switch is returned to the calculate position.

The averaging effect switch shown in the timing circuitry was used to test the affect of averaging a number of output waveforms and will be discussed in the next chapter.

Figure 27 shows a photograph of the constructed prototype calculator. The printed circuit cards containing the logic elements were arranged in two rows using Elco 29-pin plug-in connectors as shown in the photograph of Figure 28(a). Wiring was done using the wire-wrap technique which allows for dense yet reliable connector interconnection. This can be seen in the photograph of Figure 28(b). The delay line with its I/O circuitry was mounted behind the front panel and the power supplies for the logic were mounted in the rear of the cabinet. Plug-in jacks on the front panel provided for signal interconnections with the prototype.



(a)



(b)

Figure 28. Card Locations and Wiring of Prototype.

CHAPTER VI

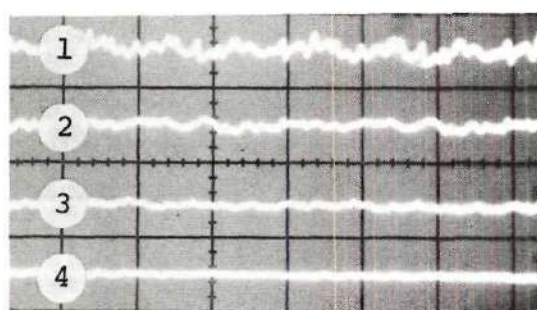
EXPERIMENTAL TESTS AND RESULTS

Test Configurations

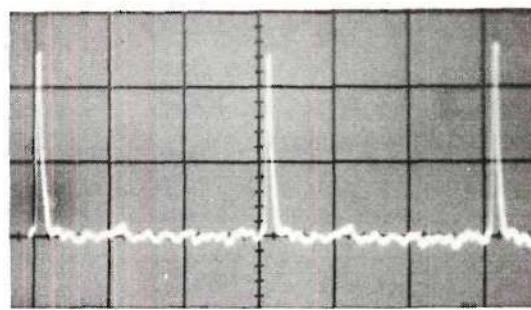
After the construction of the prototype calculator was completed, a series of tests were conducted to determine the effectiveness of its operation. The first tests made were conducted to determine the limitations of and the inherent error in the digital crosscorrelation process. A number of tests were then made using the prototype calculator to determine the impulse responses of several test systems. Series RC circuits with various time constants were used as first order test systems. An EAI TR-20 analog computer was used to simulate several different second order systems. The analog computer also served as a voltage summing junction for combining the various input test signals. Results of these tests are illustrated in the oscilloscope photographs of Figure 29.

Tests and Results

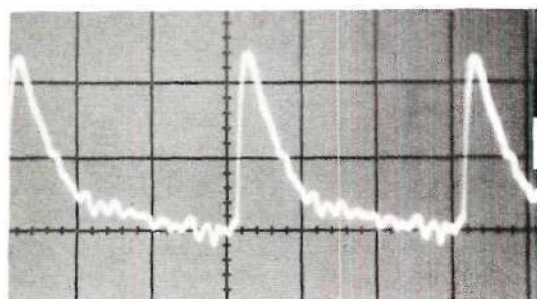
Curve Number 1 of Figure 29(a) is a sample of the crosscorrelator output with a DC input. In this case, the input to the crosscorrelator was grounded. Because the crosscorrelation of a pseudo-random noise sequence with a constant is also a constant, this waveform should be a smooth



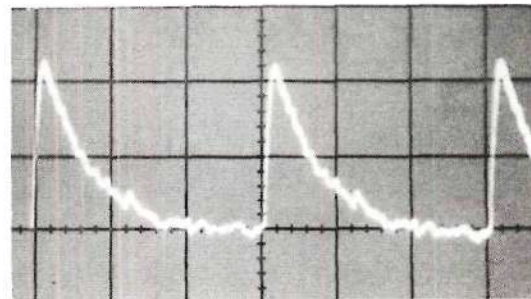
(a)



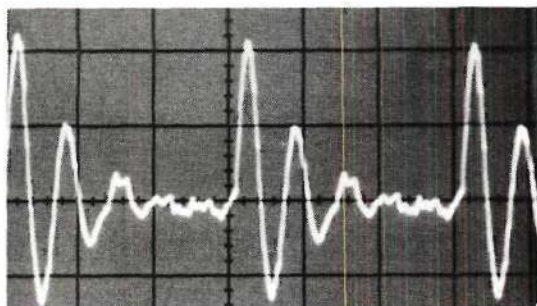
(b)



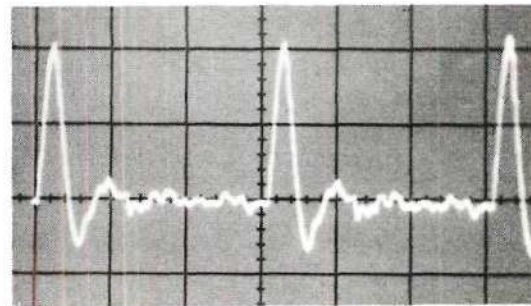
(c)



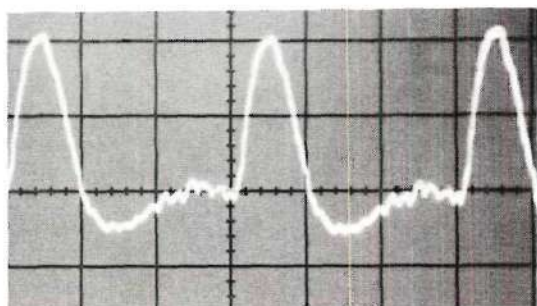
(d)



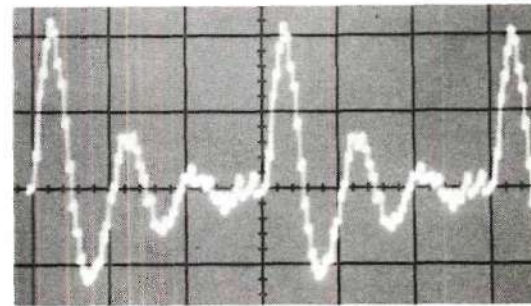
(e)



(f)



(g)



(h)

Figure 29. Test Waveforms.

straight line. The variation in the actual output waveform is primarily due to the error in the analog-to-digital conversion process. This variation would be reduced if a larger number of samples were taken which would involve using a pseudo-random sequence with a larger period. In this manner more points would be calculated per waveform, and each point would be the average of a larger number of values. The effect of more points per waveform could not be simulated with the prototype as constructed; however, the result of averaging over a larger number of values could be approximated and is illustrated in Figure 29(a).

Curve Number 2 of this figure illustrates the result of averaging two output waveforms. It was produced by allowing the prototype to calculate two sets of data, one upon the other, before shifting them to the stored position for display. This is accomplished with the averaging effect switch (mentioned in Chapter V and illustrated in the timing circuitry of Figure 25) in the ON position and the sample rate selector switch in position 2. Similarly, Curve Number 3 of Figure 29(a) illustrates the result of averaging four output waveforms. In this case, the sample rate selector switch was in position 3. With the sample rate selector switch in position 4, the result of averaging eight output waveforms was produced as illustrated in Curve Number 4. The relative variations illustrated by curves 1, 2, 3 and 4 approximate roughly the variations that would be obtained

by using pseudo-random sequences of lengths 63, 127, 255 and 511 respectively.

By using the output noise sequence of the prototype as the input to the correlator, the resulting output will be the *autocorrelation* of the pseudo-random sequence. This waveform is illustrated in Figure 29(b).

To determine the extent to which the pseudo-random noise test signal would be uncorrelated with a normal test system output, a series of tests were conducted in which sinusoidal signals of various frequencies were applied to the input of the correlator. The output waveform was unaffected by the input for most frequencies; however, some correlation was noted for a number of the different frequencies tested. By using the averaging technique mentioned above, it was found that more frequencies were uncorrelated as more output calculations were averaged. This indicates that with a longer pseudo-random sequence, fewer frequency components will possess a noticeable correlation. It should be noted, however, that since the pseudo-random sequences are periodic, there will always be some frequency components that will be correlated.

The prototype calculator was next used to calculate the impulse response of several test systems. Figure 29(c) is the calculated impulse response of an RC network with time constant $\tau = 0.025$ seconds. In this figure three horizontal scale divisions equal 0.127 seconds. Figure 29(d)

illustrates a similar first order response curve with $\tau = 0.25$ seconds. In this figure, the sample rate selector switch was adjusted so that three horizontal scale divisions equal 1.017 seconds. The rest of the waveforms illustrated in Figure 29 also have this same horizontal time scale.

Using the analog computer to simulate a second order system having a natural frequency ω_0 of 30 radians per second and a damping ratio δ of 0.15, the impulse response curve of Figure 29(e) was produced by the prototype. The curve of Figure 29(f) was calculated using a simulated second order system with ω_0 equal to 30 radians per second and δ increased to 0.4. Figure 29(g) illustrates the calculated response for a second order system with $\delta = 0.4$ and ω_0 decreased to 10 radians per second.

Output waveforms taken directly from the analog-to-digital converter have a "stair step" appearance as illustrated in Figure 29(h) which is the calculated impulse response of a second order system with $\omega_0 = 20$ radians per second and $\delta = 0.2$. To reduce this appearance and produce the smooth curves previously discussed, the output signal was low-pass filtered.

Conclusions

From the test results it can be concluded that the digital approach to the implementation of the crosscorrelation algorithm for calculating system impulse responses is highly

effective. This is primarily due to the simplified algorithm which greatly reduces the crosscorrelation calculation. This in turn reduces the amount of logic required and hence the physical size of the calculator. In taking only 63 samples per response time, impulse response waveforms were produced that compared very well with their corresponding theoretical curves. By using longer pseudo-random noise sequences and taking more samples per response time, the results would be greatly improved.

An important problem encountered in testing the prototype and one that must be considered in a practical application of this technique is that of the A/D conversion magnitudes. For optimum A/D conversion the analog voltage range of operation was adjusted to cover the digital range of an 11 bit sign plus two's complement number. However, while operating in this manner, the data being calculated in the recirculating loop often exceeded the magnitude of the 11 bit words. This could be avoided by reducing the amplitude of the analog input signal, but not without a reduction in the accuracy of the A/D conversion because of fewer quantization levels per volt. One solution to the "overflow" problem in the calculation process would be to provide a word space larger than 11 bits in the recirculating loop. When the calculation process is completed, it would be a simple task to digitally normalize the data to within the range of an 11 bit number.

There are many areas in the engineering field in which such a calculator could be used. For example, in a predominantly second order system the overshoot of a step response is uniquely related to the ratio of positive to negative area of the impulse response.⁹ Also, the final value of the step response is equal to the total area of the impulse response.² These values could be easily calculated and compared digitally to produce a measure of the overshoot.

The first zero crossing of the impulse response corresponds to the peak of the step response.² This information could be used to control the rise time of a control system.

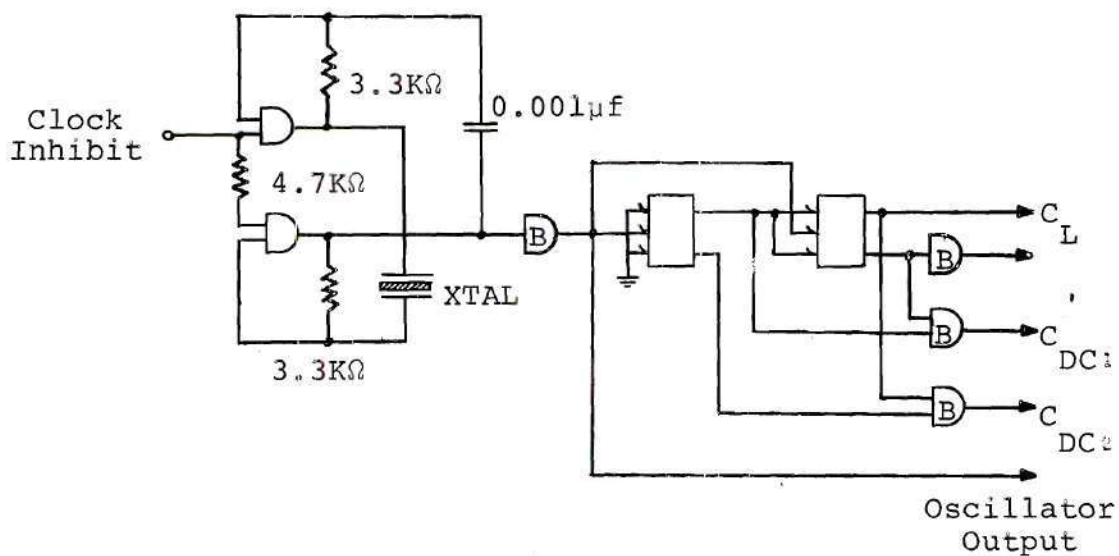
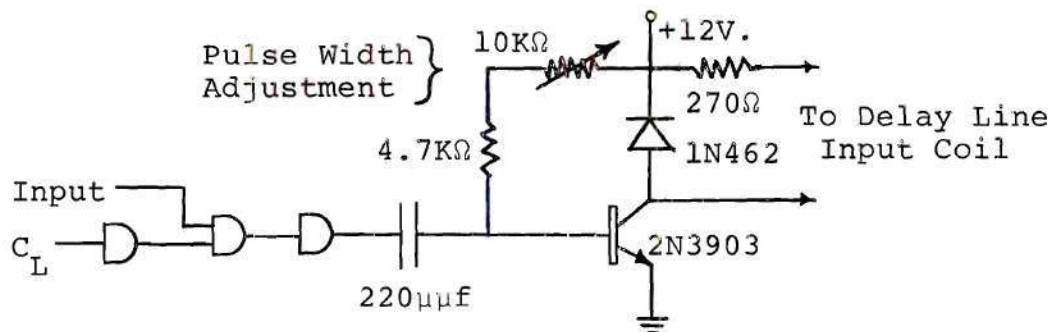
Since the system gain is proportional to the area of the impulse response, an error signal for gain control could easily be calculated.²

At the present time, much work is being done in the area of modeling human dynamic response characteristics. The technique presented here may possibly be used along these lines.

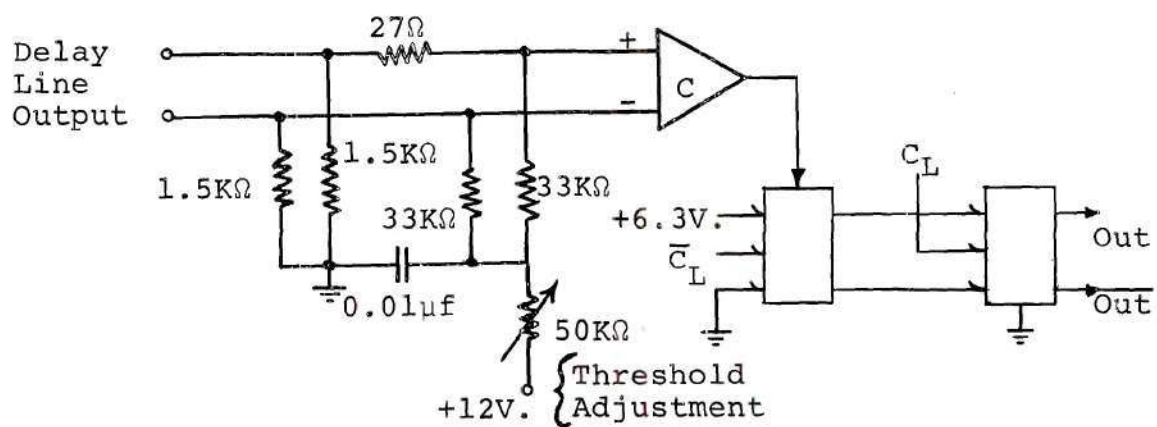
These are, of course, but a few of the potential uses for an "on-line" system or process identifier. The algorithm and digital implementation presented here should provide a useful tool for system modeling in many areas of the engineering field.

APPENDIX

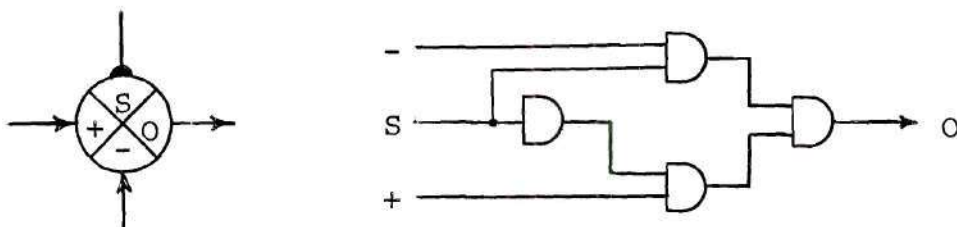
CIRCUIT DIAGRAMS

Crystal Clock CircuitDelay Line Input Circuit

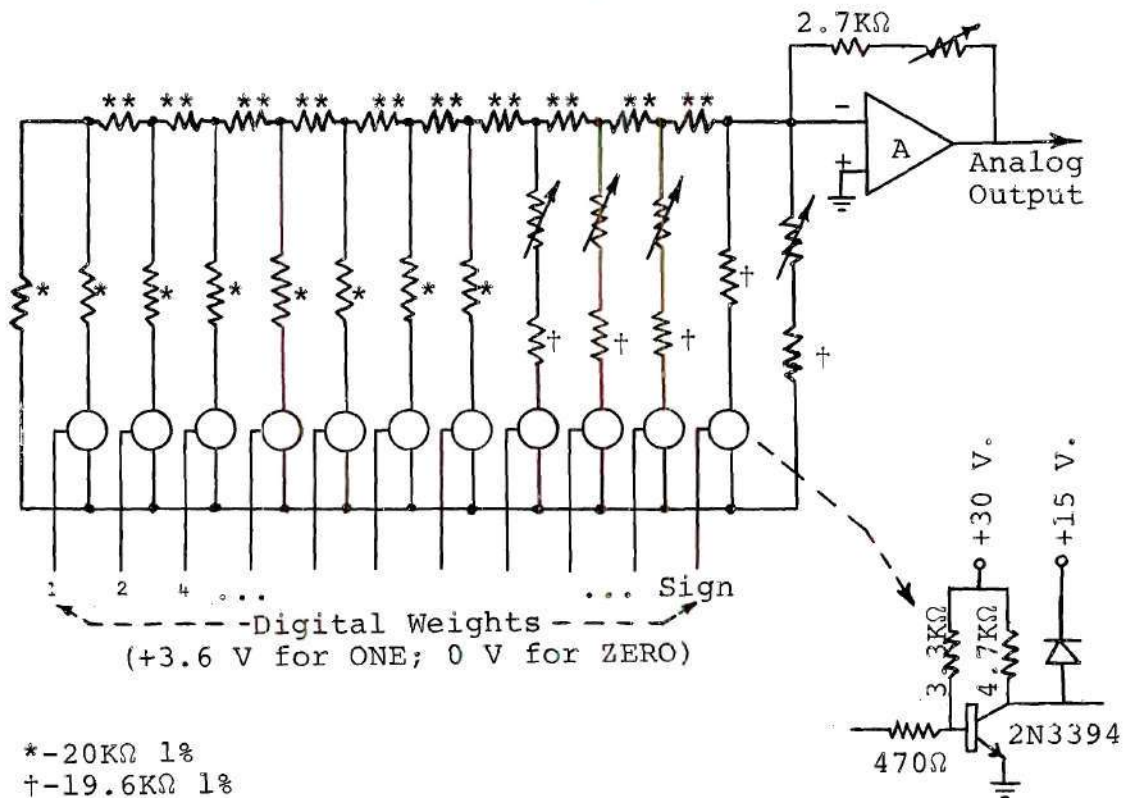
Delay Line Output Circuit



Serial Data Flow Switch



Ladder Decoder Circuit



*-20K Ω 1%
 †-19.6K Ω 1%
 **-10K Ω 1%
 All potentiometers 1K Ω

BIBLIOGRAPHY

Literature Cited

1. P. Eykhoff and Others, Survey-Systems Modelling And Identification, Third Congress of the International Federation of Automatic Control, June, 1966.
2. William G. Anderson and Others, Use of Crosscorrelation in an Adaptive Control System, Procedures of the National Electronics Conference, Vol. 15, October, 1959.
3. L. Padulo, The Time Domain Inversion of Convolution, PhD. Thesis, Georgia Institute of Technology, 1966.
4. D. C. J. Poortvliet, The Measurement of System-Impulse Response by Means of Crosscorrelation with Binary Signals, Tijdschrift Nederlands Elektronica en Radiogenootschap, Vol. 28, No. 4, pp. 253-270, 1963.
5. John C. Hancock, An Introduction to the Principles of Communication Theory, McGraw-Hill Book Company, Inc., New York, pp. 105-106, 113, 118-119, 1961.
6. J. G. Truxal, Automatic Feedback Control System Synthesis, McGraw-Hill Book Company, Inc., New York, pp. 429-438, 1955.
7. G. R. Cooper, Crosscorrelation With Binary Signals, School of Electrical Engineering, Purdue University, Lafayette, Indiana.
8. S. W. Golomb, Digital Communication With Space Applications, Prentice Hall, Inc., Englewood Cliffs, New Jersey, pp. 2-25, 1964.
9. G. W. Anderson and Others, "A Self-Adjusting System Optimum Performance," I.R.E. Convention Record Part 4, p. 182, 1958.

Other References

Davenport, Wilbur B., and Root, William L., Random Signals and Noise, McGraw-Hill Book Company, Inc., New York, 1958.

Golomb, Solomon W., Shift Register Sequences, Holden-Day, Inc., San Francisco, 1967.

Peatman, John B., "Digital Delay Lines," Electromechanical Design, September, 1967.

Papoulis, Anthanasios, Probability, Random Variables, and Stochastic Processes, McGraw-Hill Book Company, New York, 1965.

Papoulis, Anthanasios, The Fourier Integral and its Applications, McGraw-Hill Book Company, Inc., New York, 1962.

Stein, Seymour and Jones, J. Jay, Modern Communication Principles, McGraw-Hill Book Company, New York, 1967.

Systems Engineering Laboratories, Inc., "Sel 8500 Series Micrologic Modules," Bulletin 9031, 1965.