Advanced Communication Systems, ELEN90051 Workshop week 11 (=*week of 14 May 2017*): Convolutional Coding

Department of Electrical and Electronic Engineering
The University of Melbourne

Written by Margreta Kuijper and Xiangyue Meng, May 2017, updated May 8, 2018

1 Introduction

A convolutional encoder generates parity bits in a different way than a block encoder: the convolutional encoder has memory and employs a a sliding window. In fact, the encoded bit sequence can be interpreted as the output sequence of an input-state-output discrete time system that takes the information bit sequence as its input sequence. Convolutional codes have been around for a long time and are currently adopted in wireless standards, such as IEEE 802.11. In this workshop, you will get to

1. understand the principles of convolutional encoding methods
2. understand the principles of convolutional decoding methods based on the Viterbi algorithm
3. implement a convolutional encoder as well as several Viterbi decoding algorithms
4. research the history and applications of convolutional coding.

2 Background

See lecture notes Chapter ”Codes on Graphs” as well as textbook pp.491-500 and 505-516.

3 Organisation and assessment

You are expected to **be prepared before attending the workshop session**. For this, an **individual** pre-workshop report answering all questions of the Pre-Lab Task below, worth 12 marks, is to be submitted on LMS before the start of this workshop session.
For your workshop preparation you are also advised to try and answer remaining questions, as many as you can – this will help you use the two hours of the workshop session effectively.

You will be working in groups of 2 students, you can choose your own partner or, alternatively, you can ask your workshop demonstrator to assign a partner to you. Each group needs to submit one group report, worth 50 marks. In the report, you should include the answers to all questions of the Lab Tasks below as well as the section ”Reflections and Comparisons”, also include your **annotated** Matlab codes, describing clearly how your Viterbi decoders are implemented.
The report is to be submitted on LMS before the start for your workshop session in week 12 (=the week of 21 May).

4 Pre-Lab Task (12 marks)

This task deals with the basic principles of a convolutional encoder. Suppose that the code has 2 **generators** with coefficients $g_1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ and $g_2 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$, respectively. Recall that then the convolutional codes' difference equations are given by

$$y_i[t] = \left(\sum_{j=0}^{2} g_i[j] u[t-j] \right) \mod 2 \quad \text{for } i = 1, 2.$$

1. What is the constraint length L of this code?
2. What is the code rate of this code?
3. Give the shift register block diagram of this code.
4. Give the state machine diagram of this code, clearly labeling the state values and the input/output transitions. Which state value is the initial state value?
5. Draw the trellis diagram for this convolutional code and clearly distinguish the different input values and also label the state values and the output values.
6. Give matrices A, B, P and F so that the state representation (A, B, P, F) represents the convolutional code.
7. Explain **hard decision decoding** and **soft decision decoding**. What metric is used respectively?
8. Give the factor graph (see lectures) that corresponds to the hard decoder; also give the factor graph that corresponds to the soft decoder.

5 Lab Tasks

5.1 Lab Task 1 (12 marks)

In this task you will implement the above convolutional encoder in Matlab. More specifically, your task is to write a Matlab script function

```
EncodedCodeSequence=ConvEncoder(SourceSequence,GenPoly)
```

where `SourceSequence` is the sequence of the source bits, `GenPoly` is the matrix of the coefficients of the generator polynomials, and `EncodedCodeSequence` is the output of the convolutional encoder.

5.2 Lab Task 2 (8 marks)

This task deals with the basic principles of a convolutional decoder. Assume that the received sequence is `1101011000`.

1. Use the **exhaustive search** method to determine the sequence that was most likely transmitted. State all assumptions that you are making about the source and about the channel.

2. Copy your trellis diagram of the pre-lab task 5, writing the received bit sequence on the top, as in lectures. Suppose that you use a Viterbi ML hard decision decoder. Indicate in your trellis diagram: the **branch metrics** and the **path metrics**. Which sequence does the Viterbi ML hard decision decoder produce? Compare with your answer of the previous question.

3. To get a Viterbi ML **soft** decision decoder, what needs to be changed in your Viterbi trellis diagram of the previous question?

5.3 Lab Task 3 (18 marks)

In this task you will implement the above convolutional decoders in Matlab and then verify and evaluate your decoders via simulation.

1. Write a Matlab script function

   ```
   DecodedCodeSequence=ConvDecoder(ReceivedSequence,Type)
   ```

   where `ReceivedSequence` is the sequence of received values, `Type` specifies whether it is a hard-decision decoder or a soft-decision decoder, and `DecodedCodeSequence` is the output of the convolutional decoder.

2. Generate a random one-thousand-bit source sequence and encode this source sequence by `ConvEncoder`. Assume BPSK modulation, AWGN channel and a ML demodulator. Plot the BER against the SNR for hard decision and soft decision respectively, by varying the variance of the white Gaussian noise as $0.001 \leq \sigma^2 \leq 1$. Also plot the BER against the SNR for the uncoded case.

3. Reflect on your findings in the previous question. Compare and explain.

6 Reflections and Comparisons (12 marks)

1. Research and write about the history and applications of convolutional codes, encoders and decoders (1 page single spaced 11pt font). Add an extra page with a list of references (only include references that you refer to in the text); as a guideline, **use the file "ieecitationref" downloadable from LMS**; it is fine to use wikipedia in your research, however do not use it in your references, instead refer to original papers and sources.

2. In this subject ELEN90051 you saw two different uses for the Viterbi algorithm. In this workshop it is used for the decoding of a linear convolutional code. Earlier you saw it used for equalization purposes. If you were to use your MATLAB code of the Viterbi equalizer, what would you need to change to get a soft decision convolutional decoder? What to get a hard decision convolutional decoder?

End of workshop questions