# ECS455: Chapter 4

## Multiple Access

**4.4 m-sequence**

Dr.Prapun Suksompong
prapun.com/ecs455

**Office Hours:**
**BKD, 6th floor of Sirindhralai building**

| | |
|---|---|
| **Tuesday** | **14:20-15:20** |
| **Wednesday** | **14:20-15:20** |
| **Friday** | **9:15-10:15** |



## Binary Random Sequences

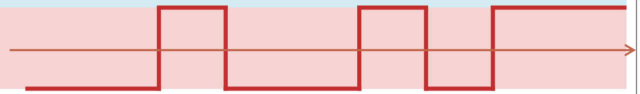
- While DSSS chip sequences must be generated *deterministically*, properties of binary random sequences are useful to gain insight into deterministic sequence design.
- A random binary chip sequences consists of i.i.d. bit values with probability one half for a one or a zero.
  - Also known as **Bernoulli sequences/trial**s, "coin-flipping" sequences
- A random sequence of length $N$ can be generated, for example, by flipping a fair coin $N$ times and then setting the bit to a one for heads and a zero for tails.



## Binary Random Sequence

| | $X_{-4}$ | $X_{-3}$ | $X_{-2}$ | $X_{-1}$ | $X_{-0}$ | $X_1$ | $X_2$ | $X_3$ | $X_4$ |
|---|---|---|---|---|---|---|---|---|---|
| Coin-flipping sequence | H | H | T | H | H | T | H | T | T |
| Bernoulli trials/sequence | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| Binary (indp.) random sequence | -1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 |

Waveform



- These names are simply many versions of the same sequence/process.
- You should be able to convert one version to others easily.
- Some properties are conveniently explained when the sequence is expressed in a particular version.



## Properties of Binary Random Sequences:

- Consider the sequence $X_1, X_2, X_3, \dots, X_n, \dots$
- Disadvantages
  - Can not further "compress" the sequence
  - Difficult to convey the sequence from the Tx to Rx
  - Require large storage at both Tx and Rx
- Advantages
  - Random = unpredictable
  1. Balanced property
  2. Run length property
  3. Shift property

## Properties of Binary Random Sequences: Balanced Property

- $\{0,1\}$ version

$$\frac{1}{N}\sum_{i=1}^{N} X_i \xrightarrow[\text{LLN}]{N\to\infty} \mathbb{E}[X_i] = 0\times\frac{1}{2} + 1\times\frac{1}{2} = \frac{1}{2}$$

- $\{\pm 1\}$ version

$$\frac{1}{N}\sum_{i=1}^{N} X_i \xrightarrow[\text{LLN}]{N\to\infty} \mathbb{E}[X_i] = (-1)\times\frac{1}{2} + 1\times\frac{1}{2} = 0$$



## Runs: An Example

- A run is a subsequence of consecutive identical symbols within the sequence.
- The following sequence contains 16 runs

000111110011010010000101010111011

A run of 1s (length = 5)

A run of 0s (length = 3)

- Rel. Freq of Run Lengths

| Run Length | Rel. Freq. |
|---|---|
| 5 | 1/16 |
| 4 | 1/16 |
| 3 | 2/16 |
| 2 | 4/16 |
| 1 | 8/16 |

- Rel. Freq of Runs

| | |
|---|---|
| 11111 | 1/16 |
| 0000 | 1/16 |
| 111 | 1/16 |
| 000 | 1/16 |
| 11 | 2/16 |
| 00 | 2/16 |
| 1 | 4/16 |
| 0 | 4/16 |



## Properties of Binary Random Sequences: Run Length Property

011001111100001

Start of a run of 1s

$$P[\text{run length} = \ell] = \frac{1}{2^{\ell}}$$ ➡ As $\ell$ increases, $P$ is reduced.

➡ Small probability of having log runs.



## FYI: Run-Length Encoding (RLE)

- A very simple form of lossless data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run.
- Most useful on data that contains many such runs.
- Example: Consider a screen containing plain black text on a solid white background.
  A line, with B representing a black pixel and W representing white, might read as follows:

  WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW

  With a RLE data compression algorithm applied to the above line, it can be rendered as follows:

  12W1B12W3B24W1B14W

# Properties of Binary Random Sequences: Shift Property

Original sequence: $X_1 \quad X_2 \quad X_3 \quad X_4 \quad \cdots \quad X_N$

Shifted sequence: $X_{-1} \quad X_0 \quad X_1 \quad X_2 \quad \cdots \quad X_{N-2}$

(amount of shift (delay) = 2)

- When the shifted amount = 0, the two sequences are exactly the same.
- When the shifted amount = $s$, we want to compare $X_j$ and $X_{j-s}$.
  - What proportion are the same?
  - What proportion are different?
- Recall that the numbers in the sequence are independent results (from several Bernoulli trials)

| $X_j$ | $X_{j-s}$ | Probability |
|---|---|---|
| 0 | 0 | ¼ |
| 0 | 1 | ¼ |
| 1 | 0 | ¼ |
| 1 | 1 | ¼ |



# Properties of Binary Random Sequences: Shift Property

- $\{0,1\}$ version: The comparison is done via the XOR ($\oplus$) operation
  - $x \oplus y = 0$ iff they are the same
  - $x \oplus y = 1$ iff they are different
- $\{\pm 1\}$ version: The comparison is done via the multiplication operation
  - $x \times y = 1$ iff they are the same
  - $x \times y = -1$ iff they are different



# Key randomness properties

[Golomb, 1967][Viterbi, 1995, p. 12] Binary random sequences with length $N$ asymptotically large have a number of the properties desired in spreading codes

- **Balanced property**: Equal number of ones and zeros.
  - Should have no DC component to avoid a spectral spike at DC or biasing the noise in despreading
- **Run length property**: The run length is generally short.
  - half of all runs are of length 1
  - a fraction $1/2^n$ of all runs are of length $n$ (Geometric)
  - Long runs reduce the BW spreading and its advantages
- **Shift property**: If they are shifted by any nonzero number of elements, the resulting sequence will have half its elements the same as in the original sequence, and half its elements different from the original sequence.

[Goldsmith, 2005, p. 387 & Viterbi, p. 12]



# Pseudorandom Sequence

- A deterministic sequence that has the balanced, run length, and shift properties as it grows *asymptotically large* is referred to as a **pseudorandom sequence** (noiselike or pseudonoise (PN) signal).
- Ideally, one would prefer a random binary sequence as the spreading sequence.
- However, practical synchronization requirements in the receiver force one to use **periodic** Pseudorandom binary sequences.
- m-sequences
- Gold codes
- Kasami sequences
- Quaternary sequences
- Walsh functions

## m-Sequences

- **Maximal-length sequences**

Longer name: Maximal length linear shift register sequence.

- A type of **cyclic code**
  - Generated and characterized by a generator polynomial
  - Properties can be derived using algebraic coding theory

[Goldsmith, 2005, p 387]

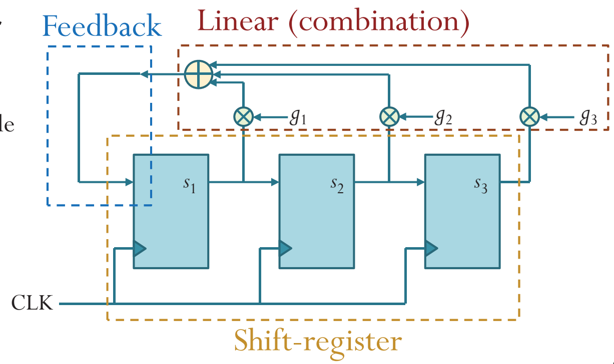
- Simple to generate with **linear feedback shift-register** (**LFSR**) circuits
  - Automated
- Approximate a random binary sequence.
- Disadvantage: Relatively easy to intercept and regenerate by an unintended receiver

[Ziemer, 2007, p 11]



## Linear Feedback Shift-Register (LFSR)

- Binary sequences drawn from the alphabet $\{0,1\}$ are shifted through the shift register in response to clock pulses.
  - Each clock time, the register shifts all its contents to the right.
- The particular 1s and 0s occupying the shift register stages after a clock pulse are called **states**.
- Suppose there are $r$ FFs. Then a state $\underline{\mathbf{s}}$ of the SR can be represented by $r$ bits.
  - There are $2^r$ possible states.
  - There are $2^r - 1$ non-zero states.






## GF(2)

- **Galois field** (finite field) of two elements
- Consist of
  - the symbols 0 and 1 and
  - the (binary) operations of
    - **modulo-2** addition (XOR) and
    - **modulo-2** multiplication.
- The operations are defined by

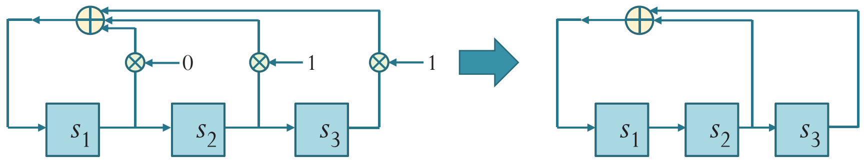
| + | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

| · | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |



## Linear Feedback Shift-Register (LFSR)

- All the values are in GF(2) which means they can only be 0 or 1.
- The value of $g_i$ determines whether the output of the $k^{th}$ FF will be in the sum that produce the feedback bit.
  - 1 signifies closed or a connection and
  - 0 signifies open or no connection.
- Ex. Suppose $g_1 = 0,\ \ g_2 = 1,\ \ g_3 = 1$ in our LFSR.

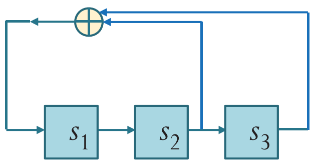


(See Section 13.4.1 in [Lathi, 1998])

# m-sequence generator (1)

- Start with a "**primitive polynomial**"
  - $g(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_r x^r$
  - $r$ = degree of the polynomial
- Use $r$ flip-flops.
- The feedback taps in the feedback shift register are selected to correspond to the coefficients of the primitive polynomial.
- Ex. $g(x) = 1 + x^2 + x^3$ is a primitive polynomial.
  $= 1 + 0x + 1x^2 + 1x^3$

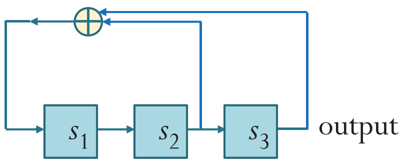
(Degree: r = 3 ➔ use 3 flip-flops)






# m-sequence generator (2)

- We start with state 100.
  - You may choose different non-zero state.
  - Note that if we start with 000, we won't go anywhere.



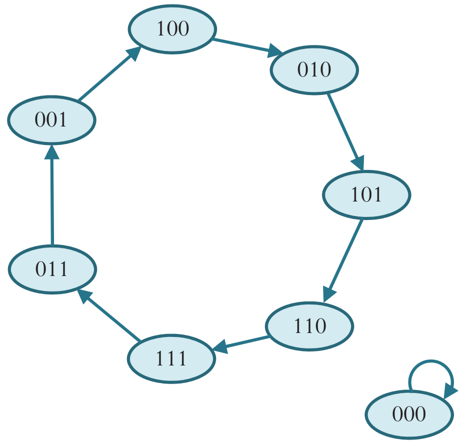

output

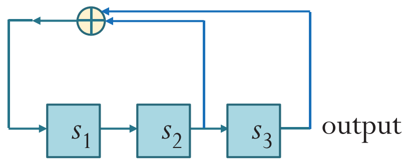
| Time | $s_1$ | $s_2$ | $s_3$ |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |

- Any polynomial generates periodic sequence.
  - The maximum period is $2^r - 1$ .
- In this example, the state cycles through all $2^3 - 1 = 7$ non-zero states.



# State Diagram




output

| Time | $s_1$ | $s_2$ | $s_3$ |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 0 |
| 4 | 1 | 1 | 1 |
| 5 | 0 | 1 | 1 |
| 6 | 0 | 0 | 1 |
| 7 | 1 | 0 | 0 |



# Primitive Polynomial

- Definition: A LFSR **generates an m-sequence** if and only if (starting with any nonzero state,) it visits all possible nonzero states (in one cycle).
- Technically, one can define primitive polynomial using concepts from finite field theory.
- Fact: A polynomial generates m-sequence if and only if it is a primitive polynomial.
  - Therefore, we use this fact to define primitive polynomial.
- For us, a polynomial is **primitive** if **the corresponding LFSR circuit generates m-sequence**.

# Sample Exam Question

Draw the complete **state diagrams** for linear feedback shift registers (LFSRs) using the following polynomials.
Does either LFSR generate an m-sequence?

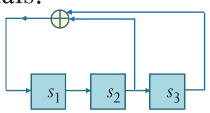
1. $g(x) = 1 + x^2 + x^3$
2. $g(x) = 1 + x + x^2 + x^3$

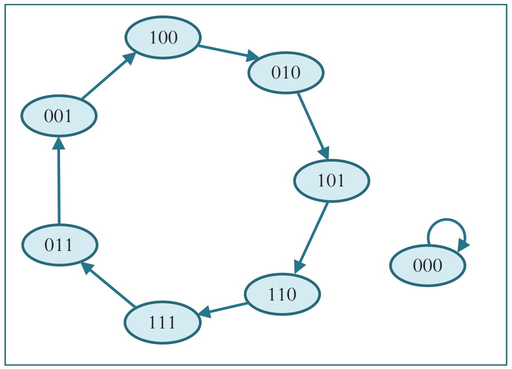


# Solution (1)

Draw the complete **state diagrams** for linear feedback shift registers (LFSRs) using the following polynomials.
Does either LFSR generate an m-sequence?

1. $g(x) = 1 + x^2 + x^3$




The corresponding LFSR **generates an m-sequence** because the state diagram contains a cycle that visits all possible nonzero states.
We can also conclude that $g(x) = 1 + x^2 + x^3$ **is a primitive polynomial**.



# Solution (2)

$g(x) = 1 + x + x^2 + x^3$



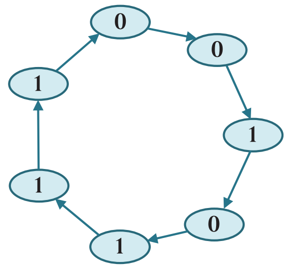
# m-Sequences: More properties

1. The contents of the shift register will cycle over all possible $2^r$-1 nonzero states before repeating.
2. Contain one more 1 than 0 (Slightly unbalanced)
3. **Shift-and-add property**: Sum of two **(cyclic-)shifted** m-sequences is another (cyclic-)shift of the same m-sequence
4. If a window of width $r$ is slid along an m-sequence for $N = 2^r-1$ shifts, each $r$-tuple except the all-zeros r-tuple will appear exactly once
5. For any m-sequence, there are
   - One run of ones of length $r$
   - One run of zeros of length $r$-1
   - One run of ones and one run of zeroes of length r-2
   - Two runs of ones and two runs of zeros of length r-3
   - Four runs of ones and four runs of zeros of length r-4
   - …
   - $2^{r-3}$ runs of ones and $2^{r-3}$ runs of zeros of length 1

# m-Sequences: More Properties

1. The contents of the shift register will cycle over all possible $2^r$-1 nonzero states before repeating.
2. Each cycle contains exactly one more 1s than 0s (Slightly unbalanced)

$g(x) = 1 + x^2 + x^3$




001011100101110010111001011100101110010111001011100101110010111

[S.W. Golomb, *Shift Register Sequences*, Holden-Day, San Francisco, 1967.]



# m-Sequences: More Properties

3. **Shift-and-add property**: Sum of two **(cyclic-)shifted** m-sequences is another (cyclic-)shift of the same m-sequence

0010111001011100101110010111001011100101110010111001011100101110010111

0 phase shift: 0010111
1 phase shift: 0101110
2 phase shift: 1011100
3 phase shift: 0111001
4 phase shift: 1110010
5 phase shift: 1100101
6 phase shift: 1001011

$\oplus$ = 1100101

4. If a window of width $r$ is slid along an m-sequence for $N = 2^r$-1 shifts, each $r$-tuple except the all-zeros r-tuple will appear exactly once

001011100101110010111001011100101110010111001011100101111

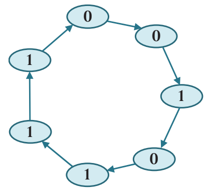
[S.W. Golomb, *Shift Register Sequences*, Holden-Day, San Francisco, 1967.]



# m-Sequences: More Properties

5. For any m-sequence, there are $2^{r-1}$ runs.
   - One run of ones of length $r$
   - One run of zeros of length $r$-1
   - One run of ones and one run of zeroes of length $r$-2
   - Two runs of ones and two runs of zeros of length $r$-3
   - Four runs of ones and four runs of zeros of length $r$-4
   - …
   - $2^{r-3}$ runs of ones and $2^{r-3}$ runs of zeros of length 1

In other words, relative frequency for runs of length $\ell$ is $\begin{cases} \frac{1}{2^\ell}, & \ell < r, \\ \frac{1}{2^{\ell-1}}, & \ell = r. \end{cases}$




0010111001011100101110010111001011100101110010111

Runs:
111
00
1,0

[S.W. Golomb, *Shift Register Sequences*, Holden-Day, San Francisco, 1967.]



# m-Sequences: Another Example

- $2^5$-1 = 31-chip m-sequence
- The following sequence contains 16 runs

0001111100110100100001010111011

- Rel. Freq of Run Lengths

| Run Length | Rel. Freq. |
|---|---|
| 5 | 1/16 |
| 4 | 1/16 |
| 3 | 2/16 |
| 2 | 4/16 |
| 1 | 8/16 |

$\begin{cases} \frac{1}{2^\ell}, & \ell < 5, \\ \frac{1}{2^{\ell-1}}, & \ell = 5. \end{cases}$

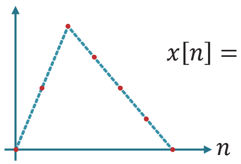
- Rel. Freq of Runs

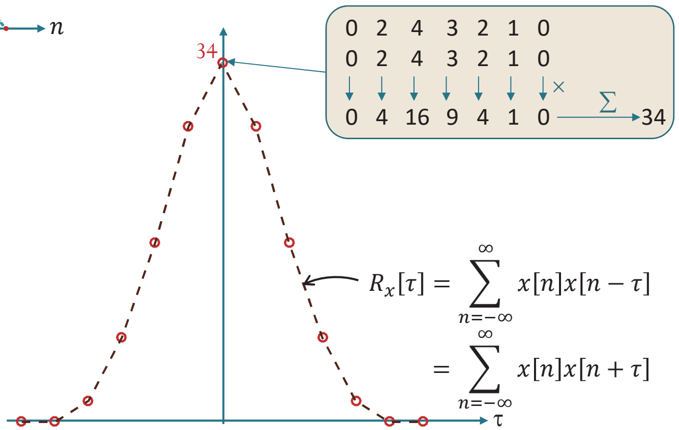
| | |
|---|---|
| 11111 | 1/16 |
| 0000 | 1/16 |
| 111 | 1/16 |
| 000 | 1/16 |
| 11 | 2/16 |
| 00 | 2/16 |
| 1 | 4/16 |
| 0 | 4/16 |

[S.W. Golomb, *Shift Register Sequences*, Holden-Day, San Francisco, 1967.]

## (Time) Autocorrelation Function for Energy Sequence

$x[n] = (0 \quad 2 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0)$



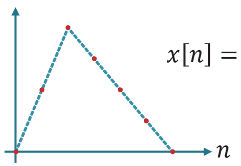



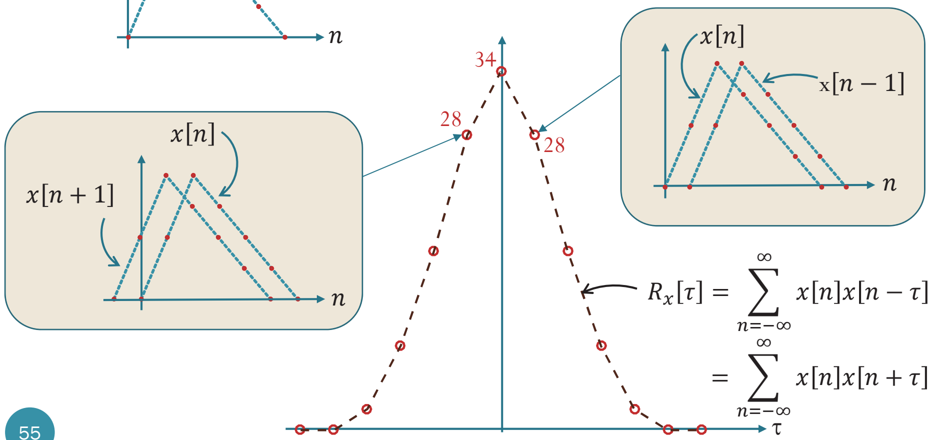

$$R_x[\tau] = \sum_{n=-\infty}^{\infty} x[n]x[n-\tau] = \sum_{n=-\infty}^{\infty} x[n]x[n+\tau]$$



## (Time) Autocorrelation Function for Energy Sequence

$x[n] = (0 \quad 2 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0)$



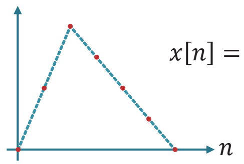



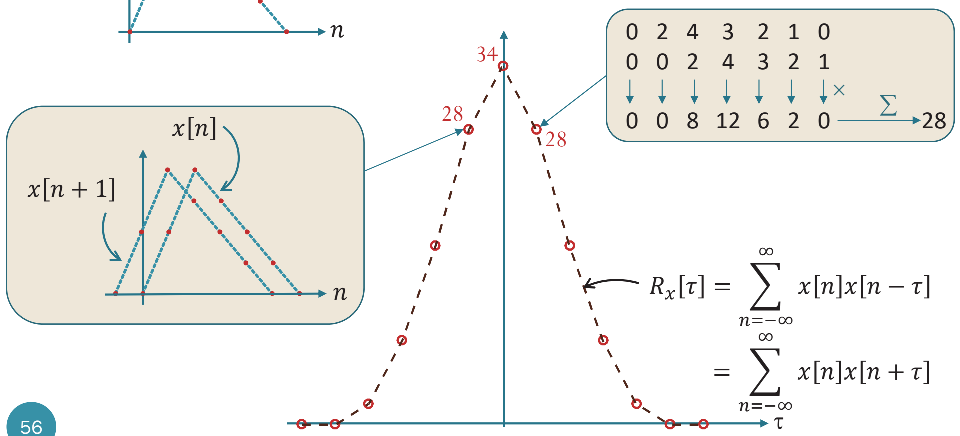

$$R_x[\tau] = \sum_{n=-\infty}^{\infty} x[n]x[n-\tau] = \sum_{n=-\infty}^{\infty} x[n]x[n+\tau]$$



## (Time) Autocorrelation Function for Energy Sequence

$x[n] = (0 \quad 2 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0)$




$$R_x[\tau] = \sum_{n=-\infty}^{\infty} x[n]x[n-\tau] = \sum_{n=-\infty}^{\infty} x[n]x[n+\tau]$$

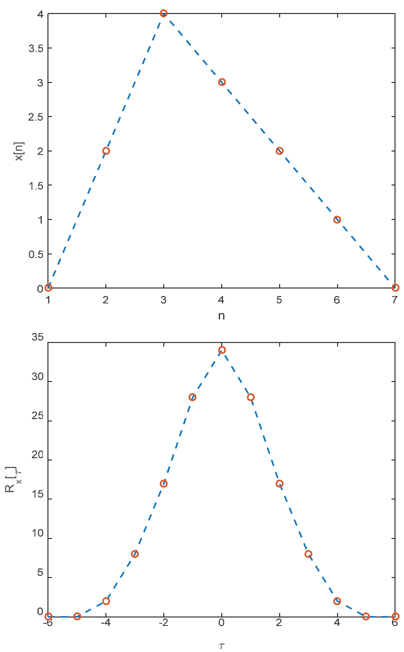


## MATLAB: `xcorr`

- `r = xcorr(x,y)`
  - Return the cross-correlation of two discrete-time sequences, `x` and `y`.
  - If `x` and `y` have different lengths, the function appends zeros at the end of the shorter vector so it has the same length as the other.
  - The lag ($\tau$) is varied from $-(N-1)$ to $(N-1)$ where $N$ is the longer length of the two sequences.
- `[r,lags] = xcorr(___)`
  - Also returns vector with the lags ($\tau$) at which the correlations are computed.

# (Time) Autocorrelation Function for Energy Sequence

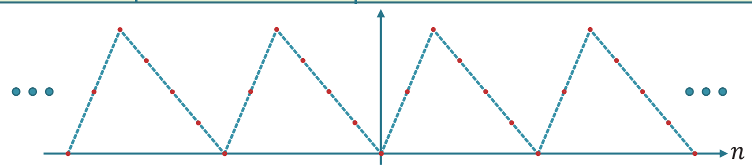
```
close all
x = [0 2 4 3 2 1 0];

% plot the signal
plot(x,'--','LineWidth',1.5)
hold on
plot(x,'o','LineWidth',1.5)
ylabel('x[n]')
xlabel('n')

% plot auto-correlation function
figure
[R lag] = xcorr(x,x);
plot(R,'--','LineWidth',1.5)
hold on
plot(R,'o','LineWidth',1.5)
ylabel('R_x[\tau]')
xlabel('\tau')
```





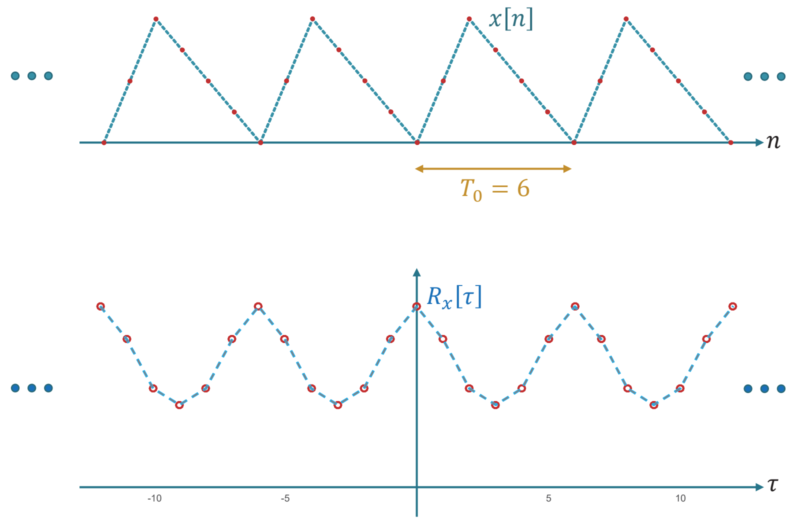

# (Time) Autocorrelation Function for Power and Periodic Sequence

| | Time average $\langle x[n] \rangle$ | Autocorrelation $R_x[\tau]$ |
|---|---|---|
| Power Sequence | $\lim_{T\to\infty} \frac{1}{2T} \sum_{n=-T}^{T} x[n]$ | $\langle x[n]x[n-\tau] \rangle = \lim_{T\to\infty} \frac{1}{2T} \sum_{n=-T}^{T} x[n]x[n-\tau]$ <br> $\langle x[n]x[n+\tau] \rangle = \lim_{T\to\infty} \frac{1}{2T} \sum_{n=-T}^{T} x[n]x[n+\tau]$ |
| Periodic Sequence with period $T_0$ | $\frac{1}{T_0} \sum_{T_0} x[n]$ | $\frac{1}{T_0} \sum_{T_0} x[n]x[n-\tau] = \frac{1}{T_0} \sum_{T_0} x[n]x[n-\tau]$ |



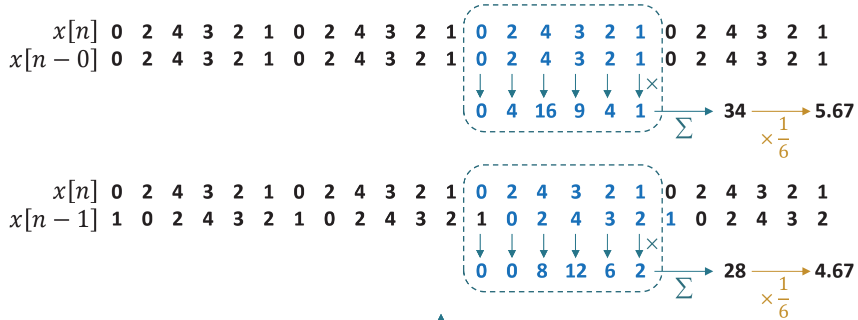


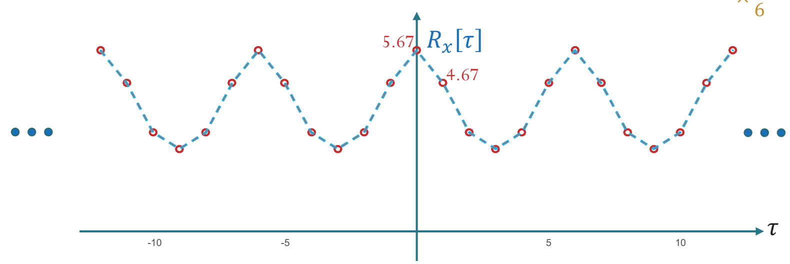

# Example: (Time) Autocorrelation Function for Periodic Sequence






# Example: (Time) Autocorrelation Function for Periodic Sequence



# Back to m-Sequences

$c[n]$: 00101110010111001011100101110010111001011100101110010111

0010111

$\oplus$

1001011

In actual transmission, we will map "0 and 1" to "+1 and -1", respectively.

| | | | | |
|---|---|---|---|---|
| 0 | $\oplus$ | 0 | = | 0 |
| 0 | $\oplus$ | 1 | = | 1 |
| 1 | $\oplus$ | 0 | = | 1 |
| 1 | $\oplus$ | 1 | = | 0 |

| | | | | |
|---|---|---|---|---|
| −1 | ? | −1 | = | −1 |
| −1 | ? | 1 | = | 1 |
| 1 | ? | −1 | = | 1 |
| 1 | ? | 1 | = | −1 |

| | | | | |
|---|---|---|---|---|
| 1 | ? | 1 | = | 1 |
| 1 | ? | −1 | = | −1 |
| −1 | ? | 1 | = | −1 |
| −1 | ? | −1 | = | 1 |



# Back to m-Sequences

$c[n]$: 00101110010111001011100101110010111001011100101110010111

0010111

$\oplus$

1001011

In actual transmission, we will map "0 and 1" to "+1 and -1", respectively.

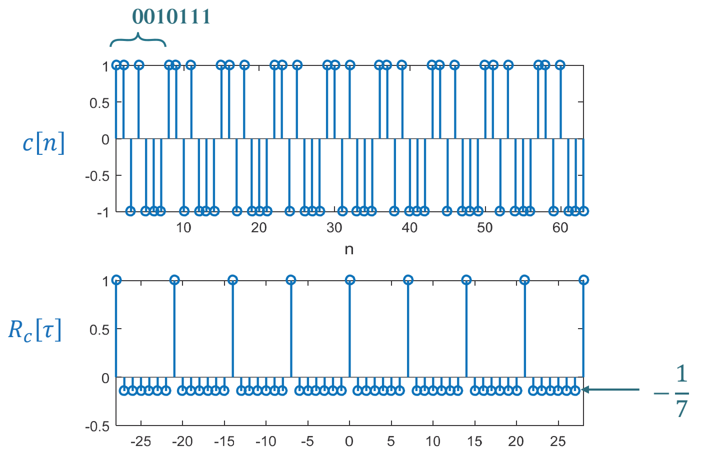
Autocorrelation when not aligned:

```
-1  1 -1 -1 -1  1  1   ×
 1  1 -1  1 -1 -1 -1
---------------------
-1  1  1 -1  1 -1 -1
```

$$\longrightarrow \sum = -1 \xrightarrow[\times \frac{1}{7}]{} -\frac{1}{7}$$

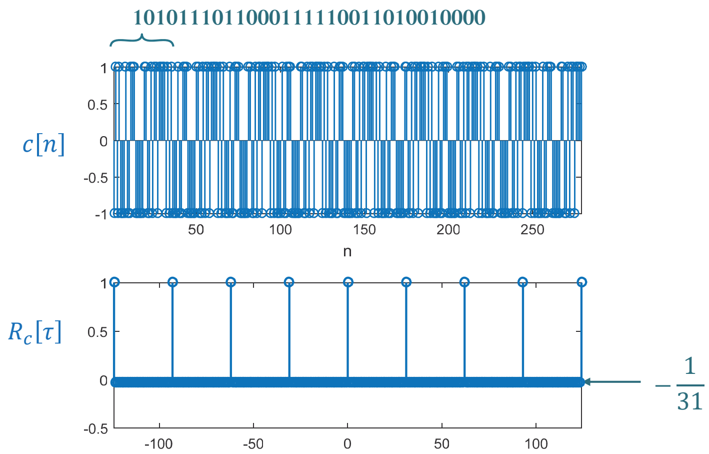


# m-Sequences: Autocorrelation function





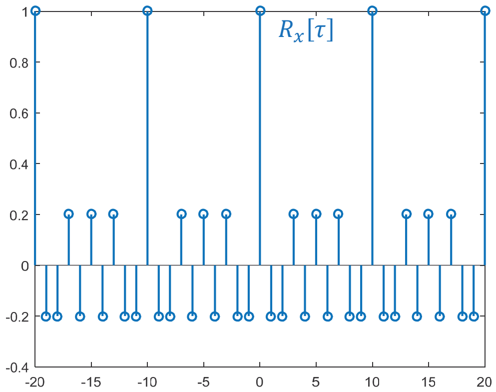

# m-Sequences: Autocorrelation function



# Autocorrelation Function for Periodic Binary Random Sequence

Consider a periodic sequence whose one period is given by

[-1  1  -1  -1  1  -1  1  1  -1  -1]



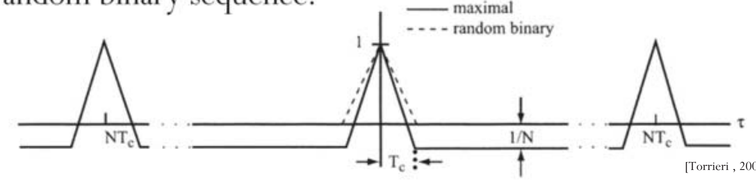

The shift property of binary random sequence implies that

$$R_x[\tau] = \langle x[n]x[n-\tau] \rangle \xrightarrow[\text{LLN}]{n\to\infty} \mathbb{E}\big[x[n]x[n-\tau]\big] = 1\times\frac{1}{2} + (-1)\times\frac{1}{2} = 0$$



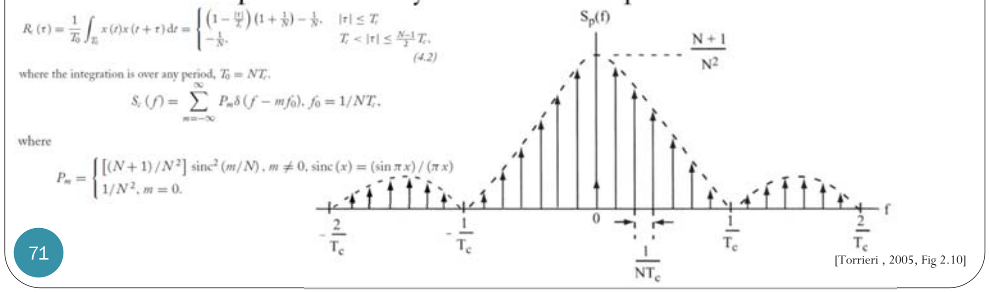
# Autocorrelation and PSD

- (Normalized) autocorrelations of maximal sequence and random binary sequence.




[Torrieri , 2005, Fig 2.9]

- Power spectral density of maximal sequence.



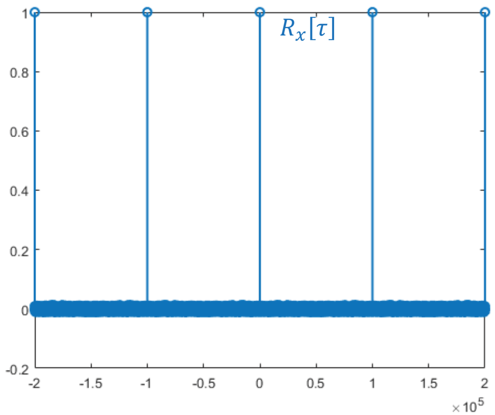

[Torrieri , 2005, Fig 2.10]



# Autocorrelation Function for Periodic Binary Random Sequence

Consider a periodic sequence whose one period is given by

```
1-2*randi([0 1],1,100000)
```



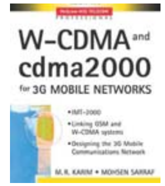

The shift property of binary random sequence implies that

$$R_x[\tau] = \langle x[n]x[n-\tau] \rangle \xrightarrow[\text{LLN}]{n\to\infty} \mathbb{E}\big[x[n]x[n-\tau]\big] = 1\times\frac{1}{2} + (-1)\times\frac{1}{2} = 0$$

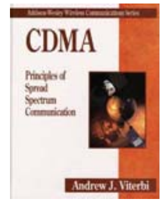


# References: m-sequences

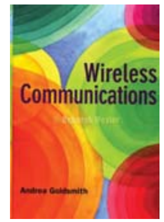

- Karim and Sarraf, ***W-CDMA and cdma2000 for 3G Mobile Networks***, 2002.
  - Page 84-90
- Viterbi, ***CDMA: Principles of Spread Spectrum Communication***, 1995
  - Chapter 1 and 2
- Goldsmith, ***Wireless Communications***, 2005
  - Chapter 13
- Tse and Viswanath, ***Fundamentals of Wireless Communication***, 2005
  - Section 3.4.3


[TK5103.452 K37 2002]




[TK5103.45 V57 1995]



# Review: m-sequence

DSSS: $m(t) \times c(t)$

Spectral spreading waveform

Spreading code/sequence   1  1  -1  1  -1  -1  -1  1  1  -1  1  -1

$c[n]$   0  0  1  0  1  1  1  0  0  1  0  1

Imitate properties of Bernoulli trials

Pseudo-random

One important collection of these is the collection of **m-sequences**.

Generated with LFSR whose connections corresponds to coefficients of primitive polynomials. The resulting sequence achieves the maximum period (length) of $N = 2^r - 1$ where $r$ is the degree of primitive polynomial.

# ECS455: Chapter 4

## Multiple Access

**4.4 m-sequence**

**(Additional Remarks)**

Dr.Prapun Suksompong
prapun.com/ecs455

**Office Hours:**
BKD, 6th floor of Sirindhralai building

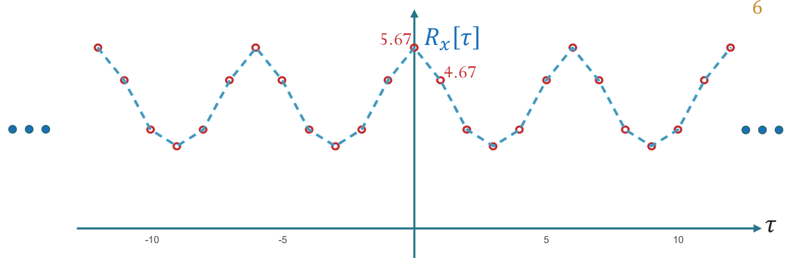
| | |
|---|---|
| **Tuesday** | **14:20-15:20** |
| **Wednesday** | **14:20-15:20** |
| **Friday** | **9:15-10:15** |



## Example: (Time) Autocorrelation Function for Periodic Sequence

$x[n]$ 0 2 4 3 2 1 0 2 4 3 2 1 **0 2 4 3 2 1** 0 2 4 3 2 1
$x[n-0]$ 0 2 4 3 2 1 0 2 4 3 2 1 **0 2 4 3 2 1** 0 2 4 3 2 1
× ↓ **0 4 16 9 4 1** → $\Sigma$ → 34 → $\times\frac{1}{6}$ → 5.67

$x[n]$ 0 2 4 3 2 1 0 2 4 3 2 1 **0 2 4 3 2 1** 0 2 4 3 2 1
$x[n-1]$ 1 0 2 4 3 2 1 0 2 4 3 2 **1 0 2 4 3 2** 1 0 2 4 3 2
× ↓ **0 0 8 12 6 2** → $\Sigma$ → 28 → $\times\frac{1}{6}$ → 4.67

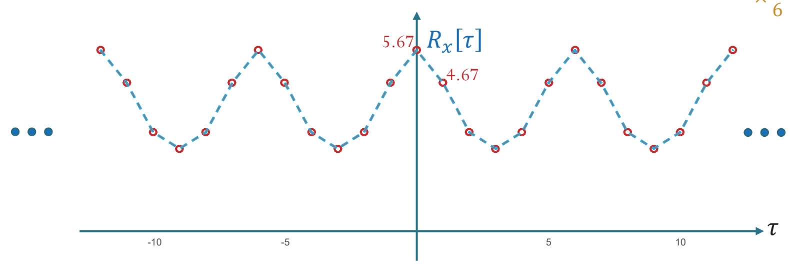





## Example: (Time) Autocorrelation Function for Periodic Sequence

$x[n]$ 0 2 4 3 2 1 0 2 4 3 2 1 **0 2 4 3 2 1** 0 2 4 3 2 1
$x[n-0]$ **0 2 4 3 2 1**
× ↓ **0 4 16 9 4 1** → $\Sigma$ → 34 → $\times\frac{1}{6}$ → 5.67

$x[n]$ 0 2 4 3 2 1 0 2 4 3 2 1 0 **2 4 3 2 1 0** 2 4 3 2 1
$x[n-1]$ **0 2 4 3 2 1**
× ↓ **0 8 12 6 2 0** → $\Sigma$ → 28 → $\times\frac{1}{6}$ → 4.67





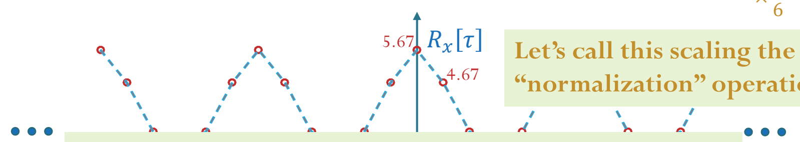

## Example: (Time) Autocorrelation Function for Periodic Sequence

$x[n]$ 0 2 4 3 2 1 0 2 4 3 2 1 **0 2 4 3 2 1** 0 2 4 3 2 1
$x[n-0]$ **0 2 4 3 2 1**
× ↓ **0 4 16 9 4 1** → $\Sigma$ → 34 → $\times\frac{1}{6}$ → 5.67

1 0 **2 4 3 2 1 0** 2 4 3 2 1
**0 2 4 3 2 1**
× ↓ **0 8 12 6 2 0** → $\Sigma$ → 28 → $\times\frac{1}{6}$ → 4.67

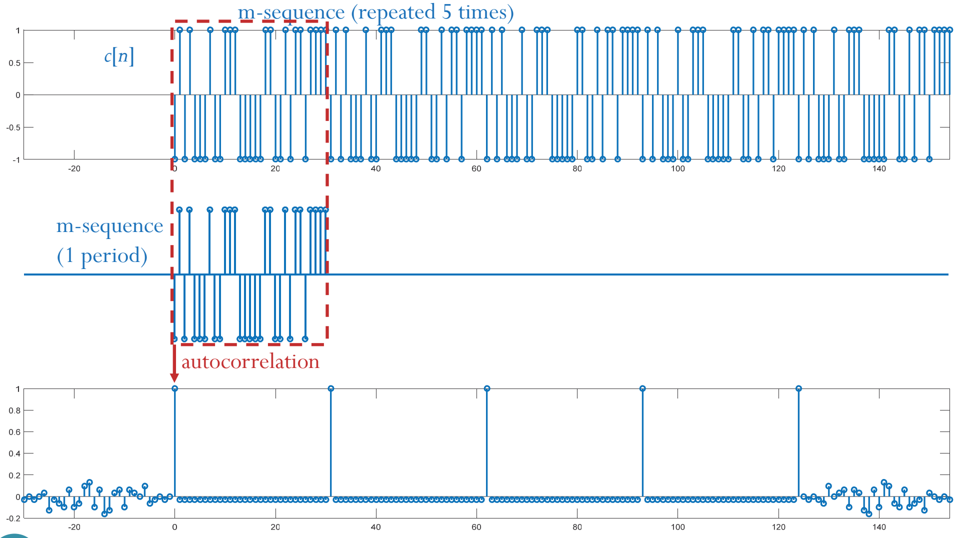
Let's call this the "sumproduct" operation. (This exact name is used in Excel for this kind of operation.) Mathematically, this is simply the dot product between two real-valued vectors.

Let's call this scaling the "normalization" operation.



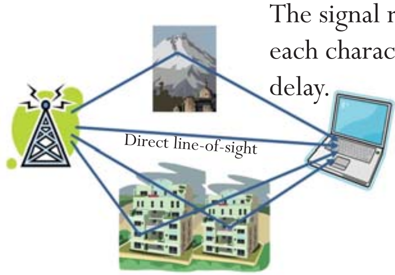

So, the combined computation can be called "normalized sumproduct" operation. We may also refer to this as the (sliding computation of) "autocorrelation" operation as well.

# Sliding computation of autocorrelation






# Wireless Comm. and Multipath Fading




The signal received consists of a number of reflected rays, each characterized by a different amount of attenuation and delay.

Here, let's consider the discrete-time version of fading:

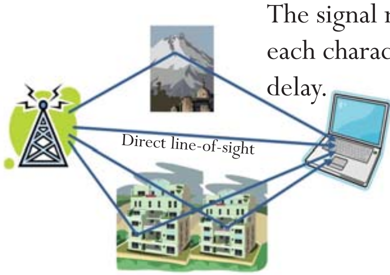
$$y[n]=\sum_{i=0}^{v}\beta_i c[n-\tau_i]$$

In particular, let's try

$$y[n]=5c[n]-3c[n-4]+c[n-10]$$



# Wireless Comm. and Multipath Fading



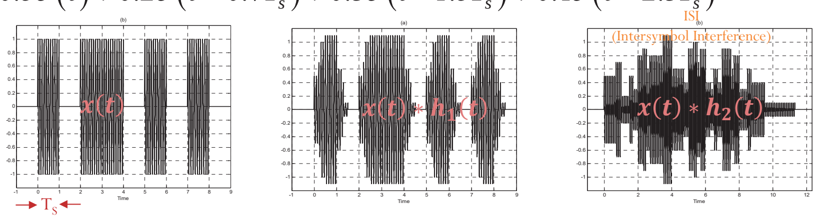

The signal received consists of a number of reflected rays, each characterized by a different amount of attenuation and delay.

$$y(t)=x(t)*h(t)+n(t)=\sum_{i=0}^{v}\beta_i x(t-\tau_i)+n(t)$$

$$h(t)=\sum_{i=0}^{v}\beta_i\delta(t-\tau_i)$$

$$h_1(t)=0.5\delta(t)+0.2\delta(t-0.2T_s)+0.3\delta(t-0.3T_s)+0.1\delta(t-0.5T_s)$$

$$h_2(t)=0.5\delta(t)+0.2\delta(t-0.7T_s)+0.3\delta(t-1.5T_s)+0.1\delta(t-2.3T_s)$$






# Identifying Parameters of Multipath Fading via Autocorrelation

$y[n]=5c[n]-3c[n-4]+c[n-10]$