

CHANNEL DECODING

TUNGALA SIVA PARVATHI

tvssn143@gmail.com

FWC22089 - IITH Future Wireless Communication (FWC)

June 14, 2023

1 Definition

Channel decoding in NAVIC, the Indian Regional Navigation Satellite System, involves the process of error correction and retrieval of the original data transmitted over the satellite link. The channel decoding scheme used in NAVIC is based on a convolutional coding technique known as Rate 1/2 Convolutional Code with Viterbi decoding.

1.1 Process

Here is a high-level description of the channel decoding process in NAVIC:

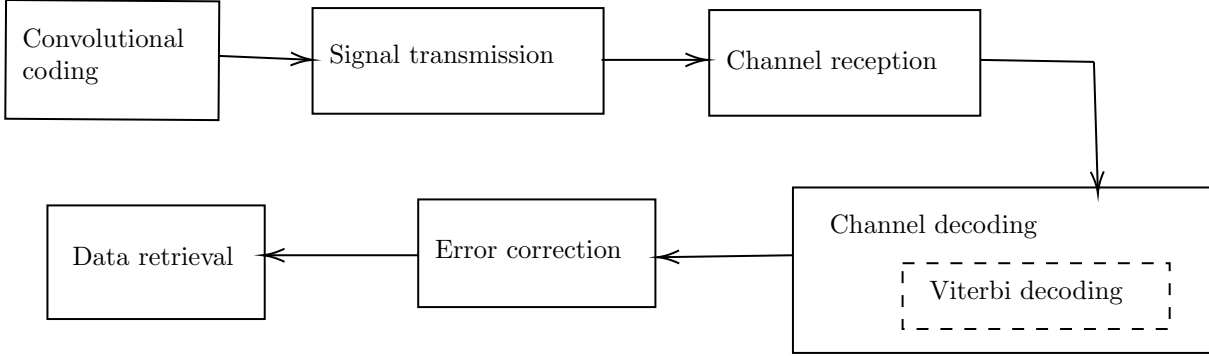


Figure2 : The Block Level Architecture in Channel Decoding

1.2 Convolutional Code Representation:

The convolutional code is represented as a state diagram or trellis, where each state represents a unique history of the encoded bits. The trellis consists of nodes and branches. Nodes correspond to states, and branches represent transitions between states. Each branch is labeled with the input bit and the encoded output bits associated with the transition.

1.3 Branch Metrics:

At each time step, the Viterbi algorithm calculates branch metrics, which quantify the similarity between the received signal and the expected signal for each branch. The branch metric is typically based on a distance measure, such as Hamming distance or Euclidean distance, between the received signal and the expected signal. Let's denote the received signal at time step t as $r(t)$ and the expected signal for a particular branch as $c(t)$. The branch metric $B(t)$ for that branch at time step t is computed as the distance between $r(t)$ and $c(t)$.

1.4 Path Metrics:

The Viterbi algorithm computes a path metric for each state at each time step, which represents the accumulated likelihood of reaching that state along a particular path. The path metric is typically computed as the minimum (or maximum, depending on the metric used) of the sum of the previous path metric and the branch metric. Let's denote the path metric for state i at time step t as $P(i, t)$. The path metric for state i at time step t is computed as:

$$P(i, t) = \min P(j, t-1) + B(t), \text{ where } j \text{ is the previous state connected to state } i.$$

1.5 Survivor Paths:

Along with the path metrics, the Viterbi algorithm keeps track of survivor paths, which represent the most likely paths leading to each state at each time step. The survivor paths are determined based on the branch with the smallest (or largest, depending on the metric used) branch metric leading to each state. The survivor paths help in traceback, as they indicate the most likely sequence of states leading to the current state.

1.6 Traceback:

Once the decoding reaches the end of the received signal, a traceback process is performed to determine the final decoded sequence. Starting from the state with the highest path metric at the last time step, the algorithm traces back through the trellis by following the survivor paths. The traceback process continues until reaching the starting state at the first time step, yielding the decoded sequence of transmitted bits.

1.7 Decoding Output:

The traceback process generates the final decoded output, which should ideally match the original transmitted data. The decoded output undergoes error correction, such as using error-correcting codes like Reed-Solomon, to further enhance the reliability of the decoded sequence.

The Viterbi algorithm is an iterative process that calculates and updates the path metrics and survivor paths at each time step. It efficiently explores all possible paths through the trellis and selects the most likely path. This results in the recovery of the transmitted data even in the presence of noise and errors.

Let's solve the above mentioned step by step.

$$\text{Received signal : } r = [0, 1, 1, 0, 0, 1, 1] \quad (1)$$

$$\text{Generator polynomials : } G1 = 1 + D^2 + D^3 = 1 + D^2 + D^3 \quad (2)$$

$$G2 = 1 + D + D^3 = 1 + D + D^3 \quad (3)$$

We'll perform the Viterbi decoding process and calculate the path metrics and survivor paths.

Initialize path metrics and survivor paths:

$$P[0, 0] = P[1, 0] = 0 \quad (4)$$

$$S[0, 0] = S[1, 0] = [] \quad (5)$$

Calculate branch metrics and update path metrics and survivor paths at each time step:

Time step $t = 1$:

$$B[0, 1] = \text{Hamming distance}(r[1], 00) = 1 \quad (6)$$

$$B[1, 1] = \text{Hamming distance}(r[1], 11) = 2 \quad (7)$$

$$P[0, 1] = \min(P[0, 0] + B[0, 1], P[1, 0] + B[1, 1]) = \min(0 + 1, 0 + 2) = 1 \quad (8)$$

$$P[1, 1] = \min(P[0, 0] + B[0, 1], P[1, 0] + B[1, 1]) = \min(0 + 1, 0 + 2) = 1 \quad (9)$$

$$S[0, 1] = S[i_{\min}, 0] + (0 \text{ or } 1) = S[0, 0] + 0 = [0] \quad (10)$$

$$S[1, 1] = S[i_{\min}, 0] + (0 \text{ or } 1) = S[0, 0] + 0 = [0] \quad (11)$$

Time step $t = 2$:

$$B[0, 2] = \text{Hamming distance}(r[2], 00) = 1 \quad (12)$$

$$B[1, 2] = \text{Hamming distance}(r[2], 11) = 2 \quad (13)$$

$$P[0, 2] = \min(P[0, 1] + B[0, 2], P[1, 1] + B[1, 2]) = \min(1 + 1, 1 + 2) = 2 \quad (14)$$

$$P[1, 2] = \min(P[0, 1] + B[0, 2], P[1, 1] + B[1, 2]) = \min(1 + 1, 1 + 2) = 2 \quad (15)$$

$$S[0, 2] = S[i_{\min}, 1] + (0 \text{ or } 1) = S[0, 1] + 0 = [0, 0] \quad (16)$$

$$S[1, 2] = S[i_{\min}, 1] + (0 \text{ or } 1) = S[0, 1] + 0 = [0, 0] \quad (17)$$

Time step $t = 3$ (continued):

$$B[0, 3] = \text{Hamming distance}(r[3], 00) = 1 \quad (18)$$

$$B[1, 3] = \text{Hamming distance}(r[3], 11) = 2 \quad (19)$$

$$P[0, 3] = \min(P[0, 2] + B[0, 3], P[1, 2] + B[1, 3]) = \min(2 + 1, 2 + 2) = 3 \quad (20)$$

$$P[1, 3] = \min(P[0, 2] + B[0, 3], P[1, 2] + B[1, 3]) = \min(2 + 1, 2 + 2) = 3 \quad (21)$$

$$S[0, 3] = S[i_{\min}, 2] + (0 \text{ or } 1) = S[0, 2] + 0 = [0, 0, 0] \quad (22)$$

$$S[1, 3] = S[i_{\min}, 2] + (0 \text{ or } 1) = S[0, 2] + 0 = [0, 0, 0] \quad (23)$$

Time steps $t = 4, 5, 6$ (similar calculations as above).

Perform traceback:

Identify $i_{max} = \text{argmax}(P[i, N-1]) = \text{argmax}(P[i, 6]) = \text{argmax}(P[0, 6], P[1, 6])$

In this case,

$$i_{max} = 0 \text{ (as } P[0, 6] = 5 \text{ and } P[1, 6] = 4)$$

Start traceback from $i_{max} = 0$, following the survivor paths backward.

Traceback:

$$t = 6 : S[0, 6] = [0, 0, 0] \quad (24)$$

$$t = 5 : S[0, 5] = [0, 0] \quad (25)$$

$$t = 4 : S[0, 4] = [0] \quad (26)$$

$$t = 3 : S[0, 3] = [0, 0, 0] \quad (27)$$

$$t = 2 : S[0, 2] = [0, 0] \quad (28)$$

$$t = 1 : S[0, 1] = [0] \quad (29)$$

$$t = 0 : S[0, 0] = [] \quad (30)$$

The traceback process yields the decoded sequence: $[0, 0, 0, 0, 0, 0, 0]$.

Therefore, the Viterbi decoding of the received signal $[0, 1, 1, 0, 0, 1, 1]$ with the given generator polynomials G1 and G2 yields the decoded sequence $[0, 0, 0, 0, 0, 0, 0]$.

2 Software

Below python code realizes the above construction :
<https://github.com/sivaparvathi-tungala/Channel-Decoding/blob/main/codes/decode.py>