Codes over Gaussian integer rings

Farhad Ghaboussi
Department of Computer Science
University of Applied Sciences Constance
Email: farhad.ghaboussi@htwg-konstanz.de
Web: www.edc.in.htwg-konstanz.de

Jürgen Freudenberger
Department of Computer Science
University of Applied Sciences Constance, Germany
Email: jfreuden@htwg-konstanz.de
Web: www.edc.in.htwg-konstanz.de

***Abstract*—This work presents block codes over Gaussian integer rings. Rings of Gaussian integers extend the number of possible QAM signal constellations over Gaussian integer fields. Many well-known code constructions can be used for codes over Gaussian integer rings, e.g., the Plotkin construction or product codes. These codes enable low complexity soft decoding in the complex domain.**

***Index Terms*—Gaussian integers, Plotkin construction, product codes, soft input decoding**

I. Introduction

Gaussian integers are a subset of the complex numbers such that the real and imaginary parts are integers. Block codes over Gaussian integers were first studied by Huber in [1]. Huber also introduced the Mannheim distance as a performance measure for codes over Gaussian integers. Codes over Gaussian integers can be used for coding over two-dimensional signal spaces, e.g. using QAM signals. Similar code constructions were later considered in [2]. More recently, Gaussian integers were applied to construct space-time codes [3], [4].

Most of the mentioned code constructions are linear codes based on finite Gaussian integer fields which are constructed from primes p of the form $p \equiv 1 \bmod 4$. The number of signal points in these complex QAM constellations is therefore limited to prime numbers satisfying this condition, e.g. $p = 5, 13, 17, 29, \ldots$. Additionally, multiplicative groups were considered in [5].

In this work we consider code constructions over Gaussian integer rings. Such rings can be used to construct codes which are very similar to linear codes over Gaussian integer fields. Gaussian integer rings can be constructed for perfect squares. They have interesting algebraic properties. We show that simple codes can be constructed similar to the one Mannheim error correcting (OMEC) codes presented by Huber in [1], by building product codes as suggested by Elias [6] or using the so-called Plotkin construction [7]. The recursive Plotkin construction can also be exploited for low-complexity decoding [8], [10]. Similarly, the OMEC and product codes can be decoded using Chase-type algorithms [11].

II. Preliminaries

Gaussian integers are complex numbers such that the real and imaginary parts are integers. The modulo function $\mu(z)$ of a complex number z is defined as

$$\mu(z) = z \bmod \pi = z - \left[\frac{z\pi^*}{\pi \cdot \pi^*}\right] \cdot \pi\,, \tag{1}$$

where π^* is the conjugate of the complex number π. $[\cdot]$ denotes rounding to the closest Gaussian integer. That is, for a complex number $z = a + ib$, we have $[z] = [a] + i\,[b]$.

We use the Mannheim weight and Mannheim distance as introduced in [1]. Let the Mannheim weight of the complex number $z \in \mathcal{G}_m$ be defined as

$$wt_{\mathrm{M}}(z) = |\mathrm{Re}\,\{\mathrm{z}\}\,| + |\mathrm{Im}\,\{\mathrm{z}\}\,|\,, \tag{2}$$

then the Mannheim distance between two complex numbers y and z is defined as

$$d_{\mathrm{M}}(y,z) = wt_{\mathrm{M}}(\mu(z-y))\,. \tag{3}$$

Similarly, the Mannheim weight of the vector $\mathbf{z}$ is

$$wt_{\mathrm{M}}(\mathbf{z}) = \sum_i wt_{\mathrm{M}}(\mu(z_i)) \tag{4}$$

and the Mannheim distance for vectors is defined as

$$d_{\mathrm{M}}(\mathbf{y},\mathbf{z}) = wt_{\mathrm{M}}(\mu(\mathbf{z}-\mathbf{y}))\,. \tag{5}$$

The Mannheim distance defines a metric.

III. Rings of Gaussian integers

Most of the code constructions in [1]–[4] are linear codes based on finite Gaussian integer fields which are constructed from primes p of the form $p \equiv 1 \bmod 4$. For prime numbers $p = 4c + 1, c \in \mathbb{N}$ the Gaussian field structure is shown in [12].

We show in the following that the residue class ring of Gaussian integers arises from the residue class ring of certain integers with a unique quadratic decomposition.

Theorem 1: Given any non-prime integer $m \in \mathbb{N}$ with a unique decomposition

$$m = a^2 + b^2 = (a+ib)(a-ib) = \Pi \cdot \Pi^*; a, b \in \mathbb{Z}$$

there exists the residue class ring of Gaussian integers modulo Π

$$\mathcal{G}_m = \{0, z_1^c, z_2^c, \ldots, z_{m-1}^c\}$$

with elements

$$z_j^c := z_j - [\frac{z_j \cdot \Pi^*}{\Pi \cdot \Pi^*}] \cdot \Pi \tag{6}$$

where $[\cdot]$ denotes rounding to the next Gaussian integer.

Proof: $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$ is a residue class ring [9]. Equation (6) is an isomorphic map, because there exist an inverse map

$$z_j = (z_j^c \cdot s \cdot \Pi^* + z_j^{c*} \cdot t \cdot \Pi)\ mod\ m, \tag{7}$$

where $1 = s \cdot \Pi^* + t \cdot \Pi$. s and t can be calculated using the Euclidean algorithm [1].

Then for all $z_1, z_2 \in \mathbb{Z}_m$ and all $z_1^c, z_2^c \in \mathcal{G}_m$ we have the following ring homomorphism

$$\begin{aligned} z_1 + z_2 \; mod \; m \quad &\leftrightarrow \quad z_1^c + z_2^c \; mod \; \Pi \\ z_1 \cdot z_2 \; mod \; m \quad &\leftrightarrow \quad z_1^c \cdot z_2^c \; mod \; \Pi \end{aligned}$$

according to

$$\begin{aligned} z_1^c + z_2^c \; mod \; \Pi \quad &= \quad z_1^c + z_2^c - [\frac{(z_1 + z_2) \cdot \Pi^*}{\Pi \cdot \Pi^*}] \cdot \Pi \\ z_1^c \cdot z_2^c \; mod \; \Pi \quad &= \quad z_1^c \cdot z_2^c - [\frac{(z_1 \cdot z_2) \cdot \Pi^*}{\Pi \cdot \Pi^*}] \cdot \Pi. \end{aligned}$$

■

Example 1: For example, with $n = 25 = 4^2 + 3^2$ we can construct the complex residue ring $\mathcal{G}_{25}$ isomorph to the residue class ring $\mathbb{Z}_{25}$ according to $25 = (4 + 3i)(4 - 3i)$

$$\begin{aligned} \mathcal{G}_{25} \quad &= \quad \{z \bmod (4 + 3i), z \in \mathbb{Z}_{25}\} \\ &= \quad \{0, 1, 2, 3, -3i, -2 + i, -1 + i, i, 1 - i, \\ &\quad\; 2 + i, -1 - 2i, -2i, 1 - 2i, -1 + 2i, 2i, \\ &\quad\; 1 + 2i, -2 - i, -1 - i, -i, 1 - i, 2 - i, 3i, \\ &\quad\; -3, -2, -1\}. \qquad (8) \end{aligned}$$

This complex constellation is also depicted in Fig. 1. The complex residue class ring $\mathcal{G}_{25}$ is by definition an additive group of Gaussian integers and a monoid under multiplication. However, note that all elements of $\mathcal{G}_{25} \setminus \{0, 5, 10, 15, 20\} = \mathcal{G}_{25} \setminus \{0, (-2+i), (-1-2i), (1+2i), (2 - i)\}$ have a multiplicative inverse. Furthermore, similar to primitive elements of a finite field there exist primitive roots that generate the ring $\mathcal{G}_{25}$ up to the elements $\{5, 10, 15, 20\}$, i.e., the powers α^i of any element $\alpha \in \{2, 3, 8, 12, 13, 17, 22, 23\}$ generate the set $\mathcal{G}_{25} \setminus \{0, 5, 10, 15, 20\}$.

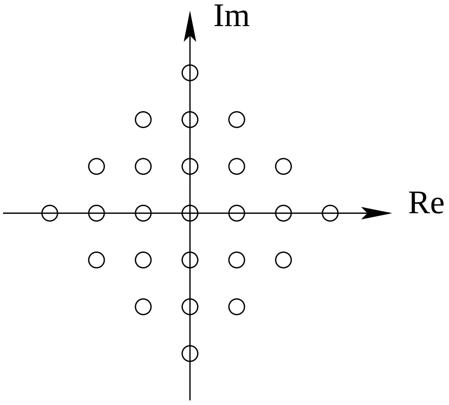

Fig. 1. Complex constellation of the Gaussian integer ring $\mathcal{G}_{25}$.

Theorem 1 requires an integer m with a unique decomposition, nevertheless there are non-prime integers with multiple decompositions like $65^2 = a_j^2 + b_j^2 = \Pi_j \cdot \Pi_j^*; j = 1, \ldots, 4$ for which one may generalize the theorem with multiple mappings to the Gaussian residue class ring corresponding to the different decompositions.

IV. Code construction

A code $\mathcal{C}$ of length n over the ring $\mathcal{G}_m$ is a set of codewords $\mathbf{v} = (v_0, \ldots, v_{n-1})$ with $v_i \in \mathcal{G}_m$. We first define some simple codes similar to codes over finite fields. Then we use these simple codes to construct more powerful codes based on well known code constructions. We only consider codes where the sum of two codewords is also a codeword. Hence, we have

$$d = \min_{\mathbf{v}', \mathbf{v}'' \in \mathcal{C}, \mathbf{v}' \neq \mathbf{v}''} d_{\mathrm{M}}(\mathbf{v}', \mathbf{v}'') = \min_{\mathbf{v} \in \mathcal{C}, \mathbf{v} \neq \mathbf{0}} wt_{\mathrm{M}}(\mathbf{v}) \quad (9)$$

for the minimum Mannheim distance d of the code $\mathcal{C}$.

a) OMEC codes: Many code constructions presented in [1] can be applied to Gaussian integer rings. Consider for example the following one Mannheim error correcting (OMEC) code. The parity check matrix of the code of length up to $n = p(p - 1)/4$ over the Gaussian residue class ring $\mathcal{G}_{p^2}$ is constructed by the elements generated by powers of a primitive root α, i. e.

$$\mathbf{H} = (\alpha^0, \alpha^1, \ldots, \alpha^{n-1}). \qquad (10)$$

Codewords are all vectors $\mathbf{v} = (v_0, v_1, ..., v_{n-1})$ with $v_i \in \mathcal{G}_{p^2}$ for which $\mathbf{H}\mathbf{v}^T = \mathbf{0}$. A systematic encoding for the information vector $\mathbf{u}$ is obtained by

$$\begin{aligned} \mathbf{v} \quad &= \quad (v_0, u_0, \ldots, u_{k-1}) \text{ with} \qquad (11) \\ v_0 \quad &= \quad -\alpha^1 u_0 - \alpha^2 u_1 - \ldots - \alpha^{n-1} u_{k-1}. \end{aligned}$$

Example 2: Using the ring $\mathcal{G}_{25}$ from example 1 with $p = 5$ we can construct a code of length $n = 5$ with parity check matrix

$$\mathbf{H} = (1, 1 + i, 2i, 1 - 2i, 3i)$$

where we have used the primitive root $\alpha = \mu(8) = 1 + i$. This code has minimum Mannheim distance $d = 3$ and is able to correct any Mannheim error of weight one, because any single error from $\{1, -1, i, -i\}$ will produce a different syndrome.

b) Encoding binary information vectors: The mapping of a binary information vector can be done by using the Horner algorithm. We demonstrate the encoding in the following example.

Example 3: The code from example 2 has dimension $k = n - 1 = 4$ and therefore has $m^k = 390625$ codewords. However, we can only map $\lfloor \log_2(m^k) \rfloor = 18$ bits to each codeword which results in a rate of $R = \frac{18}{5} = 3.6$ bits per symbol.

Let $\mathbf{u}_{(b)}$ be the information vector in base b. For example, we can interpret the vector

$$\mathbf{u}_{(2)} = (1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0)$$

as the binary representation of the integer

$$u_{(10)} = \sum_{j=0}^{17} u_j 2^j = 91149.$$

Using the Horner algorithm we can find the corresponding representation in base 25

$$\mathbf{u}_{(25)} = (24, 20, 20, 5).$$

Using (6) we can map the elements of this vector to the ring $\mathcal{G}_{25}$

$$\mathbf{u}_{(4+3i)} = (-1, 2 - i, 2 - i, -2 + i).$$

Finally, with (11) we obtain the codeword

$$\mathbf{v} = (1 + i, -1, 2 - i, 2 - i, -2 + i).$$

The encoding is systematic. Hence, using the inverse mapping in (7) we can calculate the integer representation $u_{(10)}$ of the information vector by $\sum_{j=0}^{3} v_j 25^j$ and the binary representation using the Horner scheme again.

More powerful codes can be constructed by building product codes or using the so-called Plotkin construction.

c) Product codes: The construction of product codes was suggested by Peter Elias [6]. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be (n_1, k_1, d_1) and (n_2, k_2, d_2) group codes over the Gaussian integer ring $\mathcal{G}_m$, respectively. We first encode k_2 codewords of the code $\mathcal{C}_1$ and store these codewords column wise into the first k_2 columns of a $(n_1 \times n_2)$-matrix. Then we use the code $\mathcal{C}_2$ n_1-times to encode each row of this matrix. The resulting code $\mathcal{C}$ has length $n = n_1 n_2$ and dimension $k = k_1 k_2$.

Theorem 2: A product code over a Gaussian integer ring has minimum Mannheim distance $d \geq d_1 d_2$.

Proof: According to (9) the minimum Mannheim distance of the product code is equivalent to the minimum weight of a non-zero codeword. A non-zero codeword has at least one non-zero column. This column is a codeword of the code $\mathcal{C}_1$ and has at least weight d_1. However, each non-zero element of this column results in a non-zero row. Each non-zero row is a codeword of the code $\mathcal{C}_2$ and has at least weight d_2. Hence, a non-zero codeword has at least d_1 non-zero rows each with minimum weight d_2. ■

Example 4: We can use the code from example 2 to construct a product code of length $n = 25$, dimension $k = 16$, and minimum Mannheim distance $d = 9$. This code has rate $R = \frac{75}{25} = 2.96$ bits per symbol.

d) Plotkin construction: Given two codes of length n, the Plotkin construction can be used to obtain a code of length $2n$ [7]. This construction works for linear and non-linear codes. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two block codes of length $n_1 = n_2$ over the Gaussian integer ring $\mathcal{G}_m$. We construct a code $\mathcal{C}$ by direct vector addition

$$\mathcal{C} = \{|\mathbf{v}'|\mathbf{v}' + \mathbf{v}''|, \mathbf{v}' \in \mathcal{C}_1, \mathbf{v}'' \in \mathcal{C}_2\} \qquad (12)$$

Then $\mathcal{C}$ is also a block code over $\mathcal{G}_m$. $\mathcal{C}$ has length $n = 2n_1$ and dimension $k = k_1 + k_2$, where k_1 and k_2 are the dimensions of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively.

Theorem 3: Code $\mathcal{C}$ resulting from the Plotkin construction has minimum Mannheim distance $d \geq \min\{2d_1, d_2\}$, where d_1 and d_2 are the distances of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively.
The proof is very similar to the proof of the Plotkin construction for binary codes in [12].

Example 5: We can use the code from example 2 as code $\mathcal{C}_1$ and a repetition code $\mathcal{C}_2$ of length $n_2 = n_1 = 5$ to construct a Plotkin code of length $n = 2n_1 = 10$. A repetition code of length n_2 is obtained by repeating the information symbol $u \in \mathcal{G}_m$ n_2-times. The code has dimension $k_2 = 1$ and minimum Mannheim distance $d_2 = n_2$. The resulting Plotkin code has dimension $k = 5$ and minimum Mannheim distance $d = 5$. Thus, this code has rate $R = \frac{23}{10} = 2.3$ bits per symbol.

V. Decoding

Let $\mathbf{r} = \mathbf{v} + \mathbf{e}$ denote the received vector where $\mathbf{e}$ is the error vector. We first consider hard input decoding, i.e., we assume that $e_i \in \mathcal{G}_m$. Reference [1] presents a simple algebraic decoding algorithm for the code from example 2. This code can correct any single error of Mannheim weight one. For the algebraic decoding we calculate the syndrome $s = \mathbf{H}\mathbf{r}^T$. We obtain the error location as $l = \log_\alpha s \, mod \, n$ and the error value $e_l = S\alpha^{-l}$.

a) Hard input decoding of Plotkin codes: Now consider the code resulting from the Plotkin construction as in example 5. The code has minimum Mannheim distance $d = 5$ and can correct any error up to Mannheim weight two using the following decoding procedure. We assume that the received vector is

$$\mathbf{r} = |\mathbf{r}'|\mathbf{r}''| = \mathbf{v} + \mathbf{e}$$

with $\mathbf{e} = |\mathbf{e}'|\mathbf{e}''|$, i.e., $\mathbf{r}'$ and $\mathbf{e}'$ denote the first halves of the received vector and the error vector, $\mathbf{r}''$ and $\mathbf{e}''$ denote the second halves, respectively.

Let t be the Mannheim weight of $\mathbf{e}$. First, we decode the vector $\mathbf{r}'' - \mathbf{r}' = \mathbf{v}'' + \mathbf{e}'' - \mathbf{e}'$ with respect to the repetition code $\mathcal{C}_2$ using majority logic decoding, i.e., selecting the symbol with the largest number of occurrences. Note that the vector $\mathbf{e}'' - \mathbf{e}'$ has Mannheim weight less or equal t. Hence, $\mathcal{C}_2$ corrects all errors with $t \leq \lfloor \frac{d}{2} \rfloor$. Let $\hat{\mathbf{v}}''$ denote the resulting estimate.

In the second step, we decode $\mathbf{r}'$ and $\mathbf{r}'' - \hat{\mathbf{v}}''$ with respect to $\mathcal{C}_1$ which results in two estimates $\hat{\mathbf{v}}'_1$ and $\hat{\mathbf{v}}'_2$, respectively. Note that either $\mathbf{e}'$ or $\mathbf{e}''$ has Mannheim weight less or equal $t/2$. Therefore, either $\hat{\mathbf{v}}'_1$ or $\hat{\mathbf{v}}'_2$ is equal to the transmitted codeword $\mathbf{v}' \in \mathcal{C}_1$ if $t \leq \lfloor \frac{d}{2} \rfloor$.

To obtain the final estimate $\hat{\mathbf{v}}$, we construct the codewords $|\hat{\mathbf{v}}'_1|\hat{\mathbf{v}}'_1 + \hat{\mathbf{v}}''|$, $|\hat{\mathbf{v}}'_2|\hat{\mathbf{v}}'_2 + \hat{\mathbf{v}}''|$ and select the one which is closest to the received vector $\mathbf{r}$.

b) Chase-type decoding: Chase algorithms [11] are sub-optimum decoding algorithms for linear binary block codes. This class of algorithms provides an efficient trade-off between error performance and decoding complexity. They are based on conventional algebraic decoders. Where the decoder generates a set of candidate codewords by flipping some bit positions (test patterns) previous to the algebraic decoding. The most likely codeword in this list is chosen as the final estimate.

In the following we discuss low-complexity soft input decoding procedures. In particular we show that the Chase approach can be applied to codes over Gaussian integers. Although the Chase-type algorithms are rather old, such decoders are still highly relevant (see e.g. [13]).

Let $\mathbf{r} = (r_0, \ldots, r_{n-1})$ be the received vector after transmission over an additive white Gaussian noise (AWGN) channel. $\hat{r}_i = [r_i]$ denotes the hard-decision for the i-th symbol, where $[\cdot]$ denotes rounding to the closest Gaussian integer.

A Chase decoder generates a list of possible codewords, which is used to calculate the estimated codeword by comparing each codeword in the list with the received vector $\mathbf{r}$. In the case of binary codes, the list of codewords is obtained by flipping bits in certain positions of the received sequence. In the case of codes over Gaussian integers, we add the symbols $\{1, -1, i, -i\}$ to the most unreliable symbol positions, i.e., to the positions with largest weight $wt_{\mathrm{M}}(\hat{r}_i - r_i)$.

Simulation results for the code from example 2 are depicted in Fig. 2. With hard input decoding the simple

OMEC code achieves a gain of approximately 2.5dB for a symbol error rate of 10^{-4}. Soft input decoding obtains an additional gain of 0.5dB with a list of 13 codewords.

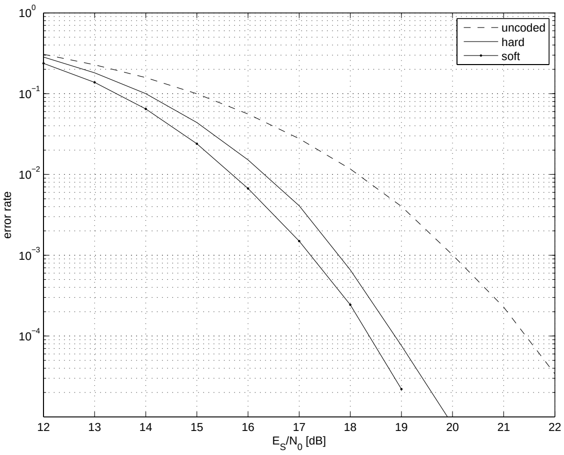

Fig. 2. Simulation results for the OMEC code from example 2 over $\mathcal{G}_{25}$ for the AWGN channel.

This concept of soft decoding can also be applied to Plotkin codes. Consider the hard input decoding described in the previous paragraph. A simple soft input decoder is obtained if, in the final decoding step, the closest codeword is determined by using the Euclidean distance. Corresponding simulation results for the code from example 5 are depicted in Fig. 3. In this case, hard input decoding achieves a gain close to 4dB for a symbol error rate of 10^{-4}. With this simple soft input decoding we obtain an additional gain of 0.5dB with only two candidate codewords.

A larger list can be obtained by using list decoding for the constituent codes. In the first decoding step, we can use soft input maximum-likelihood decoding to decode the repetition code and obtain a list $\mathcal{L}_2$ of codewords from code $\mathcal{C}_2$ by adding the symbols $\{1, -1, i, -i\}$ to the maximum-likelihood estimate. In the second step, we use Chase decoding for $\mathbf{r}'$. Furthermore, we use Chase decoding for $\mathbf{r}'' - \hat{\mathbf{v}}''_l$ for each element $\hat{\mathbf{v}}''_l \in \mathcal{L}_2$ of the list. This results in a list $\mathcal{L}_1$ of codewords from the code $\mathcal{C}_1$.

Finally, we construct all possible codewords

$$|\hat{\mathbf{v}}'_j|\hat{\mathbf{v}}'_j + \hat{\mathbf{v}}''_l|, \quad \hat{\mathbf{v}}'_j \in \mathcal{L}_1, \hat{\mathbf{v}}''_l \in \mathcal{L}_2$$

and select the one which is closest to the received vector.

VI. Conclusions

In this work we have considered codes over Gaussian integer rings. All constructions are possible for Gaussian integer fields as well as for rings. Gaussian integer rings extend the possible complex signal constellations for codes over Gaussian integers. We have shown that binary information vectors can easily be encoded to such codes.

Most previous publications on codes over Gaussian integers considered only hard input or optimum maximum-likelihood decoding (e.g. [1]-[5]). However, maximum-likelihood decoding is only feasible for small signal constellations and short codes. We have shown that low-complexity soft input decoding is possible by using a Chase-type algorithm [11]. Likewise, the recursive structure of the Plotkin construction can be exploited for decoding similar to the sub-optimum decoding of Reed-Muller codes [8], [10], [14].

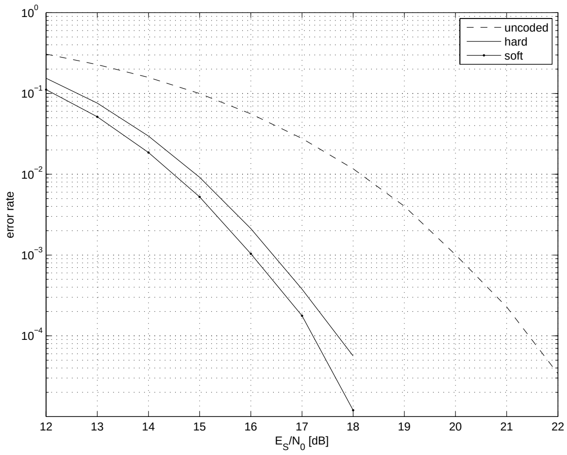

Fig. 3. Simulation results for the Plotkin code from example 5 over $\mathcal{G}_{25}$ for the AWGN channel.

References

[1] K. Huber, *Codes over Gaussian integers*, *Information Theory, IEEE Transactions on*, Volume 40, Issue 1, Jan. 1994 Page(s):207 - 216 *Information Theory, IEEE Transactions on*, Volume: 41 , Issue 5, 1995 , Page(s): 1512 - 1517

[2] Xue-dong Dong, Cheong Boon Soh, E. Gunawan, Li-zhong Tang, *Groups of algebraic integers used for coding QAM signals*, *Information Theory, IEEE Transactions on*, Volume 44, Issue 5, Sept. 1998 Page(s):1848 1860

[3] M. Bossert, E.M. Gabidulin, P. Lusina, *Space-time codes based on Gaussian integers*, in *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, 2002 Page(s):273 - 421

[4] P. Lusina, S. Shavgulidze, M. Bossert, *Space - time block factorisation codes over Gaussian integers*, *IEE Proceedings*, Volume 151, Issue 5, 24 Oct. 2004 Page(s):415 - 421

[5] J. Rifa, *Groups of complex integers used as QAM signals*,

[6] P. Elias, *Error-free coding*, *IRE Trans. on Inform. Theory*, PGIT-4: 29-37, 1954.

[7] Morris Plotkin, *Binary Codes with Specified Minimum Distance*, *IRE Transactions on Inform. Theory*, Vol. 6, pp. 445-450, 1960

[8] G. Schnabl, M. Bossert, *Soft-Decision Decoding of Reed-Muller Codes as Generalized Multiple Concatenated Codes*, *Information Theory, IEEE Transactions on*, Vol. IT-41, pp. 304-308, 1995

[9] H. Scheid and A. Frommer, ”Number theory”, (German), (Elsevier-Spektrum Akademiv publishing, 2007).

[10] N. Stolte, *Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung*, *[Dissertation], TU Darmstadt*, 2002

[11] D. Chase, *A class of algorithms for decoding block codes with channel measurement information*, *IEEE Trans. Inform. Theory*, Vol. 18, pp. 170- 182, Jan. 1972.

[12] M. Bossert, *Channel Coding for Telecommunications*, John Wiley & Sons, 1999

[13] C. Argon, S.W. McLaughlin, and T. Souvignier, *Iterative application of the Chase algorithm on Reed-Solomon product codes*, *Proceedings IEEE ICC 2001*, pp. 320-324, 2001.

[14] I. Dumer , G. Kabatiansky, C. Tavernier, *List decoding of Reed-Muller codes up to the Johnson bound with almost linear complexity*, *Proceedings Information Theory, 2006 IEEE International Symposium on* , pp.138-142, 9-14 July 2006