SIES Graduate school of Technology

Department of Electronics and Telecommunication

Institute Vision

"To be a center of excellence in Education and Technology committed towards Socio Economic advancement of the country".

Institute Mission

1. To impart advanced knowledge in Engineering and Technology.
2. To transform young minds towards professional competence by inculcating values and developing skills.
3. To promote research and ensure continuous value addition among students and employees.
4. To strengthen association with industry, research organizations and alumni to enhance knowledge on current technologies.
5. To promote next generation technocracy and nurture entrepreneurial culture for social-economic growth.

Department Vision

To be a Premier Department in Electronics & Telecommunications Engineering.

Department Mission

1. To provide quality education satisfying the requirements of corporate world across diverse fields.
2. To develop life-long learning skills to cater to socio-economic needs.
3. To strengthen Industry-Institute Interaction to bridge the gap between academic and industrial requirements.
4. To equip students with leadership and entrepreneurial skills.

Program Educational Objectives: Graduates will be able to

1. Identify, formulate and solve engineering problems in the industry, complying with ethical standards and societal needs.

2. Pursue higher studies and professional development courses leading to significant advancement in the field of specialization.

3. Apply technical concepts to develop applications and design products.

4. Exhibit leadership and entrepreneurial acumen in career.

Program Specific Outcomes: Students will be able to

1. Achieve eminence in domains like signal processing, VLSI, embedded IoT, RF & microwave.
2. Become technocrats capable of working in multi-disciplinary fields.

Program Outcomes:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities

with an understanding of the limitations.

6. **The Engineer and The World**: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment.

7. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

8. **Individual and teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

9. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

10. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

11. **Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcome:

1.To develop a platform for achieving eminence in domains like signal processing, VLSI, embedded IOT, RF & microwave.

2. To build technocrats capable of identifying, analyzing & designing solutions to real world problems independently or as a team

Digital Communication Lab (ECL 501)

Course Outcomes: Students will be able to

CO1. Compare various source coding schemes.
CO2. Design and implement different error detection codes.
CO3. Design and implement different error correction codes.
CO4. Compare various line coding techniques.
CO5. Illustrate the impulse response of a matched filter for optimum detection.
CO6. Demonstrate various digital modulation techniques.

Sr No	**Name of the Experiment**	**LOs**
1	Calculation of self-information, source entropy and average information rate	LO1
2	Calculation of Gaussian channel capacity, bandwidth, and signal to noise	LO1
3	Calculation of Shannon's limit	LO1
4	Huffman code generation and efficiency calculation	LO2
5	Shannon-Fano code generation and efficiency calculation	LO2
6	PCM Encoding and Decoding	LO3
7	DM Encoding and Decoding	LO3
8	Digital Multiplexing and De-multiplexing	LO3
9	Line codes Generation and Comparison	LO4
10	Impulse response and output of matched filter	LO4
11	Generation and Analysis of Amplitude Shift Keying	LO5
12	Generation and Analysis of Phase Shift Keying	LO5
13	Generation and Analysis of Frequency Shift Keying	LO5
14	PN Sequence Generation and Analysis	LO5
15	Spread Spectrum Signal Generation and Detection	LO5
16	Group Activity	

DC Experiment 1

Name:
Roll No:
Date:

Calculation of self-information, source entropy and average information rate -Python program

Aim: Write a program for the calculation of self-information, source entropy and average information rate using Python.

Language: Python

Theory: Self Information of a symbol $x_i = I(x_i) = \log_2 (1/p(x_i))$ bits,
where $p(x_i)$ is the probabilities of the symbols x_i
Source entropy $H(x) = \Sigma\, p(x_i) \log_2 (1/p(x_i))$ bits per symbol
Average Information Rate R = Source entropy x Average symbol rate
$= H(x)\, r_s$ bps

Program Code:

Simulation Results:

Conclusion:

DC Experiment 2

Name:
Roll No:
Date:

Calculation of Channel capacity, Bandwidth and Signal to noise Ratio for a AWGN channel- Python Program

Aim: Write a program for the calculation of Channel capacity, Bandwidth and Signal to Noise Ratio for an AWGN channel using Python.

Language: Python

Theory: According to Shannon-Hartley theorem,

Channel capacity of a Gaussian channel $C = B \log_2(1+SNR)$ bps

Bandwidth $B = C / \log_2(1+SNR)$ Hz

Signal to Noise Ratio $SNR = 2^{C/B} - 1$

SNR in dB $= 10 \log_{10} (SNR)$

Program Code:

Simulation Results:

Conclusion:

DC Experiment 3

Name:
Roll No:
Date:

Calculation of Shannon's Limit- Python Program

Aim: Write a program for the calculation of Shannon's limit for an AWGN channel using Python.

Language: Python

Theory: According to Shannon-Hartley theorem,

Channel capacity of a Gaussian channel C = B $\log_2(1+SNR)$ bps

When the bandwidth is infinite, the capacity of the Gaussian channel is called Shannon's limit.
Shannon's limit $C_\infty = 1.44\ S/\eta$
Where η is the power spectral density of White Gaussian Noise

Program Code:

Simulation Results:

Conclusion:

Name:
Roll No:
Date:

Generation and efficiency calculation of Huffman code- Python Program

Aim: Write a program for the generation of Huffman code and the calculation of code efficiency using Python.

Language: Python

Theory: Huffman code generation algorithm:
step 1: arrange the symbols in descending order of probability.
step 2: combine the last 2 symbols and write the total probability.
step 3: assign 0 to upper symbol and 1 to lower symbol.
step 4: repeat the above 3 steps up to the total probability is equal to 1.

Source entropy H(x) = $\Sigma\ p(x_i) \log_2(1/p(x_i))$ bits per symbol
Where $p(x_i)$ is the probability of the symbol x_i

Average code length L = Σ $p(x_i)$ **N**(x_i)
Where **N**(x_i) is the code length of the symbol x_i

Code Efficiency = [H(x)/L] 100%

Redundancy = 100 – Code efficiency

Program Code:

Simulation Results:

Conclusion:

DC Experiment 5

Name:
Roll No:
Date:

Generation and efficiency calculation of Shannon-Fano code-Python Program

Aim: Write a program for the generation of Shannon-Fano code and the calculation of code efficiency using Python.

Language: Python

Theory: Shannon-Fano code generation algorithm:
step 1: arrange the symbols in descending order of probability.
step 2: divide the symbols into 2 groups with minimum probability difference.
step 3: assign 0 to each upper group symbol and 1 to each lower group symbol.
step 4: repeat the above 2 steps until further division is not possible.

Source entropy $H(x) = \Sigma\, p(x_i) \log_2 (1/p(x_i))$ bits per symbol
Where $p(x_i)$ is the probability of the symbol x_i

Average code length $L = \Sigma\, p(x_i)\ N(x_i)$
Where $N(x_i)$ is the code length of the symbol x_i

Code Efficiency = [H(x)/L] 100%

Redundancy = 100 – Code efficiency

Program Code:

Simulation Results:

Conclusion:

DC Experiment 6

Name:
Roll No:
Date:

Generation of VRC code, Error Detection and Decoding- Python Program

Aim: Write a program for the generation of Vertical Redundancy Check code, error detection and decoding using Python.

Language: Python

Theory: Code generation:

VRC bit = EX-OR of the data bits

Code = Data bits and VRC bit

Error Detection: EX-OR of all the received bits = 0, no error
= 1, error

Decoding: If there is no error, select only data bits from the received code. If error exists, reject the received code (no decoding)

Program Code:

Simulation Results:

Conclusion:

DC Experiment 7

Name:
Roll No:
Date:

Generation of HRC code, Error Detection and Decoding- Python Program

Aim: Write a program for the generation of Horizontal Redundancy Check code (Longitudinal Redundancy Check LRC code), error detection and decoding using Python.

Language: Python

Theory: Code generation:
Step1: decide the HRC size (n bits)
Step2: divide the data into n-bits characters.

Step3: HRC character = EX-OR of all the data characters

Step4: Code = Data characters and HRC character

Error Detection: EX-OR of all the received characters $= 0$, no error
$\neq 0$, error

Decoding: If there is no error, select only data bits from the received code. If error exists, reject the received code (no decoding)

Program Code:

Simulation Results:

Conclusion:

DC Experiment 8

Name:
Roll No:
Date:

Generation of (6,3) and (7,4) linear block codes- Python Program

Aim: Write a program for the generation of (6,3) and (7,4) systematic linear block codes, calculation of minimum Hamming distance, detection capability and correction capability using Python.

Language: Python

Theory: (6,3) Code generation:
$[D]_{1x3}\ [G]_{3x6} = [C]_{1x6}$
Total number of data blocks are 8 and total number of code words are 8

(7,4) code generation:

$[D]_{1x4}\ [G]_{4x7} = [C]_{1x7}$
Total number of data blocks are 16 and total number of code words are 16
Minimum Hamming distance d_{min} = minimum nonzero Hamming weight of the generated code
Hamming weight = number of 1's in a code

Detection capability = $d_{min} - 1$

Correction capability = $(d_{min} - 1)/2$, if d_{min} is an odd value
= $(d_{min} - 2)/2$, if d_{min} is an even value

Program Code:

Simulation Results:

Conclusion:

DC Experiment 9

Name:
Roll No:
Date:

Hamming code error detection, correction, and decoding- Python Program

Aim: Write a program for error detection, correction and decoding of (7,4) Hamming code using Python.

Language: Python

Theory: Step1: Find $[H]_{3x7}$ from the given $[G]_{4x7}$
If $[G] = [I\ P]$, $[H] = [P^T\ I]$
If $[G] = [P\ I]$, $[H] = [I\ P^T]$

Step2: Find $[H^T]$

Step3: Construct the decoding table (syndrome table)

Step4: Determine syndrome for the given received code R
$[S] = [R]\ [H^T]$

Step5: If S = 0, no error
$\neq 0$, error

Step6: If there is no error, select 4 data bits from R

Step7: If an error exists, select error pattern (E) according to the calculated syndrome.

Step8: Corrected code $C = R \oplus E$

Step9: Select 4 data bits from C

G=[1 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1 1 0 1]
R = [1 0 1 0 1 1 0]

Construct Syndrome table

The syndrome table consists of a list of error patterns (7, 4) Hamming code can correct and the corresponding syndromes. Against each error pattern, corresponding syndrome can be calculated using the formula $S = e.H^T$.

H = [1 0 1 1 1 0 0 1 1 1 0 0 1 0 0 1 1 1 0 0 1]

H^T = [1 1 0 0 1 1 1 1 1 1 0 1 1 0 0 0 1 0 0 0 1]

E	S
1000000	110
0100000	011
0010000	111
0001000	101
0000100	100
0000010	010
0000001	001

Calculate Syndrome from R

$S= R.H^T$ = [1010110] [1 1 0 0 1 1 1 1 1 1 0 1 1 0 0 0 1 0 0 0 1] = [111]

$S \neq 0$ indicates that error in R.

Selection of error pattern from syndrome table.

For S = [1 1 1] , E = [0 0 1 0 0 0 0]

Error correction

$C = R \oplus E$ = [1 0 1 0 1 1 0] $\oplus$ [0 0 1 0 0 0 0] = 1000110

Decoding

In corrected code C, the left most 4 bits are data bits. D = 1000

Program Code:

Simulation Results:

Conclusion:

DC Experiment 10

Generation of line codes- Python Program

Name:

Roll No:
Date:

Aim: Write a program for the generation of NRZ unipolar, RZ unipolar, NRZ polar, RZ polar, NRZ AMI and Manchester line codes using Python. Compare all the line codes in terms of bandwidth, average power, synchronization capability, DC level, polarity inversion error and transmission capability.

Language: Python

Theory: Line codes the pulse representation of digital data. It is used for the baseband transmission of digital data. Different parameters for selecting a specific line code for an application are bandwidth, average power, synchronization capability, DC level, polarity inversion error and transmission capability.

Program Code:

Simulation Results:

Comparison Table:

Conclusion:

DC Experiment 11

Name:
Roll No:
Date:

Generation of matched filter impulse response and filter output-Python Program

Aim: Write a program for the generation of matched filter impulse response and output for a given input using python program.

Language: Python

Theory: Let the input signal is p(t) and the signal duration is T_b.
Impulse response of matched filter $h(t) = k\, p(T_b - t)$
Where k is the gain of the filter.
Output of matched filter = p(t) convolution h(t)

Program Code:

Simulation Results:

Conclusion:

DC Experiment 12

Name:
Roll No:
Date:

Study of ASK Modulation and Demodulation

Aim: To study Amplitude Shift Keying modulation and demodulation.

EQUIPMENTS:
Patch cords, DCL-06(ASK MODULATION and DEMODULATION kit), CRO.

THEORY:
Digital communications systems represent the information by binary digits 1 and 0. There are various modulation techniques. This kit is for explaining the function of amplitude shift keying. In ASK, amplitude of the carrier varies in accordance with the input bits 1 and 0. The ASK modulator generates output against the input data 1 and no output against the input data 0. It can be generated using a balanced modulator. ASK demodulator employs full wave rectifier, low pass filter and decision device.

$V_{ASK}(t) = b(t) A\cos\omega_0 t$, where b(t) is 0 volt for input data 0 and 2 volt for input data 1. So, there is no transmission for input data 0 and $2A\cos\omega_0 t$ for input data 1.

BLOCK DIAGRAM of MODULATOR:

DEMODULATOR:

PROCEDURE:

1) Refer to the block diagram and carry out the following connections and switch settings.
2) Connect the power supply in proper polarity to the modulator and demodulator kit (DCL-06) and switch it on.
3) Connect SERIAL DATA generated on board to CONTROL IN of CARRIER MODULATOR.
4) Select ASK modulation using switch S1, the ASK LED will glow.
5) Observe the waveforms at SERIAL DATA, SINE1, and MOD OUT.
6) Connect ASK modulated signal MOD OUT to the ASK IN of the ASK DEMODULATOR.
7) Observe waveforms on CRO.

Waveforms on graph paper:

Conclusion:

Date:

Study of PSK Modulation and Demodulation

Aim: To study Phase Shift Keying modulation and demodulation.

EQUIPMENTS:

Patch cords, DCL-06(PSK MODULATION and DEMODULATION kit), CRO.

THEORY:

Digital communications systems represent the information by binary digits 1 and 0. There are various modulation techniques. This kit is for explaining the function of phase shift keying. In PSK, the phase of the carrier varies in accordance with the input bits 1 and 0. The PSK modulator generates output against the input data 1 and inverted output against the input data 0. PSK signal can be generated using a balanced modulator. PSK demodulator employs coherent detection. It consists of a carrier regenerator, demodulator, integrator, and decision device.

$V_{PSK}(t) = b(t) A\cos\omega_0 t$, where b(t) is -1 volt for input data 0 and 1 volt for input data 1. So, there are: **$- A\cos\omega_0 t$** transmission for input data 0 and **$A\cos\omega_0$** for input data 1.

BLOCK DIAGRAM of MODULATOR:

DEMODULATOR:

PROCEDURE:

1) Refer to the block diagram and carry out the following connections and switch
   settings.
2) Connect the power supply in proper polarity to the modulator and Demodulator kit (DCL-06) and switch it on.
3) Connect SERIAL DATA generated on board to CONTROL IN of CARRIER MODULATOR.
4) Select PSK modulation using switch S1, the PSK LED will glow.
5) Observe the waveforms at SERIAL DATA, SINE1, and MOD OUT.
6) Connect PSK modulated signal MOD OUT to the PSK IN of the PSK DEMODULATOR.
7) Observe waveforms on CRO.

Waveforms on graph paper:

Conclusion:

DC Experiment 14

Name:
Roll No:

Date:

Study of FSK Modulation and Demodulation

OBJECTIVE: To study Frequency Shift Keying modulation and demodulation.

EQUIPMENTS:
Patch cords, DCL-06(FSK MODULATION and DEMODULATION kit), CRO.

THEORY:
Digital communications systems represent the information by binary digits 1 and 0. There are various modulation techniques. This kit is for explaining the function of frequency shift keying. In FSK, frequency of the carrier varies in accordance with the input bits 1 and 0. The FSK modulator generates high frequency output against the input data 1 and low frequency output against the input data 0. FSK signal can be generated using two balanced modulators and an adder. FSK demodulator employs noncoherent detection. It consists of a bandpass filter which passes a high frequency carrier, full wave rectifier, low pass filter and decision device.

$V_{FSK}(t) = A\cos\omega_H t$ for input data 1 and $V_{FSK}(t) = A\cos\omega_L t$ for input data 0, where ω_H and ω_L are high frequency and low frequency for representing logic 1 and logic 0 respectively.

BLOCK DIAGRAM of MODULATOR:

DEMODULATOR:

PROCEDURE:

1) Refer to the block diagram and carry out the following connections and switch
settings.

2) Connect the power supply in proper polarity to the modulator and Demodulator kit (DCL-06) and switch it on.

3) Connect SERIAL DATA generated on board to CONTROL IN of CARRIER MODULATOR.

4) Select FSK modulation using switch S1, the FSK LED will glow.

5) Observe the waveforms at SERIAL DATA, SINE1, SINE2 and MOD OUT.

6) Connect PSK modulated signal MOD OUT to the FSK IN of the FSK DEMODULATOR.

7) Observe waveforms on CRO.

Waveforms on graph paper:

Conclusion: