Dhirubhai Ambani Institute of
Information and Communication
Technology

Convolution Code

CT-216: Introduction To Communication Systems

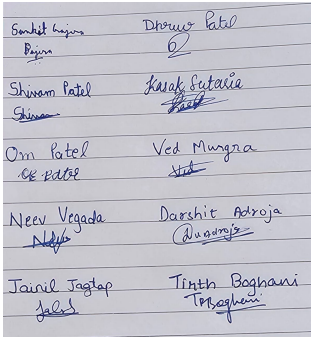
Prof. Yash Vasavada

Group 3

Honour Code

We declare that

- The work that we are presenting is our own work.,
- We have not copied the work (the code, the results, etc.) that someone else has done.,
- Concepts, understanding and insights we will be describing are our own.,
- We make this pledge truthfully. We know that violation of this solemn pledge can carry grave consequences.,



Contents

Problem Statement

We are tasked with implementing a convolutional encoder for different rates and constraint lengths, and analyzing the performance of the system under different conditions. Specifically, we need to implement the following:

- r $= 1/2$, Kc $= 3$,
- r $= 1/3$, Kc $= 4$,
- r $= 1/3$, Kc $= 6$.

The encoded output is then passed through a BPSK modulator, and transmitted over an AWGN channel that introduces a per-symbol SNR of $\frac{E_s}{N_0}$.

We will then perform both soft decision and hard decision Viterbi decoding and demonstrate the system performance by varying the SNR $\frac{E_b}{N_0} \in \{0, 10\}$ in steps of 0.5 dB. The goal is to produce curves showing the probability of detection errors as a function of $\frac{E_b}{N_0}$ for each rate r.

Contents

Introduction

- Convolutional codes were introduced in 1955 by Peter Elias.
- A convolutional code is a type of error-correcting code that generates parity symbols via the sliding application of a boolean polynomial function to a data stream. The sliding application represents the 'convolution' of the encoder over the data, which gives rise to the term 'convolutional coding'. The sliding nature of the convolutional codes facilitates trellis decoding using a time-invariant trellis.
- The most widely used application of convolutional codes is in wireless and mobile communication systems—especially in standards like 3G, 4G (LTE), and Wi-Fi.
- It is also used in Space Technology by NASA.

Introduction

- A convolutional encoder processes the information sequence continuously.
- The n-bit encoder output at a particular time depends not only on the k-bit information sequence, but also on m previous input blocks using shift registers.
- A convolutional encoder has a memory order of m.
- The set of sequences produced by a k-input, n-output encoder of memory order m is called an (n, k, m) convolutional code.
- The values of n and k are much smaller for convolutional codes compared to block codes.

Contents

Encoding

- In encoding, we first add padded bits (zeros) at the end of the input sequence of size $K-1$.
- We use the formula:

$$p_i[n] = \left(\sum_{j=0}^{K-1} g_i[j] \cdot x[n-j] \right) \mod 2$$

  to generate encoded bits for each sliding window. Here, the window size is K.
- For a $\frac{1}{N}$ rate encoder, we need to have a generator matrix of length N (i.e., N rows).
- The length of the encoded sequence will be:

$$N \times (k + K - 1)$$

where:

- K = length of a row of the generator matrix,
- k = length of the input sequence.

State Diagram and Encoder

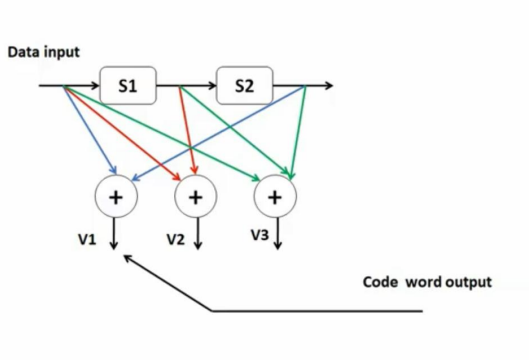


Figure: Block Diagram

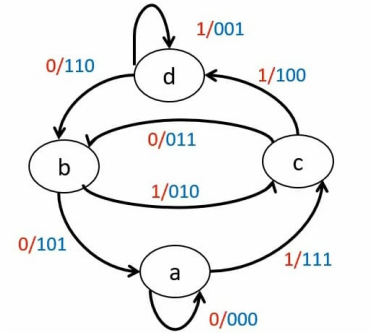


Figure: State Diagram

Truth Table

State	S1	S2
a	0	0
b	0	1
c	1	0
d	1	1

Figure: Truth Table 1: State Transition and Output

Input	Present state		Next state		Output		
data	S1	S2	S1	S2	V1	V2	V3
0	0	0	0	0	0	0	0
1	0	0	1	0	1	1	1
0	0	1	0	0	1	0	1
1	0	1	1	0	0	1	0
0	1	0	0	1	0	1	1
1	1	0	1	1	1	0	0
0	1	1	0	1	1	1	0
1	1	1	1	1	0	0	1

Figure: Truth Table 2: State Mapping

Trellis Diagram

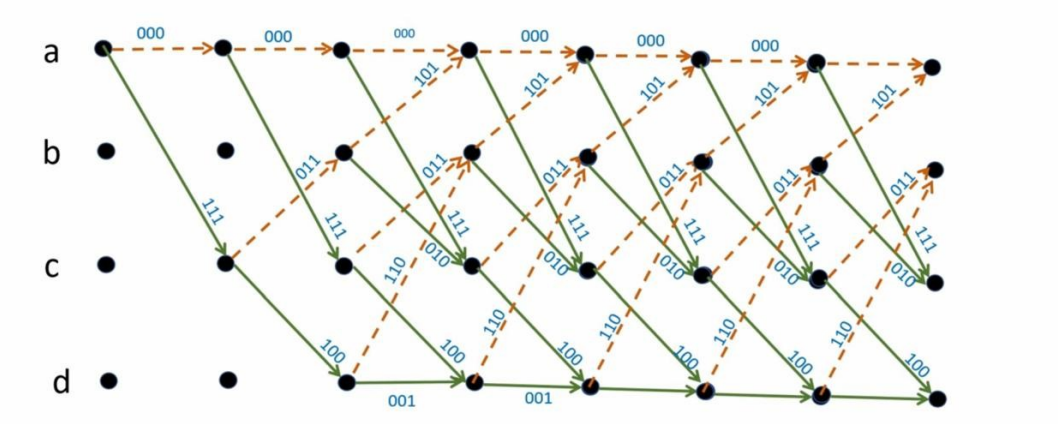


Figure: Trellis diagram representing state transitions

Contents

Modulation

We use BPSK Modulation in the encoded codeword.**BPSK (Binary Phase Shift Keying)** is a digital modulation scheme used in communication systems to transmit binary data (0s and 1s) over a communication channel.

- Bit 0 is modulated to the symbol 1.
- Bit 1 is modulated to the symbol -1.

Modulation formula Used:

$$s = 1 - 2b \quad \text{where } b \in \{0, 1\}$$

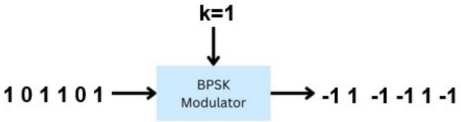


Figure: BPSK Modulation In Convolution Code

Contents

AWGN

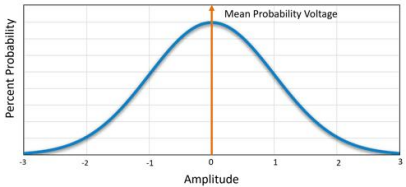
Additive White Gaussian Noise (AWGN) is characterized by being *additive*, having a constant power spectral density across frequencies (*white*), and following a Gaussian probability distribution.

The AWGN is typically modeled as a random variable with:

- Zero mean: $\mu = 0$
- Variance: σ^2

It is denoted as:

$$n \sim \mathcal{N}(0, \sigma^2)$$



Contents

Demodulation

- Demodulation is the reverse process of modulation i.e. extracting the information from the modulated signal.
- The received information is demodulated / digitized i.e. reverse mapping, symbols (voltage), s are converted to binary bits, b. Here, we have used BPSK modulator hence for Convolution code the demodulation scheme is,

$$b = \begin{cases} 1, & s < 0 \\ 0, & \text{else} \end{cases}$$

Contents

Decoding

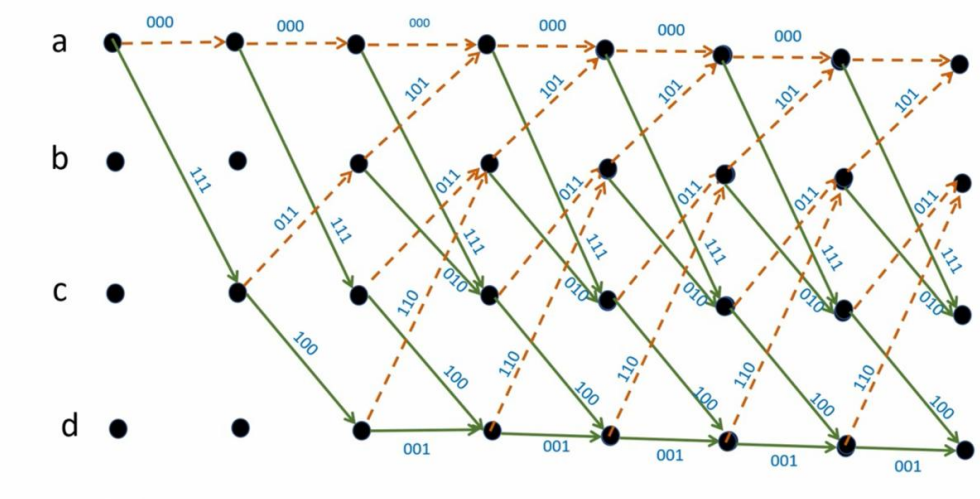
For Decoding, there are various algorithms to decode like Viterbi,BCJR. But for our project we will consider the Viterbi Algorithm.

- In the Viterbi algorithm, we use a trellis data structure. There are two types of Viterbi decoding techniques: HDD (Hard Decision Decoding) and SDD (Soft Decision Decoding).
- In the case of HDD, we use **Hamming Distance**.
- In the case of SDD, we use **Euclidean Distance**.
- Let's see the process of decoding using the Viterbi algorithm.

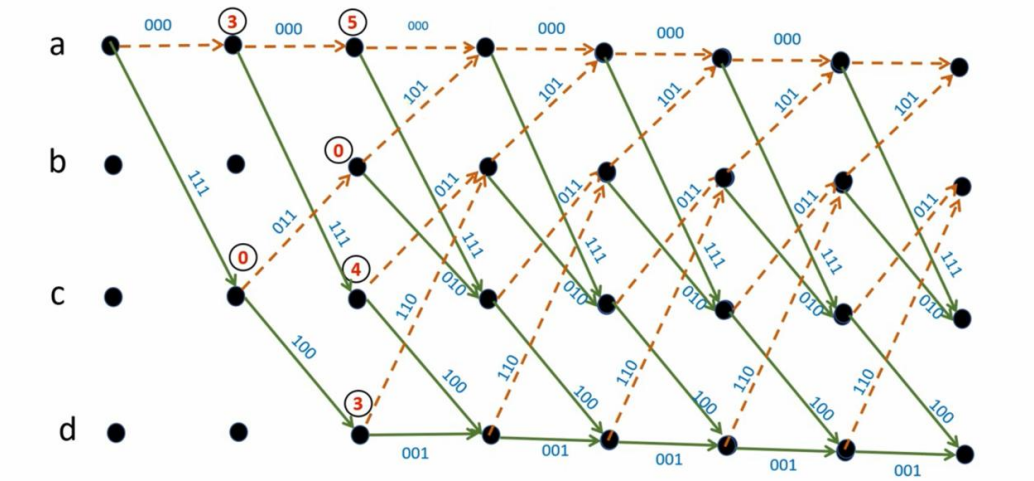
For example, let's consider the following:

- Original message: `[1 0 0 1 1 0 0]|`
- `Received message: [111 011 101 111 100 110 101]|`

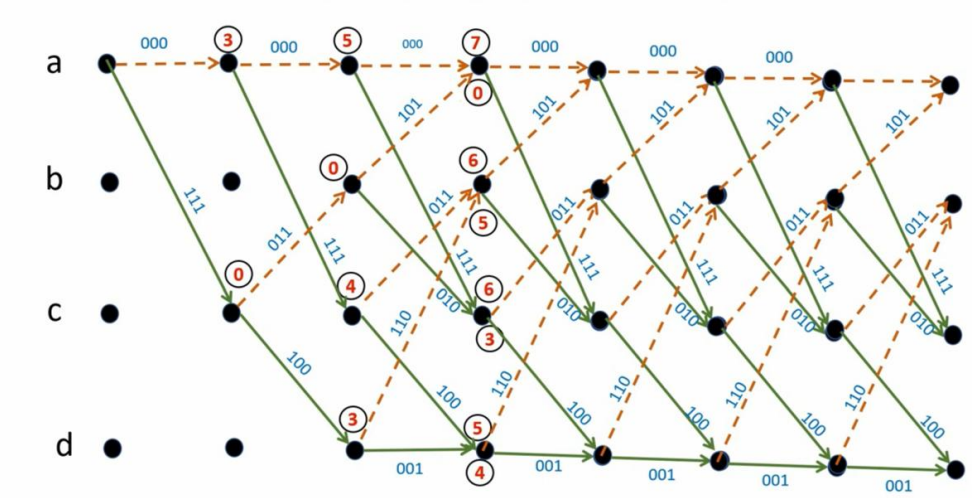
Hard Decision Decoding



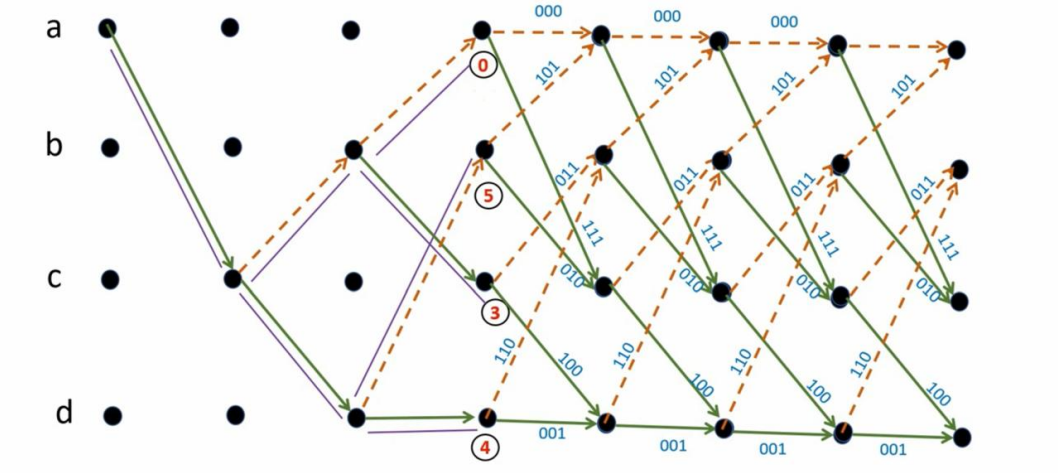
Hard Decision Decoding



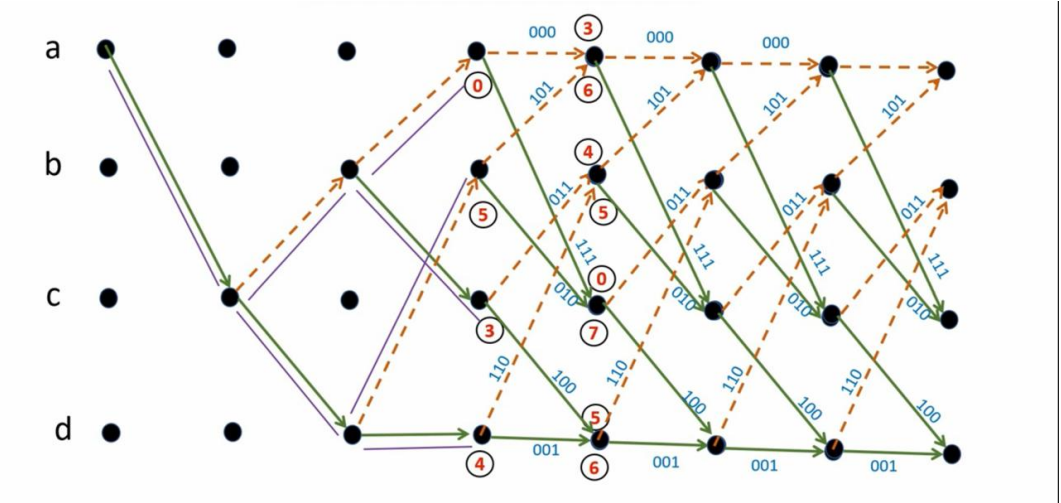
Hard Decision Decoding



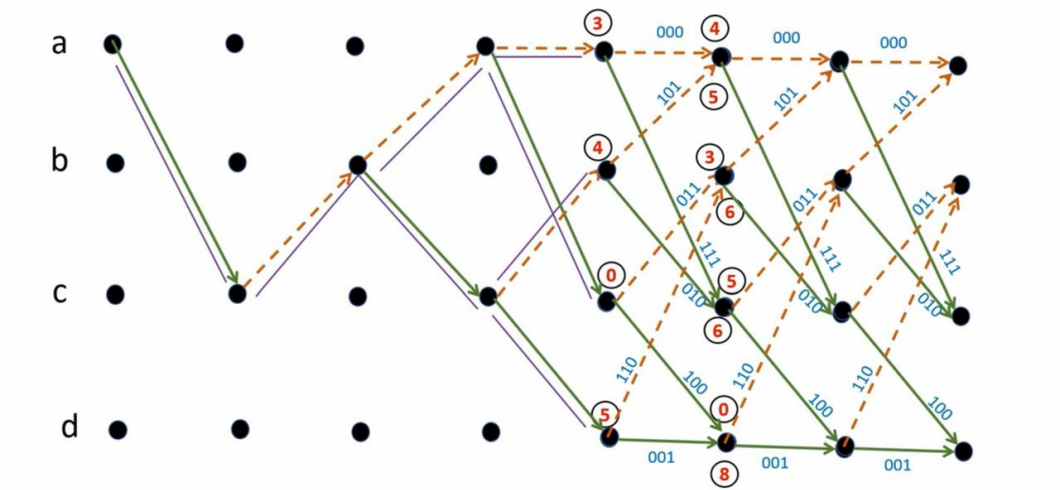
Hard Decision Decoding



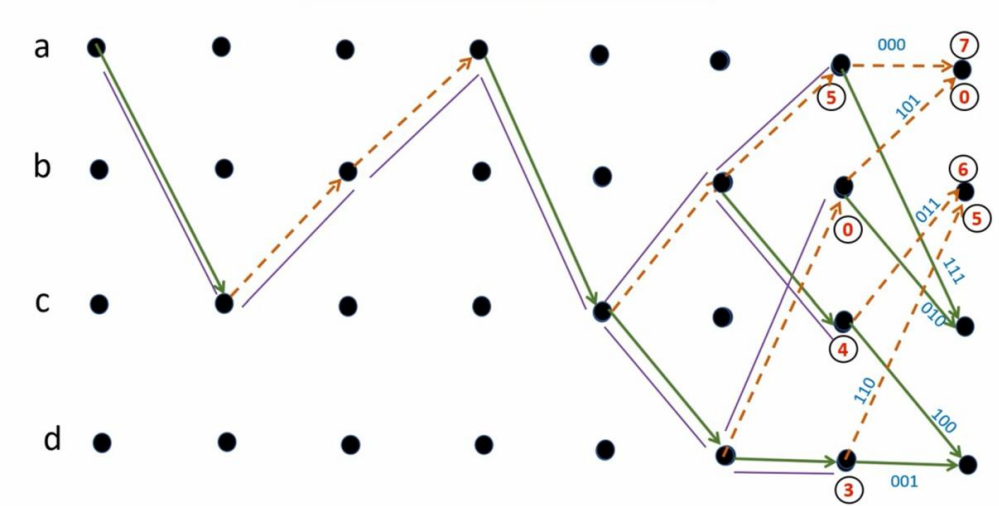
Hard Decision Decoding



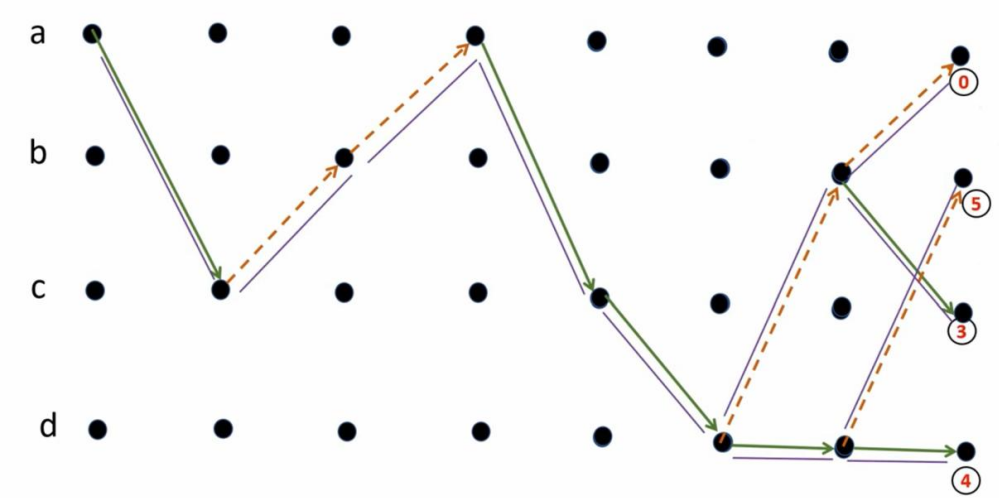
Hard Decision Decoding



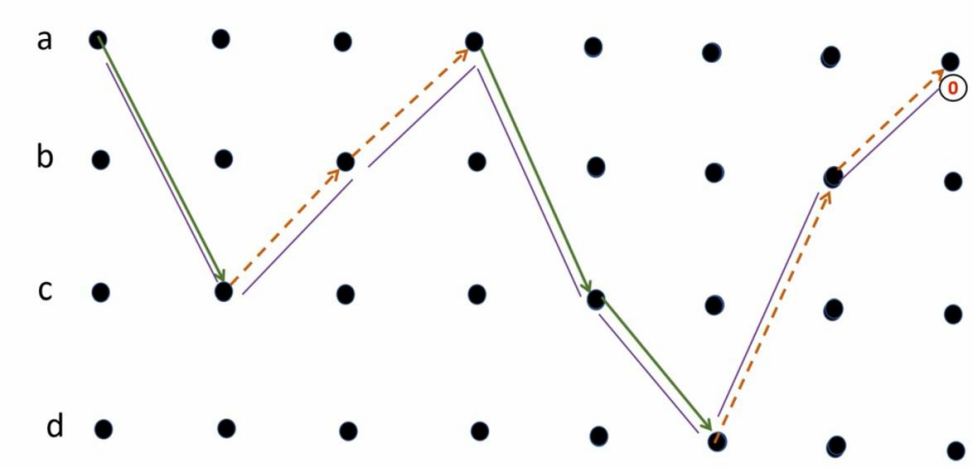
Hard Decision Decoding



Hard Decision Decoding



Hard Decision Decoding



Soft Decision Decoding

In principle, the soft decision decoding works on the Bayesian Theorem, whereas the hard decision decoding is an ad-hoc method that is preferred because it may be computationally simpler to implement.

In general, soft decision decoding improves the performance by $\approx$ 2 dB.

Instead of using hard decisions on the bits, soft decisions can be used. This requires that the input to the decoder be a Log-Likelihood Ratio (LLR) in the form:

$$L_i = \log\left(\frac{P(c_i = 1 \mid r_i)}{P(c_i = 0 \mid r_i)}\right)$$

For BPSK modulation in AWGN, the LLR becomes:

$$L_i = \frac{2\, r_i}{\sigma^2}$$

Soft Decision Decoding

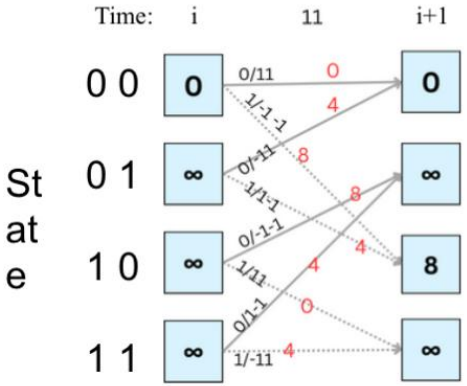
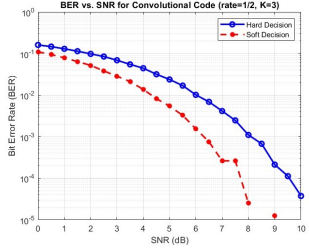
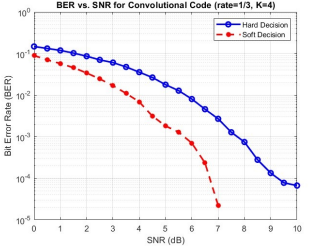
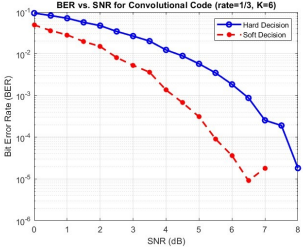


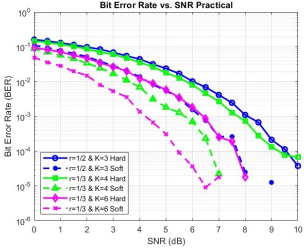
Figure: State diagram representation over time

Results


BER vs. SNR for Convolutional Code (rate=1/2, K=3)
Hard Decision
Soft Decision
Bit Error Rate (BER)
SNR (dB)

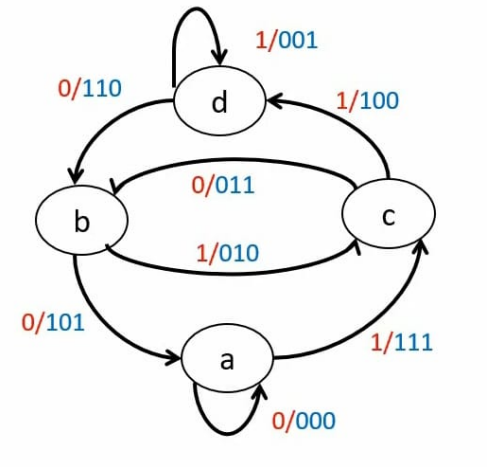

BER vs. SNR for Convolutional Code (rate=1/3, K=4)
Hard Decision
Soft Decision
Bit Error Rate (BER)
SNR (dB)


BER vs. SNR for Convolutional Code (rate=1/3, K=6)
Hard Decision
Soft Decision
Bit Error Rate (BER)
SNR (dB)

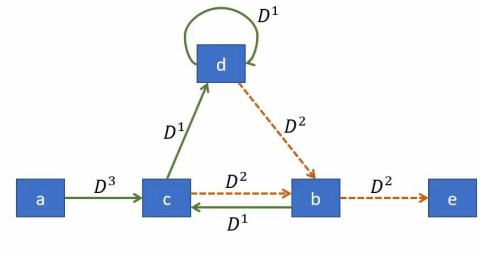

Bit Error Rate vs. SNR Practical
r=1/2 & K=3 Hard
r=1/2 & K=3 Soft
r=1/3 & K=4 Hard
r=1/3 & K=4 Soft
r=1/3 & K=6 Hard
r=1/3 & K=6 Soft
Bit Error Rate (BER)
SNR (dB)

Contents

Transfer Function



State Diagram



Transfer Function Representation

Transfer Function

$$X_c = D^3 X_a + D^1 X_b$$

$$X_b = D^2 X_c + D^2 X_d$$

$$X_d = D X_c + D X_d$$

$$X_e = D^2 X_b$$

$$Transfer\ function\ \mathrm{T(D)} = \frac{\mathrm{X_e}}{\mathrm{X_a}}$$

Transfer Function

Transfer Function of Convolution Code using Mason's Gain Formula

$$M = \frac{1}{\Delta}\sum_{k=0}^{N} P_k \Delta_k = T(D)$$

Here,

$N = number\ of\ forward\ paths,$

$P_k = gain\ of\ k_{th}\ forward\ path,$

$\Delta = 1 - (sum\ of\ loop\ gains\ of\ all\ indivdual\ loops) + (gain\ products\ of\ all\ possible\ two\ non\ touching\ loops\ ) + \cdots$

$\Delta_k = \Delta\ for\ that\ part\ of\ the\ graph\ that\ is\ not\ touching\ the\ forward\ path$

Transfer Function

$$\Delta = 1 - (D^1 + D^3 + D^4) + D^4$$

For, forward path acbe	For forward path acdbe
$P_k = D^7$	$P_k = D^8$
$\Delta_k = 1 - D$	$\Delta_k = 1$

$$T(D) = \frac{1}{\Delta}\sum_{k=0}^{N} P_k \Delta_k = \frac{1}{1 - D - D^3} * (D^7(1 - D) + D^8) = \frac{D^7}{1 - D - D^3}$$

$$T(D) = \frac{D^7}{1 - D - D^3}$$

We can define dfree as power of numerator term,i.e. 7 in this case.

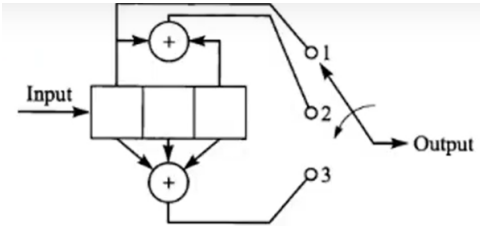
A Convolution Code with dfree can correct t errors if

$$dfree > 2t$$

Contents

Some Extra Knowledge

Polynomial Representation of Convolution Code



In this example, we are given a convolutional code with rate $\frac{1}{3}$ and constraint length $L = 3$. Let u be the input sequence to the encoder. We are also given the impulse responses g_1, g_2, g_3 from the encoder input to the three outputs.

The three outputs are given by:

$$c^{(1)} = u \otimes g_1$$

$$c^{(2)} = u \otimes g_2$$

$$c^{(3)} = u \otimes g_3$$

Some Extra Knowledge

Convolution in time domain is equivalent to multiplication in transform domain.

We define the D-transform of u as:

$$u(D) = \sum_{i=0}^{\infty} u_i D^i$$

Transform functions for the three impulse responses g_1, g_2, g_3 are:

$$g_1(D) = 1$$
$$g_2(D) = 1 + D^2$$
$$g_3(D) = 1 + D + D^2$$

The output transforms are given by:

$$c^{(1)}(D) = u(D) \cdot g_1(D)$$

Some Extra Knowledge

$$c^{(2)}(D) = u(D) \cdot g_2(D)$$
$$c^{(3)}(D) = u(D) \cdot g_3(D)$$

The final output after decoding is given by:

$$c(D) = c^{(1)}(D^3) + D \cdot c^{(2)}(D^3) + D^2 \cdot c^{(3)}(D^3)$$

Example: Let $u = (1\ 0\ 0\ 1\ 1\ 1)$

$$u(D) = 1 + D^3 + D^4 + D^5$$
$$c^{(1)}(D) = 1 + D^3 + D^4 + D^5$$
$$c^{(2)}(D) = 1 + D^2 + D^3 + D^4 + D^6 + D^7$$
$$c^{(3)}(D) = 1 + D + D^2 + D^3 + D^5 + D^7$$

Some Extra Knowledge

$$c(D) = 1 + D + D^2 + D^5 + D^7 + D^8 + D^9 + D^{10} + D^{11} + D^{12} + D^{13} + D^{15} + D^{17} + D^{19} + D^{22} + D^{23}$$

Corresponding code sequence:
$c = (111001011111110101010011)$

Why 24 bits instead of 18?
$\rightarrow$ Due to flushing of 0's for resetting the register.

Contents

Summary

1. **Convolutional Encoding:**
   Input bits are passed through shift registers and combined using generator polynomials to produce redundant output bits, enabling error correction.

2. **Modulation & Demodulation:**
   Encoded bits are mapped to signal waveforms (e.g., BPSK), transmitted, and then received through demodulation—either hard (binary) or soft (real-valued probabilities).

3. **Hard Decision Decoding:**
   Uses the Viterbi algorithm on binary demodulated bits to trace the most likely path through the encoder's trellis, minimizing Hamming distance.

4. **Soft Decision Decoding:**
   A more accurate version of Viterbi decoding that uses soft demodulated values (likelihoods), improving error correction by considering signal confidence levels.