Implementation of Viterbi decoding of convolutional code

Yanan Chen

December 17, 2015

Problem Introduction

Implementation of soft-input Viterbi decoding of convolutional codes over AWGN channel.

1.Matlab or C simulation program of the (5,7) and (15,17) ,rate 1/2 convolutional codes(with memory length of 2 and 3 respectively) over AWGN channel with BPSK modulation.

2.Compare your results with the uncoded BPSK on a log-log scale(e.g.,)where is the energy per information bit which is related to the energy per binary phase shift keying(BPSK) coded via (is the variance of the AWGN).

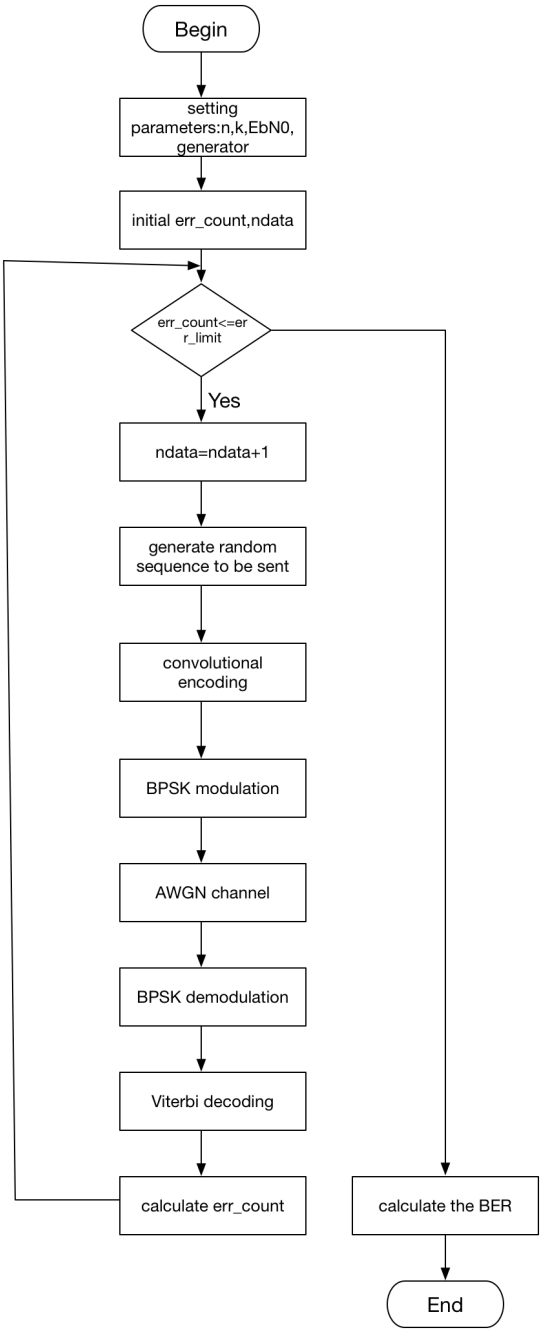
3.Compare the coding gains observed in simulations with the nominal gain$10\log_{10} d_{free}R$ .To run reliable Monte-Carlo simulations of the error probability per bit (BER) over an AWGN channel,a rule of thumb is to keep running the simulation at the given value of until you count 100 information bit errors at the output of the decoder.

4.Compare your result with the viterbi algorithm over a sliding window of length 10K where K is the code constraint length.

Parameters Setting

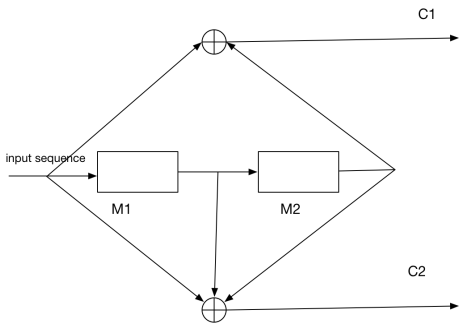
	(5,7)	(15,17)
Input (k)	1	1
Output(n)	2	2
Input Length(bit)	1000	1000
constraint Length	3	4
Eb/N0(in dB)	0:8	0:8

Flow Chart Illustration



Design Plan

1.The encoding process



the SR structure of (5,7)code

To generate corresponding convolutional code,we can have two ways(to simplify the description process,we use only (5,7)as an example,the (15,17) code is the same process as (5,7) is).

On one hand,for MATLAB,it has embedded function 'code=convenc(msg,trellis)' to generate the corresponding convolutional code given the message sequence to be translated and trellis structure of generator.

So we need to get 'trellis',we can easily realize it by using function:

trellis=poly2trellis(ConstraintLength,CodeGenerator)

On the other hand,we can generate the code by using 'generator' directly.For (5,7)code,the generator polynomial is $G(D)=[1+D+D^2,1+D^2]$.

If msg=(1 1 0 1 1…),then the convolutional code for this is C=(11,01,01,00,01,01,…).We separate 'G' into two columns,g1,g2.,therefore,C1=conv(msg,g1);C2=codec(m,g2).That means we do the convolution separately.This is the whole idea.

2.Transmitted process

At this code,we use BPSK to do the modulation.

$$2*m-1=\begin{cases}1 & if \quad m=1\\ -1 & if \quad m=0\end{cases}$$

for demodulation,the idea is shown below.

$$demo=\begin{cases}1 & if \quad receive>0\\ 0 & if \quad reveive\le 0\end{cases}$$

3.Viterbi decoding process

Viterbi algorithm is achieved by Adding - Compare - Select process, the state metric is calculated as follows : the state metric points on the first two states and the corresponding branch metric sum of two possible path metrics obtained as a measure of the new state candidates , sent them to logic unit for comparison , in which the maximum likelihood (smallest distance) is stored as a new state metric state , while storing the state of the recording as well as a new path , the main steps of the algorithm is :

a.put the received sequence into each of m group whose length is n0;

b.draw the trellis figure,and for the last L(the number of memories) spots,just draw the lines with input '0'.

c.set 's=0' and the initial state of all-zero;

d.for all branches connecting s to s+1,we get the Hamming distance between branch output and the received output.

e.do the summation of Hamming distance before 's' state and the new branch distance,they are candidate for 's+1' state

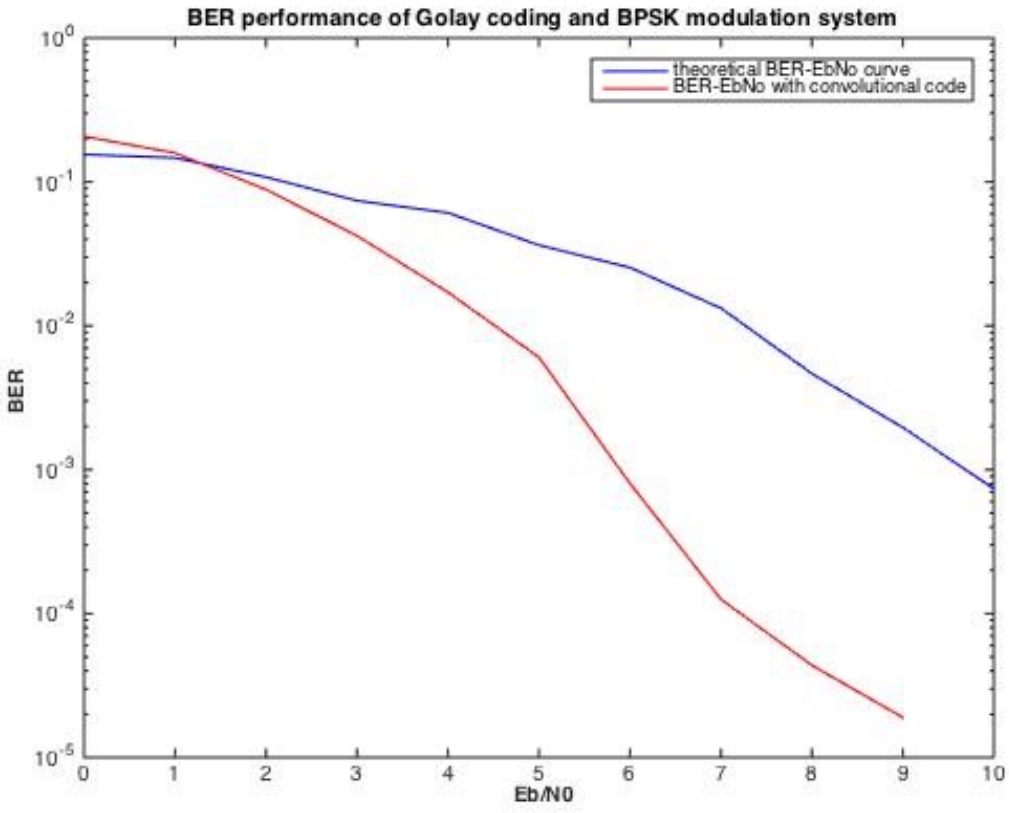
f.choose the smaller one as the saved path and state 's+1'

g.if s=m,to the next step;or s=s+1 and go back to d;

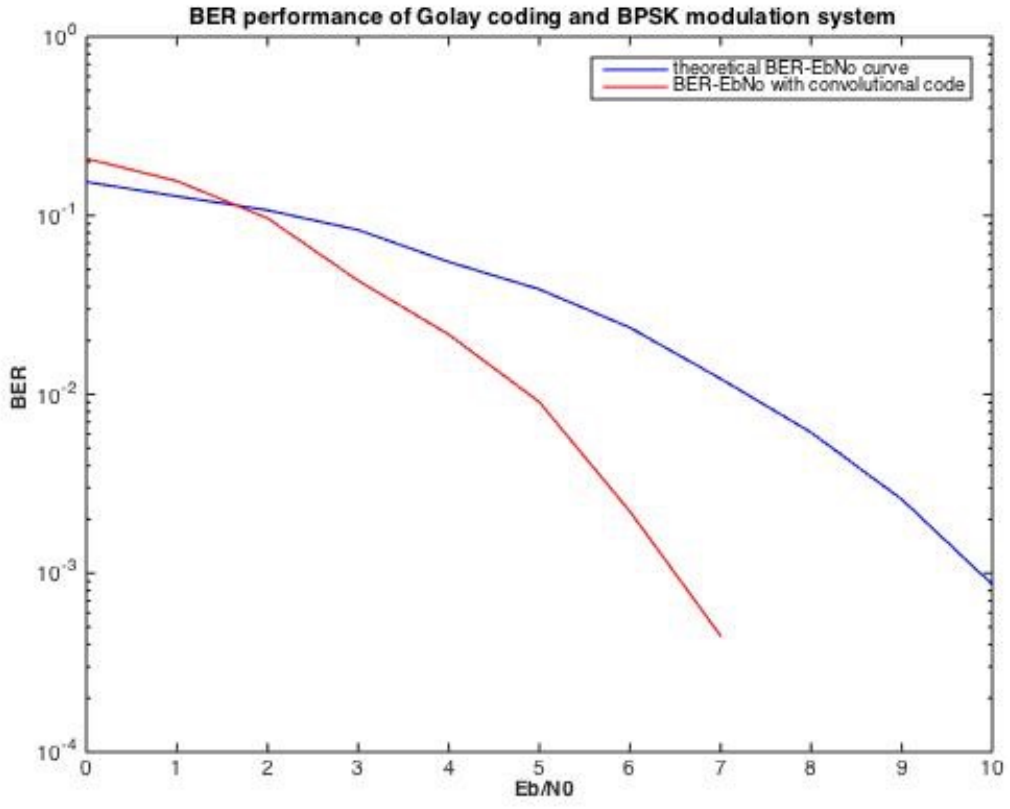
h.when we reach m,for m+1,we start at 'input=0' until the end.and we look at the reserved path,that is the best choice we make.

Result illustration

Firstly,I use the embeded MATLAB function of viterbi decoding to implement this process,and the results have been shown below.
1.Result of (7,5)code



2.Result of (15,17)code



a.From the curve of Eb/N0-BER curve,we can easily discover that the BER has decreased a lot compared to the uncoded data transmitted in the channel.When the SNR is small,uncoded one has lower BER,that is because too many errors are transmitted that the received data is almost useless.But with the increase of SNR,the performance of convolutional code has improved a lot.In the second figure,it is clear that lower Eb/N0 is needed at the same bit error ratio limit.

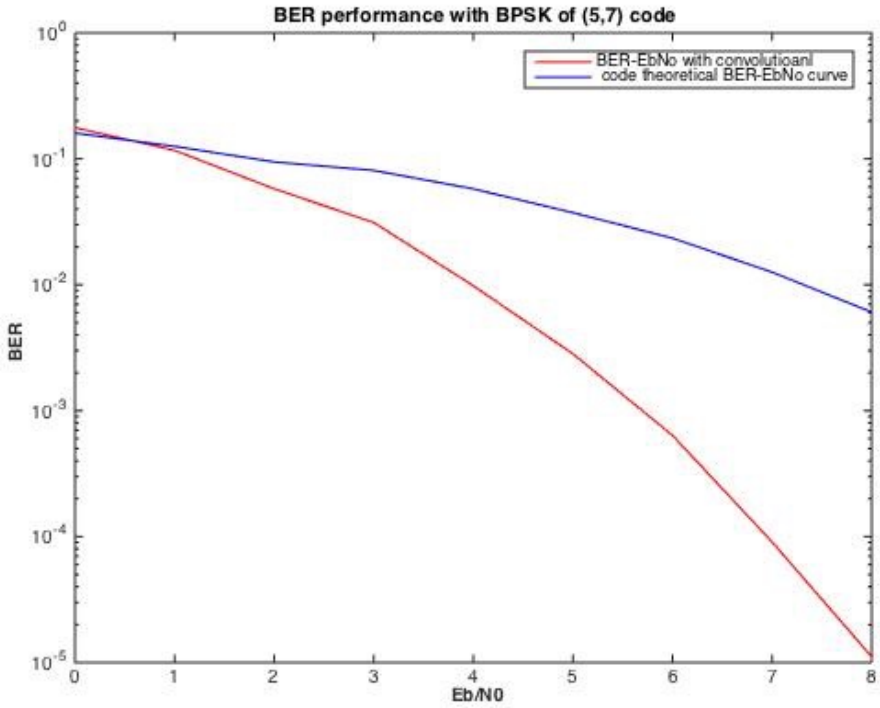
b.For (7,5) code, $d_{free} = 5$,the nominal coding gain $10\log_{10} d_{free}R = 10 * \log_{10}(5 * 1/2) = 3.979$ from the curve,we can roughly calculate the coding gain for different BER.

BER	Coding Gain
10^-1	0.1
10^-2	2.7
10^-2.5	3.4

For (17,15) code, $d_{free} = 7$,the corresponding nominal coding gain:
$10\log_{10} d_{free}R = 10 * \log_{10} 7 * 1/2 = 5.441$,
from the curve,we can roughly calculate the coding gain for different BER.

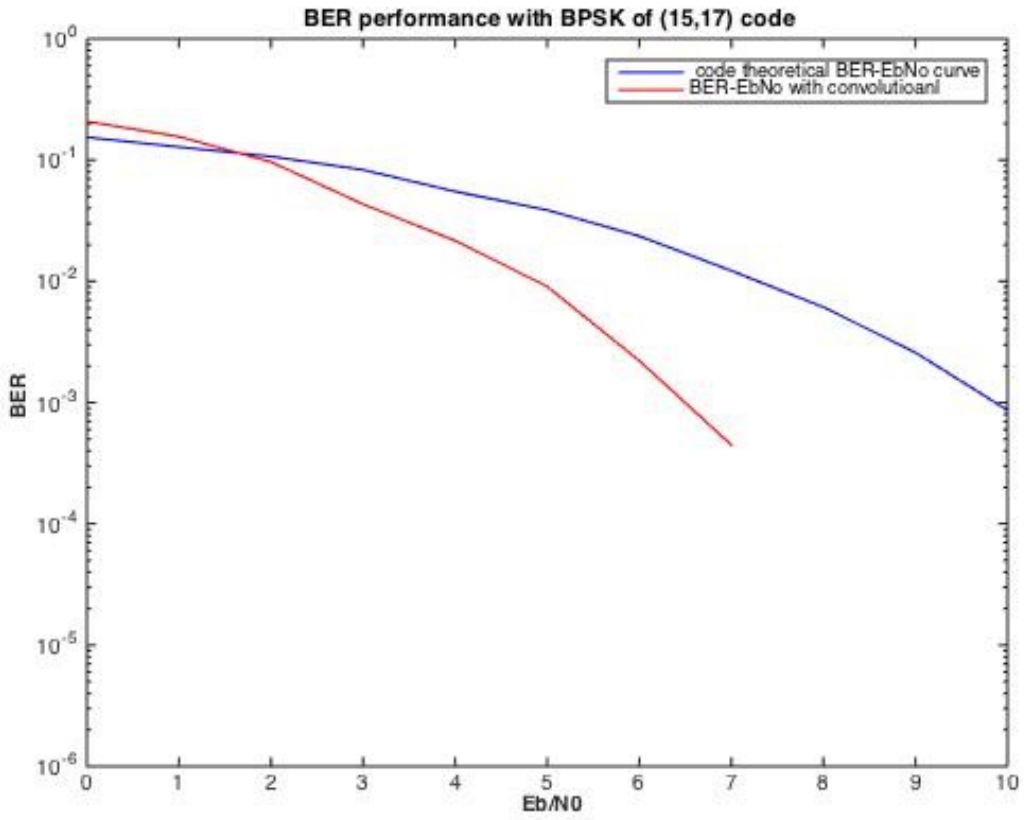
BER	Coding Gain
10^-1	0
10^-2	2.2
10^-3	2.8

As BER increases,the coding gain also increases,especially for (7,5) code,up to BER=10^-2.5,the coding gain is very close to the nominal gain.
Then,I tried to implement the decoding process by myself.And the result is shown below.
1.Result of (7,5)code



BER	Coding Gain
10^-1	0.2
10^-2	3.2
10^-2.5	3.8

2.Result of (15,17)code



BER	Coding Gain
10^-1	0
10^-2	2.2
10^-3	2.7

This implementation also prove that convolutional code has better performance than uncoded one,the coding gain is increasing with better signal-to-noise ratio.And compared with the two different codes,the (15,17) has less coding gain increase with the same BER.