Kasetsart J. (Nat. Sci.) 41 : 364 - 370 (2007)

Performance Investigation of Convolutional Vector Symbol Decoding with Larger than Two Choices and with Incomplete Second Choices

Usana Tuntoolavest* and Auttaphud Seubnaung

ABSTRACT

Vector Symbol Decoding (VSD) is a decoding technique for both block and convolutional nonbinary encoders. Convolutional VSD uses a convolutional encoder with nonbinary symbols at the encoder and VSD at the decoder. VSD can employ diversity quite easily to improve the performance and simplify the decoding because diversity provides alternative choices for the decoder. Previous work always assumed two complete choices for simplicity. This paper explores the performance of a convolutional VSD for the case of up to four complete choices and the case of two incomplete choices. The results showed that 1) although the second choice clearly improved the performance, the third and fourth choices do not provide ample improvement to justify the higher complexity of the hardware. 2) VSD can use incomplete second choices to improve performance. In summary, second choices should be applied whenever possible, but no more than two choices should be used. These results are very useful for hardware implementation.

Key words: vector symbol decoding, error correcting code, nonbinary, reliable communications

INTRODUCTION

In wireless communications, the quality of the channel is sometimes low under the condition that is changing all the time. To increase the reliability of the received information, there are two main approaches. The first one is to use simple diversity such as space diversity by using multiple antennas. The second one is to use channel coding, which is actually another form of diversity that is more complex. Usually if the receiver uses simple diversity, it will make decision based on some simple rule such as selecting the choice that has highest signal to noise ratio and throws away the rest of the choices. It can also combine the received signals using maximum ratio combining (MRC) proposed by Breanan (1959) to get the highest signal to noise ratio, with a compromise: MRC assumes perfect knowledge of the channel attenuation and phase shifts. After that, it may input this result to the channel decoder if there is one. Some examples of macro and micro diversity systems are in Chung *et al*. (2000) and Jakes *et al*. (1974). This paper is different from this usual application of diversity and channel coding in that the receiver will input all choices above some threshold to the channel decoder, which is convolutional VSD. Then VSD will use this information to help decode the received information. This is suitable because VSD concept

Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10900, Thailand.
* Corresponding author, e-mail: fengunt@ku.ac.th

allows for the use of list decoding where each received symbol has each own list and may have different number of alternative symbols in the list as shown by Metzner (2003).

The concept of VSD for block code was proposed by Metzner and Kapturowski (1990). Later, Haslach and Han Vinck (1999), rediscovered it where it was called an array code. Seo (1991) presented some concept for convolutional VSD. Metzner (2003) added the use of symbol list with block VSD. Previous simulations in Tuntooolavest (2004) on convolutional VSD with lists were for two complete choices only. That is, all received symbols always had two complete alternative choices and these choices were always different. This was valid because it was assumed that the nonbinary symbols of VSD resulted from the use of a concatenated code where the inner decoder was the list Viterbi decoder (LVA) in Chen and Sundberg (2001) and the outer decoder is VSD.

This paper extends the work in two ways 1) to show whether more than two choices should be used since the hardware would be more complex especially in terms of memory sizes 2) to show whether VSD should make use of the second choices that are incomplete. The results will be used in implementation of VSD lab prototype under development by Intharasakul and Tuntoolavest (2006).

BACKGROUND

Alternative choices

To understand the idea of alternative choices in this paper, assume that the receiver uses 4 receiving antennas. Then, the receiver makes the preliminary decision of valid alternative choices by comparing the SNR of the received waveform for each received symbol at each antenna with a threshold. Up to 4 alternative choices for each received symbol that met threshold are considered valid alternative choices and will be forwarded to VSD. The one with the highest SNR is called "first choice" or "main choice". The ones with lower SNR are called "second choice", "third choice" and "fourth choice" respectively. It is important to note that from this setup, some symbols may have only the "first choice"; other symbols may have two, three or four choices. When all received symbols have all choices, the received sequence is referred to as having complete alternative choices. If some received symbols do not have some choices, the received sequence is referred to as having incomplete alternative choices.

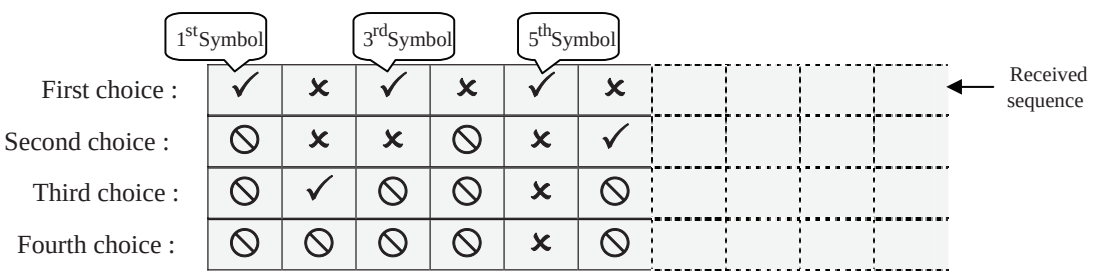

Figure 1 Example of a received sequence with incomplete alternative choices [adapted from Figure 2.1 in Nukmind and Suebnaung (2003)]

when ✓ represents correct received symbol

× represents erroneous received symbol

⊘ represents blank or no available choice

VSD: Correct with choices

Vector symbol decoding (VSD) can correct with or without the help of alternative choices. If the choices are available, VSD will screen and attempt to correct as many error symbols as possible before performing the correction by verification process, which sometimes called "correct with null combination". The theory on correct with alternative choices and correct with null combinations can be found in Metzner (2003). This paper will show an example of the case when the choices are incomplete only.

Basically, VSD uses large nonbinary symbols typically of 32-bit size. The syndrome and the received sequence are considered in the matrix format

$$\mathbf{S} = \mathbf{HY}, \quad (1)$$

where **S** is the syndrome matrix, **H** is the parity check matrix, and **Y** is the received matrix.

Each row of the received matrix represents a 32-bit symbol and the number of columns represents the number of received symbols used in the calculation of the syndrome matrix at that time. The number of column varies because convolutional VSD computes the syndrome vector (or row of the syndrome matrix) one at a time. If it cannot correct with one syndrome row (where the syndrome matrix contains one row), it will increase the number of syndromes to two (where the syndrome matrix contains two rows) and so on.

To understand why VSD can handle various list size for each symbol with little added complexity other than the increase in storage space, consider that VSD corrects with choices by appending the differences between the first choice and each of the alternative choices at the bottom of the syndrome matrix. Any added row that is in the row space of the original syndrome matrix is found as the correct symbol and will be used to replace the error symbols in the first choice. The index of the error symbols to be replaced is found by keeping track of which symbol that the appended row comes from.

As an example, consider a case of one syndrome where each syndrome is computed by 3 received symbols. Figure 2 shows the received symbols available for the current computation. The position of the correct and the wrong symbols are the same as in Figure 1.

The modified syndrome matrix in this case is

$$\mathbf{S}_{\mathbf{mod}} = \begin{bmatrix} syndrome1 \\ \text{A2} - \text{B2} \\ \text{A2} - \text{C2} \\ \text{A3} - \text{B3} \end{bmatrix} \quad (2)$$

Since A2 is wrong and C2 is correct, the difference is

A2-C2 = (error value $\underline{e}$ + correct symbol) – (correct symbol) = error value $\underline{e}$ (3)

Also, the "*syndrome*1" is computed from A1, A2 and A3 and must be zero if all three are correct. Since only A2 is wrong, the "*syndrome*1" also equals the error value $\underline{e}$. By finding the differences in the appended row that have the same value as "*syndrome*1", the decoder finds the correct symbol and by keeping track of which differences that row came from, the decoder decides that C2 is correct and can correct the error in A2 by replacing A2 by C2. Note that examples for more syndromes can

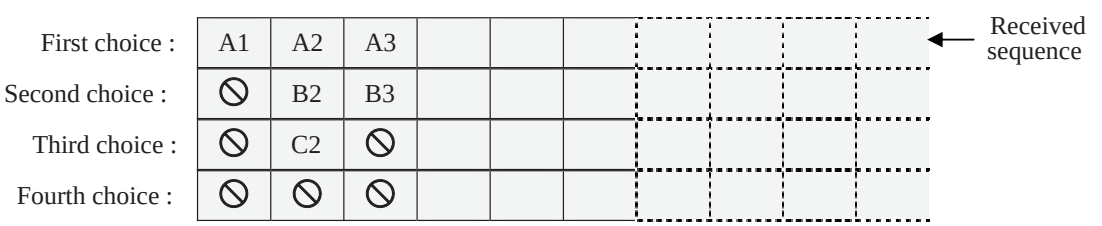

Figure 2 Shows the received symbols available for one syndrome.

be found in Nukmind and Suebnaung (2003).

METHOD

In the simulations, it was assumed that the encoder uses a (3,2,2) nonbinary convolutional code with 32-bit symbols and

$$G(D) = \begin{bmatrix} 1+D+D^2 & D^2 & 1 \\ D & 1+D^2 & 1+D+D^2 \end{bmatrix}.$$

The codeword is terminated at 21 encoded symbols. The vector symbol decoder uses a maximum of six syndromes, which is less than those in the previous work by Tuntoolavest (2003) since more than six syndromes are highly impractical in hardware. The simulation was divided into two main parts.

1. *Up to Four complete alternative choices*. In this part, the received sequences always had one, two, three or four complete alternative choices. For example, three complete choices mean there are three different alternative choices for every symbol. To simulate this part, the probabilities of symbols error for each alternative choice were assumed. Two different sets of probability of symbol error were used in the simulations. Case one assumed that $p_2 = 2p_1$, $p_3 = 4p_1$ and $p_4 = 8p_1$. Case two assumed that $p_2 = 2p_1$, $p_3 = 3p_1$ and $p_4 = 4p_1$ where

p_1 = probability of symbol error for the first choice

p_i = probability of symbol error for the i^{th} choice *given that the* $(i\text{-}1)^{th}$ *choice is wrong*; $i = 2, 3, 4$.

Note that p_i were defined in this format because there can be no more than one correct symbol in the list of four alternative choices for each symbol. If the first choice symbol at a particular position is correct, choice two to four at that position must be wrong. However, if the first choice symbol at a particular position is wrong, the second choice has a chance to be correct. Similarly, the fourth choice only has a chance to be correct if all three previous choices are wrong.

2. *Two incomplete alternative choices*. In this part, the received sequences always had the complete first choice but the second choices may be incomplete. The probability of not having second choices is P_n. For example if $P_n = 20\%$, each symbol has an 80% chance of having second choice and 20% chance of not having second choice. In the simulations, four different values of P_n were considered, i.e, $P_n = 0\%$, 25%, 50% and 100%. To obtain more realistic results, the probabilities of symbol error (p_1 and p_2) in this case were chosen from the simulation of the list Viterbi decoder (LVA) with the model from Tuntoolavest and Metzner (2002). This model assumed that concatenated code was used. The inner code was a convolutional code with List Viterbi Decoder (LVA) and the outer code used VSD as the decoder. The channel was a simplified two-state fading channel where the non-fade state provided perfectly demodulated symbols.

RESULTS

Figure 3 and Figure 4 show the symbol error probability after VSD finished the decoding process for the case of complete alternative choices. These two figures used two different sets of probabilities of the third and the fourth choices, but the results are almost identical. Both figures also contain four lines, but the results for using three and four complete choices are overlapping. The "i-choice(s)" in the graph refer to the number of complete choices that VSD was provided.

Figure 5 shows symbol error probability after VSD finished the decoding process for the case of two incomplete alternative choices when the probabilities of not having the second choice (P_n) are 0%, 25%, 50% and 100%. It is clear that smaller P_n value gives better decoding performance than larger P_n as expected since smaller P_n means the decoder has more information.

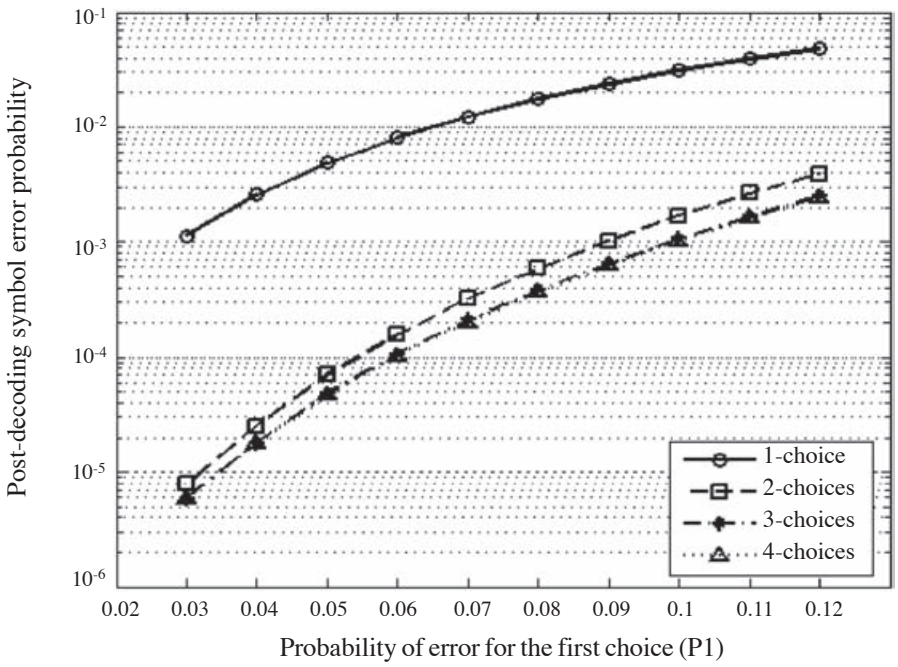

Figure 3 Post-decoding symbol error probability of VSD for the case of up to four complete alternative choices: input symbol error probabilities are $p_2 = 2p_1$, $p_3 = 4p_1$ and $p_4 = 8p_1$.

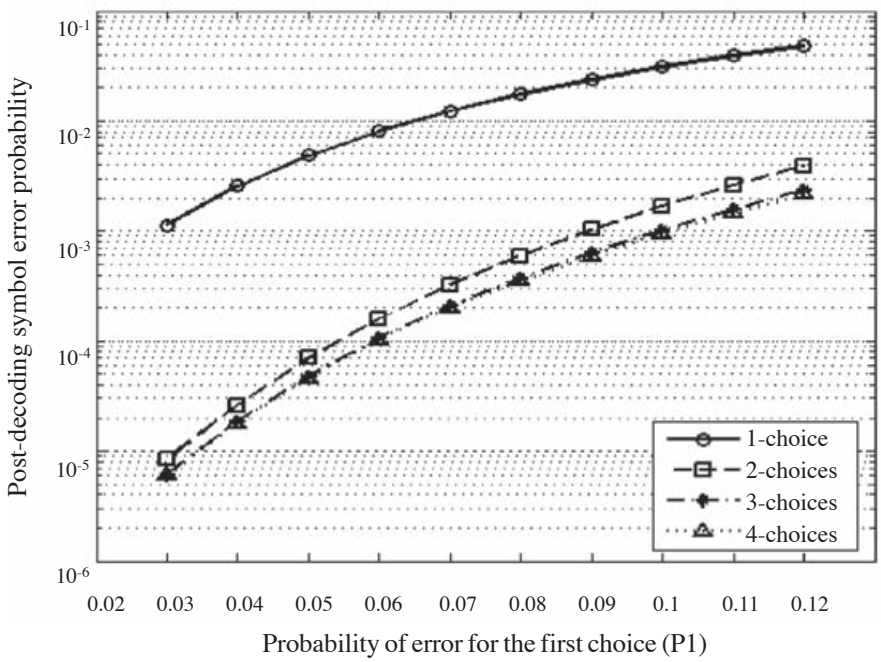

Figure 4 Post-decoding symbol error probability of VSD for the case of up to four complete alternative choices: input symbol error probabilities are $p_2 = 2p_1$, $p_3 = 3p_1$ and $p_4 = 4p_1$.

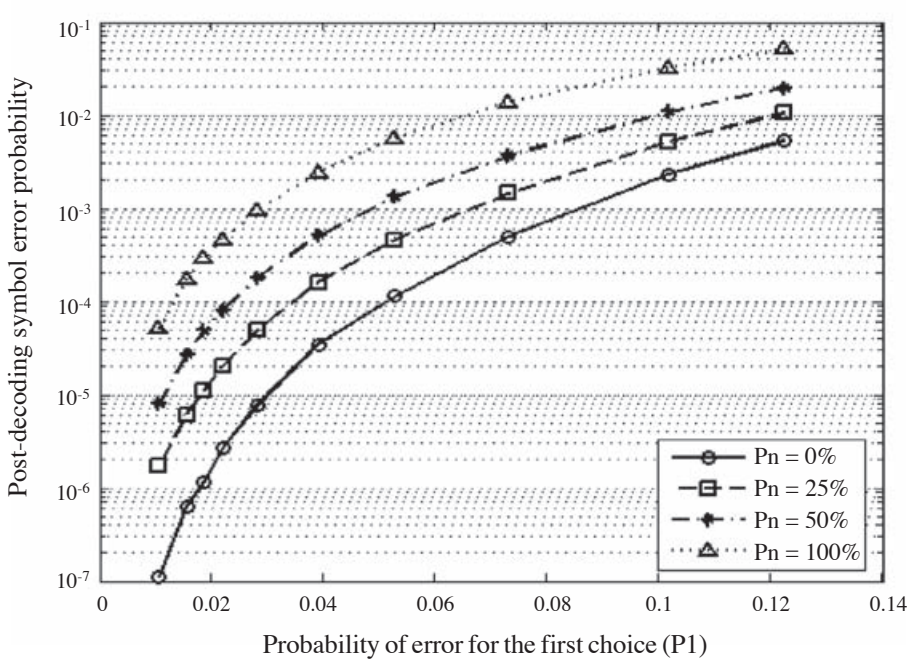

Figure 5 Post-decoding symbol error probability of VSD for the case of incomplete second choices: p_1, p_2 are from LVA simulation, probability of not having second choice $P_n = 0, 25, 50, 100\%$.

DISCUSSIONS

From Figure 3 and Figure 4, the alternative choices can improve performance of VSD but two choices are enough since more than two choices slightly improve the performance. The two sets of input probabilities used were different only in the third and the fourth choices because the result of the second choice has been shown in previous work and was not the objective of the paper. LVA was not used in this part of the simulation because simulation of LVA with a list of 4 is highly complex and Figure 3 and 4 showed that they did not affect the results.

Since the third and fourth choices contributed very little to the performance, the simulation in part two used only the first and second choices to simulate the incomplete alternative choices case. By considering only two incomplete choices instead of four, the complexity and the resource consumption, especially the memory unit, was reduced significantly. In Figure 5, the $P_n = 0\%$ curve means the second choice is complete. When $P_n = 25\%$ and 50%, only 75% and 50% of second choice were available to the decoder respectively. For the $P_n = 100\%$ case, there was no second choice at all. Obviously,the cases of incomplete second choice ($P_n = 25\%$ and 50%) have lower performance than that of complete second choice. However, it is clear that the more the information on the second choice was provided, the higher the improvement on the performance would be. Therefore, this second choice should be used in the VSD decoding process as much as possible.

It should be noted that the case of incomplete alternative choice is more realistic than the case of complete alternative choices because when the channel is good, the correct one will be the only likely choice. This case also shows that VSD is flexible in handling additional information.

CONCLUSIONS

This paper showed that second choices should be applied whenever possible, but no more than two choices should be used. This information is very useful for the implementation of a VSD decoder, which is currently under development in Intharasakul and Tuntoolavest (2006). More than two choices would have much more delay during the interfacing and need much larger memory size as well as the number of gates required in the FPGA (Field Programmable Gate Array) board that is used for the lab prototype of the decoder.

LITERATURE CITED

Brennan, D.G. 1959. **Linear Diversity Combining Techniques**. Proc. IRE., 47: 1075-1102.

Chen, B. and C-E. W. Sundberg. 2001. List Viterbi algorithms for continuous transmission. **IEEE Transactions on Communications**, 49, (5): 784-792.

Haslach, C. and A.J. Han Vinck. 1999. A decoding algorithm with restrictions for array codes. **IEEE Trans. Inform. Theory**, 45, (7): 2339-2344.

Intharasakul, R. and U. Tuntoolavest. 2006. State diagram design for implementing phase I of a Vector Symbol Decoder on an FPGA board. **International Symposium on Communications and Information Technologies Proceeding (ISCIT 2006)**, Bangkok, Thailand.

Metzner, J.J. and E.J. Kapturowski. 1990. A general decoding technique applicable to replicated file disagreement location and concatenated code decoding. **IEEE Trans. Inform. Theory**, 36: 911-917.

Metzner, J.J. 2003. Vector symbol decoding with list inner symbol decisions. **IEEE Trans. Comm.**, 51, (3): 371 – 380.

Nukmind N. and A. Suebnaung. 2003. **Effect of varying number of alternative choices on Vector Symbol Decoding**. Senior project in Electrical engineering, Kasetsart University.

Seo, Y.S. 1991. **A new decoding technique for convolutional codes**, Ph.D. Thesis, Penn State U, USA.

Tuntoolavest, U., J.J. Metzner. 2002. Vector symbol decoding with list symbol decisions and outer convolutional codes for reliable communications. **Integrated Computer-Aided Engineering Journal**, 9, (2): 101-116, IOS Press, ISBN 1069-2509.

Tuntoolavest, U. 2003. Performance comparison between a maximum and a non-maximum distance outer code decoding. **The 3rd International Symposium on Communications and Information Technologies Proceeding (ISCIT 2003)**, Songkhla, Thailand.

Tuntooolavest, U. 2004. A simple method to improve the performance of convolutional vector symbol decoding with small symbol size. **IEEE TENCON 2004**, B:676-679, Chiang Mai, Thailand.