One of the early methods for pulse compression is by phase coding. We start from a pulse of duration $T$. The pulse is divided into $M$ bits of identical duration $t_b = T/M$, and each bit is assigned (or coded) with a different phase value. The complex envelope of the phase-coded pulse is given by

$$u(t) = \frac{1}{\sqrt{T}} \sum_{m=1}^{M} u_m \text{rect} \left( \frac{t - (m - 1)t_b}{t_b} \right)$$

where $u_m = \exp(j\phi_m)$ and the set of $M$ phases $\{\phi_1, \phi_2, \ldots, \phi_M\}$ is the phase code associated with $u(t)$.

Finding optimal sets of $M$ phases (or codes) for different radar application has kept radar engineers busy from the early days of radar. The number of possibilities of generating phase codes of length $M$ is unlimited. The criteria for selecting a specific code are the resolution properties of the resulting waveform (shape of the ambiguity function), frequency spectrum, and the ease with which the system can be implemented. Sometimes the design is even more complicated by using different phase codes for the transmitted pulse and the reference pulse used at the receiver (possibly even with different lengths). This can improve resolution at the expense of a suboptimal signal-to-noise ratio.

The problem of finding a code that leads to a predetermined range–Doppler resolution is very complicated. A more manageable problem is finding a code with a good correlation function rather than an ambiguity function. The correlation function of a phase-coded pulse is a continuous function of the delay $\tau$. In general,
when examining the properties of the correlation function, one should look at the correlation function for all \(-T < \tau < T\). However, as we show in Box 6A, it is sufficient to calculate the correlation function at integer multiples of the bit duration. The values in between are then obtained by connecting the values at \(\tau = nt_b\) using straight lines in the complex plane. It is easy to see using Fig. 6.1 that the interpolation in the complex plane results with concave sections of the correlation function magnitude (i.e., the autocorrelation peaks are obtained at an integer multiples of the bit duration). Thus, minimizing the peak value, or the area (integral) under the continuous correlation function \(|R(\tau)|\), is simplified to finding the peak value and sum of the discrete correlation function \(|R_k|\).

**BOX 6A: Aperiodic Correlation Function of a Phase-Coded Pulse**

Consider a transmitted pulse \(u(t)\) with \(M_u\) phase elements defined by \(u_m\) (1 \(\leq\) \(m\) \(\leq\) \(M_u\)) and a reference pulse \(v(t)\) with \(M_v\) elements defined by \(v_n\) (1 \(\leq\) \(n\) \(\leq\) \(M_v\)). The cross-correlation function of the two phase-coded pulses is defined by

\[
R_{uv}(\tau) = \int_{-\infty}^{\infty} u(t)v^*(t + \tau) \, d\tau
\]

\[
= \int_{-\infty}^{\infty} \frac{1}{\sqrt{M_u t_b}} \sum_{m=1}^{M_u} u_m \text{rect} \left[ \frac{t - (m - 1)t_b}{t_b} \right] \frac{1}{\sqrt{M_v t_b}}
\]
\[
\sum_{n=1}^{M_v} v_n^* \operatorname{rect} \left[ \frac{t + \tau - (n - 1)t_b}{t_b} \right] d\tau
\]

(6A.1)

\[
= \frac{1}{t_b \sqrt{M_u M_v}} \sum_{m=1}^{M_u} \sum_{n=1}^{M_v} u_m v_n^* \int_{-\infty}^{\infty} \operatorname{rect} \left[ \frac{t - (m - 1)t_b}{t_b} \right] \operatorname{rect} \left[ \frac{t + \tau - (n - 1)t_b}{t_b} \right] d\tau
\]

Since

\[
\int_{-\infty}^{\infty} \operatorname{rect} \left[ \frac{t - (m - 1)t_b}{t_b} \right] \operatorname{rect} \left[ \frac{t + \tau - (n - 1)t_b}{t_b} \right] d\tau
\]

\[
= \begin{cases} t_b - |\tau - (n - m)t_b|, & |\tau - (n - m)t_b| < t_b \\ 0, & |\tau - (n - m)t_b| \geq t_b \end{cases}
\]

the integral in (6A.1) represents a triangle having a peak of \( t_b \) and centered at \( (n - m)t_b \) with a base width of \( 2t_b \). If we write the delay as \( \tau = kt_b + \eta \), where \( 0 \leq \eta < t_b \) and \( k \) is an integer, the integral will be nonzero only for \(-t_b < kt_b + \eta - (n - m)t_b < t_b\). Since \( 0 \leq \eta < t_b \), we can solve for \( n \) as a function of \( m \) and \( k \) and find that when \( \eta > 0 \), the integral is nonzero only for \( n_1 = k + m \) and \( n_2 = k + m + 1 \). When \( \eta = 0 \), the integral is nonzero only for \( n = k + m \). Using \( n_1 \) and \( n_2 \) in (6A.1), we can write the correlation function as

\[
R_{uv}(kt_b + \eta) = \frac{1}{t_b \sqrt{M_u M_v}} \left[ (t_b - \eta) \sum_{m=1}^{\min(M_u, M_v - k)} u_m v_{m+k}^* + \eta \sum_{m=1}^{\min(M_u, M_v - k - 1)} u_m v_{m+k+1}^* \right]
\]

(6A.2)

Note that the single sums in (6A.2) are now the discrete aperiodic cross-correlations between \( u_m \) and \( v_m \) evaluated at \( \tau = k \) (first sum) and \( \tau = k + 1 \) (second sum). Thus,

\[
R_{uv}(\tau) = R_{uv}(kt_b + \eta) = \frac{1}{t_b \sqrt{M_u M_v}} [(t_b - \eta) R_{uv}[k] + \eta R_{uv}[k + 1]]
\]

(6A.3)

where \( R_{uv}[k] \) is the discrete aperiodic cross-correlation function of sequences \( u \) and \( v \) evaluated at \( \tau = k \). Note that the interpolation is done in the complex plane (see Fig. 6.1).
The discrete cross-correlation of two sequences can be calculated using a procedure demonstrated below. To simplify the example, sequences with binary (±1) values are used; the same procedure can be implemented on complex-valued sequences. Consider the phase-coded pulse defined by the sequence \( \{ u_m \} = \{-1 1 -1 -1 1 1 -1 1 -1\} \) and the reference phase-coded pulse \( \{ v_m \} = \{-1 1 -1 -1 1 1 -1 -1\} \) with the same number of elements \( M_u = M_v = 9 \). To calculate the cross-correlation we multiply the conjugate of the reference signal by the elements of the transmitted sequence and add the shifted sequences as demonstrated in Table 6.1.

Figure 6.2 shows the full (continuous) cross-correlation function of \( u(t) \) and \( v(t) \) obtained by connecting the discrete values of \( R_{uv} \) using straight lines (bottom). The figure also shows the autocorrelation function where no phase coding is used (dashed) and the autocorrelation function of \( u(t) \) (top). Note how phase coding lowers the mainlobe width. Note also that the autocorrelation of \( u(t) \) is symmetric, while the cross-correlation of \( u(t) \) and \( v(t) \) is not symmetric.

Different phase codes that yield identical aperiodic autocorrelation function magnitude are called equivalent. Using the four properties of the cross-correlation function described in Box 6B, it is easy to see that the following operations on a phase code \( u_m \) give equivalent phase codes:

1. A reversal transformation: \( \hat{u}_m = u_{M-m} \).
2. A conjugate transformation: \( \hat{u}_m = u_m^* \).
3. A constant multiplication transformation: \( \hat{u}_m = \eta u_m \), where \( |\eta| = 1 \).
4. A progressive multiplication transformation: \( \hat{u}_m = \rho^m u_m \), where \( |\rho| = 1 \).

It is also easy to show that every polyphase code is equivalent to one that begins with \( \phi_1 = 0, \phi_2 = 0, \phi_3 \), where \( 0 \leq \phi_3 \leq \pi \). This form of the polyphase code is known as the normalized form.

<table>
<thead>
<tr>
<th>TABLE 6.1 Construction Table of the Discrete Cross-Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( { v_m^* } )</td>
</tr>
<tr>
<td>( { u_m } )</td>
</tr>
</tbody>
</table>
| -1 1 -1 1 1 1 -1 -1 -1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
BOX 6B: Properties of the Cross-Correlation Function of a Phase Code

Property I: Reversal transformation: The cross-correlation of reversed-order sequences is the reversed-order cross-correlation of the original sequences:

\[ R_{\hat{u}\hat{v}}[k] = \sum_{m} \hat{u}_m \hat{v}_{m+k} = \sum_{m} u_{M-m} v_{M-m-k} = \sum_{m'} u_{m'} v_{m'-k} \]

\[ = R_{uv}[-k] = R_{uv}^*[k] \]

Note that due to symmetry property of the autocorrelation the autocorrelation of the reversed-order sequence equals the conjugate of the original sequence autocorrelation.

Property II: Conjugation transformation. The cross-correlation of conjugated sequences is the conjugated cross-correlation of the original sequences:

\[ R_{\hat{u}\hat{v}}[k] = \sum_{m} \hat{u}_m \hat{v}_{m+k} = \sum_{m} u_m^* v_{m+k} = R_{uv}^*[k] \]
**Property III:** Constant multiplication transformation. The cross-correlation of sequences multiplied by a constant is the multiplied cross-correlation of the original sequences:

\[
R_{\hat{u}\hat{v}}[k] = \sum_{m} \hat{u}_{m} \hat{v}_{m+k} = \sum_{m} \eta_{u} u_{m} \eta_{v} v_{m+k} = \eta_{u} \eta_{v} \sum_{m} u_{m} v_{m+k} = \eta_{u} \eta_{v} R_{uv}[k]
\]

**Property IV:** Progressive multiplication transformation. The cross-correlation of progressively multiplied sequences is the progressively multiplied cross-correlation of the original sequences:

\[
R_{\hat{u}\hat{v}}[k] = \sum_{m} \hat{u}_{m} \hat{v}_{m+k} = \sum_{m} \rho_{m} u_{m} (\rho_{v}^{m+k}) \ast v_{m+k} = (\rho_{v}^{k}) \ast \left( \sum_{m} (\rho_{u} \rho_{v})^{m} u_{m} v_{m+k} \right)
= (\rho_{k}) \ast \left( \sum_{m} u_{m} v_{m+k} \right) = (\rho_{k}) \ast R_{uv}[k]
\]

where we used \( \rho_{u} = \rho_{v} = \rho \).

In some applications the resolution properties are determined by periodic correlation of the phase-coded pulse. In addition to the previous transformations, the following transformations preserve the periodic autocorrelation function (the aperiodic correlation function is not necessarily preserved):

5. Cyclically shifting the sequence: \( \hat{u}_{m} = u_{m+a \mod M} \).
6. Decimating the sequence by \( d \), which is relatively prime to the sequence length \( M \) results with decimating the periodic autocorrelation function: \( \hat{u}_{m} = u_{md \mod M} \). Note that some authors refer to the codes obtained by decimation as permutation codes.

Although it is not possible to design phase-coded pulses with zero aperiodic correlation sidelobes, the periodic correlation of a phase-coded signal can be zero for all nonzero shifts. Phase codes having zero periodic autocorrelation sidelobes are called perfect codes. One method for designing low aperiodic autocorrelation codes is to start with a perfect code. Transformations (5) and (6) are used to find optimal aperiodic autocorrelation function codes while keeping the code perfect. Note that for perfect codes the aperiodic correlation function has sidelobes symmetric around \( \tau = Mt_{b}/2 \) [i.e., \( R_{uv}(\tau) = -R_{uv}(M - \tau) \) for \( t_{b} \leq |\tau| \leq Mt_{b} \)].

### 6.1 Barker Codes

Probably the most famous family of phase codes is named Barker, after its inventor (Barker, 1953). Originally, the Barker codes were designed as the sets
of $M$ binary phases yielding a peak-to-peak sidelobe ratio of $M$. For example, the autocorrelation function of the $M = 13$-element Barker code is shown in Fig. 6.3. All known binary sequences yielding a peak-to-peak sidelobe ratio of $M$ were reported by Barker (1953) and Turyn (1963) and are given in Table 6.2. Although it was only proved that no binary Barker codes exist for $13 < M < 1,898,884$ and that no binary Barker codes exist for all odd $M > 13$ (Turyn, 1963; Eliahou and Kervaire, 1992), it is common belief that no Barker codes exist for all $M > 13$. In Barker and other phase-coded signals, the instantaneous phase switching causes extended spectral sidelobes. The effect of a phase-switching slope on bandwidth and autocorrelation function is discussed in Section 6.8. A Barker signal is used as an example.

6.1.1 Minimum Peak Sidelobe Codes

Binary codes that yield minimum peak sidelobes but do not meet the Barker condition (i.e., the peak-to-peak sidelobe ratio is less than $M$) are often called minimum peak sidelobe (MPS) codes. Finding MPS codes involves exhaustive
computer search. Results up to $M \leq 69$ were reported by Lindner (1975), Cohen et al. (1989, 1990) and Coxson et al. (2001). Codes with a peak sidelobe of 2 were reported for $M \leq 28$. The MPS codes reported for $28 < M \leq 48$ and $M = 51$ have a sidelobe level of 3, and the MPS codes of length $M = 50$ and $52 \leq M \leq 69$ have a sidelobe level of 4. Table 6.3 gives a single MPS code for each length up to 69. The table also gives the peak sidelobe level. For $M \leq 48$ the listed codes are those that have, from all those with minimum peak sidelobe, the minimum integrated sidelobe.

It seems that for any peak sidelobe level there is a limit of the maximal value of $M$ for which a binary sequence with that sidelobe level exists (i.e., $M = 13$ for peak sidelobe level of 1, $M = 28$ for peak sidelobe level of 2, $M = 51$ for peak sidelobe level of 3, etc.). The main advantage of the binary Barker and minimum peak sidelobe codes is in their relatively low complexity (no multiplications are needed at the receiver). The main drawback is that such codes are only known for limited values of $M$, and finding additional codes with the minimum peak sidelobe for any given length $M$ involves an exhaustive search with a size growing exponentially with $M$. Some methods for finding potentially good suboptimal peak sidelobe codes were reported (e.g., Schroeder, 1970), and some good codes (not necessarily optimal) were reported for higher values of $M$ [e.g. $M = 51$, 69, and 88 by Kerdock et al. (1986), and for $M = 53$, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, and 113 by Boehner (1967)].

### 6.1.2 Nested Codes

One early method of generating low peak sidelobe sequences for large values of $M$ is by nesting codes of shorter length. We demonstrate the concept of nested codes by forming a 39-element code using the three-element Barker code ($\{v_m\} = \{1 \ 1 \ -1\}, M_v = 3$) and the 13-element Barker code ($\{u_m\} = \{1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1\}, M_u = 13$). There are two forms of generating elements of the nested code. The first is given by $u \otimes v$ and the second is given by $v \otimes u$, where $\otimes$ is the Kronecker product. When the code is formed by $v \otimes u$, $u$ is called the inner code and $v$ is referred to as the outer code. The nested codes resulting from the two forms defined above are given by

\[
v \otimes u = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}
\]

and

\[
u \otimes v = \begin{bmatrix} 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}
\]
### TABLE 6.3 Minimum Peak Sidelobe Codes<sup>a</sup>

<table>
<thead>
<tr>
<th>M</th>
<th>PSL</th>
<th>Sample Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2</td>
<td>110100</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>10010111</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>01101011</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>01010011</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>100101110111</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>010100010011</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>0011000101011</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>0110010001110111</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>001110111010111</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>01100001101011</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>1011011110001111</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>0101000110000011011</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>10110110111011000011</td>
</tr>
<tr>
<td>22</td>
<td>3</td>
<td>00111001010101011111</td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>011010111110101001001</td>
</tr>
<tr>
<td>24</td>
<td>3</td>
<td>01100011010101111100011</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>100100101010000011100011</td>
</tr>
<tr>
<td>26</td>
<td>3</td>
<td>1000111000000010101011001</td>
</tr>
<tr>
<td>27</td>
<td>3</td>
<td>010010110111110111100011</td>
</tr>
<tr>
<td>28</td>
<td>3</td>
<td>10001110001000101011000111</td>
</tr>
<tr>
<td>29</td>
<td>3</td>
<td>1011001001010101100000011011</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1000110001010010100010000111</td>
</tr>
<tr>
<td>31</td>
<td>3</td>
<td>010101001011001100110000111</td>
</tr>
<tr>
<td>32</td>
<td>3</td>
<td>00000001110001101010101110001</td>
</tr>
<tr>
<td>33</td>
<td>3</td>
<td>01101101010101010101011111</td>
</tr>
<tr>
<td>34</td>
<td>3</td>
<td>10010110111111110000110100101011</td>
</tr>
<tr>
<td>35</td>
<td>3</td>
<td>000000000111100101101010011</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>00110011001001010100010000111111</td>
</tr>
<tr>
<td>37</td>
<td>3</td>
<td>00011011111001000111110111100111</td>
</tr>
<tr>
<td>38</td>
<td>3</td>
<td>0000000011110000110100101001100110</td>
</tr>
<tr>
<td>39</td>
<td>3</td>
<td>00111001011010100101111111111111110</td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>001000100100011111111110101001110</td>
</tr>
<tr>
<td>41</td>
<td>3</td>
<td>0001110011101010101001000000011010110000</td>
</tr>
<tr>
<td>42</td>
<td>3</td>
<td>001000100110011111111101100111110001101001</td>
</tr>
<tr>
<td>43</td>
<td>3</td>
<td>000000001011011001010101110011001110001111000</td>
</tr>
<tr>
<td>44</td>
<td>3</td>
<td>0000111111111111110000110101010101111</td>
</tr>
<tr>
<td>45</td>
<td>3</td>
<td>0001010111100001100010110111111111111111111</td>
</tr>
<tr>
<td>46</td>
<td>3</td>
<td>00001111111111111110001101111111111111111111111</td>
</tr>
<tr>
<td>47</td>
<td>3</td>
<td>00001101010110101111111101000101001100100001</td>
</tr>
<tr>
<td>48</td>
<td>3</td>
<td>000100101101101011000111111111010111111111111111</td>
</tr>
<tr>
<td>49</td>
<td>4</td>
<td>000011000101011011111110001111111111111111111111111</td>
</tr>
<tr>
<td>50</td>
<td>4</td>
<td>0000100101000110010111011111100010011111111111111111111</td>
</tr>
<tr>
<td>51</td>
<td>3</td>
<td>00011100011111111110001101000101010100101010101011</td>
</tr>
<tr>
<td>52</td>
<td>4</td>
<td>000011001101010110111100001111111111111111111111111111111</td>
</tr>
<tr>
<td>53</td>
<td>4</td>
<td>000100111101010110101111111111111111111111111111111111111</td>
</tr>
<tr>
<td>54</td>
<td>4</td>
<td>000010010101010110010110000011010010111111111111111111111111</td>
</tr>
<tr>
<td>55</td>
<td>4</td>
<td>000100110001010101101101000101100101001111111111111111111111111</td>
</tr>
<tr>
<td>56</td>
<td>4</td>
<td>00001001010101011010101111100011111111111111111111111111111111111</td>
</tr>
</tbody>
</table>
TABLE 6.3  (continued)

<table>
<thead>
<tr>
<th>M</th>
<th>PSL</th>
<th>Sample Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>4</td>
<td>0000100100110100101000111011011010100011100011010111</td>
</tr>
<tr>
<td>58</td>
<td>4</td>
<td>000010001111010011001001110001011000110100110101111</td>
</tr>
<tr>
<td>59</td>
<td>4</td>
<td>000010010011101001110000011001000101000111011001011</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>00001010101110011101111100011100011011001010101111</td>
</tr>
<tr>
<td>61</td>
<td>4</td>
<td>0000001101101011000110101100110011110110111010101000</td>
</tr>
<tr>
<td>62</td>
<td>4</td>
<td>000001011010110100111101001101001111001100101010111</td>
</tr>
<tr>
<td>63</td>
<td>4</td>
<td>0000100111011011010110100110110011011101111101110111</td>
</tr>
<tr>
<td>64</td>
<td>4</td>
<td>01001001001000010100111011001100010110111100101111100010</td>
</tr>
<tr>
<td>65</td>
<td>4</td>
<td>0000000110110110011011101011011001111110111001101010000</td>
</tr>
<tr>
<td>66</td>
<td>4</td>
<td>00000001101100110110001011001101111001111110101101111010000</td>
</tr>
<tr>
<td>67</td>
<td>4</td>
<td>0100000010100011101110101100110111111111010100110111111100010</td>
</tr>
<tr>
<td>68</td>
<td>4</td>
<td>000000011011001101110011010011011100110110110111100110101110000</td>
</tr>
</tbody>
</table>
| 69 | 4   | 00111010101111111111101101111111100110111111111111111111011111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
phases have been carried out using numerical optimization techniques. Examples of such sequences were found for all $M \leq 45$ (Bomer and Antweiler, 1989; Friese and Zottmann, 1994; Friese, 1996). Table 6.4 gives the normalized form (first two phases are zeroed) of such polyphase Barker sequences for $M \leq 45$.

### TABLE 6.4 Normalized Form of Some Polyphase Barker Codes with Large Alphabet

<table>
<thead>
<tr>
<th>$M$</th>
<th>PSL</th>
<th>Phase Values (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.50</td>
<td>104.52, 313.47</td>
</tr>
<tr>
<td>5</td>
<td>0.77</td>
<td>73.04, 225.31, 90.62</td>
</tr>
<tr>
<td>6</td>
<td>1.00</td>
<td>60.00, 180.00, 0.00, 240.00</td>
</tr>
<tr>
<td>7</td>
<td>0.53</td>
<td>106.48, 93.06, 316.72, 60.61, 270.86</td>
</tr>
<tr>
<td>8</td>
<td>0.66</td>
<td>72.33, 28.48, 294.09, 151.07, 250.77, 62.87</td>
</tr>
<tr>
<td>9</td>
<td>0.11</td>
<td>53.57, 42.22, 270.79, 215.59, 41.51, 161.92, 335.47</td>
</tr>
<tr>
<td>10</td>
<td>0.83</td>
<td>56.97, 127.04, 137.24, 12.74, 6.67, 224.63, 19.27, 233.50</td>
</tr>
<tr>
<td>11</td>
<td>0.89</td>
<td>34.17, 259.06, 266.63, 327.97, 158.47, 13.78, 22.74, 221.64, 94.65</td>
</tr>
<tr>
<td>12</td>
<td>0.91</td>
<td>104.89, 163.15, 171.04, 344.57, 241.31, 185.77, 282.58, 147.97, 209.41, 79.19</td>
</tr>
<tr>
<td>13</td>
<td>0.72</td>
<td>115.84, 114.84, 248.44, 213.38, 123.12, 154.90, 140.20, 12.75, 149.65, 303.48, 121.65</td>
</tr>
<tr>
<td>14</td>
<td>0.97</td>
<td>66.96, 133.73, 202.45, 100.74, 37.89, 236.27, 167.69, 86.72, 169.45, 34.20, 143.95, 14.33</td>
</tr>
<tr>
<td>15</td>
<td>0.80</td>
<td>17.81, 5.51, 5.37, 142.33, 211.98, 297.96, 123.75, 91.46, 1.09, 205.83, 314.02, 156.28, 23.66</td>
</tr>
<tr>
<td>16</td>
<td>0.93</td>
<td>26.46, 38.51, 97.32, 49.41, 305.85, 286.47, 197.00, 65.76, 241.32, 137.61, 319.19, 47.96, 178.58, 303.06</td>
</tr>
<tr>
<td>17</td>
<td>0.73</td>
<td>5.34, 18.49, 278.38, 307.59, 67.22, 148.91, 207.37, 70.46, 300.98, 282.64, 137.11, 6.31, 120.23, 327.59, 185.74</td>
</tr>
<tr>
<td>18</td>
<td>0.87</td>
<td>62.21, 45.41, 315.79, 282.94, 23.75, 37.27, 205.01, 186.58, 83.91, 155.50, 317.90, 337.25, 204.02, 11.17, 171.35, 31.96</td>
</tr>
<tr>
<td>19</td>
<td>0.96</td>
<td>54.66, 27.60, 91.21, 80.61, 235.94, 11.00, 333.09, 100.64, 241.78, 319.89, 162.37, 309.54, 162.20, 139.49, 33.30, 341.49, 218.99</td>
</tr>
<tr>
<td>20</td>
<td>0.98</td>
<td>99.16, 125.86, 232.99, 251.37, 133.93, 144.09, 354.74, 304.39, 192.21, 302.68, 219.51, 161.35, 283.77, 145.40, 250.28, 106.25, 228.47, 107.05</td>
</tr>
<tr>
<td>21</td>
<td>0.97</td>
<td>15.27, 30.14, 161.48, 203.92, 220.08, 190.88, 61.27, 126.36, 221.20, 340.41, 168.52, 153.07, 26.83, 255.87, 120.93, 209.07, 54.58, 239.74, 105.94</td>
</tr>
<tr>
<td>22</td>
<td>0.97</td>
<td>23.70, 53.81, 80.91, 74.08, 349.91, 264.11, 313.80, 245.26, 146.83, 73.98, 284.18, 159.95, 334.41, 77.11, 315.33, 145.77, 245.13, 343.88, 84.12, 206.36</td>
</tr>
<tr>
<td>23</td>
<td>0.91</td>
<td>7.39, 275.98, 286.43, 253.92, 256.71, 351.73, 58.40, 60.24, 226.34, 353.15, 100.57, 168.65, 41.05, 208.60, 347.90, 219.32, 126.05, 349.85, 315.45, 182.27, 56.51</td>
</tr>
<tr>
<td>24</td>
<td>0.99</td>
<td>5.05, 316.44, 257.26, 216.68, 202.65, 319.31, 311.49, 357.35, 297.32, 111.75, 36.71, 281.47, 137.65, 10.87, 116.56, 260.04, 135.27, 269.02, 29.14, 143.46, 209.62, 335.07</td>
</tr>
<tr>
<td>$M$</td>
<td>PSL</td>
<td>Phase Values (deg)</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>---------------------</td>
</tr>
<tr>
<td>25</td>
<td>0.93</td>
<td>81.86, 64.96, 316.26, 273.05, 326.28, 339.78, 62.69, 18.77, 270.52, 198.01, 98.77, 126.61, 256.47, 350.69, 105.91, 270.80, 295.42, 162.27, 334.2400, 155.49, 339.81, 147.69, 4.40</td>
</tr>
<tr>
<td>26</td>
<td>0.88</td>
<td>51.32, 117.02, 138.11, 265.30, 266.83, 175.22, 117.20, 259.95, 199.95, 195.78, 153.84, 178.61, 340.56, 186.87, 306.44, 193.87, 91.91, 189.55, 16.52, 109.34, 249.58, 37.87, 198.94</td>
</tr>
<tr>
<td>27</td>
<td>0.98</td>
<td>10.51, 21.75, 28.59, 324.46, 308.15, 117.83, 98.81, 111.75, 284.00, 199.23, 313.20, 115.64, 326.00, 184.04, 52.60, 7.95, 193., 96.11, 239.93, 334.30, 101.85, 227.48, 330.80, 91.72</td>
</tr>
<tr>
<td>28</td>
<td>0.95</td>
<td>46.92, 84.28, 166.29, 145.70, 199.79, 105.11, 116.57, 58.71, 109.70, 325.89, 24.31, 189.90, 21.40, 196.18, 58.82, 326.50, 129.18, 258.98, 306.73, 123.51, 111.19, 312.71, 298.50, 173.83, 97.89, 327.81</td>
</tr>
<tr>
<td>29</td>
<td>0.87</td>
<td>6.99, 318.38, 240.11, 264.97, 239.45, 160.78, 302.00, 327.98, 19.20, 320.43, 85.58, 109.43, 224.94, 7.08, 32.32, 185.08, 168.89, 91.05, 326.61, 228.81, 146.73, 331.28, 93.10, 265.24, 95.64, 254.51, 61.3100</td>
</tr>
<tr>
<td>30</td>
<td>1.00</td>
<td>33.11, 34.66, 33.75, 11.97, 300.21, 281.63, 26.64, 54.40, 155.79, 212.11, 231.81, 134.65, 76.33, 318.00, 276.10, 67.92, 299.31, 184.97, 73.01, 154.21, 7.02, 263.07, 94.57, 243.28, 359.63, 150.24, 306.92, 72.0800</td>
</tr>
<tr>
<td>31</td>
<td>0.94</td>
<td>28.39, 117.68, 165.05, 236.45, 308.63, 304.92, 236.37, 216.25, 327.24, 279.34, 211.11, 246.98, 191.79, 95.21, 117.41, 16.74, 272.73, 52.50, 330.82, 223.69, 303.38, 146.86, 21.37, 245.23, 28.92, 145.10, 296.68, 61.96, 190.35, 7.32</td>
</tr>
<tr>
<td>32</td>
<td>0.99</td>
<td>13.43, 16.18, 90.15, 109.65, 94.43, 59.91, 332.27, 306.11, 288.10, 280.54, 83.80, 162.77, 247.23, 333.62, 169.95, 72.93, 62.09, 219.53, 296.08, 107.98, 34.71, 270.22, 177.13, 16.92, 176.19, 285.32, 79.75, 289.11, 130.24, 326.30</td>
</tr>
<tr>
<td>33</td>
<td>0.97</td>
<td>143.07, 153.71, 339.20, 332.87, 180.92, 134.16, 19.42, 109.38, 166.70, 217.34, 226.71, 228.70, 319.75, 239.69, 185.92, 227.47, 143.26, 115.36, 76.97, 38.11, 187.58, 329.27, 228.90, 110.75, 304.75, 119.00, 275.49, 352.86, 190.92, 359.70, 167.67</td>
</tr>
<tr>
<td>34</td>
<td>0.96</td>
<td>11.60, 2.29, 308.93, 247.04, 202.65, 186.51, 234.77, 296.22, 303.84, 351.93, 49.89, 232.02, 253.31, 62.16, 340.45, 6.75, 133.71, 256.94, 76.56, 98.87, 323.47, 230.03, 65.62, 125.68, 248.19, 68.60, 297.53, 137.95, 284.29, 138.69, 17.64, 229.39</td>
</tr>
<tr>
<td>35</td>
<td>1.00</td>
<td>93.27, 65.44, 166.46, 132.46, 344.16, 279.49, 337.71, 301.42, 197.69, 56.27, 36.90, 9.33, 325.94, 334.44, 24.54, 157.98, 291.27, 301.26, 148.55, 113.09, 141.50, 296.81, 128.97, 125.66, 341.59, 130.11, 244.84, 74.10, 321.76, 157.86, 301.00, 107.79, 254.69</td>
</tr>
<tr>
<td>36</td>
<td>0.93</td>
<td>81.76, 117.41, 228.30, 227.39, 58.27, 59.76, 153.37, 108.01, 19.41, 233.18, 211.51, 260.82, 235.13, 195.44, 219.49, 114.51, 10.64, 224.56, 176.47, 119.93, 124.89, 74.97, 263.65, 112.81, 254.57, 106.85, 318.27, 98.35, 264.66, 28.37, 121.69, 244.37, 57.87, 183.84</td>
</tr>
</tbody>
</table>

(continued overleaf)
<table>
<thead>
<tr>
<th>$M$</th>
<th>PSL</th>
<th>Phase Values (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>0.91</td>
<td>66.51, 90.83, 123.03, 235.61, 235.65, 325.40, 14.62, 279.47, 224.10, 332.97, 5.58, 280.26, 202.90, 334.58, 51.37, 330.14, 260.07, 293.54, 159.60, 150.16, 83.19, 356.98, 124.60, 25.28, 314.51, 129.47, 362.27, 242.00, 50.51, 181.69, 315.88, 111.51, 189.30, 363.49, 148.70</td>
</tr>
<tr>
<td>38</td>
<td>0.93</td>
<td>39.45, 95.89, 111.27, 227.27, 253.77, 283.15, 336.93, 233.52, 209.97, 277.20, 332.82, 222.51, 179.24, 290.76, 55.61, 218.04, 37.83, 286.84, 5.30, 165.25, 337.35, 16.57, 244.77, 288.43, 167.97, 127.73, 27.45, 27.16, 227.00, 42.48, 302.45, 185.14, 344.71, 204.83, 108.78, 344.40</td>
</tr>
<tr>
<td>39</td>
<td>0.95</td>
<td>39.36, 90.73, 83.15, 186.46, 246.21, 230.04, 217.97, 269.86, 173.34, 233.32, 192.78, 308.41, 100.50, 196.44, 160.18, 327.80, 351.55, 180.86, 191.47, 358.49, 310.55, 218.60, 202.53, 77.73, 182.71, 1.76, 41.73, 260.78, 280.77, 129.13, 276.37, 132.96, 27.83, 252.61, 107.29, 331.78</td>
</tr>
<tr>
<td>40</td>
<td>0.94</td>
<td>28.66, 24.61, 46.23, 58.97, 95.28, 76.31, 337.40, 310.91, 258.76, 219.91, 49.64, 349.15, 197.79, 163.95, 216.11, 88.58, 44.59, 234.28, 23.36, 356.16, 201.31, 173.80, 252.90, 323.46, 104.99, 101.31, 318.13, 154.56, 259.66, 340.54, 85.72, 206.06, 66.94, 271.36, 80.67, 244.81, 34.78, 195.49</td>
</tr>
<tr>
<td>41</td>
<td>0.96</td>
<td>53.61, 63.59, 29.40, 3.64, 3.67, 24.31, 97.10, 114.90, 222.94, 238.18, 334.92, 304.48, 206.26, 143.67, 85.22, 80.54, 2.52, 260.82, 159.64, 309.06, 305.94, 123.34, 150.84, 297.45, 91.30, 82.12, 238.93, 349.39, 208.29, 70.77, 313.76, 167.10, 317.19, 92.70, 218.75, 23.80, 189.12, 357.01, 162.08</td>
</tr>
<tr>
<td>42</td>
<td>0.98</td>
<td>9.25, 39.87, 37.38, 136.87, 176.61, 191.79, 228.64, 85.22, 82.51, 113.02, 137.51, 57.10, 333.65, 70.13, 160.69, 364.86, 182.75, 360.18, 87.82, 232.68, 46.84, 358.94, 299.83, 212.66, 181.03, 31.53, 56.10, 278.54, 254.91, 30.65, 172.09, 256.54, 83.35, 26.66, 213.41, 372.71, 223.50, 100.92, 311.27, 161.96</td>
</tr>
<tr>
<td>43</td>
<td>0.97</td>
<td>47.01, 73.36, 91.12, 60.44, 51.29, 68.66, 91.88, 190.03, 246.45, 348.16, 17.39, 76.59, 328.08, 249.54, 161.54, 89.18, 261.26, 345.80, 359.22, 212.45, 176.39, 15.34, 172.90, 210.41, 75.23, 80.12, 1.44, 291.79, 72.16, 254.21, 310.20, 160.38, 282.71, 102.22, 284.57, 149.04, 61.46, 249.48, 45.27, 215.80, 37.36</td>
</tr>
<tr>
<td>44</td>
<td>0.97</td>
<td>30.29, 9.38, 42.55, 57.42, 313.42, 307.10, 351.41, 283.15, 48.75, 7.98, 190.94, 118.23, 120.89, 164.72, 313.19, 253.16, 133.43, 355.46, 46.04, 280.15, 247.72, 331.73, 137.00, 67.75, 157.72, 82.76, 298.75, 316.97, 191.27, 95.02, 233.35, 248.21, 50.51, 59.16, 221.70, 353.62, 92.75, 284.48, 137.54, 257.20, 38.80, 165.29</td>
</tr>
<tr>
<td>45</td>
<td>0.95</td>
<td>28.88, 5.41, 305.70, 287.05, 307.70, 255.71, 229.28, 297.61, 354.33, 42.59, 42.95, 64.81, 109.31, 221.97, 258.33, 31.78, 141.23, 152.19, 353.50, 291.18, 174.43, 319.14, 244.62, 121.78, 53.24, 350.51, 243.66, 248.42, 104.62, 41.61, 229.91, 378.89, 215.27, 68.77, 296.56, 129.20, 275.14, 51.12, 175.68, 320.13, 105.83, 262.13, 79.18</td>
</tr>
</tbody>
</table>

*aThe first two phase elements in each code are 0 and are excluded*
In many applications, the phases are restricted to values that are the \( k \)th roots of unity (e.g., \( k = 2 \) gives the original Barker codes or \( k = 6 \) for sextic Barker codes). If such restrictions are made on the values of the generalized Barker sequence, there seems to be a limit on the maximal length \( M \), depending on \( k \), for such sequences (e.g., \( M \leq 13 \) for \( k = 2 \)). Ein-dor et al. (2002) used statistical methods to show that for high values of \( M \), a polyphase Barker sequence exists for any \( M \) as long as \( k = M \).

Any unrestricted sequence can be approximated by a sequence of the \( k \)th roots of unity in such a way that the generalized Barker sequence condition is maintained for a sufficiently large value of \( k \). Ternary (\( k = 3 \)) Barker codes were shown to exist for \( 2 \leq M \leq 5 \), \( M = 7 \) and 9. Quaternary (\( k = 4 \)) Barker codes (Welti, 1960) were shown to exist for \( 2 \leq M \leq 5 \), \( M = 7 \), 11, 13, and 15. Sextic (\( k = 6 \)) Barker codes were shown to exist for all \( 2 \leq M \leq 13 \) (Golomb and Scholtz, 1965). Sixty-phase generalized Barker codes were shown to exist for all \( 2 \leq M \leq 19 \) (Zhang and Golomb, 1989) and for higher values of \( M \) up to \( M = 37 \) (Brenner, 1998). Finally, polyphase Barker codes with small alphabets (up to 120 phases) were shown to exist up to \( M = 45 \) (Brenner, 1998).

Figure 6.5 shows the autocorrelation function magnitude for the 15-element polyphase Barker sequence. Note that the peak sidelobe (\( = 1 \)) is obtained at \( \tau = \pm 14t_b \). The magnitudes of all other sidelobes are less than 1. Note also that the autocorrelation function in the polyphase case is complex valued.

### 6.2 CHIRPLIKE PHASE CODES

One of the main drawbacks of the codes discussed in Section 6.1 is their Doppler tolerance. The polyphase Barker codes were derived using numerical methods that optimized only the correlation function sidelobes. Once the target return is Doppler shifted, the expected sidelobes (and interference level) are much higher than those predicted from observing only the zero-Doppler cut of the ambiguity function (the autocorrelation magnitude). For example, the ambiguity function of
a Barker code with length 13 is shown in Fig. 6.6. A contour line marks the $\frac{1}{13}$ level (except for zero Doppler). Because of the random nature of the polyphase codes, the ambiguity function approaches a thumbtack shape.

Recall that the frequency-modulated signals described in Chapter 5 were shown to yield an ambiguity function exhibiting a ridge passing through the origin. Similar results are obtained for phase-coded pulses that have regularities similar to the frequency-modulated pulses of Chapter 5. In the following sections we describe some phase codes derived from the phase history of frequency-modulated pulses. The Frank code (Frank and Zadoff, 1962, 1963; Frank, 1963a,b) is derived from the phase history of a linearly frequency stepped pulse. The main drawback of the Frank code is that it only applies for codes of perfect square length ($M = L^2$). Frank and Zadoff first reported chirplike codes for any length $M$ in a U.S. patent filed in 1957 (Frank and Zadoff, 1963). The patent specifically excludes claiming codes of perfect square length, which were described in a Frank patent filed earlier that year. Although Frank and Zadoff reported the chirplike code of square length in 1962 (now well known as Frank code), its application to any length was not reported at that time. At approximately the same time, similar codes of perfect square length were reported by Heimiller (1961, 1962), who constrained the sequence length to the square of a prime.

Twenty years later, apparently motivated by the work of Schroeder (1970) showing the favorable aperiodic properties of codes with quadratic phase
dependence, chirplike sequences were reinvented by Chu in 1972 (Chu 1972, Frank 1973). Originally, the Frank and Zadoff–Chu codes were identified for their ideal periodic autocorrelation function. Later, the chirplike codes were also identified for their aperiodic properties (i.e., aperiodic correlation sidelobe level, bandwidth precompression limitations, etc.). Lewis and Kretschmer (1981a,b, 1982) and Kretschmer and Lewis (1983) reported specific versions of chirplike codes with good aperiodic properties. The specific versions were named P1, P2, P3, and P4. The P1 and P2 sequences are permutations of the Frank code and are applicable only for $M = L^2$. Several authors (e.g., Antweiler and Bomer, 1990; Zhang and Golomb, 1993; Wai Ho Mow and Li, 1997) studied the different variants of the Frank and Zadoff–Chu code. It was shown that the minimum aperiodic autocorrelation peak sidelobe and minimum integrated sidelobe codes are the code versions originally defined by Frank and Chu.

Other chirplike codes of length $L^2$ with good aperiodic properties (but not ideal periodic autocorrelation) were also reported by Rapajic (1998). Lewis and Kretschmer also report a palindromic variant of the P4 code (applicable to any length $M$) with better aperiodic properties (although not perfect).

### 6.2.1 Frank Code

There are several ways to describe the elements of the Frank code. For mathematical convenience, we define the elements of the original Frank code $s_m$ ($1 \leq m \leq M$) of a square length $M = L^2$ as $s_{(n-1)L+k} = \exp(j\phi_{n,k})$, for $1 \leq n \leq L$ and $1 \leq k \leq L$, where $\phi_{n,k} = 2\pi(n-1)(k-1)/L$. Frank originally expressed the values of the code using the elements of an $L \times L$ discrete Fourier transform matrix given explicitly by

$$
\begin{bmatrix}
0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 2 & \cdots & L - 1 \\
0 & 2 & 4 & \cdots & 2(L - 1) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & L - 1 & 2(L - 1) & \cdots & (L - 1)^2
\end{bmatrix}
$$

The construction method is demonstrated for $M = 16$ ($L = 4$). To calculate the phase values of the 16-element Frank code, we first write the $4 \times 4$ Frank matrix given by

$$
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 2 & 3 \\
0 & 2 & 4 & 6 \\
0 & 3 & 6 & 9
\end{bmatrix}
$$

The 16-element Frank code is formed by concatenating the rows of the Frank matrix and multiplying by $2\pi/L = 2\pi/4 = \pi/2$, resulting in the 16-element
phase code given by

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & \frac{\pi}{2} & \pi & \frac{3\pi}{2} & 0 & \pi & 2\pi & 3\pi & 0 & \frac{3\pi}{2} & 3\pi & \frac{9\pi}{2}
\end{bmatrix}
\]

Taking the phase value modulo \(2\pi\) gives

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & \frac{\pi}{2} & \pi & \frac{3\pi}{2} & 0 & \pi & 0 & 0 & \frac{3\pi}{2} & \pi & \frac{\pi}{2}
\end{bmatrix}
\]

The aperiodic autocorrelation function of the 16-element Frank code is shown in Fig. 6.7. Note that the correlation function is zero for displacements of multiples of \(L\) since the rows of the Frank matrix are orthogonal. Note also that the autocorrelation has a magnitude of unity for a displacement of one more or one less than multiples of \(L\). The original Frank code has two important properties: (1) the code is perfect (see Box 6C); (2) the aperiodic autocorrelation exhibits relatively low sidelobes (as can also be observed from Fig. 6.7).

Frank conjectured (1963a,b) that using cyclically shifted or decimated versions will not yield lower aperiodic autocorrelation sidelobes than the ones obtained using the original Frank code. This was also verified using a computer search (Antweiler and Bomer 1990, Popovic et al. 1991). Wai Ho Moe and Li (1997) use simple trigonometric identities to express the peak-to-peak sidelobe ratio of the Frank code as \(\sin(\pi/L)^{-1}\) for \(L\) even and \(2\sin(\pi/2L)^{-1}\) for \(L\) odd. Figure 6.8 shows the peak sidelobe level of the Frank code normalized by \(L\). Note that it is given by two smooth monotonically decreasing curves, depending on whether \(L\) is even or odd, converging toward their common asymptote \(1/\pi\) from above. Note also that the autocorrelation function peak for \(L\) even is higher than for \(L\) odd.

The connection between the Frank code and a frequency stepped pulse is demonstrated in Fig. 6.9, showing the phase history of a 16-element Frank code. Note that the code is made from four equal segments of linear phase dependence.
BOX 6C: Perfectness of the Frank Code

The Frank code of length $M = L^2$ is written as

$$s_{(n-1)L+k} = \exp[j2\pi(n - 1)(k - 1)/L],$$

for $1 \leq n \leq L$ and $1 \leq k \leq L$, where $n$ is the row index and $k$ is the column index of the Frank matrix. Since adding integer multiples of $L$ to either $k$ or $n$ has no effect on the phase value (modulo $2\pi$), the periodic autocorrelation of the Frank code for a delay corresponding to a shift of $0 \leq m < L$ rows and $0 \leq l < L$ columns is given by

$$\tilde{R}[Ll + m] = \sum_{n=1}^{L} \sum_{k=1}^{L} \exp[j2\pi(n - 1)(k - 1)/L] \times \exp[-j2\pi(n + m - 1)(k + l - 1)/L]$$

which can be rearranged to give

$$\tilde{R}[Ll + m] = \exp[j2\pi(m - ml + l)/L] \sum_{n=1}^{L} \exp(-j2\pi nl/L) \times \sum_{k=1}^{L} \exp(-j2\pi mk/L)$$

Using the closed-form expressions for the sum of a geometric progression, the sum over $n$ can be expressed by

$$\sum_{n=1}^{L} \exp(-j2\pi nl/L) = \exp(-j2\pi l/L) \frac{1 - \exp(-j2\pi l)}{1 - \exp(-j2\pi l/L)}$$

The sum equals zero for all $0 < l < L$ and equals $L$ for $l = 0$. In the same way we can show that the second sum is zero for all $0 < k < L$ and is equal to $L$ for $k = 0$. Thus, the autocorrelation will take nonzero values only when both $l = 0$ and $k = 0$. In this case the autocorrelation value is $L^2 = M$.

(constant frequency). The segments’ phase slope changes linearly from segment to segment implying linear frequency stepping between segments. If the frequency of the first section is zero, the frequency in the $L$th section is $1/(Lt_b) = L/T$. Note that an artificial phase (integer multiple of $2\pi$) was added to some of the sections for the plot. Note also that the first phase in each section is obtained by a linear prediction based on the phase values of the previous section. Finally, note that the resolution of phase values used for the Frank code is $2\pi/L$ compared to the resolution of $\pi$ for the binary codes presented in Section 6.1.
The partial ambiguity function of the 16-element Frank code is shown in Fig. 6.10. Note the diagonal ridge passing through the origin and extending to the first quadrant of the range–Doppler plane. Note also a secondary attenuated ridge duplicated $1/T$ in Doppler. Using decimated versions of the original Frank code yields a different slope of the diagonal ridge.

### 6.2.2 P1, P2, and Px Codes

The P1, P2, and Px codes are all modified versions of the Frank code, with the dc frequency term in the middle of the pulse instead of at the beginning. The Px code was introduced by Rapajic and Kennedy (1998) and was shown to yield the same aperiodic peak sidelobe as the Frank code but having lower integrated sidelobe level. The elements of the Px code are defined mathematically in a similar way to the Frank code as $s_{(n-1)L+k} = \exp(j\phi_{n,k})$, for $1 \leq n \leq L$ and $1 \leq k \leq L$, where in this case the $\phi_{n,k}$ are given by

$$
\phi_{n,k} = \begin{cases} 
\frac{2\pi}{L}[(L + 1)/2 - k][(L + 1)/2 - n], & L \text{ even} \\
\frac{2\pi}{L}[L/2 - k][(L + 1)/2 - n], & L \text{ odd}
\end{cases}
$$

(6.2)
In a similar way to calculating the Frank matrix, we can calculate an analogous \( P_x \) matrix. For \( M = 16 \) \((L = 4)\) the resulting matrix is

\[
\begin{bmatrix}
1 \\
\frac{1}{4} \\
-3 \\
-9
\end{bmatrix}
\begin{bmatrix}
9 & 3 & -3 & -9 \\
3 & 1 & -1 & -3 \\
-3 & -1 & 1 & 3 \\
-9 & -3 & 3 & 9
\end{bmatrix}
\]

The 16-element \( P_x \) code is formed by concatenating the rows of the \( P_x \) matrix and multiplying by \( 2\pi/L = 2\pi/4 = \pi/2 \), resulting in the 16-element phase code given by

\[
\begin{bmatrix}
\frac{9\pi}{8} & \frac{3\pi}{8} & -\frac{3\pi}{8} & -\frac{9\pi}{8} & \frac{3\pi}{8} & \frac{\pi}{8} & -\frac{\pi}{8} & -\frac{3\pi}{8} \\
-\frac{3\pi}{8} & -\frac{\pi}{8} & \frac{\pi}{8} & \frac{3\pi}{8} & -\frac{9\pi}{8} & -\frac{3\pi}{8} & \frac{3\pi}{8} & \frac{9\pi}{8}
\end{bmatrix}
\]

Taking the phase value modulo \( 2\pi \) gives

\[
\begin{bmatrix}
\frac{9\pi}{8} & \frac{3\pi}{8} & \frac{13\pi}{8} & \frac{7\pi}{8} & \frac{3\pi}{8} & \frac{\pi}{8} & \frac{15\pi}{8} & \frac{13\pi}{8} & \frac{13\pi}{8} & \frac{15\pi}{8} & \frac{\pi}{8} \\
\frac{3\pi}{8} & \frac{7\pi}{8} & \frac{13\pi}{8} & \frac{3\pi}{8} & \frac{9\pi}{8}
\end{bmatrix}
\]
The autocorrelation function and phase history of the 16-element Px code are shown in Figs. 6.11 and 6.12, respectively. The phase history of the 25-element Px code is shown in Fig. 6.13. Note that the sidelobes of the 16-element Px correlation function have a concave shape rather than the convex shape of the 16-element Frank code (compare Fig. 6.11 to Fig. 6.7). Although the peak sidelobe
of the Px and Frank codes is the same, the integrated sidelobe of the Px code is superior to that of the Frank code (and to that of other codes discussed in later sections). Note that whereas the Frank code is a perfect code (having an ideal periodic correlation function), the Px code is not perfect.

The P1 and P2 codes are due to Lewis and Kretschmer (1981a,b). The two codes are derived based on the Frank code and are very similar to the Px code. Both codes, as the Frank and Px codes, are applicable only for perfect square length \( M = L^2 \). The P2 code is valid only for \( L \) even and is defined exactly as the Px code for even \( L \). Note that the P2 code (and Px for even \( M \)) is palindromic in that it has conjugate symmetry across each frequency (matrix row) and even symmetry about the center of the code (see Fig. 6.12).

The P1 code elements \( s_m(1 \leq m \leq M) \) are defined as \( s_{(n-1)L+k} = \exp(j\phi_{n,k}) \), where \( 1 \leq n \leq L, 1 \leq k \leq L \), and

\[
\text{P1: } \quad \phi_{n,k} = \frac{2\pi}{L}\left[\frac{(L+1)/2 - n}{(n-1)L + (k-1)}\right]
\]

(6.3)

Figures 6.14 and 6.15 show the 16-element \((L = 4)\) and 25-element \((L = 5)\) P1 code phase history calculated using (6.3). Integer multiples of \(2\pi\) were added.

**FIGURE 6.14** Phase history of a 16-element P1 code.

**FIGURE 6.15** Phase history of a 25-element P1 code.
such that the phase difference between slices is less than $\pi$. Note that the P1 code has the same phase increments within each phase group as the P2/Px code, except that the starting phases are different. Note that while the Px and P2 codes are not perfect, the P1 code is perfect (as is the Frank code). The ambiguity function of the P1 code for odd $L$ is identical to that of the Frank code. For $L$ even the ambiguity function of the P2/Px code is very similar to that of the P1 code and to that of the Frank code.

Bandwidth limitations are found in all radar systems. Such limitations are usually the result of attempts to maximize signal to thermal noise and hardware limitations. The result of bandlimiting is to average the elements constituting the code waveform. For the P1 and P2/Px codes, the phase difference between adjacent code elements is low in the center of the code and higher closer to the ends of the code. Thus, bandlimiting results in amplitude intensifying the central part of the code relative to the code ends. As we already demonstrated in Chapter 4 and will also see in Chapter 7, this results in an increase to mainlobe width and lower sidelobes. The phase increments of the Frank code (taken modulo $2\pi$) are higher in the central part of the code and lower close to the code ends. Thus, bandlimiting results in amplitude intensifying the ends of the code relative to the code center and yields a decrease in mainlobe width and higher sidelobes.

### 6.2.3 Zadoff–Chu Code

While the Frank, P1, P2, and Px codes are only applicable for perfect square lengths ($M = L^2$), the Zadoff code (Zadoff, 1963) is applicable for any length and is given by $s_m = \exp(j\phi_m)$, where

\[
\phi_m = \frac{2\pi}{M}(m - 1) \left( r \frac{M - 1 - m}{2} - q \right)
\]

\[1 \leq m \leq M, \quad 0 \leq q < M\] is any integer, and $r$ is any integer relatively prime to $M$.

For any given length $M$, different variants of the Zadoff code are obtained by changing $r$ or $q$ and adding a constant phase shift to all elements. It can be shown that the permutations that preserve the ideal periodic autocorrelation function property (cyclic shift, decimation, conjugation, etc.) are equivalent to changing $q$ and $r$. An important permutation of the Zadoff code was presented by Chu (1972) and is given by $s_m = \exp(j\phi_m)$, where

\[
\phi_m = \begin{cases} 
\frac{2\pi}{M}r'(m - 1)^2/2, & M \text{ even} \\
\frac{2\pi}{M}r'(m - 1)m/2, & M \text{ odd}
\end{cases}
\]

\[1 \leq m \leq M, \quad r' \text{ is any integer relatively prime to } M.\] The Chu code can be obtained from the Zadoff code by setting $r = -r'$ and $q = -(M - 2)r'/2$ for
CHIRPLIKE PHASE CODES

$M$ even or by setting $r = -r'$ and $q = -(M - 1)r'/2$ for $M$ odd. We calculate the Zadoff–Chu phase code of length $M = 16$ and minimal chirp slope by using $r' = 1$ in (6.5) or $r = -1$ and $q = 9$ in (6.4). Figure 6.16 shows the phase values as a function of $m$. The autocorrelation function of the minimal chirp slope 16-element Zadoff–Chu phase-coded pulse is shown in Fig. 6.17.

Taking the phase values modulo $2\pi$ gives

$$
\begin{bmatrix}
0 & \frac{3\pi}{16} & \frac{\pi}{2} & \frac{15\pi}{16} & \frac{3\pi}{16} & \frac{\pi}{2} & \frac{31\pi}{16} & \frac{\pi}{2} & \frac{3\pi}{16} & \frac{3\pi}{2} & \frac{15\pi}{16} \\
\frac{\pi}{2} & \frac{3\pi}{16} & 0 & \frac{31\pi}{16}
\end{bmatrix}
$$

Note that taking the phase value modulo $2\pi$ shows close to palindromic phase behavior. Note also that many authors refer to the minimal chirp slope Zadoff–Chu code simply as Chu code. In the Chu code the phase is always zero for $m = 1$ and quadratic increasing or decreasing for higher $m$ (as for the Frank code), while for the Zadoff code, choosing $q = 0$ gives a phase shape closer to that of the Px/P2 or P1 codes having a minimum or maximum phase in the center of the code.

**FIGURE 6.16** Phase history of a 16-element Zadoff–Chu code.

**FIGURE 6.17** Autocorrelation function of a minimal chirp slope 16-element Zadoff–Chu coded pulse.
Like the Frank and P1 codes, the Zadoff–Chu code is perfect (see Box 6D). Antweiler and Bomer (1990) used an exhaustive computer search to show that considering all possible decimations (equivalent to changing the parameters of the Zadoff code), the minimal chirp slope Chu code obtained by setting $|r| = 1$ in (6.5) has the lowest aperiodic autocorrelation peak sidelobe. Note that all cyclic versions of the Zadoff–Chu code have the same aperiodic correlation magnitude (see Box 6E).

**BOX 6D: Perfectness of the Zadoff–Chu Code**

We assume that $s_m = \exp(j\phi_m)$, where $1 \leq m \leq M$ and where $\phi_m$ is given by (6.4). The condition for the code to be perfect is equivalent to showing that the $M - 1$ nonredundant equations

$$
\tilde{R}[k] = \sum_{m=1}^{M} \exp[j(\phi_m - \phi_{m+k \mod M})] = 0 \quad (6D.1)
$$

are satisfied for all $1 \leq k \leq M - 1$. Using (6.4) and simplifying, it is easy to show that

$$
\phi_{m+M} - \phi_m = -\frac{2\pi}{M}qM + \frac{2\pi}{M}rmM \quad (6D.2)
$$

Thus, the set of equations can be rewritten as

$$
\tilde{R}[k] = \sum_{m=1}^{M} \exp[j(\phi_m - \phi_{m+k})] = 0 \quad (6D.3)
$$

Using (6.4) again and simplifying gives

$$
\phi_m - \phi_{m+k} = -\frac{2\pi}{M}qk + \frac{2\pi}{M}k(-M + 2m + k) \quad (6D.4)
$$

$$
= \frac{2\pi}{M}[-rmk + g(k)]
$$

where $g(k)$ is some function of $k$. Substituting (6D.4) in (6D.3) and using the fact that $r$ is relatively prime to $M$ gives

$$
\tilde{R}[k] = \exp\left[\frac{j2\pi}{M}g(k)\right] \sum_{m=1}^{M} \exp\left(-\frac{j2\pi}{M}rmk\right) = 0 \quad (6D.5)
$$

for all $1 \leq k \leq M - 1$. 
BOX 6E: Rotational Invariance of the Zadoff–Chu Code Aperiodic ACF Magnitude

First we write the phase of the Zadoff–Chu code cyclically shifted by \( a \) positions in term of the phase values of the original Zadoff–Chu code:

\[
\phi'_m = \phi_{(m+a)\mod M} = \phi_m + \phi_{(m+a)\mod M} - \phi_m \tag{6E.1}
\]

Using (6D.2) and (6D.4), we can write

\[
\phi'_m = \phi_m - \frac{2\pi}{M}[-rma + g(a)] \tag{6E.2}
\]

Now, writing the aperiodic autocorrelation function (ACF) for positive delay \( k \) gives

\[
R_a[k] = \sum_{m=1}^{M-k} \exp[j(\phi'_m - \phi'_{m+k})] = \sum_{m=1}^{M-k} \exp \left[ j \left( \phi_m - \frac{2\pi}{M}[-rma + g(a)] - \phi_{m+k} + \frac{2\pi}{M}[-r(m+k)a + g(a)] \right) \right] = \sum_{m=1}^{M-k} \exp \left[ j \left( \phi_m - \phi_{m+k} - \frac{2\pi}{M}rka \right) \right] \tag{6E.3}
\]

\[
= \exp \left( -j \frac{2\pi}{M}rka \right) \sum_{m=1}^{M-k} \exp[j(\phi_m - \phi_{m+k})] = \exp \left( -j \frac{2\pi}{M}rka \right) R[k]
\]

Thus, \( |R_a[k]| = |R[k]| \) for positive delay \( k \). Since the ACF is conjugate symmetric around zero delay, \( |R_a[k]| = |R[k]| \) for all positive and negative delays.

The autocorrelation sidelobe pattern of the Zadoff–Chu code shown in Fig. 6.17 is typical of all Zadoff–Chu codes with \( |r| = 1 \) (higher sidelobs close to the mainlobe and close to \( \pm T \)). When using different values of \( r \) (different chirp slopes), the area of higher aperiodic autocorrelation sidelobes “moves” on the delay axis to different locations. The position of the higher sidelobes can
intuitively be predicted by examining the ambiguity function of the Zadoff–Chu phase-coded pulse. As can be seen from the example in Fig. 6.18, the higher sidelobes are obtained in the area where the secondary diagonal ridges, centered on $\pm n/T$ in Doppler ($n > 0$), cross the zero-Doppler axis.

Wai Ho Moe and Li (1997) show that the peak sidelobe of the aperiodic autocorrelation of the Chu code is given by $\left| \sin \left( \pi \frac{\sqrt{\alpha M}}{\pi} \right)^2 / \sin \left( \pi \frac{\sqrt{\alpha M}}{\pi} \right) \right|$, where $\alpha = 1.16556118520721 \ldots$ is the first positive root of the equation $\tan(\alpha) = 2\alpha$ and $\lfloor \cdot \rfloor$ stands for rounding toward the nearest integer. The expression is valid for all $M \geq 2$ except for $M = 33$, where $\sqrt{\alpha 33/\pi} \approx 3.499045$ and the sidelobe level is given by $\left| \sin \left( \pi \frac{4^2}{M} \right) / \sin(\pi 4/M) \right|$. Figure 6.19 shows the peak sidelobe level of the Zadoff–Chu code as a function of the sequence length normalized to the square root of the sequence length. Note that the sidelobe level is higher than that for the Frank code (compare to Fig. 6.8). Asymptotically, the peak sidelobe converges to $\sin \alpha / \sqrt{\pi \alpha} \approx 0.480261$.

### 6.2.4 P3, P4, and Golomb Polyphase Codes

Analogously to the connection between P1 and P2/Px codes to the original Frank code, the P3, P4, and Golomb polyphase codes are specific cyclically shifted and decimated versions of the Zadoff–Chu code. Like the P1 and P2 codes, the P3 and P4 codes, are due to Lewis and Kretschmer (1982). Unlike the P1 and P2 codes, the P3 and P4 codes, are due to Lewis and Kretschmer (1982). Unlike the P1 and P2
codes applicable only for square length (such as the Frank code), the P3 and P4 codes are, like the Zadoff–Chu code, applicable for any length $M$. The P3 and P4 codes are defined for any length $M$ by

$$P3 : \quad \phi_m = \frac{2\pi (m - 1)^2}{M^2}$$  \hspace{1cm} (6.6)$$

$$P4 : \quad \phi_m = \frac{2\pi}{M} (m - 1) \left( \frac{m - 1 - M}{2} \right)$$  \hspace{1cm} (6.7)$$
where $1 \leq m \leq M$. The P3 code is identical to the Chu code for even $M$ with $r' = 1$ and is perfect only for even values of $M$. The P4 code can be obtained by setting $r = -1$ and $q = 1$ in the expression for the Zadoff phase code. Like the Zadoff code, the P4 code has an ideal periodic correlation for both even and odd $M$.

A slightly different version of the Zadoff–Chu/P4 codes was defined by Zhang and Golomb (1993) and is given by

$$\phi_m = \frac{2\pi}{M} r'' (m - 1)m \quad (6.8)$$

where $1 \leq m \leq M$, and $r''$ is any integer relatively prime to $M$. Note that the Golomb polyphase sequences are identical to the Chu code for odd $M$. Zhang and Golomb showed that the polyphase Golomb code is perfect for all $M$.

When the target Doppler is unknown, using highly Doppler-tolerant FM waveforms allows for simplified receiver hardware with negligible degradation in performance. Kretschmer and Lewis (1983) showed that the P3 and P4 codes are much more Doppler tolerant than the Frank or P1 and P2/Px codes. This result is very intuitive and can be extended to the other Zadoff–Chu variants, based on sampling the phase history of a continuous frequency-modulated waveform other than stepped frequency-modulated waveforms.

The P4 code differs from the P3 code by having the largest code element to code element phase changes at the ends of the code instead of the middle, as in the P3 code. In this way, the P4 code differs from the P3 code in the same way that the P1 code differs from the Frank code, implying that the bandwidth tolerance of the P4 is better than the bandwidth tolerance of the P3 code. A palindromic P4 version having optimized bandwidth tolerance was defined by Kretschmer and Lewis (1983) and is given by

$$\phi_m = \frac{\pi}{M} \left( m - \frac{1}{2} \right)^2 - \pi \left( m - \frac{1}{2} \right) \quad (6.9)$$

where $1 \leq m \leq M$. Note that the palindromic version of P4 (as the Px and P2 codes) is not a perfect code.

### 6.2.5 Phase Codes Based on a Nonlinear FM Pulse

The phase codes discussed in previous sections were shown to be strongly connected to the phase history of a linear frequency-modulated or frequency-stepped pulse. Other phase codes can be derived by sampling the phase history of a nonlinear frequency-modulated (or frequency-stepped) pulse. Selection of the phase (or frequency) history of the continuous pulse determines the properties (side-lobe level, mainlobe width, Doppler tolerance, bandwidth limitations, etc.) of the resulting phase code. As shown in Chapter 5, using nonlinear instead of linear FM can yield lower autocorrelation sidelobes but a higher mainlobe width.
One example of a phase code derived from the phase history of a frequency-modulated pulse is the \( P(n, k) \) code (Felhauer, 1994). The \( P(n, k) \) code is based on step approximation of the phase function of a nonlinear FM chirp signal with the energy density function given by

\[
W(f) = \begin{cases} 
  k + (1 + k) \cos^n \left( \frac{\pi f}{B} \right), & |f| \leq \frac{B}{2} \\
  0, & |f| > \frac{B}{2}
\end{cases} \quad (6.10)
\]

where \( k \) and \( n \) are free parameters and \( B \) is the bandwidth considered for the expanded pulse. Generation of the phase code first involves designing the continuous pulse yielding the desired energy density function with parameters \( n \) and \( k \). The calculation is based on the principle of stationary phase (see Chapter 5). Next, the phase history of the continuous pulse is sampled, resulting in the phase code desired.

Figure 6.20 illustrates generation of the \( P(n, k) \) code. The upper parts of Fig. 6.20 show a typical weighting function \( W(f) \) and the corresponding instantaneous frequency function \( f(t) \) calculated using the principle of stationary phase.
The lower part of Fig. 6.20 illustrates generation of the P(n, k) code by step approximation of the phase function calculated from the instantaneous frequency \( f(t) \). For \( k = 1 \), \( W(f) \) is a constant rectangular spectrum, and the algorithm described here leads to the linear chirp signal with a corresponding phase function identical to the phase function of the P4 code. For arbitrary values of \( n \) and \( k \), the corresponding phase code elements can only be calculated using numerical methods.

The P(n, k) code of length N can be optimized for minimal peak sidelobe by controlling the free parameters \( n \) and \( k \). For example, when \( N = 100 \), the optimal values of \( k \) are 0.05 and 0.015 for \( n = 2 \) and 4, yielding a peak sidelobe level of 34.8 and 37.1 dB, respectively. The autocorrelation function of the \( P(2, 0.05) \) and \( P(4, 0.015) \) codes are shown in Fig. 6.21 together with the P4 ACF. Note the peak sidelobe of the \( P(4, 0.015) \) is only 2.9 dB lower than the theoretical optimum for uniform phase codes and more than 10 dB better than the peak sidelobe of the P4 code. The price for this significant peak sidelobe improvement is a loss in range resolution in comparison with the P4 code, as seen by the widening of the ACF mainlobe [The mainlobe width to the chip length ratio is 1.8 for \( P(2, 0.05) \), 2.8 for \( P(4, 0.015) \) and 1 for P4.]

Table 6.5 compares the peak sidelobe and mainlobe width of the P(n, \( k_{opt} \)) codes for 1 \( \leq n \leq 5 \), the Frank/P1/Px codes, and the P3/P4/Zadoff–Chu codes.

**FIGURE 6.21** Autocorrelation function of the \( P(4, k_{opt}) \), \( P(2, k_{opt}) \), and P4 codes.
TABLE 6.5  Optimal Value $k_{opt}$, Peak Sidelobe Ratio, and Mainlobe Width for Some P$(n,k)$ Codes

<table>
<thead>
<tr>
<th>Phase Code</th>
<th>$k_{opt}$</th>
<th>Peak Sidelobe Width $\times t_b$</th>
<th>$k_{opt}$</th>
<th>Peak Sidelobe Width $\times t_b$</th>
<th>$k_{opt}$</th>
<th>Peak Sidelobe Width $\times t_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(1, k_{opt})$</td>
<td>0.0055</td>
<td>22.7</td>
<td>0.006</td>
<td>30.7</td>
<td>0.01</td>
<td>33</td>
</tr>
<tr>
<td>$P(2, k_{opt})$</td>
<td>0.12</td>
<td>23.5</td>
<td>0.039</td>
<td>31.9</td>
<td>0.05</td>
<td>34.8</td>
</tr>
<tr>
<td>$P(3, k_{opt})$</td>
<td>0.085</td>
<td>24.1</td>
<td>0.015</td>
<td>33</td>
<td>0.03</td>
<td>35.9</td>
</tr>
<tr>
<td>$P(4, k_{opt})$</td>
<td>0.11</td>
<td>24.1</td>
<td>0.025</td>
<td>34.2</td>
<td>0.015</td>
<td>37.1</td>
</tr>
<tr>
<td>$P(5, k_{opt})$</td>
<td>0.045</td>
<td>24.1</td>
<td>0.024</td>
<td>35.1</td>
<td>0.015</td>
<td>38.1</td>
</tr>
<tr>
<td>Frank/P1/Px</td>
<td>—</td>
<td>21.9</td>
<td>—</td>
<td>28</td>
<td>—</td>
<td>29.9</td>
</tr>
<tr>
<td>P3/P4/Chu</td>
<td>—</td>
<td>18.7</td>
<td>—</td>
<td>24.3</td>
<td>—</td>
<td>26.2</td>
</tr>
</tbody>
</table>

for $N = 16, 64, and 100$. Note that for suboptimal values of $k$ the mainlobe width loss decreases with increasing $k$ until for $k = 1$ there is no increase in mainlobe width. Note also that as $n$ increases, $k_{opt}$ is lower and the mainlobe width loss is higher. Finally, note that for lower values of $N$, the peak/peak sidelobe ratio approaches $N$ (as for polyphase Barker codes). For example, Felhauer (1994) showed that for $P(4,k_{opt})$ the peak-sidelobe level is 1 for all $N \leq 25$.

The partial ambiguity function of a 16-element $P(4,0.11)$ is shown in Fig. 6.22. Note that the Doppler tolerance is high (as expected). Felhauer (1994) showed
that $P(n, k)$ is less affected by bandwidth limitation than even the P4 code. This is because the phase increments from code element to code element are smaller, compared to other phase codes, over most of the code length $N$. The result is further clarified by inspecting the Fourier spectra of the phase-coded pulses. Because of the rectangular chip of duration $t_b$, all phase-coded pulses have $\sin(\pi f t_b)/\pi f t_b$ baseband spectrum envelopes. The choice of the modulating phase code determines the fine structure in the baseband spectrum.

Figure 6.23 shows the partial baseband spectrum of a 100-element P4 coded pulse, 100-element Frank code baseband spectrum, and the 100-element $P(4,0.015)$-coded pulse spectrum. The figure illustrates that the $P(4,0.015)$ has a much smaller effective bandwidth, which on the one hand, is the reason for the higher mainlobe width, and on the other hand, is the explanation for the improved tolerance to bandwidth limitations. Finally, note that other phase codes can be derived using a different energy density function than the one described in (6.10), yielding different ACF and frequency spectrum properties.

### 6.3 Asymptotically Perfect Codes

The Frank and Zadoff–Chu codes discussed in previous sections are examples of perfect codes having zero periodic autocorrelation function for all nonzero
ASYMPTOTICALLY PERFECT CODES

Perfect codes are useful, for example, in radar applications employing CW waveforms. Asymptotically perfect codes have the property that the peak-to-peak PACF sidelobe goes to zero as the code length $M$ increases. The main drawback of Frank and Zadoff–Chu codes is that they are polyphase. The asymptotically perfect codes described in this section are binary.

A prime example of such sequences are maximum-length linear feedback sequences, or $m$-sequences, having a two-valued PACF (i.e., the PACF magnitude is 1 for all nonzero shifts and $M$ at integer multiples of the code period). All $m$-sequences are limited to length $M = 2^n - 1$ and are derived mathematically based on irreducible polynomials over GF($2^n$) (see Appendix 6A). Figure 6.24 shows the shift register that generates an $m$-sequence of period $2^3 - 1 = 7$. The PACF magnitude of the seven-element $m$-sequence is shown in Table 6.6.

The peak sidelobe of the aperiodic autocorrelation of $m$-sequence codes is usually suboptimal (e.g., for $M = 63$ we can find an $m$-sequence with a peak sidelobe of 6, whereas the MPS code has a peak sidelobe of 4). However, since the PACF sidelobe of the $m$-sequence is low, it is usually the case that the aperiodic autocorrelation sidelobe level is not high. Note that for different cyclic shifts of the $m$-sequence we get different ACF sidelobes. Thus, for each irreducible polynomial, it is necessary to check $M$ different cyclic shifts of the periodic code to find the optimal aperiodic code. Recall that for large values of $M$ this

![Figure 6.24 Generating the m-sequence of period 2^3 - 1 = 7 using a shift register.](image)

<table>
<thead>
<tr>
<th>${s_n^*}$</th>
<th>1 1 1 -1 -1 1 -1</th>
</tr>
</thead>
<tbody>
<tr>
<td>${s_m}$</td>
<td>1 1 -1 -1 1 -1 -1</td>
</tr>
<tr>
<td></td>
<td>1 1 1 1 -1 -1 -1</td>
</tr>
<tr>
<td></td>
<td>1 -1 -1 1 -1 -1 -1</td>
</tr>
<tr>
<td></td>
<td>-1 1 -1 1 -1 -1 -1</td>
</tr>
<tr>
<td></td>
<td>-1 1 1 1 1 1 -1</td>
</tr>
<tr>
<td></td>
<td>1 -1 -1 -1 -1 -1 -1</td>
</tr>
<tr>
<td></td>
<td>-1 -1 -1 -1 -1 -1 -1</td>
</tr>
<tr>
<td>${\tilde{R}}$</td>
<td>-1 -1 -1 -1 -1 -1 -1</td>
</tr>
</tbody>
</table>

### TABLE 6.6 Calculation of the PACF for the Seven-Element m-Sequence of Fig. 6.24

- $a_k = a_{k-2} \oplus a_{k-3}$
- $a_{k-1} \oplus a_{k-2} \oplus a_{k-3}$
- $\ldots 11100101110010111 \ldots$
is significantly less than searching over $2^M$ possible binary codes (e.g., when $M = 63$ there are six irreducible polynomials).

Two-valued asymptotically perfect sequences correspond to combinatorial objects known as cyclic Hadamard difference sets (CDSs). Known constructions of CDS are the $m$-sequence, Legendre sequences, Hall’s sextic residue sequence, twin prime sequence, GMW sequence, and the Maschietti construction (Maschietti, 1998). A cyclic $\{M, l, \lambda\}$ difference set is a set $D$ of $l$ elements, distinct modulo $M$, such that the congruence $d_i - d_j \equiv r \pmod{M}$ has exactly $\lambda$ solution pairs $(d_i, d_j)$ with $d_i$ and $d_j$ in $D$ for each $1 \leq r \leq M - 1$. A simple example of cyclic difference set is given by $D = \{1, 2, 4\}$, which is a cyclic $\{7, 3, 1\}$ difference set. It is easy to verify that the periodic sequence with period $M$ obtained by setting the sequence element to 1 where the element index is in $D \pmod{M}$ and to $-1$ where it is not in $D$ and has a two-valued periodic autocorrelation function given by

$$R[k] = \sum_{m=1}^{M} s_m s_{m+k}^\ast = \begin{cases} M, & k = 0 \pmod{M} \\ M - 4(l - \lambda), & k \neq 0 \pmod{M} \end{cases} \quad (6.11)$$

For example, $D = \{1, 2, 4\}$ implies $\{1, 1, -1, 1, -1, -1, -1\}$, which is the $m$-sequence generated by the shift register in Fig. 6.24.

### 6.4 GOLOMB’S CODES WITH IDEAL PERIODIC CORRELATION

A method for generating two-valued (biphase) perfect codes using cyclic difference sets was described by Golomb (1992). In general, the out-of-phase value $M - 4(l - \lambda)$ in (6.11) will not be zero for $M > 4$. However, if we replace the $-1$ elements in $\{s_m\}$ by a suitable complex number $\beta$, it is possible to obtain zero PACF sidelobes.

First we consider the case where $|\beta| = 1$ (i.e., we replace the $180^\circ$ phase shifts between the two types of elements in the asymptotically perfect sequence described in Section 6.3 with a different phase). In this case it is possible to show (see Box 6F) that if we set $\beta = \exp(j\phi)$, where

$$\cos \phi = -\frac{M - 2l + 2\lambda}{2(l - \lambda)} \quad (6.12)$$

the resulting biphase code is perfect. Note that (6.12) does not have a solution $\phi$ for all CDS parameters. However, for the family of Hadamard cyclic difference sets satisfying $M = 4n - 1$, $l = 2n - 1$, and $\lambda = n - 1$, we get that

$$\cos \phi = -\frac{M - 1}{M + 1} \quad (6.13)$$

which always has a solution. Table 6.7 gives some Hadamard cyclic difference sets and the corresponding phase value used according to (6.13) to construct a Golomb biphase perfect code.
TABLE 6.7 Parameters of Some Nonequivalent Golomb Biphase Perfect Codes

<table>
<thead>
<tr>
<th>M</th>
<th>Shift</th>
<th>λ</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>90</td>
<td>0</td>
<td>{1}</td>
</tr>
<tr>
<td>3</td>
<td>120</td>
<td>1</td>
<td>{1, 2}</td>
</tr>
<tr>
<td>4</td>
<td>180</td>
<td>2</td>
<td>{1, 2, 4}</td>
</tr>
<tr>
<td>7</td>
<td>138.6</td>
<td>2</td>
<td>{1, 2, 3, 5}</td>
</tr>
<tr>
<td>11</td>
<td>146.4</td>
<td>3</td>
<td>{1, 2, 3, 5, 6, 8}</td>
</tr>
<tr>
<td>15</td>
<td>151.0</td>
<td>4</td>
<td>{1, 2, 3, 5, 7, 9, 10, 11, 12} or {1, 4, 5, 7, 9, 10, 11, 12}</td>
</tr>
<tr>
<td>19</td>
<td>154.2</td>
<td>5</td>
<td>{1, 2, 3, 4, 6, 8, 13, 14, 16, 17}</td>
</tr>
<tr>
<td>23</td>
<td>156.4</td>
<td>6</td>
<td>{1, 2, 3, 4, 5, 7, 9, 10, 13, 14, 17, 19}</td>
</tr>
<tr>
<td>31</td>
<td>159.6</td>
<td>7</td>
<td>{1, 2, 3, 4, 6, 8, 12, 15, 16, 17, 23, 24, 27, 29, 30}</td>
</tr>
<tr>
<td>35</td>
<td>160.8</td>
<td>8</td>
<td>{1, 2, 4, 5, 8, 10, 12, 13, 14, 15, 17, 18, 22, 28, 29, 30, 34}</td>
</tr>
<tr>
<td>43</td>
<td>162.7</td>
<td>10</td>
<td>{1, 2, 3, 4, 5, 8, 11, 12, 16, 19, 20, 21, 22, 27, 32, 33, 35, 37, 39, 41, 42}</td>
</tr>
<tr>
<td>63</td>
<td>165.6</td>
<td>15</td>
<td>{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 17, 18, 19, 21, 24, 25, 28, 30, 33, 34, 35, 37, 41, 44, 46, 47, 49, 54, 55, 59}</td>
</tr>
</tbody>
</table>

BOX 6F: Deriving the Perfect Golomb Biphase Code

Assume a two-valued sequence \{s_m\} with period \(M\) obtained by setting the sequence element to 1 where the element index is in the \(\{M, l, \lambda\}\) cyclic difference set \(D (\text{mod} M)\) and to \(\beta = \exp(j\phi)\) where it is not in \(D\). Then when calculating the PACF for each shift \(1 \leq k \leq M - 1\), we have the following four situations:

1 \times 1 \quad \lambda \text{ times}

1 \times \beta^* \quad l - \lambda \text{ times}

\beta \times 1 \quad l - \lambda \text{ times}

\beta \times \beta^* \quad M - 2l + \lambda \text{ times}

We than have

\[ \tilde{R}[k] = \sum_{m=1}^{M} s_ms_{m+k} \]

\[ = \begin{cases} 
  l + (M - l)|\beta|^2 = M, & k = 0 \\
  \lambda + (M - 2l + \lambda)|\beta|^2 + (l - \lambda)(\beta^* + \beta) = M - 2l + 2\lambda + 2(l - \lambda)\cos\phi, & 1 \leq k \leq M - 1 
\end{cases} \]

(6F.1)

Requiring that the code is perfect immediately yields (6.12).
An alternative method also described by Golomb employs both phase and amplitude coding. The idea is to use the original binary sequence in the transmitting end. The receiver reference signal amplitudes are altered from the original value of \(-1\) to \(-b\) such that the periodic cross-correlation is ideal. In this case it is possible to show (see Box 6G) that if

\[
b = -\frac{2\lambda - l}{M - 3l + 2\lambda}
\]  

(6.14)

**BOX 6G: Deriving the Golomb Two-Valued Code with Ideal Periodic Cross-Correlation**

Assume a two-valued sequence \(\{s_m\}\) with period \(M\) obtained by setting the sequence element to 1 where the element index is in the \(\{M, l, \lambda\}\) cyclic difference set \(D \pmod{M}\) and to \(-1\) where it is not in \(D\). Next, assume a two-valued code \(\{q_m\}\) formed by replacing the \(-1\) elements in \(\{s_m\}\) with \(-b\), where \(b > 0\) and real. Then, when calculating the periodic cross-correlation for each shift \(1 \leq k \leq M - 1\), we have the following four situations:

\[
\begin{align*}
1 \times 1 & \quad \lambda \text{ times} \\
1 \times (-b) & \quad l - \lambda \text{ times} \\
-1 \times 1 & \quad l - \lambda \text{ times} \\
-1 \times (-b) & \quad M - 2l + \lambda \text{ times}
\end{align*}
\]

We then have

\[
\tilde{R}_{sq}[k] = \sum_{m=1}^{M} s_m q_{m+k}^* \\
= \begin{cases} 
  l + (M - l)b < M, & k = 0 \\
  \lambda + (M - 2l + \lambda)b - (l - \lambda)(b + 1), & k \neq 0 \pmod{M}
\end{cases} \quad (6G.1)
\]

To have zero periodic cross-correlation sidelobes, we need to have

\[
\lambda + (M - 2l + \lambda)b - (l - \lambda)(b + 1) = 2\lambda - l + (M - 3l + 2\lambda)b = 0 
\]  

(6G.2)

which immediately gives (6.14).

the condition on zero periodic cross-correlation sidelobes is met. Note that using a reference pulse different from the transmitted pulse results in a mismatch loss given by \([l + (M - l)b]/M\). Note that for the Hadamard CDS, shown previously to yield good biphase sequences, we get \(b = \infty\). Golomb showed that projective
TABLE 6.8  Parameters of Some CDS Yielding Real-Valued Codes with Ideal Cross-Correlation

<table>
<thead>
<tr>
<th>M</th>
<th>b</th>
<th>λ</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>3/2</td>
<td>6</td>
<td>{1 2 3 4 5 8 9 11 13}</td>
</tr>
<tr>
<td>21</td>
<td>3/8</td>
<td>1</td>
<td>{1 2 5 15 17}</td>
</tr>
<tr>
<td>31</td>
<td>4/15</td>
<td>1</td>
<td>{1 2 4 9 13 19}</td>
</tr>
<tr>
<td>57</td>
<td>6/35</td>
<td>1</td>
<td>{1 2 4 14 33 37 44 53}</td>
</tr>
</tbody>
</table>

planes CDS yield finite values of \( b \). Table 6.8 gives some finite projective planes cyclic difference sets and the corresponding value of \( b \) used according to (6.14) to construct a Golomb real-valued code with ideal cross-correlation. Finally, note that due to the random nature of Golomb codes, the Doppler tolerance is similar to that of the Barker codes and is much lower than that of Frank/Zadoff–Chu or \( P(n, k) \) codes.

### 6.5 IPATOV CODE

The Golomb codes discussed at the end of Section 6.4 resulted in a mismatch power loss. Ipatov (1992) considered the problem of finding the code pairs with minimal peak response loss. The search is constrained only for those code pairs where the code used at the transmitting end is binary, and the resulting binary code is referred to as an optimal code of length \( M \). We start by observing that for a code to have an ideal periodic correlation function, the DFT of the code elements must be constant. When there is a mismatch between the transmitting and receiving ends, the two codes must have complementary DFTs such that the multiplication of the two transforms yields a constant value.

Using the DFT property for finding optimal codes is exemplified for \( M = 3 \). There are \( 2^3 = 8 \) possible binary codes of length \( M = 3 \). For each possible code we calculate the DFT. For example, for the three-element Barker code \{1 1 -1\}, we get the following DFT values:

\[
\{1 \quad 1 - j1.732 \quad 1 + j1.732\}
\]

The reciprocal DFT values are given by

\[
\{1 \quad 0.25 + j0.433 \quad 0.25 - j0.433\}
\]

and the reference code yielding perfect periodic cross-correlation is obtained by taking the inverse DFT and is given by \{0.5 0 0.5\}. Normalizing the reference signal to get the same peak response yields the normalized reference code \{1.5 0 1.5\}. While the periodic autocorrelation of the original code is
PHASE-CODED PULSE

Table 6.9 Globally Optimal Sequences

<table>
<thead>
<tr>
<th>$M$</th>
<th>Sequence</th>
<th>Loss (dB)</th>
<th>$M$</th>
<th>Sequence</th>
<th>Loss (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>100</td>
<td>1.76</td>
<td>17</td>
<td>11001101011000000</td>
<td>0.85</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>0.00</td>
<td>18</td>
<td>10101100011000000</td>
<td>1.09</td>
</tr>
<tr>
<td>5</td>
<td>10000</td>
<td>0.46</td>
<td>19</td>
<td>1100010111100100000</td>
<td>0.46</td>
</tr>
<tr>
<td>6</td>
<td>100000</td>
<td>1.19</td>
<td>20</td>
<td>10010001011100010000</td>
<td>0.46</td>
</tr>
<tr>
<td>7</td>
<td>1000000</td>
<td>1.88</td>
<td>21</td>
<td>1110100010010000000</td>
<td>0.45</td>
</tr>
<tr>
<td>8</td>
<td>11010000</td>
<td>1.25</td>
<td>22</td>
<td>11000100101111000000</td>
<td>0.46</td>
</tr>
<tr>
<td>9</td>
<td>100100000</td>
<td>2.21</td>
<td>23</td>
<td>100011010010001100000</td>
<td>0.46</td>
</tr>
<tr>
<td>10</td>
<td>111010000</td>
<td>2.24</td>
<td>24</td>
<td>110000100101110100000</td>
<td>0.25</td>
</tr>
<tr>
<td>11</td>
<td>11100100000</td>
<td>1.11</td>
<td>25</td>
<td>1011110010101000100000000</td>
<td>0.49</td>
</tr>
<tr>
<td>12</td>
<td>110010100000</td>
<td>0.51</td>
<td>26</td>
<td>1000010111100010011000000</td>
<td>0.51</td>
</tr>
<tr>
<td>13</td>
<td>11001010000000</td>
<td>0.17</td>
<td>27</td>
<td>1001111101000100010010000000</td>
<td>0.40</td>
</tr>
<tr>
<td>14</td>
<td>110010100000000</td>
<td>0.85</td>
<td>28</td>
<td>1011001001011110010100100000000</td>
<td>0.53</td>
</tr>
<tr>
<td>15</td>
<td>10100111101000000</td>
<td>0.62</td>
<td>29</td>
<td>1001111000011011000100100000000</td>
<td>0.50</td>
</tr>
<tr>
<td>16</td>
<td>110011101010000000</td>
<td>1.00</td>
<td>30</td>
<td>1001010011111001000100000000000</td>
<td>0.51</td>
</tr>
</tbody>
</table>

{3 1 1}, the periodic cross-correlation of \{1 1 −1\} and \{1.5 0 1.5\} is \{3 0 0\}. The power loss involved with using \{1.5 0 1.5\} instead of \{1 1 −1\} as a reference is \((1.5^2 + 1.5^2)/3 = 1.5\) (1.76 dB).

Table 6.9 gives optimal codes for higher lengths. Note that it is not always possible to find a reference code yielding zero periodic cross-correlation (consider, for example, the case when one of the DFT values of the transmitted code is zero), but for any length there exists at least one binary code with all nonzero DFT elements. Note that the machine time needed to conduct the exhaustive search used to derive the codes of Table 6.9 becomes considerably large with $M$ above 30.

Note also that the periodic autocorrelation function of the optimal codes given in Table 6.9 has an irregular form, the result of which is an extended size of alphabet (phase and amplitude values) of the reference code used at the receiving end (e.g., for $M = 30$ there are 30 different values in the reference code). To decrease the size of the reference code alphabet, Ipatov suggested using binary sequences with only two- or three-valued periodic autocorrelation function values. Ipatov showed that using specific families of codes with two- or three-valued periodic correlation yields a two- or three-valued spectrum (and reference alphabet) and extremely low mismatch power loss.

Ipatov code generation is based on constructing an $m$-sequence over $GF(q)$ and replacing its elements by the binary symbols $±1$ according to some a priori selected law based on a corresponding cyclic difference set. Table 6.10 gives some Ipatov code parameters with the resulting mismatch loss. Consider, for example, the family for which $q = 5$, $v = 4$, and $r = 3$. Let $n = 2$. A corresponding binary sequence with length $M = (q^n − 1)v/(q − 1) = 24$ is generated on the basis of $m$-sequence with length $q^n − 1 = 24$. A primitive polynomial of degree
Table 6.10  Ipatov Parameters

<table>
<thead>
<tr>
<th>( q, v, r )</th>
<th>Difference set ( { v, v - r, \lambda } )</th>
<th>( n )</th>
<th>( M = (q^n - 1)v/(q - 1) )</th>
<th>Mismatch loss (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3, 1, 1</td>
<td>( {1, 0, 0} )</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>13</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>40</td>
<td>0.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>121</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>364</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>1093</td>
<td>0.51</td>
</tr>
<tr>
<td>4, 1, 1</td>
<td>( {1, 0, 0} )</td>
<td>2</td>
<td>5</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>21</td>
<td>1.00</td>
</tr>
<tr>
<td>5, 4, 3</td>
<td>( {4, 1, 0} )</td>
<td>2</td>
<td>24</td>
<td>0.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>124</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>624</td>
<td>0.45</td>
</tr>
<tr>
<td>7, 3, 2</td>
<td>( {3, 1, 0} )</td>
<td>2</td>
<td>24</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>171</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>1200</td>
<td>0.39</td>
</tr>
<tr>
<td>8, 7, 3</td>
<td>( {7, 4, 2} )</td>
<td>2</td>
<td>63</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>511</td>
<td>0.44</td>
</tr>
<tr>
<td>13, 4, 3</td>
<td>( {4, 1, 0} )</td>
<td>2</td>
<td>56</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>732</td>
<td>0.80</td>
</tr>
<tr>
<td>16, 15, 7</td>
<td>( {15, 8, 4} )</td>
<td>2</td>
<td>255</td>
<td>0.05</td>
</tr>
<tr>
<td>23, 11, 6</td>
<td>( {11, 5, 2} )</td>
<td>2</td>
<td>264</td>
<td>0.15</td>
</tr>
<tr>
<td>29, 7, 4</td>
<td>( {7, 3, 1} )</td>
<td>2</td>
<td>210</td>
<td>0.90</td>
</tr>
<tr>
<td>31, 15, 8</td>
<td>( {15, 7, 3} )</td>
<td>2</td>
<td>480</td>
<td>0.12</td>
</tr>
<tr>
<td>32, 31, 15</td>
<td>( {31, 16, 8} )</td>
<td>2</td>
<td>1023</td>
<td>0.01</td>
</tr>
<tr>
<td>47, 23, 12</td>
<td>( {23, 11, 5} )</td>
<td>2</td>
<td>1104</td>
<td>0.08</td>
</tr>
<tr>
<td>53, 13, 9</td>
<td>( {13, 4, 1} )</td>
<td>2</td>
<td>702</td>
<td>0.73</td>
</tr>
<tr>
<td>61, 15, 8</td>
<td>( {15, 7, 3} )</td>
<td>2</td>
<td>930</td>
<td>0.52</td>
</tr>
</tbody>
</table>

\( n = 2 \) over GF(5) is, for example, \( \pi(x) = x^2 + 4x + 2 \). The \( m \)-sequence \( \{d_i\} \) is generated by the recursion \( d_i = d_{i-1} + 3d_{i-2} \) with an arbitrary choice of initial conditions. Using \( d_1 = 0, d_2 = 1 \), we get the following 24-element \( m \)-sequence over GF (5):

\[
\{0 1 1 4 2 4 0 2 2 3 4 3 0 4 4 1 3 1 0 3 3 2 1 2\}
\]

The mapping law for the code is based on the \( \{4, 1, 0\} \) ordinary cyclic difference set by placing +1 only in the positions where the original \( m \)-sequence has a zero and at those positions that belong to some \( \{4, 1, 0\} \) difference set. A possible \( \{4, 1, 0\} \) CDS is the set \( \{1\} \); thus, we get the binary sequence given by

\[
\{1 1 1 -1 -1 -1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1\}
\]
Taking the inverse DFT of the reciprocal DFT of the binary code yields the following reference code:

\[
\{ 5 \ 11 \ 11 \ -7 \ -7 \ -7 \ 5 \ -7 \ -7 \ -7 \ -7 \ 5 \\
-7 \ -7 \ 11 \ -7 \ 11 \ 5 \ -7 \ -7 \ -7 \ 11 \ -7 \}
\]

with a three-element alphabet. The mismatch loss is only 1.061 (0.28 dB).

## 6.6 Optimal Filters for Sidelobe Suppression

Whereas the Ipatov code shows a way of designing code pairs with perfect periodic cross-correlation and minimal mismatch loss, the method described here shows for any given transmitted code, a way of finding a suboptimal reference code, yielding lower aperiodic correlation sidelobes. Assume that the transmitted signal is defined by an \(M\)-element code \(\{s_m\}\). We are interested in designing a suboptimal filter defined by a \(K \geq M\) element code \(\{q_k\}\) such that the cross-ambiguity function between the two signals gives minimal ambiguity function values for a specific selection of points on the delay–Doppler grid (for brevity, we use ambiguity instead of cross-ambiguity). We start by describing a simplified problem where we wish to control only the cross-correlation function on grid points given by integer multiples of the chip duration (Griep et al., 1995).

Define a filter vector \(\mathbf{q}\) (with length \(K\)) and a signal vector \(\mathbf{s}\) (with length \(M\)) such that the elements of \(\mathbf{q}\) and \(\mathbf{s}\) are given by

\[
\mathbf{q} = [q_1 \ q_2 \ \cdots \ q_K], \quad \mathbf{s} = [s_1 \ s_2 \ \cdots \ s_M]
\]

The cross-correlation function for integer delay \(l\) between the pulse coded by \(\{s_m\}\) and the reference pulse defined by \(\{q_m\}\) is given by

\[
R_{sq}[l] = \sum_{m=1}^{K} s_m q_{m+l}^* \tag{6.16}
\]

where \(-(K - 1) \leq l \leq (K - 1)\). Define the correlation vector \(\mathbf{y}\), where \(y_m = R_{sq}[m]\); then (6.16) can be represented in matrix form as \(\mathbf{y} = \mathbf{q}^* \mathbf{A}\), and

\[
\mathbf{A} = \begin{bmatrix}
    s_K & \cdots & s_2 & s_1 & 0 & \cdots & 0 \\
    0 & s_K & \cdots & \cdots & \cdots & \cdots & \cdots \\
    \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
    0 & \cdots & 0 & s_K & \cdots & s_2 & s_1
\end{bmatrix} \tag{6.17}
\]

Define a \(P \times P\) weight diagonal matrix \(\mathbf{F}\) such that \(E\), the total weighted energy of \(\mathbf{y}\), is given by

\[
E = \mathbf{y} \mathbf{F} \mathbf{y}^H = (\mathbf{q}^* \mathbf{A}) \mathbf{F} (\mathbf{q}^* \mathbf{A})^H = \mathbf{q}^* (\mathbf{A} \mathbf{F} \mathbf{A}^H) \mathbf{q}^T = \mathbf{q}^* \mathbf{A} \mathbf{q}^T \tag{6.18}
\]
The vector $q$ that minimizes $E$ under the restriction that $sq^H = ss^H$ (same response for zero Doppler and zero delay), and that $A = \Lambda F A^H$ is not singular, is given by $q = sA^{-1}(ss^H)/(sA^{-1}s^H)$. The minimal value of $E$ is given by $(ss^H)^2/(sA^{-1}s^H)$.

The example below shows the design of a minimum integrated sidelobe level (ISLL) reference code in the case where the code used for the transmitted pulse is the five-element Barker code. Griep et al. (1995) also give additional methods for designing reference codes for a minimum peak sidelobe level and minimum cross-correlation peak power signals. We start with a Barker binary code of length $M = 5$. Assume that the length $K$ of the reference signal $q$ is also 5. The cross-correlation function between $s$ and $q$ is to be minimized on grid points laying on the zero Doppler axis and given by $\tau = \pm k t_b (1 \leq k \leq 5)$. Define

$$\Lambda = \begin{bmatrix} 1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 1 \end{bmatrix},$$

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and get that $q = [0.7692 \ 1.3462 \ 0.7692 \ -1.3462 \ 0.7692]$. The resulting cross-correlation is shown in Fig. 6.25. Note the reduction in the sidelobe level. Using a longer reference signal can yield much lower correlation sidelobes. An example of a longer reference code ($K = 11$) using $q = [-0.2769 \ -0.3175 \ 0.0554 \ 0.7540 \ 1.3293 \ 0.8334 \ -1.3293 \ 0.7540 \ -0.0554 \ -0.3175 \ 0.2769]$ is shown in Fig. 6.26. The SNR loss involved with using a suboptimal receiver (nonmatched) is only 0.33 dB with $K = 5$ and only 0.59 dB with $K = 11$.

Note that for a given signal $\{s_m\}$, the ambiguity function and its derivatives of any order for any value of $\tau$ and $\nu$ are a linear function of $\{q_k\}$. Thus, the procedure described by Griep et al. (1995) can be generalized. We define an output vector $y$ where each element $y_p (p = 1, 2, \ldots, P)$ represents a point $\langle \tau_p, \nu_p \rangle$ where we wish to impose restrictions on the $r_p$ derivative of the ambiguity function in a direction $\varepsilon_p$ relative to the delay axis ($r_p \geq 0$; $r_p = 0$ stands for the value of the ambiguity function itself). Since each element $y_p$ is a linear function of $\{q_k\}$, we can still write $y = q^* \Lambda$, where $\Lambda$ is a matrix with $K$ rows and $P$ columns and the elements of $\Lambda$ are a function of $A$, $\tau_p, \nu_p, r_p$, and $\varepsilon_p$. 

FIGURE 6.25 ACF of the five-element Barker and cross-correlation with the minimum ISLL reference of length $K = 5$.

FIGURE 6.26 ACF of the five-element Barker and cross-correlation with the minimum ISLL reference code of length $K = 11$.

6.7 Huffman Code

Huffman (1962) considered the idealized compressed pulse that contains no side-lobes except the unavoidable sidelobes at the two ends of the compressed pulse. The end sidelobes are unavoidable because at the corresponding delay only one chip remains in the integrated product of the received and reference signals (see Fig. 6.27). The end sidelobe level is a design parameter. It turns out that the resulting pulses vary in amplitude as well as in phase. Because of the amplitude variations the waveforms were not very practical for radar applications. However, with the use of solid-state transmitters, they may become more attractive.

FIGURE 6.27 Reason for the edge sidelobe.
The Huffman code is derived simply by describing the time sequence using polynomials. Accordingly, the transmitted sequence polynomial $S(z)$ is given by

$$S(z) = s_1 + s_2 z^1 + s_3 z^2 + \cdots + s_{M} z^{M-1}$$  \hfill (6.19)

where \( \{s_m\} \) is the complex-valued code. Huffman showed that to obtain the desired criteria, the roots of the $S(z)$ polynomial should lie in the $z$-plane at intervals of $2\pi/(M-1)$ on two circles whose radii $R$ and $R^{-1}$ are given by

$$\left[ \left| \frac{1}{2a} \right| \pm \left( \frac{1}{4a^2} - 1 \right)^{1/2} \right]^{1/(M-1)}$$  \hfill (6.20)

The design of the Huffman code consists of specifying the number of code elements $M$, the edge sidelobe level $a$, and the particular choice of $z$-plane zeros which results from choosing one zero from each of the $M-1$ pairs. The design of a Huffman sequence with length $M = 6$ is given below.

We chose to design a Huffman code with peak-to-peak sidelobe level of $a = \frac{1}{6}$. The roots of the $S(z)$ polynomial lie in the $z$-plane at intervals of $2\pi/5$ on two circles whose radii $R$ and $R^{-1}$ are 1.423 and 0.703. We calculate the $S(z)$ polynomial by selecting one zero from each of the five pairs. For example,
selecting all the inner zeros that lie on radii $R = 0.703$ gives the trivial solution of $S(z) = z^5 - 0.1716$, implying the code $s = [1 \ 0 \ 0 \ 0 \ 0 \ -0.1716]$. A different selection of zeros is shown in Fig. 6.28. The resulting polynomial is $S(z) = z^5 - 0.7198z^4 + 1.024z^3 + 1.4569z^2 + 2.0727z - 2.8796$, implying the code $s = [1 \ -0.7198 \ 1.0240 \ 1.4569 \ 2.0727 \ -2.8796]$. The aperiodic autocorrelation of the Huffman code of length $M = 6$ is shown in Fig. 6.29. Selecting the zero patterns in a random manner generally results in codes that vary considerably in amplitude from bit to bit. This represents a loss in terms of the power that could be transmitted at the maximum level.

An example of a longer Huffman code ($M = 23$) is given in Figs. 6.30 to 6.32. Figure 6.30 displays the real envelope of the Huffman signal. The peak-to-mean envelope power ratio (PMEPR) in this signal is 3.8. Methods to reduce the PMEPR of Huffman-coded signals were outlined by Ackroyd (1970). Figure 6.31 displays the autocorrelation function (in dB). The peak ACF sidelobe, which is a design parameter, is $-63$ dB. Figure 6.32 displays the partial ambiguity function. Clearly evident are the sidelobe-free zero-Doppler cut and the fast sidelobe buildup with Doppler.

![FIGURE 6.29](image)

**FIGURE 6.29** ACF of the six-element Huffman code.

![FIGURE 6.30](image)

**FIGURE 6.30** Real envelope of a Huffman code ($M = 23$).
The instantaneous phase switching and amplitude rise time in Barker and other phase-coded signals causes extended spectral sidelobes. For example, the autocorrelation function (ACF) and spectrum (both in dB) of an ideal Barker 13 signal appear in Fig. 6.33. Note the expected ACF sidelobe peaks of $20 \log\left(\frac{1}{13}\right) = -22.3$ dB. The spectral sidelobes, referred to as spectral skirt, decay rather slowly, at a rate of 6 dB/octave. Of special interest is the first null at $f = 13/T = 1/t_b$, namely at the inverse of the subpulse “bit” duration. The spectral skirt poses
problems at both the transmitter and the receiver. The signal transmitted may not meet spectral emission regulations. It may also exceed the bandwidth of the final transmitting stages (e.g., the antenna).

In a practical receiver (see Fig. 6.34), the matched filter will be implemented digitally; in a correlator, that follows an analog-to-digital (A/D) converter. To limit the noise power reaching the A/D, an analog narrowband filter is likely to precede it. That filter will cut off the spectrum tail. In some receivers (Farnett and Stevens, 1990; Taylor, 1990) the bandpass filter matches (or nearly matches) the subpulse, and the sample interval equals the subpulse duration. That approach is not likely to produce an output that is the exact ACF. For example, the highly spaced samples could miss the output’s peak, causing a “straddle” or “cusping” loss of 2.3 dB (Klein and Fujita, 1979).

Rather than allow the bandpass filter to influence the delay response, the spectrum tail can be narrowed by modifying the signal in the transmitter (e.g., by slowing the phase switching rate). Figure 6.35 shows the phase evolution of a Barker 13 signal in which the phase transition occupies one-fifth of the bit duration. The resulted ACF and spectrum appear in Fig. 6.36. The ACF peak sidelobe increased only slightly (to −20 dB), while the −30 dB spectral width was nearly halved.

A more drastic modification to biphase coding is the biphase-to-quadriphase (BTQ) transformation (Taylor and Blinchikoff, 1988) that yields the quadriphase
BANDWIDTH CONSIDERATIONS IN PHASE-CODED SIGNALS

FIGURE 6.34 Matched-filter implementation for phase-coded signal.

code. The complex envelope of a quadriphase code, which follows a biphase code with elements $W_k$, is

$$u(t) = \sum_{k=1}^{M} j^{s(k-1)} W_k p(t - kt_b) \quad (6.21)$$

where the subpulse $p(t)$ is a half-cycle of a cosine wave of length $2t_b$, namely,

$$p(t) = \cos \frac{\pi t}{2t_b}, \quad -t_b \leq t \leq t_b \quad (6.22)$$

The constant coefficient $s$ can be either 1 or $-1$. The quadriphase radar signal is similar to the minimum shift keying (MSK) digital modulation technique. The magnitude $a(t)$ and phase $\phi(t)$ of the complex envelope

$$u(t) = a(t) \exp[j\phi(t)]$$
are continuous functions. If the original sequence of phases is \( \theta_k, k = 1, \ldots, M \), the transformed phases, at multiples of the subpulse duration \( t_b \), are given by

\[
\phi(k \tau_b) = \begin{cases} 
0, & k = 0 \\
s(k-1)\pi/2 + \theta_k, & k = 1, \ldots, M \\
0, & k = M + 1
\end{cases}
\]  \hspace{1cm} (6.23)

In between multiples of \( t_b \), the phase is given by straight-line segments, connecting the values at multiples of \( t_b \). For Barker 13 the phases derived from (6.23) appear as shown in Table 6.11.

The magnitude \( a(t) \), which is a rectangle in the biphase case, is given in the quadriphasic case as

\[
a(t) = \begin{cases} 
A \sin(2\pi t/4t_b), & 0 \leq t \leq t_b \\
A, & t_b \leq t \leq M t_b \\
A \cos[2\pi(t - M t_b)/4t_b], & M t_b \leq t \leq (M + 1)t_b
\end{cases}
\]  \hspace{1cm} (6.24)

Figure 6.37 shows the magnitude, phase, and frequency evolution of the quadriphase signal that corresponds to Barker 13. The frequency subplot indicates a frequency switching between \(-1/(4t_b)\) and \(+1/(4t_b)\), except at the edge bits, where the frequency of the complex envelope is zero.
TABLE 6.11  Biphase-to-Quadriphase Transformation
\((s = +1)\) of a Barker 13 Sequence

<table>
<thead>
<tr>
<th>(k)</th>
<th>(\theta_k)</th>
<th>(\phi_k \mod 2)</th>
<th>(k)</th>
<th>(\theta_k)</th>
<th>(\phi_k \mod 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>—</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>(3\pi/2)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>(2\pi)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>(\pi/2)</td>
<td>10</td>
<td>(\pi)</td>
<td>(3\pi/2)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>(\pi)</td>
<td>11</td>
<td>0</td>
<td>(\pi)</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>(3\pi/2)</td>
<td>12</td>
<td>(\pi)</td>
<td>(\pi/2)</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>(2\pi)</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>(\pi)</td>
<td>(3\pi/2)</td>
<td>14</td>
<td>—</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>(\pi)</td>
<td>(2\pi)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The ACF and spectrum of quadriphase code 13 are shown in Fig. 6.38, and should be compared to the ACF and spectrum of Barker 13 (Fig. 6.33). The ACF is very similar to Barker 13. However, the spectral skirt falloff rate has doubled (12 dB/octave). The first null is now at \(f \approx 1/t_b \sqrt{2}\) and the \(-30\) dB point is now at \(f \approx 1/t_b\). Although the ACF changed little from Barker 13 to quadriphase 13, the AF (Fig. 6.39) off the zero-Doppler axis is quite different (Levanon and
FIGURE 6.38  ACF (top) and spectrum (bottom) of quadriphase code 13.

FIGURE 6.39  Ambiguity function of quadriphase code 13.
BANDWIDTH CONSIDERATIONS IN PHASE-CODED SIGNALS

Freedman, 1989) from that of Barker 13 (Fig. 6.6). Quadriphase code 13 was generated using the MATLAB code in Appendix 6B and the ambiguity plotting codes.

Chen and Cantrell (2002) suggested another bandlimiting technique, in which the rectangular shape of the phase bits is replaced by the shape of a Gaussian-windowed sinc function. A generalization of their signal, which allows polyphase coding (rather than just binary), will define the complex envelope of a single bit as

\[
-u_m(t) = \exp(j\phi_m) \frac{\sin(\pi t/t_b)}{\pi t/t_b} \exp \left[ -\frac{t^2}{2(\sigma t_b)^2} \right],
\]

\[-2t_b \leq t \leq 2t_b, \text{ zero elsewhere} \quad (6.25)

A pulse constructed from \( M \) bits is defined as

\[
u(t) = \sum_{m=1}^{M} u_m[t - (m - 1)t_b]
\]

(6.26)

It is important to note that the bit duration now extends over \( 4t_b \), yet consecutive bits are spaced only \( t_b \) apart. Hence there is an overlap of \( 3t_b \). This also implies that the duration of the entire pulse is \( T = (M + 3)t_b \) and not \( Mt_b \), as in a conventional phase-coded signal. The magnitude of a bit is shown in Fig. 6.40

**FIGURE 6.40**  Gaussian-windowed sinc function.
(solid), which also displays its two components: the sinc (dotted) and the Gaussian (dashed). $\sigma = 0.7$ was used in (6.25).

The use of this bit shape will first be demonstrated on a Barker 13 signal. The magnitude and phase of the signal are shown in Fig. 6.41. The variable amplitude points to the main difficulty with this signal—the requirement for linear power amplification. The autocorrelation function and spectrum are shown in Fig. 6.42. They should be compared to Figs. 6.33 and 6.38. The ACF (top subplot of Fig. 6.42) peak sidelobe level is not much different than in the conventional Barker or the quadriphase Barker. However, the spectrum (lower subplot of Fig. 6.42) is dramatically narrower, with a negligible spectral skirt. The ambiguity function (AF) is plotted in Fig. 6.43 and is very similar to the AF of a conventional Barker 13 signal.

The second example will be a 25-element P4 signal utilizing a Gaussian-windowed sinc bit shape. In the P4 case the overlap of 3 bits will cause a variable phase within a bit. The amplitude and phase of the signal are given in Fig. 6.44. $\sigma = 0.7$ is still used in (6.25). The need for a linear power amplifier is obvious from the top subplot. The phase, which is a smoothed P4 phase, begins to resemble the phase behavior of a linear-FM signal. The ACF and spectrum of the modified 25-element P4 signal are plotted in Fig. 6.45 and should be compared to those of a conventional 25-element P4 (Fig. 6.46). The modified P4

![Figure 6.41](image-url)  
**FIGURE 6.41**  Barker 13 with a bit shape of Gaussian-windowed sinc.
FIGURE 6.42 ACF and spectrum of Barker 13 with a bit shape of Gaussian-windowed sinc.

FIGURE 6.43 Ambiguity function of Barker 13 with a bit shape of Gaussian-windowed sinc.
FIGURE 6.44 P4 (25-bit) with a bit shape of Gaussian-windowed sinc.

FIGURE 6.45 ACF and spectrum of 25-element P4 with a bit shape of Gaussian-windowed sinc.
exhibits a small reduction of ACF sidelobes and a dramatic (almost complete) reduction of the spectral sidelobes. The examples of phase-coded signal with a Gaussian-windowed sinc bit waveform in this chapter were generated using the MATLAB program in Appendix 6C.

Although not shown, it should be pointed out that the ideal periodic autocorrelation of a conventional P4 signal is lost in the modified P4 signal. In contrast, while Barker 13 PACF has rather high sidelobes, its quadriphase modification turns out to be very nearly ideal, with a PACF sidelobe level near $-60$ dB. This property is not repeated in the quadriphase signal based on Barker 7.

### 6.9 CONCLUDING COMMENTS

We have discussed some sets of codes that can be operated either periodically or aperiodically under various conditions and optimization rules. Most of the codes described here are used in operational radar systems. Many other codes were not presented in detail. Among these codes we choose to mention the primitive root and quadratic residue codes (Schroeder, 1986), generalized Frank codes (Suehiro and Hatori, 1988a; Kretschmer and Gerlach 1991), $N$-shift cross-orthogonal...
sequences (Suehiro and Hatori, 1988b), reciprocal perfect phase codes (Kretschmer and Gerlach, 1991), generalized P4/Zadoff–Chu codes (Kretschmer and Gerlach, 1991; Popovic, 1992), index codes (Kretschmer and Gerlach, 1991), and binary alexis sequences (Luke, 2001). Many other codes have been published. Finally, we note the theoretical work initiated by Welch (1974) and Sarwate (1979) putting bounds on the cross-correlation and autocorrelation of sets of periodic and aperiodic sequences.

**APPENDIX 6A: GALOIS FIELDS**

A Galois field is an algebraic identity having a finite number of members and obeying a set of roles. The simplest and smallest example of such a field is the set containing the elements 1 and 0. We denote this field as GF(2). Galois fields play a fundamental role in the theory and application of error-control coding, multiuser communication, cryptography, and digital signal processing. This appendix is not a replacement for a good textbook on the subject or a dedicated course. Instead, it can serve as a quick tutorial for those who are not familiar with the subject at all, or as a quick reference for those who know the basics of the subject but do not practice it. We start by defining some basic algebraic identities.

**Definition:** A field is a set \( F \) of elements in which it is possible to add, subtract, multiply, and divide (except that division by 0 is not defined). Addition and multiplication must satisfy the commutative, associative, and distributive laws. That is, for all elements \( x, y, \) and \( z \) of \( F \):

1. \( x + y = y + x \) (addition is commutative).
2. \( (x + y) + z = x + (y + z) \) (addition is associative).
3. \( x \cdot y = y \cdot x \) (multiplication is commutative).
4. \( x \cdot (y \cdot z) = (x \cdot y) \cdot z \) (multiplication is associative).
5. \( x \cdot (y + z) = x \cdot y + x \cdot z \) and \( (x + y) \cdot z = x \cdot z + y \cdot z \) (the distributive law).

The field must also contain elements 0, 1, \(-x\), and \(x^{-1}\) with the properties that \( x + 0 = x\), \(1 \cdot x = x\), \(x + (-x) = 0\), and \(x \cdot x^{-1} = 1\) for all elements \(x\) of \(F\).

A finite field (or Galois field) contains a finite number of elements, this number being called the order of the field. For example, let \(p\) be a prime number. Then the integers modulo \(p\) form a field of order \(p\), denoted by GF(\(p\)). The elements of GF(\(p\)) are \{0, 1, 2, \ldots, \(p - 1\)\} and \(+, -, \cdot, \) and \(\div\) are carried out mod \(p\). GF(3) is the ternary field \{0, 1, 2\} with \(1 + 2 = 3 = 0\) (mod 3), \(2 \cdot 2 = 4 = 1\) (mod 3), \(1 - 2 = -1 = 2\) (mod 3), \(1 \div 2 = 4 \div 2 = 2\) (mod 3), and so on.
The multiplication and addition tables for the GF(3) are given by

\[
\begin{array}{c|ccc}
+ & 0 & 1 & 2 \\
\hline
0 & 0 & 1 & 2 \\
1 & 1 & 2 & 0 \\
2 & 2 & 0 & 1 \\
\end{array}
\quad \quad \quad
\begin{array}{c|ccc}
\cdot & 0 & 1 & 2 \\
\hline
0 & 0 & 0 & 0 \\
1 & 0 & 1 & 2 \\
2 & 0 & 2 & 1 \\
\end{array}
\]

Finite fields of order \( p^m \) exist for all \( p \) and \( m \) where \( p \) is a prime and \( m \) is some integer. Furthermore, all fields of the same order are equivalent. An example of a field with \( 2^2 = 4 \) elements is now given. Each element of GF(4) can be represented by two elements of GF(2). This gives the four elements given by 00, 01, 10, and 11.

Any two elements of GF(4) can clearly be added by adding the elements in the corresponding location using the addition rule in GF(2) (i.e., modulo 2). To define multiplication we associate with each element a polynomial in \( \alpha \):

<table>
<thead>
<tr>
<th>Vector representation</th>
<th>Polynomial representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>( \alpha )</td>
</tr>
<tr>
<td>11</td>
<td>( 1 + \alpha )</td>
</tr>
</tbody>
</table>

The multiplication of two elements in the field is defined as the multiplication of the two polynomials. Since simple multiplication of any two polynomials in the field does not always give a polynomial with degree \( \leq 1 \), we must also agree that \( \alpha \) will satisfy a certain fixed equation of degree \( 2 \). A suitable equation is \( \pi(\alpha) = \alpha^2 + \alpha + 1 = 0 \). For the multiplication to have an inverse, which it must if our system is to be a field, \( \pi(x) \) must be irreducible over GF(2).

**Definition:** A **primitive element** of GF\((p^m)\) is a cyclic generator of the group of nonzero elements of GF\((p^m)\). This means that every nonzero element of the field can be expressed as the primitive element raised to some integer power. Usually, the primitive element is denoted by \( \alpha \). In the example given before for the construction of GF(4), note that the “10” element is \( 1 = \alpha^0 \), “01” is \( \alpha = \alpha^1 \), and “11” is \( 1 + \alpha = \alpha^2 \), since \( \pi(\alpha) = \alpha^2 + \alpha + 1 = 0 \).

**Definition:** A **primitive polynomial** for GF\((p^m)\) is the minimal polynomial of some primitive element of GF\((p^m)\). That is, it is the \( p \)-ary coefficient polynomial of smallest nonzero degree having a certain primitive element as a root in GF\((p^m)\). Consequently, a primitive polynomial has degree \( m \) and is irreducible (i.e., it is not the product of two polynomials of lower degree in the field). For example, the polynomial \( \pi(x) = x^2 + x + 1 \) has degree 2 and is irreducible over GF(2) since \( \pi(1) = 1^2 + 1 + 1 = 1 \) and \( \pi(0) = 0^2 + 0 + 1 = 1 \).
Using the definition of the primitive polynomial and primitive element, we can now generalize the example given before for the construction of Galois fields of order \( p^m \).

**Definition:** Suppose that \( \pi(x) \) is irreducible over \( \text{GF}(p) \) and has degree \( m \). Then the set of all polynomials in \( x \) of degree \( \leq m - 1 \) and with coefficients from \( \text{GF}(p) \), with calculations performed modulo \( \pi(x) \), form a field of order \( p^m \). The prime number \( p \) is called the characteristics of the field.

Computation in a finite field with characteristics 2 is easily manipulated using digital circuits or in a binary computer. An element of \( \text{GF}(2^m) \) is stored in a row of \( m \) binary storage elements (a register). To multiply by \( \alpha \) where \( \pi(\alpha) = 0 \), we use a linear feedback shift register with the taps connected according to the coefficients of the primitive polynomial \( \pi(x) \). For example, if the shift register in Fig. 6A.1 initially contains \( a_0, a_1 \leftrightarrow a_0 + a_1\alpha \), one time instant later it contains \( a_0\alpha + a_1\alpha^2 = a_0\alpha + a_1(\alpha + 1) = a_1 + (a_0 + a_1)\alpha \).

If the initial condition is not 00 and \( \alpha \) is primitive, the output of the circuit of Fig. 6A.1 is periodic with period \( 2^{2^2} - 1 = 3 \). This is the maximum possible period with two storage elements. The resulting sequence of 0’s and 1’s is called a **maximal-length shift register code**, or **m-sequence**. In a similar way, circuits with ternary or \( p \)-ary elements can be implemented, forming **m-sequences** over a \( p \)-ary alphabet.

**APPENDIX 6B: QUADRIPHASE BARKER 13**

The following MATLAB program generates a numerical quadriphase Barker 13 signal, which is then used with the ambiguity function plot programs in Appendix 3A to study its performances. After running the code given here, the workspace includes the variables \( u_{\text{amp}} \) and \( f_{\text{basic}} \), which define the quadriphase Barker 13 signal complex envelope. Use \( u_{\text{amp}}=u_{\text{amp}} \) and \( f_{\text{basic}} =f_{\text{basic}} \) and run the ambiguity function code of Appendix 3A to plot the ambiguity function of the code.
APPENDIX 6C: GAUSSIAN-WINDOWED SINC

The following MATLAB program generates several phase-coded pulses in which the bit is a Gaussian-windowed sinc extended over four code bits. After running the code given here, the workspace includes the variables $u_{\text{amp1}}$ and $f_{\text{basic1}}$, which define the Gaussian-windowed sinc signal complex envelope. Use $u_{\text{amp}}=u_{\text{amp1}}$ and $f_{\text{basic}}=f_{\text{basic1}}$ and run the ambiguity function code of Appendix 3A to plot the ambiguity function of the code.

% gaussian_sinc.m - Gaussian windowed sinc waveform
% written by Nadav Levanon

% single bit shape
nn=201; nn2=(nn-1)/2; nn22=nn2/2;
small=.00000001;
nnn=-nn2:nn2;
arg_bit=small+4*pi/nn*nnn;
amp_bit=sin(arg_bit)./arg_bit;
s_gauss=.7;
gauss_weight=exp(-0.5*(nnn./(nn/4*s_gauss)).^2);
ab_a=amp_bit.*gauss_weight;
sig_type=input(' Barker13 =1, P4 =2, single bit =3, = ? '); 
if sig_type==1
  phase_vec=pi*[0 0 0 0 0 1 1 0 0 1 0 1 0]; % Barker 13
elseif sig_type==2
  mm=input(' No. of elements of P4 signal = ? '); 
  m=1:mm;
  phase_vec=pi*(1/mm*(m-1).*2./(m-1)); % P4
else
  fprintf('Invalid input.
');
end

% quadriphase_barker13.m - Barker 13 using Taylor's quadriphase modification
% written by Nadav Levanon

nn=14;

mm=100;
mmm=1:mm;
amp_rise=sin(2*pi/4/mm*mmm);
amp_fall=fliplr(amp_rise);
a1 =ones(1,mm);
b1 =-a1;
c1=zeros(1,mm);

f_basic1=0.25/mm*[c1 a1 a1 a1 b1 a1 b1 a1 b1 b1 b1 c1];
u_amp1=[amp_rise a1 a1 a1 a1 a1 a1 a1 a1 a1 a1 a1 a1 amp_fall];

figure(10)
plot( u_amp1,'+')
grid
axis([0 mm*nn -.1 1.1])
elseif sig_type==3
    phase_vec=[0 0 0 0 0 0 0 0 0];
end

lb=length(phase_vec);
vec_length=(lb+3)*nn22+1;
ab=zeros(lb,vec_length);
bpv=ones(1,nn); % bit phase vector

for k=1:lb
    ab(k,:)=[zeros(1,(k-1)*nn22), ab_a.*exp(j*phase_vec(k)*bpv),
            zeros(1,(lb-k)*nn22)];
end

u_amp_complex=sum(ab);

u_amp1=abs(u_amp_complex);
u_phase1=1/pi*angle(u_amp_complex);
t_axis=nnn/nn22;

figure(10)
plot(u_amp1,'k','linewidth',1.5)
grid

figure(11), clf, hold off
plot( t_axis,ab_a,'k','linewidth',2.5)
hold on
plot(t_axis,gauss_weight,'k--','linewidth',1.5)
plot(t_axis,amp_bit,'k:','linewidth',1.5)
grid
axis([-inf inf -.3 1.1])
xlabel('τ/τb')

PROBLEMS

6.1 Equivalent codes
Show that every polyphase code is equivalent to one that begins \(\{φ_1 = 0, \; φ_2 = 0, \; φ_3, \ldots\}\), where \(0 ≤ φ_3 ≤ π\). What does the equivalence imply regarding:

(a) The autocorrelation function of the code?
(b) The cross-correlation function of the code with a different code?
(c) The frequency spectrum of the code?

6.2 Cyclic shift operation
Show that cyclically shifting a code preserves the periodic autocorrelation function. What happens to the periodic cross-correlation when cyclically shifting one of the codes?

6.3 Decimation
Show that decimating a sequence by \(d\), which is relatively prime to the code length \(M\), results with decimating the periodic autocorrelation function.
6.4 Symmetric aperiodic ACF sidelobe property of a perfect code
Show that for a perfect code, the aperiodic correlation function has side-
lobes symmetric around $\tau = M t_b / 2$.

6.5 New perfect codes
Consider a new code obtained by multiplying the elements of a Frank code and a Chu code of the same odd length.
(a) Calculate the code elements for a length $M = 25$.
(b) Show that the new code is also a perfect code.
(c) Prove that all codes formed in the same way are perfect.

6.6 Code comparison I
For the Barker, polyphase Barker, Chu, P3, and P4 codes with $M = 13$ elements, calculate and plot:
(a) The phase history.
(b) The aperiodic autocorrelation function.
(c) The periodic correlation function.
(d) The power spectrums.
What can be said regarding the periodic and aperiodic correlation functions of codes from observing the power spectrums?

6.7 Code comparison II
Calculate the aperiodic autocorrelation function of a minimum peak side-
lobe code of length $M = 6$. Give a four-phase code of length 6 with a peak sidelobe of 2, using progressive multiplication.
(a) Plot the autocorrelation function of the new codes and compare to the MPS code obtained using Table 6.3.
(b) Compare the two codes’ structure, integrated sidelobe levels, frequency spectrums, and ambiguity functions.

6.8 Code comparison III
For a P4, palindromic P4, and P($4,k_{opt}$) codes of length $M = 64$, compare:
(a) The aperiodic and periodic mainlobe width.
(b) Autocorrelation peak response and autocorrelation peak sidelobe.
Use the results to define scenarios or applications, for which one of the three types of signals is more suitable than the others.

6.9 Code comparison IV
(a) Calculate the phase elements of an $m$-sequence of length $M = 15$
(use the shift register initial condition where all states are “1”).
(b) Plot the aperiodic and periodic autocorrelation functions of the result-
ing code.
(c) Calculate Golomb’s biphase code based on the $m$-sequence just cal-
culated and plot the aperiodic and periodic autocorrelation function
of the Golomb code.
(d) Compare the resulting periodic autocorrelation and aperiodic autocorrelation with a globally optimal Ipatov binary code of length $M = 15$.

6.10 Code comparison V
(a) Calculate an Ipatov binary code of length $M = 63$ (use Table 6.10).
(b) Compare a code with an $m$-sequence and a Golomb biphase code of length $M = 63$.
(c) Compare the codes to Frank and Px codes of length $M = 64$.

6.11 Suboptimal reference code
(a) For a Barker code of length $M = 13$, calculate an optimal filter with length 13 and 27 such that the integrated cross-correlation sidelobe level is minimized.
(b) What is the SNR loss when using the longer reference code compared to a shorter code and compared to the matched-filter case?

6.12 Primitive root code
The $M$ code words of the primitive root code are defined as
$$u_m = \exp[-j2\pi/(M + 1)\alpha^{m-1}],$$
where $M - 1$ must be a prime number and $\alpha$ is a primitive root modulo $M + 1$.

(a) Calculate and plot the phase and autocorrelation functions of the primitive root codes of length $M = 100$ (101 is prime) with $\alpha = 2$ and 3.
(b) Plot the aperiodic cross-correlation between the two codes.
The results are typical for the cross-correlations between two primitive root codes of the same length or number of elements; the sidelobes are down from the peak by approximately the pulse compression ratio.

6.13 Quaternary residue code
We introduce the Legendre symbol $\{q/p\}$. This symbol is defined for all $q$ that are not divisible by $p$; it is equal to 1 if $q$ is a quadratic residue of $p$ and is equal to $-1$ otherwise. Note that $q$ is a quadratic residue of $p$ if the congruence $z^2 = q \mod p$ has a solution. The code is defined as $u_m = \{(m - 1)/M\}$, where $M$ is a prime number of the form $4N - 1$. Note that we define $\{0/M\} = 1$.

(a) Calculate the quaternary residue codes of length $M = 11$ and $M = 19$.
(b) Plot the aperiodic autocorrelation functions of the two codes.

6.14 Suboptimal filtering of nested codes
A nested Barker code of length 65, shown below, was created from an outer Barker code of length 5 in which each element is a Barker code of length 13.

```
+++++−−++−+−+ +++++−−++−+−+
+++++−−++−+−+ −−−−−++−−+−+
+++++−−++−+−+ −−−−−++−−+−+
+++++−−++−+−+ −−−−−++−−+−+
```
(a) Design a filter of length 31 with impulse response $H_{31}$ that will yield a minimum integrated sidelobe for Barker code of length 13.

(b) Perform convolution between $H_{31}$ and the entire code of length 65. The product will be labeled OUT1. Plot OUT1 using a linear scale.

(c) Design a filter of length 17 with impulse response $H_{17}$ that will yield a minimum integrated sidelobe for a Barker code of length 5.

(d) Extend $H_{17}$ by inserting 12 zero elements after each element of $H_{17}$. The new impulse response, of length $13 \times 17 = 221$, will be labeled $H_{17W}$.

(e) Perform convolution between OUT1 and $H_{17W}$. The overall convolution product will be labeled OUT. Plot OUT in decibels.

6.15 Quadriphase code I

Derive a quadriphase signal based on Barker 11. Calculate and plot:

(a) Its amplitude, phase, and frequency evolution.

(b) Its autocorrelation function (in dB).

(c) Its spectrum (in dB).

(d) Its ambiguity function.

![Quadriphase signal based on minimum peak sidelobe code of length 28.](FIGURE P6.16)
6.16 Quadriphase code II
Show that the minimum peak sidelobe code of length 28 (see Table 6.3) can be modified to the quadriphase signal shown in Fig. P6.16. Calculate and plot:
(a) Its autocorrelation function (in dB).
(b) Its spectrum (in dB).
(c) Its ambiguity function.

REFERENCES
REFERENCES


Lindner J., Binary sequences up to length 40 with best possible autocorrelation function, *Electronics Letters*, vol. 11, no. 21, 16 October 1975, p. 507.


Welti, G. R., Quaternary codes for pulsed radar, *IRE Transactions on Information Theory*, vol. IT-6, June 1960, pp. 400–408.

