Chapter 2: Pseudorandom sequences

Toni Huovinen
Tampere University of Technology
toni.huovinen@tut.fi

Slides are mainly based on the original lecture notes by prof. Tapani Ristaniemi.
Basic synchronous CDMA model

- A baseband model for a $K$-user system with additive white Gaussian noise:

$$y(t) = \sum_{k=1}^{K} A_k b_k s_k(t) + \sigma n(t), \quad t \in [0, T]$$  \hspace{1cm} (1)

- $T$ is the inverse of the data rate
- $s_k(t)$ is the unit-energy ($\| s_k(t) \|^2 = \int_0^T s_k(t)^2 dt = 1$) deterministic signature waveform of the user $k$, supported by $[0, T]$.
- $A_k$ is the received amplitude of the user $k$
- $b_k \in \{-1, 1\}$, a bit transmitted by user $k$
- $n(t)$ is white Gaussian noise with unit power spectral density.
The performance of demodulation of a particular user depends (given that the phase of the signature sequence is known at the receiver) mainly on the signal-to-noise ratio (SNR) of a particular user, $A_k/\sigma$, and degree of similarity between the signature waveforms.

- **Similarity measure: crosscorrelation $\rho_{kl}$**:

$$\rho_{kl} = \langle s_k, s_l \rangle = \int_0^T s_k(t)s_l(t)dt$$  \hspace{1cm} (2)

- **Properties**:
  - $\rho_{kk} = 1$ for any $k$
  - $\rho_{kl} \leq 1$
Basic asynchronous CDMA model

- The synchronous model (1) is also called a one-shot model, since it is sufficient to restrict attention only to the received waveform in an interval of a bit.
- In case where transmissions of different users are not aligned in time, an asynchronous model arises:

\[
y(t) = \sum_{k=1}^{K} \sum_{i=0}^{M} A_k b_k[i] s_k(t - iT - \tau_k) + \sigma n(t)
\]  

(3)

- Here, \( \tau_k \) defines a (relative) delay of the \( k \)-th user.
- The model takes into account \( M + 1 \) symbols (a block or a frame), since due to asynchronism the demodulation of a particular user’s symbol is now affected by more than one symbol of a interfering user.
• The crosscorrelation just defined for a synchronous model is no longer sufficient
• Asynchronous crosscorrelations are defined as

\[ \rho_{kl}^{-}(\tau) = \int_{0}^{\tau} s_{k}(t) s_{l}(t + T - \tau) dt \]  

(4)

\[ \rho_{kl}^{+}(\tau) = \int_{\tau}^{T} s_{k}(t) s_{l}(t - \tau) dt \]  

(5)
Signature waveforms

Direct sequence SS

- Direct-sequence refers to a specific approach to construct SS waveforms, characterized by
  - chip waveform $p_{T_c}$ with “duration” $T_c$ such that
    \[
    \int_{-\infty}^{\infty} p_{T_c}(t)p_{T_c}(t - nT_c)\,dt = 0, \quad n = 1, 2, \ldots
    \]  
  - $N$, the number of chips per bit
  - binary sequence $(c_1, \ldots, c_N)$
If the binary sequence is used to modulate the chip antipodally, we obtain the DS-SS waveform with duration $NT_c$:

$$s(t) = A \sum_{i=1}^{N} (-1)^{c_i} p_{T_c}(t - (i - 1)T_c)$$  \hspace{1cm} (7)

where $T_c$ is the duration of a chip.

Examples of chip waveforms:

- **rectangular**
  
  $$p_{T_c}(t) = \begin{cases} 
  1, & \text{if } 0 \leq t < T_c \\
  0, & \text{otherwise} 
  \end{cases}$$  \hspace{1cm} (8)

- **sinc**
  
  $$p_{T_c}(t) = \text{sinc}\left(\frac{2t}{T_c} - 1\right)$$  \hspace{1cm} (9)

These are examples of time-limited and bandwidth-limited waveforms.
Variants of the latter chip waveform are used mostly in practice due to the achieved spectral efficiency. For example, raised cosine,

\[
p_\beta(t) = \text{sinc} \left( \frac{t}{T_c} \right) \frac{\cos \left( \frac{\beta \pi t}{T_c} \right)}{1 - \left( \frac{2\beta t}{T_c} \right)^2}
\]

has a band-width \((1 + \beta)/(2T_c)\), \(0 \leq \beta \leq 1\).
**Spreading factor**

The number of chips $N$ per bit is also called *spreading factor* or a *processing gain*, and it has the following properties:

- Bandwidth of the DS-SS signal is proportional to $N$ (for a fixed duration of the signature waveform)
- For a given signal-to-noise ratio ($A/\sigma$), the single-user bit-error-rate in a white Gaussian noise is independent of $N$
- If $N$ is a power of 2, an orthogonal synchronous CDMA system can support $N$ users
- Large values on $N$ contribute to the privacy of the system, as it makes eavesdropping more difficult
- Large values contribute to reduce the interference caused on/by a co-existing narrowband transmission (cf. CDMA overlay)
- The reliability for chip timing determination increases with $N$
Signature sequences

- In a DS-CDMA system, the number of chips per bit and the chip waveform are the same for each user (given equal bit-rates).
- What distinguishes different signature waveforms (that is, users in the system) is the assignment of the chip sequence $c_1, \ldots, c_N$.
- Therefore sequence design is one of the most important issues in spread spectrum systems.
- There are several obvious questions regarding the issue of sequence design:
  - How to choose a spreading sequence?
  - How to generate a spreading sequence?
  - How to choose and generate a set of sequences when more than one sequences are needed, for example, in CDMA systems?
The questions are not independent. For example, practical constraint on sequence generation may limit the set of sequence we can choose from.

To start with, let’s first list the most important properties we want the spreading sequences to possess:

- The elements of a chip sequence should behave like i.i.d. random variables, i.e., the sequence should be pseudo-random.
- It should be easy to distinguish a spreading signal from a time-shifted version of it.
- A chip sequence should be easy to distinguish from other sequences, including time-shifted versions of them.
- It should be easy for the transmitter and the intended receiver to generate chip sequence.
- It should be difficult for any unintended receiver to acquire and regenerate the chip sequence.
**Shift register**

- Due to the restriction imposed by the fourth property above, chip sequences are usually generated by feedback shift registers in practice since shift registers are easy to build.
- The spreading sequences generated by feedback shift registers are periodic and they are usually pseudo-random.
- The figure shows Fibonacci implementation of Linear Feedback Shift Register (LFSR)
‘⊕’ represents modulo-2 addition (XOR gate), and $h_i$’s take values either 1 or 0. (connection / no connection).

Hence, the output of the shift register is

$$u_{l+n} = h_n u_l \oplus h_{n-1} u_{l+1} \oplus \cdots \oplus h_1 u_{l+n-1} \quad (11)$$

In fact, we always choose $h_0 = h_n = 1$. If $h_0 = 0$, the output values of the register would be all 0 after $n$ shifts. On the other hand, if $h_n = 0$ we could remove the first storage element without changing the output sequence.

‘□’ represents a single binary storage element, resulting in maximum of $2^n$ different states for a shift register.

Therefore, the output is periodic, with a period determined by the tap values ($h_i$’s) and initial fill of the storage elements.
An example: Let $n = 4$ and $h_0 = h_1 = h_4 = 1$ and $h_2 = h_3 = 0$. Given an initial fill ("state 1") equal to "0001" the other states are:

- state 1: 0 0 0 1
- state 2: 0 0 1 1
- state 3: 0 1 1 1
- state 4: 1 1 1 1
- state 5: 1 1 1 0
- state 6: 1 1 0 1
- state 7: 1 0 1 0
- state 8: 0 1 0 1
- state 9: 1 0 1 1
- state 10: 0 1 1 0
- state 11: 1 1 0 0
- state 12: 1 0 0 1
- state 13: 0 0 1 0
- state 14: 0 1 0 0
- state 15: 1 0 0 0
- state 16: 0 0 0 1

* States 1 and 16 are equal → the period of the output is 15, and the resulting sequence equals 000111101011001.
Polynomial representations of LFSRs and sequences

Our goal now is to derive some of the key properties that a sequence generated by a linear feedback shift register (LFSR) have.

- First, let’s start with a LFSR output $u_l$. From Eq. 11 we see that (replacing $l$ with $l - n$)

$$u_l = h_1 u_{l-1} + h_2 u_{l-2} + \cdots + h_n u_{l-n} = \sum_{i=1}^{n} h_i u_{l-i} \quad (12)$$

- Considering only the non-negative indexes $l$, the sequence $u_l = (u_0, u_1, u_2, \ldots)$ can be expressed as a polynomial $G(D)$ as follows:

$$G(D) = u_0 + u_1 D + u_2 D^2 + \cdots = \sum_{l=0}^{\infty} u_l D^l, \quad (13)$$

where $D$ is a unit delay operator, and the power of $D$ correspond to the number of unit delays.
For example, a binary sequence “10011” can be expressed as a polynomial $G(D) = 1 + D^3 + D^4$.

Combining Eqs. 12 and 13 we have

\[
G(D) = \sum_{l=0}^{\infty} u_l D^l = \sum_{l=0}^{\infty} \sum_{i=1}^{n} h_i u_{l-i} D^l \\
= \sum_{i=1}^{n} h_i D^i \left[ \sum_{l=0}^{\infty} u_{l-i} D^{l-i} \right] \\
= \sum_{i=1}^{n} h_i D^i \left[ u_{i} D^{-i} + \cdots + u_{-1} D^{-1} + G(D) \right]
\]

Thus, we have

\[
G(D) \left( 1 - \sum_{i=1}^{n} h_i D^i \right) = \sum_{i=1}^{n} h_i D^i \left[ u_{i} D^{-i} + \cdots + u_{-1} D^{-1} \right]
\]
From this we can solve $G(D)$ in closed form:

$$G(D) = \frac{\sum_{i=1}^{n} h_i D^i [u_{-i} D^{-i} + \cdots + u_{-1} D^{-1}]}{1 - \sum_{i=1}^{n} h_i D^i}$$  \hspace{1cm} (18)$$

We thus have $G(D)$ defined as a ratio of finite polynomials, defined as

$$G(D) \triangleq \frac{g_0(D)}{f(D)}$$  \hspace{1cm} (19)$$

$f(D)$ is called a characteristic polynomial of the shift register; it depends only on the connection variables $h_i$.

g_0(D)$ depends further on the initial conditions of the register $(u_{-n} \cdots u_{-2} u_{-1})$. That is, the initial state of the shift register just before $u_0$ is generated (but not outputted yet).
• Rewriting \( g_0(D) \) gives

\[
g_0(D) = \sum_{i=1}^{n} h_i D^i [u_{-i} D^{-i} + \cdots + u_{-1} D^{-1}] \quad (20)
\]

\[
= \sum_{i=1}^{n} h_i [u_{-i} + u_{-i+1} D + \cdots + u_{-1} D^{i-1}] \quad (21)
\]

\[
= h_1 u_{-1} \quad (22)
\]

\[
+ h_2 (u_{-2} + u_{-1} D) \quad (23)
\]

\[
+ h_3 (u_{-3} + u_{-2} D + u_{-1} D^2) \quad (24)
\]

\[
+ \cdots \quad (25)
\]

\[
+ h_n (u_{-n} + u_{-n+1} D + \cdots + u_{-1} D^{n-1}) \quad (26)
\]

• Hence, \( g_0(D) \) is not necessarily exactly the polynomial representation for the initial fill. For the time being we consider the initial fill \( u_{-n} = 1, u_{-n+1} = \cdots = u_{-1} = 0 \) giving \( g_0(D) = 1 \).
Example

Suppose $f(D) = 1 + D + D^3$. Draw the related shift register configuration and find the output sequence by polynomial division with initial condition $g_0(D) = 1$. You can check the result by a truth table.

The division gives

$$G(D) = \frac{1}{1 + D + D^3} = 1 + D + D^2 + D^4 + D^7 + D^8 + D^9 + D^{11} + \cdots \quad (27)$$

giving hence the binary output 111010011101 \cdots
Maximal length LFSR sequences

- What is the maximum period of the output?
- Recall that the output of the shift register is the sequence of successive values for the storage element $u_l$.
- Thus, to have a maximum period for the output, the shift register of $n$ storage elements have to experience all the possible binary combinations, that is, $2^n$ of them.
- Disregarding the undesired state of the shift register (‘all zeros’ resulting in an ‘all zeros’ output for the register), the periodicity of the register is upper bounded by $2^n - 1$.
- Sequences that achieve that bound are called maximal-length sequences, or shortly $m$-sequences.
- Polynomial $f(D)$ which generates an $m$-sequence, i.e., characteristic polynomial of an $m$-sequence, is called primitive.
Properties of m-sequences

**Property A:** A maximal length sequence contains one more “1” than “0”.

- This is also called a “Balanced property”: there are exactly $2^{n-1}$ ones and $2^{n-1} - 1$ zeros.
- This is because from all the possible $2^n - 1$ states of a shift register plus the “all zeros” state, exactly one half of them begin with 0 and the other half begin with 1. Thus, excluding the forbidden state, we have the Property A.

Notice, that if we map the binary values $u_l$ into the values $\pm 1$ according to $c_l = (-1)^{ul}$, then the Property A equals the following temporal property:

$$\frac{1}{N} \sum_{i=1}^{N} c_i = -\frac{1}{N} \quad (28)$$

Thus, the imbalance from a zero-mean variable is $N^{-1}$, which for $n = 10, 30,$ and $50$ is approximately $10^{-3}, 10^{-9},$ and $10^{-15},$ respectively.
Property B: The sum of an m-sequence and any phase shift of the same m-sequence is another phase of the same m-sequence.

- Suppose $G_0(D)$ is the original sequence and $G_\tau(D)$ is the shifted one, both of length $N = 2^n - 1$.
- Denote $g_0(D)$ and $g_\tau(D)$ their initial conditions.
- Then we have
  \[ G_0(D) = \frac{g_0(D)}{f(D)}, \quad G_\tau(D) = \frac{g_\tau(D)}{f(D)}, \]  
  where $f(D)$ is the primitive for both sequences.
- Hence, $G_0(D) + G_\tau(D) = \frac{g_0(D) + g_\tau(D)}{f(D)}$.
- This means that the sum sequence is an m-sequence generated by the very same primitive $f(D)$ but with a different initial condition $g_0(D) + g_\tau(D)$.
- Hence, $G_0(D) + G_\tau(D)$ is itself another phase of $G_0(D)$.
Property B (cont’d)

Property B is also called a “Shift-and-add property”

- If we map the binary values $u_l$ into the values $\pm 1$ according to $c_l = (-1)^{u_l}$, then the Property B equals the following temporal property:

$$\frac{1}{N} \sum_{i=1}^{N} c_i c_{i+j} = -\frac{1}{N}$$  \hspace{1cm} (30)

- Thus, m-sequences have good autocorrelation properties. However, crosscorrelations with other m-sequences are not always that good.
Security of m-sequence

- SS systems are used to protect digital transmission from being jammed. This is achieved if the jammer has no knowledge of the spreading waveform.
- Unfortunately, if the jammer receives a relatively noise-free copy of the transmitted signal, the related shift register is pretty easy to be determined. For this reason, short m-sequences are a poor choice.
- How to find out the LFSR used by the transmitter? Recall that m-sequence satisfies Eq. 12:

\[ u_l = h_1 u_{l-1} + h_2 u_{l-2} + \cdots + h_n u_{l-n} = \sum_{i=1}^{n} h_i u_{l-i} \]  

(31)

- As soon as we have \( n \) such equations, \( h_i \)'s can be easily found.
**Security of m-sequence: example**

Suppose a sequence “00100110...” is received, and we have a knowledge that the period is 15. Hence, \( n = 4 \) and we have a set of equations:

\[
\begin{align*}
0 &= h_1 \times 1 + h_2 \times 1 + h_3 \times 0 + h_4 \times 0 \quad (32) \\
1 &= h_1 \times 1 + h_2 \times 0 + h_3 \times 0 + h_4 \times 1 \\
1 &= h_1 \times 0 + h_2 \times 0 + h_3 \times 1 + h_4 \times 0 \\
0 &= h_1 \times 0 + h_2 \times 1 + h_3 \times 0 + h_4 \times 0 \quad (35)
\end{align*}
\]

From Eq. 34 we have \( 1 = h_3 \times 1 = h_3 \), and from Eq. 35 we have \( 0 = h_2 \times 1 = h_2 \). Substituting \( h_2 = 0 \) and \( h_3 = 1 \) to Eqs. 32 and 33 we have \( h_1 = 0 \) and \( h_4 = 1 \).

Therefore, remembering that \( h_0 = 1 \) always, we have \( f(D) = 1 + D^3 + D^4 \).
Gold sequences

- In CDMA systems large sets of codes with good auto- and crosscorrelation properties are needed.

- M-sequences have very good autocorrelation properties, but cross-correlations with two m-sequences (generated by two different LFSR’s) might be poor.

- In fact, the cross-correlation spectrum of pairs of m-sequences might be three-valued, four-valued, or possibly many-valued.

- Certain special pairs, called preferred pairs have three-valued cross-correlation:

\[-\frac{1}{N}t(n), \quad \frac{1}{N}, \quad \frac{1}{N}[t(n) - 2],\]

where

\[t(n) = \begin{cases} 
1 + 2^{0.5(n+1)}, & \text{for odd } n \\
1 + 2^{0.5(n+2)}, & \text{for even } n
\end{cases}\]

- For example, with \(n = 5, N = 2^5 - 1 = 31\) the three-valued cross-correlations would be equal to \(-\frac{9}{31}, \frac{1}{31}, \frac{7}{31}\)
Decimation - a way to find a preferred pair for a m-sequence

- If every $q$th value of a sequence $G(D)$ is taken and a new sequence $G(D)^I$ is built, $G(D)^I$ is called a decimation of $G(D)$ and denoted $G(D)^I = G(D)[q]$.
- In fact, it has been shown, that any pair of m-sequences of equal period are related by a decimation with some $q$.
- However, only few $q$'s result in preferred pairs. The criteria are:
  - D1: $n \neq 0 \mod 4$ (that is, 4 is not a factor of $n$)
  - D2: $q$ is odd and either of the form $q = 2^k - 1$ or $2^{2k} - 2^k + 1$ for some $k$.
  - D3: Greatest common divisor of $(n, k)$ is either 1 (for $n$ odd) or 2 (for $n = 2 \mod 4$)
- Then, given a preferred pair $G(D)$ and $G(D)^I$, a family of sequences defined by
  \[
  \{G(D), G(D)^I, G(D) + G(D)^I, G(D) + DG(D)^I, ..., G(D) + D^2 G(D)^I, ..., G(D) + D^{N-1} G(D)^I\}
  \]
  is called the set of Gold codes.
- Notice that there are $N + 2 = 2^n + 1$ codes in the set.
Example 1: Gold code generator

A typical shift register configuration used to generate Gold sequences.
Example 2: Cross-correlation spectrum

Find a preferred pair for a m-sequence $G(D) \ "1110100\ "$ and evaluate their cross-correlations. (" ") is now used just to shorten notations.

Since $N = 7$ and $n = 3$. Criterion D1 is fulfilled, since $3 = 3 \mod 4$. Take $q = 3$ which satisfies D2 and D3. Now $G(D)[3]$ equals "1001011"

The set of Gold sequences now equals:

\[
\begin{align*}
G(D) & \quad 1110100 \\
G(D)[3] & \quad 1001011 \\
G(D) + G(D)[3] & \quad 0111111 \\
G(D) + DG(D)[3] & \quad 1100011 \\
G(D) + D^2G(D)[3] & \quad 1011010 \\
\cdots & \quad \cdots \\
G(D) + D^6G(D)[3] & \quad 0010001
\end{align*}
\]

A straightforward but tedious manual calculation will show that, given any pair of gold codes, any phase shift gives a three-valued cross-correlation value $-\frac{5}{7}, -\frac{1}{7}, \frac{3}{7}$. 
Example 3: Cross-correlation spectrum

Gold codes of length 31. In the figure one code is selected as a reference code and the related auto- and crosscorrelations with respect to the other 30 codes are presented.
Kasami sequences

- As seen previously, combining preferred pairs of m-sequences results in a set of Gold sequences which have much better cross-correlation properties than m-sequences alone.
- Kasami sequences are also derived from m-sequences and their cross-correlation properties are similar to Gold codes.
- Shortly, Kasami sequence is got by combining a Gold sequence and a decimated version of one of the two m-sequences that form the Gold sequence.
- Such a sequence can be expressed in more details as

\[ G(D) = G_1(D) + G_2(D) + G_i(D)[q] \]  

(40)

Here \( G_1 \) and \( G_2 \) is a preferred pair of m-sequences of length \( 2^n - 1 \). \( G_3 \) is a decimated version of either \( G_1 \) or \( G_2 \), where \( q \) is chosen to be \( q = 2^{n/2} + 1 \).  

- Sequences generated according to Eq. 40 form a large set of Kasami sequences. The number of sequences is \( 2^{n/2}(2^n + 1) = \sqrt{N+1}(N + 2) \).
Orthogonal code sequences

- Gold and Kasami sequences have rather good cross-correlation properties, which make them attractive in CDMA systems.
- So, are pure orthogonal codes of any use?
- Orthogonal codes usually have poor autocorrelation properties making them useless in asynchronous access.
- However, if the transmissions would be synchronous, the users are completely transparent for each others given a single-path channel.
- But now the existence of multi-paths would cause strong ISI/IPI ;(
- To circumvent that, one solution to maintain orthogonality is chip-equalization. This means that the channel is equalized into a single-path channel prior to despreading, which thus restores orthogonality between the users.
- Hence, orthogonal codes would be a sensible choice in downlink (base-to-mobile) communication provided that mobile units perform chip-equalization.
One way to generate orthogonal codes is to apply the following recursive procedure.

\[ H_1 = 1, \quad H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \]  

(41)

\( H_N \) is called a Hadamard matrix of dimension \( N \times N \).

For example,

\[ H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \]  

(42)

are Hadamard matrices having 2 and 4, respectively, mutually orthogonal sequences on their columns.

The recursive procedure with orthogonal matrices results in a very useful property of \( H_N \). Namely, each column of \( H_N \) can be viewed as a repetition of shorter orthogonal code, that is 2 successive orthogonal codes of length \( N/2 \), or 4 successive orthogonal codes of length \( N/4 \), and so on.
- This enable orthogonal access of multiple users with variable spreading factor, which, in turn, enable the system to support a variety of data services (bit rates).

- This is why these codes are also called OVSF-codes (Orthogonal Variable Spreading Factor), and in practise they are generated in a slightly different manner:

- OVSF-codes are used in WCDMA downlink channelization.
**Theory and practice: Sequences in WCDMA**

<table>
<thead>
<tr>
<th></th>
<th>Channelization code</th>
<th>Scrambling code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Usage</strong></td>
<td>UL: separation of data and control channels</td>
<td>UL: separation of terminals</td>
</tr>
<tr>
<td></td>
<td>DL: separation of DL connections</td>
<td>DL: separation of sectors/cells</td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>4 – 256 chips</td>
<td>UL: 256 or 38400 chips in frame</td>
</tr>
<tr>
<td></td>
<td>DL also 512 chips</td>
<td>DL: 38400 chips</td>
</tr>
<tr>
<td><strong>Number of codes</strong></td>
<td>4 – 512</td>
<td>UL: several millions, DL: 512</td>
</tr>
<tr>
<td><strong>Code family</strong></td>
<td>OVSF-codes</td>
<td>Gold codes (long, 38400 chips) and Kasami codes (short, 256 chips)</td>
</tr>
<tr>
<td><strong>Spreading</strong></td>
<td>Increases bandwidth</td>
<td>Does not increase bandwidth</td>
</tr>
</tbody>
</table>

Diagram:
- **channelization codes** and **scrambling codes**
- Data flow with **bit rate** and **chip rate**

- Channelization code: Spreads signals to increase bandwidth
- Scrambling code: Does not increase bandwidth