Analysis of Analog Modulo Block Codes

Tim Schmitz, Matthias Rüngeler, Peter Vary
Institute of Communication Systems and Data Processing (ind)
RWTH Aachen University, Germany
{schmitz|ruengeler|vary}@ind.rwth-aachen.de

***Abstract*—Digital transmission of analog speech, audio, or video requires source coding which introduces unavoidable quantization errors. The bit stream produced by the source encoder needs to be protected against transmission errors by channel coding. The split of a given gross bit rate between source and channel coding is a compromise taking into account the design target of the worst case channel. Thus, even in clear channel conditions the quality of the decoded source signal is limited because of the irreversible quantization errors.**

Analog transmission systems with discrete-time and quasi-continuous-amplitude encoding and decoding do not show this saturation effect. Instead, their performance improves for increasing channel qualities. Examples are Linear Analog Block Codes, which show a poor overall performance, and (nonlinear) Archimedes spirals, which are just suited for the case of transmitting one source symbol with two channel uses.

Here, Analog Modulo Block Codes are proposed, which employ the matrix multiplication used in Linear Analog Block Codes while adding a modulo operation as nonlinearity. These codes combine the advantages of outperforming all purely digital codes in very good channel conditions (because of the absent quantization error) and requiring less energy per symbol than Linear Analog Block Codes (because of the modulo operation). Two different types of decoders are presented, and the results are evaluated and discussed.

I. Introduction

Digital transmission systems have been proven to be very good if the channel code is designed for the correct channel quality. However, the most prominent disadvantage is the irreversible quantization error that is induced by the source encoder. One way to circumvent this disadvantage are hybrid digital-analog (HDA) systems [1] which transmit the quantization error as an analog channel use.

Alternatively, the quantization can be avoided completely by applying a continuous-amplitude, discrete-time channel code. It has been shown in [2] that the simple multiplication of an analog information vector with a matrix (Linear Analog Block Code, LABC) does not improve the transmission quality at all. On the one hand, the matrix stretches the code symbols, which is why the channel noise has a smaller effect. On the other hand, the needed transmission energy is increased by the same amount, thus relativizing this effect.

Most digital codes operate on numbers from a Galois field and can be described using integers and modulo arithmetic. The idea of using modulo arithmetic can be transferred to LABCs as well. By applying a modulo operation after the matrix multiplication, the energy of each coded symbol is limited and therefore reduced. Thus, a coding gain can be obtained.

These codes, called Analog Modulo Block Codes or AMB codes, are introduced in this paper. They may be applied in low-delay applications where continuous-amplitude source symbols are transmitted over channels having a better-than-assumed channel quality at particular times or locations. This includes broadcast scenarios with the lack of feedback channels or unicast systems with changing channel quality. Also systems with low-complexity transmitters not designed to adapt to changing channels, e.g., transmitters used for wireless sensors, microphones or headsets, are a possible field of application.

Although the discrete-time Analog Modulo Block Codes use continuous-amplitude operations, they can be implemented with digital signal processing, while only the channel remains analog. The encoding (and decoding) can be done on a digital signal processor with analog-to-digital (A/D) and digital-to-analog (D/A) converters that have a sufficiently high resolution. If the induced quantization noise of the A/D-converter is small compared to the channel noise the system can be seen as an analog system. Therefore, the A/D-conversion is not considered in this paper.

The paper is organized as follows. In Section II the system model is introduced. The AMB codes are analyzed in Section III, methods for decoding are evaluated in Section IV, and results are shown in Section V. Finally, the work is concluded in Section VI.

II. System Model

The transmission system is depicted in Fig. 1.

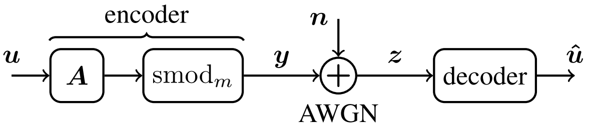


Figure 1. Block diagram of the transmission system.

A source vector $\boldsymbol{u} \in \mathbb{R}^M$ is considered with elements u_i, $-m \leq u_i < m$. Encoding with an Analog Modulo Block Code (AMB code) is done by multiplying this vector[1] with a matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ and then element-wise applying a modified (symmetric) modulo operation

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

which maps the input symbols onto the range $(-m, +m)$ as visualized in Fig. 2. The value of m can be arbitrarily

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

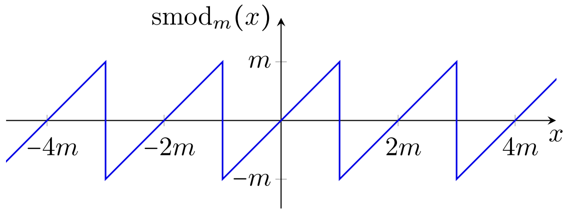


Figure 2. Modified (symmetric) modulo function.

chosen without any effect on the performance, provided that the matrix $\boldsymbol{A}$ is scaled accordingly. We assume $m = 1$ for all codes, but explicitly mention m in the equations in order to keep track of its influence. As the encoding maps M source symbols onto N channel symbols, the code rate is $r = \frac{M}{N}$.

The encoder can be summarized as:

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

with the *jump vector*

$$\boldsymbol{s} = -\left\lceil \frac{\boldsymbol{u}\boldsymbol{A}}{2m} \right\rfloor \in \mathbb{Z}^N \tag{3}$$

coming from the modulo operation. The notation $\lceil x \rfloor = \lfloor x + \frac{1}{2} \rfloor$ denotes rounding element-wise to the nearest integer.

For simplicity, we focus on *systematic* AMB codes with

$$\boldsymbol{A} = \begin{bmatrix} \mathbb{1} & \tilde{\boldsymbol{A}} \end{bmatrix} \tag{4}$$

whose code words contain the information words because of the $(M \times M)$ identity matrix $\mathbb{1}$, assuming that the elements of $\boldsymbol{u}$ are limited between $-m$ and $+m$.

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

The received vector

$$\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{n} \tag{5}$$

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

In order to evaluate the performance of the code we need an expression for the channel quality, i.e., the signal-to-noise ratio on the channel

$$CSNR = \frac{\mathrm{E}\left\{\|\boldsymbol{y}\|^2\right\}}{\mathrm{E}\left\{\|\boldsymbol{n}\|^2\right\}} = \frac{N \cdot \sigma_y^2}{N \cdot \sigma_n^2} = \frac{\sigma_y^2}{\sigma_n^2}. \tag{6}$$

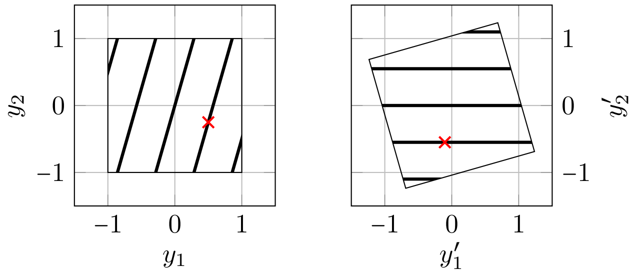
The quality of the transmission is denoted by the signal-to-distortion ratio

$$SDR = \frac{\mathrm{E}\left\{\|\boldsymbol{u}\|^2\right\}}{\mathrm{E}\left\{\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^2\right\}} = \frac{M \cdot \sigma_u^2}{M \cdot MSE} = \frac{\sigma_u^2}{MSE}. \tag{7}$$

MSE stands for *Mean Square Error*.

III. Analysis

Fig. 3a shows the valid code words of a simple systematic AMB code, where $\boldsymbol{A}$ is a 1×2 matrix. This means that each input symbol $\boldsymbol{u}$ (in this case scalar, as $M = 1$) is mapped to $N = 2$ output symbols $[y_1 \;\; y_2] = \boldsymbol{y}$.



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

Figure 3. Valid code words with $\boldsymbol{A} = [1 \;\; 3.5]$ and $m = 1$. As the code is systematic, it holds $y_1 = \boldsymbol{u}$. The box visualizes the limitations by the modulo operation and the limited range of input values (modulo cube). Cross: example.

Because of the restricted domain of the input symbols ($-m$ to m) and the modulo function, all code words are inside a (hyper-)cube with side length $2m$. This (hyper-)cube is called *modulo cube*.

The code words are located on distinguishable lines, which are in general parallel M-dimensional subspaces of the code space $\mathbb{R}^N$.

III-A. Rotation

Using an appropriate rotation $\boldsymbol{G} \in \mathbb{R}^{N \times N}$, the continuous M-dimensional subspaces (arms) can be aligned with the cartesian dimensions (y'_1 in Fig. 3b). This results in $D = N - M$ remaining dimensions which exhibit only discrete values and are orthogonal to the arms (y'_2 in Fig. 3b).

After the rotation, D columns of the code matrix $\boldsymbol{A}$ are zero:

$$\boldsymbol{A}\boldsymbol{G} = \boldsymbol{P} = \begin{bmatrix} \tilde{\boldsymbol{P}} & \mathbb{0}_{M \times D} \end{bmatrix}, \tag{8}$$

where $\mathbb{0}$ is an all-zero matrix[2] and $\boldsymbol{G}$ is orthogonal, i.e., $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} = \mathbb{1}$. If the input vector $\boldsymbol{u}$ is multiplied with $\boldsymbol{A}$ and then rotated by $\boldsymbol{G}$ without applying the modulo operation, the D last entries of the resulting vector $\boldsymbol{u}\boldsymbol{P}$ are zero. One possibility[3] to find $\boldsymbol{G}$ is the singular value decomposition [3]

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}, \text{ where } \boldsymbol{G} = \boldsymbol{V} \text{ and } \boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Sigma}. \tag{9}$$

Reformulating the rotation of the noisy received vector $\boldsymbol{z}$ from (5) with $\boldsymbol{G}$ using (2) leads to:

$$\boldsymbol{z}\boldsymbol{G} = \boldsymbol{u}\underbrace{\begin{bmatrix} \tilde{\boldsymbol{P}} & \mathbb{0} \end{bmatrix}}_{\boldsymbol{P}} + 2m\boldsymbol{s}\underbrace{\begin{bmatrix} \boldsymbol{G}_{\mathrm{c}} & \boldsymbol{G}_{\mathrm{d}} \end{bmatrix}}_{\boldsymbol{G}} + \boldsymbol{n}\underbrace{\begin{bmatrix} \boldsymbol{G}_{\mathrm{c}} & \boldsymbol{G}_{\mathrm{d}} \end{bmatrix}}_{\boldsymbol{G}}, \tag{10}$$

where $\boldsymbol{G}_{\mathrm{c}} \in \mathbb{R}^{N \times M}$ and $\boldsymbol{G}_{\mathrm{d}} \in \mathbb{R}^{N \times D}$ are submatrices of $\boldsymbol{G}$. The rotated vector $\boldsymbol{z}\boldsymbol{G} = \begin{bmatrix} \boldsymbol{z}_{\mathrm{c}} & \boldsymbol{z}_{\mathrm{d}} \end{bmatrix}$ can be split into a continuous part

$$\boldsymbol{z}_{\mathrm{c}} = \boldsymbol{z}\boldsymbol{G}_{\mathrm{c}} = \boldsymbol{u}\tilde{\boldsymbol{P}} + 2m\boldsymbol{s} \cdot \boldsymbol{G}_{\mathrm{c}} + \boldsymbol{n}\boldsymbol{G}_{\mathrm{c}} \quad \in \mathbb{R}^M \tag{11}$$

and a discrete part

$$\boldsymbol{z}_{\mathrm{d}} = \boldsymbol{z}\boldsymbol{G}_{\mathrm{d}} = 2m\boldsymbol{s} \cdot \boldsymbol{G}_{\mathrm{d}} + \boldsymbol{n}\boldsymbol{G}_{\mathrm{d}} \quad \in \mathbb{R}^D. \tag{12}$$

[2] Notation: $\mathbb{0}$ is an all-zero matrix whereas $\boldsymbol{0}$ is an all-zero vector.

[3] Note that there are many possibilities to find a valid $\boldsymbol{G}$, as it is not unique.

The equations for $\boldsymbol{z}$ also hold for $\boldsymbol{y}$ in case $\boldsymbol{n} = \boldsymbol{0}$. In Fig. 3b, y_1' corresponds to the continuous part and y_2' corresponds to the discrete part of the rotated code word $\boldsymbol{yG}$.

Example: As an example we encode the symbol $\boldsymbol{u} = 0.5$ with $\boldsymbol{A} = [1 \;\; 3.5]$ and $m = 1$. The first element $y_1 = 0.5$ of the code word equals the input symbol $\boldsymbol{u}$, because the code is systematic. The second element is $y_2 = \operatorname{smod}_m(0.5 \cdot 3.5) = -0.25$, and $\boldsymbol{s}$ is $[0 \;\; -1]$. The code word and its rotated version are shown in Fig. 3.

Rotating $\boldsymbol{y} = [y_1 \;\; y_2] = [0.5 \;\; -0.25]$ by 74.1° with the rotation matrix

$$\boldsymbol{G} \approx \begin{bmatrix} 0.274 & -0.962 \\ 0.962 & 0.274 \end{bmatrix} \quad \text{for } \boldsymbol{A} = [1 \;\; 3.5] \tag{13}$$

yields $\boldsymbol{yG} = [\boldsymbol{y}_\mathrm{c} \;\; \boldsymbol{y}_\mathrm{d}] \approx [-0.103 \;\; -0.549]$ (Fig. 3b).

It holds $\boldsymbol{P} \overset{(8)}{=} \boldsymbol{AG} \approx [3.64 \;\; 0]$, which means that the symbols have been stretched by 3.64 before the modulo operation.

III-B. Lattice Structure of the Discrete Part

The discrete part of the valid code words is

$$\boldsymbol{y}_\mathrm{d} = \boldsymbol{y} \cdot \boldsymbol{G}_\mathrm{d} = \boldsymbol{z}_\mathrm{d}\big|_{\boldsymbol{n}=\boldsymbol{0}} = 2m\boldsymbol{s} \cdot \boldsymbol{G}_\mathrm{d}. \tag{14}$$

For *systematic* codes, the first M entries of

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{0} & \tilde{\boldsymbol{s}} \end{bmatrix} \in \mathbb{Z}^N \tag{15}$$

are zero. In this case, the valid discrete points

$$\boldsymbol{y}_\mathrm{d} = \tilde{\boldsymbol{s}}\boldsymbol{B} \tag{16}$$

lie on a lattice [4] with the base matrix $\boldsymbol{B} \in \mathbb{R}^{D \times D}$, i.e., they are integer linear combinations of the row vectors in $\boldsymbol{B}$. The base matrix $\boldsymbol{B}$ is contained in the discrete rotation matrix

$$\boldsymbol{G}_\mathrm{d} = \begin{bmatrix} \boldsymbol{G}_{\mathrm{d,up}} \\ \frac{1}{2m}\boldsymbol{B} \end{bmatrix}. \tag{17}$$

The lattice structure is exploited in the design of a decoder (Section IV-B), see also Fig. 6.

The base matrix $\boldsymbol{B} \approx 0.274 \cdot 2m = 0.549$ of the example above ($\boldsymbol{A} = [1 \;\; 3.5]$) is a scalar, because the discrete part has only $D = 1$ dimension.

III-C. Performance bound

For decoders which decode the discrete and continuous part separately (Section IV: DML, ZF), the maximum coding gain is reached when the decoder always chooses the correct arm. In this case, only the error along the continuous dimensions (Section IV-C) has an influence on the total error. It depends on the singular values of $\boldsymbol{A}$, i.e., the diagonal elements of $\boldsymbol{\Sigma}$ from (9):

$$\boldsymbol{\Sigma} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbb{O} \end{bmatrix}, \quad \tilde{\boldsymbol{\Sigma}} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_M). \tag{18}$$

The SDR (derived in the Appendix) is bounded by

$$SDR \le \frac{\sigma_u^2}{\sigma_y^2} \cdot \frac{M}{\sum_{k=1}^{M} \frac{1}{\sigma_k^2}} \cdot CSNR = g_{\max} \cdot CSNR \tag{19}$$

for this type of decoders.

It can be easily imagined where the gain $g_{\max}$ comes from if the multiplication with the coding matrix is interpreted as stretching the code words along the information dimensions and then rotating it. The noise is then added onto the stretched symbols. During decoding, the (pseudo-)inverse of the matrix is applied. This implies squeezing with the reciprocal value of the stretching factor. In total, the information values have not been stretched at all, while the noise has been squeezed with the inverse of the stretching factors. This yields an improved ratio between information signal and distortion power (*SDR*).

If the values in $\boldsymbol{u}$ are uniformly distributed between $-m$ and m, σ_u^2 equals $\frac{m^2}{3}$. The code words $\boldsymbol{y}$ are then nearly[4] uniformly distributed between $-m$ and m for large entries ($\gg m$) in $\boldsymbol{A}$. In this case, the ratio $\frac{\sigma_u^2}{\sigma_y^2}$ is approximately 1.

A slightly higher performance can be achieved by an MMSE decoder (Section IV-D). However, typically, for high channel qualities this difference is very small.

IV. Decoding

In this section, three different methods for decoding AMB code words are described.

Two of them, the *Discrete Maximum Likelihood Decoder* (DML) and the *Zero Forcing Decoder* (ZF), decode the discrete part (find $\hat{\boldsymbol{s}}$ from $\boldsymbol{z}$) and the continuous part (find $\hat{\boldsymbol{u}}$ from $\boldsymbol{z}$ and $\hat{\boldsymbol{s}}$) of systematic codes separately (Fig. 4). Sections IV-A (DML) and IV-B (ZF) describe how they find the nearest valid lattice point $\boldsymbol{y}_\mathrm{d}$, specified by the jump vector $\boldsymbol{s}$. For systematic codes, it suffices to find the part of $\boldsymbol{s} \overset{(15)}{=} \begin{bmatrix} \boldsymbol{0} & \tilde{\boldsymbol{s}} \end{bmatrix}$ that is not zero.

A method for decoding the continuous part is shown in Section IV-C. Finally, the approach for an optimal decoder is shown in Section IV-D.

IV-A. Discrete Maximum Likelihood (DML) Decoder

The nearest valid discrete point $\hat{\boldsymbol{y}}_\mathrm{d}$ is found by evaluating

$$\hat{\boldsymbol{y}}_\mathrm{d} = \underset{\boldsymbol{y}_\mathrm{d}}{\arg\min} \left\| \boldsymbol{z}_\mathrm{d} - \boldsymbol{y}_\mathrm{d} \right\|^2 \tag{20}$$

in the discrete part, thus the name *Discrete Maximum Likelihood.* An estimate $\hat{\boldsymbol{s}}$ of the jump vector $\boldsymbol{s}$ from (3) can be calculated using $\hat{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{0} & \hat{\tilde{\boldsymbol{s}}} \end{bmatrix}$ as in (15) and

$$\hat{\tilde{\boldsymbol{s}}} = \hat{\boldsymbol{y}}_\mathrm{d} \boldsymbol{B}^{-1} \tag{21}$$

for systematic codes (see upper branch in Fig. 4).

Fig. 5 shows the discrete part of an AMB code with $\boldsymbol{A} = [1 \;\; 2.5 \;\; 10.5]$. As the valid code words are inside the modulo cube (Section III), the valid lattice points are inside the projection of the modulo cube. All possibly received channel symbols which are mapped to the same lattice point by the decoder are located in a common *decision region.* The lines denote the decision borders which enclose the decision regions. The DML decoder always chooses the lattice point which is closest to the received symbol. This leads to a regular structure in the middle of the figure and unlimited decision regions of the outer lattice points.

[4] They are exactly uniformly distributed if all entries of $\boldsymbol{A}$ are integers.

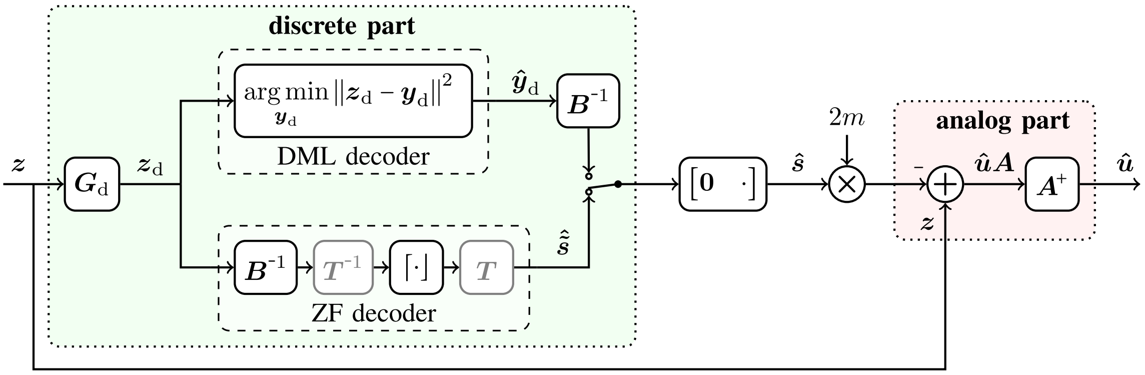


Figure 4. Block diagram of rotating decoders.

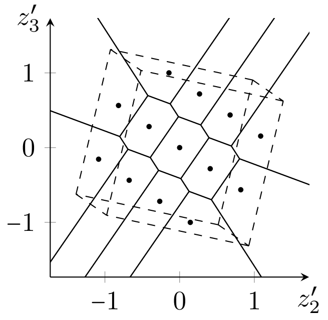
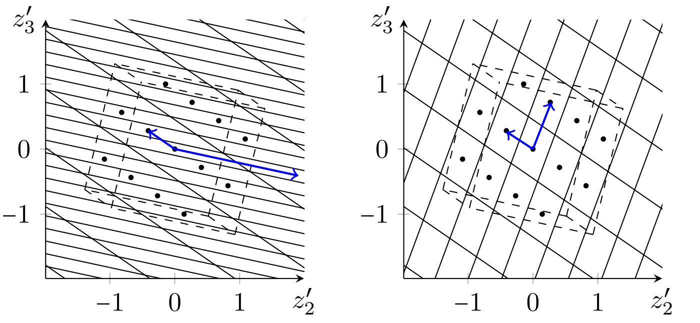


Figure 5. Decision regions of the DML decoder ($A = [1 \quad 2.5 \quad 10.5]$). The image shows a projection on the discrete dimensions ($zG_d = [z_2' \quad z_3']$). Dashed: Projection of modulo cube. Dots: valid code words (discrete part).

IV-B. Zero Forcing (ZF) Decoder

In contrast to the DML decoder from the previous section, which has to find the nearest arm by comparing the received point to all valid points in a naïve approach, the Zero Forcing (ZF) decoder exploits the lattice structure of the discrete part. The discrete part $y_d = yG_d$ of a valid code word lies on a lattice point. Therefore, $y_d B^{-1} \overset{(16)}{=} \tilde{s}$ contains integer values. The jump vector s can be estimated using (12) and (14) with

$$\hat{\tilde{s}} = \lceil z_d B^{-1} \rfloor = yG_d B^{-1} + \lceil nG_d B^{-1} \rfloor. \qquad (22)$$



(a) No lattice reduction

(b) With lattice reduction

Figure 6. Decision regions of ZF decoders ($A = [1 \quad 2.5 \quad 10.5]$). The images show a projection on the discrete dimensions ($zG_d = [z_2' \quad z_3']$). Dashed lines: modulo cube. Dots: valid code words (discrete part). Arrows: base vectors of the used lattice.

If every element of the noise term $nG_d B^{-1}$ is smaller than $\frac{1}{2}$, the effect of the noise is eliminated by the rounding function and thus, in this case, the estimation is correct ($\hat{\tilde{s}} = \tilde{s}$).

The decision regions of the Zero Forcing decoder can be derived from the lattice base vectors. Fig. 6a shows the decision regions of a code for which they are very narrow. This comes from the unequal lengths of the base vectors. In some regions, the decoder chooses a lattice point that is probably not the correct one, as it is not the nearest. These unfavorable decision regions lead to a performance that is far from a code's optimal performance (Fig. 9).

The decoding properties can be improved by *lattice reduction* (LR) [4]. Reducing the lattice means finding a more compact representation of the base using a suitable linear superposition of its vectors,

$$B_{\text{red}} = TB, \qquad (23)$$

where T is a unimodular matrix with integer entries. The new estimate of the non-zero part of the jump vector is

$$\hat{\tilde{s}} = \lceil z_d \cdot B^{-1} T^{-1} \rfloor \cdot T. \qquad (24)$$

A decoder that uses this estimate is called ZFLR decoder (Zero Forcing with Lattice Reduction). A block diagram of this decoder is depicted in the lower branch of Fig. 4. If no lattice reduction is desired, T can be set to $\mathbb{1}$. Fig. 6b shows the decision regions of this decoder. They are much more similar to those of the DML decoder and therefore yield better results (see Section V).

IV-C. Decoding of the Continuous Part

When an estimate of the transmitted lattice point and thus an estimate $\hat{s}$ of the jump vector s has been found, the continuous dimensions are decoded along the arm which corresponds to the found lattice point, using the pseudoinverse of A denoted by

$$A^+ = A^{\mathrm{T}} \cdot \left(AA^{\mathrm{T}}\right)^{-1} = G_c \tilde{\Sigma}^{-1} U^{\mathrm{T}}. \qquad (25)$$

Using the jump vector estimate $\hat{s} \overset{(15)}{=} [0 \quad \hat{\tilde{s}}]$,

$$\hat{u} = (z - 2m\hat{s}) \cdot A^+ \qquad (26)$$

yields an estimate of the information vector u. This estimate is correct for undisturbed transmission ($\boldsymbol{n} = \mathbf{0} \Rightarrow \boldsymbol{z} = \boldsymbol{y} \wedge \hat{\boldsymbol{s}} = \boldsymbol{s}$):

$$\hat{\boldsymbol{u}} = (\boldsymbol{y} - 2m\boldsymbol{s}) \cdot \boldsymbol{A}^{+} \overset{(2)}{=} \boldsymbol{u}\boldsymbol{A} \cdot \boldsymbol{A}^{+} \tag{27}$$

$$\overset{(25)}{=} \boldsymbol{u} \cdot \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} \cdot \left(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\right)^{-1} = \boldsymbol{u}. \tag{28}$$

Summary of the ZF decoder: Inserting (24) into (15) and (26) yields a summarized equation

$$\hat{\boldsymbol{u}} = \left(\boldsymbol{z} - 2m\left[\mathbf{0} \quad \left\lceil \boldsymbol{z}\boldsymbol{G}_{\mathrm{d}} \cdot \boldsymbol{B}^{-1}\boldsymbol{T}^{-1} \right\rfloor \cdot \boldsymbol{T}\right]\right) \cdot \boldsymbol{A}^{+} \tag{29}$$

for the ZF decoders.

Adding noise to the code word $\boldsymbol{y} = [0.5 \;\; -0.25]$ from the example above yields, e.g., $\boldsymbol{z} = [0.6 \;\; -0.3] \Rightarrow \boldsymbol{z}_{\mathrm{d}} \approx -0.659$ and $\hat{\tilde{\boldsymbol{s}}} = \left\lceil -0.659 \cdot 0.549^{-1} \right\rfloor = \lceil -1.2 \rfloor = -1 = \tilde{\boldsymbol{s}}$ with $\boldsymbol{T} = 1$ (because lattice reduction does not make sense for $D = 1$). The resulting $\hat{\boldsymbol{u}} \approx (\boldsymbol{z} - 2m[0 \;\; -1]) \cdot \begin{bmatrix} 0.0754 \\ 0.264 \end{bmatrix} \approx 0.494$ is very close to the correct source symbol $\boldsymbol{u} = 0.5$.

Summary of the DML decoder: The same can be done with (20), (21), (15) and (26) for the DML decoder:

$$\hat{\boldsymbol{u}} = \left(\boldsymbol{z} - 2m\left[\mathbf{0} \quad \left(\arg\min_{\boldsymbol{y}_{\mathrm{d}}} \|\boldsymbol{z}\boldsymbol{G}_{\mathrm{d}} - \boldsymbol{y}_{\mathrm{d}}\|^{2}\right) \cdot \boldsymbol{B}^{-1}\right]\right) \cdot \boldsymbol{A}^{+}. \tag{30}$$

IV-D. Minimum Mean Square Error (MMSE) decoder

In order to maximize the transmission quality SDR from (7), the expected mean square error $MSE = \frac{1}{M}\,\mathrm{E}\left\{\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^{2}\right\}$ is minimized. Thus, the optimal estimation is

$$\hat{\boldsymbol{u}} = \arg\min_{\hat{\boldsymbol{u}}} \mathrm{E}_{\boldsymbol{u}}\left\{\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^{2} \,\middle|\, \boldsymbol{z}\right\}. \tag{31}$$

Therefore, the probability density functions, which have to be known for this approach, are evaluated [5]:

$$\hat{\boldsymbol{u}} = \mathrm{E}\left\{\boldsymbol{u}|\boldsymbol{z}\right\} = \int \boldsymbol{u} \cdot \mathrm{p}\left(\boldsymbol{u}|\boldsymbol{z}\right) \mathrm{d}\boldsymbol{u}. \tag{32}$$

A decoder which implements this estimation is called Minimum Mean Square Error (MMSE) decoder. This decoder is preferably used as a reference, because its performance in terms of SDR cannot be exceeded by any other decoder. Yet, the computations that have to be performed for this decoder are very complex and the distribution of $\boldsymbol{u}$ as well as the channel conditions have to be known.

V. Simulation Results

V-A. Code Rate $\frac{1}{2}$, $M = 1$

The simulation results in Fig. 7 show the performance of several very simple codes with code matrix $\boldsymbol{A} = [1 \;\; t]$. The information symbols are uniformly distributed between −1 and 1 throughout this section. The encoded symbols are transmitted over an AWGN channel and decoded with the MMSE decoder. As a reference, the optimum performance theoretically attainable (OPTA) [2] for the same code rate $r = \frac{M}{N} = \frac{1}{2}$ is depicted as well. OPTA is obtained by equating the rate-distortion function with the channel capacity for partial or multiple channel uses. Another reference is the code with $\boldsymbol{A} = [1 \;\; 1]$. It corresponds to an LABC or repetition code, because the modulo operation has no effect.

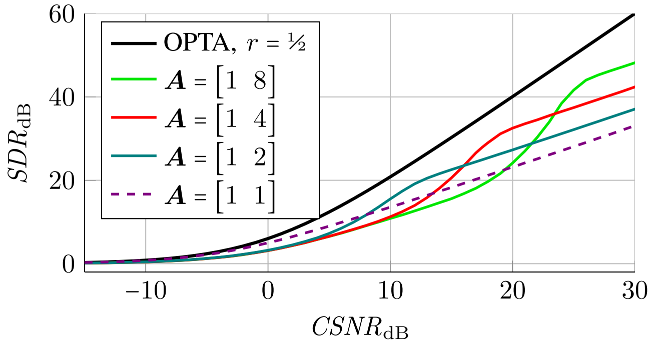


Figure 7. Simulation results for different AMB codes ($r = \frac{1}{2}$, $M = 1$, MMSE decoder).

The simulation results reveal the nature of analog codes: there is no quantization noise, and therefore an unlimited increase in performance for increasing channel conditions is observed. The DML performance of these codes is bounded by $SDR \le (1 + t^{2}) \cdot CSNR$ in this case.[5] This bound increases with t. On the other hand, a better channel is needed in order to achieve a gain from decoding at all for higher values of t. This tradeoff has to be evaluated individually for each transmission scenario.

V-B. Code Rate $\frac{1}{3}$, $M = 1$

Fig. 8 shows results of several codes with $\boldsymbol{A} = [1 \;\; a \;\; b]$ decoded with a DML decoder. All the codes have the same singular values (σ_i in (18)) and therefore the SDR is the same for very good channels (see (19)).

As an example, the code with $a = b$ ($\boldsymbol{A} = [1 \;\; 26.4 \;\; 26.4]$) performs very badly in contrast to the other codes, because all its lattice points are aligned in one row. Therefore, this code behaves similar to a code with $D = 1 \Rightarrow N = 2$, but still needs 50 % more transmission energy per source symbol.

As another example, the code with $\boldsymbol{A} = [1 \;\; 2 \;\; 37.3]$ has an intermediate “plateau”, because the lattice points are very close in one dimension (needing a high channel SNR to be distinguishable) but quite distant in another direction (which leads to basic decodability for medium channel SNRs).

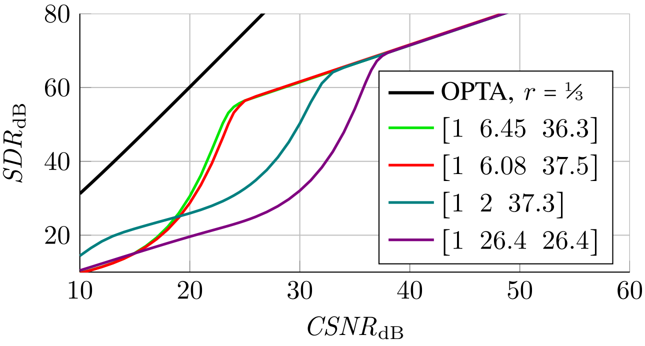


Figure 8. Comparison of codes with $\boldsymbol{A} \in \mathbb{R}^{1\times 3}$ ($r = \frac{1}{3}$, DML decoder).

[5] For $t \in \mathbb{N}$ and equally distributed $\boldsymbol{u}$ it holds $\sigma_{u}^{2} = \sigma_{y}^{2}$. The bound follows from (19) with $M = 1$ and $\sigma_{1}^{2} = 1 + t^{2}$.

The decision regions of an optimal code would be spherical, so that the (spherical [6]) noise always stays inside the decision region for a sufficiently high channel quality. Yet, this is only possible for an infinitely large number of dimensions D. For $D = 2$ dimensions, hexagonal decision regions are a good approximation.[6] By analyzing the lattice, it can be shown that the code with $\boldsymbol{A} = [1 \;\; 6.45 \;\; 36.3]$ has hexagonal DML decision regions. Therefore, this code performs better than the one with $\boldsymbol{A} = [1 \;\; 6.08 \;\; 37.5]$, which has square decision regions.

V-C. Decoder Performance

Fig. 9 shows a comparison of decoder performance. As expected, the MMSE decoder yields the best results for all channel qualities.

The performance of the DML decoder is very close to that of the MMSE decoder.

Like the other decoders, the Zero Forcing decoder reaches the SDR bound (19) for good channel conditions. However, the shape of the decoding regions of the discrete part of the ZF decoder (Fig. 6a) are far from the optimal shape (DML decoder) and thus the channel quality has to be almost 10 dB higher to find the correct lattice points.

The ZFLR decoder with lattice reduction exhibits decision regions which are much more similar to the optimal case, and thus the performance is much better and nearly as good as that achieved by the DML decoder. Simulations with different small codes have shown that the ZFLR decoder always performs only slightly worse than the DML decoder, whereas the ZF decoder without lattice reduction needs significantly higher channel qualities to decode the discrete part with a negligible error probability. Its performance loss (compared to ZFLR) is especially high if the base differs much from the reduced base, but vanishes for $\boldsymbol{T} = \mathbb{1}$.

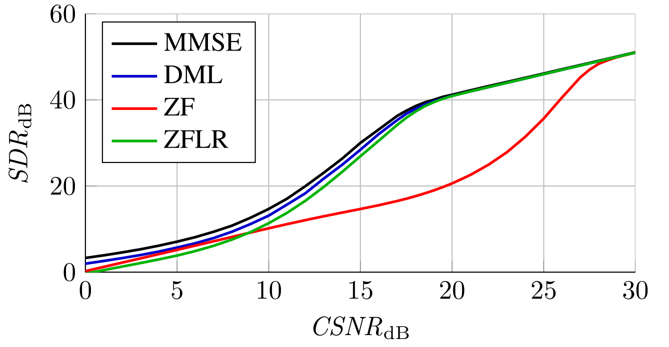


Figure 9. Simulation results for different decoders (AMB code with $\boldsymbol{A} = [1 \;\; 2.5 \;\; 10.5]$).

VI. Conclusion

In this paper, Analog Modulo Block Codes have been introduced. In contrast to conventional digital transmission systems which suffer from a saturation effect of the performance due to the uncorrectable quantization error, AMB codes use continuous-amplitude processing at the transmitter and the receiver and therefore improve their performance for increased channel qualities.

With an appropriate rotation of the AMB code words, the received symbols may be split into two parts: The discrete dimensions, which consist of a lattice, and continuous dimensions, which may be analyzed and decoded independently. Based on this, several types of decoders are proposed: a discrete maximum likelihood (DML) decoder, which achieves ML performance in the discrete part, and a zero forcing (ZF) decoder, which exploits the lattice structure of the code words. An enhanced zero forcing decoder with additional lattice reduction (ZFLR) almost achieves the performance of the DML decoder at a significantly lower complexity.

Appendix

Derivation of the Performance Bound (19)

The performance bound is derived for decoding of the analog part with the pseudoinverse of $\boldsymbol{A}$ as in Section IV-C.

Without loss of generality we choose $\boldsymbol{u} = \boldsymbol{0} \Rightarrow \boldsymbol{s} = \boldsymbol{0}$. The maximum performance is achieved if the decoder chooses the correct discrete point ($\hat{\boldsymbol{s}} = \boldsymbol{s}$). In this case, only the noise along the continuous dimensions has an influence on the estimation:

$$\hat{\boldsymbol{u}} \overset{(26)}{=} (\boldsymbol{z} - 2m\hat{\boldsymbol{s}}) \cdot \boldsymbol{A}^{+} \tag{33}$$

$$\overset{(5),(2)}{=} (\boldsymbol{u}\boldsymbol{A} + 2m\boldsymbol{s} + \boldsymbol{n} - 2m\hat{\boldsymbol{s}}) \cdot \boldsymbol{A}^{+} \tag{34}$$

$$= \boldsymbol{u} + \boldsymbol{n}\boldsymbol{A}^{+} \quad \Rightarrow \quad \hat{\boldsymbol{u}} - \boldsymbol{u} = \boldsymbol{n}\boldsymbol{A}^{+}. \tag{35}$$

With $\boldsymbol{A}^{+} = \boldsymbol{V}[\mathrm{diag}(\sigma_k^{-1}) \;\; \mathbb{O}]^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}$ [3] and $\boldsymbol{V}$, $\boldsymbol{U}$ being unitary (i.e., they do not change the length of a vector) it holds

$$\mathrm{E}\left\{\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2\right\} = \mathrm{E}\left\{\left\|\boldsymbol{n} \cdot \begin{bmatrix}\mathrm{diag}(\sigma_k^{-1}) \\ \mathbb{O}\end{bmatrix}\right\|^2\right\} = \sigma_n^2 \sum_{k=1}^{M} \sigma_k^{-2} \tag{36}$$

as the noise energy $\|\boldsymbol{n}\|^2$ is equally distributed among the elements of the noise vector $\boldsymbol{n}$ ($\|\boldsymbol{n}\|^2 = N \cdot \sigma_n^2$). In this case ($\hat{\boldsymbol{s}} = \boldsymbol{s}$) it holds

$$SDR \overset{(7)}{=} \frac{\mathrm{E}\left\{\|\boldsymbol{u}\|^2\right\}}{\mathrm{E}\left\{\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^2\right\}} = \frac{M\sigma_u^2}{\sigma_n^2 \sum_{k=1}^{M} \sigma_k^{-2}} \tag{37}$$

$$= \frac{\sigma_y^2}{\sigma_n^2} \frac{\sigma_u^2}{\sigma_y^2} \cdot \frac{M}{\sum_{k=1}^{M} \sigma_k^{-2}} \overset{(6)}{\Rightarrow} (19). \tag{38}$$

References

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

[2] M. Rüngeler, B. Schotsch, and P. 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.

[3] G. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[4] D. Wübben, D. Seethaler, J. Jaldén, and G. Matz, "Lattice reduction: A survey with applications in wireless communications," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 70–91, May 2011.

[5] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall Signal Processing Series, Apr. 1993.

[6] C. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, Jan 1949.

[6] However, other effects (considering the continuous dimensions) can have a more important influence on the performance of the overall system.