Adv. Radio Sci., 5, 209–214, 2007
www.adv-radio-sci.net/5/209/2007/

Advances in
Radio Science

Adaptive decoding of convolutional codes

K. Hueske, J. Geldmacher, and J. Götze

Information Processing Lab - AG DT, University of Dortmund, Germany

Abstract. Convolutional codes, which are frequently used as error correction codes in digital transmission systems, are generally decoded using the Viterbi Decoder. On the one hand the Viterbi Decoder is an optimum maximum likelihood decoder, i.e. the most probable transmitted code sequence is obtained. On the other hand the mathematical complexity of the algorithm only depends on the used code, not on the number of transmission errors. To reduce the complexity of the decoding process for good transmission conditions, an alternative syndrome based decoder is presented. The reduction of complexity is realized by two different approaches, the syndrome zero sequence deactivation and the path metric equalization. The two approaches enable an easy adaptation of the decoding complexity for different transmission conditions, which results in a trade-off between decoding complexity and error correction performance.

1 Introduction

The Viterbi Decoder (VD) (Viterbi, 1967) is the standard approach for decoding convolutional codes. The decoder is based on the application of the Viterbi Algorithm (VA) to the trellis representation of the convolutional encoder. Forney Jr. (1973) furthermore showed that the Viterbi Decoder is an optimum Maximum Likelihood (ML) decoder, i.e. the valid code sequence with minimum distance to the received sequence is obtained.

The mathematical complexity only depends on the used code, i.e. it is not depending on the channel behaviour, which will be described as Signal to Noise Ratio (SNR) in this paper. This means, a constant high amount of decoding operations is required, even if few or no errors occurred. This is a disadvantage, especially for applications that require energy efficient implementations (e.g. mobile terminals).

This paper presents an alternative syndrome based convolutional decoder, whose complexity can be adaptively reduced in case of good transmission conditions. The decoder also uses the VA, but the algorithm is applied to the trellis representation of the syndrome former. While the VD determines the most likely transmitted code sequence directly, the syndrome based decoder determines the most probable error sequence first. It will be shown that both methods allow optimum ML decoding, but using the syndrome former trellis is advantegous in terms of adaptivity and complexity.

Syndrome based convolutional decoders were also described by Schalkwijk and Vinck (1975); Ariel and Snyders (1999); Reed and Truong (1985), but the presented decoder allows further adaptivity in terms of a trade-off between computational complexity and error correction performance. The reduction in complexity is realized using two approaches:
The syndrome zero sequence deactivation considers the dependencies between the syndrome and the error sequence, which allows the deactivation of the decoder for error free sequences. This leads to a reduction of decoding complexity for high SNR with no or marginal loss in decoding performance.

The path metric equalization reduces the required number of Add-Compare-Select (ACS) operations for the VA by using identical path metrics for different error patterns. The error correction performance is decreased but the decoding complexity can be reduced by 25%. This is especially useful for applications that require a BER threshold and do not benefit from further BER degradation. While the performance of the presented decoder without adaptation is equivalent to the standard VD, we can reduce the complexity of the decoding process if the BER is below the required threshold.

Section 2 presents the basic syndrome decoding algorithm, which is based on Schalkwijk and Vinck (1975). Furthermore the equivalence to the VD will be derived, which shows

Correspondence to: K. Hueske
(klaus.hueske@uni-dortmund.de)

that both are optimum ML decoders. In Sect. 3 two methods for complexity reduction are introduced, syndrome zero sequence deactivation and path metric equalization. Furthermore an estimation technique for the Bit Error Rate (BER) is introduced, which is used for an adaptive complexity reduction. For performance and complexity analysis simulation results are presented in Sect. 4. Conclusions are given in Sect. 5.

2 Syndrome decoding

In this section the syndrome decoding algorithm proposed by Schalkwijk and Vinck (1975) is presented.

2.1 Syndrome decoding

The syndrome decoding algorithm is presented for code rates $R{=}k/n$. Sequences and transfer functions are represented in frequency domain as power series in D with coefficients from $GF(2)$. For the sake of clarity, the delay operator D is omitted in the following.

An information sequence is encoded by multiplication with a generator matrix

$$\mathbf{v} = \mathbf{u}\mathbf{G}, \tag{1}$$

where $\mathbf{G}$ is a polynomial generator matrix with k rows and n columns, $\mathbf{v}$ is a row vector of length n containing the coded bits and $\mathbf{u}$ is a row vector of length k containing the information bits.

The sequence $\mathbf{v}$ is transmitted over a memoryless, noisy channel, causing the corrupted received sequence

$$\mathbf{r} = \mathbf{v} + \mathbf{e}, \tag{2}$$

where $\mathbf{e}$ is the error sequence resulting from channel noise.

A syndrome sequence $\mathbf{b}$ is computed from the received sequence as

$$\mathbf{b} = \mathbf{r}\mathbf{H}^T, \tag{3}$$

where $\mathbf{H}^T$ is a polynomial matrix with n rows and $n-k$ columns and is called the syndrome former matrix of $\mathbf{G}$. The syndrome former matrix is defined to be orthogonal to $\mathbf{G}$, thus the syndrome sequence $\mathbf{b}$ is the zero sequence if and only if $\mathbf{r}$ is a valid code sequence. It follows that the syndrome sequence only depends on the error sequence and is independent of the transmitted information:

$$\mathbf{b} = \mathbf{r}\mathbf{H}^T = (\mathbf{v}+\mathbf{e})\mathbf{H}^T = \mathbf{e}\mathbf{H}^T \tag{4}$$

The syndrome decoder has to map the syndrome sequence back to the error sequence. However the mapping from $\mathbf{b}$ to $\mathbf{e}$ is not unique, but there is a set of error sequences corresponding to one syndrome sequence. For ML decoding the decoder has to determine the code sequence with minimum distance to the received sequence, which corresponds to the error sequence with minimum weight. Thus the decoding problem can be formulated in terms of an optimization problem with constraint:

$$\min_{\hat{\mathbf{e}}} ||\hat{\mathbf{e}}|| \quad \text{with} \quad \mathbf{b} = \hat{\mathbf{e}}\mathbf{H}^T \tag{5}$$

The norm $||...||$ is defined as hamming distance for hard decision decoding and as 2-norm for soft decision decoding. The minimization can be done by searching the syndrome former trellis for the error sequence with minimum weight using the VA. The constraint is incorporated by only allowing the transitions belonging to the current syndrome value at every stage in the syndrome former trellis.

When the ML error sequence $\hat{\mathbf{e}}$ has been found, the transmission errors are corrected by subtracting the estimated error sequence from the received sequence

$$\hat{\mathbf{v}} = \mathbf{r} - \hat{\mathbf{e}}, \tag{6}$$

where $\hat{\mathbf{v}}$ is the estimated code sequence. Finally the information sequence can be obtained as the product of the estimated code sequence and the right inverse generator matrix

$$\hat{\mathbf{u}} = \hat{\mathbf{v}}\mathbf{G}^{-1}. \tag{7}$$

2.2 Syndrome former matrix and trellis representation

The syndrome former $\mathbf{H}^T$ can be calculated from the invariant factor decomposition (IFD) of the generator matrix $\mathbf{G}$. The IFD of a matrix $\mathbf{G}$ is defined as

$$\mathbf{G} = \mathbf{A}\Gamma\mathbf{B} \tag{8}$$

where $\mathbf{A}$ is a polynomial $(k{\times}k)$ matrix and $\mathbf{B}$ is a polynomial $(n{\times}n)$ matrix with det($\mathbf{A}$)=det($\mathbf{B}$)=1. Γ is a $(k{\times}n)$ matrix and is called Smith-Form of $\mathbf{G}$. An algorithm for the computation and the properties of the IFD are given in Johannesson and Zigangirov (1999).

The first k rows of $\mathbf{B}$ form an equivalent generator matrix of $\mathbf{G}$ and the last $n{-}k$ rows are the transpose of the inverse of $\mathbf{H}$:

$$\mathbf{B} = \begin{pmatrix} \mathbf{G}_b \\ (\mathbf{H}^{-1})^T \end{pmatrix} \tag{9}$$

As $\mathbf{B}$ is non-singular, it can be inverted. The right inverse equivalent generator matrix $\mathbf{G}_b^{-1}$ and the syndrome former $\mathbf{H}^T$ can be identified as the first k columns and the last $n-k$ columns of $\mathbf{B}^{-1}$, respectively:

$$\mathbf{B}^{-1} = \begin{pmatrix} \mathbf{G}_b^{-1} & \mathbf{H}^T \end{pmatrix} \tag{10}$$

For $R{=}k/n$ the syndrome former takes a sequence of n polynomials as input and produces a syndrome sequence represented by $n{-}k$ polynomials. As the number of memory elements required to realize the syndrome former is equal to the number of the memory elements of the encoder (Forney Jr., 1970), the syndrome former trellis has the same number of states as the encoder trellis. The syndrome trellis has

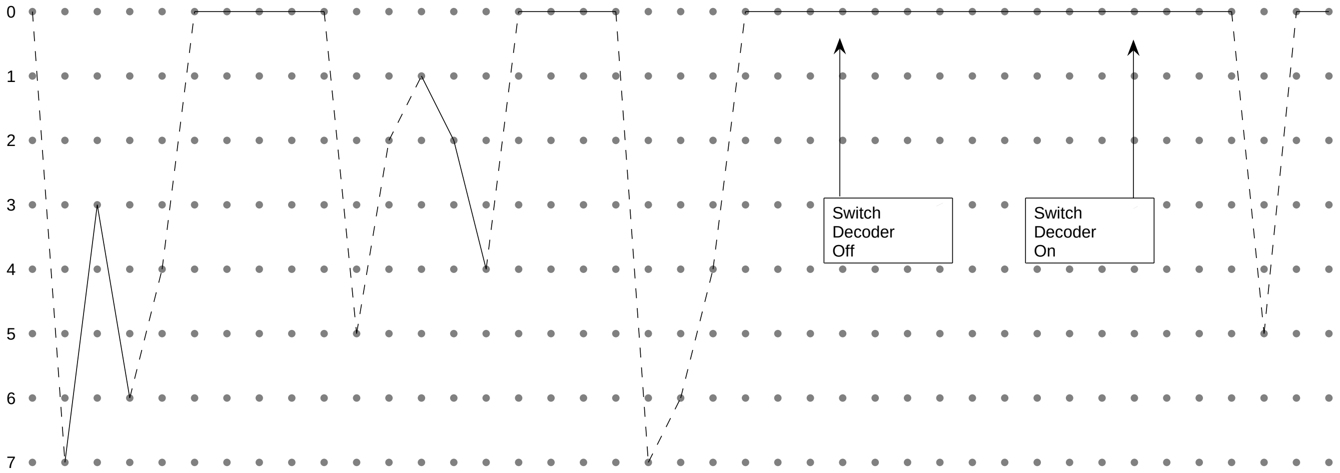

Fig. 1. Example of an error path, “0“-syndrome sequences are plotted as lines, ”1“-syndrome sequences are plotted dashed.

2^v states, where v is the number of memory elements of the encoder, and 2^n edges leaving each state.

For $R=1/2$ the syndrome trellis has 4 edges leaving each state. The trellis can be split into two parts, with each part containing only the transitions of one syndrome symbol. This results in two trellises with each 2^v states and 2 edges leaving each state.

To find the minimum weight error sequence, a shortest path algorithm like the VA is used. As branch metric for the transition of a state to a successor state, the weight of the corresponding error pattern is taken. To incorporate the constraint from Eq. (5), at each decoding stage only the trellis part corresponding to the current syndrome symbol is considered.

2.3 Equivalence to the Viterbi Decoder

The equivalence of the syndrome based decoder to the VD can be shown by replacing the estimated error $\hat{\mathbf{e}}$ in Eq. (5) by $\mathbf{r}-\hat{\mathbf{v}}$. Expression (5) then becomes

$$\min_{\hat{\mathbf{v}}} ||\mathbf{r}-\hat{\mathbf{v}}|| \text{ with } \mathbf{b}=(\hat{\mathbf{v}}+\hat{\mathbf{e}})\mathbf{H}^T, \tag{11}$$

and using $\mathbf{b}=\hat{\mathbf{e}}\mathbf{H}^T$ leads to

$$\min_{\hat{\mathbf{v}}} ||\mathbf{r}-\hat{\mathbf{v}}|| \text{ with } \mathbf{0}=\hat{\mathbf{v}}\mathbf{H}^T, \tag{12}$$

which represents the decoding problem in “Viterbi sense“. As both decoders are equivalent, the syndrome based decoder is also an optimum ML decoder.

We note that the syndrome computation in Eq. (3) and the mapping from estimated code sequence to information sequence in Eq. (7) can be realized as simple XOR operations. Thus the estimation of the error sequence has a similar complexity as the VD.

3 Adaptive decoding

3.1 Syndrome zero sequence deactivation

Good transmission conditions will lead to error free sequences in $\mathbf{r}$, i.e. to sequences where $\mathbf{e}=\mathbf{0}$. As the syndrome sequence only depends on the channel errors, as shown in Eq. (4), error free periods will lead to zero sequences in the syndrome $\mathbf{b}$, too. For zero sequences, no decoding is necessary, the zero path in the syndrome former trellis is taken.

The first proposed optimization of the syndrome decoder now consists of detecting zero sequences in the syndrome, and switching the decoder off for these sequences. A reasonable length of the period has to be assumed, to be sure, that the ML path in the trellis has returned to zero state. Decoding is done by always tracing back from zero state.

If the minimum zero period length is chosen long enough, the decoding performance can be as good as the performance of the unmodified decoder. There is a trade-off, decoding performance can be exchanged with decoding complexity by reducing or increasing the required length of the zero sequences.

There are two critical parameters for the adaptive decoder: The first parameter l_{off} is the number of syndrome zeros, after which the decoder can be safely switched off. The second parameter l_{on} is the number of stages before the first syndrome one, when the decoder has to be switched on again. The second parameter depends on the number of path registers v. Because an error event is propagated with maximum delay of v stages to the syndrome, the decoder has to be switched on v stages before the first syndrome one. The first parameter also depends on v but is also influenced by the received data in case of soft decision, which makes a general choice infeasible. Therefore, simulations are used to determine a reasonable value.

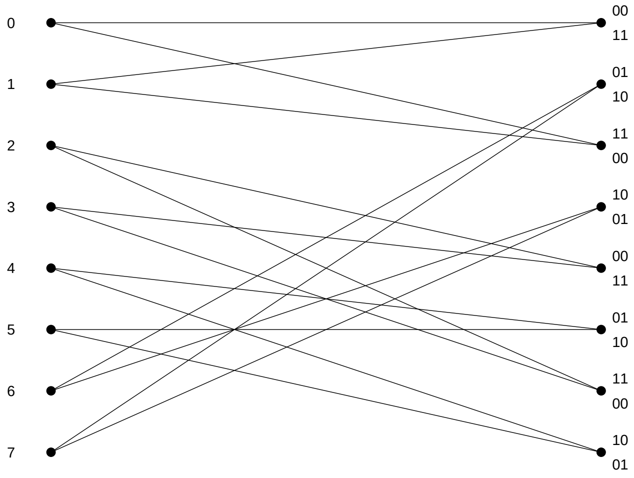

Fig. 2. Syndrome former trellis for the example code, Syndrome 0 part.

As an example, Fig. 1 shows an error path for a v=3 Code. In this example the decoder is switched off, when six successive syndrome zeros are detected. The decoder is turned off after l_{off}=3 syndrome zeros and turned on again l_{on}=3 stages before the next syndrome one.

3.2 Path metric equalization

A second reduction of the computational complexity is based on a closer examination of the syndrome former trellis structure. As Schalkwijk et al pointed out in Schalkwijk and Vinck (1976), the syndrome former trellis has a special structure. We can identify pairs of states which have the same predecessor states and at the same time have identical weights of the corresponding transition in case of hard decision.

For example, Fig. 2 shows a syndrome former trellis for a code with memory length v=3 and generator polynomials $\mathbf{G}$=[D^3+D^2+1, D^3+D^2+D+1]. For simplicity only the zero part is considered. The trellis is annotated on the left side with the state numbers and on the right side with the error input corresponding to the transition. One can see that the states 1 and 3 have the same predecessor states (6 and 7) with the transitions having the same weight 0+1=1 and 1+0=1. The same holds for the states 5 and 7 which have the predecessors 4 and 5.

Let $M^{(i)}(t)$ be the metric of state i at time t and $\mu^{(i,j)}(t)$ the branch metric for a transition from state i to state j. At every decoding stage and for every state the VA has to compute the metrics for all transitions from each predecessor state, compare the metrics and choose the survivor path as the path with minimum metric (ACS). For example, for the states 1 and 3 the metric is computed as

$$\begin{aligned}&M^{(1)}(t+1)=\\&\min\{M^{(6)}(t)\quad+\mu^{(6,1)}(t),\,M^{(7)}(t)+\mu^{(7,1)}(t)\}\\&M^{(3)}(t+1)=\\&\min\{M^{(6)}(t)\quad+\mu^{(6,3)}(t),\,M^{(7)}(t)+\mu^{(7,3)}(t)\}\end{aligned}\qquad(13)$$

If harddecision is applied, the branch metrics are computed as

$$\begin{aligned}&\mu^{(6,1)}(t)=\mu^{(7,3)}(t)=0+1=1\\&\mu^{(7,1)}(t)=\mu^{(6,3)}(t)=1+0=1\end{aligned}\qquad(14)$$

and Eq. (13) becomes

$$\begin{aligned}&M^{(1)}(t+1)=\min\{M^{(6)}(t)+1,\,M^{(7)}(t)+1\}\\&M^{(3)}(t+1)=\min\{M^{(6)}(t)+1,\,M^{(7)}(t)+1\}\end{aligned}\qquad(15)$$

The metric computation is identical for both state 1 and state 3 and thus for state 3 no computation is required. The same holds for states 6 and 7.

We can use this for reducing the complexity of the decoder and save one ACS operation for each of the 2^{v-2} pairs. If the decoder uses harddecision, 25% of the ACS operations can be saved without loss of decoding performance (Schalkwijk and Vinck, 1976).

However as hard decision is rarely used, we now discuss the metric computation for the soft decision case. Let $\mathbf{r}(t)$=[$r_1(t)$ $r_2(t)$] be the 3-Bit quantized vector received at time t, zero symmetric BPSK modulation and AWGN channel assumed. The absolute value of $\mathbf{r}(t)$ can be seen as a measure for the confidence of the received word. The lower the absolute value of $r_i(t)$, the more likely a transmission error occurred at this position. Thus the branch metric calculation for softdecision can formulated as

$$<\mathbf{e}(t),\mathbf{r}(t)>=[e_1(t)\ e_2(t)][|r_1(t)|\ |r_2(t)|]^T\qquad(16)$$

to incorporate the confidence informations. The metric calculation for the example becomes

$$\begin{aligned}&M^{(1)}(t+1)=\min\{M^{(6)}(t)+|r_2(t)|,\,M^{(7)}(t)+|r_1(t)|\}\\&M^{(3)}(t+1)=\min\{M^{(6)}(t)+|r_1(t)|,\,M^{(7)}(t)+|r_2(t)|\}\end{aligned}\qquad(17)$$

So the metric calculation is not identical for soft decision and a simplification is no longer possible without decoding performance loss.

However, the metric calculation can be modified to allow a simplification analog to the hard decision case. This is realized by assigning the same branch metrics to the 01 and 10 error patterns, which can be done by choosing the minimum of $|r_1(t)|$ and $|r_2(t)|$ as branch metric for the 01 and 10 transitions:

$$\mu^{(i,j)}(t)=\begin{array}{ll}0 & \text{for } \mathbf{e}(t)=[00]\\ \min\{|r_1(t)|,|r_2(t)|\} & \text{for } \mathbf{e}(t)=[10]\\ \min\{|r_1(t)|,|r_2(t)|\} & \text{for } \mathbf{e}(t)=[01]\\ |r_1(t)|+|r_2(t)| & \text{for } \mathbf{e}(t)=[11]\end{array}$$

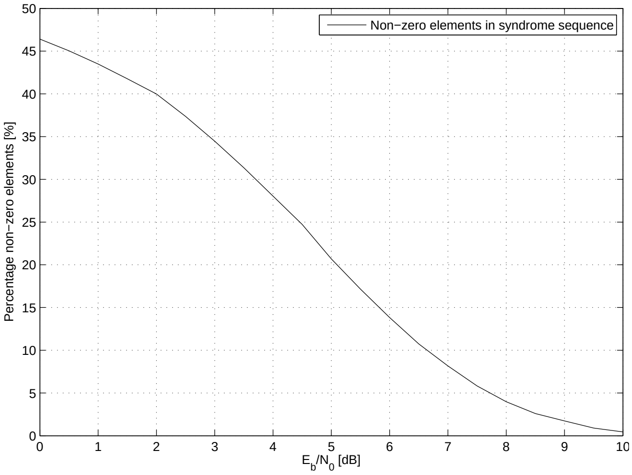

Fig. 3. Percentage of non-zero elements in syndrome vector.

The metric computation for the states 3 and 6 then becomes

$$\begin{aligned}M^{(1)}(t+1) &= \min\{M^{(6)}(t)+\min\{|r_1(t)|,|r_2(t)|\},\\ &\qquad M^{(7)}(t)+\min\{|r_1(t)|,|r_2(t)|\}\}\\ M^{(3)}(t+1) &= \min\{M^{(6)}(t)+\min\{|r_1(t)|,|r_2(t)|\},\\ &\qquad M^{(7)}(t)+\min\{|r_1(t)|,|r_2(t)|\}\}\end{aligned} \tag{18}$$

If the metric calculation is modified in this way, the reduction of complexity can be applied as in the hard decision case. However, there will be a loss of decoding performance compared to the full complexity soft decision decoder.

3.3 BER estimation

The path metric equalization is especially suitable for transmission systems that require a specific BER and do not benefit from better channel conditions. If the desired BER is reached, the complexity of the decoder can be reduced adaptively. The total error correction performance will be decreased, but the required BER threshold will always be achieved.

The problem that comes up with this approach is that the BER is generally unknown in the receiver. But using the syndrome vector we can roughly estimate the BER by calculating the syndrome vector weight. This means for good transmission conditions only few errors occur, which results in a syndrome vector with only few non-zero elements. On the other hand if the SNR is very low, we expect many transmission errors, which produces many non-zero elements. The dependencies between SNR and number of non-zero elements in the syndrome vector is depicted in Fig. 3.

Using this relation we can define a syndrome vector weight threshold in the receiver, which allows an adaptive switching between the normal and the reduced complexity syndrome based decoder.

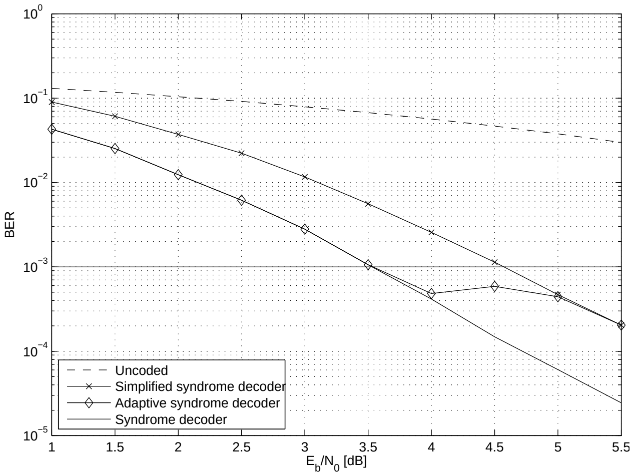

Fig. 4. Comparison of the Bit-Error-Rates of the reduced complexity decoder and the full complexity decoder.

3.4 Summary

There are two options for reduction of decoding complexity. The first option is an adaptive decoding approach, which is based on analyzing the syndrome sequence and turning off the decoder in error-free periods. A reasonable minimum length assumed, decoding costs can be reduced without loosing decoding performance. The second option is based on the syndrome former trellis structure and allows a reduction of decoding complexity by about 25% but resulting in a loss of decoding performance.

4 Simulation results

This section gives simulation results for both options. All simulations have been done for a code with generator matrix

$$\mathbf{G} = [D^3 + D^2 + 1,\ D^3 + D^2 + D + 1] \tag{19}$$

with v=3 path registers and corresponding syndrome former

$$\mathbf{H}^T = [D^3 + D^2 + D + 1,\ D^3 + D^2 + 1]^T \tag{20}$$

The codewords are transmitted over a memoryless AWGN channel using BPSK modulation. For the BER simulations soft decision was applied, considering 2000 simulated errors.

Figure 4 shows the results for the reduced complexity syndrome decoder using path metric equalization. The performance of the simplified syndrome decoder is about 1dB worse compared to the regular syndrome decoder, but it requires 25% less operations. The same figure shows the adaptive decoding approach with a desired BER of 10^{-3}. For a $E_b/N_0 < 3.5$ dB only the unmodified syndrome decoder is used, i.e. the decoding complexity is equivalent to the VD. For higher SNR the reduced complexity decoder is used, depending on the syndrome vector weight of the actual received

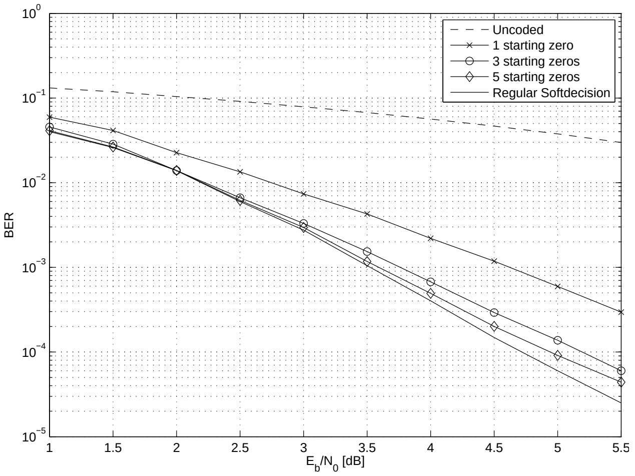

Fig. 5. Comparison of the Bit-Error-Rates of the adaptive decoders using 3 different minimum zero sequence lengths and the regular decoder for Softdecision.

data block. For E_b/N_0>5.0 dB all decoding operations are done by the reduced complexity decoder. This shows, that the decoding complexity can be easily adapted for different channel conditions.

Figure 5 shows the performance of the decoder using syndrome zero sequence deactivation. Simulations are shown for l_{on}=3 and $l_{off} \in \{1, 3, 5\}$. In Fig. 6 the percentage of the decoding time the decoder is turned off is plotted. For example at E_b/N_0=4.5 dB the decoder with l_{off}=5 is turned off about 33% of the time with about 0.15 dB loss in performance compared to the regular decoder. This allows a reduction of complexity with only marginal performance loss. If desired, the complexity can be further decreased by reducing l_{off}. At E_b/N_0=4.5 dB the decoder with l_{off}=1 is turned off about 48% of the time while taking a loss of about 1 dB.

5 Conclusions

While performance and computational complexity of the Syndrome Decoder are equivalent to the Viterbi Decoder for worst case transmission conditions, the syndrome based decoder allows an adaptive reduction of complexity for good channel conditions. With the proposed methods the adaptive reduction can be performed in two steps. First, the syndrome zero sequence deactivation will reduce the decoding complexity without significant loss in decoding performance. If desired, the required number of operations can be further decreased by either using a reduced length l_{off} or using the path metric equalization. The last step is especially applicable for transmission systems that require a BER threshold and do not benefit from further SNR improvement.

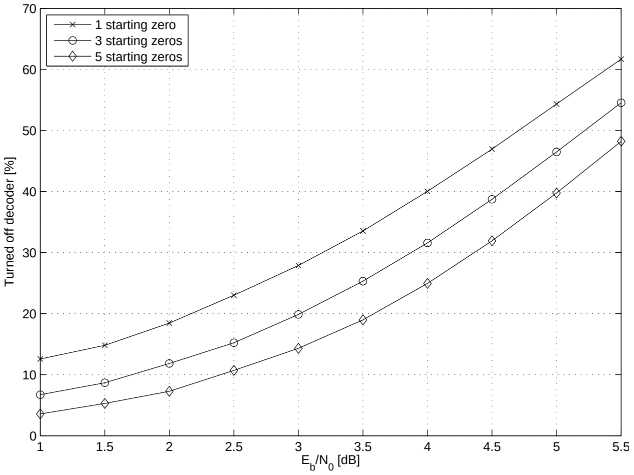

Fig. 6. The percentage of turned off decoders for th 3 different minimum zero sequence lengths.

References

Ariel, M. and Snyders, J.: Error-trellises for convolutional codes .II. Decoding methods, Communications, IEEE Transactions on, 47, 1015–1024, doi:10.1109/26.774852, 1999.

Forney Jr., G.: Convolutional codes I: Algebraic structure, Information Theory, IEEE Transactions on, 16, 720–738, 1970.

Forney Jr., G.: The Viterbi Algorithm, IEEE Proceedings, 61, 268–278, 1973.

Johannesson, R. and Zigangirov, K. S.: Fundamentals of Convolutional Coding, IEEE press, 1999.

Reed, I. and Truong, T.: Error-trellis syndrome decoding techniques for convolutional codes, IEE Proceedings Pt. F, 132, 77–83, 1985.

Schalkwijk, J. and Vinck, A.: Syndrome Decoding of Convolutional Codes, Communications, IEEE Transactions on, 23, 789–792, 1975.

Schalkwijk, J. and Vinck, A.: Syndrome Decoding of Binary Rate-1/2 Convolutional Codes, Communications, IEEE Transactions on, 24, 977–985, 1976.

Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, Information Theory, IEEE Transactions on, 13, 260–269, 1967.