High-Speed Viterbi Decoder

Mário Pereira Véstias
https://orcid.org/0000-0001-8556-4507
INESC-ID, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Portugal

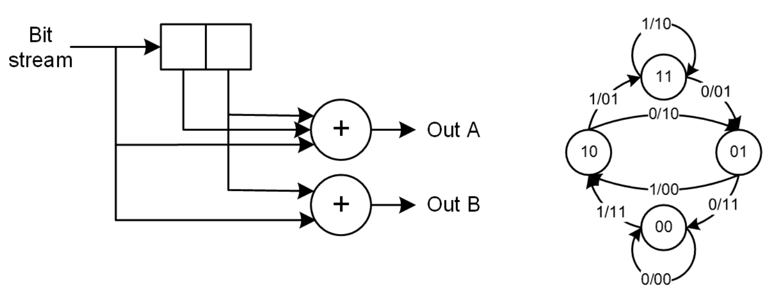
BACKGROUND

Generically, a communication system includes a channel encoder at the transmitter and a channel decoder at the receiver. Encoding the bit stream to be transmitted reduces the energy per data bit needed to achieve the same bit error rate (BER) over a noisy channel compared to a non-coded transmission. A better channel code needs lower energy to obtain a particular BER.

Convolutional Encoder

Two major classes of binary codes are block codes and convolution codes. In this article, the authors are only concerned with convolution codes (P. Elias, 1955). Convolution codes operate over a continuous bit stream. A convolutional encoder is used to generate the encoded bitstream. A typical implementation consists on a shift register and a set of modulo-2 addition. The content of the shift register defines the state of the encoder (s bits), from which a finite state machine with $m=2^s$ states is obtained. Figure 1 shows an example of a convolution encoder with rate 1/2 (rate 1/c generates c bits for each input bit) and its state machine.

Figure 1. Convolution encoder with rate 1/2



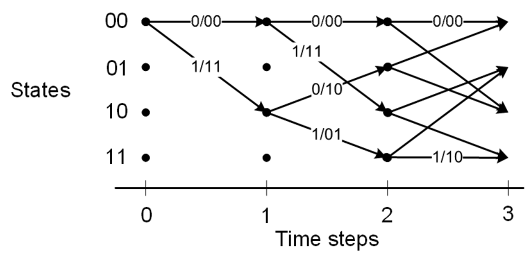
An alternative way to represent the behavior of the encoder was proposed by Forney (G. D. Forney, Jr., 1967) designated *trellis representation*. A trellis is one of the most convenient ways to visualize the behavior of the decoding algorithms (see Figure 2).

Each node in the trellis represents a state of the FSM. After m time steps, the trellis is full and the branch pattern repeats indefinitely.

Most work on trellis decoding was dedicated to the codes with a rate of $1/r$. This type of codes will provide the simplest trellis decoding since there are only two nodes leaving and entering a state.

DOI: 10.4018/978-1-7998-3479-3.ch019

Figure 2. Trellis diagram of the encoder in figure 1



Viterbi Algorithm

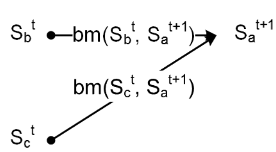
The Viterbi algorithm (VA) was proposed in 1967 by Andrew Viterbi (Viterbi, 1967). The algorithm decodes convolutional codes with an optimum non-sequential algorithm. Contrary to previous algorithmic solutions whose complexity increases exponentially with the length of the encoded sequence, the computational complexity of the Viterbi algorithm increases linearly with the length of the bit stream. The algorithm has three main steps: (1) branch metric calculation; (2) trellis calculation; and (3) traceback decoding.

The branch metric, *bm*, calculation determines the squared Euclidian distance, between the received symbol, r_k, and the expected symbol associated with a branch of the trellis diagram, c_k, that is, $|r_k-c_k|^2$. At every time step, $2\times m$ branch metrics are calculated since each state has two input branches.

The linear computation complexity of the VA is achieved by discarding unlikely branches in the trellis diagram using the Principle of Optimality (Bellman and Dreyfus, 1962). According to this, whenever two paths of the trellis diagram merge into the same state only the path with the best metric (most likely) must be kept. This path is designated *survivor path*. At each time step, this operation is applied to all paths and *m* survivor paths are obtained at each time step. Formally, a state metric, *sm(state)* (see figure 3), is updated as follows:

$$sm(S_a^{t+1}) = \min(S_b^t + bm(S_b^t, S_a^{t+1}), S_c^t + bm(S_c^t, S_a^{t+1})) \quad (1)$$

Figure 3. State metric update



At each time step, the trellis calculation step obtains *m* decision symbols, one for each surviving branch. These symbols must be stored to help recover the symbols responsible for specific transitions during the traceback decoding step.

The trellis computation stops after a predetermined number of steps or after the decoding of the complete stream. From the *m* survivor paths, the final step (traceback decoding) starts in the state with the best metric and tracebacks through the path identified by the decision bits until reaching the first state and, consequently, obtain the most likely sequence of symbols.