TOWARD A BETTER UNDERSTANDING OF DIRTY PAPER TRELLIS CODES

Chin Kiong Wang, Gwenaël Doërr and Ingemar J. Cox

University College London Adastral Park Postgraduate Campus
{c.wang, g.doerr, i.cox}@adastral.ucl.ac.uk

ABSTRACT

Dirty paper trellis codes have been introduced as an alternative to lattice codes to implement watermarking as communications with side information. Their key feature is robustness against value-metric scaling in comparison with lattice codes. Despite the strong academic recognition, parametrization issues remain unclear. For instance, the impact of the trellis configuration on performance is still not well understood. In this paper, experiments on synthetic signals are reported to investigate how the trellis configuration influences the bit error rate and the computational complexity.

1. INTRODUCTION

Dirty paper trellis codes are a form of watermarking that is based on modeling watermarking as a communications system with side information. Channel coding with side information refers to a communications system in which the transmitter has additional knowledge (or side information) about the channel. In the early 1980's, theoretical studies of a communications channel with two noise sources, one of which is completely known to the transmitter, but neither of which is known to the receiver, revealed that, under Gaussian assumptions, the channel capacity was equivalent to a channel in which the first (known) noise source was absent [1, 2]. From a watermarking perspective, this (known) first noise source is equivalent to the cover Work e.g. an image in which we want to embed a message in, and the (unknown) second noise represents the distortions that the watermark undergoes between the time of embedding and detection.

Costas' result [2] implies that the cover Work need not interfere with the hidden message and thus offers the potential to hide a very large number of bits [3–5]. In practice, this is accomplished by using dirty paper codes which associate several codewords to a single message. For a given message, the associated codeword is chosen according to the available side information, i.e. the cover Work. Three main techniques have been proposed for watermarking with side information. These are lattice codes [6], syndrome codes [5] and dirty paper trellis codes [7].

Lattices codes, also referred to as Quantization Index Modulation (QIM), have received most attention due to their ease of implementation and their low computational cost. Nevertheless, they are usually criticized for being highly sensitive to value-metric scaling e.g. changes to the volume of audio signal can lead to complete loss of the watermark message. Although significant progress has been made toward resolving this issue [8–10], dirty paper trellis codes offer the potential for superior performance. These codes were originally proposed [7, 11] to overcome the issue of value-metric scaling. Although the original article briefly investigated the impact of the trellis structure on performance [7], it is still unclear whether the configuration chosen for experiments is optimal. The goal of this paper is to provide a better understanding of the interaction between the structure of the trellis and the performances in terms of bit error rate (BER).

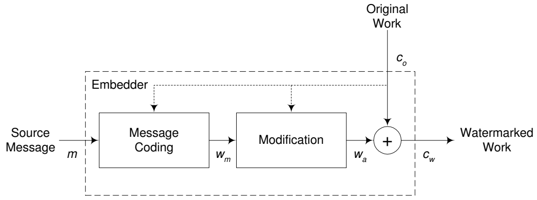

Fig. 1. Informed watermarking scheme.

Section 2 provides a brief review of dirty paper trellis codes. Section 3 then provides further insight about the performance of different dirty paper trellis configurations, especially in regard of the BER and the computational complexity. Experiments results are presented to validate the analysis. Finally, conclusions and open issues are discussed in Section 4.

2. TRELLIS DIRTY PAPER WATERMARKING

Fig. 1 depicts a typical informed watermarking system. For a given message m to be hidden, the message encoder proposes a set of watermark patterns and one of them ($\mathbf{w}_m$) is chosen for embedding based on the original cover Work $\mathbf{c}_\mathrm{o}$. Next, this watermark pattern undergoes some modifications with the influence of $\mathbf{c}_\mathrm{o}$ to produce an added mark $\mathbf{w}_\mathrm{a}$. Finally, this mark is added to the cover Work to produce the watermarked Work $\mathbf{c}_\mathrm{w}$. In this paper, we focus on the message coding step.

2.1. Blind Coding

The word *blind* is used to emphasize the fact that blind message coding does not make any use of available side information, i.e. the cover Work $\mathbf{c}_\mathrm{o}$. In other words, for a given message m, the encoder will always output the same watermark pattern $\mathbf{w}_m$. This is a one-to-one mapping and can be obtained using a traditional trellis as depicted in Fig. 2. Each node has two arcs emanating from it to two different nodes in the next column of nodes. A step is defined to be the transition from one column of nodes to the next column of nodes, moving from left to right. Each step corresponds to one message bit and each arc is labeled with a reference pattern of length N i.e. a N-samples pseudo-random sequence. Starting from node A0, the trellis is traversed from left to right by choosing a bold arc if the bit is "1" or an non-bold arc if the bit is "0". Thus,

1-4244-0469-X/06/$20.00 ©2006 IEEE

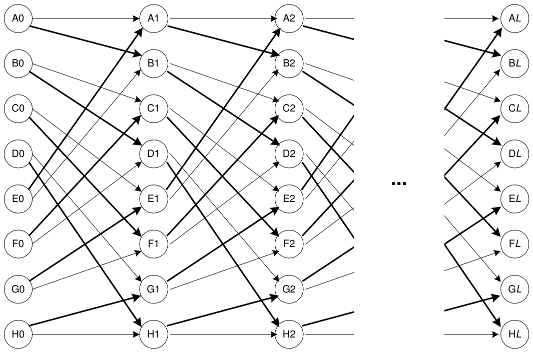

Fig. 2. Traditional 8-states trellis: two arcs enter/leave from each state.

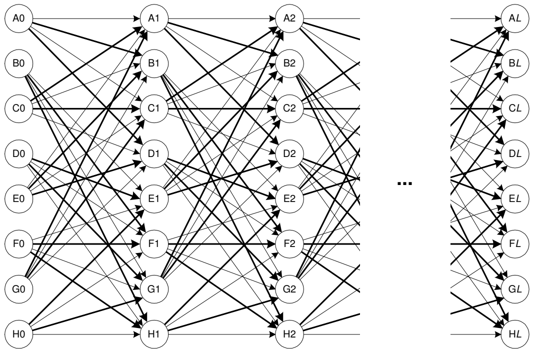

Fig. 3. Dirty paper 8-states trellis: four arcs enter/leave each state.

each L-bits message is mapped to a unique L-steps path through the trellis and the output watermark is obtained by concatenating the labels associated with the arcs of the path. It should be noted that the cover Work $\mathbf{c}_\text{o}$ is not involved in this process.

The resulting watermark $\mathbf{w}_m$ is subsequently embedding into the cover Work using a simple blind additive approach:

$$\mathbf{c}_\text{w} = \mathbf{c}_\text{o} + \alpha \mathbf{w}_m \tag{1}$$

where α is the embedding strength. During decoding, the most likely path through the trellis is determined by using the Viterbi algorithm [12]. To do so, each step of the trellis is associated with a portion of the cover Work $\mathbf{c}_\text{o}$ e.g. a few samples or coefficients. The cost of traversing an arc is then defined as the linear correlation between the reference pattern associated with the arc and the relevant portion of the cover Work. As a result, the Viterbi algorithm finds the path which exhibits the highest linear correlation with the cover Work.

2.2. Informed Coding

For informed coding, a given message m may be represented by several alternative codewords, one of which is chosen based on some criterion. Thus there is a one-to-many mapping between messages and codewords. A computationally efficient way to map a message to a desired codeword, consists of modifying a traditional trellis so that more than two arcs leave and enter a node. Such a trellis is shown in Fig. 3 and is referred to as dirty paper trellis [11]. A dirty paper trellis has the property that several paths through the trellis encode the same message. It is consequently necessary to tailor a procedure which decides which path, and by extension which watermark signal, will be retained for embedding. This is where the original cover Work $\mathbf{c}_\text{o}$ plays a role.

During the embedding process, the choice of which codeword to embed is determined by first modifying the dirty paper trellis so that all paths through the trellis encode the same message, This is accomplished by removing all the arcs which do not encode the desired message. For example, if the first message bit is a "0", the bold arcs are removed in the first step (nodes A0 ... H0 to nodes A1 ... H1). The Viterbi decoder is then run to find the path through this modified trellis which has the highest linear correlation with the input cover Work $\mathbf{c}_\text{o}$. Once again, the watermark $\mathbf{w}_m$ is obtained by concatenating all the labels of the arcs along the identified best path. In contrast with blind coding, both the message and the cover Work influence the encoding process.

The resulting watermark $\mathbf{w}_m$ is then embedded blindly according to Eq. (1). At the receiver, the decoder applies the Viterbi algorithm to the entire dirty paper trellis, as depicted in Fig. 3. This identifies the path through the trellis which has the highest linear correlation with the watermarked cover Work $\mathbf{c}_\text{w}$. The hidden message can then be determined by examining each arc in the optimum path to determine whether it encodes a "1" or "0".

3. PERFORMANCE CHARACTERIZATION FOR DIRTY PAPER TRELLISES

In [7, 11] an empirical investigation revealed that a good compromise between computation efficiency and bit error rate was obtained with a trellis structure consisting of 64 states and 64 arcs per state. However, it remains unclear how the structure of the trellis affects performance. In the following subsections, two criteria will be investigated to compare alternative trellis configurations: (i) the bit error rate and (ii) the computational complexity of Viterbi decoding of the full trellis.

3.1. Trellis Structure vs. Bit Error Rate

For a given dirty paper trellis, the total number of paths through the trellis, i.e. the number of codewords in the codebook, is determined by the following formula:

$$n_c = S.A^L, \tag{2}$$

where S is the number of states in the trellis, A the number of arcs per state and L is the number of steps in the trellis. The comparison of different trellis architectures can be based on a number of different decoding performance metrics, including Bit Error Rate (BER), Message Error Rate (MER) or Path Error Rate (PER). The BER is the probability that a message bit is incorrectly decoded. The MER is the probability that a message is incorrectly retrieved after embedding; a single bit error induces a message error. The path error rate (PER) is the probability that the path output by the Viterbi decoder differs between message encoding and watermark detection; if these two path are different, it is a path error even if the extracted path encodes the same message. Because of its popular use, we chose the BER to assess performance.

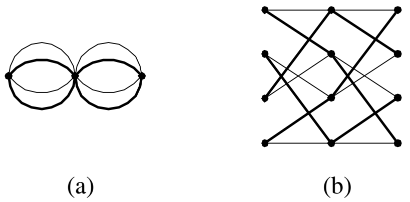

Fig. 4. Different trellis configurations for 16 codewords: (a) 1 state, 4 arcs/state, length 2 trellis and (b) 4 states, 2 arcs/state, length 2 trellis.

When alternative dirty paper trellis configurations are compared, it is necessary to ensure that some parameters are fixed to allow a fair comparison. One can for instance keep the number of codewords fixed. In this perspective, Fig. 4 depicts two alternative trellis configurations which share the same total number of codewords, $n_c = 16$. Configuration (b) is basically a traditional 4-states trellis whereas configuration (a) is a degenerate trellis with only a single state. Keeping the number of codewords constant means that the measured BER variations are only due to the changes in the trellis configuration. In the proposed example, configuration (a) is memoryless, i.e. an error can occur at each step independently from decisions in the previous steps. Therefore, the BER is likely to be higher than with configuration (b). In the latter case, errors are indeed more costly since an error at the first step induces necessarily a second error.

In practice, maintaining a fixed number of codewords is not always possible. Referring back to Eq. (2), if the number of arcs per states A is divided by 2, then the number of states should be multiplied 2^L, which can rapidly grow huge and intractable. As a result, the decoding performance of all possible trellis configurations have to be evaluated.

3.2. Experiments

For simplicity, experiments have been carried out with synthetic signals. To reduce computational complexity, the number of steps in the trellis has been set to 10 ($L = 10$) i.e. the hidden message consists of 10 bits. The length of the arc labels has been set to 64 ($N = 64$) and each instance of those patterns is drawn from a Gaussian distribution with zero mean and unit variance. Since we are interested in the affect of the trellis structure, the number of states S and the number of arcs per state A are parameters. Therefore for a selected set of couples (S, A), the following experiment is run 10^6 times.

1. Generate a random cover Work $\mathbf{c}_{\mathrm{o}} \sim \mathcal{N}(0, 1)$ of length $N.L$
2. Generate a dirty paper trellis with S states and A arcs per state
3. Generate a random L-bits message m
4. Identify the path $\mathbf{p}_1$ in the trellis which encodes the message m and has the highest linear correlation with $\mathbf{c}_{\mathrm{o}}$ using the Viterbi decoder
5. Use Eq. (1) to embed the resulting watermark $\mathbf{w}_m$ with an embedding strength $\alpha = \sqrt{0.1}$ so that the Document-to-Watermark Ratio (DWR) is equal to 10 dB[1]

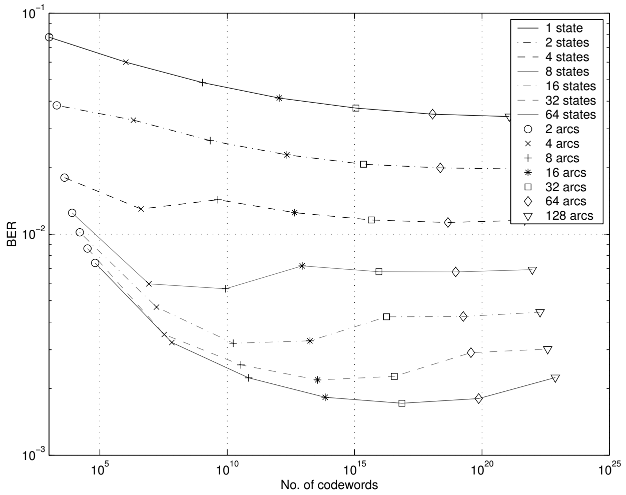

Fig. 5. BER as a function of the number of codewords for different trellis configurations.

6. Corrupt the watermarked Work $\mathbf{c}_{\mathrm{w}}$ with additive zero mean Gaussian noise whose variance is adjusted to obtain a Watermark to Noise Ratio (WNR) equal to -7 dB
7. Identify the path $\mathbf{p}_2$ in the *entire* trellis which has the highest linear correlation with the corrupted watermarked Work using the Viterbi decoder
8. Compute the number of different arcs between $\mathbf{p}_1$ and $\mathbf{p}_2$

The BER can then be computed by dividing the total number of reported arc errors by the number of iterations times the length L of the path.

Fig. 5 reports the measured BER for different dirty paper trellises with respect to the number of codewords n_c. As one can expect, the BER first decreases as the number of codewords increases. This is intuitive, since the larger the codebook is, the more likely it is that a codeword exists which is similar to the cover Work. Moreover, because of noise addition, the BER raises up again after some point. Indeed, increasing the number of codewords decreases the detection region of each individual codeword and when they become to small, the watermarked Work start to be sensible to noise addition. Nevertheless, some “bumps” can be isolated which seem to contradict this generic behaviour. When looking closely at Fig. 5, it can be noted that the bumps occur when some parallel arcs, i.e. arcs linking the same states in the trellis, are introduced in the trellis ($A > S$). In this case, single errors can occur without inducing additional errors. In other words, making errors is cheap and thus happen more often. This is an important difference with trellis configurations on the left side of these bumps where an error necessarily induces other ones i.e. making an error is costly and thus happens more rarely. This suggests that configurations with $A > S$ should be avoided. Finally, it can be noticed that for a given number of codewords (roughly, A constant), very different BER can be obtained depending on the trellis configuration.

[1] This value has been chosen to observe enough bit errors to estimate the BER without running huge number of iterations, but not to many which would stack all the curves at the top of the figure.

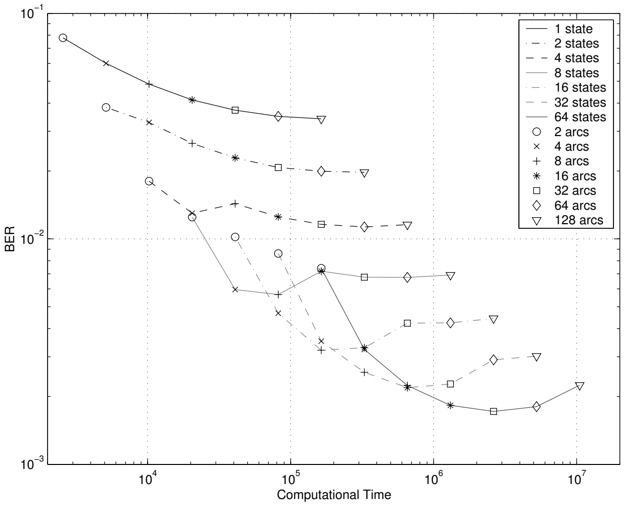

Fig. 6. BER vs. computational time for different trellis structures.

3.3. Trellis Structure vs. Computational Time

Even if a trellis configuration is found to give good decoding performance, its computational complexity should also be measured to determine whether it can be used in practice or not. With trellis dirty paper watermarking, the most costly operation is the Viterbi decoding of the whole trellis. To do it, it is necessary to compute the cost function for each arc. Since linear correlation is used, this requires N multiplications, $(N-1)$ additions and once division. Since there are $A.S.L$ arcs in the trellis, the computational time is given by:

$$\tau = A.S.L\big(N\tau_{\times} + (N-1)\tau_{+} + \tau_{/}\big) \qquad (3)$$

where $\tau_{\times}$ (resp. τ_{+} and $\tau_{/}$) is the computational time of a multiplication (resp. addition and division). Assuming that these three values are equal, then the computational time is simply given by $2.A.S.L.N$. Referring back to Fig. 4, it means that configuration (a) has half the computational cost of configuration (b). In other words, the good performance in terms of BER is counterbalanced by a higher computational cost.

Fig. 6 shows the performance of different dirty paper trellises in terms of BER with respect to the computational time which is given by Eq. (3). This shifts the previous curves horizontally so that configurations sharing the same total number $A.S.L$ of arcs are aligned vertically. The main point to notice is that a few specific trellis configurations appear to offer a significantly better compromise. For instance, assuming that we want to obtain a BER about 3.10^{-3}, Fig. 5 indicates that three alternative configurations can be used, namely $(S, A) = (16, 8)$, $(32, 4)$ or $(64, 4)$. However, if they have similar performance in terms of BER, Fig. 3 clearly shows that configuration $(64, 4)$ should be discarded because of its higher computational cost. In fact, looking closely at this figure, one can easily isolate the set of configurations which optimize the considered trade-off at the studied WNR regime. In the reported experiment, the following configurations should be used: $(1, 2)$, $(2, 2)$, $(4, 2)$, $(8, 2)$, $(8, 4)$, $(16, 4)$, $(16, 8)$, $(32, 8)$ and so on.

4. DISCUSSION

Despite the strong academic recognition, several factors have delayed the adoption of dirty paper trellis codes. One of them was that the influence of the trellis structure on performance was unclear. In this paper, we have measured the BER and the computational time for many different configurations of the dirty paper trellis. The reported experiments have provided a better understanding of the impact of different trellis configurations. The following conclusions can be drawn: (i) parallel arcs should be avoided in the trellis structure since they introduce single errors, (ii) alternatives trellis configurations sharing the same number of codewords lead to very different BER and (iii) alternative trellis configurations sharing the same BER may have very different computational costs. In future work, we will further investigate how the trellis configuration affects the distribution of the dirty paper trellis codewords on the surface of the unit hypersphere. Our recent studies have indeed suggested that this distribution can be far from uniform or regular as one would expect.

5. REFERENCES

[1] S. I. Gel'fand and M. S. Pinsker, "Coding for channel with random parameters," *Problems of Control and Information Theory*, vol. 9, no. 1, pp. 19–31, 1980.

[2] M. Costa, "Writing on dirty paper," *IEEE Trans. Inform. Theory*, vol. 29, pp. 439–441, 1983.

[3] B. Chen and G. W. Wornell, "An information-theoretic approach to the design of robust digital watermarking systems," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1999.

[4] I. J. Cox, M. L. Miller, and A. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, no. 7, pp. 1127–1141, 1999.

[5] Jim Chou, S. Sandeep Pradhan, and Kannan Ramchandran, "On the duality between distributed source coding and data hiding," *Thirty-third Asilomar conference on signals, systems, and computers*, vol. 2, pp. 1503–1507, 1999.

[6] B. Chen and G Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1423–1443, May 2003.

[7] M. Miller, G. Doërr, and I. Cox, "Applying informed coding and informed embedding to design a robust, high capacity watermark," *IEEE Transactions on Image Processing*, vol. 13, pp. 792–807, June 2004.

[8] K. Lee, D. Kim, T. Kim, and K. Moon, "Em estimation of scale factor for quantization-based audio watermarking," in *Proceedings of the Second International Workshop on Digital Watermarking*, October 2003, vol. 2939, pp. 316–327.

[9] Q. Li and I. Cox, "Using perceptual models to improve fidelity and provide invariance to valumetric scaling for quantization index modulation watermarking," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, March 2005, vol. II, pp. 1–4.

[10] F. Pérez-González, F. Mosquera, M. Barni, and A. Abrardo, "Rational dither modulation: A high-rate data-hiding method invariant to gain attacks," *IEEE Transactions on Signal Processing, Supplement on Secure Media*, vol. 53, pp. 3960–3975, October 2005.

[11] G. L. Miller, G. Doërr, and I. J. Cox, "Dirty-paper trellis codes for watermarking," in *IEEE Int. Conf. on Image Processing*, 2002.

[12] A. J. Viterbi, *CDMA: principles of spread spectrum communications*, Addison Wesley Longman Inc., 1995.