VI. QUASI-SELF-RECIROC POLYNOMIALS

Quasi-self-reciprocal polynomials (QSRP's), were introduced in [2] as a generalization of SRP's. They represent a superset of SRP's, since QSRP's are defined

\[ f^*(x) = \pm f(x). \]

This extends SRP's over GF \((p^k)\), \(p > 2\), to include those for which

\[ f^*(x) = -f(x). \quad (15) \]

Thus, if \( f(x) = x^r + f_{r-1}x^{r-1} + \cdots + f_0, x \geq 2 \), then \( f(x) = -f(x) \). This is true only if \( f_{r/2} = 0 \).

For \( r \) even, the coefficient of \( x^{r/2} \) in a QSRP must be zero if \( p > 2 \), since by definition \( f_{r/2} = -f_{r/2} \), which is true only if \( f_{r/2} = 0 \).

Division of a QSRP by \( x - 1 \) produces a polynomial which is self-reciprocal with even degree. Therefore, all QSRP's can be derived by multiplying a SRP by \( x - 1 \), and the number of QSRP's of degree \( r = 2t + 1 \) is equal to the number of SRP's of degree \( r = 2t \). Since the exponent of \( x - 1 \) is 1, the exponent of a QSRP is the same as the underlying SRP, so no gains in exponent are possible by considering a QSRP over a SRP.

Like quadratic residues of nonresidues, the product of a QSRP and a QSRP produces a SRP, i.e., QSRP \(*\) QSRP = SRP, SRP \(*\) QSRP = QSRP and SRP \(*\) SRP = SRP, since \((x - 1)^t \) is a SRP if \( t \) is even.

VII. SUMMARY

The maximum possible exponent for a self-reciprocal polynomial over GF \((q)\), \(q \) a prime power \( p^k \), has been derived for \( q \) odd, and a bound determined for \( q \) even. A construction method has been given for \( q \) even which improves the algorithm given in [8]. quasi-self-reciprocal polynomials have also been examined.

REFERENCES


SOLVING EQUATIONS IN FINITE FIELDS AND SOME RESULTS CONCERNING THE STRUCTURE OF GF \((p^m)\)

Klaus Huber, Member, IEEE

Abstract—New methods are given to solve equations over finite fields. The methods exploit the structure of GF \((p^m)\) and are first developed for fields of characteristic two. Then the results are generalized for fields of characteristic \( p > 2 \).

Index Terms—Finite fields, polynomials, root-finding, coset-cycles, Zech's logarithms, Fibonacci numbers, polyhedra.

I. INTRODUCTION

For many applications, e.g., to algebraically decode error correcting block codes, the roots of equations over GF \((p^m)\) must be found. For coding applications this problem is often solved using the Chien-search (see [4]). For other algorithms see [2], [9, ch. 4], [1]. In this correspondence, we propose new deterministic methods which exploit the structure of finite fields, notably coset-cycles. The correspondence is organized as follows: In Section II, we present facts and notations which are needed in the sequel, in particular results on coset-cycles in fields of characteristic two, which are essential for the new methods. Then, in Section III-A, we first present a systematic method to find all roots of an equation in GF \((2^m)\). Section III-B treats a variant of the method. This variant can be extended iteratively. The iterative extension—called iterative root mapping—is described in Section IV. In Section V, some remarks on the complexity of the algorithms for fields of characteristic two are made. The algorithm of Section III A is mainly of theoretical interest. The variant in Section III-B is efficient for small fields (say \( m = 6, 7, \cdots, 17 \)), whereas the iterative method of Section IV appears to give an efficient method even in larger fields. Quantitative results on the efficiency of the iteration method are given for the fields GF \((2^m)\) for \( m = 5, 6, \cdots, 16 \). Then, in Section VI, the results are extended for fields of arbitrary characteristic \( p \). We consider several kinds of coset-cycles, as well as \( IZ \)-conglomerations. The maximal number of cosets in a coset-cycle of the first kind in GF \((p^m)\) is related to Fibonacci numbers. The cycle-length of a coset-cycle of the second kind is twelve. It is also shown that for \( p = 3 \) and \( p = 5 \) polyhedra give nice and natural partitions of GF \((p^m)\). Then the main ideas for finding the roots of polynomials are extended for fields of characteristic \( p > 2 \).

II. BACKGROUND

Before describing the methods, we need some notation and results concerning Zech's logarithm (see [6]) and coset-cycles. Let the field GF \((p^m)\) be constructed using the primitive element \( \alpha \), which is

Manuscript received January 25, 1991. This work was presented in part at AACEC-9, New Orleans, LA, October 7-11, 1991.

The author is with Deutsche Bundespost Telekom, Research Institute, P.O. Box 100003, 6100 Darmstadt, Germany.

IEEE Log Number 9106696.
root of the primitive polynomial $p(x)$. The set $C_s$—called the 
cyclotomic coset modulo $p^m - 1$ (see [8, p. 104]) is defined by
$$C_s = \{ s, p^1 s, p^2 s, \ldots, p^{m-1} s \},$$
where $m_s$ is the smallest number such that $p^{m_s} \cdot s = s \mod p^m - 1$. 
The smallest element of $C_s$ is called coset-representative. $I(x)$, 
the additive inverse modulo $p^m - 1$, is given by
$$I(x) = p^m - 1 - x.$$ 
To represent the element 0 in polar notation (i.e., using exponents 
of a primitive element $\alpha$), we formally introduce the symbol $-\infty$. 
Thus, $\alpha^{-m} = 0$, and the whole set of possible exponents is
$$N_{p^m} = \{ 0, 1, 2, \ldots, p^{m-2} \} \cup \{ -\infty \}.$$ 
Given a polynomial $q(x)$, we refer to the reversed polynomial by
$$\bar{q}(x) = x^{\deg(q(x))} \cdot q(1/x).$$
Zech's logarithm $Z(x)$ is defined by
$$Z(x) = 1 + \alpha^x.$$ 
The following equations follow easily (see [6], [7]):
$$Z(I(x)) = Z(x) - x \mod p^m - 1, \quad x \neq -\infty,$$
$$Z(\mu x) = \mu \cdot Z(x) \quad (\text{mod } p^m - 1),$$
$$Z^{-1}(x) = \begin{cases} Z(x), & \text{for } p = 2, \\
    r + Z(Z(x) - r), & \text{for } p > 2, \alpha' = p - 2, 
\end{cases}$$
$$Z(0) = -\infty, \quad \text{for } p = 2,$$
$$Z\left(\frac{p^m - 1}{2}\right) = -\infty, \quad \text{for } p > 2,$$
$$Z(-\infty) = 0.$$ 
We will also need
$$Z^{-1}(I(x)) = \frac{p^m - 1}{2} + Z^{-1}(x) - x \mod p^m - 1, \quad \text{for } p > 2,$$
which follows from
$$-\alpha^{-x-1} Z^{-1}(x) = -\alpha^{-x}(1 + \alpha x) =$$
$$\alpha^{z-1}(-x), \quad \text{as } -1 = \alpha^{p^{m-1}/2}. \quad \text{Define}$$
$$\alpha^{z(x)} = i + \alpha^x;$$
then in a similar manner we get
$$Z(\xi_j - x) = Z(\xi_j)(x) - x,$$
where $\xi_j = i.$

$Z(x)$ and $I(x)$ map cosets onto cosets of the same size. We now 
focus on fields of characteristic $p = 2$ and introduce the concept of 
coset-cycle (see [6]). Therefore, consider the identity
$$\alpha^t = 1 + \frac{1}{1 + \frac{1}{\alpha^t}}.$$
From this equation, we see that starting from a coset and applying 
$Z$ and $I$ repeatedly, we encounter at most 6 different 
cyclotomic cosets (including the first coset), see Fig. 1. As an example consider 
the field $GF(2^8)$ constructed with the primitive polynomial $p(x) = x^7 + x^3 + 1$. Here, we have three cycles as shown in Fig. 1. The 
coset representatives of the cosets involved in the three cycles are
$$(s_1, s_2, s_3, s_4, s_5, s_6) = (1, 31, 3, 7, 15, 63),$$
$$(s_1, s_2, s_3, s_4, s_5, s_6) = (5, 21, 43, 19, 27, 47),$$
$$(s_1, s_2, s_3, s_4, s_5, s_6) = (9, 11, 29, 23, 13, 55).$$
We refer to cycles such as those shown above as coset-cycles. Let $s$ be an element of a coset $C_j$, where $j$ is the coset representative. 
Then we define the set $CC_s$ by
$$CC_s = \{ \cup_{i=1}^{m_t} C_{s_i} \},$$
where
$$s \in C_j, j \in \{ s_1, s_2, s_3, s_4, s_5, s_6 \}.$$ 
For instance in the above example we have $CC_1 = C_1 \cup C_9 \cup C_{14}$ 
$\cup C_{19} \cup C_{24} \cup C_{29}$. We call the smallest element of $CC_s$ a 
coset-cycle representative, and usually refer to $CC_s$ by setting $s$ equal to 
the coset-cycle representative. It can be shown (see [6]), that for 
increasing $m$ most elements of $N_{2^m}$ lie in coset-cycles involving 6 
different cosets, and in most of the coset-cycles there are $6m$ elements. If there are coset-cycles that contain less than 6 different 
cosets, they contain either 1, 2, or 3 different cosets. 
If we are given a pair $(s, Z(s))$, we can easily compute Zech's 
logarithm of any element of $CC_s$. First by $Z(2x) = 2 \cdot Z(x)$, all 
Zech's logarithm of $2s, 4s, 8s, \ldots, (i.e., of the whole coset) 
follow, and then using $Z^{-1}(x) = Z(x)$ and $Z(I(x)) = Z(x) - x$, 
we can cycle around and compute all the other Zech's logarithms. 
For example in the above field $GF(2^8)$ we have $Z(3) = 7 \Rightarrow Z(6) = 14, \ldots$, and the whole coset follows. Then $Z(3) = 7 \Rightarrow Z(7) = 3 \Rightarrow Z(120) = 123 \Rightarrow Z(123) = 120 \Rightarrow Z(4) = 124$, 
$Z(124) = 4 \Rightarrow Z(3) = 7$. 
Now we define the list $L_s$ by
$$L_s = (s, Z(s), -Z(s), -Z(-s), Z(-s), -s).$$
$L_s$ contains three $I$-pairs $(x, -x)$, and three $Z$-pairs $(x, Z(x))$ 
(note that $Z(-Z(x)) = -Z(-x)$), e.g., in the previous example we have 
$L_s = (5, 82, 45, 50, 77, 122)$. If we know one $Z$-pair $(x, Z(x))$ of a list, we can determine all the other elements of the list. 
Clearly $\sum_{i \in L_s} i = 0 \mod 2^m - 1$. For the algorithms of the 
next sections we need the following polynomials:
$$H_s(x) = \prod_{i \in CC_s} (x - \alpha^i).$$
and
$$h_s(x) = \prod_{i \in L_s} (x - \alpha^i).$$
Obviously, $H_s(x)$ is a polynomial over $GF(2)$, as it is the product of 
all minimal polynomials of a coset-cycle. If $L_s$ contains six 
different elements, we have
$$H_s(x) = \text{lcm}\left\{ h_s(x), h_s(x^{1/2}), \ldots, h_s(x^{1/2m-1}) \right\}^{2^m-1},$$

\begin{center}
\begin{tabular}{c c c c}
$C_{s_1}$ & $\xrightarrow{Z}$ & $C_{s_2}$ & $\xrightarrow{I}$ & $C_{s_3}$ \\
\hline
1 & $\quad\downarrow$ & 2 \\
$C_{s_6}$ & $\xrightarrow{Z}$ & $C_{s_5}$ & $\xrightarrow{I}$ & $C_{s_4}$ \\
\end{tabular}
\end{center}
Fig. 1. Coset-cycle for $p = 2$. 

where \( lcm \) denotes the least common multiple. Below, we will show that there is only one case for \( m \) even, where \( L \) does not contain six different elements.

Let \( M \) be the set of all coset-cycle representatives \( \geq 1 \) (e.g., for the field \( GF(2^7) \) above we get \( M = \{1, 5, 9\} \)). For \( m \) not too big, \( M \) can easily be found, e.g., from the coset-cycles. If we know \( M \), then any polynomial \( H_i(x) \) and \( h_i(x) \) can be determined immediately. In fact, to get \( H_i(x) \), we only need to know the minimal polynomial of any single element \( \alpha_i, i \in CC_{L} \). Then, \( H_i(x) \) follows from

\[
H_i(x) = \text{lcm}\{q_i(x), q_{i2}(x), q_{i3}(x), q_{i4}(x),
\]
\[
q_{i5}(x), q_{i6}(x)\}, \quad (14)
\]

where \( q_i(x) \) is the minimal polynomial of \( \alpha_i \), and

\[
q_i(x) = q_i(x + 1), \quad q_{i2}(x) = q_i(x),
\]
\[
q_{i3}(x) = q_i(x + 1), \quad q_{i4}(x) = q_i(x), \quad q_{i5}(x) = q_i(x + 1), \quad q_{i6}(x) = q_i(x).
\]

For example, in the field \( GF(2^7) \)—from the minimal polynomial of \( \alpha_i \), which is the primitive polynomial \( p(x) \) itself—we get

\[
q_i(x) = p(x) = x^7 + x^3 + 1 \quad \text{and} \quad H_i(x) = q_i(x) \cdot q_{i1}(x) \cdot q_{i2}(x) \cdot q_{i3}(x) \cdot q_{i4}(x) \cdot q_{i5}(x) \cdot q_{i6}(x) \cdot q_i(x).
\]

Then from \( q_i(x) = m_i(x) = x^7 + x^6 + x^5 + x^4 + 1 \) and \( q_{i5}(x) = m_i(x) = x^7 + x^6 + x^5 + x^3 + x^2 + x + 1 \), we obtain \( H_i(x) \) and \( h_i(x) \) in the same way. Note that \( H_i(x) \) is a reversible polynomial, i.e., \( \bar{H}_i(x) = H_i(x) \), and also \( H_i(x + 1) = H_i(x) \).

The polynomials \( h_i(x) \) also have a very particular and simple structure. We get

\[
h_i(x) = (x + \alpha_i^n) \cdot (x + \alpha_i^{n+1}) \cdot (x + \alpha_i^{n+2}) \cdot (x + \alpha_i^{n+3}) \cdot (x + \alpha_i^{n+4}) \cdot (x + \alpha_i^{n+5}) \cdot (x + \alpha_i^{n+6})
\]

\[
= h_i(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1, \quad (15)
\]

where only \( \alpha_i \in GF(2^m) \), i.e., \( h_i(x) \) is specified by a single element of \( GF(2^m) \).

To see that \( h_i(x) \) is of the form in (15), first note that \( h_i(x) = h_i(x) \). Hence, we only have to find the coefficients of \( x^2 \) and \( x^3 \). Setting \( h_i(x) \) equal to \( h_i(x) \), (15) follows. From \( h_i(\alpha_i^n) = 0 \) we immediately obtain \( \alpha_i \) as

\[
\alpha_i = \frac{\alpha^n + \alpha^{n+1} + \alpha^{n+2} + 1}{\alpha^4 + \alpha^5} = \frac{(\alpha^3 + \alpha^2 + 1)^3}{(\alpha^2 + \alpha^3)^2}. \quad (16)
\]

For example for the field \( GF(2^7) \), we get

\[
h_i(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1,
\]
\[
h_i(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1,
\]
\[
h_i(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1.
\]

Table I contains some data for the fields \( GF(2^m) \), \( m = 5, 6, \cdots, 10 \). In the third column, \( m_i(x) \), taken from [10], denotes the minimal polynomial of \( \alpha_i \) in octal (clearly \( m_i(x) \) is the primitive polynomial used). The information on the degree of \( H_i(x) \) is given in such a way, that the size and numbers of cosets involved can be seen, e.g., the entry \( 3 \cdot m/2 \) means that there are three different cosets of size \( m/2 \) in the coset-cycle \( CC_i \).

The first column of Table I, the index \( CC_{L} \), gives the coset-cycle representative of \( CC_{L} \).

Table I

<table>
<thead>
<tr>
<th>( m )</th>
<th>( s )</th>
<th>( Z(s) )</th>
<th>( m_i(x) )</th>
<th>( \deg H_i(x) )</th>
<th>( t_i )</th>
<th>( C_{L} )</th>
<th>( CC_{L} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>18</td>
<td>45</td>
<td>6 \cdot m</td>
<td>26</td>
<td>211</td>
<td>\bar{CC}_{10}</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>6</td>
<td>103</td>
<td>6 \cdot m</td>
<td>1</td>
<td>21</td>
<td>\bar{CC}_{10}</td>
</tr>
<tr>
<td>7</td>
<td>26</td>
<td>111</td>
<td>3 \cdot m</td>
<td>45</td>
<td>27</td>
<td>\bar{CC}_{10}</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>45</td>
<td>15</td>
<td>2 \cdot m/2</td>
<td>0</td>
<td>29</td>
<td>\bar{CC}_{10}</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>27</td>
<td>7</td>
<td>2</td>
<td>\infty</td>
<td>\bar{CC}_{10}</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>71</td>
<td>121</td>
<td>6 \cdot m</td>
<td>118</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>82</td>
<td>235</td>
<td>6 \cdot m</td>
<td>23</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>94</td>
<td>277</td>
<td>6 \cdot m</td>
<td>113</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>51</td>
<td>435</td>
<td>6 \cdot m</td>
<td>32</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>53</td>
<td>765</td>
<td>6 \cdot m</td>
<td>164</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>711</td>
<td>551</td>
<td>3 \cdot m</td>
<td>221</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>120</td>
<td>675</td>
<td>3 \cdot m</td>
<td>204</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>99</td>
<td>453</td>
<td>6 \cdot m</td>
<td>32</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>68</td>
<td>23</td>
<td>6 \cdot m/2</td>
<td>85</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>92</td>
<td>545</td>
<td>6 \cdot m</td>
<td>6</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>170</td>
<td>7</td>
<td>2</td>
<td>\infty</td>
<td>\bar{CC}_{10}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Let \( h_i(x) \) be the formal derivative of \( h_i(x) \), then from

\[
h_i(x) = (x^2 + x + \alpha^{2i}) \cdot h_i(x) + \alpha^i,
\]

we see that for \( \alpha^i \neq 0 \) the polynomial \( h_i(x) \) contains no repeated roots. If \( \alpha^i = 0 \), we get \( h_i(x) = (x^2 + x + 1)^2 \). Hence the only case that \( h_i(x) \) contains repeated roots is for \( m \) even, where \( \alpha^i = 1/2 \) and \( \alpha^i = (2/6m-1) \) are triple roots \( h_i(x) \).

For \( m \geq 5 \) there are \( |M| = 2^m/6m \) different polynomials \( H_i(x) \) and \( h_i(x) \) (for \( 5 \leq m \leq 10 \)) the exact number can be obtained from Table I.

III. SOLVING EQUATIONS IN \( GF(2^m) \): TWO SIMPLE METHODS

A. A First Method

We are interested in finding the roots of the polynomial

\[
Y(x) = \sum_{i=0}^{s} Y_i \cdot x^i, \quad Y_i \in GF(2^m)
\]

where \( s \) denotes the smallest number for which \( Y(x) \) is zero.
in the field \( \text{GF}(2^m) \). Let us assume that the roots of \( Y(x) \) are simple (otherwise replace \( Y(x) \) by \( Y(x)/\gcd(Y(x), Y'(x)) \)), and that \( x \) and \( (x + 1) \) do not divide \( Y(x) \), i.e., \( Y_0 \neq 0 \) and \( \sum Y_i \neq 0 \) (if this is not the case simply divide \( Y(x) \) by \( x \) and/or \( (x + 1) \)).

The root-finding now proceeds as follows. We split the polynomial \( Y(x) \) into polynomials \( y_s(x) \), which have all their roots in \( \{ \alpha^i | i \in \mathbb{C}_s \} \). Therefore, we compute

\[
y_s(x) = \gcd \{ Y(x), H_s(x) \}, \quad \forall s \in M.
\]

If the degree of \( y_s(x) \) is zero, there is no root \( \alpha^i \) with \( i \in \mathbb{C}_s \). If the degree of \( y_s(x) \) is one, we have found a unique root \( \alpha^i \), \( i \in \mathbb{C}_s \). If the degree of \( y_s(x) \) is between 2 and 4, we may choose to use direct methods to find the roots of \( y_s(x) \) such as described in [3], [6]. For those \( y_s(x) \) which are left, we compute

\[
y_s(x) = \gcd \{ y_s(x), h_s(x^{1/2}) \},
\]

where \( j \in \{ 0, 1, \ldots, m - 1 \} \) runs from 0 upwards until all of the degree \( \{ y_s(x) \} \) roots of \( y_s(x) \) are separated by the polynomials \( h_s(x^{1/2}) \). Note that

\[
h_s(x) = x^6 + x^5 + x^{2i-2} x^4 + x^3 + x^{2i+2} x^2 + x + 1.
\]

The roots of \( y_s(x) \) are then all contained in the set \( R_s \):

\[
R_s = \{ \alpha^i | i \in L_{s/2} \} = \left\{ \gamma, 1 + \gamma, \frac{1}{1 + \gamma}, 1 + \gamma^{-1}, 1 + \gamma^{-1} | \gamma = \alpha^{2^i} \right\}.
\]

As the degree of either \( y_s(x) \) or \( h(x^{1/2}) y_s(x) \) is \( \leq 3 \) we may also use direct methods to find these roots. Also, the roots of \( h_s(x) \) itself can be found by solving two equations of degree 2 and 3. Namely, solve \( \omega^3 + \omega^2 + \omega + 1 = 0 \) and \( \beta^2 + \beta + \omega = 0 \), then \( \beta \) is root of \( h_s(x) \).

Thus, essentially, all we have to know is a set of \( |M| \) elements \( \alpha^i \) whose exponents \( i \) are from different coset-cycles (these elements can also be obtained by random search). The polynomials \( H_i(x) \) and \( h_s(x) \) can be computed as needed.

**B. A Second Method**

We now present a closely related second method that can be extended iteratively. Again, let \( Y(x) \) be the polynomial whose roots we want to find:

\[
Y(x) = \sum_{i=0}^c Y_i \cdot x^i = \prod_{j \in S_Y} (x + \alpha^i).
\]

We form

\[
F(x) = \prod_{j=1}^6 Y_j(x),
\]

where

\[
Y_1(x) = Y(x), \quad Y_2(x) = Y_s(x + 1), \quad Y_3(x) = \tilde{Y}_2(x), \quad Y_4(x) = Y_s(x), \quad Y_5(x) = \tilde{Y}_4(x), \quad Y_6(x) = Y_s(x + 1) = \tilde{Y}(x).
\]

Obviously the polynomial \( F(x) \) has the following form:

\[
F(x) = \prod_{j \in S_Y} h_j(x)
\]

\[
= \prod_{j \in S_Y} \left( x^6 + x^5 + x^3 + x + 1 + \alpha^{i_j} \cdot (x^4 + x^2) \right).
\]

The \( i_j \) belong to a subset \( S \) of \( N_{2m} \). Hence, a straight-forward method to find the roots of \( Y(x) \) is to take all polynomials \( h_j(x) \) which divide \( F(x) \), i.e., \( F(x) \) mod \( h_j(x) = 0 \), and then to compute \( \gcd \{ Y(x), h_j(x) \} \), for all \( \xi \). In this way, we can do even more operations in parallel than with the first method, and computing \( F(x) \) mod \( h_j(x) \) can be implemented using shift-registers. It is therefore suitable for applications in error control coding (for say \( m = 0, 7, \ldots, 12 \)). For instance in the field \( \text{GF}(2^7) \) a decoder may use \( (2^2 - 2)/6 = 21 \) almost identical shift-registers of length 6, wired to compute \( F(x) \) mod \( h_j(x) \). This gives very regular and repetitive hardware structures (in particular if normal-bases are used). Hence, using classical algorithms, the root-finding can essentially be done in time \( O(e^k) \) for forming \( F(x) \), feeding \( F(x) \) into the shift registers, and computing the \( \gcd \{ Y(x), h_j(x) \} \), for all \( \xi \). This second method is also the starting point for an iterative root finding method, which will be treated in the next section.

**IV. FINDING ROOTS IN \( \text{GF}(2^m) \) BY ITERATIVE ROOT MAPPING**

In the previous sections, we have seen that knowledge of \( |M| \) elements from \( \text{GF}(2^m) \) enables us to solve equations over \( \text{GF}(2^m) \). For \( m > 5 \), \( |M| \) is well approximated by \( 2^m/6m \). For increasing \( m \) this number grows exponentially. Therefore, it is of interest to decrease this number. This is the motivation for the iterative extension of the second method treated in this section. We consider

\[
F(x) = \prod_{j \in S_Y} \left( a(x) + \alpha^{i_j} \cdot b(x) \right) = \prod_{i=1}^6 Y_i(x),
\]

from the previous section, where \( a(x) = (x^3 + x + 1)^3 \) and \( b(x) = (x^3 + x)^2 \).

If we would know the set \( \{ \alpha^i | i \in S_Y \} \), we could find the roots of \( Y(x) \) by computing \( \delta(x) = \gcd \{ Y(x), h_i(x) \}, \forall i \in S_Y \) (recall that \( h_i(x) \) is specified by \( \alpha^i \)). Note that to find the roots of \( h_i(x) \) we never have to solve equations of degree greater than 3, since either \( \delta(x) \) or \( h_i(x)/\delta(x) \) has degree \( \leq 3 \). But even in large fields \( \text{GF}(2^m) \) there are very efficient methods to solve such low degree equations (see [3]).

The main observations for the iterative algorithm are the following.

- Knowledge of all \( \alpha^i, i \in S_Y \) leads immediately to the roots of \( Y(x) \).
- The set of all \( \alpha^i \) is smaller than the set of all \( \alpha^s \) (there are about six times less cosets involved).

To make things more precise let us introduce the mapping \( \sigma \), which relates the exponents \( s \) and \( t_s \), i.e., \( t_s = \sigma(s) \). Equation (16) yields

\[
t_s = 3 \cdot Z(s + Z(s)) - 2 \cdot (Z(s) + s). \quad (17)
\]

\( \sigma \) is a mapping from \( N_{2^m} \) to a subset of \( N_{2^m} \): \( \sigma: N_{2^m} \rightarrow S \), where (for \( m \geq 5 \)) \( |S| = |N_{2^m}|/6 \).

Now let us apply \( \sigma \) on the set \( S \):

\[
\sigma: S \rightarrow S'.
\]

If the set \( S' \) is smaller than \( S \), we can continue in this way until we reach a final set \( FS \). For the final set we have \( \sigma: FS \rightarrow FS \). Let us introduce the iterated function

\[
\sigma^{(i+1)}(x) = \sigma(\sigma^{(i)}(x)), \quad i = 1, 2, \ldots
\]
where \( \sigma^{(1)} = \sigma \), and let the iteration number \( f \) be the smallest integer such that \( \sigma^{(1)} N \rightarrow FS \). If the size \( FS \) is substantially smaller than \( N \), then the following iterative extension reduces the number of elements \( \alpha^f \) which have to be found, in the same way. We form the polynomial \( T(x) \):

\[
T(x) = \sum_{j=0}^{e-1} s_j x^j = \prod_{i \in S_y} (x - \alpha^0) = \prod_{i \in S_y} (x - \alpha^{e-0}),
\]

where \( e = \deg \{ Y(x) \} \). The polynomial \( I(x) \) can be computed rapidly from the polynomial \( F(x) \). First, note that

\[
F(x) = \sum_{i=0}^{6e} F_i x^i = a(x)^e \cdot T_e + a(x)^{e-1} \cdot b(x) \cdot T_{e-1} \cdot \ldots \]

\[+ a(x) \cdot b(x)^{e-1} \cdot T_1 + b(x)^e \cdot T_0.
\]

Hence,

\[
T_e = T_{6e} \quad \text{and} \quad T_{6e-k} = \sum_{i=0}^{6e-k} G^{(i)} x^i = F(x) - T_e \cdot a(x)^e
\]

\[
T_{e-1} = \sum_{i=0}^{6e-2} G^{(i)} x^i = F(x) - T_{e-1} \cdot a(x)^{e-1} \cdot b(x)
\]

\[
T_{e-2} = \sum_{i=0}^{6e-4} G^{(i)} x^i = F(x) - T_{e-2} \cdot a(x)^{e-2} \cdot b(x)^2
\]

\[
\vdots
\]

\[
T_{e-j} = \sum_{i=0}^{6e-2j} G^{(j)} x^i = F(x) - T_{e-j} \cdot a(x)^{e-j} \cdot b(x)^j
\]

and so forth.

Remark: Another way to compute the polynomial \( T(x) \) is by using the element \( y \) which is root of \( x^2 + x + 1 = 0 \). Then \( a(y) = 0, b(y) = 1 \), and the coefficients of \( T(x) \) follow: \( T_0 = F(y), T_1 = (F(x) - T_0 \cdot b(x))/a(x) \), etc. The disadvantage of this way of computing \( T(x) \) is that for odd \( m \) the element \( y \) lies in \( GF(2^m) \).

The iterated method now consists in forming the following sequence of polynomials:

\[
Y^{(0)}(x) = \prod_{i \in S_y} (x + \alpha^i),
\]

\[
Y^{(1)}(x) = \prod_{i \in S_y; \sigma(i) \neq -\infty, 0} (x + \alpha^{e(i)}),
\]

\[\vdots\]

\[
Y^{(J)}(x) = \prod_{i \in S_y; \sigma^{(J)}(i) \neq -\infty, 0} (x + \alpha^{e(J)(i)}),
\]

\[\vdots\]

\[
Y^{(I)}(x) = \prod_{i \in S_y; \sigma^{(I)}(i) \neq -\infty, 0} (x + \alpha^{e(I)(i)}).
\]

The computation of \( Y^{(J)}(x) \) from \( Y^{(J-1)}(x) \) is the same as that of \( T(x) \) from \( T(x) \); we denote this by \( Y^{(J)}(x) = \tilde{Y} \{ Y^{(J-1)}(x) \} \).

Then, we solve \( Y^{(J)}(x) \) by gcd operations with the minimal polynomials \( m_i(x) \), where \( \eta \in FS \):

\[
gcd \{ Y^{(J)}(x), m_\eta(x) \} \quad \forall m_\eta(x), \eta \in FS.
\]

Once we have found the solutions of \( Y^{(J)}(x) \), we can work backwards and solve all the other \( Y^{(J)}(x) \) by gcd operations:

\[
gcd \{ Y^{(J-1)}(x), a(x) + \alpha^j \cdot b(x) \},
\]

where

\[Y^{(J)}(\alpha^j) = 0.\]
most \( f_{\text{max}} \) mappings \( \sigma \). Let

\[
Y(x) = \sum_{i=0}^{e} Y_i \cdot x^i = \prod_{i \in S_y} (x - \alpha^i)
\]

and

\[
\mathcal{F}\{ Y(x) \} = \prod_{i \in S_y} (x - \alpha^{\sigma(i)}).
\]

We now compute the following sequence of polynomials:

\[
Y^{[0]}(x) = Y(x)/\gcd\{ Y(x), M(x) \},
\]

\[
Y^{[1]}(x) = \mathcal{F}\{ Y^{[0]}(x) \}/\gcd\{ \mathcal{F}\{ Y^{[0]}(x) \}, M(x) \},
\]

\[
Y^{[2]}(x) = \mathcal{F}\{ Y^{[1]}(x) \}/\gcd\{ \mathcal{F}\{ Y^{[1]}(x) \}, M(x) \},
\]

\[
\vdots
\]

\[
Y^{[j]}(x) = \mathcal{F}\{ Y^{[j-1]}(x) \}/\gcd\{ \mathcal{F}\{ Y^{[j-1]}(x) \}, M(x) \},
\]

until \( j = f_{\text{max}} \) or \( \deg \{ Y^{[j]}(x) \} \leq 1 \) (or 2, 3, 4, if direct solution by gcd operations).

The roots of \( Y(x) \) are then obtained from the roots of \( \gcd\{ \mathcal{F}\{ Y^{[j]}(x) \}, M(x) \} \) as before by gcd operations.

In Example 1, using \( M(x) = m_0(x) \), we obtain

\[
Y^{[0]}(x) = Y(x), \quad \text{as} \quad \gcd\{ Y(x), M(x) \} = 1,
\]

\[
Y^{[1]}(x) = \mathcal{F}\{ Y^{[0]}(x) \}/((x + \alpha^{60}) \cdot (x + \alpha^{11})
\]

\[
\cdot (x + \alpha^{120})],
\]

\[
Y^{[2]}(x) = \mathcal{F}\{ Y^{[1]}(x) \}/((x + \alpha^{30}) \cdot (x + \alpha^{13})).
\]

With the roots found, we can work backwards (using the corresponding \( h_i(x) \)) until all roots of \( Y(x) \) in \( \text{GF}(2^m) \) are located.

V. REMARKS ON THE COMPLEXITY OF THE ALGORITHMS

We now comment on the complexity of the algorithms for fields of characteristic two. The algorithms of Sections III-A and III-B are useful if the fields \( \text{GF}(2^m) \) are comparatively small (say \( m \leq 12 \)). Their main advantage is the resulting structural simplicity which exploits a natural partition of \( \text{GF}(2^m) \). This makes the algorithm of Section III-B suited for hardware realizations, having time complexity \( O(e^2) \) for finding the roots of \( Y(x) \).

The iterative algorithm of Section IV, however, can be used in larger fields. To judge the average time to find the roots of a polynomial \( Y(x) \) of degree \( e \), first note that the root-mapping and gcd operations with \( h_i(x) \) can be done in time \( O(e^2) \). In Table II, the average number \( a_e \) of mappings, necessary to send a random element \( i \in \{ 1, 2, 2^m - 2 \} \) into the set \( FS \cup C_0 \) is given for \( m = 5, 6, \ldots, 16 \). The results obtained so far indicate that \( a_e \) is a slowly growing function of the logarithm of the size of the field. \( f \) gives the maximum number of mappings to send any element of \( N_2 \) to \( FS \cup C_0 \).

The determination of polynomials \( M(x) \) or of the set \( FS \) can be done in two ways. First, by applying the root mapping to random elements of \( \text{GF}(2^m) \). Second, analytically using

\[
x^j = f^{[j]}(x), \quad j = 0, 1, \ldots, m - 1, \quad l = 1, 2, \ldots,
\]

where

\[
f^{[j+1]}(x) = f(f^{[j]}(x)), \quad f^{[1]}(x) = f(x) = a(x)/b(x).
\]

\( i \) gives the number of cosets in \( FS \), which are connected by the mapping \( \sigma \). In the last row of Table II, \( i \) gives the smallest number of minimal polynomials (excluding \( x + 1 \) and \( x \)) which can be chosen as factors of \( M(x) \) for \( m = 5, \ldots, 16 \).

VI. GENERALIZATION FOR FIELDS OF CHARACTERISTIC \( p \)

In the following we extend the range of applications to fields of characteristic \( p > 2 \). First it is natural to ask how the cosets are related in \( \text{GF}(p^m) \). With respect to the mapping \( I(x) \) there is no problem. \( I(x) \) is still its own inverse. However, this is no more true for \( Z^{-1}(x) \) (see (4)). Equation (9) shows, that \( p \) Elements of \( N_{p^m} \) are connected in a \( Z \)-cycle: \( x_1 \xrightleftharpoons{5} x_2 \xrightleftharpoons{3} x_3 \xrightleftharpoons{4} \ldots \xrightleftharpoons{2} x_0 \). The structure of the coset-cycles in \( \text{GF}(p^m) \) is only slightly more complicated than for fields of characteristic two. We distinguish two kinds of coset-cycles.

Coset Cycles of the First Kind: Let us start with \( p = 3 \). Here, we get

\[
\alpha' = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \alpha^3}}},
\]

Written as a simple continued fraction we have \( \alpha' = [1, 1, 1, 1, \alpha^3] \), i.e., for \( p = 3 \) we get coset-cycles which involve at most 8 different cosets. This is shown in Fig. 3. Note that the arrows for \( Z \) are only in one direction. To determine the length of the coset-cycles for general \( p \), we write

\[
\alpha^x = [1, 1, 1, \ldots, 1, \alpha^x].
\]

After some mathematical manipulations, we find that the maximal length \( w_p \) of the coset-cycles is given by

\[
w_p = 2 \cdot u_p,
\]

where \( u_p \) is the index of the smallest Fibonacci number which contains \( p \) as divisor (Fibonacci numbers are defined by \( F_{i+2} = F_{i+1} + F_i \) and \( F_0 = 0, F_1 = 1 \)). Thus, to find the maximal length of the coset-cycles, we simply have to find the smallest Fibonacci-number \( u_p \) which is a multiple of \( p \), then the maximal number of cosets in a coset-cycle is given by twice the index \( u_p \). In Table III, the maximal number \( w_p = 2 \cdot u_p \) of cosets in a coset-cycle of the first kind is given for the first primes. A simple upper-bound of \( w_p(p + 5) \) is

\[
w_p \leq 2 \left( p - \left( \frac{5}{p} \right) \right),
\]

where \( (5/p) \) denotes Legendre's symbol. This bound follows immediately from the well-known fact that \( p \) divides \( F_{p-5}/(p) \) (see e.g., p 150, [5]). From the divisibility properties of Fibonacci numbers we further infer that \( w_p \) divides \( 2 \cdot (p - (5/p)) \).

Coset-Cycles of the Second Kind: Coset-cycles of the second kind are obtained by alternating \( Z \) and \( Z^{-1} \), i.e., we ask for the maximal cycle-length of a cycle \( x_0 \xrightleftharpoons{5} x_1 \xrightleftharpoons{3} x_2 \xrightleftharpoons{4} \ldots \xrightleftharpoons{2} x_0 \). The maximal length of a coset-cycle of the second kind

<table>
<thead>
<tr>
<th>( m )</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_e )</td>
<td>0.78</td>
<td>1.16</td>
<td>1.20</td>
<td>1.91</td>
<td>2.47</td>
<td>2.39</td>
<td>6.35</td>
<td>5.12</td>
<td>9.70</td>
<td>12.1</td>
<td>9.81</td>
<td>12.6</td>
</tr>
<tr>
<td>( f )</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>13</td>
<td>16</td>
<td>23</td>
<td>33</td>
<td>37</td>
</tr>
<tr>
<td>( i )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

TABLE II

NUMERICAL DATA OF THE ITERATIVE ROOT-MAPPING FOR THE FIRST KIND \( \text{GF}(2^m) \): \( m = 5, 6, \ldots, 16 \)
TABLE III
SIZE OF ZI-CYCLES AND ZI-CONGLOMERATIONS

<table>
<thead>
<tr>
<th>p</th>
<th>w_p</th>
<th>p^3 - p</th>
<th>r_p</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>336</td>
<td>336</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>1320</td>
<td>1320</td>
</tr>
</tbody>
</table>

is found for any p to be twelve. This follows from the identity
\[ y = [1, -1, 1, -1, 1, -1, y]. \]  (21)

If we know a \((x, Z(x))\) pair, we can easily determine all elements of the corresponding coset-cycle of the second kind using (2) and (8). Namely, let \(Z(a) = b\), and \(\xi = (p^n - 1)/2\), i.e., \(\xi = -\xi \mod p^n - 1\), then
\[
\begin{align*}
x_0 &= a, \\
x_1 &= Z(a) = b, \\
x_2 &= -x_1 = -b, \\
x_3 &= Z^{-1}(x_2) = Z^{-1}(-x_1) = \xi + Z^{-1}(x_1) - x_1 \\
&= \xi + a - b, \\
x_4 &= -x_2 = \xi - b, \\
x_5 &= Z^{-1}(x_3) = \ldots = \xi + b, \\
x_6 &= -x_4 = \xi - b, \\
x_7 &= Z(x_6) = \ldots = a - b, \\
x_8 &= -x_5 = b - a, \\
x_9 &= Z^{-1}(x_8) = \ldots = -a, \\
x_{10} &= -x_7 = a.
\end{align*}
\]

To summarize in terms of elements of \(N_p\), we have considered the following cycles.

- Z-cycles connecting p elements of \(N_p\): \(x_1 \sim x_2 \sim x_3 \sim \ldots \sim x_p \sim x_1\).
- Zi-cycles of the first kind connecting at most \(w_p\) elements of \(N_{p^n}\): \(y_1 \sim y_2 \sim y_3 \sim \ldots \sim y_{w_p} \sim y_1\).
- Zi-cycles of the second kind connecting at most 12 elements of \(N_{p^n}\): \(z_1 \sim z_2 \sim z_3 \sim \ldots \sim z_{11} \sim z_{12} \sim z_1\).

To generalize the root-finding algorithms, we now consider the set \(L(s)\) which contains \(s \in N_{p^n}\) and all values which can be reached from s by the mappings \(Z\) and \(I\). We name \(L(s)\) Zi-conglomeration and denote by \(r_p\) the maximal size of \(L(s)\). For instance, for \(p = 3\) we find that at most \(r_3 = 24\) elements are connected, these relationships are displayed in Fig. 4. The one-sided arrows denote the \(Z\)- and the double-sided arrows the \(I\)-mapping.

Table III gives \(r_p\) for the first primes. As \(p\) elements are connected in a Z-cycle, we know that \(p\) is a divisor of \(r_p\). From (9) and (10), we also expect that \(p \cdot (p - 1)\) divides \(r_p\). In general, the following bound holds for \(r_p\):
\[
r_p \leq p \cdot (p - 1) \cdot (p + 1) = p^3 - p. \]  (22)

To see this consider the polynomial
\[
\begin{align*}
u_s(x) &= \sum_{i=0}^{p-1-i} x^{p-1} + y_s \cdot \left( x^p - x \right)^{p(p-1)} \\
&= \frac{x^{p^2-1}}{x^{p-1} - 1} + y_s \cdot \left( x^p - x \right)^{p(p-1)}.
\end{align*}
\]  (23)

Now, if \(\eta\) is any root of \(v_s(x)\), where \(\eta^p \neq \eta\) and let \(a \in GF(p)\), then we find that \(v_s(a + \eta) = 0\) and \(v_s(\eta^{-1}) = 0\). Hence, \(r_p\) is bounded by the degree of \(v_s(x)\) and we get \(r_p \leq p^3 - p\).

This upper bound is attained e.g., for \(p = 2, 3, 7, 11\). For \(p = 2\) we see that if we set \(y_s = 1 + a^s\) then \(h_s(x)\) equals \(v_s(x)\) (equivalently, for \(p = 2\), we could have defined \(h_s(x)\) by \(v_s(x)\) with \(y_s = a^s\)). If \(r_p = p^3 - p\) then the polynomial \(h_s(x)\) is defined by
\[
h_s(x) = a(x) + \alpha^s \cdot b(x),
\]
where
\[
a(x) = x^{p^2-1} - 1 \\
b(x) = (x^p - x)^{p(p-1)}.
\]
In this case, \(h_s(x)\) is again specified by a single element \(\alpha^s\) of the big field \(GF(p^n)\). From \(h_s(\alpha^s) = 0\) we find
\[
\alpha^s = -\frac{a(\alpha^s)}{b(\alpha^s)} = \frac{\alpha^{p^3-2} - \alpha^s}{(\alpha^{p^3-2} - \alpha^s)^{p(p-1)+1}}.
\]
For instance for \( p = 3 \) we get

\[
a(x) = x^{24} + x^{20} + x^{18} + x^{16} + x^{14} + x^{12} + x^{10} + x^8 + x^6 + x^2 + 1
\]

\[
b(x) = x^{18} + x^6 = (x^2 - x)^6 = x^3(x + 1)^6(x + 2)^6.
\]

To see when \( r_p \) is smaller than \( p^3 - p \) we first consider the possibility of multiple roots of \( u_s(x) \). Let us assume that \( \eta \) (with \( \eta^p \neq \eta \)) is a multiple root of \( u_s(x) \), i.e., the formal derivative \( u'_s(x) \) has also a root at \( \eta \). Then from \( u_s(\eta) = 0 \) and \( u'_s(\eta) = 0 \) we get \( y_\eta = 1 \). Hence, for \( y_\eta \neq 1 \) the polynomial \( u_s(x) \) contains only simple roots. However, the polynomial \( u_s(x) \) does have a property which the polynomial \( h_2(x) \) does not necessarily have to have, namely,

\[
\forall \eta \in \mathbb{GF}(p) \setminus \{0\} \text{ we have } u_s(\eta) = 0 \Rightarrow u_s(\eta \cdot \eta) = 0 \text{ (again } \eta^p \neq \eta). \]

The smallest characteristic \( p \) for which \( h_2(x) \) does not have this property is for \( p = 5 \). For \( \gamma \in \mathbb{GF}(5^m) \), the following equation holds:

\[
y = \left[ \frac{W}{\gamma} \right].
\]

Together with the \( Zl \)-cycles of the first and second kind, we can find the \( Zl \)-conglomeration for \( p = 5 \). A half of this is displayed in Fig. 5. Thus, \( r_5 = 60 \). For \( p = 5 \) the polynomial \( h_2(x) \) is still characterized by a single element of the extension field, we get \( h_2(x) = a(x) + \alpha \cdot b(x) \), where \( a(x) = (x^5 - x)^2 + 3 \cdot (x^5 - x)^3 + 3 \cdot (x^5 - x)^4 + 1 \) and \( b(x) = (x^5 - x)^5 \) (again \( \eta^p \neq \eta \)). Thus we have determined the polynomials \( h_2(x) \) for the primes \( p \leq 11 \). In general, the polynomial \( u_s(x) \) can be used in the same way as \( h_2(x) \).

From Figs. 4 and 5 we see, that for \( p = 3 \) and 5, the \( Zl \)-conglomerations can be arranged in a pleasing geometric way. For \( p = 3 \) the \( Z \)-cycles are placed on the vertices of a cube, and for \( p = 5 \) the \( Z \)-cycles are on the vertices of an icosahedron.

Before we generalize the root finding, we look at the roots of \( u_s(x) \) in the following way. Let \( u_s(\eta) = 0 \), where \( \eta^p \neq \eta \), and let the set \( A \) be defined as

\[
A = \{a_i \mid a_i^p = \eta, i \in \{0, 1, \ldots, p - 1\}\}.
\]

Then by the properties of \( u_s(x) \) all roots of \( u_s(x) \) are given by \( \alpha^i, \beta \in \mathbb{E} \), where \( \mathbb{E} \) consists of the sets \( K \) and \( U \):

\[
\mathbb{E} = U \cup K
\]

\[
U = \left\{ \pm a_i + j \cdot \frac{p^m - 1}{p - 1} \mid i = 0, 1, \ldots, p - 1; j = 0, 1, \ldots, p - 2 \right\}
\]

\[
K = \left\{ a_i - a_j + k \cdot \frac{p^m - 1}{p - 1} \mid i \neq j; i, j \in \{0, 1, \ldots, p - 1\} \right\},
\]

\[
k \in \{0, 1, \ldots, p - 2\}.
\]

Clearly, \( U \cap K \) is empty. The size of \( U \) is \( 2 \cdot p \cdot (p - 1) \) and the size of \( K \) is \( p \cdot (p - 1)^2 = |\mathbb{E}| = p^3 - p \). In this way, all roots of \( u_s(x) \) are characterized by a single root \( \eta \). Finding a root of \( u_s(x) \) can be reduced to finding a solution of three equations of degree \( p + 1, p, \) and \( p - 1 \) as follows:

\[
u_s(x) = \frac{x^{p^2 - 1} - 1}{x^p - 1} + y_\eta \cdot (x^p - x)^{p - 1}
\]

\[
= u_s(x) = g(u) = u^{p^2 - 1} + u^{p - 1} + 1 + y_\eta \cdot u^{p^2 - p},
\]

where \( u = x^{p^2 - x} \) and

\[
u_s(x) = k(z) = z^{p + 1} + z + 1 + y_\eta \cdot z^p,
\]

where \( z = u^{p - 1} \). (Here \( Y(x + 1) \) means first replace \( x \rightarrow x + 1 \) and then reverse the
polynomial. $F(x)$ is then given by

$$F(x) = \prod_{j \in S_F} \left( a(x) + a_j \cdot b(x) \right).$$

Now $T(x)$ follows in a similar way as in Section IV.

VII. CONCLUSION

New results concerning the structure of finite fields have been given and exploited for the root-finding of polynomials. The iterative algorithm given in this paper is an efficient root-finding method for fields of small characteristic. Currently a rigorous mathematical analysis of the complexity of this algorithm is lacking. The quantitative results that have been obtained for fields of characteristic two indicate a time complexity of $O(e^m)$ for a polynomial of degree $e$ over GF ($2^m$) with $\omega = 1$, i.e., the iterative algorithm is quite efficient. For $p > 2$ the size of $L(s)$ makes the method less efficient.

ACKNOWLEDGMENT

The author would like to thank the referees for helpful comments on this correspondence.

REFERENCES


Computing Bit-Error Probabilities for Avalanche Photodiode Receivers by Large Deviations Theory

K. Ben Letaief and J. S. Sadowsky

Abstract—A bit error probability analysis of direct detection optical receivers employing avalanche photodiodes is presented. An asymptotic analysis for large signal intensities is first presented. This analysis provides some useful insight into the balance between the Poisson statistics, the avalanche gain statistics, and the Gaussian thermal noise. The conjugate distribution is then developed, which is obtained by applying the large deviations exponential twisting formula. It is demonstrated that this conjugate distribution can be used to obtain numerically efficient Monte Carlo estimates of the bit-error probability via the importance sampling method.

Index Terms—Avalanche photodiode, large deviations theory.

I. INTRODUCTION

We consider an optical communications system that uses the direct detection integrate-and-dump receiver as illustrated in Fig. 1. This receiver employs an APD (avalanche photodiode) to convert the received optical signal into electrical current. The incident light first produces primary electrons. Each primary electron is then multiplied by a random avalanche gain to produce a number of secondary electrons. The number of primary electrons is a Poisson random variable that depends on the intensity of the optical signal. The avalanche diode’s output current, consisting of the secondary electrons, is then amplified, and this adds a Gaussian amplifier thermal noise component.

Consider the transmission of a single information bit by an optical pulse in the time interval [0, T]. $X_k$ will denote the avalanche gain for the $k$th primary electron. The sequence $\{X_k\}$ is i.i.d., and the univariate gain distribution has been well characterized in the literature [12]-[14]. (We assume that the avalanche process is instantaneous.) $N$ will denote the total number of primary electrons produced in the signaling interval, and $N_G$ will denote the receiver’s thermal noise response. The receiver’s decision statistic is

$$D = \sum_{k=1}^{N} X_k + N_G. \quad (1)$$

We assume OOK (on-off keyed) modulation of the optical intensity. The statistical hypotheses are

$$H_0(0 \text{ transmitted}): N \text{ is Poisson with parameter } \lambda_0,$$

$$H_1(1 \text{ transmitted}): N \text{ is Poisson with parameter } \lambda_1,$$

where $\lambda_0 < \lambda_1$. Our goal is to estimate the bit-error probabilities $P_0 = P_D(\text{bit error}) = P(D \geq \gamma)$ and $P_1 = P_D(\text{bit error}) = P(D < \gamma)$, where $\gamma$ is the test threshold and $P(D)$ is the $H_1$ probability distribution.

In this correspondence, we attack the problem of estimating $P_0$ and $P_1$, using techniques of large deviations theory. We first present an asymptotic analysis for the limit as both $\lambda_0$ and $\lambda_1 \to \infty$. In this limit, we must also control the test threshold $\gamma$ and the Gaussian noise variance $\sigma^2_N$. Without loss of generality, we use $\gamma$ as the control variable by setting $\lambda_1 = \gamma c_1$ and $\lambda_0 = \gamma c_2$, where $c_1$ and $c_2$ are constants, and we consider the limit as $\gamma \to \infty$ (In Section II, we shall discuss the physical significance of this asymptotic formulation in more detail.) The large deviations result for this large intensity limit is

$$P_1 - C_1^* \exp (- I_1^* \gamma),$$

where $C_1^*$ and $I_1^*$ are positive constants.\(^1\)

It turns out that the above limit is essentially identical to the saddlepoint approximation. Helstrom [10] has provided some numerical computations indicating that the saddlepoint approximation can be very accurate for many practical systems. So, one could interpret our asymptotic result as another verification (for the large

\(^1\) We write $a_n \sim b_n$ to mean precisely that $\lim_{n \to \infty} a_n / b_n = 1.$