IMPROVED DECODING OF ANALOG MODULO BLOCK CODES FOR NOISE MITIGATION

Tim Schmitz, Peter Jax, Peter Vary

Institute of Communication Systems (ind)
RWTH Aachen University, Germany
{schmitz|jax|vary}@ind.rwth-aachen.de

ABSTRACT

A drawback of digital transmission of analog signals is the unavoidable quantization error which leads to a limited quality even for good channel conditions.

This saturation can be avoided by using analog transmission systems with discrete-time and quasi-continuous-amplitude encoding and decoding, e.g., Analog Modulo Block codes (AMB codes). The AMB code vectors are produced by multiplying a real-valued information vector with a real-valued generator matrix using a modulo arithmetic.

Here, algorithms for improving the decoding performance are presented. The Lattice Maximum Likelihood (LML) decoder, a variant of the Discrete Maximum Likelihood (DML) decoder, is derived and analyzed. It refines the Zero Forcing (ZF) result if necessary, thus achieving near-ML signal quality with a reduced decoding complexity. A reduced complexity is essential for decoding high-dimensional code words. Additionally, pre- and post-processing methods are presented and analyzed, which increase the signal-to-distortion ratio (SDR) of the received symbols.

***Index Terms*—** analog channel coding, decoding, modulo operation, noise mitigation

1. INTRODUCTION

Digital transmission systems are very good if the used channel code is designed for the correct channel quality. The irreversible quantization error that is induced by the source encoder is their most prominent disadvantage. Hybrid digital-analog (HDA) systems [1–4], which transmit the quantization error in analog form as side information, are one way to circumvent this disadvantage.

By applying a continuous-amplitude, discrete-time channel code, the quantization can be avoided completely. In contrast to linear analog channel codes that can correct burst errors, e.g., [5, 6], Analog Modulo Block Codes (AMB codes, introduced in [7]) apply a modulo operation after multiplying an analog information vector with a generator matrix. The modulo operation adds a non-linearity, which improves the energy budget for transmission compared to Linear Analog Block Codes [8]. AMB codes can be applied in low-delay applications where continuous-amplitude source symbols are transmitted. A possible field of application are systems with low-complexity transmitters – e.g., transmitters used in hearing aids, wireless sensors, microphones, or loudspeakers – that are not designed to adapt to changing channel conditions.

Existing decoders for AMB codes are the Minimum Mean Square Error (MMSE) decoder, the Discrete Maximum Likelihood (DML) decoder and the Zero Forcing decoder with Lattice Reduction (ZFLR) [7]. The DML decoder is a compromise between the low computational complexity of the ZFLR decoder and the MMSE decoder, which achieves a minimum distortion. A novel approach to achieve near-DML results with a reduced computational complexity is presented in Section 4. Additional methods for pre- and post-processing, which further reduce the decoding error, are shown in Section 5.

2. SYSTEM MODEL

The system model from [7] is used. A block diagram of the transmission system is shown in Fig. 1.

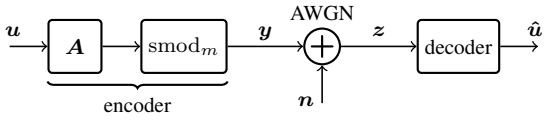


Fig. 1. Block diagram of the transmission system.

A source vector $\boldsymbol{u} \in \mathbb{R}^M$ is considered with elements u_i, where $-U \leq u_i < U$. This vector[1] is encoded with an Analog Modulo Block Code (AMB code) by multiplying it with a generator matrix $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ (where $M < N$), followed by element-wise application of a modified (symmetric) modulo operation

$$\mathrm{smod}_m(x) = \big((x+m) \mod 2m\big) - m \quad \text{for } x \in \mathbb{R}, \tag{1}$$

which maps the input symbols onto the range $(-m, +m)$ as shown in Fig. 2. The value of m does not have any effect on the performance if $\boldsymbol{A}$ is scaled appropriately. In this paper, we assume $m = 1$, but it is explicitly mentioned in the equations to keep track of its influence.

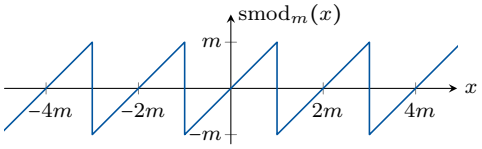


Fig. 2. Modified (symmetric) modulo function.

Briefly, the code words are defined by

$$\boldsymbol{y} = \mathrm{smod}_m(\boldsymbol{u}\cdot\boldsymbol{A}). \tag{2}$$

The code rate is $r = \frac{M}{N}$, as the encoder maps M source symbols onto N channel symbols.

Here, we focus on *systematic* AMB codes with $\boldsymbol{A} = \begin{bmatrix}\mathbb{1} & \tilde{\boldsymbol{A}}\end{bmatrix}$ for simplicity. Because of the $(M\times M)$ identity matrix $\mathbb{1}$, the information words are contained in the code words, assuming that $0 < U \leq m$.

The resulting code vector $\boldsymbol{y} \in \mathbb{R}^N$ is transmitted over a discrete-time Additive White Gaussian Noise (AWGN) channel, which is modeled by adding the noise vector $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0},\ \sigma_n^2 \cdot \mathbb{1})$, $\boldsymbol{n} \in \mathbb{R}^N$.

[1] All vectors in this paper are row vectors.

 In *International Conference on Acoustics, Speech, and Signal Processing*, Mar. 2016, pp. 3861–3865, DOI: 10.1109/ICASSP.2016.7472400

Finally, the received vector

$$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{n} \tag{3}$$

is processed by a decoder which generates an estimate $\hat{\boldsymbol{u}} \in \mathbb{R}^M$ of the source vector $\boldsymbol{u}$.

The signal-to-noise ratio on the channel

$$CSNR = \frac{\mathrm{E}\left\{\|\boldsymbol{y}\|^2\right\}}{\mathrm{E}\left\{\|\boldsymbol{n}\|^2\right\}} = \frac{N \cdot \sigma_y^2}{N \cdot \sigma_n^2} = \frac{\sigma_y^2}{\sigma_n^2} \tag{4}$$

is used as an expression for the channel quality, while the signal-to-distortion ratio

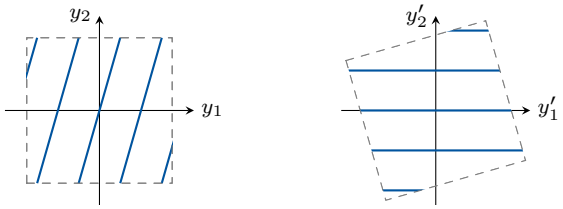
$$SDR = \frac{\mathrm{E}\left\{\|\boldsymbol{u}\|^2\right\}}{\mathrm{E}\left\{\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^2\right\}} = \frac{M \cdot \sigma_u^2}{M \cdot MSE} = \frac{\sigma_u^2}{MSE}, \tag{5}$$

denotes the quality of the transmission, where MSE is the *Mean Square Error*.

3. BASICS AND DECODING

In this section, the basics of AMB codes and different methods for decoding are summarized, which have already been elaborated in [7].

The modulo function limits all code words to a (hyper-)cube with side length $2m$, which is caleld *modulo cube*. The code words are located on distinguishable lines, which are in general parallel M-dimensional subspaces of the code space $\mathbb{R}^N$. Fig. 3 shows the code words of a simple code with $M = 1$ and $N = 2$.



(a) Original code words $\boldsymbol{y}$ (b) Rotated code words $\boldsymbol{yG}$

Fig. 3. Valid code words and cube with $\boldsymbol{A} = \begin{bmatrix}1 & 3.5\end{bmatrix}$ and $m = 1$.

3.1. Rotation

By rotating these subspaces in such a way that they are aligned with M of the N dimensions, they form a lattice in the remaining $N - M = D$ dimensions, which we call *discrete dimensions* (y_2' in Fig. 3). We can choose an $N{\times}D$ matrix $\boldsymbol{G}_\mathrm{d}$ that applies this rotation and gets rid of the M continuous dimensions (y_1' in Fig. 3). This matrix, which only depends on the generator matrix $\boldsymbol{A}$, maps the valid code words $\boldsymbol{y}$ to discrete points (horizontal lines in Fig. 3b)

$$\boldsymbol{y}_\mathrm{d} = \boldsymbol{y}{\cdot}\boldsymbol{G}_\mathrm{d} = \tilde{\boldsymbol{s}}{\cdot}\boldsymbol{B} \quad \text{with} \quad \tilde{\boldsymbol{s}} \in \mathbb{Z}^D. \tag{6}$$

These points lie on a lattice with the base matrix $\boldsymbol{B} \in \mathbb{R}^{D\times D}$, which can be derived from $\boldsymbol{G}_\mathrm{d}$ as shown in [7].

3.2. Decoding

In order to decode a received word $\boldsymbol{z}$, we use $\boldsymbol{z}_\mathrm{d} = \boldsymbol{z}{\cdot}\boldsymbol{G}_\mathrm{d}$ to get an estimate $\hat{\boldsymbol{y}}_\mathrm{d}$ of the discrete lattice point $\boldsymbol{y}_\mathrm{d}$. Then, the continuous dimensions (that had been discarded by $\boldsymbol{G}_\mathrm{d}$) are restored. Finally, a multiplication with the pseudoinverse $\boldsymbol{A}^+ = \boldsymbol{A}^\mathrm{T} \cdot \left(\boldsymbol{A}\boldsymbol{A}^\mathrm{T}\right)^{-1}$ of $\boldsymbol{A}$ yields an estimate $\hat{\boldsymbol{u}}$ of the information word

$$\hat{\boldsymbol{u}} = \left(\boldsymbol{z} - 2m\begin{bmatrix}\boldsymbol{0} & \hat{\boldsymbol{y}}_\mathrm{d} \cdot \boldsymbol{B}^{-1}\end{bmatrix}\right) \cdot \boldsymbol{A}^+. \tag{7}$$

In Section 4 the decoders *Discrete Maximum Likelihood* and *Zero Forcing with Lattice Reduction* are used. Their methods for the determination of $\hat{\boldsymbol{y}}_\mathrm{d}$ are presented in the following.

3.2a) The discrete maximum likelihood (DML) approach determines the lattice point that is closest to $\boldsymbol{z}_\mathrm{d}$:

$$\hat{\boldsymbol{y}}_{\mathrm{d}\,\mathrm{DML}} = \underset{\boldsymbol{y}_\mathrm{d}}{\arg\min} \left\|\boldsymbol{z}_\mathrm{d} - \boldsymbol{y}_\mathrm{d}\right\|. \tag{8}$$

This method yields very good results, but it is computationally complex, as the received sample has to be compared to each possible valid discrete lattice point. A computationally less complex approximation of this approach is presented in Section 4.

3.2b) The Zero Forcing (ZF/ZFLR) approach is based on the lattice structure of the discrete points. The base matrix $\boldsymbol{B}$ derived from $\boldsymbol{G}_\mathrm{d}$ is not an optimal representation of this structure for our purpose [7]. A representation $\boldsymbol{L}$ of $\boldsymbol{B}$ with preferably short base vectors is called *reduced* [9, 10]. It can be used to find an approximation of the transmitted lattice point:

$$\hat{\boldsymbol{y}}_{\mathrm{d}\,\mathrm{ZFLR}} = \left\lceil \boldsymbol{z}_\mathrm{d}{\cdot}\boldsymbol{L}^{-1} \right\rfloor{\cdot}\boldsymbol{L}. \tag{9}$$

Applying the rounding operation $\lceil\cdot\rfloor$ yields an estimate of the valid discrete part, because valid lattice points have integer entries in $\tilde{\boldsymbol{s}}$ according to (6). In contrast to the DML approach, the decision regions are parallelotopes regardless of the true Voronoi regions, and invalid lattice points outside of the modulo cube are also considered. However, this approach still yields acceptable results while having a very low computational complexity.

A detailed description of the basics and the different decoders can be found in [7].

4. LATTICE ML (LML) DECODER

The discrete part of the DML decoding approach described in Section 3.2a basically is a lattice quantization with a lattice that is limited to the projection of the modulo cube. In this section, an algorithm with reduced computational complexity is presented. It does not take this limitation into account, so that a generic lattice quantizer could be used. However, most lattice quantizers do not work for arbitrary lattices, but only for special lattices that are well-known [11–17]. Therefore, a method for quantization (i.e., maximum likelihood decoding of the discrete part) using arbitrary lattices with base matrix $\boldsymbol{L}$ is developed in this section. Similar approaches are described in [18–20].

4.1. Decision Regions: Zero Forcing vs. ML

As shown in Fig. 4, the decision regions of the Zero Forcing decoder (with lattice reduction, as in Section 3.2b) share a large central region with the Maximum Likelihood decision regions. Therefore, the result $\hat{\boldsymbol{y}}_{\mathrm{d}\,\mathrm{ZFLR}}$ from (9) of the ZF decoder can be used as a first approximation of the resulting lattice point estimate $\hat{\boldsymbol{y}}_\mathrm{d}$.

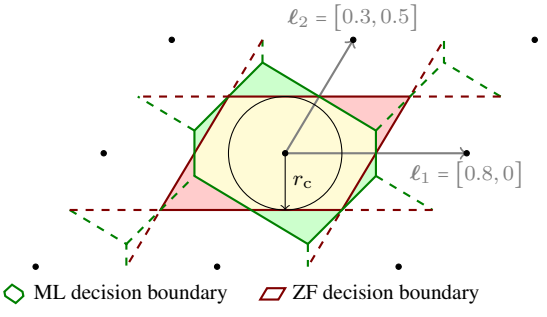


Fig. 4. ML and ZF decision regions and the radius r_c (see Section 4.3) for $\boldsymbol{L} = \begin{bmatrix}\boldsymbol{\ell}_1 \\ \boldsymbol{\ell}_2\end{bmatrix} = \begin{bmatrix}0.8 & 0 \\ 0.3 & 0.5\end{bmatrix}$, i.e., $D = N - M = 2$. Dots: lattice points.

4.2. Decoding of the Estimation Offset

In order to refine the estimation from Zero Forcing to Maximum Likelihood, the preliminarily decoded lattice point is subtracted from the received point and thus regarded as the center. This way, only the region inside its Voronoi region $[12, 21]$ has to be considered. Therefore, the point that remains to be decoded is (cf. Fig. 5)

$$\boldsymbol{e} = \boldsymbol{z}_\mathrm{d} - \hat{\boldsymbol{y}}_{\mathrm{d}\mathrm{ZFLR}}. \tag{10}$$

This estimation offset $\boldsymbol{e}$ is the current (i.e., ZF) estimation of the discrete noise (i.e., $\boldsymbol{e} = \boldsymbol{n}\boldsymbol{G}_\mathrm{d}$ for correct decoding $\hat{\boldsymbol{y}}_{\mathrm{d}\mathrm{ZFLR}} = \boldsymbol{y}_\mathrm{d}$).

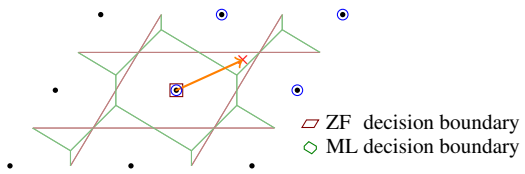


Fig. 5. Lattice points •, received point $\boldsymbol{z}_\mathrm{d}$ ×, estimation offset $\boldsymbol{e}$ →, ZF decoded point $\hat{\boldsymbol{y}}_{\mathrm{d}\mathrm{ZFLR}}$ □, candidates $\mathcal{C}$ for ML ○.

Depending on the orientation of $\boldsymbol{e}$, different candidates

$$\mathcal{C} = \left\{ \hat{\boldsymbol{y}}_{\mathrm{d}\mathrm{ZFLR}} + \boldsymbol{q}\cdot\boldsymbol{\mathcal{D}}\boldsymbol{L} \;\middle|\; \boldsymbol{q} \in \{0,1\}^D \right\} \tag{11}$$

for the final ML estimation are examined, where the orientation of the base vectors is determined by $\boldsymbol{\mathcal{D}} = \mathrm{diag}(\mathrm{sign}(\boldsymbol{e}\cdot\boldsymbol{L}^{-1}))$, and $\boldsymbol{q}$ "activates" and "deactivates" the base vectors. These candidates are all combinations of base vectors pointing roughly in the same direction as $\boldsymbol{e}$ (from the ZF estimate $\hat{\boldsymbol{y}}_{\mathrm{d}\mathrm{ZFLR}}$). The lattice points and candidates for an exemplary received vector are shown in Fig. 5.

Finally, the ML decision only has to be made among those $|\mathcal{C}| = 2^D$ candidates:

$$\hat{\boldsymbol{y}}_{\mathrm{d}\mathrm{LML}} = \underset{\boldsymbol{y}_\mathrm{d}\in\mathcal{C}}{\arg\min} \left\| \boldsymbol{z}_\mathrm{d} - \boldsymbol{y}_\mathrm{d} \right\|. \tag{12}$$

This is the discrete part of the proposed Lattice ML (LML) decoder.

Fig. 6 shows a block diagram of the Lattice ML decoder.

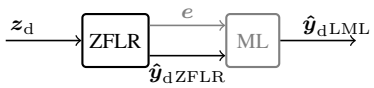


Fig. 6. Block diagram of the Lattice ML (LML) decoder.

4.3. Further Complexity Reduction by Radius Pre-Check

Fig. 4 also shows the radius r_c of the circle inside which both decision regions map to the same point. If the estimation offset $\boldsymbol{e}$ is inside this circle ($\|\boldsymbol{e}\| < r_\mathrm{c}$), no further decoding is needed and the previous step from Section 4.2 can be skipped (by leaving out (11) and (12) or the gray part in Fig. 6), resulting in a decreased decoding complexity. This happens when the noise is small,[2] so that $\|\boldsymbol{e}\| = \|\boldsymbol{n}\boldsymbol{G}_\mathrm{d}\| < r_\mathrm{c}$.

When the channel is good, $\|\boldsymbol{e}\| < r_\mathrm{c}$ holds for most transmitted code words, so that in this case the decoding complexity is not significantly higher than that of the ZF decoder.

This radius can be determined by

$$r_\mathrm{c} = \min_{i\in\mathbb{N}\le D} \frac{1}{2} \left| \ker(\boldsymbol{\ell}_j \forall j \neq i) \cdot \boldsymbol{\ell}_i \right|, \tag{13}$$

where $\boldsymbol{\ell}_i$, $\boldsymbol{\ell}_j$ are the base vectors (rows of $\boldsymbol{L}$) and ker is the kernel. The vector $\boldsymbol{k}_i = \ker(\boldsymbol{\ell}_j \forall j \neq i)$ is a single vector of length 1 that is orthogonal to all base vectors except $\boldsymbol{\ell}_i$, and thus orthogonal to the ZF decision boundary (hyper-)plane at the center of $\boldsymbol{\ell}_i$. The inner product $\boldsymbol{k}_i \cdot \boldsymbol{\ell}_i$ of this kernel vector with the base vector $\boldsymbol{\ell}_i$ is the length of the projection, as $\|\boldsymbol{k}\| = 1$. Among the D candidates, the minimum is taken.

[2] Or when the noise is large enough to "move" the code word into another (neighboring) decision region.

4.4. Simulation Results Lattice ML vs. ZFLR/DML

Fig. 7 shows simulation results[3] comparing the Lattice ML decoder with the standard DML decoder and the ZFLR decoder.

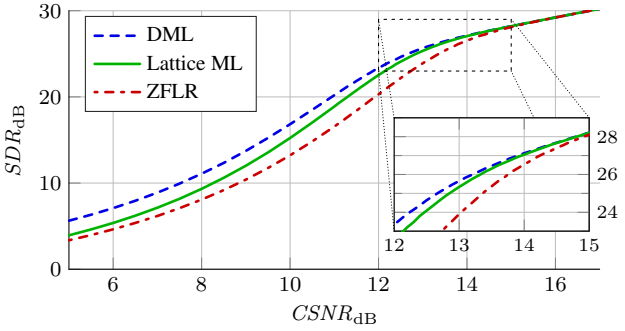
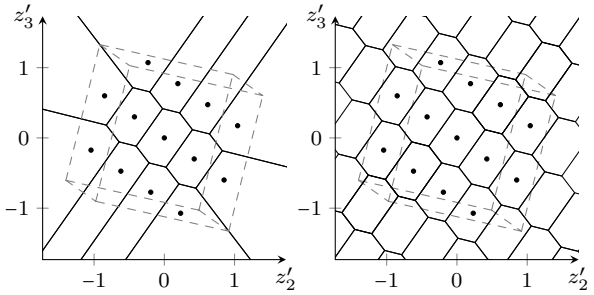


Fig. 7. Simulation results for different decoders, $\boldsymbol{A} = [1 \;\; 2 \;\; 4]$.

In the region with very high CSNR, all decoders show the same performance: the discrete part $\boldsymbol{z}_\mathrm{d}$ (cf. Fig. 8) is decoded correctly for (nearly) all code words (i.e., with significantly low error probability), and the continuous part is decoded in the same way for all decoders. However, this saturation is reached earlier (in terms of CSNR) by the LML and DML decoders than by the ZFLR decoder because of the optimal decision regions in the discrete part, although the LML decoder has a lower computational complexity.



(a) DML decision regions (b) Lattice ML decision regions

Fig. 8. Decision regions of the DML and Lattice ML decoders, $\boldsymbol{A} = [1 \;\; 2.5 \;\; 10]$. The image shows a projection on the discrete dimensions ($\boldsymbol{z}\boldsymbol{G}_\mathrm{d} = [z'_2 \;\; z'_3]$). Dashed: Projection of modulo cube. Dots: valid lattice points.

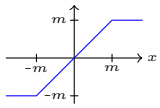
For a very low channel quality, however, the SDR of the Lattice ML decoder converges to the low SDR of the ZFLR decoder, because both decoders consider non-existent discrete lattice points outside of the modulo cube (Fig. 8b, in contrast to Fig. 8a). By applying the pre- and post-processing which will be presented in Section 5, this effect can be mitigated easily. It could be avoided completely by checking whether a discrete point is inside the modulo cube and, if so, applying a fallback solution (e.g., conventional DML decoding) – but this would significantly increase the computational complexity.

[3] All simulations in this paper have been conducted with the elements of $\boldsymbol{u}$ being independent and uniformly distributed between $-U = -1$ and $U = 1$.

5. PRE- AND POST-PROCESSING

Pre- and post-processing can be applied independently of the used decoder type to increase the quality of the decoded signal. For both proposed methods, a saturation function

$$\mathrm{sat}_m(x) = \begin{cases} -m & \text{for } x \le -m \\ x & \text{for } -m < x < m \\ m & \text{for } m \le x \end{cases} \qquad (14)$$



is needed, which limits the absolute of its input x to a given value m.

Fig. 9 shows a block diagram of both methods that will be explained in the next sections.

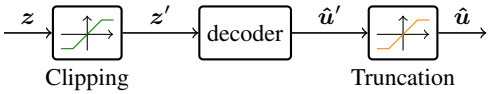


Fig. 9. Block diagrams of pre- and post-processing.

5.1. Clipping

All valid code words $\boldsymbol{y}$ are inside the modulo cube, i.e., $|y_i| \le m \ \ \forall i$. Therefore, the received values $\boldsymbol{z}$ can be limited to this cube: $\boldsymbol{z}' = \mathrm{sat}_m(\boldsymbol{z})$.

Fig. 10b shows the decoded values $\hat{\boldsymbol{u}}$ (as color) of a 1×2 code when clipping is applied. The values outside the modulo cube are mapped onto the nearest edge (or corner) before the actual decoding is performed.

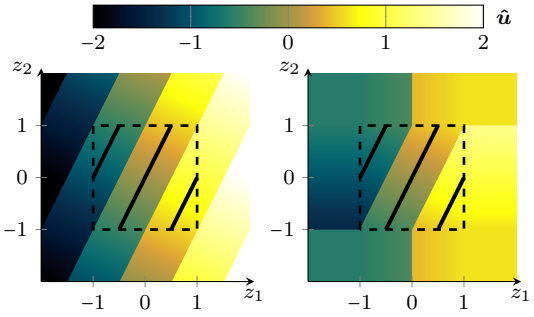


Fig. 10. ZF decoding of the $\boldsymbol{A} = [1 \ \ 2]$ code without and with clipping. Lines: valid code words. Color: decoded value.

5.2. Truncation

When the channel quality is good (i.e., the noise is small), most of the values $\hat{\boldsymbol{u}}'$ with conventional decoding are within the valid range of $\boldsymbol{u}$. However, if the channel quality is bad, the values might get very large ($\boldsymbol{z} \approx \boldsymbol{n}$, $\|\boldsymbol{z}\| \gg \|\boldsymbol{u}\|$ for $CSNR \ll 1$).

Limiting the elements of $\hat{\boldsymbol{u}}$ to the maximum possible absolute value U of $\boldsymbol{u}$, i.e.,

$$\hat{\boldsymbol{u}} = \mathrm{sat}_U(\hat{\boldsymbol{u}}') \quad \text{so that} \quad \|\hat{\boldsymbol{u}}\|_\infty \le U, \qquad (15)$$

(see Fig. 9) leads to a significantly increased SDR especially for very bad channel conditions.

By limiting the elements of $\hat{\boldsymbol{u}}$, the decoding error $\boldsymbol{u} - \hat{\boldsymbol{u}}$ is also limited: $\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^2 \le M(2U)^2$, and $\mathrm{E}\left\{\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^2\right\} \le 4MU^2$ (which is not a tight bound). Thus, the SDR is lower bounded to

$$SDR \overset{(5)}{=} \frac{M\sigma_u^2}{\mathrm{E}\left\{\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^2\right\}} \ge \frac{\sigma_u^2}{4U^2}. \qquad (16)$$

For uniformly distributed $\boldsymbol{u}$ it can be shown that $SDR \ge 1/4 \,\hat{=}\, -6\,\mathrm{dB}$.

Fig. 11 shows the influence of truncation on the decoded values $\hat{\boldsymbol{u}}$ for $U = m$. The extreme values with $|\hat{u}| > U \overset{\text{here}}{=} 1$ are avoided.

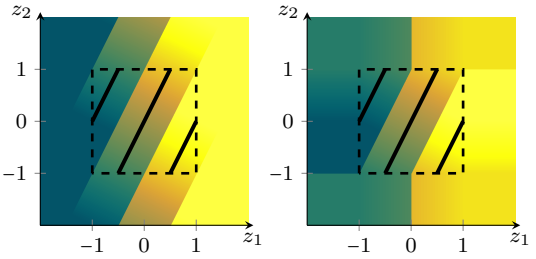


Fig. 11. ZF decoding of the $\boldsymbol{A} = [1 \ \ 2]$ code with truncation.

5.3. Simulation Results with Truncation and Clipping

Fig. 12 shows that the LML decoder nearly achieves DML performance when clipping and truncation is applied.

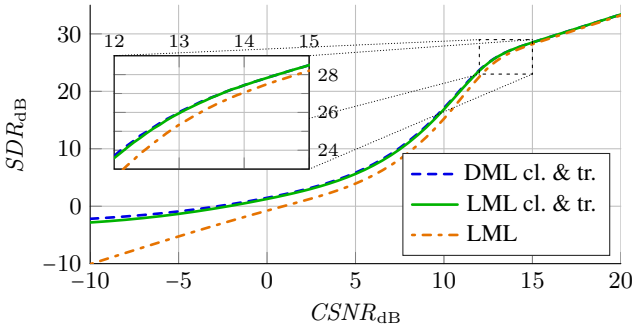


Fig. 12. Simulation results for $\boldsymbol{A} = [1 \ \ 2 \ \ 4]$ (cf. Fig. 7).

Without pre- or post-processing, the entries in $\hat{\boldsymbol{u}}$ can become arbitrarily large (and, thus, arbitrarily wrong), resulting in a very low SDR. By limiting $\boldsymbol{z}$ (clipping) or $\hat{\boldsymbol{u}}$ (truncation), the SDR can be lower bounded, as shown in Section 5.2. When AMB codes are used for speech or audio transmission, this avoids very loud and annoying errors, especially in very bad channel conditions.

6. CONCLUSION

A computationally low-complexity approximation of the discrete maximum likelihood (DML) decoder for AMB codes, the Lattice ML decoder, was derived in Section 4. It uses the low-complexity Zero Forcing decoder as an approximation which is then refined by taking the neighboring lattice points into account. This check can be skipped if the estimation offset is small, leading to a further decrease in complexity. Except for including invalid lattice points outside the modulo cube, this decoder achieves maximum likelihood performance in the discrete part with low computational complexity.

Section 5 contains two simple methods (clipping and truncation) to further increase the SDR of the decoded signals. These methods mitigate the effect that the Lattice ML decoder chooses invalid lattice points under some circumstances, so that the Lattice ML decoder (nearly) achieves DML results with a significantly reduced complexity.

AMB codes are a quantizer-free alternative for the low-delay transmission of time-discrete analog signals, like sampled speech, audio, or video. They can be used, e.g., in wireless microphones, loudspeakers, or hearing aids. The proposed methods of Lattice ML decoding and pre-/post-processing can be used for faster and better decoding, respectively.

7. REFERENCES

[1] Mikael Skoglund, Nam Phamdo, and Fady Alajaji, "Design and performance of VQ-based hybrid digital-analog joint source-channel codes," *IEEE Transactions on Information Theory*, vol. 48, no. 3, pp. 708–720, Mar. 2002.

[2] Udar Mittal and Nam Phamdo, "Hybrid digital-analog (HDA) joint source-channel codes for broadcasting and robust communications," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1082–1102, May 2002.

[3] Matthias Rüngeler and Peter Vary, "Surpassing purely digital transmission: A simplified design of hybrid digital analog codes," in *9th International ITG Conference on Systems, Communications and Coding (SCC'2013)*, Munich, Germany, Jan. 2013.

[4] Matthias Rüngeler, Johannes Bunte, and Peter Vary, "Design and evaluation of hybrid digital-analog transmission outperforming purely digital concepts," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 3983–3996, Nov. 2014.

[5] Jack Keil Wolf, "Redundancy, the discrete fourier transform, and impulse noise cancellation," *IEEE Transactions on Communications*, vol. 31, no. 3, pp. 458–461, Mar. 1983.

[6] Werner Henkel, "Multiple error correction with analog codes," *AAECC-6, Rome and Springer Lecture Notes in Computer Science*, vol. 357, pp. 239–249, 1989.

[7] Tim Schmitz, Matthias Rüngeler, and Peter Vary, "Analysis of analog modulo block codes," in *10th International ITG Conference on Systems, Communications and Coding (SCC'2015)*, Hamburg, Germany, Feb. 2015.

[8] Matthias Rüngeler, Birgit Schotsch, and Peter Vary, "Properties and performance bounds of linear analog block codes," in *Conference Record of Asilomar Conference on Signals, Systems, and Computers (ACSSC)*. IEEE, Nov. 2009, pp. 962–966.

[9] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.

[10] Dirk Wübben, Dominik Seethaler, Joakim Jaldén, and Gerald Matz, "Lattice reduction: A survey with applications in wireless communications," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 70–91, May 2011.

[11] John H. Conway and N. J. A. Sloane, "Fast quantizing and decoding and algorithms for lattice quantizers and codes," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 227–232, Mar. 1982.

[12] John H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 211–226, Mar. 1982.

[13] Allen Gersho, "On the structure of vector quantizers," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 157–166, Mar. 1982.

[14] John H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Transactions on Information Theory*, vol. 29, no. 6, pp. 820–824, Nov. 1983.

[15] Thomas R. Fischer, "A pyramid vector quantizer," *IEEE Transactions on Information Theory*, vol. 32, no. 4, pp. 568–583, July 1986.

[16] John H. Conway and N. J. A. Sloane, *Sphere packings, lattices, and groups*, Springer-Verlag, New York, 1988.

[17] Martin C. Rost and K. Sayood, "The root lattices as low bit rate vector quantizers," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1053–1058, Sept. 1988.

[18] Khalid Sayood, Jerry D. Gibson, and Martin C. Rost, "An algorithm for uniform vector quantizer design," *IEEE Transactions on Information Theory*, vol. 30, no. 6, pp. 805–814, Nov. 1984.

[19] Khalid Sayood and Steven J. Blankenau, "A fast quantization algorithm for lattice quantizer design," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP-88)*, Apr. 1988, vol. 2, pp. 1168–1171.

[20] E. Viterbo and E. Biglieri, "A universal decoding algorithm for lattice codes," in *14° Colloque sur le traitement du signal et des images*, France, 1993, Groupe d'Etudes du Traitement du Signal et des Images (GRETSI).

[21] Georgy Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs.," *Journal für die reine und angewandte Mathematik*, vol. 134, pp. 198–287, 1908.