

Decoding the Golden Space-Time Trellis Coded Modulation

Barbara Cerato, *Member, IEEE*, Guido Masera, *Senior Member, IEEE*,
and Emanuele Viterbo, *Senior Member, IEEE*

Abstract—This letter presents new techniques enabling low complexity implementation of a decoder for 2×2 MIMO using an inner space-time block code (STBC) concatenated with an outer Trellis Coded Modulation (TCM). The receiver includes a Viterbi decoder and sphere decoders to compute branch metrics. We present a technique to reduce the number of sphere decoding processes at each step of the trellis still retaining maximum-likelihood (ML) frame-error rate (FER) performance as well as a sub-optimal technique to speed up the decoding, with a negligible FER performance loss.

Index Terms—MIMO systems, space-time codes decoding, sphere decoding, concatenated codes.

I. INTRODUCTION

IN slow fading channels where coefficients are fixed over a fairly long frame, concatenated coding schemes can be used in order to improve coding-gain without incurring exponential decoding complexity. The inner space-time block code guarantees the full diversity, full multiplexing gain or their trade-off, while the outer code is used to improve the coding gain. The Golden Space-Time Trellis Coded Modulation (GST-TCM), proposed in [1], combines an outer Trellis Coded Modulation (TCM) with set partitioning of the inner Golden code for 2×2 MIMO.

This concatenation scheme is very well suited for high-rate real-time and stream communications, since it results in significant performance improvements, without incurring in large delay. The latency of the decoding can be easily adapted to the system requirements by selecting an appropriate frame length.

The inner space-time code is the Golden code, sending 2×2 complex matrix codewords. By standard vectorization and separation of real and imaginary parts the codewords can be viewed as points in an 8-dimensional Euclidean space. In particular, the Golden code can be mapped to a $(Q\text{-PAM})^8$ constellation, carved from a rotated 8-dimensional integer lattice \mathbb{Z}^8 . Set partitioning is performed to increase the minimum determinant of the Golden codewords within each subset. This corresponds to a lattice set partitioning of \mathbb{Z}^8 , using the cosets of one of its sublattices [1].

II. SYSTEM ARCHITECTURE

In Figure 1 the encoder structure is depicted for the two level partition scheme \mathbb{Z}^8/E_8 , where the sublattice E_8 is

Manuscript received April 7, 2008. The associate editor coordinating the review of this letter and approving it for publication was G. Karagiannidis.

B. Cerato and E. Viterbo are with DEIS – Università della Calabria, Rende (CS), Italy (e-mail: bcerato@deis.unical.it).

G. Masera is with CERCOT – Dipartimento di Elettronica – Politecnico di Torino, Torino, Italy.

Digital Object Identifier 10.1109/LCOMM.2008.080543.

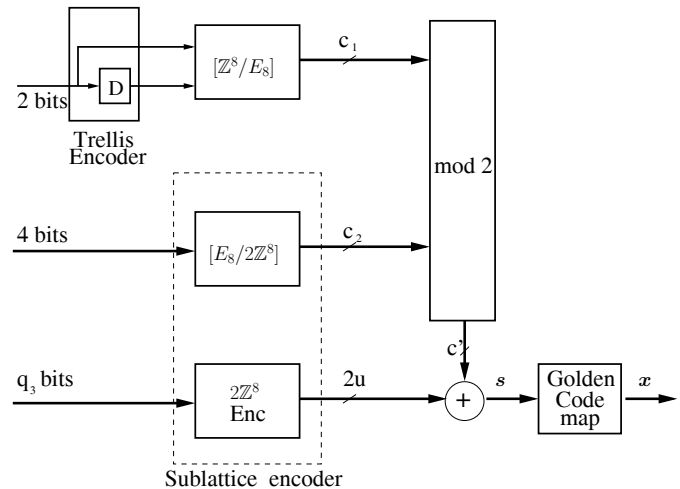


Fig. 1. Block diagram for the GST-TCM encoder.

Gosset lattice [2]. We will refer to this simple case to illustrate the key idea of this work.

Two information bits enter a rate 1/2 convolutional encoder and the four output bits are used to select one of the 16 cosets of E_8 compounding the \mathbb{Z}^8 lattice. In particular, they select one of the 16 coset leaders (c_1) in the set $[\mathbb{Z}^8/E_8]$. A second coset leader c_2 is chosen by means of 4 other information bits, which select one of the 16 cosets of $2\mathbb{Z}^8$ compounding E_8 . Finally $c' = c_1 \oplus c_2$ (\oplus denotes EXOR) represents the coset leader of one of the 256 cosets of $2\mathbb{Z}^8$ compounding \mathbb{Z}^8 . The parameter q_3 depends on the size of the $Q\text{-PAM}$ modulation and the last q_3 bits entering the encoder are used by the sublattice encoder to select a $2\mathbb{Z}^8$ lattice point within a cubic shaped constellation.

A. Decoder implementation

Maximum likelihood decoding of the proposed scheme can be achieved through a Viterbi decoder. The trellis corresponding to the adopted convolutional encoder with a single quaternary memory element “D” has four states and there are 16 parallel transitions between each pair of states as result of the sublattice encoding. As a consequence, each state has 64 incoming and 64 outgoing branches (4 groups of 16 parallel branches).

The Viterbi decoder is structured according to the well-established three-unit architecture, with a Branch Metric Unit (BMU) computing the metrics, an Add-Compare-Select Unit (ACS) that updates state metrics and a Trace Back Unit (TBU) that reconstructs the most probable trellis path and decodes finally all the codewords.

Since the Viterbi algorithm needs soft input information, Euclidean distances have to be evaluated in the BMU. The sphere decoder [3] computes the closest point by calculating the minimum Euclidean distance d between the received symbol and each point of the (rotated and faded) \mathbb{Z}^8 lattice. A trivial approach, proposed in [1], consists in evaluating the metrics associated with all the outgoing branches from all the states. This can be obtained by performing the sphere decoding on 256 different $2\mathbb{Z}^8$ lattices, each one translated by a different coset leader (c') associated with that branch.

However, for the parallel transitions not all the distance values are necessary but only the minimum value within each group is needed when adding to the state metric. Moreover, the union of the $2\mathbb{Z}^8$ sublattices associated to parallel transitions corresponds to E_8 . As a consequence, the sphere decoder has to solve:

$$\hat{s} = \arg \min_{s \in E_8} \|\mathbf{y} - \mathbf{M}s\|^2 \quad (1)$$

where \mathbf{y} is the received vector, s is the signal vector after the TCM encoding and before the Golden code encoding (see Figure 1), and \mathbf{M} is the matrix given by the combination of channel effects and Golden code matrix.

Intuitively this problem can be solved, generalizing [4], by thinking of E_8 as a \mathbb{Z}^8 lattice where some points are not allowed and finding the constraint that must be satisfied so that s belongs to the requested lattice. Recalling the tree structure for the search in sphere decoding process [3], in some components the signal can not assume all values but it is constrained by the selections that have already been made at upper levels.

Construction A of E_8 can be written as:

$$E_8 = 2\mathbb{Z}^8 + [8, 4, 4], \quad (2)$$

where [8,4,4] is the extended Hamming code, and allows to obtain the coset leader generators in [1]. From their analysis, points belonging to a translated E_8 lattice present four *free* and four *constrained* components. In particular, taking into account that the sphere decoder tree must be traversed starting from the last dimension of these vectors, we can observe that all the E_8 coset leaders c_2 can be obtained as:

$$c_2 = \begin{cases} c_2(7) = \textit{free} \\ c_2(6) = \textit{free} \\ c_2(5) = \textit{free} \\ c_2(4) = c_2(7) \oplus c_2(6) \oplus c_2(5) \\ c_2(3) = \textit{free} \\ c_2(2) = (c_2(4) \oplus c_2(3)) \cdot c_2(5) \\ c_2(1) = (c_2(4) \oplus c_2(3)) \cdot c_2(6) \\ c_2(0) = (c_2(4) \oplus c_2(3)) \cdot c_2(7) \end{cases} \quad (3)$$

Since $c' = c_1 \oplus c_2$, c_1 is a parameter fixed for the sphere decoder, while the allowed and interdicted values of the signal s_i depend on the value of c' . If $c'(i) = '0'$ then s_i can take the values $[\dots, -7, -3, 1, 5, \dots]$ otherwise it can take the values $[\dots, -5, -1, 3, 7, \dots]$. The bounds of these sets depend on the constellation used for transmission. It is worth noting that, when $c_2(i)$ is *free*, $c'(i)$ can assume both the values 0 and 1 leading s_i to assume any value in the original PAM constellation.

TABLE I
A: SYNTHESIS RESULTS FOR GST-TCM SYSTEMS

Flexible up to:	Core [mm^2]	GE Core [kGate]	Memory [mm^2]	Tot. Area [mm^2]	Freq. [MHz]
16 QAM	4.94868	824	0.12326	5,07194	200
64 QAM	5.55567	922	0.14611	5.70179	174

We consider, as an example, two central levels 4 and 3, and we assume that at upper levels have already been selected $c_2(7) = '1'$, $c_2(6) = '0'$ and $c_2(5) = '1'$. These choices, using (3), lead to $c_2(4) = '0'$. Moreover, if for the considered sphere decoder $c_1(4) = '1'$, then it results $c'(4) = '1'$. In this example, adopting a 16-QAM constellation, the values $[-1, 3]$, are allowed for s_4 . At level 3, instead, $c_2(3)$ is free so $c'(3)$ can assume both values '0' and '1' and the four branches are all admissible.

In the proposed scheme, an E_8 sphere decoder allows to explore the parallel transitions with a unique circuit. This will lead to a first considerable reduction in the number of requested branch computations from 256 to 16. In this case the decoder must work on the original \mathbb{Z}^8 grid instead of the $2\mathbb{Z}^8$. The sphere decoder will only visit the E_8 and skip the others, while scanning the \mathbb{Z}^8 points within the constellation boundaries. The complexity reduction of this single slower E_8 decoder is still large compared to the faster 16 sphere decoders for $2\mathbb{Z}^8$.

The realization of the single sphere decoder is not addressed in this paper, possible implementations can be found in [5], [6], while the adopted architecture is very similar to the decoder described in [7].

The ACS unit has the task of selecting at each trellis step the most probable path leading to a certain state and to update accordingly the state metric. In our system there are four ACS units and each of them has to select among four incoming metrics.

Branch metrics are represented on 16 bits for the synthesis presented below. The well-established approach presented in [8] is adopted to guarantee a sufficient number of bits for state metrics. Using 3 additional bits, metric wrapping in two's complement notation allows to avoid normalization circuits, with no performance loss.

The TBU has the task of identifying the minimum survivor path stored by the ACS in memory and reconstruct it. In order to limit the required amount of memory, the widely known mechanism of sliding blocks [9] has been adopted, splitting the data frame into several windows, each one of length 16.

III. FASTER SUB-OPTIMAL DECODING STRATEGY

In Table I, synthesis results on a 0.13 μm commercial CMOS standard cell technology for the described GST-TCM concatenation scheme are summarized. This architecture has never been realized in hardware before, hence no comparisons are possible. The required area is dominated by the 16 sphere decoders, each one taking about 50 KGate. The offered throughput (slightly more than 12 Mbit/s with a clock frequency of 200 MHz, as indicated in Table II, second column) is severely limited by the need that all sphere decoders find the best solution within their cosets.

TABLE II
THROUGHPUT AT VARYING OF PARAMETER K

	ML	$K=4$	$K=3$	$K=2$
Through.[Mb/s]	12.28	82.4	87.2	93

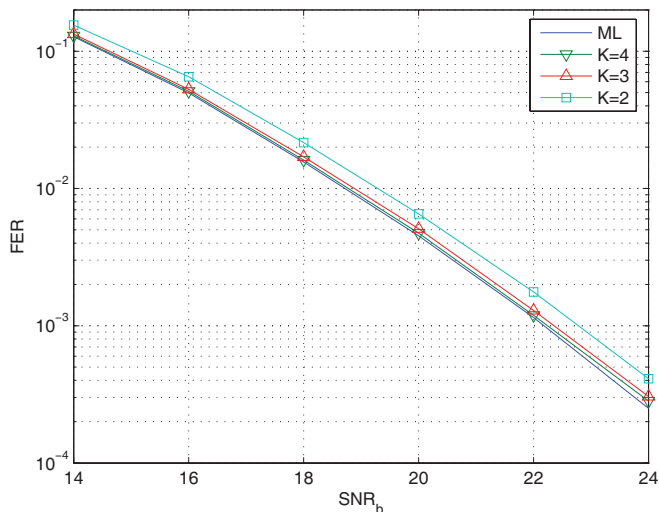


Fig. 2. FER of GST_TCM at varying of parameter K .

Moreover the throughput does not improve with the increasing SNR, as in common sphere decoding applications. This behaviour can be explained by the fact that, at high SNR, at each trellis step, only one branch has a small metric, while *all* the others have a large metric. Furthermore, in the cosets corresponding to large branch metrics the received symbol is likely to be approximately at the same distance from more than one point: this slows down the convergence of the search algorithm.

A substantial throughput improvement is achieved by stopping all the 16 parallel sphere decoders as soon as the first one ends its search, so keeping sub-optimal solutions for the other ones. Unfortunately, with this approach the distances collected in other branches are not sufficiently accurate and do not allow the Viterbi algorithm to correct many recoverable errors, leading to a consistent FER performance loss.

In order to operate an effective and not detrimental simplification, we observe that in the Viterbi algorithm it would be sufficient to find for each state the minimum path metric. Large branch metrics summed with their associated state metrics are unlikely to give competitive path metrics. We thus propose a more refined strategy: after obtaining the squared distance d^2 from the first sphere decoder that ends its search, we restrain the search of the other ones by limiting their square search radius [3], [10] to a multiple $K > 1$ of d^2 :

$$\|y - Ms\|^2 \leq Kd^2 \quad (4)$$

The value of parameter K needs to be carefully tuned: Figure 2 reports FER performance offered by the described sub-optimal technique as a function of parameter K , while in Table II we give the average throughput achievable by the unit with different choices of $K=2,3,4$ at $\text{FER}=10^{-3}$. All values refer to the 16QAM case and assume a clock frequency of 200 MHz.

This technique allows to speed up the convergence of all the sphere decoders which output at relatively low distance from the received point. On the other hand it stops the other decoders that cannot find candidate solutions within Kd^2 .

IV. CONCLUSIONS

In this letter we deal with the first hardware implementation of GST-TCM, a very recent scheme very well suited for real-time and stream communications, which require low latency in the decoding. The adopted scheme shows good performance also with short frame length thus guaranteeing a fast response.

With respect to a trivial decoding strategy, the use of the E_8 sphere decoder allows us to reduce the area occupation by more than one order of magnitude, while the proposed sub-optimal technique, as it is shown in Figure 2 and Table I, enables the realization of a six-fold speed-up with a very small FER performance loss (less than 0.1dB for $K > 2$).

REFERENCES

- [1] Y. Hong, E. Viterbo, and J.-C. Belfiore, "Golden space-time trellis coded modulation," *IEEE Trans. Inform. Theory*, May 2007.
- [2] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1992.
- [3] E. Viterbo and E. Biglieri, "A universal decoding algorithm for lattice codes," in *Quattorzieme Colloquio GRETSI*, Sept. 1993.
- [4] B. Cerato, G. Masera, and E. Viterbo, "Enabling VLSI processing blocks for MIMO-OFDM communications," *Hindawi VLSI Design*, special issue on the International Conference on Electronics, Circuits and Systems, vol. 2008, article ID 351962, 10 pages, 2008.
- [5] A. Burg, M. Borgmann, M. Wenk, M. Zellwegger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, July 2005.
- [6] L.G. Barbero and J. C. Thomson, "Rapid prototyping of the sphere decoder for MIMO systems," in *Proc. IEE/EURASIP Conference on DSPEnabledRadio, 2005 (Ref. No. 2005/11086)*, Sept. 2005.
- [7] B. Cerato, G. Masera, and E. Viterbo, "A VLSI decoder for the Golden code," in *Proc. 13th IEEE International Conference on Electronics, Circuits and Systems (ICECS 2006)*, Nice, France, Dec. 2006.
- [8] A. P. Hekstra, "An alternative to metric rescaling in Viterbi decoders," *IEEE Trans. Commun.*, vol. 37, no. 11, pp. 1220–1222, Nov. 1989.
- [9] P. J. Black and T. H. Meng, "1-Gb/s, four-state, sliding block Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 32, no. 6, pp. 797–805, June 1997.
- [10] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1642, July 1999.