Course: EEE2020 Data Structure

Homework 6   **Due date: 2012.12.11 Before class starts**

Things you should do for this homework.

(1) **Upload *Report File* on YSCEC** (Report file should contain 'Flowchart', 'source code' and 'results')

(2) **Upload all *source codes* on YSCEC**.

(3) ***Print your Report File***, and ***hand it in*** before class

Late submissions will NOT be accepted.

RSC Code Decoder

In this project, you are to write a program which decodes the Recursive Systematic Convolutional (RSC) code (The encoding program of the RSC code will be provided). The decoding of RSC code reconstructs the original signal with bit error rate (BER) improvement (compared with the case where there is no encoding and decoding process).

RSC Code is an encoding process which improves the BER in communication theory. This encoding process is done by summing up previous inputs (for example, like Figure1) to make two outputs (which should be transferred). Since there are two signals (or codes) the length of code becomes twice as long as the input signal. Figure 1 shows an example of RSC encoder. We can generate the convolutional code using the encoder given below.

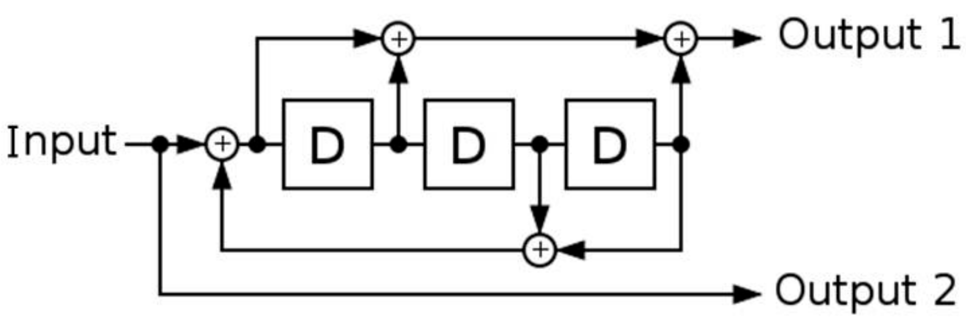


Figure 1. *Convolutional Systematic Convolutional Encoder*

From the RSC encoder, the state of system and the output may change depending on the input signal. The receiver has only access to the two output codes. The Trellis Diagram shows the relationship between the outputs and the state transition. Figure 2 is an example of Trellis diagram.

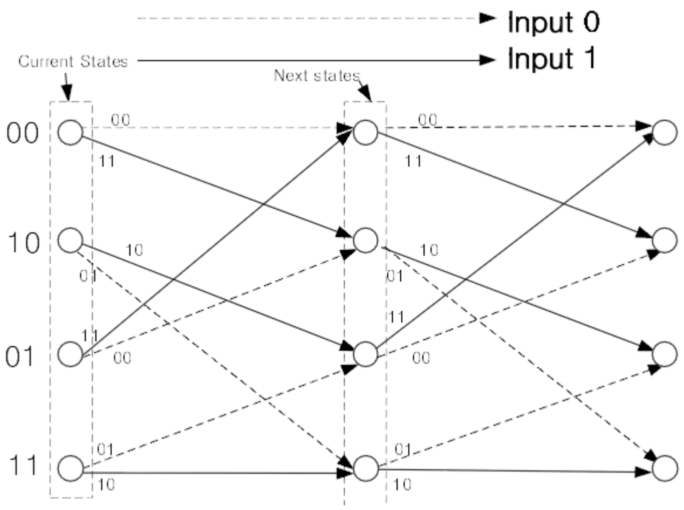


Figure 2. Example *Trellis Diagram (solid: input 1, dashed: input 0)*

For an example, when the initial state is 10 and the current input is 1, the output of encoder is 10 and the state will be changed to 01. When the state is 11 and the current input is 0, the output of encoder is 01 and the state will be changed to 01. Likewise, we can decode the output signal in order to estimate the input signal and state transition.

In the encoding process, RSC Code starts with its state 00, and should end with state 00. In this project, you should decode the attached output signal to estimate the state transition and input signals. For this decoding process, you should use an algorithm called Viterbi Algorithm.

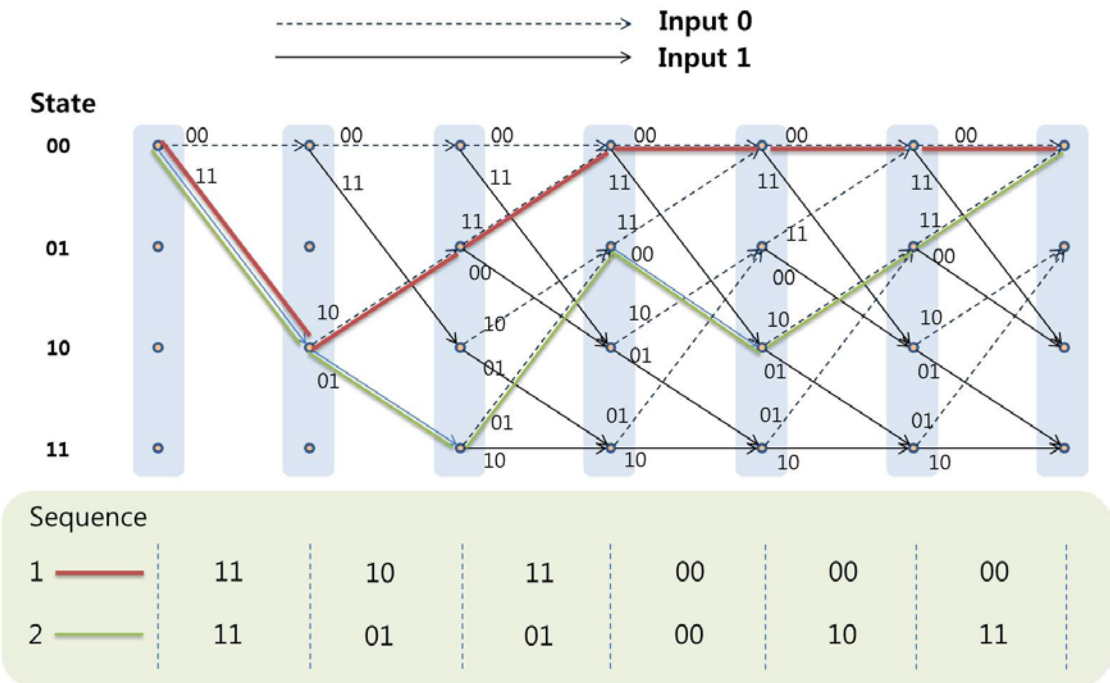


Figure 3. *Example of Viterbi Decoding for convolutional code*

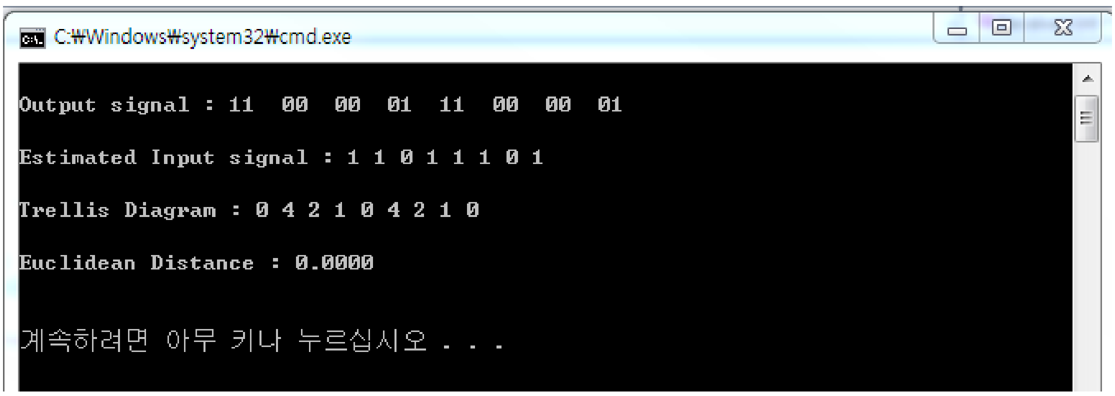
We can decide the path (a sequence of state transition) from the input signal. For example, see “sequence 1 (red line)” from Figure 3. The given output sequence is “11, 10, 11, 00, 00, 00” which means that the state changes in a series of “00, 10, 01, 00, 00, 00, 00”. Therefore, the estimated input signal is ‘1, 0, 0, 0, 0’. For the “sequence 2 (green line)”, the given output sequence is “11, 01, 01, 00, 10, 11”, while the state follows the sequence “00, 10, 11, 01, 10, 01, 00”. Therefore, the input signal to be estimated is ‘1, 1, 0, 1, 0, 0’. In this way, using Viterbi algorithm with a given output sequence, we can estimate the input signal and state transition.

While the signal is transmitted, the convolutional code is corrupted by noise. The received output signal may not be identical with the transmitted output signal. We should consider noise added to the output signal. We consider all possible paths of Trellis diagram, and choose a path to minimize the Euclidean distance between the path and noisy output signal. For example, when a given sequence of output signals is “11, 01, 01, 00, 10, 10”, compare with the sequence 1 (11, 10, 11, 00, 00, 00) to calculate the Euclidean distance. For the first two outputs, the Euclidean distance between 11 and 11 is 0. The Euclidean distance between 10 and 01 is $\sqrt{(1-0)^2+(0-1)^2}$ $=\sqrt{2}$. The distance between 11 and 01 is 1. Thus, the total Euclidean distance with sequence 1 is 0 + $\sqrt{2}$ + 1 + 0 + 1+ 1 = 3 + $\sqrt{2}$. In a similar way, the total Euclidean distance with sequence 2 (11, 01, 01, 00, 10, 11) is 0+0+0+0+0+1 = 1

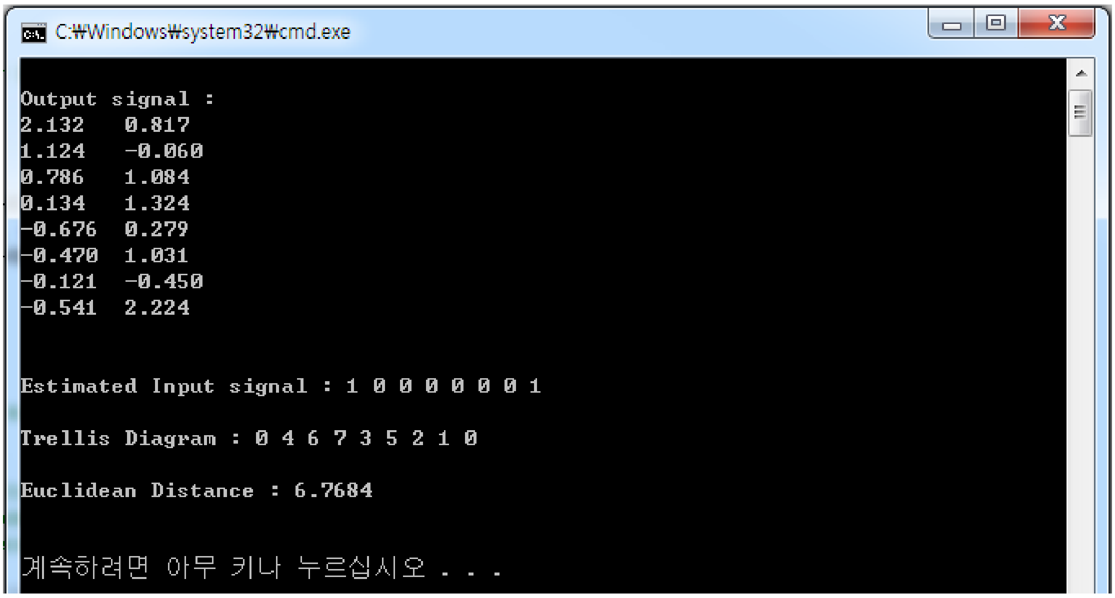
To compare these two sequences, the distance with the sequence 2 is less than sequence 1. We decide the uncorrupted output signal is sequence 2 among a set of sequences (here, when we assume that all possible sequences are sequence 1 and sequence 2). With this sequence decision, we can estimate the input signal and state transition. Likewise, we can calculate the total Euclidean distance for all possible paths, and choose the path with the minimum distance.

Program Example

① Noise free output signal



② Noise corrupted signal



```
C:\Windows\system32\cmd.exe

Output signal :
2.132   0.817
1.124   -0.060
0.786   1.084
0.134   1.324
-0.676  0.279
-0.470  1.031
-0.121  -0.450
-0.541  2.224


Estimated Input signal : 1 0 0 0 0 0 0 1

Trellis Diagram : 0 4 6 7 3 5 2 1 0

Euclidean Distance : 6.7684


계속하려면 아무 키나 누르십시오 . . .
```

(1) With the attached noise-free output signal, choose a path to minimize the Euclidean distance by using the Trellis diagram which is shown below.

(2) Estimate its input signal by using the Trellis diagram which is given below.

(3) Repeat (1) and (2) with noise-corrupted signals.

Trellis Diagram for this project