INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.

Xerox University Microfilms
300 North Zeab Road
Ann Arbor, Michigan 48106
ANALOG AND DIGITAL RESONANT SEQUENCE FILTERS
FOR WALSH FUNCTIONS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By
Donald Lee Moon, B.S., M.S.

The Ohio State University
1974

Reading Committee:
Prof. F. C. Weimer
Prof. W. C. Davis
Prof. R. B. Lackey

Approved By
Robert B. Lackey
Advisor
Department of Electrical Engineering
ACKNOWLEDGEMENTS

Special acknowledgement is given Professors William Davis and Carlin Weimer for their efforts as reading committee members for this work. Also, special thanks are due my wife, Lorraine, for her sacrifices during the period of research for and preparation of this paper.

In addition, the author extends distinct appreciation to Professor Robert Lackey for his sincere encouragement, support and advice.

Finally, appreciation is extended to the NCR Company, Dayton, Ohio, for aiding with the final drawings and Joyce Bruce for typing the manuscript.
VITA

July 5, 1936 . . . Born - Boomer, West Virginia

1963 . . . . . . B.S.E.E., West Virginia Institute of Technology, Montgomery, West Virginia

1963-1964 . . . Engineer, Avco Corp., Electronics Division, Cincinnati, Ohio

1964-1965 . . . Graduate Assistant, Department of Electrical Engineering, University of Toledo, Toledo, Ohio

1965-1967 . . . Engineer, National Cash Register Company, Dayton, Ohio

1966 . . . . . . M.S.E.E., University of Toledo, Toledo, Ohio

1967-1973 . . . Assistant Professor of Electrical Engineering, University of Dayton, Ohio

1973 . . . . . . Associate Professor of Electrical Engineering, University of North Carolina, Charlotte, North Carolina

1974 . . . . . . Senior Project Engineer, National Cash Register Company, Dayton, Ohio

PUBLICATIONS

"The Polynomial Counter - With Applications in Four-Phase Logic," Computer Design, December, 1969, pp. 135-143


FIELDS OF STUDY

Major Field: Electrical Engineering

Digital Systems Theory
Control Theory
Network Synthesis
Applied Mathematics
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>VITA</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 THE WALSH FUNCTIONS</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Representation of Walsh Functions by Difference Equations</td>
<td></td>
</tr>
<tr>
<td>2.2 Representation of Walsh Functions as Products of Rademacher Functions</td>
<td></td>
</tr>
<tr>
<td>2.3 Mathematical Properties of the Set of Walsh Functions</td>
<td></td>
</tr>
<tr>
<td>Orthogonal Systems of Functions</td>
<td></td>
</tr>
<tr>
<td>Complete System of Functions</td>
<td></td>
</tr>
<tr>
<td>The Multiplication Theorems for Walsh Functions</td>
<td></td>
</tr>
<tr>
<td>2.4 Generation of the Walsh Functions</td>
<td></td>
</tr>
<tr>
<td>General Concepts of Analog Walsh Waveform Generation</td>
<td></td>
</tr>
<tr>
<td>General Concepts of Digital Waveform Generation</td>
<td></td>
</tr>
<tr>
<td>2.5 Applications of Walsh Functions</td>
<td></td>
</tr>
<tr>
<td>3 SEQUENCE FILTERS BASED ON WALSH FUNCTIONS</td>
<td>33</td>
</tr>
<tr>
<td>3.1 Matched Sequency Filters</td>
<td></td>
</tr>
<tr>
<td>3.2 Resonant Sequency Filters</td>
<td></td>
</tr>
<tr>
<td>3.3 Developmental History of LCS Resonant Sequency Filters</td>
<td></td>
</tr>
<tr>
<td>3.4 An Application of Resonant Sequency Filters</td>
<td></td>
</tr>
<tr>
<td>4 ANALOG RESONANCE FILTERS FOR WALSH FUNCTIONS</td>
<td>41</td>
</tr>
<tr>
<td>4.1 The Ideal Single Pass LCS Filter</td>
<td></td>
</tr>
<tr>
<td>4.2 The Non-ideal Single Pass LCS Filter</td>
<td></td>
</tr>
</tbody>
</table>
Chapter

5 DISCRETE RESONANT FILTERS FOR WALSH FUNCTIONS 62

5.1 Difference Equation for the Discrete Resonant Filter
5.2 Simulation of the Discrete Resonant Filter
5.3 Results of Simulation of the Discrete Filter

6 Z-TRANSFORM ANALYSIS OF RESONANT FILTERS FOR WALSH FUNCTIONS 89

6.1 The Discrete, Time-Invariant Resonant Filter
6.2 The Discrete, Time-Variant Resonant Filter
6.3 Transform Solution for the Time-Variant Filter
6.4 Time Domain Solution for the Time-Variant Filter

7 A SPECIAL PURPOSE COMPUTER FOR DIGITAL RESONANT FILTERS 116

7.1 Organization and Operation of a Special Purpose Computer
7.2 Implementation of a Special Purpose Computer

8 CONCLUSIONS 129

APPENDIX

A 132
B 143

BIBLIOGRAPHY 154
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Truth values of Walsh functions.</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>Relationship between Walsh and coincidence of sal functions</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>Required relationships between ( s(k) ) and ( v_1(k) ) for the first eight Walsh functions.</td>
<td>67</td>
</tr>
<tr>
<td>4</td>
<td>Phase I simulation results; input set equal to the resonating signal</td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>Phase II simulation results; the sal ((k,1,8)) filter</td>
<td>82</td>
</tr>
<tr>
<td>6</td>
<td>Phase II simulation results; the sal ((k,3,8)) filter</td>
<td>83</td>
</tr>
<tr>
<td>7</td>
<td>Phase II simulation results; the cal ((k,3,8)) filter</td>
<td>84</td>
</tr>
<tr>
<td>8</td>
<td>The output of the filter tuned to resonate at ( \text{sal}(k,4,8) )</td>
<td>93</td>
</tr>
<tr>
<td>9</td>
<td>Proof by perfect induction that ( B=1-A )</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>Proof by perfect induction that ( G=1-D ) and ( F=1-C )</td>
<td>101</td>
</tr>
<tr>
<td>11</td>
<td>Table 3 under the mapping (-1\rightarrow1) and (+1\rightarrow0) for the Walsh functions</td>
<td>128</td>
</tr>
<tr>
<td>12</td>
<td>Z-transforms of the Walsh function sampled at 2, 4, 8, and 16 rates</td>
<td>139</td>
</tr>
<tr>
<td>13</td>
<td>Z-transforms of the Walsh functions sampled at the 32 rate</td>
<td>140</td>
</tr>
<tr>
<td>14</td>
<td>Simulation program listing</td>
<td>144</td>
</tr>
<tr>
<td>15</td>
<td>Input data for simulation program</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>Input-output response for the resonant filter</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>16</td>
<td>cal(k,0,8)</td>
<td>147</td>
</tr>
<tr>
<td>17</td>
<td>sal(k,1,8)</td>
<td>148</td>
</tr>
<tr>
<td>18</td>
<td>cal(k,1,8)</td>
<td>149</td>
</tr>
<tr>
<td>19</td>
<td>sal(k,2,8)</td>
<td>150</td>
</tr>
<tr>
<td>20</td>
<td>cal(k,2,8)</td>
<td>151</td>
</tr>
<tr>
<td>21</td>
<td>cal(k,3,8)</td>
<td>152</td>
</tr>
<tr>
<td>22</td>
<td>sal(k,4,8)</td>
<td>153</td>
</tr>
<tr>
<td>Figure No.</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>The Walsh functions ( \text{wal}(n,\theta) ) for ( n=0,1,\ldots,15 )</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>The Rademacher function for ( n=1,2,3 ) and 4</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Illustration showing that ( \text{wal}(13,\theta)=\text{r}(1,\theta) ) ( r(4,\theta) )</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Karnaugh maps used to find additional Walsh functions</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>Address message decoder</td>
<td>38</td>
</tr>
<tr>
<td>6</td>
<td>The serial LCS resonance circuit</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>Timing diagram for a LCS resonant filter tuned to the function applied to its input terminals</td>
<td>46</td>
</tr>
<tr>
<td>8</td>
<td>Timing diagrams for a LCS resonant filter not tuned to the function applied to its input terminals</td>
<td>48</td>
</tr>
<tr>
<td>9</td>
<td>Timing diagrams for a LCS resonant filter tuned to ( \text{sal}(3,\theta) ) with functions ( \text{wal}(0,\theta) ) to ( \text{wal}(4,\theta) ) applied to its input terminals</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>Operation times of the switch ( S ) for tuning the filter to the functions ( \text{sal}(1,\theta) ) to ( \text{sal}(4,\theta) )</td>
<td>52</td>
</tr>
<tr>
<td>11</td>
<td>Timing diagrams illustrating separation properties for the LCS resonance filter</td>
<td>53</td>
</tr>
<tr>
<td>12</td>
<td>The parallel LCS circuit</td>
<td>55</td>
</tr>
<tr>
<td>13</td>
<td>The basic low-pass filter</td>
<td>56</td>
</tr>
<tr>
<td>14</td>
<td>Sketches illustrating the response of the non-ideal filter</td>
<td>61</td>
</tr>
<tr>
<td>Page</td>
<td>Section</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Discrete, first order network</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>An overall design and analysis concept for a discrete system</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Direct form of the network for the LCS filter</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Parallel form of the network for the LCS filter</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Flow chart for simulation program</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Phase I simulation results for the $\text{cal}(2,\theta)$ filter at resonance</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Phase I simulation results for the $\text{sal}(3,\theta)$ filter at resonance</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Phase I simulation results for the $\text{cal}(3,\theta)$ filter at resonance</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Phase I simulation results for the $\text{sal}(4,\theta)$ filter at resonance</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Phase II simulation results for the $\text{sal}(k,3,8)$ filter at anti-resonance condition</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Phase II simulation results for the $\text{sal}(k,3,8)$ filter at anti-resonance condition</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Conceptual Diagram of a digital filtering system</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Functional organization of a resonant digital filter for Walsh functions</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Functional operation of a resonant digital filter for Walsh functions</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>The input section of a digital Walsh function filter</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>The filtering section of a Walsh function filter based on the direct method</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>The filtering section of a Walsh function filter based on the parallel method</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>The output section of a Walsh function filter</td>
<td></td>
</tr>
</tbody>
</table>
The discrete Walsh functions $s(k,1,N)$ for $N = 2, 4, \text{ and } 8$.

The sampling process for $N = 8$ applied to the Walsh functions with sequence $n = 0, 1, 2 \ldots, 7$. 
CHAPTER 1

INTRODUCTION

The system of sine and cosine functions plays a dominant role in filter theory. There are many historical, theoretical, and practical reasons to support this fact. From the theoretical point of view, one of the major reasons is that Fourier series and the Fourier transform permit the representation of a large class of functions by a superposition of sine and cosine functions. This representation makes it possible to apply the concept of frequency, which was originally defined for sines and cosines only, to other functions.

In recent years more general classes of complete systems of orthogonal functions have been used for theoretical investigations as well as equipment design. Furthermore, semiconductor device technological developments have made it practical to use digital devices to perform filtering functions which were heretofore dedicated to implementation by linear devices.

The purpose of this work is to investigate the
application of the generalized frequency analysis (referred to as sequency analysis) concept to resonant filtering of time dependent signals. The investigation is based on (a) the orthonormal set of Walsh functions which are square wave type functions that oscillate between plus one (+1) and minus one (-1) in magnitude and (b) the Harmuthian resonance concept.

The research conducted for this dissertation is presented in the next seven chapters. Chapter 2 emphasizes the mathematical characteristics, representation methods, and generation techniques for Walsh functions while accentuating those concepts which are most important to sequency filtering of these type functions. Chapter 3 has the primary objective of establishing the relationship between the matched and resonant filtering concepts in the sequency domain. Chapter 3 also introduces the developmental history of the theory of the resonant-type filters for Walsh functions as well as an important application of such filters.

Chapter 4 discusses the theoretical foundation of the resonant filter concept by using electrical elements (inductors, capacitors and switches) as devices which are controlled such that the system output indicates resonance versus anti-resonance. Using Laplace Transform theory, two analyses are presented; one under the assumption of ideal
elements and one under the assumption of non-ideal elements.

In Chapter 5 it is realized that the resonant filter is essentially a sampled-data system and, as such, its operation may be represented by difference equations. These equations are developed and simulated on a general purpose digital computer using the Fortran IV programming language. The system difference equations are solved analytically in Chapter 6 using Z-transform theory. Chapter 7 uses the difference equations as a basis for presentation of the hardware architecture and functional operation of a special purpose digital computer to realize resonant filters for Walsh functions.

Chapter 8 discusses conclusions drawn from the previous sections while Appendix A lends support to the Z-transform analysis of Chapter 7 and Appendix B is used for simulation data presentation.
CHAPTER 2

THE WALSH FUNCTIONS

The continuous Walsh functions were originally defined by their namesake J. L. Walsh [1] in 1923. Generally speaking, these functions take on only the two values of plus one (+1) and minus one (-1) and are defined on the unit interval. In other words, they are square-wave type functions which oscillate between +1 and -1 in magnitude. In recent years there have emerged several notations in mathematics and engineering literature for mathematically describing the Walsh functions and, also, many techniques have been discussed for their electronic generation. It is the purpose of the chapter to emphasize the mathematical characteristics and representation methods, as well as generation techniques, for Walsh functions. The objective will be to accentuate those concepts which are important with respect to sequency filtering of Walsh functions.
2.1 REPRESENTATION OF WALSH FUNCTIONS BY DIFFERENCE EQUATIONS

Walsh functions can be defined by a difference equation [2] which yields the sequence shown in Figure 1. In the figure, the functions are ordered according to the number of sign changes or zero crossings in the half open interval \(-1/2 < \theta < 1/2\). That is, the independent variable on this interval is \(\theta\), a normalized time which is equal to \(t/T\) where \(t\) is ordinary time and \(T\) is the time unit used to normalize the time \(t\). The functions \(c_{al}(n,0)\) and \(s_{al}(n,0)\) have \(2n\) zero crossings in the interval; \(n=1,2,3,\ldots\). Walsh functions are characterized by their "sequency" which is a generalized frequency defined as one-half the average number of zero crossings per second (zp/s). This notation is analogous to the sine and cosine functions normally used in frequency analysis. Furthermore, all functions \(c_{al}(n,0)\) equal +1 and all functions \(s_{al}(n,0)\) change from -1 to +1 for \(\theta = 0\).

From Figure 1 we may write that, generally:

\[
\begin{align*}
\text{wal}(2n-1,\theta) &= \text{sal}(n,\theta) & -1/2 < \theta < 1/2 \\
\text{wal}(2n,\theta) &= \text{cal}(n,\theta) & -1/2 < \theta < 1/2 \\
\end{align*}
\]

(1)

\[
\text{wal}(2n-1,\theta) = \text{wal}(2n,\theta) = 0 \quad \theta < -1/2, \quad \theta > 1/2
\]

except for the special case of \(n=0\) where all we have is \(\text{wal}(0,\theta)\).

Using this notation we may write difference equations for the Walsh functions in the following form:
Figure 1. The Walsh Functions $\text{wal}(n,0)$ for $n = 0, 1, 2, \ldots, 15$. 
wal(2k + q, θ) = (-1)^{[k/2]} wal(k, 2θ + 1/2) + (-1)^{k+q} wal(k, 2θ - 1/2) \quad (2)

wal(0, θ) = \begin{cases} 
1 & \text{if } -1/2 < θ < 1/2 \\
0 & \text{if } θ < -1/2 \text{ or } θ > 1/2 
\end{cases}

q = 0 \text{ or } 1, \quad k = 0, 1, 2, \ldots

where \([k/2]\) means the largest integer smaller than or equal to \(k/2\).

For explanation of the difference equation, consider the Walsh function wal(0, θ) of Figure 1. Shifting it by 1/2 to the left yields

wal(0, θ + 1/2), \quad -1 \leq θ < 0

and compressing it by a factor of 2 yields

wal(0, 2θ + 1/2), \quad -1/2 \leq θ < 0.

Similarly

wal(0, 2θ - 1/2), \quad 0 \leq θ < 1/2

is obtained by shifting wal(0, θ) to the right and compressing it. For example, let \(k = q = 0\), then equation (2) reduces to

wal(0, θ) = (-1)^{0+0} wal(0, 2θ + 1/2) + (-1)^{0+0} wal(0, 2θ - 1/2)

= wal(0, 2θ + 1/2) + wal(0, 2θ - 1/2)

= cal(0, θ)

which is evidently correct. For \(k = 2, q = 1\) equation (2) reduces to

wal(5, θ) = (-1)^{1+1} [wal(2, 2θ + 1/2) + (-1)^{2+1} wal(2, 2θ - 1/2)]
It is obvious that any Walsh function may be constructed from this difference equation by choosing the proper values of $k$ and $q$ and mathematical reduction using the Walsh function definitions.

There are other recursive-type or difference equations that have been presented in the literature for representing Walsh functions [3]. One of the main advantages of the above type of representation is that the derived Walsh functions are ordered such that one may write

$$\text{wal} (2n, \theta) = \text{cal} (n, \theta)$$

and

$$\text{wal} (2n - 1, \theta) = \text{sal} (n, \theta)$$

where $\text{cal} (n, \theta)$ and $\text{sal} (n, \theta)$ are even and odd functions (analogous to cosine and sine functions, respectively) such that $n$ represents the sequency of the function. The point being that there are other methods of representing the Walsh functions which do not yield this same well ordered result [2]. (The basis for the $\text{cal}$ and $\text{sal}$ notation usage should be clear at this point, i.e., the $c$ and $s$ correspond to cosine and sine functions and the $a$ corresponds to Walsh.)

It may be concluded that difference equations for Walsh functions are, in general, complex and "hard to
remember." To alleviate these problems, Walsh functions are usually represented as products of Rademacher functions [4] which are a subset of Walsh functions.

2.2 REPRESENTATION OF WALSH FUNCTIONS AS PRODUCTS OF RADEMACHER FUNCTIONS.

Rademacher functions are square wave functions, with odd symmetry, having two arguments, \( n \) and \( \theta \), such that \( r(n, \theta) \) has \( 2^{n-1} \) periods in the interval \(-1/2 < \theta < 1/2\). The amplitudes of these functions are plus one (+1) and minus one (-1) -- thus, we realize them to be a subset of the Walsh functions. Figure 2 shows the Rademacher functions for \( n = 1, 2, 3, \) and 4 on the normalized \( \theta = t/T \) scale. Note the relationship: \( r(n, \theta) = sal(2^{n-1}, \theta) = wal(2^{n-1}, \theta) \).

A method for generating the family of Walsh functions by taking products of Rademacher functions was presented without proof by Lackey and Meltzer [5] and later proven to be generally valid by Davies [6]. This method is as follows. To form the Walsh function \( wal(n, \theta) \), first form the straight binary representation of \( n \), then form the Gray code version of \( n \), and multiply the Rademacher functions corresponding to the 1 bits in the Gray code. The resultant product will be the required Walsh function.

The rule for forming Gray code numbers is as follows. For bit number \( i \) in the Gray code, add bit \( i \)
Figure 2. The Rademacher Functions for $n = 1, 2, 3,$ and $4$. 
to bit \((i+1)\) of the original binary number, modulo 2. Counting is done from the least significant bit. We can express this rule mathematically by the following relationships: If \(n\) in binary is

\[
n = b_m b_{m-1} b_{m-2} \cdots b_1
\]  

then \(n\) in Gray code is:

\[
n = g_m g_{m-1} g_{m-2} \cdots g_1
\]

where

\[
g_i = b_i \oplus b_{i+1}
\]

where \(\oplus\) indicates addition modulo 2, with no carries, and

\[
b_{m+1} = 0 \text{ so } g_m = b_m
\]

The application of the above procedure for the generalization of the Walsh function \(\text{wal}(13, 0)\) would proceed as follows:

Step 1. \(n = (13)_{10} = (1101)_{2}\) and \(m = 4\)

Step 2. \((1101)_2 = (b_4 b_3 b_2 b_1)_2 = (g_4 g_3 g_2 g_1)_{\text{Gray}} = (1011)_{\text{Gray}}\)
where
\[ g_4 = b_4 \oplus b_5 = 0 \oplus 1 = 1 \]
\[ g_3 = b_3 \oplus b_4 = 1 \oplus 1 = 0 \]
\[ g_2 = b_2 \oplus b_3 = 0 \oplus 1 = 1 \]
\[ g_1 = b_1 \oplus b_2 = 1 \oplus 0 = 1 \]

Step 3. From step 2 we see that the Gray code bits which are equal to 1 are bits 4, 2, and 1; therefore the product of Rademacher functions required to generate \( \text{wal}(13, 0) \) is \( r(4, 0), r(2, 0), \) and \( r(1, 0), \) i.e.,
\[ \text{wal}(13, 0) = r(4, 0) r(2, 0) r(1, 0) \]

This product is further illustrated by Figure 3.

We now have a complete definition, mathematical and graphical, of Walsh functions as well as a simple method of mathematically generating the functions based on the Rademacher functions. The next logical concept to consider is the special mathematical properties exhibited by the Walsh functions considered as a general set of functions. Accordingly, the following section will consider the mathematical properties of Walsh functions with particular emphasis being placed on these properties which are important with respect to sequency filtering.
Figure 3. Illustration showing that $\text{wal}(13, \theta) = r(1, \theta) \cdot r(2, \theta) \cdot r(4, \theta)$. 

$\theta = \frac{t}{T}$
2.3 MATHEMATICAL PROPERTIES OF THE SET OF WALSH FUNCTIONS.

The fundamental theory underlying the Walsh functions was presented by Walsh [1] in 1923. The functions are closely associated with matrices made up of plus ones (+1's) and minus ones (-1's) which were first studies in 1893 by Hadamard [7, 8] and referred to as Hadamard matrices. Many additional papers concerning the theory of the Walsh functions have appeared in recent years [1, 9, 10, 11, 12].

The most important mathematical properties with respect to sequency filtering of Walsh functions is that the set of functions is complete and orthonormal. Because of the special significance of these concepts, this section presents the mathematical foundations on which orthonormality and completeness are based. Also, the concept of closure and its relationship to the Walsh functions are discussed.

ORTHONORMAL SYSTEMS OF FUNCTIONS

A system \( \{f(n, \theta)\} \) of real and almost everywhere non-vanishing functions \( f(0, \theta), f(1, \theta), \ldots \) is called orthogonal in the interval \( \theta_0 \leq \theta \leq \theta_1 \), if the following conditions hold true:

\[
\int_{\theta_0}^{\theta_1} f(n, \theta) f(m, \theta) d\theta = \begin{cases} 
\frac{p_n}{n} & \text{if } n = m \\
0 & \text{if } n \neq m 
\end{cases}
\]  

(7)
The functions are called orthogonal and normalized if the constant $P_n$ is equal to 1. These two terms are usually reduced to the single term: orthonormal or orthonormalized.

It is easy to show that the set of Walsh functions \{wal (n, $\theta$)\} are orthonormal on the interval $-1/2 < \theta < 1/2$ according to the general definition of orthogonality since their products are always either $+1$ or $-1$ for any given subinterval. As an example, consider the product of the two Walsh functions: $\text{wal} (0, \theta)$ and $\text{wal} (1, \theta)$. This product is equal to $-1$ on the interval $-1/2 < \theta < 0$ and $+1$ in the interval $0 < \theta < 1/2$. The integral of this product has the following value:

\[
\int_{-1/2}^{1/2} \text{wal} (0, \theta) \text{wal} (1, \theta) \, d\theta = \int_{-1/2}^{1/2} \text{wal} (0, \theta) \text{wal} (1, \theta) \, d\theta + \int_{0}^{1/2} \text{wal} (0, \theta) \text{wal} (1, \theta) \, d\theta
\]

\[
= \int_{-1/2}^{1/2} (+1) (-1) \, d\theta + \int_{0}^{1/2} (+1) (+1) \, d\theta
\]

\[
= [-0 + 1/2] + [1/2 - 0] = -1/2 + 1/2 = 0
\]

Using this procedure one may easily verify that the integral of the product of any two of the Walsh functions
is equal to zero. A function multiplied by itself yields the products (+1) (+1) or (-1) (-1). Hence, these products have the value 1 in the whole interval \(-1/2 < \theta < 1/2\) and their integral is therefore 1. The Walsh functions are thus orthonormal.

**COMPLETE SYSTEM OF FUNCTIONS**

Let a function \(F(\theta)\) be expanded in a series of the orthonormal system \(\{f(n, \theta)\}\):

\[
F(\theta) = \sum_{n=0}^{\infty} c(n) f(n, \theta) \tag{8}
\]

where \(c(n)\) is the series expansion coefficient and may be obtained by multiplying the above equation by \(f(k, \theta)\) and integrating the products in the interval of orthogonality \(\theta_0 < \theta < \theta_1\):

\[
c(k) = \frac{\theta_1 - \theta_0}{\theta_1 - \theta_0} F(\theta) f(k, \theta) \, d\theta \tag{9}
\]

Note that the series definition of \(F(\theta)\) requires an infinite number of series terms. A question arises with respect to the "best" representation of \(F(\theta)\) if the \(c(n)\) coefficients are calculated using the last equation and the practical limit on \(n\) is placed at some finite integer \(N\). The criterion that is standardly used for "best" is the mean square error of deviation, \(e(n)\), of \(F(\theta)\) from its representation:
Here we assume that the series $\sum_{n=0}^{N} b(n)f(n, \theta)$ yields the "best" approximation for $F(\theta)$. Expanding the right side of equation (10):

$$\varepsilon(N) = \int_{\theta_0}^{\theta_1} F(\theta) \, d\theta - 2\sum_{n=0}^{N} b(n) \int_{\theta_0}^{\theta_1} F(\theta)f(n, \theta) \, d\theta + \int_{\theta_0}^{\theta_1} \left[ \sum_{n=0}^{N} b(n)f(n, \theta) \right] \, d\theta$$

Using equation (9) and the orthogonality of the functions $f(n, \theta)$ yields $\varepsilon(N)$ in the following form:

$$\varepsilon(N) = \int_{\theta_0}^{\theta_1} F(\theta) \, d\theta - \sum_{n=0}^{N} c^2(n) + \sum_{n=0}^{N} [b(n) - c(n)]^2.$$  \hspace{1cm} (11)

The last term vanishes for $c(n) = b(n)$ and the mean square deviation assumes its minimum.

The so called Bessel inequality follows from (11):

$$\sum_{n=0}^{N} c^2(n) \leq \int_{\theta_0}^{\theta_1} F^2(\theta) \, d\theta.$$  \hspace{1cm} (12)

The upper limit of the summation may be infinity ($\infty$) instead of $N$, since the integral does not depend on $N$ and must thus hold for any value of $N$.

The system $\{f(n, \theta)\}$ is called orthonormal and complete, if $\varepsilon(N)$ converges to zero with increasing $N$ for any function $F(\theta)$ that is quadratically integrable in the interval $\theta_0 \leq \theta \leq \theta_1$:

$$\lim_{N \to \infty} \int_{\theta_0}^{\theta_1} \left[ F(\theta) - \sum_{n=0}^{N} c(n)f(n, \theta) \right]^2 \, d\theta = 0.$$  \hspace{1cm} (13)
The equality sign holds in this case in the Bessel inequality so:

\[ \sum_{n=0}^{\infty} c_n^2 = \int_{\theta_0}^{\theta_1} F^2(\theta) \, d\theta. \]  \tag{14}

Equation (14) is known as the completeness theorem or Parseval's theorem. Its physical meaning is as follows: Let \( F(\theta) \) represent a voltage as a function of time across a unit resistance. The integral of \( F^2(\theta) \) represents, then, the energy dissipated in the resistor. This energy equals, according to (14), the sum of the energy of the terms in the sum \( \sum c(n)f(n,\theta) \). Stating the same idea differently, the energy is the same whether the voltage is described by the time function \( F(\theta) \) or its series representation. The Walsh functions are a complete set [10].

In signal processing or filtering terms, the orthogonal time series representation converts input data to a set of coefficients or spectrum numbers. Thus, for a selected set of orthogonal functions and a given input signal, \( f(t) \), the input can be uniquely represented by the series coefficients \( c(n) \) or by the series itself.

**THE MULTIPLICATION THEOREMS FOR WALSH FUNCTIONS**

Mathematically speaking, a complete, orthonormal system is always closed under the operation of multiplication. This property of closure means that the
product of two Walsh functions yields another Walsh function:

$$\text{wal}(h, \theta) \times \text{wal}(k, \theta) = \text{wal}(p, \theta).$$

This relation may readily be proved by writing the difference equation for \text{wal}(h, \theta) and \text{wal}(k, \theta), and multiplying them together. The resultant product \text{wal}(h, \theta) \times \text{wal}(k, \theta) satisfies a difference equation of the same form as equation (2). The determination of \(p\) from the difference equation (2) is somewhat cumbersome but can be shown to be the modulo 2 sum of \(h\) and \(k\):

$$\text{wal}(h, \theta) \times \text{wal}(k, \theta) = \text{wal}(h \oplus k, \theta). \quad (15)$$

As an example, consider the following multiplication:

$$\text{wal}(6, \theta) \times \text{wal}(12, \theta) = \text{wal}(p, \theta).$$

We may determine \(p\) by looking at the waveshape of \text{wal}(6, \theta) and \text{wal}(12, \theta) in Figure 1 or as follows by realizing that 
\((6)_{10} = (0110)_2\) and \((12)_{10} = (1100)_2\) such that \([(0110)_2 + (1100)_2] \mod 2 = (1010)_2 = (10)_{10}\). Therefore \(p = 10\). It should be noted that the product of a Walsh function with itself yields \text{wal}(0, \theta), since only the products \((+1) (+1)\) and \((-1) (-1)\) occur:

$$\text{wal}(k, \theta) \times \text{wal}(k, \theta) = \text{wal}(0, \theta) \quad (16)$$
where \( k \oplus k = 0 \). Also, the product of \( \text{wal}(k, \theta) \) with \( \text{wal}(0, \theta) \) leaves \( \text{wal}(k, \theta) \) unchanged:

\[
\text{wal}(k, \theta) \text{wal}(0, \theta) = \text{wal}(k, \theta)
\]  

(17)

where \( k \oplus 0 = k \).

The other important multiplication theorems which come directly from equations (1) and (15) are:

\[
\text{cal}(n, \theta) \text{cal}(k, \theta) = \text{cal}(n \oplus k, \theta)
\]  

(18)

\[
\text{sal}(n, \theta) \text{cal}(k, \theta) = \text{sal}([k \oplus (n-1)] + 1, \theta)
\]

\[
\text{cal}(n, \theta) \text{sal}(k, \theta) = \text{sal}([n \oplus (k-1)] + 1, \theta)
\]

\[
\text{sal}(n, \theta) \text{sal}(k, \theta) = \text{cal}[(n-1) \oplus (k-1), \theta]
\]

\[
\text{cal}(0, \theta) = \text{wal}(0, \theta).
\]

2.4 GENERATION OF THE WALSH FUNCTIONS

Although an exhaustive discussion of electronic generation of the Walsh functions is not necessary, relative to sequency filtering, such a discussion is important for completeness in a general presentation of Walsh function theory so as to illustrate the difference between the analog Walsh waveforms and digital Walsh waveforms. To this end, this section serves to point out this difference and to indicate the general concepts on which electronic generation of Walsh functions are based.
GENERAL CONCEPTS OF ANALOG WALSH WAVEFORM GENERATION [13]

One period of a Walsh function is taken to be of duration $T$ seconds. In the Walsh function set, as shown in Figure 1, $t$ is in the range $-T/2 \leq t < T/2$ so that $\theta = \frac{t}{T}$ is in the range $-1/2 \leq \theta < 1/2$. $Sal(1, \theta)$ is a function defined as

$$Sal(1, \theta) = \begin{cases} -1 & -1/2 \leq \theta < 0 \\ +1 & 0 \leq \theta < 1/2 \end{cases}$$ (19)

Beginning with this function, all other Walsh functions can be determined in the following manner:

1. Let the general terms for the $cal$ and $sal$ functions be represented by

$$sal[(2P + 1) 2^k, \theta]$$

$$cal[(2P + 1) 2^k, \theta]$$

with $P = 0, 1, 2, 3, \ldots$, and $k = 0, 1, 2, 3, \ldots$

2. Any function that has a sequency $2^k$ times that of a function with a particular value of $P$ has exactly the same form as that function but is compressed by the factor $2^k$. Thus, all binary-sequence multiples of $sal(n, \theta)$ and $cal(n, \theta)$ have the same waveshape.

3. In order to determine the $cal$ function from the corresponding $sal$ function of the same sequency, the
following rules apply:

a. If \( P \) is even, then, for a particular value of \( k \), the sal function is shifted \( T/2^{k+2} \) to the left (or in the negative direction) to produce the cal function.

b. If \( P \) is odd, then the sal function is shifted \( T/2^{k+2} \) to the right (or in the positive direction) to produce the cal function.

4. For any given cal function, the sal function with a sequency that is one greater than the cal function may be produced from

\[
\text{sal}[2P + 1]2^k + 1, \theta] = -\text{cal}[2P + 1]2^k, \theta] -1/2 \leq \theta < 0 \quad (20)
\]

\[
= \text{cal}[2P + 1]2^k, \theta] \quad 0 \leq \theta < 1/2
\]

Thus the analog waveform for \( \text{sal}(n + 1, \theta) \) is derived from \( \text{cal}(n, \theta) \) by inverting \( \text{cal}(n, \theta) \) for \( \theta \) in the range \(-1/2 \leq \theta < 0\) and retaining \( \text{cal}(n, \theta) \) for \( \theta \) in the range \( 0 \leq \theta < 1/2\). This is readily seen in Figure 1. Starting from \( \text{wal}(0, \theta) \), the sal and cal waveforms can be evolved in ascending order of sequency by the method just described.

GENERAL CONCEPTS OF DIGITAL WALSH WAVEFORM GENERATION

As indicated earlier, the analog Walsh waveforms
vary from +1 to -1 in a manner analogous to the function set illustrated in Figure 1. If one allows the Walsh value +1 to be designated by a logical one and the Walsh value -1 to be designated by the logical zero, the Walsh functions may be generated by using a sequential digital system.

The basic procedural difference between analog and digital generation is that the analog technique begins by assuming the lowest sequency term as indicated in the previous section whereas the digital technique begins with the highest sequency term and generates the waveform of the lower sequency terms. To illustrate the mechanics involved in digital generation of the Walsh functions, consider the following example:

Assume the availability of a 4-bit binary up-counter made of trigger-type flip-flops operating with a clock signal that pulses the counter 16 times during the time T. One readily realizes the sal(8, 0) is represented by the output of the low order flip-flop and, also, sal(4, 0), sal(2, 0) and sal(1, 0) are the outputs of the other three stages of the counter. Further, if the Walsh functions are represented by sequences of logic ones and zeros as in Table 1, the cal(1, 0), cal(4, 0) and cal(5, 0) may be formed from sal(1, 0), sal(2, 0), sal(4, 0), and sal(8, 0). Sal(8, 0), given in Table 1 by column d, is
### Table 1

**Truth Values of Walsh Functions**

*For $0 \leq \theta < 1$*

<table>
<thead>
<tr>
<th>Given Inputs</th>
<th>Required Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r(1,\theta)$</td>
<td>$cal(1,\theta)$</td>
</tr>
<tr>
<td>$r(2,\theta)$</td>
<td>$cal(4,\theta)$</td>
</tr>
<tr>
<td>$r(3,\theta)$</td>
<td>$cal(5,\theta)$</td>
</tr>
<tr>
<td>$r(4,\theta)$</td>
<td></td>
</tr>
<tr>
<td>$sal(1,\theta)$</td>
<td></td>
</tr>
<tr>
<td>$sal(2,\theta)$</td>
<td></td>
</tr>
<tr>
<td>$sal(4,\theta)$</td>
<td></td>
</tr>
<tr>
<td>$sal(8,\theta)$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
the highest sequency term, and sal(4, \theta), sal(2, \theta),
sal(1, \theta) consist of consecutive binary count-down terms,
shown by column c, b, and a, respectively. Using Table 1
and Karnaugh maps, shown in Figure 4, the truth values
for columns e, f, and g, in terms of a, b, c, and d, are

\[
\begin{align*}
\text{cal}(1, \theta) &= e = \bar{a} \bar{b} + ab \\
&\quad = a \odot b, \tag{21} \\
\text{cal}(4, \theta) &= f = \bar{c} \bar{d} + cd \\
&\quad = c \odot d \tag{22} \\
\text{cal}(5, \theta) &= g = \bar{a} \bar{b} (\bar{c} \bar{d} + cd) + ab (\bar{c} \bar{d} + cd) \\
&\quad + \bar{a} b (\bar{c} d + c \bar{d}) + a \bar{b} (\bar{c} d + c \bar{d}) \\
&\quad = (\bar{a} \bar{b} + ab)(\bar{c} \bar{d} + cd) + (\bar{a} b + a \bar{b})(\bar{c} d + c \bar{d}) \\
&\quad = e f + \bar{e} \bar{f} \tag{23} \\
&\quad = e \odot f \\
&\quad = a \odot b \odot c \odot d \\
&\quad = \text{sal}(1, \theta) \odot \text{sal}(2, \theta) \odot \text{sal}(4, \theta) \odot \text{sal}(8, \theta)
\end{align*}
\]

If the relationship from Figure 2 (i.e., \( r(n, \theta) =
\text{sal}(2^{n-1}, \theta) \)) is substituted into equation (23), we then
have

\[
\text{cal}(5, \theta) = r(1, \theta) \odot r(2, \theta) \odot r(3, \theta) \odot r(4, \theta) \tag{24}
\]

The symbol \( \odot \) is the standard coincidence notation which
can be realized with NAND and NOR gates. By extending
this method, it can be shown that, in general, the forma-
Figure 4. Karnaugh Maps used to find additional Walsh functions.
tion of any Walsh function requires only a coincidence gate involving two other Walsh functions. Table 2 shows the first fifteen Walsh functions generated by this method.

It should be noted that there are other methods that have been developed for generating the Walsh functions [73]. The purpose here has been only to illustrate the concepts involved in function generation.

2.5 APPLICATIONS OF WALSH FUNCTIONS

Since Walsh's original paper, Walsh function theory has been further developed and equated to Fourier theory by Paley [9, 14] in the early thirties and Fine [10, 15, 16] in the late forties. More recently, Pichler [17] developed the mathematical basis for sequency notation and the "sal" and "cal" terms often used in Walsh analysis. A development that has particular engineering appeal was reported by Chrestenson [18]. He generalized the Walsh functions from a system of bi-valued functions to a system of n-valued functions, thereby providing a conceptual bridge for engineers from Walsh to Fourier. A relationship between Walsh functions and Hadamard matrices has often been noted and has been reported on by Welch [19]. In addition, Kane [20] reported the use of Walsh functions in matrix inversion.
### TABLE 2

**RELATIONSHIP BETWEEN WALSH AND COINCIDENCE OF SAL FUNCTIONS**

<table>
<thead>
<tr>
<th>val(n, 6)</th>
<th>sal(n, 6)</th>
<th>sal(n, 6) / cal(n, 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>val(0, 6)</td>
<td>1</td>
<td>cal(0, 6)</td>
</tr>
<tr>
<td>val(1, 6)</td>
<td>sal(1, 6)</td>
<td>sal(1, 6)</td>
</tr>
<tr>
<td>val(2, 6)</td>
<td>sal(1, 6) ⊕ sal(2, 6)</td>
<td>cal(1, 6)</td>
</tr>
<tr>
<td>val(3, 6)</td>
<td>sal(2, 6)</td>
<td>sal(2, 6)</td>
</tr>
<tr>
<td>val(4, 6)</td>
<td>sal(2, 6) ⊕ sal(4, 6)</td>
<td>cal(2, 6)</td>
</tr>
<tr>
<td>val(5, 6)</td>
<td>sal(1, 6) ⊕ sal(2, 6) ⊕ sal(4, 6)</td>
<td>sal(3, 6)</td>
</tr>
<tr>
<td>val(6, 6)</td>
<td>sal(1, 6) ⊕ sal(4, 6)</td>
<td>cal(3, 6)</td>
</tr>
<tr>
<td>val(7, 6)</td>
<td>sal(4, 6)</td>
<td>sal(4, 6)</td>
</tr>
<tr>
<td>val(8, 6)</td>
<td>sal(4, 6) ⊕ sal(8, 6)</td>
<td>cal(4, 6)</td>
</tr>
<tr>
<td>val(9, 6)</td>
<td>sal(1, 6) ⊕ sal(2, 6) ⊕ sal(8, 6)</td>
<td>sal(5, 6)</td>
</tr>
<tr>
<td>val(10, 6)</td>
<td>sal(1, 6) ⊕ sal(2, 6) ⊕ sal(4, 6) ⊕ sal(8, 6)</td>
<td>cal(5, 6)</td>
</tr>
<tr>
<td>val(11, 6)</td>
<td>sal(2, 6) ⊕ sal(4, 6) ⊕ sal(8, 6)</td>
<td>sal(6, 6)</td>
</tr>
<tr>
<td>val(12, 6)</td>
<td>sal(2, 6) ⊕ sal(8, 6)</td>
<td>cal(6, 6)</td>
</tr>
<tr>
<td>val(13, 6)</td>
<td>sal(1, 6) ⊕ sal(2, 6) ⊕ sal(8, 6)</td>
<td>sal(7, 6)</td>
</tr>
<tr>
<td>val(14, 6)</td>
<td>sal(1, 6) ⊕ sal(8, 6)</td>
<td>cal(7, 6)</td>
</tr>
<tr>
<td>val(15, 6)</td>
<td>sal(8, 6)</td>
<td>sal(8, 6)</td>
</tr>
</tbody>
</table>
Investigations in coding theory applications of Walsh functions has been the source of much activity. Though much use has been made of the Reed-Muller codes, it is seldom realized that they are identical to Walsh functions. Also, a number of communications systems devised and built in the sixties [8, 21, 22, 23, 24, 25, 26] employed Walsh function formats but appeared to be unaware of their mathematical foundation. However, investigators, principally in West Germany, have developed multiplex systems designed to use Walsh function coders and decoders as reported by Harmuth [12] and Huebner [27]. These efforts have been extended, as reported by Huebner [28], to a multiplex system using Walsh functions as the carrier.

An early investigation of Walsh function analysis of linear networks was reported by Hammond and Johnson [29]. A related but independent effort by Corrington and Adams [30] applied Walsh functions to non-linear analysis. Many aspects of signal processing have been and are under continued investigation. These include correlation by Lueke [31, 32] bandwidth requirements by Schreiber [33], detection and sorting by Brown [34] and significant contributions to signal processing in general by Guinn [35] and Brown [36].
The use of Walsh functions in linear system theory has been investigated by Pichler [17], while Weiser [37] investigated the use of Walsh functions in non-linear stochastic problems. In addition, Nambiar [38] reports on the use of Walsh functions in probability problems; also, Ohnsorg [39] extends the concept of Walsh functions to include imaginary values to describe an operating sonar processing technique.

Of strong interest to communications engineers is the study of speech analysis and compression techniques. The basic problem is to maximize the efficiency of voice transmission while minimizing distortion in the character of the voice. Considerable work has been done over recent decades in providing detailed Fourier analysis of speech formants and related characteristics. However, in recent years, additional analytic work on speech signals has been done in the sequency domain using Walsh functions. Robinson and others [40] at COMSAT Laboratories report significant success in bandwidth reduction in the sequency domain.

Researchers have also applied Walsh functions in the areas of optical spectral processing. Investigations by Gibbs [41, 42, 43, 44, 45] have demonstrated the usefulness of Walsh functions in digital computations required by spectral analysis. In addition, his interest
in the area has led him to fundamental considerations of Walsh functions, logical equations and binary computations. Gebbie [46] has used Walsh functions in transform spectroscopy.

The radiation of "Walsh-waves" represents an interesting and fundamentally important area of research. Harmuth [12] has provided original theoretical work, and experimental work has been done under his direction [47].

Pratt and Andrews have done extensive work in various areas of digital image processing [48, 49, 50, 51, 52, 53]. They have investigated the use of various transform domains including Fourier, Harr, and Hadamard/Walsh. In addition, Carl [55, 56] has been using Walsh functions to provide image classification which has lead to pattern recognition solutions. Also, Meltzer [57, 58, 59] and Searle [60] have investigated the use of Walsh functions in the recognition of biological patterns and in the fundamental studies of shape analysis.

Numerous filtering techniques have been devised for use with Walsh functions. A survey of various analog techniques is given by Harmuth [12, 61]. Vandivere [62] reports a flexible sequency analyzer for use as an investigative tool as well as a real time bandwidth compressor. Roth [63] describes a "V-filter" or comb-sequency like filter and its uses in problems with variable sampling
rates. Many investigators have reported uses for fast Walsh transformations in digital filtering both in hardware and software.

More comprehensive surveys and bibliographies of Walsh function applications may be found in papers by Lee [64] and, more recently, by Lackey [65]. The most comprehensive source of information concerning Walsh function theory and applications is probably the proceedings of the annual symposium on "Applications of Walsh Functions." This symposium has been held for the last five years in Washington, D. C., under the sponsorship of the Naval Research Laboratory, the IEEE-EMC Group and the University of Maryland.
CHAPTER 3

SEQUENCY FILTERS BASED ON WALSH FUNCTIONS

Basically, there are two types of sequency filters for Walsh functions -- matched and resonant. Most of the research work in recent years has been associated with filters of the matched type [2, 12, 13, 66]. In 1970, Harmuth [61] defined his conception of the analog resonant sequency filter which was incited a few researchers to perform work towards their design, analysis, and development [67, 68].

The fundamental difference between the matched and resonant sequency filters for Walsh functions parallels that for the standard matched and resonant frequency filters for sinusoidal functions. That is, the matched filter operates in a mode which essentially destroys the input signal waveform but retains its characteristics such that the signal can be re-established if desired and tends to maximize the signal-to-noise ratio between the input signal and the desired output signal. The resonant filter operates in a mode which produces an output signal which
very closely resembles the input signal except for some
time delay and tends to minimize the mean-square-error
between the input signal and the desired output signal.

This chapter deals primarily with the "idea" of
sequency filtering, in general. The objectives are to
establish the general concepts, to outline the historical
development and to give an example application of resonant
sequency filters, in particular.

3.1 MATCHED SEQUENCY FILTERS

When the input signal is a deterministic time func­
tion, one possible criterion for optimizing a system is to
maximize the output signal to noise ratio (SNR) at some
specified time. This criterion is particularly useful
when the major objective of the system is the detection of
a signal of known shape in the presence of noise. In some
circumstances, such as in the detection of a radar signal,
the form of the output time function is not important;
merely the presence or absence of a signal is important.
The criterion mentioned above can be accomplished by the
correlation of an unknown waveform with a known waveform
by passing the unknown waveform through a linear system
whose impulse response is the time reverse of the known
waveform and observing the output at any instant of time.
If the waveforms are made the same, the filter is said to
be matched to the input waveform. The filter output as
a function of real time is then the auto-correlation function of the input waveform. This type of filter is called a matched filter. The matched filter output usually has little resemblance to the matching signal, even though the matching signal can be derived from the matched filter output.

This work is not specifically involved with the matched filter concept except to clarify its characteristics with respect to those of the resonant filter concept.

3.2 RESONANT SEQUENCY FILTERS

The maximum signal-to-noise ratio criterion is not suited to situations where it is desired that the output signal reproduce the input waveform as closely as possible at all times. A more satisfactory criterion is one which minimizes the mean-square-error between the input signal and the desired output signal. It has been established that the resonance filter gives the optimum filter when the minimum mean-square-error between the input and output criterion is applied [69]. Also, it has been shown that resonance filters for Walsh functions exist [61]. However, until recently, no analog sequency resonant filters had been built [67]. Such filters require time variable elements in addition to the inductors, capacitors, and resistors used for resonance filters based on sine-cosine
functions. For analog resonant sequency filters based on Walsh functions, switches are the time variable elements, and inductors and capacitors are the other elements used to implement the LCS filter - a name which was coined by Harmuth [61].

The digital resonant sequency filter based on Walsh functions is derived from the fact that the output of the analog resonant filter is of interest only at discrete time points and therefore may be represented by a difference equation. Once the difference equation for a specific Walsh function is known, implementation of the filter can be performed either in hardware, software or firmware.

3.3 DEVELOPMENTAL HISTORY OF LCS RESONANT SEQUENCY FILTERS

The concept of the resonant sequency filter was first reported by Harmuth [61] in 1970. He defined the analog resonant sequency filter in quite much detail and developed the fundamental mathematics of analysis for such a device utilizing inductors, capacitors and switches. In 1971, Golden and James [13, 67] reported on the design and implementation of the analog resonant sequency filter by presenting a specific circuit along with experimental results of its operation. Then, in 1972, Golden [70] introduced a general purpose digital simulation of Harmuth's
analog serial resonant filter and Nagle [71] proposed special purpose digital hardware for Harmuth's filter based on Z-transform theory.

Based on the development of the theory for the design and analysis of such filters thus far, it is apparent that a detailed study of both the analog and digital form of these types of filter for Walsh functions is in order. This work is guided toward that goal.

3.4 AN APPLICATION OF RESONANT SEQUENCY FILTERS

Data distribution systems that use the "data bus" concept have been proposed that time-multiplex all data in a serial fashion along a channel [72]. Data sinks or terminals have continuous access to the "bus" information and will sample according to preprogrammed patterns or upon command from the data bus controller (DBC). Each data terminal is identical in design philosophy; the only difference being that each must be able to detect its own unique address code. Upon detection of its address, the terminal is instructed to supply information from its buffer memory or to receive commands from the DBC. Since each data terminal need only recognize a small number of the address messages, one method of decoding at the data terminals is shown in Figure 5. This type of circuit would require bit synchronization which is necessary for proper timing of the detection device used. If a biorthog-
Figure 5. Address Message Decoder.
onal code with a self-synchronizing property such as the Reed-Muller code is used for the address codes, the detection may be a digital resonance filter for Walsh functions or an analog resonance filter for Walsh functions. The reason for this possibility is that there is a one-to-correspondence between Walsh functions and the Reed-Muller codes. Walsh functions have a value of 0 or 1. If the following mapping is used

\[ +1 \rightarrow 0, -1 \rightarrow 1, \quad +x \]

Walsh functions become Reed-Muller codes.

Suppose that it becomes necessary to access two or more terminals simultaneously with a command. One way to accomplish this would be to sequency-multiplex the signal together - that is, sum the signals. Then, a serial resonance sequency filter may be used as the detection device in order to separate the multiplexed signals at the data terminal. For example, if the two terminals to be accessed had addresses of \( \text{sal}(1, 0) \) and \( \text{sal}(3, 0) \), the sum of these two signals would be transmitted down the channel. Each terminal would recognize its address from the sum and function according to the command.

Perhaps the most important applications of resonant sequency filters will be related to code detection similar to the data distribution system mentioned in the previous
paragraphs. However, application to digital communication systems development, in general, and to multiplexing, digital filtering and coding, specifically, seem to be areas where future utility of the resonant sequency filter concept will be important.
The analog filter is one whose input signal is assumed to be an arbitrary function of time which is itself a voltage variation analogous to a time varying parameter. The concept of the resonant sequency filter to be used in the following sections is that which was established in section 3.2. To understand the relationship between the standard LC resonant frequency filter theory and the following discussion for resonant sequency filter theory, one only needs to understand the circuit function of the time variable switching element $S$ which acts as a sequency "tuning" element.

The following two sections are concerned with Inductor - Capacitor - Switch (LCS) filters for Walsh functions for the two important cases of (1) ideal element implementation and (2) non-ideal element implementation.

4.1 THE IDEAL SINGLE PASS LCS FILTER

An analog series LCS resonance filter is shown in
Figure 6. To analyze the operation of this filter, assume the initial voltages across the capacitors are $v_1(0) = V_1$ and $v_2(0) = V_2$ and, then, solve for $i(t)$, $v_1(t)$ and $v_2(t)$ for $C_1 = C_2 = C$ and $S_0$ open. If $S$ is closed at $t=0$, the circuit of Figure 6 yields the integro-differential equation:

$$V_1 - L \frac{di(t)}{dt} - V_2 \frac{1}{C_2} \int_0^t i(t)dt - \frac{1}{C_1} \int_0^t i(t)dt = 0$$

$$V_1 - V_2 = L \frac{di(t)}{dt} + \frac{2}{C} \int_0^t i(t)dt \quad (25)$$

Noting that $i(0) = 0$, we may use the Laplace Transform of equation (25) to solve for $i(t)$ as follows:

$$\frac{V_1 - V_2}{S} = L S I(s) + \frac{2}{C} I(s) \quad (26)$$

Solving for $I(s)$ gives:

$$I(s) = \frac{V_1 - V_2}{L} \frac{1}{S^2 + 2/LC} \quad (27)$$

Therefore

$$i(t) = \frac{V_1 - V_2}{\omega L} \sin \omega t \quad (27)$$

where $\omega^2 = 2/LC$. 
Figure 6. The Serial LCS Resonance Circuit.
We may solve for $v_1(t)$ and $v_2(t)$ as follows:

$$v_1(t) = v_1 - \frac{1}{C_1} \int_0^t i(t) \, dt \quad (28)$$

The Laplace transform of equation (28) is:

$$V_1(s) = \frac{V_1}{s} - \frac{1}{s} \frac{I(s)}{C_1} \quad (29)$$

By inserting $I(s)$ from equation (26), we have:

$$V_1(s) = \frac{V_1}{s} - \frac{(V_1 - V_2)}{C_1 L} \frac{1}{s(s^2 + \omega^2)}.$$ 

Therefore

$$v_1(t) = v_1 - \frac{(V_1 - V_2)}{\omega^2 C_1 L} (1 - \cos \omega t). \quad (30)$$

Similarly

$$v_2(t) = v_2 + \frac{(V_1 - V_2)}{\omega^2 C_2 L} (1 - \cos \omega t) \quad (31)$$

Note that since $C_1 = C_2 = C$ and $\omega^2 = 2/LC$, these last two equations may be written as:

$$v_1(t) = v_1 - (1/2) (V_1 - V_2) (1 - \cos \omega t) \quad (32)$$

and

$$v_2(t) = v_2 + (1/2) (V_1 - V_2) (1 - \cos \omega t). \quad (33)$$

It is extremely important to realize that, at $t = \pi/\omega$, the following relationships are true:
\[ i(\pi/\omega) = 0 \]
\[ v_1(\pi/\omega) = V_2 \]
\[ v_2(\pi/\omega) = V_1 \]

That is, the voltages across the capacitors at the times \( t = 0 \) and \( t = \pi/\omega \) are interchanged by the circuit. Consider, now, the function \( s_{1\lambda}(4,6) \) with \( \theta = t/T \) as shown on line 1 of Figure 7. It can be represented by the discrete Walsh function which is the amplitude sample of line 2. Let the switch \( S_0 \) in Figure 6 add the charge \( q = V_1C \) at the times of positive amplitude samples and the charge \( q = -V_1C \) at the times of negative amplitude samples to the charge on the capacitor \( C_1 \). The switch \( S \) shall always be closed as shown by line 3 of Figure 7 (black \( \mathbf{\Box} \) closed). Let the capacitors \( C_1 \) and \( C_2 \) of Figure 6 be initially free of charge. The first positive charge \( q = +V_1C \) applied at time \( t = -T/2 + T/16 = -7T/16 \) according to Figure 7, produces the voltage \( V_1 = q/C \) across the capacitor \( C_1 \). Line 4 of Figure 7 shows \( v_1(t) \) and \( v_2(t) \) according to equations (32) and (33) for \( \omega = 8\pi/T \). The voltage \( v_1(t) \) becomes zero and \( v_2(t) \) becomes \( V_1 \) at the time \( t = -T/2 + 3T/16 = -5T/16 \). A negative charge \( q = -V_1C \) makes \( v_1(t) \) jump to \( -V_1 \) at this moment. During the following time interval of duration \( T = 8\pi/\omega \), the voltages across \( C_1 \) and \( C_2 \) interchange, yielding \( v_1(t) = +V_1 \) and \( v_2(t) = -V_1 \) at the time \( t = -T/2 + 5T/16 = -3T/16 \). A positive charge \( q = +V_1C \) makes \( v_1(t) \) jump from
Figure 7. Timing diagrams for an LCS resonant filter tuned to the function applied to its input terminals.
+V\textsubscript{1} to 2V\textsubscript{1} at this moment and the exchange of voltages begins again. Line 4 in Figure 7 shows how \(v_1(t)\) and \(v_2(t)\) increase proportional with time. This increase is analogous to the increase of the voltage across the capacitor of a serial LC circuit (like Figure 6 except for the switches) fed from a sinusoidal current source in resonance with the circuit. Since the Walsh function has been represented by time samples of its amplitude, the current source must be replaced by a charge source.

Once steady state has been reached \(v_2(t)\) will have the same general shape as \(s(4, 0)\) with the exception that \(v_2(t)\) will be the complement of \(s(4, 0)\) due to the characteristic of the circuit. Note that the value of \(\pm V_1\) is directly proportional to the area of the amplitude sample which is the amount of charge delivered to the capacitor \(C_1\).

As another example, let the function \(s(3, 0)\) be applied to the circuit of Figure 6. The switch \(S\) is operated according to line 7 of Figure 7 where black (■ ■ ■) implies closed and white (■) implies open. The voltages \(v_1(t)\) and \(v_2(t)\) on line 8 increase again proportional with time. The circuit of Figure 6 is therefore a serial resonance circuit for time sampled or discrete Walsh functions.

For examples of anti-resonance, consider Figure 8.
Figure 8. Timing diagrams for an LCS resonance filter not tuned to the functions applied to its input terminals.
Line 1 through 4 show the application of sal(3, θ) to the circuit of Figure 6 if the switch S is closed all the time which means that the circuit is therefore tuned to sal(4, θ). The voltages do not increase with time. Similarly, lines 5 through 8 show the application of sal(4, θ) to a circuit tuned to sal(3, θ) by the switch S. Again, there is no increase of $v_1(t)$ and $v_2(t)$ proportional with time. Figure 9 shows the application of the eight functions, wal(0, θ) through sal(4, θ), to the circuit tuned to sal(3, θ) by the switch S. For simplification, the voltages $v_1(t)$ and $v_2(t)$ between the sampling times $t=7T/16$, $-5T/16$, ... are shown by straight lines rather than the correct curved lines used in Figures 7 and 8. Only wal(0, θ) and sal(3, θ) produce resonance. This is in complete analogy to the serial tuned LC circuit for sinusoidal functions, i.e., feeding a direct current or a sinusoidal current of proper frequency from a current source to such a circuit produces a voltage across the capacitor that increases proportional with time.

Note that the circuit distinguishes between sal(3, θ) and cal(3, θ) even though they both have the same sequency. The reason for this is that the switch S introduces synchronization into the circuit and, therefore, acts as a tuning element. A sinusoidal resonant circuit, of course, will not distinguish between sine and cosine functions at the same frequency.
Figure 9. Timing diagrams for an LCS resonance filter tuned to \( \text{sal}(3,\theta) \) with functions \( \text{wal}(0,\theta) \) to \( \text{sal}(4,\theta) \) applied to its input terminals. \( v_2(t) \) is dashed and \( v_1(t) \) is not dashed in each case.
Figure 10 shows when switch S has to be closed or open to cause the circuit of Figure 6 to be in resonance with respective Walsh functions sal(1, 0) through sal(4, 0). It can readily be seen that S must be closed for intervals of duration $\Delta \theta = 1/8$ or $\Delta T = T/8$ whenever the Walsh function changes sign.

The separation of the sum of two Walsh functions is illustrated in Figure 11. If the input signal is $[\text{sal}(1, 0) + \text{sal}(3, 0)]$ as shown on line 3 with the filter tuned to $\text{sal}(3, 0)$, as indicated by line 5, the output voltages $v_1(t)$ and $v_2(t)$ are shown on line 6 of Figure 11. Note that $v_2(t)$ clearly indicates the resonance condition.

To summarize, the results of this section thus far serve to establish the characteristics of an ideal analog resonance filter for Walsh functions. The ideas seem clear and correspond favorably with the concepts associated with sinusoidal function theory in the frequency domain. The next logical step in such a presentation might be to investigate the effects of the addition of circuit elements to and/or reconfiguring the circuit of Figure 6 so as to define LCS parallel filters, LCS low pass filters, LCS band pass filter, LCS high pass filters, etc. This type of analysis would serve to further strengthen the fact that sinusoidal analysis is only one of a larger class of signal analysis techniques but would add little in the way of new
Figure 10. Operation times of the switch $S$ in Figure 6 for tuning the filter to the functions $sal(1, \theta)$ to $sal(4, \theta)$. Black areas: $S$ closed; white areas: $S$ open.
Figure 11. Timing diagrams illustrating separation properties for the LCS resonance filter for the sum of two Walsh Functions.
filtering concepts. As a matter of fact, Harmuth [61] has introduced models of a parallel LCS filter as well as a low pass LCS filter and has shown that the analysis of such filters is more complex but similar to that presented here for the serial filter.

Figures 12 and 13 illustrate the conceptual circuit configurations for the parallel and low pass filters. It should be realized that the circuits mentioned thus far can be implemented by using operational amplifiers, capacitors and resistors to replace the inductors (e.g., gyrators) [67].

An important point that should be realized concerns the output voltage $v_2(t)$ of the circuit in Figure 6 if one assumes that the input is continually applied. The point is that the circuit in its ideal (lossless) form is basically unstable and that $v_2(t)$ will continue to increase in magnitude with increasing time such that the output never reaches a "steady-state" level. This is not, however, a problem for the lossy case which is to be discussed in the following section.

4.2 THE NON-IDEAL SINGLE PASS LCS FILTER

The theoretical model of the serial resonance filter has been presented in Figure 6. The analysis of the circuit operation was based on switch S being an ideal element. However, in a practical implementation, this
Figure 12. The parallel LCS circuit.
Figure 13. The basic low-pass filter.
switch would be a semiconductor switching element which would offer a non-zero finite resistance to the circuit when in the "closed" state. This characteristic of switch S must be included in the circuit analysis if the model is to be truly representative of a practical system. Such an analysis may be carried out by assuming the "closed" resistance of switch S is symbolized by R. Under this assumption, equation (25) becomes

\[ V_1 - V_2 = L \frac{di(t)}{dt} + \frac{2}{C} \int_0^t i(t) \, dt + R \cdot i(t) \]  

(35)

Noting that \( i(0) = 0 \), we may again use Laplace analysis to solve for \( i(t) \), \( v_1(t) \), and \( v_2(t) \) as in section 4.1:

\[ \frac{V_1 - V_2}{S} = LsI(s) + 2 \frac{I(s)}{C} + RI(s) \]

Solving for \( I(s) \) gives:

\[ I(s) = \frac{(V_1 - V_2)/L}{S^2 + \frac{R}{L} s + \frac{2}{LC}} \]  

(36)

We may "complete the square" of the quadratic by adding and subtracting \((R/2L)^2\) to the denominator of equation (36) which gives:

\[ I(s) = \frac{(V_1 - V_2)/L}{(S + \frac{R}{2L})^2 + \left[ \frac{2}{LC} - \frac{R}{2L} \right]^2} \]  

(37)

To simplify the form of equation (37) we may let
\[ \alpha = \frac{R}{2L} \]
\[ \omega_0^2 = \frac{2}{LC} \]

and
\[ \omega^2 = \omega_0^2 - \alpha^2 \] (38)
such that
\[ I(s) = \frac{(V_1 - V_2)/L}{(s + \alpha)^2 + \omega^2} \] (39)

Therefore
\[ i(t) = \frac{(V_1 - V_2)}{L} e^{-\alpha t} \sin \omega t. \] (40)

We may solve for \( v_1(t) \) and \( v_2(t) \) as follows:
\[ v_1(t) = V_1 - \frac{1}{C_1} \int_0^t i(t) \, dt \] (41)

The Laplace transform of equation (41) is:
\[ V_1(s) = V_1 - \frac{1}{C_1} \cdot \frac{I(s)}{s} \]

By inserting \( I(s) \) from equation (39), we have:
\[ V_1(s) = V_1 - \frac{1}{C_1 L} \cdot \frac{(V_1 - V_2)}{s \left[ (s + \alpha)^2 + \omega^2 \right]} \]

Therefore
\[ v_1(t) = V_1 - \frac{(V_1 - V_2)}{C_1 L} \left[ \frac{1 - e^{-\alpha t}((\alpha/\omega) \sin \omega t + \cos \omega t)}{\omega^2 + \alpha^2} \right] \] (42)
Equation (42) simplifies to

\[ v_1(t) = V_1 - \frac{1}{2} (V_1 - V_2) \left[ 1 - e^{-at}((\alpha/\omega) \sin \omega t + \cos \omega t) \right] \]  (43)

since \( C_1 = C, \omega_0^2 = \frac{2}{LC} \) and \( \omega_0^2 = \omega^2 + \alpha^2 \).

Similarly

\[ v_2(t) = V_2 + \frac{1}{2} (V_1 - V_2) \left[ 1 - e^{-at}((\alpha/\omega) \sin \omega t + \cos \omega t) \right] \]  (44)

Equations (40), (43) and (44) yield the following values when evaluated at \( t = \pi/\omega \):

\[ i(\pi/\omega) = 0 \]

\[ v_1(\pi/\omega) = \frac{1}{2} (V_1 + V_2) - K \]  (45)

\[ v_2(\pi/\omega) = \frac{1}{2} (V_1 + V_2) + K \]

where

\[ K = \frac{1}{2} (V_1 - V_2) e^{-\alpha \pi/\omega} \]  (46)

and, obviously, the capacitor voltages do not interchange as they do for the ideal case.

Since \( \alpha \) and \( \omega \) must be positive, real numbers, \( K \) will be a positive, real number if we make \( V_1 > V_2 \). Using this relationship between \( V_1 \) and \( V_2 \), we may determine the effect of \( R \) on \( v_1(t) \) and \( v_2(t) \) when the input to the circuit is a discrete Walsh function as follows. Assume the initial capacitor voltages are \( V_1 \) and \( V_2 = 0 \). This assumption
leads to $v_1(\pi/\omega)$ and $v_2(\pi/\omega)$ such that

$$v_1(\pi/\omega) = \left(V_1/2\right) \left(1 - e^{-\frac{\alpha\pi}{\omega}}\right)$$
and

$$v_2(\pi/\omega) = \left(V_1/2\right) \left(1 + e^{-\frac{\alpha\pi}{\omega}}\right)$$

(47)

Under these conditions the top sketch of Figure 14 illustrates how the voltages slowly build up as time increases. Since $K$ varies exponentially with $R$, an increase in $R$ decreases the constant $K$ substantially. This implies that as the series resistance of the circuit approaches infinity the constant $K$ approaches zero. Under this condition, the voltages will not build with time but will remain at a steady state value as long as an input signal at the proper sequency is applied. Upon removal of the input signal the output will dampen out very quickly. Therefore, the effect of the series resistance is to prevent the continual buildup as predicted by the lossless case. This would imply that $K$ is small enough to be considered zero - the sketch at the bottom of Figure 14 illustrates this steady state operation.

In summary, then we see that the practical circuit still performs the desired filtering function and, in fact, has a better response than the lossless model since the output is stabilized by the series resistance.
Figure 14. Sketches illustrating the response of the non-ideal filter for \( v_1(0) = V_1 \) and \( v_2(0) = V_2 = 0 \).
CHAPTER 5

DISCRETE RESONANT FILTERS FOR WALSH FUNCTIONS

Since the input to the LCS filter is a sampled signal and the output is of interest only at certain times, we may discuss the filter as a linear, discrete system by writing linear, difference equations to describe its operation. These equations may then be used to investigate the filter's performance through analytical solution or general purpose simulation.

The purpose of this chapter is to develop the discrete system difference equations and their corresponding models which may be used for simulation and, then, to present the simulation programs. This will allow us to investigate the input/output characteristics of such filters on a software level. The next chapter will present an analytical solution.

5.1 DIFFERENCE EQUATIONS FOR THE DISCRETE RESONANT FILTER

Since the voltages $v_1(t)$ and $v_2(t)$ are of interest only at certain time points $t_k = \left(\frac{k\pi}{\omega}\right) + T/16$ where $k = 0, \pm 1, \pm 2, \ldots$, according to Figures 7 through 9,
difference equations may be used for the analysis of the resonance filter circuit of Figure 6. To illustrate, denote \( v_1(t_k) \) by \( v_1(k) \) and \( v_2(t_k) \) by \( v_2(k) \) with \( S \) always closed and \( S_0 \) always open. According to equation (34) the voltages \( v_1(k) \) and \( v_2(k) \) are related by the following two difference equations of first order:

\[
v_1(k+1) = v_2(k); \quad v_2(k+1) = v_1(k)
\]

or

\[
v_1(k) = v_2(k-1); \quad v_2(k) = v_1(k-1)
\]

If the switch \( S_0 \) is momentarily closed at the time \( t_k \), the applied charge \( q_k = C v_i^1(k) \) increases the voltage across \( C_1 \) by \( v_i^1(k) \). Equations (49) assume the following form:

\[
v_1(k) = v_2(k-1) + v_i^1(k); \quad v_2(k) = v_1(k-1)
\]

Separation of the variables yields two difference equations of second order:

\[
v_1(k) = v_1(k-2) + v_i^1(k)
\]

\[
v_2(k) = v_2(k-2) + v_i^1(k-1)
\]

Hence, we realize from equation (52) that for the special case of \( S \) always closed (which corresponds to having the filter tuned to \( \text{Sal}(4, \theta) \)) the discrete time LCS filter falls into the standard classification of a recursive filter. That is, it is a linear filter which may be realized by a recursive technique wherein the output samples
of the filter are explicitly determined as a weighted sum of past output samples as well as past and/or present input samples. The general equational form of such a filter is

\[
y(k) = b_0 x(k) + b_1 x(k-1) + b_2 x(k-2) + \ldots + b_p x(k-p) + a_1 y(k-1) + a_2 y(k-2) + \ldots + a_m y(k-m)
\]

where \( y(k) \) and \( x(k) \) correspond to the filter's output and input, respectively, and the \( a_j \)'s and \( b_i \)'s are the filter's weighting coefficients. If a filter is time-variant then its coefficients vary with \( k \) and if a filter is time-invariant its coefficients are constants.

Comparing equations (52) and (53) we find that, for the special case of \( S \) always closed, the resonant filter is a linear, time-invariant digital filter where \( a_j = 0 \) for \( j = 1, 3, 4, \ldots, a_2 = 1; b_1 = 0 \) for \( i = 0, 2, 3, \ldots \) and \( b_1 = 1 \). This realization leads to a less than straightforward z-transform analysis which is presented in Chapter 6.

Let us now consider the general case where the switch \( S \) is not always closed but is operated so as to "tune" the filter to resonate at any specified Walsh function. Realizing that the switch \( S \) controls the voltage
interchange between the capacitors \( C_1 \) and \( C_2 \), we may write relationships for \( v_1(k) \) and \( v_2(k) \) which affect this interchange characteristic. The philosophy for writing the interchange relationships is based on the following specification: an interchange of voltage is instigated at \( k \) time if and only if the switch \( S \) was closed between \((k-1)\) and \( k \) times. This specification implies two sets of equations, one for the case when there is an interchange of voltage and one for the case when there is no interchange.

Obviously, the case corresponding to an interchange is represented by equations (50). The case for no voltage interchange is represented by the following equations:

\[
v_1(k) = v_1(k-1) + v_1^i(k); \quad v_2(k) = v_2(k-1)
\]

(54)

We may combine equations (50) and (54) as follows:

\[
v_1(k) = \begin{cases} 
  v_2(k-1) + v_1^i(k) & \text{if } S \text{ was closed between } (k-1) \\
  v_1(k-1) + v_1^i(k) & \text{if } S \text{ was open between } (k-1) 
\end{cases} 
\]

and \( k \) time

(55)

\[
v_2(k) = \begin{cases} 
  v_1(k-1) & \text{if } S \text{ was closed between } (k-1) \text{ and } k \text{ time} \\
  v_2(k-1) & \text{if } S \text{ was open between } (k-1) \text{ and } k \text{ time}
\end{cases}
\]

(56)

Furthermore, to simplify matters somewhat, we may write equations (55) and (56) as:
\[ v_1(k) = S \{v_2(k-1) + v_1^1(k)\} + S' \{v_1(k-1) + v_1^1(k)\} \quad (57) \]

\[ v_2(k) = S \{v_1(k-1)\} + S' \{v_2(k-1)\} \quad (58) \]

where \( S = 1 - S \) and \( S \) is a time variable coefficient (i.e., \( S = S(k) \)) defined as follows:

\[ S = \begin{cases} 1 & \text{if switch } S \text{ was closed between } (k-1) \text{ and } k \text{ time} \\ 0 & \text{if switch } S \text{ was open between } (k-1) \text{ and } k \text{ time} \end{cases} \quad (59) \]

Note that the definition of the switching function \( S \) follows directly from the originally defined operation of switch \( S \) which stated that "\( S \) must be closed for intervals of duration of \( \Delta \theta = 1/8 \) or \( \Delta T = T/8 \) whenever the Walsh function changes sign" in order to cause resonance to occur at a particular Walsh function. Table 3 shows the values of \( S \) for some of the Walsh functions. This table correlates directly with Figure 10 \((k>0)\) and the above definition of \( S \).

Since \( S \) is simply the complement (inversion) of \( S \) we may substitute \( S = 1 - S \) into equations (57) and (58) to yield:

\[ v_1(k) = S \{v_2(k-1) - v_1(k-1)\} + v_1(k-1) + v_1^1(k) \quad (60) \]

\[ v_2(k) = -S \{v_2(k-1) - v_1(k-1)\} + v_2(k-1) \quad (61) \]

These last two equations indicate that, in general, the LCS resonant filter's operation is described by a set
Table 3 Required Relationships Between \( S(k) \) and \( v_1^i(k) \) for the First Eight Walsh functions assuming eight samples/period

<table>
<thead>
<tr>
<th>( v_1^i(k) ) ( S(k) )</th>
<th>( k )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal(0,0) ( S(k) )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Sal(1,0) ( S(k) )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Cal(1,0) ( S(k) )</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>Sal(2,0) ( S(k) )</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Cal(2,0) ( S(k) )</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Sal(3,0) ( S(k) )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Cal(3,0) ( S(k) )</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>Sal(4,0) ( S(k) )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
of two simultaneous linear difference equations with time varying coefficients. Because separation of the variables in such cases generally leads to difference equations of order greater than one for which no general analytical solution is known [74] one may conclude that a more expedient method of solution will be via simulation on a general purpose computer. The following section presents such a simulation.

5.2 SIMULATION OF THE DISCRETE RESONANT FILTER

In order to verify the validity of the generalized equations (60) and (61), one may use these equations to define a discrete model or network to represent the filter and exercise that model via simulation on a general purpose digital computer. This section discusses such a modelling/simulation process.

Prior to a specific discussion of the LCS filter, it seems worth while to make a few general statements concerning network notation. Consider the general, first order, linear, difference equation (from equation 53)

$$y(k) = b_0(k) x(k) + a_1(k) y(k-1)$$  \hspace{1cm} (62)

where, as before, $y$ and $x$ represent the output and input, respectively, of a discrete system and $a_1$ and $b_0$ are, in general, time varying coefficients. One can discuss the procedural process (i.e., algorithm) for calculating $y(k)$
given \(x(k), a(k), \) and \(b_0(k)\) and, also we can discuss a network which is a visualization or picture showing the relationship between operation (62) and the algorithmic process.

An algorithm that may be used in this case, assuming the initial conditions on the variables as well as \(a_1\) and \(b_0\) are known is:

1. Calculate the products \(a_1y(k-1)\) and \(b_0x(k)\),
2. Calculate the sum of the products,
3. Set \(y(k)\) equal to the sum of products.

A convenient picture of equation (62) is shown in Figure 15. This can be related to the computation algorithm as follows: when the new data sample \(x(k)\) appears, the previously computed value of the output \(y(k)\) becomes \(y(k-1)\) for the new iteration. The multiplications and addition are then performed, resulting in the updating of the output to \(y(k)\). We call Figure 15 a digital, or discrete, network. One utilization of the network and algorithm, in combination, leads to a digital computer program and then to a simulation of a given system. Another utilization of the network leads to a digital hardware concept and then to a corresponding hardware simulation. Figure 16 gives a heuristic view of this overall process showing both utilizations. (A discussion of the hardware concept is presented in Chapter 7.)
Figure 15. Discrete, first-order network.
Figure 16. An overall design and analysis concept for a discrete system.
We can now discuss the simulation of the LCS filter by specifying its network representation and then developing a computer simulation program. Such a network for the LCS resonant filter is illustrated in Figure 17 in what is generally referred to as the "direct" form of a network (i.e., the equations are implemented directly without modification). Figure 18 shows the same information except drawn in a parallel form. Both of these models implement equations (60) and (61) directly.

The algorithmic process as suggested by equations (60) and (61) leads to the computer simulation program flow chart of Figure 19. A Fortran IV program version of Figure 19 is listed in Appendix B and a discussion of the most important simulation results is presented in the following section. A more complete presentation of the simulation program and results is given in Appendix B.

5.3 RESULTS OF SIMULATION OF THE DISCRETE FILTER

It is advisable to partition a simulation effort of this type into two phases which are defined according to the class of sampled data signals to be assumed applied to the filter's input. The two phases thus defined are as follows:

Phase I: The input signal is assumed exactly equal to the sampled Walsh function to which the simulated fil-
Figure 17. Direct form of the network for the LCS filter.
Figure 18. Parallel form of the network for the LCS filter.
Input data:
1) Specify filter to be simulated by defining \( s(k), k = 0, 1, 2, \ldots \)
2) Specify \( v_1^k(k), k = 0, 1, 2, \ldots \)

Initial conditions:
\( v_1(-1) = v_2(-1) = 0 \)

Begin simulation:
\( k = 0, k_{\text{max}} = 15 \)

\[
\begin{align*}
v_1(k) &= v_2(k-1) + v_1^k(k) \\
v_2(k) &= v_2(k-1)
\end{align*}
\]

\[
\begin{align*}
v_1(k) &= v_1(k-1) + v_1^k(k) \\
v_2(k) &= v_2(k-1)
\end{align*}
\]

\( k = k_{\text{max}} \)

Print: for \( k = 0, 1, \ldots, k_{\text{max}} \)
\( v_1^k(k) \) \( v_1(k); v_2(k) \)

stop

Figure 19. Flow Chart for Simulation Program.
ter is "tuned" for resonance. This phase will establish the output data which may serve as reference for determining resonance versus anti-resonance for other simulations.

Phase II: The input signal is assumed to be a superposition of sampled Walsh functions, one of which may be equal to the resonating function to which the simulated filter is "tuned". If the resonating signal is present in the input signal, the output should indicate this fact by responding in a manner derived in Phase I. This phase will verify the selectivity of the resonant filter as well as indicate anti-resonant characteristics.

The results of the Phase I simulation effort are presented in Table 4 for the first eight Walsh functions with an assumed sampling rate of eight samples/period and for \(k=0\) through 15. This same data is shown graphically in Figures 20 through 23 for the last four cases of Table 4. It is apparent from the Table 4 data that the filter resonates properly and that the discrete data agrees with the analog data presented in Chapter 4. It is concluded, then, that equations (60) and (61) do serve as describing equations for the LCS resonant filter for Walsh functions.

Partial results of the Phase II simulation effort are presented in Tables 5, 6, and 7 for three different resonant filters: \(s_a(1, \theta), s_a(3, \theta)\), and \(c_a(3, \theta)\). Additional data for this simulation phase are presented
Table 4  Phase I Simulation Results. Output response with the input signal set equal to the resonating signal in each case.

<table>
<thead>
<tr>
<th>$v_1(k)$</th>
<th>k</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>cal(k,0,8)</td>
<td>1/0</td>
<td>2/0</td>
<td>3/0</td>
<td>4/0</td>
<td>5/0</td>
<td>6/0</td>
<td>7/0</td>
<td>8/0</td>
<td>9/0</td>
<td>10/0</td>
<td>11/0</td>
<td>12/0</td>
<td>13/0</td>
<td>14/0</td>
<td>15/0</td>
<td>16/0</td>
<td></td>
</tr>
<tr>
<td>sal(k,1,8)</td>
<td>1/0</td>
<td>2/0</td>
<td>3/0</td>
<td>4/0</td>
<td>5/0</td>
<td>6/0</td>
<td>7/0</td>
<td>8/0</td>
<td>9/0</td>
<td>10/0</td>
<td>11/0</td>
<td>12/0</td>
<td>13/0</td>
<td>14/0</td>
<td>15/0</td>
<td>16/0</td>
<td></td>
</tr>
<tr>
<td>cal(k,1,8)</td>
<td>1/0</td>
<td>2/0</td>
<td>3/0</td>
<td>4/0</td>
<td>5/0</td>
<td>6/0</td>
<td>7/0</td>
<td>8/0</td>
<td>9/0</td>
<td>10/0</td>
<td>11/0</td>
<td>12/0</td>
<td>13/0</td>
<td>14/0</td>
<td>15/0</td>
<td>16/0</td>
<td></td>
</tr>
<tr>
<td>sal(k,2,8)</td>
<td>1/0</td>
<td>2/0</td>
<td>3/0</td>
<td>4/0</td>
<td>5/0</td>
<td>6/0</td>
<td>7/0</td>
<td>8/0</td>
<td>9/0</td>
<td>10/0</td>
<td>11/0</td>
<td>12/0</td>
<td>13/0</td>
<td>14/0</td>
<td>15/0</td>
<td>16/0</td>
<td></td>
</tr>
<tr>
<td>cal(k,2,8)</td>
<td>1/0</td>
<td>2/0</td>
<td>3/0</td>
<td>4/0</td>
<td>5/0</td>
<td>6/0</td>
<td>7/0</td>
<td>8/0</td>
<td>9/0</td>
<td>10/0</td>
<td>11/0</td>
<td>12/0</td>
<td>13/0</td>
<td>14/0</td>
<td>15/0</td>
<td>16/0</td>
<td></td>
</tr>
<tr>
<td>sal(k,3,8)</td>
<td>1/0</td>
<td>2/0</td>
<td>3/0</td>
<td>4/0</td>
<td>5/0</td>
<td>6/0</td>
<td>7/0</td>
<td>8/0</td>
<td>9/0</td>
<td>10/0</td>
<td>11/0</td>
<td>12/0</td>
<td>13/0</td>
<td>14/0</td>
<td>15/0</td>
<td>16/0</td>
<td></td>
</tr>
<tr>
<td>cal(k,3,8)</td>
<td>1/0</td>
<td>2/0</td>
<td>3/0</td>
<td>4/0</td>
<td>5/0</td>
<td>6/0</td>
<td>7/0</td>
<td>8/0</td>
<td>9/0</td>
<td>10/0</td>
<td>11/0</td>
<td>12/0</td>
<td>13/0</td>
<td>14/0</td>
<td>15/0</td>
<td>16/0</td>
<td></td>
</tr>
<tr>
<td>sal(k,4,8)</td>
<td>1/0</td>
<td>2/0</td>
<td>3/0</td>
<td>4/0</td>
<td>5/0</td>
<td>6/0</td>
<td>7/0</td>
<td>8/0</td>
<td>9/0</td>
<td>10/0</td>
<td>11/0</td>
<td>12/0</td>
<td>13/0</td>
<td>14/0</td>
<td>15/0</td>
<td>16/0</td>
<td></td>
</tr>
</tbody>
</table>
Figure 20. Phase I simulation results. Filter tuned to $\text{cal}(k, 2, 8)$ with $v_1^f(k) = \text{cal}(k, 2, 8)$, i.e., the $\text{cal}(2, \theta)$ filter at resonance.
Figure 21. Phase I Simulation results. Filter tuned to \( \text{sal}(k, 3, 8) \) with \( v_1^1(k) = \text{sal}(k, 3, 8) \), i.e., the sal \((3,8)\) at resonance.
Figure 22. Phase I Simulation results. Filter tuned to $\text{cal}(k, 3, 8)$ with $v_1^1(k) = \text{cal}(k, 3, 8)$, i.e., the $\text{cal}(3,8)$ filter at resonance.
Figure 23. Phase I Simulation results. Filter tuned to $\text{sal}(k, 4, 8)$ with $v_1(k) = \text{sal}(k, 4, 8)$, i.e., the $\text{sal}(4, 8)$ filter at resonance.
Table 5  Phase II Simulation Results. Input/output responses for the sal(k,1,8) = sal(1,8) filter. The filter is selective when cal(k,0,8) is not present in the input signal.

| \(v_1(k)/v_2(k)\) | \(k\) | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  |
|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| \(v_1^1(k)\)     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| sal(k,3,8)       |     | 1/0 | -1/0| -1/0| 1/0 | -1/0| 1/0 | -1/0| 1/0 | -1/0| 1/0 | -1/0| 1/0 | -1/0| 1/0 | -1/0|
| sal(k,2,8) +     |     | 2/0 | 2/0 | 0/0 | 0/0 | 0/4 | 0/4 | -2/4| -2/4| 2/-4| 2/-4| 0/-4| 0/-4| 0/8 | 0/8 | -2/8|
| sal(k,1,8)       |     | 2/0 | 2/0 | 2/0 | 2/0 | -2/8| -2/8| -2/8| -2/8| 2/-8| 2/-8| 2/-8| 2/-8| -2/16| -2/16| -2/16|
| 2 sal(k,1,8)     |     | 2/0 | 2/0 | 2/0 | 2/0 | -2/8| -2/8| -2/8| -2/8| 2/-8| 2/-8| 2/-8| 2/-8| -2/16| -2/16| -2/16|
| 2 sal(k,1,8) +   |     | 6/0 | -1/0| -1/0| 5/0 | -5/8| 1/8 | 1/8 | -5/8| 5/-8| 5/-8| -1/-8| 1/-8| -1/-8| 5/-8| -5/16|
| 3 sal(k,3,8)     |     | 2/0 | 2/0 | 2/0 | 2/0 | 0/8 | 0/8 | 0/8 | 2/0 | 2/0 | 2/0 | 2/0 | 2/0 | 2/0 | 2/0 | 2/0 |
| sal(k,1,8) +     |     | 2/0 | 2/0 | 2/0 | 2/0 | 0/8 | 0/8 | 0/8 | 2/0 | 2/0 | 2/0 | 2/0 | 2/0 | 2/0 | 2/0 | 2/0 |
| cal(k,0,8)       |     | 2/0 | 2/0 | 0/0 | 0/0 | 0/4 | 0/4 | 0/4 | 2/4 | 2/4 | 2/4 | 2/4 | 2/4 | 0/4 | 0/4 | 0/4 |
| cal(k,1,8) +     |     | 2/0 | 2/0 | 0/0 | 0/0 | 0/4 | 0/4 | 0/4 | 2/4 | 2/4 | 2/4 | 2/4 | 2/4 | 0/4 | 0/4 | 0/4 |
| cal(k,0,8)       |     | 2/0 | 2/0 | 0/0 | 0/0 | 0/4 | 0/4 | 0/4 | 2/4 | 2/4 | 2/4 | 2/4 | 2/4 | 0/4 | 0/4 | 0/4 |
Table 6 Phase II Simulation Results. Input/output responses for the sal(k,3,8) = sal(3,0) filter. This data agrees with Figures 9 and 11.

<table>
<thead>
<tr>
<th>( v_1(k) )</th>
<th>( k )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{cal}(k,0,8) )</td>
<td>1/0</td>
<td>1/1</td>
<td>1/1</td>
<td>1/2</td>
<td>1/2</td>
<td>1/3</td>
<td>1/3</td>
<td>1/4</td>
<td>1/4</td>
<td>1/5</td>
<td>1/5</td>
<td>1/6</td>
<td>1/6</td>
<td>1/7</td>
<td>1/7</td>
<td>1/8</td>
<td></td>
</tr>
<tr>
<td>( \text{sal}(k,1,8) )</td>
<td>1/0</td>
<td>1/1</td>
<td>1/1</td>
<td>1/2</td>
<td>-1/2</td>
<td>-1/1</td>
<td>-1/1</td>
<td>-1/0</td>
<td>1/0</td>
<td>1/1</td>
<td>1/1</td>
<td>1/2</td>
<td>-1/2</td>
<td>-1/1</td>
<td>-1/1</td>
<td>-1/0</td>
<td></td>
</tr>
<tr>
<td>( \text{cal}(k,1,8) )</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
<td>-1/0</td>
<td>-1/-1</td>
<td>-1/1</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
<td>-1/0</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
<td>-1/0</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
</tr>
<tr>
<td>( \text{sal}(k,2,8) )</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
<td>-1/0</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
<td>-1/0</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
</tr>
<tr>
<td>( \text{cal}(k,2,8) )</td>
<td>1/0</td>
<td>-1/1</td>
<td>-1/1</td>
<td>1/-2</td>
<td>1/2</td>
<td>-1/-1</td>
<td>-1/-1</td>
<td>1/0</td>
<td>1/0</td>
<td>-1/1</td>
<td>-1/1</td>
<td>1/2</td>
<td>-1/-1</td>
<td>-1/-1</td>
<td>1/0</td>
<td>-1/1</td>
<td>-1/1</td>
</tr>
<tr>
<td>( \text{sal}(k,3,8) )</td>
<td>1/0</td>
<td>-1/1</td>
<td>-1/1</td>
<td>1/-2</td>
<td>-1/2</td>
<td>1/-3</td>
<td>-1/-3</td>
<td>-1/4</td>
<td>1/-4</td>
<td>-1/5</td>
<td>-1/5</td>
<td>1/-6</td>
<td>-1/6</td>
<td>1/-7</td>
<td>1/-7</td>
<td>-1/8</td>
<td></td>
</tr>
<tr>
<td>( \text{cal}(k,3,8) )</td>
<td>1/0</td>
<td>-1/1</td>
<td>1/1</td>
<td>-1/0</td>
<td>-1/0</td>
<td>1/-1</td>
<td>-1/-1</td>
<td>1/0</td>
<td>1/1</td>
<td>-1/1</td>
<td>1/0</td>
<td>1/-1</td>
<td>-1/0</td>
<td>1/-1</td>
<td>-1/1</td>
<td>1/0</td>
<td>-1/1</td>
</tr>
<tr>
<td>( \text{sal}(k,4,8) )</td>
<td>1/0</td>
<td>-1/1</td>
<td>1/1</td>
<td>-1/0</td>
<td>1/0</td>
<td>-1/1</td>
<td>1/1</td>
<td>-1/0</td>
<td>1/0</td>
<td>-1/1</td>
<td>1/1</td>
<td>-1/0</td>
<td>1/0</td>
<td>-1/1</td>
<td>1/1</td>
<td>-1/0</td>
<td>1/0</td>
</tr>
<tr>
<td>( \text{sal}(k,1,8) + )</td>
<td>2/0</td>
<td>0/2</td>
<td>0/2</td>
<td>2/0</td>
<td>-2/4</td>
<td>0/-2</td>
<td>0/-2</td>
<td>-2/4</td>
<td>2/-4</td>
<td>0/6</td>
<td>0/6</td>
<td>2/-4</td>
<td>-2/8</td>
<td>0/-6</td>
<td>0/-6</td>
<td>-2/8</td>
<td></td>
</tr>
<tr>
<td>( \text{sal}(k,3,8) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 7 Phase II Simulation Results. Input/output responses for the $\text{cal}(k,3,8) - \text{cal}(3,0)$ filter. The filter is selective when $\text{cal}(k,0,8)$ is not present in the input signal.

<table>
<thead>
<tr>
<th>$v_1(k)$</th>
<th>$v_2(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1'(k)$</td>
<td>$k$</td>
</tr>
<tr>
<td>cal($k,3,8$) + cal($k,0,8$)</td>
<td>2/0</td>
</tr>
<tr>
<td>cal($k,3,8$) + sal($k,1,8$)</td>
<td>2/0</td>
</tr>
<tr>
<td>cal($k,3,8$) + cal($k,1,8$)</td>
<td>2/0</td>
</tr>
<tr>
<td>cal($k,3,8$) + sal($k,2,8$)</td>
<td>2/0</td>
</tr>
<tr>
<td>cal($k,3,8$) + cal($k,2,8$)</td>
<td>2/0</td>
</tr>
<tr>
<td>cal($k,3,8$) + sal($k,3,8$)</td>
<td>2/0</td>
</tr>
<tr>
<td>cal($k,3,8$) + cal($k,3,8$)</td>
<td>2/0</td>
</tr>
<tr>
<td>cal($k,3,8$) + sal($k,4,8$)</td>
<td>2/0</td>
</tr>
</tbody>
</table>
in Appendix B. Detailed analysis of the data for these three cases reveals that proper resonance and anti-resonance characteristics are exhibited for each simulation. Anti-resonance and selectivity are illustrated graphically in Figures 24 and 25 for the sal(k,3,8) filter.

As was noted in Chapter 4 and verified again in this Phase II simulation, if the input signal contains a cal(0, θ) component, the output will continue to "build-up" in amplitude. That is, if an input signal contains both a resonating component and a cal(0, θ) component, the output is controlled by the cal(0, θ) term thus obscuring the presence of the resonating signal. In a practical sense, this is interpreted to mean that if the input signal has a non-zero average, the filter will not exhibit resonance in an oscillatory manner. Thus, for arbitrary input signals with non-zero averages the filter "malfunctions". (These statements are shown to be generally true through the use of mathematical analysis in Chapter 6.)

This requirement that the input signal must have a zero-average leads directly to the conclusion that for these types of filters to be useful in a practical application, we must process the input signal so as to remove its average value. This, in turn, leads to a non-real time requirement for proper implementation. Chapter 7 pre-
Figure 24. Phase II Simulation results. Filter tuned to sal(k, 3, 8) with v_1(k) = sal(k, 4, 8).
$v_1^1(k)$

$\begin{align*}
\text{Phase II Simulation results. Filter tuned to sal} (k, 3, 8) \text{ with } v_1^1(k) &= \text{sal} (k, 1, 8) + \text{sal} (k, 3, 8).
\end{align*}$
sents such an implementation.

Another important point to be made is that there is a definite transient time for these filters during which the output is not equivalent to the inverted Walsh function during resonance. This fact is illustrated by Figure 25 which has a transient that is apparent during the time \( k < 3 \), i.e., during that period \( v_2(k) \) is not equal to the inverse of \( \text{sal}(k,3,8) \). However, for \( k > 3 \), the output at \( k \) time is equal to inversion of \( \text{sal}(k,3,8) \) as it should be.
CHAPTER 6

Z-TRANSFORM ANALYSIS OF RESONANT FILTERS FOR WALSH FUNCTIONS

In Chapter 5, difference equations were derived which described the input/output relationships for resonant filters. These equations were then investigated via computer simulation. The purpose of this chapter is to derive a general solution for the filter equations by utilization of the z-transform method of analysis. The procedure to be used is based strongly on the fact that, in order for a filter to resonate at any specific Walsh function, the "tuning" switch S (and, consequently, the variable coefficients $S(k)$ of equations (60) and (61) must change state in a periodic manner with respect to discrete time $k$. This periodic characteristic of the coefficients of the system difference equations allows one to derive a periodic solution for the output $v_2(k)$ in terms of the input $v_1(k)$.

The analysis begins by considering the special case of $S$ always closed and then proceeds to the general case wherein $S$ is assumed to open and close in a periodic fashion. The two cases are referred to as the time-invariant and time-variant cases, respectively.
Also included in this chapter are detailed examples of input/output calculations using the derived general solution. In each example, the results of the Chapter 5 simulation effort are repeated exactly.

6.1 THE DISCRETE, TIME-INVARIANT RESONANT FILTER

The special case, described in Chapter 5 corresponding to the situation where the switch S in the LCS filter of Figure 6 is always closed, leads to the simple difference relations of equations (51) and (52). Transforming these equations to the Z-domain gives

\[ V_1(z) = z^{-2} V_1(z) + V'_1(z) \]  \hspace{1cm} (63)
\[ V_2(z) = z^{-2} V_2(z) + z^{-1} V'_1(z) \]  \hspace{1cm} (64)

Here \( V'_1(z) \) is the Z-transform of the input to the system which is assumed to be a sampled Walsh function. Therefore, in order to determine \( v_1(k) \) and/or \( v_2(k) \), we must determine the Z-transforms for the family of Walsh functions. This determination is presented in Appendix A.

Using the results of Appendix A, we may illustrate the resonant characteristics for the special case of S always closed as follows. Solving equations (63) and (64) for \( V_1(z) \) and \( V_2(z) \), respectively gives

\[ V_1(z) = \frac{z^2}{(z^2 - 1)} V'_1(z) \]  \hspace{1cm} (65)
\[ V_2(z) = \frac{z}{(z^2 - 1)} V'_1(z) = z^{-1} V_1(z) \]  \hspace{1cm} (66)
According to previous results, one should be able to show that $v_1(k)$ and $v_2(k)$ increase proportional with $k$ if $v_1'(k)$ contains $\text{sal}(k, 4, 8)$ which is the sampled version of $\text{sal}(4, 8)$ as defined in Appendix A and illustrated on line 2 of Figure 7 and, also, in Figure 34. In order to show this resonance mathematically, we may assume that $\text{sal}(k, 4, 8)$ is applied to the filter. To do this in the $Z$-domain, we require the $Z$-transform of $\text{sal}(k, 4, 8)$ which is, from Table 12 and equation 162:

$$\text{sal}(Z, 4, 8) = (1 - Z^{-8})^{-1} (1 - Z^{-1}) (1 + Z^{-2}) (1 + Z^{-4})$$

Therefore, replacing $V_1'(Z)$ in equation (65) with $\text{sal}(Z, 4, 8)$, we have

$$V_1(z) = (1-z^{-1})(1+z^{-2})(1+z^{-4})z^2/(z^2-1)(1-z^{-8})$$

$$= z^3/(z+1)^2(z-1)$$

$$= 1 - z^{-1} + 2z^{-2} - 2z^{-3} + 3z^{-4} - 3z^{-5} + \ldots$$

$$v_1(k) = \delta(k) - \delta(k-1) + 2\delta(k-2) - 2\delta(k-3) + 3\delta(k-4) - \ldots$$

where $\delta(k-j)$ is the unit impulse function occurring at $k = j$. From (66)

$$V_2(z) = z^{-1} V_1(z)$$

$$= z^{-1} - z^{-2} + 2z^{-3} - 2z^{-4} + 3z^{-5} - 3z^{-6} + \ldots$$

$$v_2(k) = \delta(k-1) - \delta(k-2) + 2\delta(k-3) - 2\delta(k-4) + 3\delta(k-5) - \ldots$$
Based on these calculated values of \( v_1(k) \) and \( v_2(k) \) we may conclude that, with \( s(\text{al} (k, 4, 8) \) applied to the input of this specific system, the system output indicates the resonance condition. Table 8 gives calculated values of \( v_2(k) \) as a function of \( v_1'(k) \) for \( v_1'(k) \) equal to the Walsh functions for \( n = 0, 1, 2, 3, 4 \) and \( N = 8 \). It can be seen that the system resonates only if \( s(\text{al} (k, n, N) = s(\text{al} (k, 4, 8) \) is the filter input.

Before concluding this discussion of the time-invariant filter case, an interesting observation should be made which lends credence to such an analysis in terms of practicality: it is a simple matter to show that the general behavior of \( v_1(k) \) and \( v_2(k) \) for a specific input sequency \( n \) and samples per period \( N \) will be unchanged for input sequency equal to \( 2^p n \) and samples per period equal to \( 2^p N \) where \( p \) is a positive integer equal to or greater than zero. This is due to the equality of the Z-transforms of such inputs as illustrated by equations 165 of Appendix A.

### 6.2 THE DISCRETE, TIME-VARIANT RESONANT FILTER

We now consider the general case wherein the switch \( S \) in the LCS filter of Figure 6 is not always closed. The time-varying difference equations for this case are given by equations (60) and (61). Although there is no known general solution for arbitrary \( S(k) \), it is possible to
Table 8. The output of the filter tuned to resonant at Sal(k,4,8) for Walsh functions with n=0,1,2,3 and 4 used as the input signals.

<table>
<thead>
<tr>
<th>$v_1^i(k)$</th>
<th>$v_2(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Cal(k,0,8)</td>
<td>0</td>
</tr>
<tr>
<td>Sal(k,1,8)</td>
<td>0</td>
</tr>
<tr>
<td>Cal(k,1,8)</td>
<td>0</td>
</tr>
<tr>
<td>Sal(k,2,8)</td>
<td>0</td>
</tr>
<tr>
<td>Cal(k,2,8)</td>
<td>0</td>
</tr>
<tr>
<td>Sal(k,3,8)</td>
<td>0</td>
</tr>
<tr>
<td>Cal(k,3,8)</td>
<td>0</td>
</tr>
<tr>
<td>Sal(k,4,8)</td>
<td>0</td>
</tr>
</tbody>
</table>
develop an iterative type solution for the special case when the time-varying coefficients are known to be periodic with respect to \( k \). We will pursue such a periodic solution here since the operation of the switch \( S \) is periodic for any specific resonant filter and, therefore, the variable \( S(k) \) in equations (60) and (61) will be periodic. The fact that \( S(k) \) is periodic is illustrated in Table 3. That is, from Table 3, we note that for \( M = 1, 2, \) or 4,

\[
S(k + M) = S(k)
\]  

(67)

As an example, consider \( S(k) \) for the cal \((k, 3, 8)\) filter. For resonance to occur \( S(k) \) is periodic with a period of \( M = 4 \). Likewise, \( s(k) \) must have a period of 4 for the cal \((k, 1, 8)\) filter. Using the same reasoning, we realize that \( S(k) \) for cal \((k, 0, 8)\) and sal \((k, 4, 8)\) must have a period of \( M = 1 \) and for sal \((k, 2, 8)\) and cal \((k, 2, 8)\) must have a period of \( M = 2 \). The other two functions of Table 3 must have \( M = 4 \).

Another important idea is that for any one of the sampled Walsh functions of Table 3

\[
S(k + 4) = S(k)
\]  

(68)

This fact allows the following discussion to be concerned only with the period \( M = 4 \) and terminate in a general
solution which is valid for $M = 1, 2, \text{ and } 4$ since 1, 2, and 4 are all sub-multiples of 4.

In order to simplify the notation, we may represent equations (60) and (61) as follows:

$$x_k = a_k(y_{k-1} - x_{k-1}) + x_{k-1} + w_k \quad (69)$$

$$y_k = -a_k(y_{k-1} - x_{k-1}) + y_{k-1} \quad (70)$$

where

$$x_k \triangleq v_1(k); \quad a_k \triangleq s(k)$$

$$y_k \triangleq v_2(k); \quad w_k \triangleq v_1'(k)$$

We may further simplify equations (69) and (70) by letting $b_k \triangleq 1 - a_k$, thus,

$$x_k = b_kx_{k-1} + a_ky_{k-1} + w_k \quad (71)$$

$$y_k = a_kx_{k-1} + b_ky_{k-1} \quad (72)$$

It should be noted that, since $a_k$ and $b_k$ are defined as binary variables with value of 0 or 1:

$$a_k b_k = 0; \quad a_k + b_k = 1 \quad (73)$$

We now proceed to solve equations (71) and (72) under the special condition that $a_k$ (and, hence, $b_k$) is a periodic coefficient with period $M$; that is, $a_i = a_{M+i}$ for $i = 0, 1, 2, 3, \ldots, M-1$. In this case, $M = 4$ as indicated by equation (68). Based on the periodicity of $a_k$, we will first form a constant coefficient difference equation that
describes the solution for \( k = n M = 4n, n = 0, 1, 2, \ldots \), and then use this result to find the solution for all other values of \( k \). This procedure, in essence, removes the periodicity of the coefficients from the analysis.

Let \( W_{4n}(z) = w_0 + w_4z^{-4} + w_8z^{-8} + w_{12}z^{-12} + \ldots \) (74)

\[
W_{4n+1}(z) = w_1 + w_5z^{-4} + w_9z^{-8} + w_{13}z^{-12} + \ldots \quad (75)
\]

\[
W_{4n+2}(z) = w_2 + w_6z^{-4} + w_{10}z^{-8} + w_{14}z^{-12} + \ldots \quad (76)
\]

\[
W_{4n+3}(z) = w_3 + w_7z^{-4} + w_{11}z^{-8} + w_{15}z^{-12} + \ldots \quad (77)
\]

With this pattern established for the input \( w_k \), we may now solve equations (71) and (72) simultaneously using the following iterative procedure. Let \( k = 4n \) so that \( a_k = a_0 \) (that is, \( a_k = a_{4n} \) and letting, at the start, \( n = 0 \)). We can write from equation (71):

\[
x_{4n} = b_0x_{4n-1} + a_0y_{4n-1} + w_{4n} \quad (78)
\]

\[
x_{4n+1} = b_1x_{4n} + a_1y_{4n} + w_{4n+1} \quad (79)
\]

\[
x_{4n+2} = b_2x_{4n+1} + a_2y_{4n+1} + w_{4n+2} \quad (80)
\]

\[
x_{4n+3} = b_3x_{4n+2} + a_3y_{4n+2} + w_{4n+3} \quad (81)
\]

\[
x_{4n+4} = b_0x_{4n+3} + a_0y_{4n+3} + w_{4n+4} \quad (82)
\]

Likewise from equation (72), we have
Equations (78) through (87) may be transformed into two constant coefficient difference equations by first "plugging" equations (79) and (84) into (80) and (85), respectively, to yield

\[ x_{4n+2} = (b_2 b_1 + a_2 a_1) x_{4n} + (b_2 a_1 + a_2 b_1) y_{4n} + b_2 w_{4n+1} + w_{4n+2} \]  
\[ y_{4n+2} = (a_2 b_1 + b_2 a_1) x_{4n} + (a_2 a_1 + b_2 b_1) y_{4n} + a_2 w_{4n+1} \]

Secondly, these two equations may be "plugged" into equations (81) and (86) to yield new equations for \( x_{4n+3} \) and \( y_{4n+3} \). This procedure may be continued until the following two equations are derived for \( y_{4n+4} \) and \( x_{4n+4} \):

\[ y_{4n+4} = A x_{4n} + B y_{4n} + C w_{4n+1} + D w_{4n+2} + E w_{4n+3} \]
\[ x_{4n+4} = Bx_{4n} + Ay_{4n} + Fw_{4n+1} + Gw_{4n+2} + Hw_{4n+3} + \omega_{4n+4} \] (91)

where \( A, B, C, D, E, F, G, \) and \( H \) are constants which are functions of \( a_i \) and \( b_i \) for \( i = 0, 1, 2, 3 \). These functions are as follows:

\[ A = a_0b_1b_2b_3 + a_0a_1a_2b_3 + a_0b_1a_2a_3 + a_0a_1b_2b_3 \]

\[ + b_0b_1b_2a_3 + b_0a_1a_2a_3 + b_0b_1a_2b_3 + b_0a_1b_2b_3 \] (92)

\[ B = a_0a_1b_2b_3 + a_0b_1a_2b_3 + a_0a_1a_2a_3 + a_0b_1b_2a_3 \]

\[ + b_0a_1b_2a_3 + b_0b_1a_2a_3 + b_0a_1a_2b_3 + b_0b_1b_2b_3 \] (93)

\[ C = a_0b_2b_3 + a_0a_2a_3 + b_0b_2a_3 + b_0a_2b_3 \] (94)

\[ D = a_0b_3 + b_0a_3 \] (95)

\[ E = a_0 \] (96)

\[ F = b_0b_2b_3 + b_0a_2a_3 + a_0b_2a_3 + a_0a_2b_3 \] (97)

\[ G = b_0b_3 + a_0a_3 \] (98)

\[ H = b_0 \] (99)

By using perfect induction and the definition of \( a_1 \) as a binary variable (i.e.; \( a_1 \) can take on only the values of 0
or 1 in this discussion) and the relationships of equation (73), it can be verified that

\begin{align*}
B &= 1 - A \quad (100) \\
F &= 1 - C \quad (101) \\
G &= 1 - D \quad (102) \\
H &= 1 - E \quad (103)
\end{align*}

The proof of equation (100) is given in Table 9 and for equations (101) and (102) in Table 10. We may now express equations (90) and (91) in a simpler form:

\begin{align*}
y_{4n+4} &= A x_{4n} + (1 - A) y_{4n} + C w_{4n+1} + D w_{4n+2} + B w_{4n+3} \quad (104) \\
x_{4n+4} &= (1 - A) x_{4n} + A y_{4n} + (1 - C) w_{4n+1} + (1 - D) w_{4n+2} \\
&\quad + (1 - E) w_{4n+3} + w_{4n+4} \quad (105)
\end{align*}

These last two equations are actually constant coefficient difference equations involving \( x_{4n} \), \( y_{4n} \) and the various forcing function terms. As such, solutions for \( x_{4n} \) and \( y_{4n} \) may be developed using the Z-transform method. This development is presented in the following section.

6.3 TRANSFORM SOLUTION FOR THE TIME VARIANT FILTER

Prior to the development of the solution of equations (104) and (105), it is advisable to establish the
Table 9. Proof by perfect induction that $B = 1 - A$.

<table>
<thead>
<tr>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_0 b_1 a_2 b_3$</th>
<th>$a_0 a_1 b_2 a_3$</th>
<th>$a_0 a_1 a_2 a_3$</th>
<th>$b_0 b_1 b_2 a_3$</th>
<th>$b_0 b_1 a_2 b_3$</th>
<th>$b_0 a_1 b_2 b_3$</th>
<th>$b_0 a_1 a_2 b_3$</th>
<th>$b_0 a_1 a_2 b_3$</th>
<th>$A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$a_0 b_1 a_2 b_3$</th>
<th>$a_0 a_1 b_2 a_3$</th>
<th>$a_0 a_1 a_2 a_3$</th>
<th>$b_0 b_1 b_2 a_3$</th>
<th>$b_0 b_1 a_2 b_3$</th>
<th>$b_0 a_1 b_2 b_3$</th>
<th>$b_0 a_1 a_2 b_3$</th>
<th>$b_0 a_1 a_2 b_3$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

100
Table 10  Proof by Perfect Induction that
(a)G=1-D and (b)F=1-C.

(a)

\begin{tabular}{cccccccc}
\text{a}_0 & \text{a}_2 & \text{a}_3 & \text{b}_0\text{a}_3 & \text{D} & \text{b}_0\text{b}_3 & \text{a}_0\text{a}_3 & \text{G} \\
\hline
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{tabular}

(b)

\begin{tabular}{cccccccccccc}
\text{a}_0 & \text{a}_2 & \text{a}_3 & \text{a}_0\text{b}_2\text{b}_3 & \text{a}_0\text{a}_2\text{a}_3 & \text{a}_0\text{a}_2\text{a}_3 & \text{b}_0\text{b}_2\text{a}_3 & \text{b}_0\text{b}_2\text{b}_3 & \text{C} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
\end{tabular}

\begin{tabular}{cccccccc}
\text{b}_0\text{b}_2\text{b}_3 & \text{b}_0\text{a}_2\text{a}_3 & \text{a}_0\text{b}_2\text{a}_3 & \text{a}_0\text{a}_2\text{b}_3 & \text{F} \\
\hline
1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 \\
\end{tabular}
initial conditions on the variables $x$ and $y$ as well as the generalized $Z$-transforms of the functions $x_{4n}$, $y_{4n}$, $x_{4n+4}$, and $y_{4n+4}$.

The initial conditions may be established from equations (69) and (70) as follows:

$$x_0 = a_0(y_{-1} - x_{-1}) + x_{-1} + w_0$$

$$= a_0(0 - 0) + 0 + w_0$$

$$= w_0$$

(106)

$$y_0 = -a_0(y_{-1} - x_{-1}) + y_{-1}$$

$$= -a_0(0 - 0) + 0$$

$$= 0$$

(107)

Therefore, the initial value of $x$, i.e., $x_0$, will always be equal to the initial value of the input, $w_0$, and the initial value of $y$ will always be zero.

The required transforms of $x_{4n}$, $y_{4n}$, $x_{4n+4}$, and $y_{4n+4}$ may be established by considering a general discrete function $f_k$ and its $Z$-transform $F_k(z)$ in the following manner:

$$F_k(z) = \sum_{k=0}^{\infty} f_k z^{-k}$$

$$= f_0 + f_1 z^{-1} + f_2 z^{-2} + f_3 z^{-3} + \ldots$$
If \( k = 4n, \ n = 0, 1, 2, 3, \ldots \), we have

\[
F_{4n}(z) = \sum_{n=0}^{\infty} f_{4n} z^{-4n} \tag{108}
\]

\[
= f_0 + f_4 z^{-4} + f_8 z^{-8} + f_{12} z^{-12} + \ldots .
\]

Furthermore, we may write that

\[
F_{4n+4}(z) = \sum_{n=0}^{\infty} f_{4n+4} z^{-4n} \tag{109}
\]

\[
= f_4 + f_8 z^{-4} + f_{12} z^{-8} + f_{16} z^{-16} + \ldots .
\]

\[
= z^4 [f_4 z^{-4} + f_8 z^{-8} + f_{12} z^{-12} + f_{16} z^{-16} + \ldots .]
\]

\[
= z^4 [-f_0 + f_4 z^{-4} + f_8 z^{-8} + f_{12} z^{-12} + \ldots .]
\]

\[
= z^4 [F_{4n}(z) - f_0] \tag{110}
\]

where in the third and fourth steps above the right side has \( z^4 \) factored out and \( \pm f_0 \) added to it, respectively.

Utilizing the results of equations (108) through (110), we readily obtain the transforms of equations (104) and (105):

\[
z^4 [Y_{4n}(z) - y_0] = AX_{4n}(z) + (1-A)Y_{4n}(z) + CW_{4n+1}(z)
\]

\[
+DW_{4n+2}(z) + EW_{4n+3}(z) \tag{111}
\]

\[
z^4 [X_{4n}(z) - x_0] = (1-A)X_{4n}(z) + AY_{4n}(z) + (1-C)W_{4n+1}(z)
\]

\[
+(1-D)W_{4n+2}(z) + (1-E)W_{4n+3}(z) + z^4 [W_{4n}(z) - w_0] \tag{112}
\]
It will be recalled that $W_{4n+i}(z)$ for $i = 0, 1, 2, 3$ are assumed to be the transforms of the components of a known input forcing function and have been specified in equations (74) through (77).

To complete the solution process one may now solve equations (111) and (112) simultaneously to yield the following general equations for $Y_{4n}(z)$ and $X_{4n}(z)$:

$$Y_{4n}(z) = \frac{1}{[z^4-(1-A)]^2-A^2} \left\{ Az^4 W_{4n}(z) + [(z^4-1)C+A] W_{4n+1}(z) 
+ [(z^4-1)D+A] W_{4n+2}(z) + [(z^4-1)B+A] W_{4n+3}(z) 
+ y_0 z^4 [(z^4-1)+A] + Ax_0 z^4 - Aw_0 z^4 \right\} \quad (113)$$

This last expression may be simplified by using the initial conditions from equations (106) and (107) and by renaming the constants such that

$$K_0 \triangleq A \quad (114)$$
$$K_1 \triangleq C \quad (115)$$
$$K_2 \triangleq D \quad (116)$$
$$K_3 \triangleq E \quad (117)$$

The simplified form of equation (113) becomes:

$$Y_{4n}(z) = \frac{3y_0 [(z^4-1)K_1 + K_0] W_{4n+1}(z)}{[z^4-(1-K_0)]^2-K_0^2} \quad (118)$$
Solving for $X_{4n}(z)$ we have:

$$X_{4n}(z) = \frac{1}{z^4-(1-K_0)} \left\{ AY_{4n}(z) + (1-C)W_{4n+1}(z) \right. $$

$$+(1-D)W_{4n+2}(z) + (1-E)W_{4n+3}(z)$$

$$+z^4W_{4n}(z) + x_0z^4 - w_0z^4 \right\} \quad (119)$$

or

$$X_{4n}(z) = \frac{1}{z^4-(1-K_0)} \left\{ \frac{K_0^3}{1-K_0} [(z^4-1)K_1 + K_0]W_{4n+1}(z) \right. $$

$$+\frac{3}{1-K_1}W_{4n+1}(z) + z^4W_{4n}(z) \right\} \quad (120)$$

Finally, we have arrived at general solutions for $X_{4n}(z)$ and $Y_{4n}(z)$ as functions of the input components $W_{4n+i}$ for $i = 0, 1, 2, 3$. The next section presents examples of the application of these results to a complete solution of the problem under consideration. That is, we may now consider the time domain solution for $y_{4n}$ and $x_{4n}$ under a specified input, $w_{4n}$. Also, we may then use equations (78) through (87) to determine the solutions for all $k \geq 0$.

6.4 TIME DOMAIN SOLUTION FOR THE TIME VARIANT FILTER

The general procedure to be used in this section is defined as follows:

**Step 1.**

Assume that a resonant filter for a specific Walsh function is to be considered. (This assumption will
dictate the values of $a_i$ and, hence, $K_i$ for $i = 0, 1, 2, 3$.) The required values for $K_i$ will be inserted in the equations for $Y_{4n}(z)$ and $X_{4n}(z)$ from the previous section.

**Step 2.**
Assume a specific input $w_k$ and, hence, $W_{4n}$, from which $W_{4n+i}(z)$ for $i = 0, 1, 2, 3$ may be determined. The functions $W_{4n+i}(z)$ will be inserted into the equations for $Y_{4n}(z)$ and $X_{4n}(z)$ arrived at in Step 1 above.

**Step 3.**
The inverse Z-transform will be determined for $Y_{4n}(z)$ and $X_{4n}(z)$, i.e., $y_{4n}$ and $x_{4n}$ will be calculated.

**Step 4.**
The values of $y_{4n}$ and $x_{4n}$ will be used in conjunction with equations (78) through (87) to determine a complete solution for all $k \geq 0$.

As a first example, let us assume that a filter for the sampled version of the Walsh function $S_{al}(3,0)$ being sampled eight times per period, i.e., a filter for $S_{al}(k, 3, 8)$, is to be analyzed. From Table 3 we find that

\[ a_0 = a_1 = a_3 = 1 \]

\[ a_2 = 0 \]
and, hence

\[ K_0 = K_3 = 1 \]
\[ K_1 = K_2 = 0 \]

These values inserted into equations (118) and (120) give

\[ Y_{4n}(z) = \frac{1}{(z^8-1)} \left\{ z^4W_{4n}(z) + W_{4n+1}(z) + W_{4n+2}(z) + z^4W_{4n+3} \right\} \]

or

\[ Y_{4n}(z) = \frac{z^4}{(z^8-1)} \left\{ W_{4n}(z) + z^{-4}W_{4n+1}(z) + z^{-4}W_{4n+2} + z^4W_{4n+3} \right\} \]

Equation (121) gives

\[ Y_{4n}(z) = \frac{1}{z^4} \left\{ \frac{z^4W_{4n}(z) + W_{4n+1}(z) + W_{4n+2}(z) + z^4W_{4n+3}}{(z^8-1)} \right\} \]

\[ + W_{4n+1}(z) + W_{4n+2}(z) + z^4W_{4n}(z) \}

or

\[ Y_{4n}(z) = \frac{z^4}{(z^8-1)} \left\{ z^4W_{4n}(z) + W_{4n+1}(z) + W_{4n+2}(z) \right\} \]

Equations (121) and (122) will yield the output \( y_{4n} \) of the Sal \((k, 3, 8)\) filter if the input \( w_k \) is known. Let us assume that the input is the Sal \((k, 3, 8)\) function and continue the analysis and check for proper resonance. The
results may be checked against results from the computer simulation of Chapter 5.

From Table 3 one finds that for a single complete period of $\text{Sal}(k, 3, 8)$ we have

$$\text{Sal}(k, 3, 8) = W_k = \{1, -1, -1, 1, -1, 1, 1, -1\} \quad (123)$$

Therefore, from equations (74) through (77)

$$W_{4n}(z) = w_0 + w_4 z^{-4} + w_8 z^{-8} + w_{12} z^{-12} + \ldots \ldots$$

$$= 1 - z^{-4} + z^{-8} - z^{-12} + \ldots \ldots = z^4/(z^4+1) \quad (124)$$

$$W_{4n+1}(z) = w_1 + w_5 z^{-4} + w_9 z^{-8} + w_{13} z^{-12} + \ldots \ldots$$

$$= -1 + z^{-4} - z^{-8} + z^{-12} + \ldots \ldots = -z^4/(z^4+1) \quad (125)$$

$$W_{4n+2}(z) = w_2 + w_6 z^{-4} + w_{10} z^{-8} + w_{14} z^{-12} + \ldots \ldots$$

$$= -1 + z^{-4} - z^{-8} + z^{-12} - z^{-16} + \ldots \ldots = -z^4/(z^4+1) \quad (126)$$

$$W_{4n+3}(z) = w_3 + w_7 z^{-4} + w_{11} z^{-8} + w_{15} z^{-12} + \ldots \ldots$$

$$= 1 - z^{-4} + z^{-8} - z^{-12} + z^{-16} + \ldots \ldots = z^4/(z^4+1) \quad (127)$$

Inserting equations (124) through (127) into equation (121) and (122) we have
\[ Y_{4n}(z) = \frac{z^4}{(z^8-1)} \left\{ \frac{z^4}{z^{4}+1} - \frac{1}{z^{4}-1} - \frac{1}{z^{4}+1} + \frac{z^4}{z^{4}-1} \right\} \]

\[ = \frac{2z^4}{(z^{4}+1)^2} \] \hspace{1cm} (128)

\[ = 2z^4(z^{-8} - 2z^{-12} + 3z^{-16} - 4z^{-20} + 5z^{-24} - \ldots) \]

\[ = 2z^4 - 4z^8 + 6z^{12} - 8z^{16} + 10z^{20} - \ldots \] \hspace{1cm} (129)

Also,

\[ X_{4n}(z) = \frac{z^4}{(z^8-1)} \left\{ \frac{z^8}{z^{4}+1} - \frac{z^4}{z^{4}+1} + \frac{z^4}{z^{4}+1} \right\} \]

\[ = \frac{z^4(z^4-1)}{(z^4+1)^2} \] \hspace{1cm} (130)

\[ = 1 - 3z^{-4} + 5z^{-8} - 7z^{-12} + 9z^{-16} - 11z^{-20} + \ldots \] \hspace{1cm} (131)

Finally, the inverse transforms of equations (129) and (131) are

\[ Y_{4n} = 2\delta(4n-4) - 4\delta(4n-8) + 6\delta(4n-12) \]

\[ - 8\delta(4n-16) + \ldots \ldots \] \hspace{1cm} (132)

\[ = -\frac{4n}{\sqrt{2}} \cos \left(\frac{(4n-1)\pi}{4}\right) \quad n = 0,1,2,3,\ldots \] \hspace{1cm} (133)

\[ X_{4n} = \delta(4n) - 3\delta(4n-4) + 5\delta(4n-8) - 7\delta(4n-12) \]

\[ + 9\delta(4n-16) + \ldots \ldots \] \hspace{1cm} (134)
\begin{align*}
= \frac{4n+2}{\sqrt{2}} \cos \frac{(4n-1)n}{4} \quad n = 0, 1, 2, 3, \ldots \tag{135}
\end{align*}

The closed form for \( y_{4n} \) and \( x_{4n} \) presented in the equations (133) and (135) may be arrived at by first using de Moivre's Theorem to show that the fourth roots of \(-1\) are

\[- \left( \frac{\sqrt{2}}{2} + j \frac{\sqrt{2}}{2} \right) + \left( \frac{\sqrt{2}}{2} - j \frac{\sqrt{2}}{2} \right)\]

such that the denominator of equations (128) and (130) may be written as follows, with \( a = \sqrt{2}/2 \):

\begin{align*}
(z^4+1)^2 &= [z^+(a+ja)]^2 [z^-(a-ja)]^2 \\
&= [z-(a+ja)]^2 [z+(a+ja)]^2 \\
&= [z-(a-ja)]^2 [z+(a-ja)]^2 \tag{136}
\end{align*}

Next, by using Heaviside's Expansion Theorem for multiple complex roots and Euler's Formulas for sine and cosines, it's a rather laborious, yet straightforward, mathematical procedure to develop the closed form relationships for \( y_{4n} \) and \( x_{4n} \).

Another somewhat heuristic technique for deriving the closed form is to consider equations (132) and (134) as series of numbers whose values are represented by the coefficients of the delta functions and attempt to determine the \( k^{th} \) term of the series. This procedure will usually be fruitful in dealing with series such as those
derived for resonant filter outputs during the resonant condition since such outputs are always oscillatory and increasing in magnitude.

The conclusion of this example is arrived at by using equations (133) and (135) in conjunction with equations (79) through (81) and (84) through (86) and the following relationships for $w_k = \text{Sal}(k, 3, 8)$ where $k = 4n$:

$$w_{4n} = w_{4n+3} = -w_{4n+1} = -w_{4n+2} = (-1)^{\frac{4n}{4}} \quad (137)$$

The final solution is presented below for $y_k$ only.

$$y_{4n} = -\frac{4n}{\sqrt{2}} \cos \left(\frac{4n-1}{4}\right) \pi \quad (138)$$

$$y_{4n+1} = \frac{4n+2}{\sqrt{2}} \cos \left(\frac{4n-1}{4}\right) \pi \quad (139)$$

$$y_{4n+2} = \frac{4n+2}{\sqrt{2}} \cos \left(\frac{4n-1}{4}\right) \pi \quad (140)$$

$$y_{4n+3} = -\frac{4n}{\sqrt{2}} \cos \left(\frac{4n-1}{4}\right) \pi - 2(-1)^{\frac{4n}{4}} \quad (141)$$

It can be noted that these results agree exactly with the simulation results of Chapter 5.

As a second example, consider the output of the $\text{Sal}(k, 1, 8)$ filter whose input is assumed to be $\text{Sal}(k, 1, 8)$. Using the same procedure as before, it can be shown that
\[ y_{4n} = 4\delta(4n-4) - 4\delta(4n-8) + 8\delta(4n-12) - 8\delta(4n-16) + 12\delta(4n-20) + \ldots \] (142)

and

\[ x_{4n} = 6(4n) - 6(4n-4) + 36(4n-8) - 36(4n-12) + 96(4n-16) - 96(4n-20) + \ldots \] (143)

In closed form

\[ y_{4n} = 1 - (-1)^{\frac{4n}{4}} - \frac{4n}{\sqrt{2}} \cos \left(\frac{(4n-1)\pi}{4}\right) \] (144)

and

\[ x_{4n} = 1 - (-1)^{\frac{4n}{4}} + \frac{4n+2}{\sqrt{2}} \cos \left(\frac{(4n-1)\pi}{4}\right) \] (145)

To complete this second example we may write the following equations which describe \( y_k \) for all \( k \geq 0 \). In this case

\[ w_{4n} = w_{4n+1} = w_{4n+2} = w_{4n+3} = (-1)^{\frac{4n}{4}} \] (146)

and from equations (79) through (81) and (84) through (86) we have

\[ y_{4n} = 1 - (-1)^{\frac{4n}{4}} - \frac{4n}{\sqrt{2}} \cos \left(\frac{(4n-1)\pi}{4}\right) \] (147)

\[ y_{4n+1} = y_{4n+2} = y_{4n+3} = y_{4n} = y_k, \quad k = 0, 1, 2, \ldots \] (148)
Again it is noted that these results agree exactly with the simulation data developed in Chapter 5 for the Sal (k, 1, 8) filter in resonance.

As a third example let us evaluate the anti-resonance response of the Sal (k, 1, 8) filter whose input is assumed to be

\[ w_k = \text{Sal} (k, 3, 8) \]

From the general equations for \( Y_{4n}(z) \) and \( X_{4n}(z) \), i.e., equations (118) and (119), for the Sal (k, 1, 8) filter, we have

\[ Y_{4n}(z) = \frac{z^4}{(z^6-1)} \sum_{i=0}^{3} W_{4n+i}(z) \]  
(149)

and

\[ X_{4n}(z) = z^{-4} Y_{4n}(z) + W_{4n}(z) \]  
(150)

where, from our first example and equations (124) through (127), we find that

\[ \sum_{i=0}^{3} W_{4n+i}(z) = 0 \]  
(151)

Hence

\[ Y_{4n}(z) = 0 \]  
(152)

and

\[ X_{4n}(z) = W_{4n}(z) \]  
(153)
Therefore, we may conclude that the filter will not resonate in response to this input function. Again, the agreement between the Chapter 5 simulation data and these results is noted.

As a concluding observation for this chapter and in support of the remarks of Chapter 5 concerning the selectivity of the LCS resonant filter in cases where the input signal has a non-zero average value, we consider the following ideas.

The general equation derived for the LCS filter is that given in equation (118) which shows that, regardless of the value of \( n \) or \( K_i \), \( Y(a) \) will always have z-plane poles dictated by the term in the denominator of equation (118), i.e., by

\[
[z^4 - (1-K_0)]^2 - K_0^2 = 0 \tag{154}
\]

Also, since \( K_0 \) may take on only values of 0 or 1, equation (154) can only take on one of the following forms:

\[
(z^4-1)^2 = 0 \tag{155}
\]

\[
(z^8-1) = 0 \tag{156}
\]

Either of these forms will always result in a pole at \( z = 1 \). Such a pole will cause \( Y(z) \) to equal infinity if the input signal has a non-zero average [76]. In the LCS
filter, this situation causes the filter to become non-selective to the resonating component of an input signal.

One procedure that may be used to eliminate this detrimental effect is to "subtract" the average value of the input signal before applying the signal to a resonant filter. This procedure, in turn, requires non-real time operation and is utilized in the following chapter to implement a digital version of the resonant filter for Walsh functions.
CHAPTER 7

A SPECIAL PURPOSE COMPUTER FOR
DIGITAL RESONANT FILTERS

Previous chapters in this work have dealt with the theory of operation of resonant filters for Walsh functions and have pointed out the relationship between the continuous time and discrete time signal versions of such filters. Equations (60) and (61) have been used to simulate the resonant filter thus allowing the realization of these filters as digital filters via a general purpose computer. In this chapter we treat the problem of realizing these discrete time resonant filters via a special purpose computer wherein the architecture of the special purpose hardware executes a program dictated by equations (60) and (61) along with the practical limitations derived from Chapters 5 and 6.

Figure 26 is a conceptual block diagram showing the functional requirements of a generalized digital filtering system. That is, the input to the system is a continuous signal $e_i(t)$ and the output is a continuous signal $e_o(t)$. The functional operation of such a system is apparently to convert the continuous input to binary digital form, filter
Figure 26. Conceptual Diagram of a digital filtering system.
out the undesirable frequency (or sequency, in our case) components by digitally processing the binary representation, and then re-converting the signal to continuous form. The analog-to-digital (A/D) and digital-to-analog (D/A) converters are the devices which perform the necessary conversions and serve as the interfaces between the digital filtering process and the outside analog world. There are standard techniques and hardware available to perform the A/D and D/A functions and, therefore, the following discussion will only be concerned with the filtering process. With respect to Figure 26, this means that we will discuss the "digital filter" section whose input and output are \( v_1(k) \) and \( v_2(k) \), respectively, which have the same meaning as in previous chapters. It should be noted that \( v_1(k) \) and \( v_2(k) \) are now assumed to be binarily coded versions of the input and output.

Another extremely important point to recall with respect to the resonant Walsh function filter is that it is a single sequency filter whose output in the resonant mode tends to oscillate in an unbounded manner. To eliminate the necessity of having to specify large register bit sizes, the following implementation will require that the filter output simply "indicate" (in a yes/no, i.e., binary manner) whether the filter is in a resonant mode.
Another important idea to note is that only conceptual models will be presented as opposed to detailed logic design of the resonant filter. The models will be clear enough so that detailed logic design would lend no new concepts and would be dependent on device technology. The models presented will be directly related to the equations and models developed in previous chapters which have been shown to be operationally valid through simulation.

7.1 ORGANIZATION AND OPERATION OF A SPECIAL PURPOSE COMPUTER

The functional organization of a resonant digital filter for Walsh functions is shown in Figure 27 and the operational sequence of such a system is shown in flow chart form by Figure 28. These two figures are related by the following system operational description.

There are basically six control states that the filtering must accomplish, namely

1. Initialization
2. Input Data
   a. Real Time Sampling of Input Signal
   b. Memorization of Input Data for one period
   c. Accumulation (i.e., addition) of Input Data for one period
3. Calculation of the Average Value
Figure 27. Functional organization of a resonant digital filter for Walsh functions.
Figure 28. Functional operation of a resonant digital filter for Walsh functions.
4. Removal of the Average Value

5. Filtering of the Input Data minus the Average Value (through implementation of Equations (60) and (61)).
   a. Transient Determination (first filtering cycle)
   b. Resonance Determination (second filtering cycle)

6. Indication of the Resonant (or anti-resonant) Mode at the end of the second cycle

7.2 IMPLEMENTATION OF A SPECIAL PURPOSE COMPUTER

A hardware implementation based upon the operational description of Section 7.1 appears in Figures 29, 30, 31, and 32. Figure 29 shows the generalized form of the input data memory buffer, the input accumulator and the input average remover. Figures 30 and 31 represent two different implementations of the actual resonant digital filtering process; one based on the "direct" implementation of Figure 17 and one based on the "parallel" implementation of Figure 18. Finally, Figure 32 shows an implementation of the resonance detector output.

The function of the resonance detector is performed on the second cycle of the filtering process by detecting the signs (plus or minus) of the $v_2(k)$ signal and comparing these signs to the Walsh function to which
Figure 29. The input section of a digital Walsh function filter.
Figure 30. The filtering section of a Walsh function filter based on the direct method of Figure 17. Note that $v_1(k)$ here is actually the "delayed" $v_1(k)$ with average value removed.
Figure 31. The filtering section of a Walsh function filter based on the parallel method of Figure 18. Note that $v_1(k)$ and the tuning register are the same as for Figure 30.
Figure 32. The output section of a Walsh function filter.

Note: \( \text{Walsh function register } = v_2(k) \text{ sign register } \): 1 + output
the filter is tuned under the following mapping

<table>
<thead>
<tr>
<th>Sign</th>
<th>Continuous Walsh Function</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>+</td>
<td>+1</td>
<td>0</td>
</tr>
</tbody>
</table>

Under these conditions, the resonance mode relationship that would exist between the Walsh function register contents and the S(k) tuning register is shown in Table 11 for an assumed eight samples per period sampling rate. It should be noted that Table 11 is equivalent to Table 3 under the above mentioned mapping.

In summary we see that the implementation of these types of filters follows directly from their theoretical analysis presented in previous chapters. Each of the components are either readily available in integrated circuit form or can easily be built up from basic integrated circuitry presently available on the semiconductor market. It should also be pointed out that the generalized system presented here can in principle be used to implement filtering for Walsh functions of any order.
Table 11. Table 3 under the mapping $-1 \rightarrow 1$ and $+1 \rightarrow 0$ for the Walsh function

<table>
<thead>
<tr>
<th>$v_i^j (k)$</th>
<th>$s(k)$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s(k)$</td>
<td>$s(k)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$cal(0, \theta)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$s(k)$</td>
<td>$s(k)$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$cal(1, \theta)$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s(k)$</td>
<td>$s(k)$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$cal(2, \theta)$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$s(k)$</td>
<td>$s(k)$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$cal(3, \theta)$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s(k)$</td>
<td>$s(k)$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$cal(4, \theta)$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$s(k)$</td>
<td>$s(k)$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
CHAPTER 8

CONCLUSIONS

One of the motivations for this work has been the need to extend the sequency filtering concept into the method of resonant filtering as opposed to matched filtering which has consumed most of the previous research in the Walsh function arena. It would seem that the mathematical, simulation, and implementation techniques used herein most certainly serve to establish the resonant filter as a viable method of selectively recognizing the presence of Walsh functions when they appear as integral parts of arbitrary functions. More specifically, one of the primary conclusions that can be made concerning the results of this paper is that the concepts of resonant sequency filtering are amenable to investigation through LaPlace Transform theory (for the analog filter) and Z-transform theory (for the discrete, or digital, filter). This conclusion is important in that it lends support to possible use of the classical techniques in other filtering procedures such as sequency bandpass, sequency low pass, and sequency high pass filtering.
With respect to the LCS filter we can make several fundamental conclusions:

The LaPlace transform analysis serves to establish the theoretical and practical concepts of a resonant sequency filter and corresponds favorably with procedures associated with sinusoidal function theory in the frequency domain.

When visualized as a linear, discrete system the LCS resonant filter may be described by linear, difference equations with periodic, time-varying coefficients. Once derived, these equations are seen to fit into the standard form for a recursive digital filter and can be solved by using the Z-transform theory along with special procedures based upon the periodicity of the coefficients.

Real time resonant sequency filtering of arbitrary time functions is possible only if the average value of the function is zero in magnitude.

Once the describing equations for such filters are derived, their general purpose simulation is straightforward. Also, the filters are easily implemented in digital form using standard hardware elements.

For reasons of simplicity, the examples used in this work assumed an input sampling rate of eight samples.
per period but the development procedures are general enough that they can be directly extended to any other sampling rate. The key factor necessary for such an extension would be that the operation of these filters is based on the periodicity of the Walsh functions and, correspondingly, the required periodicity of the "tuning" switches which cause an individual filter to resonate.
APPENDIX A

THE Z-TRANSFORMS OF THE WALSH FUNCTIONS

The purpose of this section is to present the development of the one-sided \( Z \)-transform for the family of discrete Walsh functions. This development will be strongly based on the periodic characteristics of the continuous Walsh functions and will result in a closed form \( Z \)-transform for each Walsh function in terms of its sequency \( n \) and the number of samples per period \( N \).

The discrete Walsh functions are obtained by sampling the continuous Walsh functions as illustrated in Figure 33 for the cases of \( N = 2, 4, \) and 8 applied to the continuous function \( \text{Sal}(1, 0) \). Figure 34 illustrates the sampling process for \( N = 8 \) applied to the Walsh functions with sequency \( n = 0, 1, 2, 3, 4 \). Also, the symbolic representation to be used for the discrete Walsh functions is shown in Figures 33 and 34 by the use of the three arguments \( k, n, \) and \( N \) where \( k \) is an independent, discrete variable which may take on \( N \) values of \( 0, 1, 2, \ldots, (N-1) \).
Figure 33. The discrete Walsh functions $s(a)(k, 1, N)$ for $N = 2, 4,$ and 8.
Figure 34. The sampling process for $N = 8$ applied to the Walsh functions with sequence $n = 0, 1, 2, \ldots, 7$. 
The one-sided Z-transform of a sampled function \( f(k\tau) \), where \( \tau \) is the sample size (i.e., time between samples), is defined [77] as:

\[
F(Z) = \sum_{k=0}^{\infty} f(k\tau)Z^{-k}
\] (157)

Using this definition, we may determine the Z-transform of the Walsh functions directly by replacing \( f(k\tau) \) in equation (157) with a particular discrete Walsh function. As an example, consider the Z-transform of the discrete function sal \((k, 1, 2)\):

\[
F(Z) = \sum_{k=0}^{\infty} \text{sal} \,(k, 1, 2)Z^{-k}
\]

where sal \((k, 1, 2)\) may be written in closed form as

\[
\text{sal} \,(k, 1, 2) = (-1)^k
\] (158)

so that

\[
F(Z) = \sum_{k=0}^{\infty} (-1)^kZ^{-k} = 1 - Z^{-1} + Z^{-2} - Z^{-3} + Z^{-4} - ... \]

\[
= (1 - Z^{-1}) \left(1 + Z^{-2} + Z^{-4} + Z^{-8} + ...\right)
\]

\[
= (1 - Z^{-1}) \left(1/(1 - Z^{-2})\right), \quad Z > 1
\]

\[
= Z/(Z + 1) \quad , \quad Z > 1
\] (159)

Unfortunately, the determination of the Z-transforms for the Walsh functions using the direct method requires
us to write each of the discrete functions in a closed form (similar to equation (158) for sal (k, 1, 2)) which is not always as simple as in the previous example. Consequently, we may use the periodic characteristics of the Walsh functions to simplify matters somewhat. Realizing that each discrete Walsh function is periodic (i.e., 
\( f(k \tau + N \tau) = f(k \tau) \), where \( N \tau = T \) (time of one period), we may assume that \( \tau \) is unity and expand equation (157) as follows:

\[
F(Z) = \sum_{k=0}^{\infty} f(k)Z^{-k}
\]

\[
F(Z) = \sum_{k=0}^{N-1} f(k)Z^{-k} + \sum_{k=N}^{2N-1} f(k)Z^{-k} + \ldots
\]

\[
F(Z) = (1+Z^{-N} + Z^{-2N} + Z^{-3N} + \ldots) \sum_{k=0}^{N-1} f(k)Z^{-k}
\]

Using the binomial theorem to identify the infinite series, we have

\[
F(Z) = (1-Z^{-N})^{-1} \sum_{k=0}^{N-1} f(k)Z^{-k}
\]  

(160)

where the finite summation in equation (160) represents the Z-transform of \( f(k) \) if \( f(k) \) existed only from 0 to \( (N-1) \). In other words, given the Z-transform for a Walsh function for its first period, we need only to multiply it by the factor \( (1-Z^{-N})^{-1} \) to arrive at the closed form
Z-transform for all k.

Applying this "indirect" method to the previous example for sal (k, 1, 2), we have

\[ \sum_{k=0}^{N-1} f(k)z^{-k} = \frac{1}{1 - z^{-1}} \]

Therefore, from equation (160):

\[ F(z) = (1 - z^{-N})^{-1} \sum_{k=0}^{N-1} f(k)z^{-k} \]

\[ = (1 - z^{-2})^{-1} (1 - z^{-1}), \quad z > 1 \]

\[ = \frac{z}{z + 1}, \quad z > 1 \]

This result is identical to the previous result shown by equation (159).

Before applying the indirect method of determining the Z-transform to a more complicated function, let us define the notation to be used for indicating the Z-transforms of the Walsh functions. The definitions are as follows for the cal and sal functions:

\[ \text{sal}(Z, n, N) = \sum_{k=0}^{N-1} \text{sal}(k, n, N)z^{-k} \]  

(161)

\[ = \frac{1}{1 - z^{-N}} \text{sal}(Z, n, N) \]  

(162)

where

\[ \text{sal}(Z, 2, N) = \sum_{k=0}^{N-1} \text{sal}(k, n, N)z^{-k} \]  

(163)
Note that $Z_l$ is used to specify the Z-transform for one period of a given Walsh function whereas $Z$ is used to specify the Z-transform of the function for all $k$. (The definitions for the cal functions are the same except "sal" is replaced by "cal" in equations (161), (162), and (162).)

To illustrate the utility of the indirect method of determining the Z-transform for the Walsh functions, consider the case for sal $(k, 1, 8)$ as shown in Figure 34. From equation (163) and Figure 34, we may write:

$$\text{sal} (Z_l, 1, 8) = \frac{7}{k=0} \sum_{k=0}^{7} \text{sal} (k, 1, 8) Z^{-k}$$

$$\text{sal} (Z_l, 1, 8) = 1 + Z^{-1} + Z^{-2} + Z^{-3} - Z^{-4} - Z^{-5} - Z^{-6} - Z^{-7}$$

$$\text{sal} (Z_l, 1, 8) = (1 + Z^{-1}) (1 + Z^{-2}) (1 - Z^{-4})$$

From equation (162),

$$\text{sal} (Z, 1, 8) = (1 - Z^{-8})^{-1} \text{sal} (Z_l, 1, 8)$$

$$\text{sal} (Z, 1, 8) = (1 - Z^{-8})^{-1} (1 + Z^{-1}) (1 + Z^{-2}) (1 - Z^{-4})$$

$$\text{sal} (Z, 1, 8) = \frac{Z (Z + 1) (Z^2 + 1)}{Z^4 + 1}$$

Table 12 lists sal $(Z_l, n, N)$ for $n = 1, 2, 3, \ldots 8$; $N = 2, 4, 8, 16$ and cal $(Z_l, n, N)$ for $n = 0, 1, 2, \ldots 7$; $N = 2, 4, 8, 16$. Table 13 extends the list in Table 12 to the case for $N = 32$. It can be seen from the symmetry characteristics of the transforms in these tables
Table 12. Z-transforms of the Walsh functions
samples at the 2, 4, 8, and 16 rates

<table>
<thead>
<tr>
<th>( f(k,n,N) )</th>
<th>( F(z,1,n,N) )</th>
<th>( F(z,n,N) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{cal}(k,0,2) )</td>
<td>( 1+z^{-1} )</td>
<td>( z/(z-1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,1,2) )</td>
<td>( 1-z^{-1} )</td>
<td>( z/(z+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,0,4) )</td>
<td>( 1+z^{-1} ) ( (1+z^{-2}) )</td>
<td>( z/(z-1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,1,4) )</td>
<td>+</td>
<td>( z/(z+1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,1,4) )</td>
<td>-</td>
<td>( z/(z-1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,2,4) )</td>
<td>+</td>
<td>( z/(z+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,0,8) )</td>
<td>( (1+z^{-1})(1+z^{-2})(1+z^{-4}) )</td>
<td>( z/(z-1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,1,8) )</td>
<td>+</td>
<td>( z/(z+1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,1,8) )</td>
<td>-</td>
<td>( z/(z-1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,2,8) )</td>
<td>+</td>
<td>( z/(z+1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,2,8) )</td>
<td>-</td>
<td>( z/(z-1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,3,8) )</td>
<td>-</td>
<td>( z/(z-1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,3,8) )</td>
<td>+</td>
<td>( z/(z+1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,4,8) )</td>
<td>-</td>
<td>( z/(z+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,0,16) )</td>
<td>( (1+z^{-1})(1+z^{-2})(1+z^{-4})(1+z^{-8}) )</td>
<td>( z/(z-1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,1,16) )</td>
<td>+</td>
<td>( z/(z+1)/(z^2+1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,1,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2+1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,2,16) )</td>
<td>+</td>
<td>( z/(z+1)/(z^2+1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,2,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2+1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,3,16) )</td>
<td>+</td>
<td>( z/(z+1)/(z^2-1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,3,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2-1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,4,16) )</td>
<td>+</td>
<td>( z/(z+1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,4,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,5,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2+1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,5,16) )</td>
<td>-</td>
<td>( z/(z-1)/(z^2-1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,6,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2-1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,6,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2+1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,7,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2+1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{cal}(k,7,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2+1)/(z^4+1) )</td>
</tr>
<tr>
<td>( \text{sal}(k,8,16) )</td>
<td>+</td>
<td>( z/(z-1)/(z^2+1)/(z^4+1) )</td>
</tr>
</tbody>
</table>
Table 13. Z-transforms of the Walsh functions sampled at the 32 rate

<table>
<thead>
<tr>
<th>f(k,n,N)</th>
<th>P(zl,n,N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cal(k,0,32)</td>
<td>(1+z&lt;sup&gt;-1&lt;/sup&gt;) (1+z&lt;sup&gt;-2&lt;/sup&gt;) (1+z&lt;sup&gt;-4&lt;/sup&gt;) (1+z&lt;sup&gt;-8&lt;/sup&gt;) (1+z&lt;sup&gt;-16&lt;/sup&gt;)</td>
</tr>
<tr>
<td>sal(k,1,32)</td>
<td>+ + + + -</td>
</tr>
<tr>
<td>cal(k,1,32)</td>
<td>+ + + - -</td>
</tr>
<tr>
<td>sal(k,2,32)</td>
<td>+ + + - +</td>
</tr>
<tr>
<td>cal(k,2,32)</td>
<td>+ + - - +</td>
</tr>
<tr>
<td>sal(k,3,32)</td>
<td>+ + - - -</td>
</tr>
<tr>
<td>cal(k,3,32)</td>
<td>+ + - + -</td>
</tr>
<tr>
<td>sal(k,4,32)</td>
<td>+ + - - +</td>
</tr>
<tr>
<td>cal(k,4,32)</td>
<td>+ + - + -</td>
</tr>
<tr>
<td>sal(k,5,32)</td>
<td>+ + - + -</td>
</tr>
<tr>
<td>cal(k,5,32)</td>
<td>+ + - + -</td>
</tr>
<tr>
<td>sal(k,6,32)</td>
<td>+ + - - -</td>
</tr>
<tr>
<td>cal(k,6,32)</td>
<td>+ + - - -</td>
</tr>
<tr>
<td>sal(k,7,32)</td>
<td>+ + - - -</td>
</tr>
<tr>
<td>cal(k,7,32)</td>
<td>+ + - - -</td>
</tr>
<tr>
<td>sal(k,8,32)</td>
<td>+ + - - -</td>
</tr>
<tr>
<td>cal(k,8,32)</td>
<td>+ + - - -</td>
</tr>
<tr>
<td>sal(k,9,32)</td>
<td>+ + - - -</td>
</tr>
<tr>
<td>cal(k,9,32)</td>
<td>+ + - - -</td>
</tr>
<tr>
<td>sal(k,10,32)</td>
<td>- - - - -</td>
</tr>
<tr>
<td>cal(k,10,32)</td>
<td>- - - - -</td>
</tr>
<tr>
<td>sal(k,11,32)</td>
<td>- - - - -</td>
</tr>
<tr>
<td>cal(k,11,32)</td>
<td>- - - - -</td>
</tr>
<tr>
<td>sal(k,12,32)</td>
<td>- - - - -</td>
</tr>
<tr>
<td>cal(k,12,32)</td>
<td>- - - - -</td>
</tr>
<tr>
<td>sal(k,13,32)</td>
<td>- + - - -</td>
</tr>
<tr>
<td>cal(k,13,32)</td>
<td>- + - - -</td>
</tr>
<tr>
<td>sal(k,14,32)</td>
<td>- + - - -</td>
</tr>
<tr>
<td>cal(k,14,32)</td>
<td>- + - - -</td>
</tr>
<tr>
<td>sal(k,15,32)</td>
<td>- + - - -</td>
</tr>
<tr>
<td>cal(k,15,32)</td>
<td>- + - - -</td>
</tr>
<tr>
<td>sal(k,16,32)</td>
<td>- + - - -</td>
</tr>
</tbody>
</table>
that if the Zl-transforms of cal and sal functions of sequency \( n \) and length \( N/2 \) are known, then Zl-transforms of cal and sal functions of length \( N \) can be computed recursively by using the following relationships:

\[
\begin{align*}
cal (Zl, 2n, N) &= cal (Zl, n, N/2) \cdot (1 + Z^{-N/2}) \\
sal (Zl, 2n+1, N) &= cal (Zl, n, N/2) \cdot (1 - Z^{-N/2}) \\
cal (Zl, 2n-1, N) &= sal (Zl, n, N/2) \cdot (1 - Z^{-N/2}) \\
sal (Zl, 2n, N) &= sal (Zl, n, N/2) \cdot (1 + Z^{-N/2})
\end{align*}
\]

where recursion starts with \( N = 2 \).

Note that since the general form of the Zl-transform is a product of \( \log_2 N \) terms like \((1 \pm Z^{-1})\), \((1 \pm Z^{-2})\), \ldots, \((1 \pm Z^{-N/2})\), in Tables 12 and 13 only, the + or - sign is shown in each term.

It should also be pointed out that only values of the transforms are given here for \( N = 2^P \) where \( P \) is a positive integer. The reason for considering only these values of \( N \) is based on the sampling requirements for the resonant sequency filter.

Also, in Table 12, a listing of Z-transforms for \( N = 2, 4, 8, 16 \) is given. We are able to realize the following recursive relationships in addition to those
given earlier:

\[ \text{cal} (Z, n, N) = \text{cal} (Z, 2^P n, 2^P N), \ n = 0, 1, 2, \ldots \ (165) \]

\[ \text{sal} (Z, n, N) = \text{sal} (Z, 2^P n, 2^P N), \ n = 1, 2, 3, \ldots \]

where

\[ N = 2, 4, 8, 16, \ldots \ldots, 2n \leq N \text{ and } P \text{ is a positive integer.} \]
APPENDIX B

THE SIMULATION PROGRAM

PROGRAM LISTING

The Fortran IV version of the simulation program used to simulate the resonant Walsh function filter discussed in Chapter 5 is listed in Table 14. The relationship between the program variables and the filter variables as used in the Chapter 5 text are as follows:

\[
\begin{align*}
KS(K) & \triangleq S(k) \\
KV11(K) & \triangleq v_1'(k) \\
KV1(K) & \triangleq v_1(k) \\
KV2(K) & \triangleq v_2(k)
\end{align*}
\]

Inasmuch as the Fortran IV compiler used during the simulation did not accommodate zero subscripted variables, Table 15 shows the relationships used in the program for setting up the filter inputs in order to overcome the subscription problem.

It should be noted that the program was used to simulate a specific resonant filter during a given program run. Consequently, the program statements marked by an asterisk (*) were changed from run to run in order to simulate different filters.
Table 14 Simulation Program Listing

DIMENSION KWAL(8,17), KS(18), KV11(18), KV1(18), KV2(18)
* KS(2) = 0
* KS(3) = 0
* KS(4) = 0
* KS(5) = 0
KMAX = 17
NMAX = 8
DO 1 K = 2, KMAX - 3, 1
1 KS(K+4) = KS(K)
DO 2 N = 1, 8, 1
2 READ 3, (KWAL(N, K), K = 2, 9)
3 FORMAT (8I2)
   DO 4 N = 1, NMAX, 1
   D) 4 K = 2, KMAX - 8, 1
4 KWAL(N, K+8) = KWAL(N, K)
DO 5 K = 2, KMAX, 1
* 5 KV11(K) = KWAL(1, K)
KV1(1) = 0
KV2(1) = 0
DO 6 K = 1, KMAX
   KV1(K+1) = (KS(K+1)) * (KV2(K) - KV1(K)) + KV1(K) + KV11(K+1)
6 KV2(K+1) = KV2(K) - ((KS(K+1)) + KV2(K) - KV1(K))
   PRINT 8, (KV10K), K = 2, KMAX
   PRINT 8, (KV2(K), K = 2, KMAX)
8 FORMAT (1H0, 18I3)
   CALL EXIT
END
Table 15 Input Data for Simulation Program

<table>
<thead>
<tr>
<th>Programmed Input</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Corresponding Walsh Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>KWAL(1,k)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>cal(0,0)</td>
</tr>
<tr>
<td>KWAL(2,k)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>sal(1,0)</td>
</tr>
<tr>
<td>KWAL(3,k)</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>cal(1,0)</td>
</tr>
<tr>
<td>KWAL(4,k)</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>sal(2,0)</td>
</tr>
<tr>
<td>KWAL(5,k)</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>cal(2,0)</td>
</tr>
<tr>
<td>KWAL(6,k)</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>sal(3,0)</td>
</tr>
<tr>
<td>KWAL(7,k)</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>cal(3,0)</td>
</tr>
<tr>
<td>KWAL(8,k)</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>sal(4,0)</td>
</tr>
</tbody>
</table>
ADDITIONAL OUTPUT DATA FOR PHASE II SIMULATION

Tables 16 through 22 are tabulations of the input/output responses derived via the Phase II simulation of Chapter 5. These seven tables of data plus the data given in Table 6 of Chapter 5 serve to verify the selectivity and anti-resonant characteristics of the resonant Walsh function filter.
Table 16 Input-Output Response for the cal (k, o, 8) Resonant Filter

<table>
<thead>
<tr>
<th>$v_1'(k)$</th>
<th>$v_2(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>cal (k, 0, 8)</td>
<td>0</td>
</tr>
<tr>
<td>sal (k, 1, 8)</td>
<td>0</td>
</tr>
<tr>
<td>cal (k, 1, 8)</td>
<td>0</td>
</tr>
<tr>
<td>sal (k, 2, 8)</td>
<td>0</td>
</tr>
<tr>
<td>cal (k, 2, 8)</td>
<td>0</td>
</tr>
<tr>
<td>sal (k, 3, 8)</td>
<td>0</td>
</tr>
<tr>
<td>cal (k, 3, 8)</td>
<td>0</td>
</tr>
<tr>
<td>sal (k, 4, 8)</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 17 Input-Output Response for the sal (k, 1, 8) Resonant Filter

<table>
<thead>
<tr>
<th>$\bar{v}_1(k)$</th>
<th>$k$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>cal (k, 0, 8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>sal (k, 1, 8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>-4</td>
<td>-4</td>
<td>-4</td>
<td>-4</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>cal (k, 1, 8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sal (k, 2, 8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cal (k, 2, 8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sal (k, 3, 8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cal (k, 3, 8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sal (k, 4, 8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( v_1'(k) )</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>( \text{cal} , (k, 0, 8) )</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>( \text{sal} , (k, 1, 8) )</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( \text{cal} , (k, 1, 8) )</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-4</td>
<td>-4</td>
<td>-4</td>
<td>-4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>-8</td>
<td>-8</td>
<td>-8</td>
<td></td>
</tr>
<tr>
<td>( \text{sal} , (k, 2, 8) )</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( \text{cal} , (k, 2, 8) )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( \text{sal} , (k, 3, 8) )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( \text{cal} , (k, 3, 8) )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( \text{sal} , (k, 4, 8) )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Table 19 Input-Output Response for the sal (k, 2, 8) Resonant Filter

<table>
<thead>
<tr>
<th>( v_1(k) )</th>
<th>( v_2(k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{cal (k, 0, 8)} )</td>
<td>0.0 0.0 2.2 2.2 2.2 4.4 4.4 4.4 6.6 6.6 6.6 6.6 8.8 8.8</td>
</tr>
<tr>
<td>( \text{sal (k, 1, 8)} )</td>
<td>0.0 0.0 2.2 2.2 2.2 2.2 0.0 0.0 0.0 2.2 2.2 2.2 2.2 0.0 0.0</td>
</tr>
<tr>
<td>( \text{cal (k, 1, 8)} )</td>
<td>0.0 0.0 2.2 2.2 2.2 2.2 0.0 0.0 0.0 2.2 2.2 2.2 2.2 0.0 0.0</td>
</tr>
<tr>
<td>( \text{sal (k, 2, 8)} )</td>
<td>0.0 0.0 2.2 2.2 2.2 2.2 4.4 4.4 4.4 6.6 6.6 6.6 6.6 8.8 8.8</td>
</tr>
<tr>
<td>( \text{cal (k, 2, 8)} )</td>
<td>0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>( \text{sal (k, 3, 8)} )</td>
<td>0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>( \text{cal (k, 3, 8)} )</td>
<td>0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>( \text{sal (k, 4, 8)} )</td>
<td>0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>$v_1(k)$</td>
<td>$k$</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
</tr>
<tr>
<td>cal $(k, 0, 8)$</td>
<td>0</td>
</tr>
<tr>
<td>cal $(k, 1, 8)$</td>
<td>0</td>
</tr>
<tr>
<td>cal $(k, 2, 8)$</td>
<td>0</td>
</tr>
<tr>
<td>cal $(k, 3, 8)$</td>
<td>0</td>
</tr>
<tr>
<td>cal $(k, 4, 8)$</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 21 Input-Output Response for the cal (k, 3, 8) Resonant Filter

<table>
<thead>
<tr>
<th>$v_1(k)$</th>
<th>$v_2(k)$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>cal (k, 0, 8)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>sal (k, 1, 8)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>cal (k, 1, 8)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>sal (k, 2, 8)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>cal (k, 2, 8)</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>sal (k, 3, 8)</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>cal (k, 3, 8)</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>2</td>
<td>-3</td>
<td>3</td>
<td>-4</td>
<td>-4</td>
<td>5</td>
<td>-5</td>
<td>6</td>
<td>6</td>
<td>-7</td>
<td>7</td>
<td>-8</td>
<td></td>
</tr>
<tr>
<td>sal (k, 4, 8)</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Table 22 Input-Output Response for the sal (k, 4, 8)
Resonant Filter

<table>
<thead>
<tr>
<th>$v_1'(k)$</th>
<th>$k$</th>
<th>$v_2(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>cal (k, 0, 8)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sal (k, 1, 8)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>cal (k, 1, 8)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sal (k, 2, 8)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>cal (k, 2, 8)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sal (k, 3, 8)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>cal (k, 3, 8)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sal (k, 4, 8)</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


75. Ibid., pp. 57-59.
