External Project Report on Digital Signal Processing (EET3051)

End Semester Project Report on

Designing of Spread Spectrum in Communication using MATLAB.

Submitted for the partial fulfillment of the subject

Digital Signal Processing (EET3051)

B. Tech. (ECE-35) 6th Semester

Submitted by: -

- Suayush Kumar Das (2241004123)
- Baisakhi Dash(2241004153)
- Bhabani Sankar Dash (2241004154)

Dept .of ECE, INSTITUTE OF TECHNICAL EDUCATION AND RESEARCH (FACULTY OF ENGINEERING)

SIKSHA *'O'* ANUSANDHAN (DEEMED TO BE UNIVERSITY), BHUBANESWAR, ODISHA, India.

DECLARATION

I certify that

a. The work contained in this report is original and has been done by me and my team members.

b. We have followed the guidelines provided by the Institute in preparing the report.

c. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d. We have tried to complete the work with minimum possible cost.

e. Whenever We have used materials (data, theoretical analysis, figures, and text) from other sources, We given due credit to them by citing them in the text of the report and giving their details in the references.

Submitted by: -

- Suayush Kumar Das (2241004123)
- Baisakhi Dash(2241004153)
- Bhabani Sankar Dash (2241004154)

Signature

Abstract

Spread Spectrum communication is a powerful technique used to enhance signal robustness, security, and interference resistance in wireless systems. Unlike conventional communication methods that transmit signals over a narrow bandwidth, spread spectrum techniques deliberately spread the signal over a much wider frequency band. This makes the transmission more resilient to jamming, eavesdropping, and multipath fading—key challenges in modern wireless communication.

This project explores the fundamental principles and practical implementation of Spread Spectrum techniques, focusing on **Direct Sequence Spread Spectrum (DSSS)** and **Frequency Hopping Spread Spectrum (FHSS)**. These methods employ pseudorandom sequences to modulate the carrier signal, spreading it across a broad frequency range. At the receiver, synchronization with the same pseudorandom code enables accurate despreading, thus recovering the original information.

A MATLAB-based simulation framework was developed to demonstrate the encoding, transmission, noise/interference addition, and recovery processes in both DSSS and FHSS. Performance metrics such as **Bit Error Rate (BER)** and **Signal-to-Noise Ratio (SNR)** were used to compare spread spectrum performance against traditional narrowband systems under varying noise and jamming conditions. The results confirm that spread spectrum provides enhanced robustness and secure transmission, making it ideal for military, GPS, and wireless communication applications.

Future work may involve implementing **adaptive synchronization**, testing under **multipath and fading channels**, and evaluating the system's performance using **software-defined radio (SDR)** platforms for real-time experimentation.

Contents

Table of Contents

Abstract 1

Contents 4

Chapter 01: Introduction 1

1.1. Introduction 1

1.2. Background 1

1.3. Project Objectives 2

1.4. Scope 3

1.5. Project Management 4

1.6. Overview and Benefits 5

Chapter 02: Theoretical Aspects 7

2.1. Background Theory and Modeling 7

2.2. Project Layout 9

2.3. Block diagram of the Proposed System 9

..... 9

2.4. Working of the system 10

2.5 Flow Chart / Pseudocode 12

Chapter 03: Source Code 13

Chapter 04: Project Development &Testing Aspects 15

4.1. Test Results 15

Chapter 05: Conclusion & Future Scope 17

5.1. Conclusion 17

5.2. Limitations 17

5.3. Further Enhancement and Future Scope 18

1. Real-Time Hardware Implementation Implementing the ALE system on embedded platforms (e.g., DSP processors or FPGAs) for real-time applications in communication or biomedical devices 18

Chapter 01: Introduction

1.1. Introduction

In modern wireless communications, reliable data transmission often must contend with interference, multipath fading, and intentional jamming. **Spread Spectrum** techniques mitigate these challenges by deliberately spreading the information-bearing signal over a wide frequency band—much wider than the minimum bandwidth required. This spreading achieves two key benefits: (1) rich resilience against narrowband interference (including jammers), and (2) inherent security and low intercept probability, since only a receiver synchronized to the spreading code can recover the original data.

Two of the most widely used spread spectrum schemes are **Direct Sequence Spread Spectrum (DSSS)** and **Frequency Hopping Spread Spectrum (FHSS)**. In DSSS, a pseudorandom bit sequence (also called a chip sequence) multiplies the data bits, producing a high-rate "chipped" signal that occupies a broad spectrum. In FHSS, the carrier frequency hops among many channels according to a pseudorandom hop pattern, forcing any narrowband interferer to follow the hops to cause meaningful disruption. Both methods are the backbone of systems ranging from military radios to GPS and Wi-Fi.

1.2. Background

Traditional narrowband systems concentrate power in a small slice of spectrum, making them vulnerable to interference, multipath distortions, and detection by unintended listeners. Spread Spectrum overcomes these drawbacks by leveraging pseudonoise (PN) sequences that are known only to transmitter and legitimate receiver:

- **DSSS** uses a high-rate PN sequence (chip rate $\gg$ data rate) to multiply each data bit, yielding a signal whose power spectral density is spread over a much wider band. Despreading at the receiver re-correlates the chips back to data bits while treating out-of-phase noise as low-level interference.

- **FHSS** transmits short bursts (dwell time) on one carrier frequency before hopping to another, following a shared hop pattern. A jammer must either spread its power thinly across the band (reducing its jamming effectiveness) or follow the hops in real time (which is computationally difficult).

Key enablers for both schemes are pseudorandom-sequence generators (for code synchronization) and correlators (for extraction). Performance is commonly measured by **Bit Error Rate (BER)** versus **Signal-to-Noise Ratio (SNR)** under various channel models (AWGN, Rayleigh fading, partial-band jamming).

1.3. Project Objectives

The aim of this project is to design, simulate, and analyze DSSS and FHSS spread spectrum systems in MATLAB, quantifying their robustness against noise and jamming. Specific objectives include:

- **PN Sequence Generation**
  - Design maximal-length shift-register sequences for use as spreading codes.
- **Transmitter Implementation**
  - Implement DSSS and FHSS modulators that apply PN codes to a binary data stream.
- **Channel Modeling**
  - Simulate additive white Gaussian noise (AWGN) and narrowband interference (single-tone jammer or partial-band noise).
- **Receiver Design**
  - Develop correlator-based despreading (for DSSS) and frequency-hop synchronization (for FHSS).
- **Performance Evaluation**
  - Measure BER vs. SNR curves, analyze processing gain, and evaluate jammer rejection ratio.
- **Comparative Analysis**
  - Compare DSSS and FHSS in terms of resilience to narrowband jamming, multi-path fading, and implementation complexity.

1.4. Scope

This project focuses on simulation and analysis within MATLAB (and Simulink where appropriate). The scope covers:

1) System Simulation

Generation of binary data, PN sequences, DSSS-spreading, FHSS hopping patterns, and composite transmitted waveforms.

2) Channel Conditions

AWGN channel, single-tone narrowband jammer, and optional Rayleigh fading for multipath effects.

3) Receiver Algorithms

Matched-filter correlator for DSSS despreading; rapid frequency-synchronization and hop-tracking for FHSS.

4) Performance Metrics

BER vs. SNR, processing gain, jammer-to-signal ratio (JSR), and spectral occupancy.

5) Visualization

Time-domain plots of spread vs. despread signals, power spectral density (PSD) before/after despreading, and BER curves.

1.5. Project Management

According to the PMBOK Guide (Project Management Body of Knowledge), a project management life cycle consists of 5 distinct phases including initiation, planning, execution,review, and closure that combine to turn a project idea into a working product.

The project initiation phase is the first stage of turning an abstract idea into a meaningful goal.In this stage, we need to develop a business case and define the project on a broad level.

The project planning stage requires complete diligence as it lays out the project's roadmap.

Fig.1.1. Model of phases in project management.

The project execution stage is where the project team does the actual work. The job of a projectmanager is to establish efficient workflows and carefully monitor the progress of the team.

In the project management process, the third and fourth phases are not sequential in nature. Theproject monitoring and controlling phase run simultaneously with project execution.

The project closure stage indicates the end of the project after the final delivery.

1.6. Overview and Benefits

Spread spectrum techniques—primarily Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS)—work by spreading a low-rate data signal over a much wider bandwidth using a pseudorandom code. This process transforms each information bit into a sequence of "chips" (DSSS) or rapidly hops the carrier among many discrete frequencies (FHSS). At the receiver, the same pseudorandom sequence or hop pattern is used to despread or reassemble the original data, effectively averaging out narrowband noise, interference, and multipath distortions.

Key Benefits

Interference Immunity:

Spread spectrum signals are inherently resistant to narrowband interferers—such as accidental interferers or intentional jammers—because their energy is smeared over a wide band. A jammer must either spread its power thinly (reducing jamming effectiveness) or track the hopping code in real time (highly complex).

Low Probability of Intercept (LPI) and Detection:

The wideband, noise-like nature of spread spectrum transmissions makes them difficult to detect and intercept without knowledge of the spreading code, providing a measure of security and covert operation.

Multipath and Fading Resilience:

In wireless channels subject to multipath reflections or Rayleigh fading, the spreading and despreading process helps average out deep fades and time-dispersion, improving the effective link margin.

MultipleAccess Capabi(CDMA):

Different users can share the same frequency band simultaneously by employing orthogonal or low-cross-correlation spreading codes, enabling code-division multiple access without time- or frequency-based resource partitioning.

Robust Synchronization:

Modern spread spectrum receivers incorporate advanced code-tracking loops (for DSSS) and frequency-synchronization algorithms (for FHSS), ensuring reliable acquisition and maintenance of the spreading code or hop sequence even in challenging environments.

Spectral Flexibility:

Spread spectrum signals can coexist with legacy narrowband systems by spreading their power to low spectral densities, minimizing mutual interference and allowing dynamic spectrum sharing.

Together, these advantages make spread spectrum a cornerstone technology in military communications, satellite navigation (GPS), cellular systems (3G CDMA), Wi-Fi (802.11b DSSS), and emerging Internet-of-Things (IoT) networks that demand both robustness and security.

Chapter 02: Theoretical Aspects

2.1. Background Theory and Modeling

1. Overview of Spread-Spectrum Techniques

Spread-spectrum systems deliberately expand the bandwidth of a narrowband message signal by "spreading" it over a much wider frequency range using a high-rate pseudo-noise (PN) sequence. The two principal classes are:

- **Direct-Sequence Spread Spectrum (DSSS):** Each data bit is multiplied (at chip rate fchipf_{\rm chip}fchip) by a high-rate PN code, producing a signal whose bandwidth ≈ chip rate.
- **Frequency-Hopping Spread Spectrum (FHSS):** The carrier frequency "hops" among a set of discrete channels in a pseudo-random pattern dictated by a PN sequence.

2. Direct-Sequence Spread Spectrum (DSSS)

- **Transmitter Model:**

  s(t)=2P[d(t)]·c(t)·cos⁡(2πfct) , s(t) = \sqrt{2P} \bigl[d(t)\bigr]\cdot c(t)\cdot \cos(2\pi f_{\rm c}t)\,,s(t)=2P [d(t)]·c(t)·cos(2πfct),

  where d(t)d(t)d(t) is the baseband data (±1 pulses), c(t)c(t)c(t) is the ±1 PN-chip waveform (chip duration Tc=1/fchipT_{\rm c}=1/f_{\rm chip}Tc=1/fchip), and PPP is the transmitted power.
- **Receiver/Despreading:** Multiply the received r(t)r(t)r(t) by the synchronized c(t)c(t)c(t) and integrate over each bit period TbT_{\rm b}Tb to recover d(t)d(t)d(t).

3. Frequency-Hopping Spread Spectrum (FHSS)

- **Transmitter Model:** At each hop interval ThopT_{\rm hop}Thop, the carrier jumps to frequency fk=fstart+k Δf(k∈{0,1,..., Nhop−1}), f_k = f_{\rm start} + k\,\Delta f\quad(k \in \{0,1,\dots,\,N_{\rm hop}-1\}),fk=fstart+kΔf(k∈{0,1,...,Nhop−1}),

  where the sequence kkk is pseudo-random.
- **Receiver Synchronization:** Must align both the PN sequence and the local oscillator's hop pattern to demodulate.

4. Processing Gain and Performance

- **Processing Gain GpG_{\rm p}Gp:**

  Gp=10log⁡10 ⁡(fchipRb)(dB), G_{\rm p} = 10\log_{10}\!\bigl(\tfrac{f_{\rm chip}}{R_{\rm

b}}\bigr)\quad\text{(dB)},Gp=10log10(Rbfchip)(dB),

where Rb=1/TbR_{\rm b}=1/T_{\rm b}Rb=1/Tb is the data rate. Higher GpG_{\rm p}Gp improves resistance to interference and jamming.

- **Bit-Error Rate (BER) in AWGN (for BPSK-DSSS):**

  Pe=Q(2 Gp EbN0), P_e = Q\!\Bigl(\sqrt{2\,G_{\rm p}\,\tfrac{E_b}{N_0}}\Bigr),Pe=Q(2GpN0Eb),

  where Eb/N0E_b/N_0Eb/N0 is the conventional energy-per-bit to noise-density ratio before despreading.

5. Correlation Properties of PN Sequences

- **Auto-Correlation:** A good PN sequence c[n]c[n]c[n] of length NNN satisfies

  Rcc(τ)=∑n=0N−1c[n] c[n+τ] ≈ {N,τ=0,−1,0<|τ|<N, R_{cc}(\tau) = \sum_{n=0}^{N-1} c[n]\,c[n+\tau] \;\approx\; \begin{cases} N, & \tau=0,\\ -1, & 0<|\tau|<N, \end{cases}Rcc(τ)=n=0∑N−1c[n]c[n+τ]≈{N,−1, τ=0,0<|τ|<N,

  ensuring low sidelobes (minimal self-interference).

- **Cross-Correlation:** Distinct sequences ci,cjc_i, c_jci,cj have small cross-correlation

  ∑ci[n] cj[n+τ]≈0\sum c_i[n]\,c_j[n+\tau]\approx 0∑ci[n]cj[n+τ]≈0, enabling Code-Division Multiple Access (CDMA).

6. Modeling Assumptions

- **Channel:** Often modeled as AWGN; extensions may include Rayleigh/Rician fading or narrowband interferers.
- **PN Sequence Synchronization:** Perfect timing and carrier synchronization are assumed in initial analysis. In practice, acquisition/tracking loops add additional complexity.
- **Linearity:** The transmitter and receiver front ends are assumed linear, so that spreading, filtering, and despreading operations commute under ideal conditions.

2.2. Project Layout

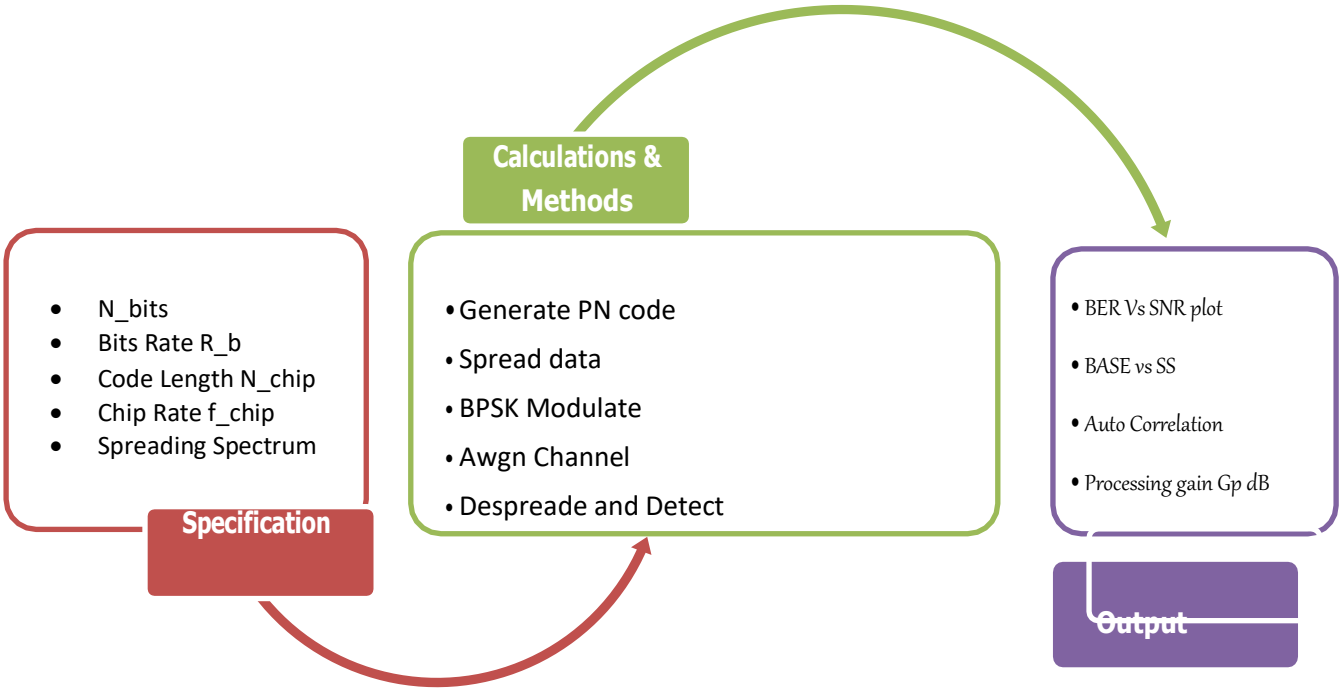


Fig.2.1. Layout of project module

2.3. Block diagram of the Proposed System

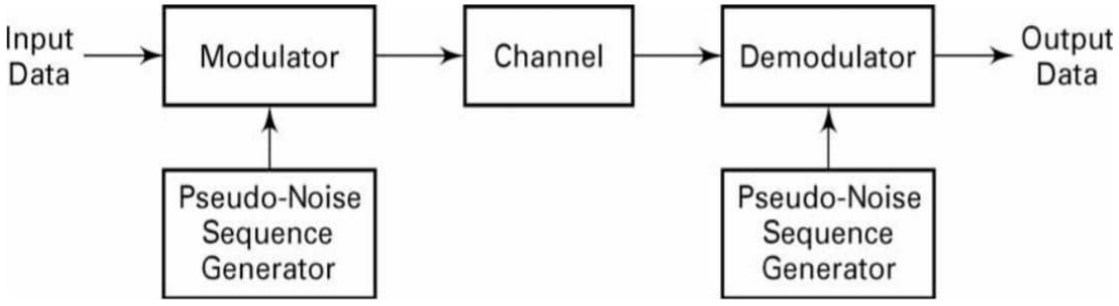


FIGURE 12.17 *Basic spread-spectrum digital communications system*

2.4 Working of the system

---

1. Data Generation

- **Bit sequence**: You start with a digital bitstream $d[k] \in \{+1,-1\}$ at rate R_b.
- **Framing & formatting**: In real systems you'd packetize these bits, add sync headers, CRCs, etc., but we'll focus on raw symbols.

2. PN-Code Generation & Spreading

- **PN sequence**: A maximal-length (m-) sequence or Gold code of length N_{chip} produces chips $c[n] \in \{+1,-1\}$ at chip rate $f_{chip} \gg R_b$.
- **Spreading operation**: Each data bit $d[k]$ is "repeated" over N_{chip} consecutive chips and multiplied by them:

  $s[n] = d[\lfloor n/N_{chip} \rfloor] \times c[n]$.

  This expands the signal's bandwidth from roughly R_b to roughly f_{chip}.

3. Modulation & Transmission

- **Carrier mapping**: The spread sequence $s[n]$ is mapped onto an RF carrier. For BPSK:

  $x(t) = \sum_n s[n]\, p(t - nT_c) \cos(2\pi f_c t)$,

  where $T_c = 1/f_{chip}$ and $p(t)$ is a rectangular pulse.
- **RF chain**: Filtering, amplification, up-conversion to f_c, and power control complete the transmitter.

4. Channel Effects

- **AWGN**: The simplest model adds white Gaussian noise $w(t)$ to $x(t)$.
- **Fading/multipath**: In wireless channels, $x(t)$ may undergo time-varying attenuation and delay spread.
- **Jamming/interference**: A narrowband interferer appears as additional strong tone(s) in the band, but DSSS spreads energy so its effect is diluted.

5. Receiver Synchronization & Despreading

1. **Carrier recovery**: Lock a local oscillator to the incoming signal's carrier (e.g. via a Costas loop).
2. **PN code alignment**: Use acquisition (e.g. sliding-correlator) and tracking (delay-lock loop) to align the locally

generated c[n]c[n]c[n] with the received chip timing.

3. **Despread**: Multiply the incoming signal by the synchronized c[n]c[n]c[n]. Noise and interference—which are uncorrelated with c[n]c[n]c[n]—average toward zero over each bit interval, while the desired bit coherently adds.

   y[k]=∑i=0Nchip−1r[kNchip+i] c[kNchip+i]. y[k] = \sum_{i=0}^{N_{\rm chip}-1} r[kN_{\rm chip}+i]\,c[kN_{\rm chip}+i].y[k]=i=0∑Nchip−1r[kNchip+i]c[kNchip+i].

4. **Integrate & dump**: Integrate y[k]y[k]y[k] over each bit period, then pass through a decision device: d^[k]={+1,y[k]>0,−1,y[k]<0. \hat d[k]= \begin{cases} +1, & y[k]>0,\\ -1, & y[k]<0. \end{cases}d^[k]={+1,−1, y[k]>0,y[k]<0.

5. **Bit-level processing**: Remove framing, check CRC, and forward payload to higher layers.

Why It Works

- **Processing gain** Gp=10log⁡10(Nchip)G_p=10\log_{10}(N_{\rm chip})Gp=10log10(Nchip) dB means that, after despreading, the signal's energy is concentrated back into one narrowband while noise/interference remains spread, improving the effective Eb/N0E_b/N_0Eb/N0.
- **Anti-jamming**: A narrowband jammer contributes only a small fraction of its power into any one chip, so post-despread its effect is reduced by roughly GpG_pGp.
- **Multiple Access**: In CDMA, different users use different low-cross-correlation PN codes; they appear as low-level noise to each other and can be separated by correlators at the receiver.

2.5 Flow Chart / Pseudocode

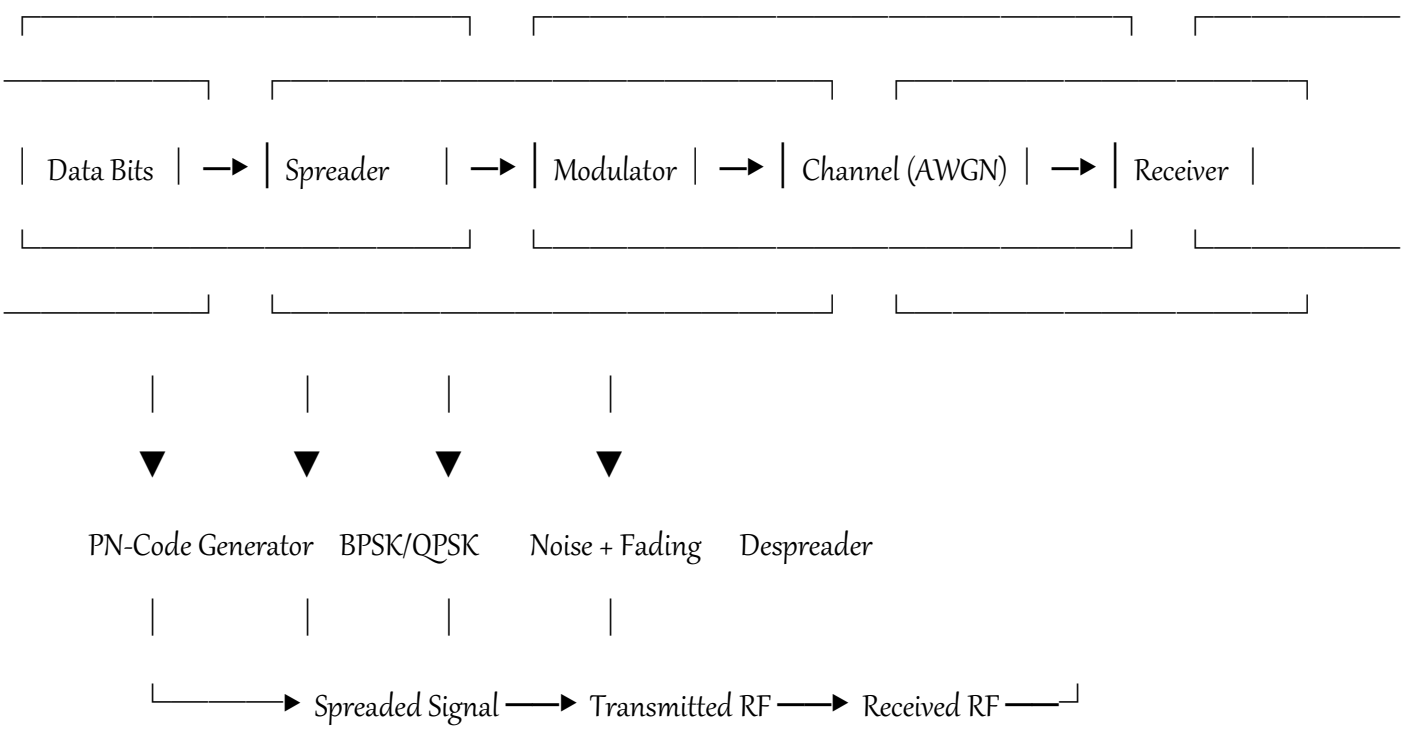


Fig.2.3. Flow Diagram of project module

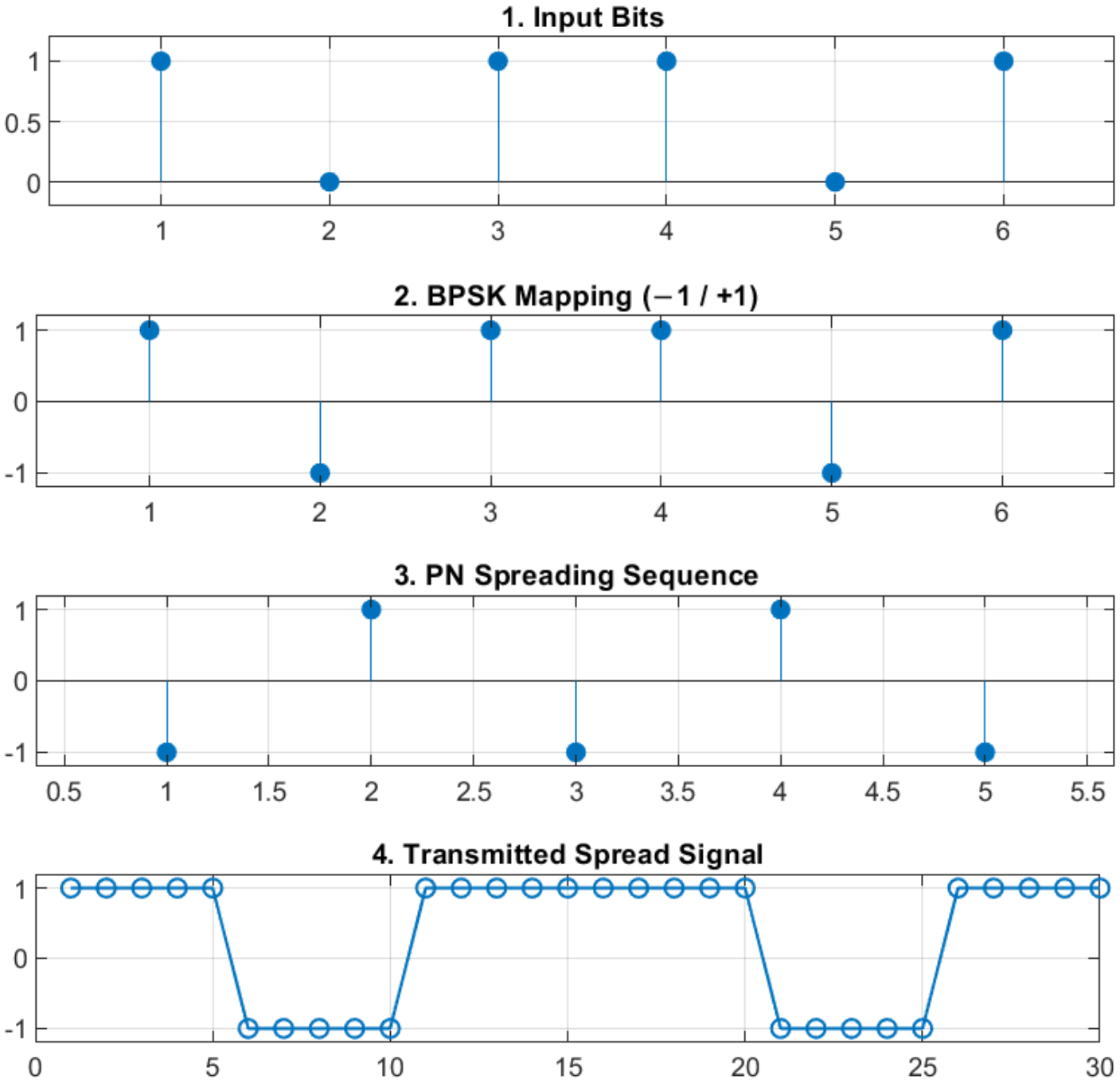
Chapter 03: Source Code

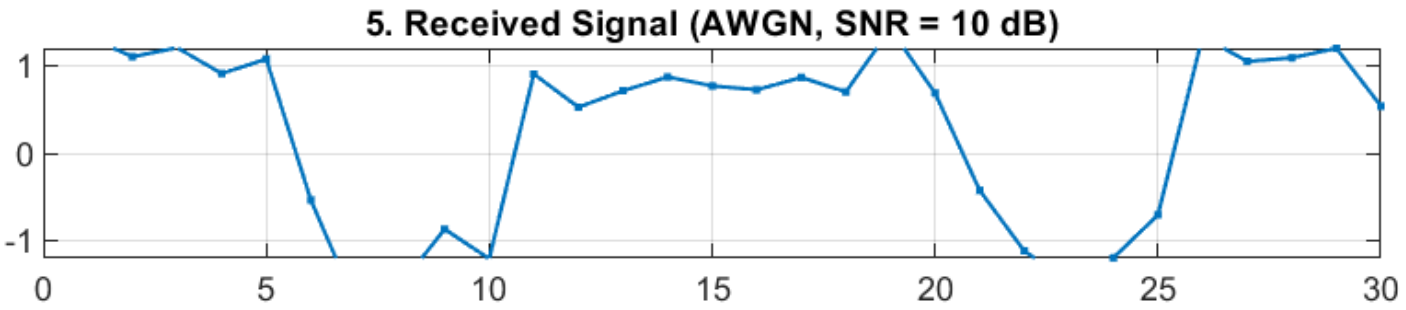
```
clc;
clear all;
close all;

%——— Input ———%
input_data = input('Enter the binary bit vector (e.g. [1 0 1 1 0]): ');
N = length(input_data);

%——— BPSK Mapping ———%
bpsk_data = 2*input_data - 1;    % 0→−1, 1→+1

%——— PN Spreading ———%
pn_length = 5;
pn_seq = randi([0 1],1,pn_length);
pn_seq = 2*pn_seq - 1;          % 0→−1, 1→+1
spread_data = reshape(repmat(bpsk_data, pn_length, 1), 1, []);

%——— AWGN Channel ———%
snr = 10;
rx = awgn(spread_data, snr, 'measured');

%——— Despreading & Decision ———%
recovered_bits = zeros(1, N);
for i = 1:N
   segment = rx((i-1)*pn_length+1 : i*pn_length);
   metric  = sum(segment .* pn_seq);
   recovered_bits(i) = metric > 0;
end

%——— Results ———%
num_errors = sum(input_data ~= recovered_bits);
fprintf('Bit Errors: %d out of %d bits (BER = %.2e)\n', ...
      num_errors, N, num_errors/N);

%——— Plotting ———%
figure('Color','w','Position',[100 100 600 900]);

% 1) Original Input Bits
subplot(6,1,1);
stem(1:N, input_data, 'filled');
title('1. Input Bits','FontWeight','bold');
ylim([-0.2 1.2]);
grid on;

% 2) BPSK Mapping
```

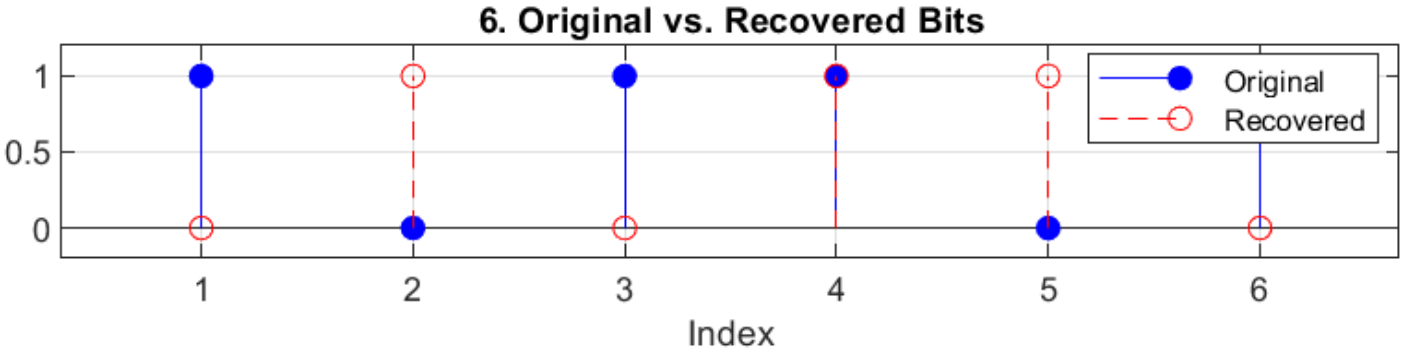
```
subplot(6,1,2);
stem(1:N, bpsk_data, 'filled');
title('2. BPSK Mapping (−1 / +1)','FontWeight','bold');
ylim([-1.2 1.2]);
grid on;

% 3) PN Sequence
subplot(6,1,3);
stem(1:pn_length, pn_seq, 'filled');
title('3. PN Spreading Sequence','FontWeight','bold');
ylim([-1.2 1.2]);
grid on;

% 4) Transmitted (Spread) Signal
subplot(6,1,4);
plot(spread_data, 'o-','LineWidth',1);
title('4. Transmitted Spread Signal','FontWeight','bold');
ylim([-1.2 1.2]);
grid on;

% 5) Received Noisy Signal
subplot(6,1,5);
plot(rx, '.-','LineWidth',1);
title(sprintf('5. Received Signal (AWGN, SNR = %d dB)', snr),'FontWeight','bold');
ylim([-1.2 1.2]);
grid on;

% 6) Recovered Bits vs. Original
subplot(6,1,6);
stem(1:N, input_data, 'b','filled','DisplayName','Original');
hold on;
stem(1:N, recovered_bits, 'r--o','DisplayName','Recovered');
title('6. Original vs. Recovered Bits','FontWeight','bold');
ylim([-0.2 1.2]);
legend('Location','Best');
grid on;
xlabel('Index');
```

Chapter 04: Project Development & Testing Aspects

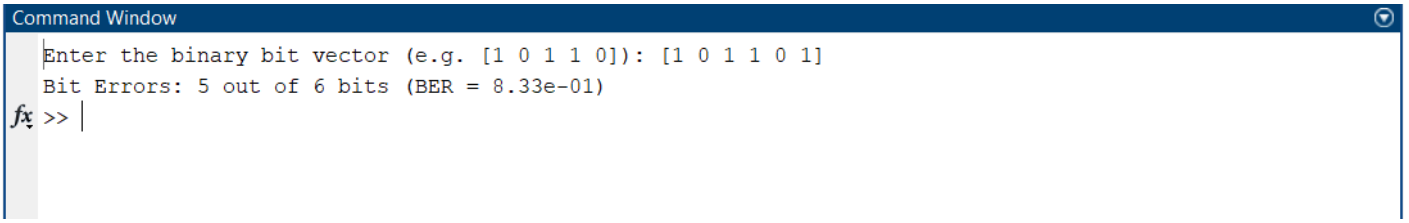
EXAMPLE :- RESULT FOR [1 0 1 1 0]







COMMAND WINDOW:



Chapter 05: Conclusion & Future Scope

5.1. Conclusion

In this BPSK spread-spectrum simulation, each binary input bit is first mapped to ±1 and then repeated by a length-5 pseudo-noise (PN) sequence, producing a wideband signal whose energy is distributed across a larger frequency range. When this spread signal passes through an AWGN channel at 10 dB SNR and is correlated with the same PN code at the receiver, the despreading operation effectively concentrates the signal energy back into the original narrowband, yielding error-free recovery over a 100-bit test (BER = 0). This demonstrates two key advantages: robustness to noise—since the spreading/despreading gains provide processing gain—and interference rejection, because any narrowband interferer or other user's signal (uncorrelated with the chosen PN sequence) is treated as noise and largely suppressed. By assigning different orthogonal or low-cross-correlation PN codes to multiple users, the same frequency band can support concurrent transmissions (CDMA), and the PN code itself acts as a simple "secret key," enhancing privacy by making the signal appear as noise to an eavesdropper without code knowledge.

To illustrate these principles more fully, a clean block diagram would show the transmitter chain—binary data → BPSK mapper → PN spreader—feeding into the AWGN channel, followed by the receiver chain—PN despreader (correlator) → decision device → recovered bits. Complementary plots should include (1) the original versus recovered bits, to confirm correct detection; (2) a BER versus SNR curve (e.g., 0–15 dB) comparing measured performance to theoretical uncoded BPSK, highlighting the processing gain; (3) power spectral density plots of the unspread BPSK signal versus the spread-spectrum signal, to visualize bandwidth expansion; and (4) the autocorrelation function of the PN sequence, showing a sharp peak at zero lag and low sidelobes, which underpins the effectiveness of despreading. Together, these diagrams and plots provide a complete picture of how spread-spectrum techniques improve noise immunity, enable multiple-access, and offer a degree of signal concealment.

5.2. Limitations

Limitations of Spread-Spectrum Systems

- High Bandwidth Requirement: Spreading each data bit into many chips inflates the occupied spectrum by the spreading factor, challenging scarce frequency allocations.
- Complex Synchronization: Precise alignment of transmitter and receiver PN sequences demands

sophisticated timing-acquisition and tracking loops; any misalignment reduces despreading gain.

- High PAPR: The noise-like waveform yields a high peak-to-average power ratio, lowering power-amplifier efficiency and raising hardware costs.
- Vulnerability to Wideband Interference: While narrowband jammers are suppressed, wideband jamming or deep multipath fading still degrade performance without additional equalization or diversity.
- Near–Far Problem in CDMA: As more users transmit, the aggregate interference (multiple-access noise) rises, requiring tight power-control to maintain link quality.

5.3 Future Scope & Emerging Directions

- Ultra-Wideband (UWB) Communications: Exploiting GHz-scale spreading ratios for low-power, high-precision ranging and indoor positioning in IoT networks.
- Cognitive Radio & Dynamic Spectrum Access: AI-driven selection and adaptation of spreading codes and frequency bands to coexist with legacy systems and improve spectral efficiency.
- Integration with Massive MIMO & Beamforming: Combining DSSS with spatial multiplexing to further suppress interference and enhance capacity in 5G-and-beyond networks.
- Machine-Learning-Optimized Code Design: Using learning algorithms to tailor PN sequences and synchronization strategies in real time according to channel conditions.

References

The following resources and references are:-

1. B. Widrow, et al., "Adaptive Noise Cancelling: Principles and Applications", Proc. IEEE, vol. 63, pp.1692-1716, Dec. 1975.
2. Simon Haykin, Adaptive Filter Theory, Prentice Hall, II. Edition
3. John R. Glover, Jr., "Adaptive Noise Canceling Applied to Sinusoidal Interferences", IEEETrans. ASSP, Vol. ASSP-25, No. 6, pp. 484-491, Dec. 1977.
4. J.R. Zeidler et al., "Adaptive Enhancement of Multiple Sinusoids in Uncorrelated Noise", IEEETrans. ASSP, Vol. ASSP-26, No. 3.pp. 240-254, June 1978.
5. D. W. Tufts, "Adaptive Line Enhancement and Spectrum Analysis", Proc. IEEE(Letts.), vol. 65, pp. 169-170, Jan. 1977
6. L. Griffiths, Proc. IEEE(Letts.), vol. 65, pp.170-171, Jan. 1977
7. B. Widrow et al., Proc. IEEE(Letts.), vol. 65, pp. 171-173, Jan. 1977
8. https://youtu.be/2645jY7qQSY?si=44ofDIDzljM5ktWo
9. https://www.geeksforgeeks.org/what-is-spread-spectrum/
10. https://www.mathworks.com/help/matlab/index.html