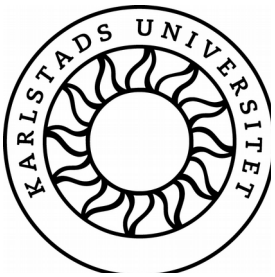

Construction of Superimposed Codes Using Graphs and Galois Fields

Konstruktion av överlagrade koder med grafer och Galoiskroppar

David Johansson

Faculty of Health, Science and Technology

Mathematics, Bachelor Degree Project

15.0 ECTS Credits

Supervisor: Igor Gachkov

Examiner: Niclas Bernhoff

July 2017

Abstract

In this thesis some constructions of superimposed codes are presented. Many of the known nontrivial constructions arise from $t-$designs, and the constructions discussed in this thesis is also based on a block design idea. Superimposed codes are rather combinatorial in nature, so the connection to $t-$designs is not too surprising. What may be a little surprise, however, is the connection between superimposed codes and linear codes and Galois fields. Linear codes are quite intuitive and have nice properties, as is the case for Galois fields; combinatorial structures are quite often the contrary, not intuitive and quite difficult to understand. Because of this, it is interesting that a combinatorial structure like superimposed codes can be constructed from structures like linear codes and Galois fields.

The main goal of this thesis is to present two possibly new approaches to construct superimposed codes. The constructions are described, but not proved to be correct. The first construction presented is using graphs. In practice, this is not a good way to construct codes, since it requires the construction of a graph and finding certain cycles in the graph. It is still an interesting construction, however, since it provides a connection between constant weight codes and superimposed codes. Another construction is presented, one that seems much more useful when constructing codes. In [7] one particular superimposed code is constructed from a Galois field. In this thesis we will see that this construction using Galois fields can be generalized.

Sammanfattning

I denna uppsats presenteras några konstruktioner av överlagrade koder. Många av de redan kända konstruktionerna har sitt ursprung i t-designer, och även konstruktionerna som behandlas i denna uppsats är baserade på en blockdesignsidé. Överlagrade koder är tämligen kombinatoriska till sin natur, så kopplingen mellan överlagrade koder och t-designer är inte speciellt överraskande. Däremot kan kopplingen mellan överlagrade koder, linjära koder och Galoiskroppar vara överraskande. Linjära koder är ganska intuitiva och har trevliga egenskaper, likaså Galoiskroppar; kombinatoriska strukturer är ofta tvärt om, inte intuitiva och svåra att förstå. På grund av detta är det intressant att kombinatoriska strukturer som överlagrade koder kan konstrueras med hjälp av strukturer som linjära koder och Galoiskroppar.
Det primära målet med denna uppsats är att presentera två möjligen nya konstruktioner av överlagrade koder. Konstruktionerna beskrivs men deras korrekthet bevisas inte. Den första konstruktionen som presenteras är baserad på grafer. I praktiken är denna konstruktionen inte bra för att skapa koder, eftersom den kräver konstruktion av en graf och sedan att hitta vissa cykler i grafen. Det är dock fortfarande en intressant konstruktion, eftersom den bidrar till en intressant koppling mellan konstantvikt koder och överlagrade koder. En annan konstruktion presenteras, och den är mycket mer praktiskt användbar. I [7] skapas en specifik överlagrad kod med hjälp av en Galoiskropp. I denna uppsats ser vi hur denna konstruktion med Galoiskroppar kan generaliseras.

Contents

1 Introduction **1**

2 Finite Fields **3**
2.1 Introduction to Ring and Field Theory 3
2.2 Polynomial Rings 6
2.3 Field Extensions 8

3 An Introduction to Codes **10**
3.1 General Concepts 10
3.2 Linear Codes 11
3.3 Nonlinear Codes 15

4 Constant Weight Codes **18**

5 Superimposed Codes **20**

6 Main Results **22**
6.1 Construction Using Graphs 22
6.2 Construction Using Galois Fields 27
6.2.1 Concluding Remarks 33

A Code Verification Program **35**

Chapter 1

Introduction

Many of the known constructions of superimposed codes are based on combinatorial designs. Not too much research seems to be done directly on the subject of superimposed codes, but quite a bit of work is done indirectly via combinatorial designs. Several constructions of superimposed codes is presented in [9], one of which uses super-simple t-designs. Because of this connection between t-designs and superimposed codes, some research on t-designs is also very useful for the study of superimposed codes. Some researchers uses this connection as additional motivation for their study of super-simple t-designs and related combinatorial designs. For example, all of [6],[2],[3],[4] and [1] are research articles on super-simple t-designs, but the main results in those articles can also be used to create superimposed codes. As demonstrated in [7] and [8], however, super-simple t-designs were studied before the results in [9] were known.
The main goal of this thesis is to propose some new constructions of superimposed codes. Initially, the goal was to find geometrical constructions, but after some not very successful attempts, the focus was shifted to a more algebraic approach. The best constructions that was found are based on Galois fields, but also a construction based on cycles in graphs is proposed. The graph construction is not very good for constructing superimposed codes, but it provides an interesting connection between constant weight codes and superimposed codes. None of the constructions are proved to be correct, but several examples are provided for both constructions. However, the codes constructed from graphs was already known before this thesis, thus no new codes was found that way. The construction using Galois fields is much more practical and is demonstrated in a few constructions of not previously known nontrivial examples. The construction using Galois fields may in fact be a construction of some combinatorial design that sometimes also happens to be a superimposed code. This is discussed further in Chapter 6.
Since the construction of superimposed codes using Galois fields seems to be the best one found in this thesis, we start the thesis off by a review of field theory and construction of Galois fields in Chapter 2.
In Chapter 3 a brief review of the theory of linear codes and $t-$designs is given. Superimposed codes seems not to be directly connected to linear codes. But one particu-

lar very good superimposed code can be constructed from the extended $[8, 4, 4]$ Hamming code and thus we review the theory of linear codes up to and including Hamming codes. Many of the previously known constructions of superimposed codes are directly constructed from the incidence matrices of $t-$designs. Therefore, these concepts are also discussed in Chapter 3.
While constant weight codes are not directly used for the construction of superimposed codes, every construction of superimposed codes presented in this thesis also turns out to be a constant weight code. For this reason, a very brief introduction to the theory of constant weight codes is given in Chapter 4. Also, in this chapter, a slight motivation is given for why the idea of geometrical constructions was studied.
In Chapter 5 the concept of superimposed code is defined and some well known and useful theorems are presented. Here the topics of linear codes, t-designs and superimposed codes are tied together with theorems and an example.
In Chapter 6 the constructions found during the project are presented. This chapter also contains a short discussion of possible future research topics directly related to the constructions presented.
In Appendix A is the program code that was used to verify the $(2, 2)$ superimposed codes.
The source of all the theory presented in Chapter 2 is [11].
In Chapter 3, the source of Theorem 3.5, Theorem 3.12 and their proofs is [11]. The source of the rest of the theory in Chapter 3 is [10].
In Chapter 4, the source of Theorem 4.1, Theorem 4.2 and the proof of Theorem 4.1 is [10]. The source of Theorem 4.3 and its proof is [5].
In Chapter 5, the source of Theorem 5.2 and Theorem 5.3 and the proof of Theorem 5.3 is [9]. Example 5.4 can be found as Lemma 5.3 in [7].

Chapter 2

Finite Fields

The theory in this chapter mainly deals with the construction of fields. The goal of this chapter is to describe some constructions of fields, as well as to briefly describe the structure of Galois fields.

2.1 Introduction to Ring and Field Theory

In coding theory, Galois fields are a powerful tool for constructing codes and for proving properties of codes. Because of this, we here give a brief review of the field theory needed in this thesis.

Definition 2.1. A ring $(R, +, *)$ is a set R with two operations $+$ and $*$, called addition and multiplication, such that for any $a, b, c \in R$

$$a + b = b + a,$$
$$a + (b + c) = (a + b) + c, a * (b * c) = (a * b) * c,$$
$$a * (b + c) = (a * b) + (a * c),$$

and there exists an element $0 \in R$ and for every $a \in R$ there exists $-a \in R$ such that

$$a + 0 = 0 + a = a, \text{and}$$
$$a + (-a) = 0.$$

Even though the proper notation for a ring is of the form $(R, +, *)$, we will for convenience often use the shorter notation of R unless this may cause confusion. A commutative ring R is a ring in which multiplication is commutative, that is, when $a, b \in R$, we have $a * b = b * a$. A ring R with unity is a ring in which there exists an element 1 such that for any $a \in R$, we have $a * 1 = 1 * a = a$. A commutative ring F with unity is called a *field* if for any nonzero element $a \in F$ there exists $a^{-1} \in F$ such that $a * a^{-1} = 1$. A finite field will be referred to as a *Galois field*. For convenience, and to make expressions easier to read, we will generally write ab instead of $a * b$. The *order* of a ring R is the size of the ring. If R is a ring with unity, the *characteristic* of R is

defined to be the least positive integer c such that 1 added to itself c times equals 0. If no such c exists, the characteristic is said to be $-\infty$.
A simple example of a field is the real numbers $\mathbb{R}$ with addition and multiplication. The real numbers have infinitely many elements, but for our purposes we generally need Galois fields. An example of a Galois field is the set $\{0, 1\}$ with addition and multiplication done modulo 2.

Definition 2.2. Let F and G be fields. A bijective function $\phi : F \to G$ is called an *isomorphism* if

$$\phi(a + b) = \phi(a) + \phi(b)$$
$$\phi(a * b) = \phi(a) * \phi(b)$$

is true for any $a, b \in F$. If an isomorphism exists, F and G are said to be *isomorphic* and we denote this by $F \cong G$.

Galois fields have many nice properties but we will not provide proofs for them. For example, the number of elements in a Galois fields must be p^n where p is a prime number and n a natural number. Furthermore, there exists only *one* Galois field of order p^n in the sense that if we have two seemingly different fields of order p^n then they are isomorphic. Thus we can say *the* Galois field of order p^n. This field is denoted $GF(p^n)$. It can be proved that for $GF(p^n)$, the characteristic is p.

Definition 2.3. Let $(R, +, *)$ be a ring and let $S \subseteq R$. $(S, +, *)$ is a *subring* of $(R, +, *)$, denoted $(S, +, *) \leq (R, +, *)$ or $S \leq R$, if $(S, +, *)$ is a ring with the operations addition and multiplication of $(R, +, *)$ restricted to the set S. If S is a proper subset of R we may use the notation $(S, +, *) < (R, +, *)$. Similarily, let $(F, +, *)$ be a field and $G \subseteq F$. If $(G, +, *)$ is a field with the operations addition and multiplication of $(F, +, *)$ restricted to the set G, we say that $(G, +, *)$ is a *subfield* of $(F, +, *)$, also denoted $(G, +, *) \leq (F, +, *)$. Also, if G is a proper subset of F we may again use the notation $(G, +, *) < (F, +, *)$.

Definition 2.4. Let R be a ring. A subring I of R is said to be an *ideal* if and only if $ar, ra \in I$ for all $a \in I$ and all $r \in R$.

Example 2.5. The integers $\mathbb{Z}$ with addition and multiplication defined the usual way forms a ring. All subrings, and thus all ideals, of $\mathbb{Z}$ are of the form $n\mathbb{Z} = \{nz : z \in \mathbb{Z}\}$ where n is an integer. Furthermore, if $a \in n\mathbb{Z}$ and $r \in \mathbb{Z}$ then $ar = nzr \in n\mathbb{Z}$ for some $z \in \mathbb{Z}$ and thus $n\mathbb{Z}$ is an ideal.

To describe the following construction of rings properly, we need the notion of *coset* from group theory. If G is a group, H is a subgroup of G, and $g \in G$, then the set $gH = \{gh : h \in H\}$ is called the left coset of H with respect to g and $Hg = \{hg : h \in H\}$ is called the right coset of H with respect to g. If g_1H and g_2H contain the same elements, that is if $g_1H = g_2H$, then we consider g_1H and g_2H to be the same coset. Given a coset g_1H, then for any $g_2 \in g_1H$ the coset g_2H is the same as g_1H. For all

groups dealt with in this thesis, the left coset gH is the same as the right coset Hg. When $gH = Hg$ the subgroup H is said to be a *normal* subgroup and then we need not specify whether a coset is a left coset or right coset. Of course, if G is an abelian group then $gH = Hg$, and thus all subgroups of G are normal. It can be proved that all cosets of a given normal subgroup H are pairwise disjoint and that their union is equal to the group G. In the remainder of this thesis we will only deal with normal subgroups, thus we will say coset instead of left or right coset. Let G/H denote the set of all cosets of H. If we define the operation $*$ in G/H by $(aH) * (bH) = (ab)H$, we get the *quotient group* of G modulo H. In a quotient group, the elements are the cosets of some other group. To make notation simpler, we may choose an element in each coset that represents that coset. If a, b and ab are representatives of the cosets aH, bH and abH, respectively, we may write $a * b = ab \pmod{H}$ instead of writing $(aH) * (bH) = (ab)H$. Since rings are groups under addition, the notion of coset carries over to rings in a natural way. If F is a field, we may refer to $(F, +)$ as the additive group of F and $(F \setminus \{0\}, *)$ as the multiplicative group of F. The multiplicative group of F is denoted by F^*.

Example 2.6. Let $\mathbb{Z}$ be the ring of integers and $5\mathbb{Z}$ an ideal in $\mathbb{Z}$. $5\mathbb{Z}$ is a subgroup of the additive group of Z and $\mathbb{Z}/5\mathbb{Z} = \{0 + 5\mathbb{Z}, 1 + 5\mathbb{Z}, 2 + 5\mathbb{Z}, 3 + 5\mathbb{Z}, 4 + 5\mathbb{Z}\}$ is the quotient group. Here, we have chosen $0, 1, 2, 3$ and 4 to be representatives of the cosets, but we could have chosen many others. For the coset $1 + 5\mathbb{Z} = \{1 + z : z \in 5\mathbb{Z}\} = \{1 + 5n : n \in \mathbb{Z}\}$ we can choose any element $1 + 5n$ with $n \in \mathbb{Z}$ to be a representative. But with the choice of representatives being $0, 1, 2, 3$ and 4 we see that this looks a lot like the group of integers with addition done modulo 5, which we denote $\mathbb{Z}_5$. In fact, $\mathbb{Z}/5\mathbb{Z}$ and $\mathbb{Z}_5$ are isomorphic. Just like the group $\mathbb{Z}_5$ can be provided an extra operation, the operation multiplication, to become a ring, the group $\mathbb{Z}/5\mathbb{Z}$ can similarly be made into a ring with addition and multiplication of cosets.

The following theorem, along with its proof, can be found in [11].

Theorem 2.7. *Let R be a ring and I an ideal in R then R/I is a ring under addition and multiplication modulo I, that is, addition and multiplication of cosets of I.*

Definition 2.8. Let R be a ring. An ideal $M \neq R$ in R is said to be a *maximal ideal* if there exists no ideal I such that $M \subset I \subset R$.

Example 2.9. Both $6\mathbb{Z}$ and $3\mathbb{Z}$ are ideals in $\mathbb{Z}$. Since $6\mathbb{Z} \subset 3\mathbb{Z}$, $6\mathbb{Z}$ is not a maximal ideal. However, there exists no ideal $n\mathbb{Z} \neq \mathbb{Z}$ such that $3\mathbb{Z} \subset n\mathbb{Z}$, so $3\mathbb{Z}$ is a maximal ideal in $\mathbb{Z}$.

The following theorem will be important for further constructions of fields. A proof of the theorem can be found in [11].

Theorem 2.10. *Let R be a commutative ring with unity and M an ideal in R. M is a maximal ideal if and only if the quotient ring R/M is a field.*

Since $3\mathbb{Z}$ is a maximal ideal in $\mathbb{Z}$, the quotient ring $\mathbb{Z}/3\mathbb{Z}$ is in fact a field.

2.2 Polynomial Rings

Consider the field $\mathbb{Z}_2$ and denote by $\mathbb{Z}_2[x]$ all polynomials with coefficients in $\mathbb{Z}_2$. $\mathbb{Z}_2[x]$ is a ring, not a field. In the previous section we saw how a field can be constructed from a ring using a maximal ideal. In this section we will see how such ideals can be found in polynomial rings. To do this we begin with the following result, which states that all ideals in a polynomial ring are in some sense similar to each other. The source of this theorem and its proof is [11].

Theorem 2.11. *Let F be a field. Then all ideals in $F[x]$ is of the form $\langle g(x)\rangle = \{g(x)h(x) : h(x) \in F[x]\}$ for some $g(x) \in F[x]$.*

Proof. Let I be an ideal in $F[x]$. If $I = \{0\}$ is the trivial ideal, then $I = \langle 0\rangle$. So in the remainder of the proof, we assume that I contain at least one nonzero polynomial. Let $g(x) \neq 0$ be a polynomial with minimal degree in I. We want to show that all polynomials in I is a multiple of $g(x)$. Let $f(x) \in I$ be arbitrary. The division algorithm gives $f(x) = g(x)q(x) + r(x)$ for some $q(x), r(x) \in F[x]$, where $r(x) = 0$ or $\deg r(x) < \deg g(x)$. Since $f(x)$ and $g(x)$ belongs to I, and I is an ideal, then $r(x) = f(x) - g(x)q(x)$ must also belong to I. Since $g(x)$ has minimal degree in I, we cannot also have $\deg r(x) < \deg g(x)$. Thus, $r(x) = 0$ and we have $f(x) = g(x)q(x)$. Q.E.D.

If $I = \langle g(x)\rangle$ is an ideal, it is not unreasonable to think that the properties of the polynomial $g(x)$ may carry over into properties of the ideal. One property that a polynomial might have that decides some properties for the corresponding ideal, is whether or not the polynomial can be factorized. Polynomials that cannot be factorized plays a similar role for polynomial rings as prime numbers does in $\mathbb{Z}$. This will be very important for the construction of Galois fields, as the following example suggests.

Example 2.12. The quotient ring $\mathbb{Z}_2[x]/\langle x^3 + x + 1\rangle$ is a field. The multiplicative identity is the constant polynomial 1 and below is the elements with their multiplicative inverses.

$p(x)$	$p^{-1}(x)$
0	undefined
1	1
x	$x^2 + 1$
$x + 1$	$x^2 + x$
x^2	$x^2 + x + 1$
$x^2 + 1$	x
$x^2 + x$	$x + 1$
$x^2 + x + 1$	x^2

Obviously multiplication in $\mathbb{Z}_2[x]/\langle x^3 + x + 1\rangle$ is commutative. Thus $\mathbb{Z}_2[x]/\langle x^3 + x + 1\rangle$ is a field of order $2^3 = 8$.

In the above example, the polynomial x^3+x+1 was chosen carefully. If x^3+1 was used, the resulting quotient ring would not be a field. The important property that x^3+x+1 has but x^3+1 has not, is that x^3+x+1 cannot be factorized as the product of two polynomials in $\mathbb{Z}_2[x]$ of lower degree than 3. Unlike x^3+x+1, x^3+1 can be factorized as the product of two lower degree polynomials, namely $x^3+1=(x+1)\left(x^2+x+1\right)$. The polynomials $x+1$ and x^2+x+1 cannot, however, be written as the product of two lower degree polynomials in $\mathbb{Z}_2[x]$. This is the property that we need to create fields from polynomial rings. These polynomials are what is called *irreducible*, which is defined as follows.

Definition 2.13. Let F be a field and $F[x]$ a polynomial ring over F. A polynomial $f(x) \in F[x]$ of degree greater than 1 is said to be *irreducible* over F if there exists no polynomials $g(x), h(x) \in F[x]$, both with lower degree than $f(x)$, such that $f(x) = g(x)h(x)$.

The following theorem is important for our construction of fields, and its proof can be found in [11].

Theorem 2.14. *Let F be a field and $f(x)$ polynomial in $F[x]$. Then the quotient ring ${}^{F[x]}/_{\langle f(x)\rangle}$ is a field if and only if $f(x)$ is irreducible over F.*

The following theorem states an important property of Galois fields, and the source of theorem and its proof is [11].

Theorem 2.15. *The multiplicative group $GF^*(p^n)$ of a Galois field $GF(p^n)$ is a cyclic group.*

Proof. It is obvious that $GF^*(p^n)$ is a finite abelian group under multiplication. According to the structure theorem for finite abelian groups, every finite abelian group is isomorphic with the direct product of some primary cyclic groups. So we know that

$$GF^*(p^n) \cong \mathbb{Z}_{d_1} \times \mathbb{Z}_{d_2} \times \cdots \times \mathbb{Z}_{d_r} \tag{2.1}$$

where $d_i = {p_i}^{n_i}$ and p_i is a prime number and $n_i \in \mathbb{N}$. What we want to show is that the direct product on the right-hand side of 2.1 is a cyclic group. We will do this by showing that all d_i are pairwise relatively prime.
Let $m = \mathrm{LCM}\,(d_1, d_2, \ldots, d_r)$. Then $m \leq d_1 d_2 \cdots d_r = |GF^*(p^n)|$. If $a_i \in \mathbb{Z}_{d_i}$ then $a_i^{d_i} = 1$. Since d_i divides m we also have $a_i^m = 1$. Consequently we have

$$(a_1, a_2, \ldots, a_r)^m = (a_1^m, a_2^m, \ldots, a_r^m) = (1, 1, \ldots, 1)$$

for all $(a_1, a_2, \ldots, a_r) \in \mathbb{Z}_{d_1} \times \mathbb{Z}_{d_2} \times \cdots \times \mathbb{Z}_{d_r}$. Thus all elements in $GF^*(p^n)$ is a root of the polynomial $x^m - 1 \in GF(p^n)[x]$. The polynomial $x^m - 1$ can have at most m roots in $GF(p^n)$. Since all elements in $GF^*(p^n)$ are roots of $x^m - 1$, we have

$$m \geq |GF^*(p^n)| = d_1 d_2 \cdots d_r.$$

Thus, we have

$$m = d_1 d_2 \cdots d_r.$$

But now we have $d_1 d_2 \cdots d_r = \text{LCM}\,(d_1, d_2, \ldots, d_r)$ which implies that all d_i are pairwise relatively prime. Since all d_i are pairwise relatively prime, we have $\mathbb{Z}_{d_1} \times \mathbb{Z}_{d_2} \times \cdots \times \mathbb{Z}_{d_r} \cong \mathbb{Z}_{d_1 d_2 \cdots d_r}$, which is a cyclic group. Q.E.D.

An element α in a Galois field $GF(p^n)$ is said to be a *primitive element* of $GF(p^n)$ if $\langle \alpha \rangle = GF^*(p^n)$. A primitive element is guaranteed to exist, since every finite cyclic group have at least one generator α. Every nonzero element in $GF(p^n)$ can thus be written as α^k for some integer k. When powers of α is a useful notation of the elements in the field and we need to use 0, we use the notation $\alpha^{-\infty}$ for the zero element.

2.3 Field Extensions

In the previous section we saw how polynomial rings can be made into fields by using modular arithmetic. In some sense, what we did in the previous section was to shrink a ring into a structure that meets the requirements of a field. By figuring out a way in which certain elements in the ring can be considered equivalent, we could in some sense remove a lot of elements in the ring until the structure became a field. The way in which elements were considered equivalent is that the elements belong to the same residue class modulo some fixed polynomial. What we do in this section is in a way the opposite. In this section we describe a way in which we can *extend* an existing field into a bigger field by *adjoining* an element that was not previously in the initial field.

Definition 2.16. If F and G are fields such that $F \leq G$, then G is said to be a *field extension* of F. If $G = \{a + b\alpha : a, b \in F\}$ for some $\alpha \neq F$, then G is said to be constructed from F by *adjoining* the element α and this is denoted by $F(\alpha) = G$.

Example 2.17. The field of complex numbers $\mathbb{C}$ is a field extension of the field of real numbers $\mathbb{R}$. The complex numbers is obtained by adjoining the element i to $\mathbb{R}$, that is, $\mathbb{C} = \mathbb{R}(i)$.

It is well known that the imaginary unit i can be defined as a root to the irreducible polynomial $x^2 + 1 \in \mathbb{R}[x]$. At first it may not be obvious how the element to be adjoined to a field is obtained. Any element that satisfies the axioms of a field but is not already in the field, will do the job. A way to guarantee that this is the case is to define the element as a root of a polynomial which is irreducible over the field, because irreducible polynomials have no roots in the field. This enables us to create fields in a similar way to how it was done in the previous section; in essence, all we need to do is finding an irreducible polynomial.

Example 2.18. Consider the field $\mathbb{Q}$ of rational numbers and the polynomial ring $\mathbb{Q}[x]$. The polynomial $x^2 - 3$ is irreducible over $\mathbb{Q}$. We can define $\sqrt{3}$ as a root of $x^2 - 3$ and extend $\mathbb{Q}$ to $\mathbb{Q}(\sqrt{3})$. The polynomial $x^2 - 3$ is not irreducible over $\mathbb{Q}(\sqrt{3})$, since $x^2 - 3 = \left(x + \sqrt{3}\right)\left(x - \sqrt{3}\right)$.

In Chapter 5 and Chapter 6, Galois fields will be used to construct superimposed codes. So, let us see an extension of a Galois field.

Example 2.19. For a prime number p a Galois field $GF(p)$ can be constructed. A simple example of such a field is $\mathbb{Z}_p$ with arithmetic done modulo p. To create $GF(p^n)$ we will adjoin a root of an irreducible polynomial of order n. For simplicity, let $p = 2$ and $n = 3$. To extend $GF(2)$ to $GF\left(2^3\right)$ we can use the polynomial $x^3 + x + 1$ which is irreducible over $GF(2)$. So let α be a root of $x^3 + x + 1$. Now $GF(2)$ can be extended to $GF\left(2^3\right) = \left\{g_0 + g_1\alpha + g_2\alpha^2 + g_3\alpha^3 + \cdots : g_0, g_1, g_2, g_3, \ldots \in GF(2)\right\}$. But since α is a root of $x^3 + x + 1$, $\alpha^3 = \alpha + 1$. Thus α^k, $k \in \mathbb{Z}$, can always be written as a sum of powers of α where all powers are strictly less than 3. For example, $\alpha^4 = \alpha\left(\alpha + 1\right)$. Finally, we have $GF\left(2^3\right) = \left\{g_0 + g_1\alpha + g_2\alpha^2 : g_0, g_1, g_2 \in GF(2)\right\}$.

The Galois field constructed in the above example looks very similar to the polynomial field constructed in Example 2.12. Indeed these two fields are isomorphic, but one is constructed as a quotient ring and the other is constructed as a field extension.

Chapter 3

An Introduction to Codes

3.1 General Concepts

In this section we introduce the notion of codes. Codes are interesting for several reasons. A common application of codes is error correction in communication systems. While error correcting codes are not the focus of this thesis, the codes discussed can still be used to perform error correction. The codes that are of main interest in this thesis are interesting for their combinatoric properties, rather than their error correcting capabilities.

Definition 3.1. Let $\mathbb{B}$ be a set of symbols and n a natural number. A *code* C is a nonempty subset of $\mathbb{B}^n$. The elements of $\mathbb{B}^n$ are called *words* and the elements of $\mathcal{C}$ are called *codewords*.

If $\boldsymbol{x}$ is a word in $\mathbb{B}^n$ we write $\boldsymbol{x} = x_1 x_2 \dots x_n$, where each x_i represents one symbol. In this thesis we will deal with binary codes, that is, the case $\mathbb{B} = \{0, 1\}$.

Example 3.2. Let $\mathbb{B}^4$ be the set of all binary words with four symbols. The code $\mathcal{C}$ of all words with even number of ones is $\mathcal{C} = \{0000, 0011, 0101, 0110, 1100, 1010, 1001, 1111\}$.

Definition 3.3. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{B}^n$. The *Hamming distance,* or just *distance,* $\operatorname{dist}(\boldsymbol{x}, \boldsymbol{y})$ between $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined to be the number of positions where $x_i \neq y_i$. Similarly, define the *Hamming weight,* or just *weight,* to be $\operatorname{wt}(\boldsymbol{x}) = \operatorname{dist}(\boldsymbol{x}, \mathbf{0})$, where $\mathbf{0}$ is the word with 0 in all positions.

Note that the notion of Hamming distance and Hamming weight is defined in the set of *all* words, not only in a given code. Therefore the Hamming distance and Hamming weight functions can be used on words regardless of which code the words belong to.

Example 3.4. Let the set of all words be $\mathbb{B}^4$. For the words $0000, 0110, 1001$, and 1111 we have $\operatorname{wt}(0110) = \operatorname{dist}(0110, 0000) = 2, \operatorname{wt}(1111) = 4$, and $\operatorname{dist}(0110, 1001) = 4$.

The next theorem provides a poverful tool for proving theorems and solving problems in coding theory. The presented proof of the theorem can also be found in [10].

Theorem 3.5. *For all* $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{B}^n$ *the following is true*

1. $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ *with equality if and only if* $\boldsymbol{x} = \boldsymbol{y}$
2. $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = \text{dist}(\boldsymbol{y}, \boldsymbol{x})$
3. $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) \leq \text{dist}(\boldsymbol{x}, \boldsymbol{z}) + \text{dist}(\boldsymbol{y}, \boldsymbol{z})$

Proof. Statements 1 and 2 are trivial, so we will only prove statement 3. Let $\boldsymbol{x} = x_1 x_2 \dots x_n, \boldsymbol{y} = y_1 y_2 \dots y_n$, and $\boldsymbol{z} = z_1 z_2 \dots z_n$.
First, suppose $x_i \neq z_i$ for some i, then position i will contribute 1 to $\text{dist}(\boldsymbol{x}, \boldsymbol{z})$. Now either $x_i = 0$ and $z_i = 1$, or $x_i = 1$ and $z_i = 0$. Since y_i is either 0 or 1, y_i has to equal one of x_i and z_i, but differ from the other. Thus position i will contribute 1 to either $\text{dist}(\boldsymbol{x}, \boldsymbol{y})$ or $\text{dist}(\boldsymbol{z}, \boldsymbol{y})$, but not both. That is, position i will contribute 1 to both sides of statement 3, or position i will contribute 0 to the left-hand side and 2 to the right-hand side.
Now suppose $x_i = z_i$ for some i, then position i will contribute 0 to $\text{dist}(\boldsymbol{x}, \boldsymbol{z})$. If, further, $x_i = y_i = z_i$, then position i will also contribute 0 to both $\text{dist}(\boldsymbol{x}, \boldsymbol{y})$ and $\text{dist}(\boldsymbol{z}, \boldsymbol{y})$. If, on the other hand, $x_i = z_i \neq y_i$, then position i will contribute 1 to both $\text{dist}(\boldsymbol{x}, y)$ and $\text{dist}(\boldsymbol{z}, y)$. That is, position i will always contribute the same amount to both sides of statement 3.
Thus, any position i will contribute the same amount to both sides of statement 3, or more to the right-hand side than to the left-hand side. Q.E.D.

Definition 3.6. The *minimum distance* d of a code $\mathcal{C}$ is defined to be

$$d = \min\{\text{dist}(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}, \boldsymbol{x} \neq \boldsymbol{y}\}$$

It may not be immediately obvious why the notion of minimum distance is important, but it is possibly the most important property for deciding the error correcting capabilities of a code. As stated before, this thesis is not about error correction, so the details of the role minimum distance plays in error correction will not be discussed. The concept of minimum distance is still important, though, but will be used mainly for proving theorems.

3.2 Linear Codes

Thus far, we have not assigned any properties to the individual symbols or words that we are dealing with. To develop the theory further, however, we need to do that. If the size of the set of symbols we are dealing with is a prime power, we can in a natural way consider the set to be a finite field. In our case, with $\mathbb{B} = \{0, 1\}$, we can consider $\mathbb{B}$ to be a finite field with addition and multiplication done in the usual way mudulo 2. This enables us to consider $\mathbb{B}^n$ as an $n-$dimensional vector space over the finite field $\mathbb{B}$. In the remaining part of the thesis, the terms *vector* and *word* will be used interchangeably to describe elements in $\mathbb{B}^n$. Addition of vectors in $\mathbb{B}^n$ is done component by component, $\boldsymbol{x}+\boldsymbol{y} = (x_1, x_2, \cdots, x_n)+(y_1, y_2, \cdots, y_n) = (x_1 + y_1, x_2 + y_2, \cdots, x_n + y_n)$ and multiplication by scalar is done as follows, $r\boldsymbol{x} = (rx_1, rx_2, \cdots, rx_n)$.

Definition 3.7. A code $\mathcal{C}$ is said to be *linear* if $\mathcal{C}$ is a subspace of $\mathbb{B}^n$.

Since any linear code $\mathcal{C}$ is a subspace of the $n-$dimensional vector space $\mathbb{B}^n$, the dimension of $\mathcal{C}$ is at most n. If the dimension of $\mathcal{C}$ is k, we say that $\mathcal{C}$ is an $[n,k]$ code. If $\mathcal{C}$ has minimum distance d, we may emphasize this by saying that $\mathcal{C}$ is an $[n,k,d]$ code. Codewords in a linear $[n,k]$ code can be said to have k symbols of information, while the remaining $n-k$ symbols are used for other purposes, often to detect or correct errors.
If $\mathcal{C}$ is an $[n,k]$ code, there exists a basis for $\mathcal{C}$ of k linearly independent vectors. If $\boldsymbol{G}$ is a $k \times n$ matrix whose rows are all the vectors of a given basis for $\mathcal{C}$, then we can conveniently describe $\mathcal{C}$ as $\mathcal{C} = \left\{\boldsymbol{x}\boldsymbol{G} : \boldsymbol{x} \in \mathbb{B}^k\right\}$. The matrix $\boldsymbol{G}$ may be used as a convenient way to construct the code. If $\boldsymbol{G}$ has the form $\boldsymbol{G} = [\boldsymbol{I}_k|\boldsymbol{A}]$, where $\boldsymbol{I}_k$ is the $k \times k$ identity matrix and $\boldsymbol{A}$ is some fixed $k \times (n-k)$ matrix, then the encoded codewords will have the property that the first k symbols of a codeword is identical to the word that is encoded.

Definition 3.8. Let $\mathcal{C}$ be an $[n,k]$ code. A $k \times n$ matrix $\boldsymbol{G}$ whose rows are all the vectors of a given basis for $\mathcal{C}$ is called a *generator matrix* for $\mathcal{C}$. If $\boldsymbol{G} = [\boldsymbol{I}_k|\boldsymbol{A}]$ for some $k \times (n-k)$ matrix $\boldsymbol{A}$, then we call $\boldsymbol{G}$ a *standardized generator matrix.*

Example 3.9. Consider the code $\mathcal{C}$ defined by the standardized generator matrix

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

After some calculations, we get $\mathcal{C} = \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\}$. $\mathcal{C}$ is the code of all even weight words of length 4.

Next, we will define another useful matrix.

Definition 3.10. Let $\mathcal{C}$ be a linear $[n,k]$ code. An $(n-k) \times n$ matrix $\boldsymbol{H}$ with the property that $\boldsymbol{H}\boldsymbol{x}^{tr} = \boldsymbol{0}$ for all $\boldsymbol{x} \in \mathcal{C}$ is called a *parity check matrix.* If $\boldsymbol{H} = [\boldsymbol{A}|\boldsymbol{I}_{n-k}]$ for some $(n-k) \times k$ matrix $\boldsymbol{A}$, then we call $\boldsymbol{H}$ a *standardized parity check matrix.*

Both a parity check matrix and a generator matrix can be used to define a linear code and they are both useful in applications. When using a linear code for error correction, it is natural to use a generator matrix for encoding. Before decoding an encoded word, we want to know if the word is corrupted or not. We know that the linear code contains all codewords that equals zero when multiplying by the parity check matrix. So to check if the word to be decoded is corrupt or not, we can multiply the word by the parity check matrix. If the product is zero we assume the codeword was not corrupted, and if the product is nonzero we know that the codeword was corrupted. It is also possible to use the parity check matrix to correct errors, but that is not important for our work so we will not discuss that here. However, a parity check matrix will be used for a very convenient definition of a code that is important for this thesis, so the concept of parity check matrix is still important.

Example 3.11. In Example 3.9 we defined the even weight code of length 4 using the standardized generator matrix

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

A corresponding parity check matrix is the standardized parity check matrix $\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$. We can now easily check whether a word is a codeword or not. For example, 0110 is a codeword but 1110 is not, since

$$\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = 0, \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = 1.$$

The following theorem states a correspondence between standardized generator and parity check matrices. The proof provided for the theorem can be found in [11].

Theorem 3.12. *Let $\mathcal{C}$ be a linear $[n, k]$ code defined by the standardized generator matrix $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I_k} | -\boldsymbol{A^{tr}} \end{bmatrix}$. Then the matrix $\boldsymbol{H} = [\boldsymbol{A}|\boldsymbol{I}_{n-k}]$ is a standardized parity check matrix for $\mathcal{C}$.*

Proof. Let $\mathcal{K} = \left\{ \boldsymbol{x} \in \mathbb{B}^n : \boldsymbol{H}\boldsymbol{x}^{tr} = \boldsymbol{0} \right\}$. We are going to prove that $\mathcal{K} \subseteq \mathcal{C}$ and $\mathcal{C} \subseteq \mathcal{K}$. To this end, suppose $\boldsymbol{u} \in \mathcal{C}$, that is, $\boldsymbol{u} = \boldsymbol{v}\boldsymbol{G}$ for some $\boldsymbol{v} \in \mathbb{B}^k$. Now we have

$$\boldsymbol{H}\boldsymbol{u}^{tr} = \boldsymbol{H}(\boldsymbol{v}\boldsymbol{G})^{tr} = \boldsymbol{H}\left(\boldsymbol{G}^{tr}\boldsymbol{v}^{tr}\right) = \left(\boldsymbol{H}\boldsymbol{G}^{tr}\right)\boldsymbol{v}.$$

Written in blockform, the product $\boldsymbol{H}\boldsymbol{G}^{tr}$ is

$$\boldsymbol{H}\boldsymbol{G}^{tr} = [\boldsymbol{A}|\boldsymbol{I}_{n-k}] \begin{bmatrix} \boldsymbol{I}_k{}^{tr} \\ -\left(\boldsymbol{A}^{tr}\right)^{tr} \end{bmatrix} = [\boldsymbol{A}|\boldsymbol{I}_{n-k}] \begin{bmatrix} \boldsymbol{I}_k \\ -\boldsymbol{A} \end{bmatrix} = \boldsymbol{A}\boldsymbol{I}_k + \boldsymbol{I}_{n-k}(-\boldsymbol{A}) = \boldsymbol{A} - \boldsymbol{A} = \boldsymbol{0}$$

So if $\boldsymbol{u}$ is a codeword, then $\boldsymbol{H}\boldsymbol{u}^{tr} = \boldsymbol{0}$; thus $\mathcal{C} \subseteq \mathcal{K}$. To show that $\mathcal{K} \subseteq \mathcal{C}$, let $\boldsymbol{y} \in \mathcal{K}$. It will be useful to write $\boldsymbol{y}$ in the form $\boldsymbol{y} = [\boldsymbol{x}|\boldsymbol{z}]$ where $\boldsymbol{x} \in \mathbb{B}^k$ and $\boldsymbol{z} \in \mathbb{B}^{n-k}$. Since $\boldsymbol{y} \in \mathcal{K}$, we have

$$\boldsymbol{0} = \boldsymbol{H}\boldsymbol{y}^{tr} = [\boldsymbol{A}|\boldsymbol{I}_{n-k}] \begin{bmatrix} \boldsymbol{x}^{tr} \\ \boldsymbol{z}^{tr} \end{bmatrix} = \boldsymbol{A}\boldsymbol{x}^{tr} + \boldsymbol{I}_{n-k}\boldsymbol{z}^{tr} = \boldsymbol{A}\boldsymbol{x}^{tr} + \boldsymbol{z}^{tr}$$

which yields $\boldsymbol{z}^{tr} = -\boldsymbol{A}\boldsymbol{x}^{tr}$. Now we have

$$\boldsymbol{y} = [\boldsymbol{x}|\boldsymbol{z}] = \left[\boldsymbol{x}|\left(-\boldsymbol{A}\boldsymbol{x}^{tr}\right)^{tr}\right] = \left[\boldsymbol{x}|-\boldsymbol{x}\boldsymbol{A}^{tr}\right] = \boldsymbol{x}\left[\boldsymbol{I}_k|-\boldsymbol{A}^{tr}\right] = \boldsymbol{x}\boldsymbol{G}.$$

Since $\boldsymbol{y}$ is an arbitrary element in $\mathcal{K}$, this implies $\mathcal{K} \subseteq \mathcal{C}$. We conclude that $\mathcal{C} = \mathcal{K}$.

Q.E.D.

An important class of linear codes are the Hamming codes.

Definition 3.13. A linear $[n = 2^r - 1, 2^r - 1 - r, d = 3]$ code is called a *Hamming code* and has a parity check matrix whose columns are all possible nonzero binary vectors of length r. Denote this code by $\mathcal{H}_r$.

We have not yet come to the main topic of this thesis, which is superimposed codes. As we will later see, superimposed codes are difficult to construct. But by clever use of Hamming codes, we can construct some very good superimposed codes. We have already seen some Hamming codes in earlier examples, but here is another example.

Example 3.14. For $r = 3$ we have the $[7, 4, 3]$ Hamming code. The columns of a parity check matrix of the $[7, 4, 3]$ Hamming code are all possible nonzero binary vectors of length 3. The following matrix $\boldsymbol{H}_3$ is a standardized parity check matrix for the $[7, 4, 3]$

$$\boldsymbol{H}_3 = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

The corresponding standardized generator matrix $\boldsymbol{G}_3$ is

$$\boldsymbol{G}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

The codewords in the Hamming code $\mathcal{H}_3$ defined by these two matrices are

0000000, 1000111
0001011, 1001100
0010101, 1010010
0011110, 1011001
0100110, 1100001
0101101, 1101010
0110011, 1110100
0111000, 1111111.

There are multiple common techniques for constructing new codes from old codes. One such technique that will be important for our work is to add an over all parity bit to a code. Let $\mathcal{C}$ be an $[n, k, d]$ code that contains words of both odd and even weight. We can create a new *extended* code $\hat{\mathcal{C}}$ by adding a 1 at the end of every codeword of odd weight in $\mathcal{C}$ and a 0 at the end of every codeword of even weight in $\mathcal{C}$. The new code $\hat{\mathcal{C}}$ contains only codewords of even weight. The distance between any two even weight codewords has to be even. Thus, if $\mathcal{C}$ has odd minimum distance d, $\hat{\mathcal{C}}$ will have minimum distance $d + 1$.
One important code that will turn up again later is the code constructed by extending the $[7, 4, 3]$ Hamming code. This code is shown in the next example.

Example 3.15. Let $\boldsymbol{H}_3$ be the parity check matrix for the $[7,4,3]$ Hamming code from the previous example. The $[8,4,4]$ extended Hamming code $\hat{\mathcal{H}}_3$ has the the following parity check matrix $\boldsymbol{H}$

$$\boldsymbol{H} = \begin{bmatrix} 1 & \cdots & 1 \\ & & 0 \\ & \boldsymbol{H}_3 & \vdots \\ & & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The codewords in the $[8,4,4]$ extended Hamming code defined by this parity check matrix are

00000000, 10001110
00010111, 10011001
00101011, 10100101
00111100, 10110010
01001101, 11000011
01011010, 11010100
01100110, 11101000
01110001, 11111111.

3.3 Nonlinear Codes

Linear codes have many advantages over nonlinear codes. Since linear codes are vector spaces they have many nice properties, and they are easy to encode and decode. In some respects, however, there exist better codes. If it is important to have as many codewords as possible in some application, then linear codes are not always the best choice. Hence, the development of the theory of nonlinear codes.

Definition 3.16. A code $\mathcal{C}$ of M codewords of length n with minimum distance d is denoted by (n, M, d).

As for linear codes, we can consider the codewords of a nonlinear code to be vectors in a vector space. The nonlinear code itself is not a vector space, but it is a subset in the vector space $\mathbb{B}^n$. This allows us to use the notion of Hamming distance and Hamming weight in nonlinear codes as well.
Since we are not studying nonlinear codes for the sake of studying nonlinear codes, we will restrict our focus to just the set of nonlinear codes that will be useful to us. Basically, this means that we will restrict our focus to $t-$designs and incidence matrices.

Definition 3.17. Let X be a set with $|X| = v$. The elements of X are called *points*. A $t-(v,k,\lambda)$ *design* is a pair (X,B) where B is a set whose elements are $k-$subsets (sets of size k) of X called *blocks*, such that any $t-$subset of X is a subset of exactly λ blocks.

Definition 3.18. Given a $t-(v,k,\lambda)$ design with the points $P_1, P_2, \ldots, P_v$ and blocks $B_1, B_2, \ldots, B_b$. The *incidence matrix* of this $t-$design is a $v \times b$ matrix $\boldsymbol{A} = (a_{ij})$ defined by

$$a_{ij} = \begin{cases} 1 & \text{if } P_j \in B_i, \\ 0 & \text{if } P_j \notin B_i. \end{cases}$$

The incidence matrix of a given $t-(v,k,\lambda)$ design serves as a good tool to make the fairly abstract concept of a $t-$design into something more concrete. The incidence matrix obviously has v columns. Each row in the incidence matrix has exactly k ones. For any t columns of the incidence matrix there are exactly λ rows where all t columns has a one.

Example 3.19. An example of a $2-(7,3,1)$ design is the set of points $X = \{1, 2, \ldots, 7\}$ together with set of blocks $B = \{124, 235, 346, 457, 561, 672, 713\}$, where 124 denotes the set $\{1,2,4\}$ just to keep it somewhat readable. The following matrix is the incidence matrix of this $t-$design.

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Definition 3.20. A codeword $\boldsymbol{u}$ is said to *cover* a codeword $\boldsymbol{v}$ if $\boldsymbol{u}$ has ones in at least all positions where $\boldsymbol{v}$ has ones.

The following theorem is important since it will allow us to construct a superimposed code from the extended $[8,4,4]$ Hamming code. The source of this theorem and its proof is [10].

Theorem 3.21. *The codewords of weight* 4 *in the extended* $[n = 2^m, k = 2^m - m - 1, d = 3]$ *Hamming code* $\hat{\mathcal{H}}_m$ *forms a* $3-(2^m, 4, 1)$ *design.*

Proof. The codewords of weight 4 in $\hat{\mathcal{H}}_m$ will be blocks in the $t-$design. That is, each codeword of weight 4 in $\hat{\mathcal{H}}_m$ will be a row in the corresponding incidence matrix and each position in the codewords correspond to one column in the incidence matrix. What we want to show is that for any 3 columns in the incidence matrix, there is exactly one row that has ones in those 3 columns. In the terminology of Hamming codes, what we want to show is that given any word $\boldsymbol{u}$ of length n and weight 3, there is exactly one codeword in $\hat{\mathcal{H}}_m$ that covers $\boldsymbol{u}$. Denote the positions in the codewords by $P_1, P_2, \ldots, P_n$. Assume $\boldsymbol{u}$ has ones in position P_h, P_i and P_j with $h < i < j$. There cannot be two codewords of weight 4 in $\hat{\mathcal{H}}_m$ that covers $\boldsymbol{u}$, because then they would have 3 of their

4 ones in common, which requires them to be at distance less than 3; a contradiction. We will look at two cases, $j < n$ and $j = n$.
Assume $j < n$. Then, since $\mathcal{H}_m$ is a perfect single-error-correcting code, there is a unique codeword $\boldsymbol{c} \in \mathcal{H}_m$ at distance at most 1 from $\boldsymbol{u}$. If $\boldsymbol{c} = \boldsymbol{u}$ then the extended codeword $\hat{\boldsymbol{c}} = [\boldsymbol{c}\,|1] \in \hat{\mathcal{H}_m}$ has weight 4 and covers $\boldsymbol{u}$. If $\boldsymbol{c} \neq \boldsymbol{u}$ then $\boldsymbol{c}$ has weight 4 and has 3 ones in common with $\boldsymbol{u}$. Now $\hat{\boldsymbol{c}} = [\boldsymbol{c}\,|0]$ belongs to $\hat{\mathcal{H}_m}$, has weight 4 and covers $\boldsymbol{u}$.
Now assume $j = n$. Let $\boldsymbol{u}'$ have ones in position P_h and P_i. There exists a unique codeword $\boldsymbol{c} \in \mathcal{H}_m$ with weight 3 that is distance 1 away from $\boldsymbol{u}'$ and thus covers $\boldsymbol{u}'$. The extended codeword $\hat{\boldsymbol{c}} = [\boldsymbol{c}\,|1] \in \hat{\mathcal{H}_m}$ covers $\boldsymbol{u}$. Q.E.D.

Chapter 4

Constant Weight Codes

Constant weight codes are of interest to us mainly because all constructions of superimposed codes that later will be presented also turns out to be constructions of constant weight codes. The connection between constant weight codes and superimposed codes will be discussed more in the next chapter.
The main question of interest when studying constant weight codes is how many codewords there exist of a given length n, weight w and minimum distance d. Let $A(n, d, w)$ be a function that to each triple (n, d, w) assigns a value; the value is the maximum number of codewords possible for a code with length n, weight w and minimum distance d. It is not always easy to directly calculate the exact value of $A(n, d, w)$, so we often only have bounds on $A(n, d, w)$.
As described in [5], some constant weight codes can be constructed with geometrical methods. A previous student of my supervisor[1] found a way to use graphs to obtain bounds on $A(n, d, w)$, and sometimes even the exact value. This method is in some ways reminiscent of a construction of superimposed codes that will be presented in Chapter 6.
Below is a theorem that states some basic results on $A(n, d, w)$, and the source of the theorem is [10].

Theorem 4.1. *1.* $A(n, 2d-1, w) = A(n, 2d, w)$

2. $A(n, 2d, w) = A(n, 2d, n-w)$

3. $A(n, 2d, w) = 1$ *if* $w < d$

4. $A(n, 2d, w) = \lfloor \frac{n}{d} \rfloor$

Generally we do not have a formula for $A(n, d, w)$, but if we put some restrictions on the parameters, then some formulas are known, as the next theorem shows. The source of this theorem is also [10].

[1] I am having trouble finding this particular thesis again

Theorem 4.2.

$$A(n,4,4) = \begin{cases} \frac{n(n-1)(n-2)}{4\cdot 3\cdot 2} & \text{if } n \equiv 2 \text{ or } 4 \pmod 6 \\ \frac{n(n-1)(n-3)}{4\cdot 3\cdot 2} & \text{if } n \equiv 1 \text{ or } 3 \pmod 6 \\ \frac{n\left(n^2-3n-6\right)}{4\cdot 3\cdot 2} & \text{if } n \equiv 0 \pmod 6 \end{cases}$$

According to [10] it is unknown what happens for $n \equiv 5 \pmod 6$ in the above theorem. There are several similar formulas presented in [10] and all of them requires the minimum distance d and weight w to be some fixed values. If we want d and w also to be variable, we can for example look at the results in [5]. By using geometrical constructions, exact formulas for $A(n,d,w)$ was found, where n, d and w all depend on some other variable v. In the constructions described in [5], each codeword is represented by a line in a 2-dimensional plane and the components in codewords are represented by points in said plane. If a codeword $\boldsymbol{v}$ has a one in position i then the line that represents $\boldsymbol{v}$ contains the point that represents position i. Using some clever constructions based on this idea, the following two formulas were found in [5].

Theorem 4.3. *For $v \geq 3$,*

$$A\left(\frac{v\left(9v-7\right)}{2}, 6\left(v-1\right), 3v-2\right) = 3v \text{ and}$$
$$A\left(v\left(8v-7\right), 8\left(v-1\right), 4v-3\right) = 4v.$$

The above theorem was actually found as a special case for another formula found in [5], which provided a lower bound on $A(n,d,w)$.
We have already seen a few examples of constant weight codes. In Example 3.19 both the rows and columns of the incidence matrix make up codewords in constant weight codes. In Example 3.15 we constructed the $[8,4,4]$ extended Hamming code $\hat{\mathcal{H}}_3$. If we choose the codewords of weight 4 in $\hat{\mathcal{H}}_3$ we of course get a constant weight code. Also, if we let the codewords of weight 4 in $\hat{\mathcal{H}}_3$ be rows in a matrix, then the columns of the same matrix also make up codewords in a constant weight code.
We will not delve any deeper into the field of constant weight codes. Mainly what we want to see is that geometrical constructions can sometimes be useful when solving certain problems in coding theory and thus we have some motivating examples behind the use of graph constructions that is presented in Chapter 6.

Chapter 5

Superimposed Codes

Definition 5.1. An $N \times T$ matrix $\boldsymbol{C} = (c_{ij})$ is called an (N, T, w, r) superimposed code if for every pair of subsets $W, R \subset \{1, 2, \ldots, T\}$ with $|W| = w$ and $|R| = r$ and $W \cap R = \varnothing$, there exists an $i \in \{1, 2, \ldots, N\}$ such that $c_{ij} = 1$ for $j \in W$ and $c_{ij} = 0$ for $j \in R$.

A simple example of a superimposed code is a matrix $\boldsymbol{C}$ with T columns whose rows are all possible binary vectors of weight w. This code has $N = \binom{T}{w}$ and r can be any positive integer less than $T - w$. This code will be referred to as the trivial (N, T, w, r) superimposed code.
The main goal when studying superimposed codes is to find the minimal N, given T, w and r or to find the maximal T, given N, w and r. A code with said minimal N or maximal T is said to be *optimal*. Superimposed codes are difficult to construct directly, so we will try to find theoretical constructions. Some theoretical constructions are already known, as for example the following theorem, the source of which is [9].

Theorem 5.2. *A $(t+1) - (v, k, 1)$ design is an (N, v, t, r) superimposed code with*

$$N = \frac{\binom{v}{t+1}}{\binom{k}{t+1}} = \frac{(v-t)\binom{v}{t}}{(k-t)\binom{k}{t}} \text{ and } r < \frac{v-t}{k-t}.$$

It was proved in Theorem 3.21 that the codewords of weight 4 in the extended Hamming $[8, 4, 4]$ code is a $3{-}(8, 4, 1)$ design. Using this design, we can construct a $(14, 8, 2, 2)$ superimposed code. In [9] it is also proved that this superimposed code is optimal. Another construction presented in [9] uses the notion of *super-simple* $t{-}$design. A super-simple $t{-}$design is a $t{-}$design in which the intersection of any two blocks has at most t elements. The following theorem is proved in [9].

Theorem 5.3. *A super-simple $t - (v, k, \lambda)$ design is an $(N, v, t, \lambda - 1)$ superimposed code with $N = \frac{\lambda\binom{v}{t}}{\binom{k}{t}}$.*

In [7] it was proved that a $2{-}(v, 4, 3)$ design exists if and only if $v \equiv 0$ or $1 \pmod 4, v \geq 8$. When $v = 9$ Theorem 5.3 says that we have an $(18, 9, 2, 2)$ superimposed code.

A construction of such a $2-(9,4,3)$ design is presented in the next example. It is an interesting construction since it is based on Galois fields. Later, we will generalize this construction of superimposed codes using Galois fields to parameters representing $t-$designs of block size different from 4 and λ different from 3.

Example 5.4. Let $X = GF(3^2)$, $f(x) = x^2 + x + 2$ and α a root of f. Note that f is irreducible over $GF(3)$. Further let $B = \left\{\left(\alpha^0, \alpha^2, \alpha^4, \alpha^6\right), \left(\alpha^1, \alpha^3, \alpha^5, \alpha^7\right)\right\}$ and define $\operatorname{dev} B = \bigcup_{b \in B} \left\{b + g : g \in GF(3^2)\right\}$, where $b + g$ means addition of g to every component of b, just like it is done when constructing cosets. Now X is the set of points of a super-simple $2-(9,4,3)$ design and $\operatorname{dev} B$ is the set of blocks. The incidence matrix of this super-simple $2-(9,4,3)$ is a superimposed code. The blocks, that is the elements of $\operatorname{dev} B$, are listed in the following table

$\langle\alpha^2\rangle$	$\alpha\langle\alpha^2\rangle$
$(\alpha^0, \alpha^2, \alpha^4, \alpha^6)$	$(\alpha^1, \alpha^3, \alpha^5, \alpha^7)$
$(\alpha^4, \alpha^3, \alpha^{-\infty}, \alpha^1)$	$(\alpha^7, \alpha^5, \alpha^2, \alpha^6)$
$(\alpha^{-\infty}, \alpha^5, \alpha^0, \alpha^7)$	$(\alpha^6, \alpha^2, \alpha^3, \alpha^1)$
$(\alpha^7, \alpha^0, \alpha^6, \alpha^3)$	$(\alpha^5, \alpha^4, \alpha^{-\infty}, \alpha^2)$
$(\alpha^2, \alpha^7, \alpha^3, \alpha^4)$	$(\alpha^{-\infty}, \alpha^6, \alpha^1, \alpha^0)$
$(\alpha^6, \alpha^4, \alpha^1, \alpha^5)$	$(\alpha^2, \alpha^{-\infty}, \alpha^0, \alpha^3)$
$(\alpha^3, \alpha^6, \alpha^5, \alpha^{-\infty})$	$(\alpha^0, \alpha^1, \alpha^7, \alpha^4)$
$(\alpha^1, \alpha^{-\infty}, \alpha^7, \alpha^2)$	$(\alpha^3, \alpha^0, \alpha^4, \alpha^5)$
$(\alpha^5, \alpha^1, \alpha^2, \alpha^0)$	$(\alpha^4, \alpha^7, \alpha^6, \alpha^{-\infty})$

and the incidence matrix is the following

$$\begin{bmatrix}
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}.$$

Chapter 6

Main Results

6.1 Construction Using Graphs

In this section we will take a closer look at the $(14, 8, 2, 2)$ superimposed code obtained from the $[8, 4, 4]$ extended Hamming code and see that it can be described as a graph with a very specific structure. This construction will not use any difficult graph theory, so there is little need to review more graph theory than the following definitions.

Definition 6.1. Let V be a nonempty finite set of vertices and $E \subseteq V \times V$ be a set of unordered pairs called edges. The pair $G = (V, E)$ is called an *undirected graph*.

Since we only deal with undirected graphs in this thesis, we will simply say graph instead of undirected graph.

Definition 6.2. Let $G = (V, E)$ be a graph, $a, b, a_0, a_1, \cdots, a_n \in V$ be vertices and $e_1, e_2, \cdots, e_n \in E$ be edges. An *a-b walk* is a finite alternating sequence

$$a = a_0, e_1, a_1, e_2 \ldots, e_{n-1}, a_{n-1}, e_n, a_n = b$$

of vertices and edges. The *length* of the walk is the number n of edges in the walk.

Definition 6.3. An *a-b* walk with $a = b$ is called a *closed* walk. A closed walk is called a *cycle* if the only repeated vertex is the starting and ending vertex $a = b$.

The discussion in this section about construction of superimposed codes using graphs

is based on the following $(14, 8, 2, 2)$ superimposed code

$$\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}.$$

Let the rows in the $(14, 8, 2, 2)$ superimposed code be represented by vertices in a graph. In the graphs discussed in this section, row i of a superimposed code is represented by vertex p_i. Each codeword is represented by a cycle containing the vertices corresponding to the positions in which the codeword has ones. There are several ways to draw the edges that make up the cycles. In Figure 6.1 the edges that are required for each column are created as needed when reading the column from top to bottom. In Figure 6.2 the vertices corresponding to the ones in each column was first marked and then the edges was drawn so that the graph looked as subjectively pleasing as possible. There is a remarkable difference in the looks of the graphs in Figure 6.1 and Figure 6.2, yet they represent the same superimposed code.

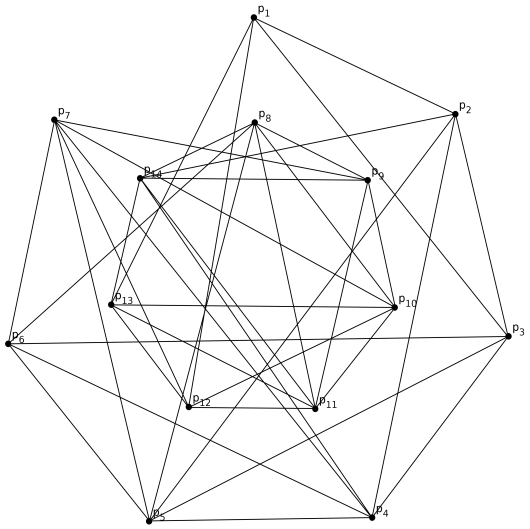

Figure 6.1: A graph represention of a $(14, 8, 2, 2)$ superimposed code.

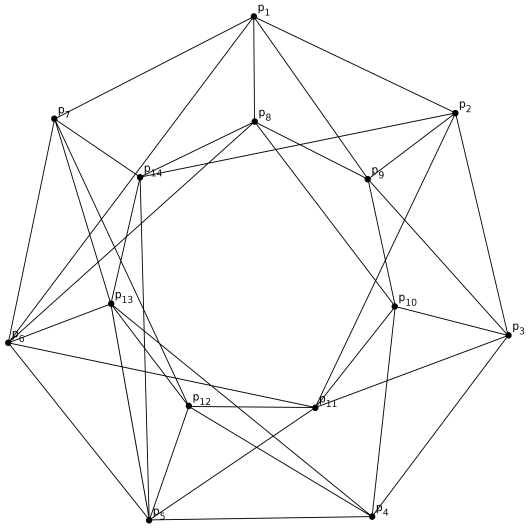

Figure 6.2: A subjectively nicer looking graph represention of a $(14, 8, 2, 2)$ superimposed code.

There are 14 vertices in each graph representing the $(14, 8, 2, 2)$ superimposed code. In these graphs there are 8 cycles such that if any two cycles $\boldsymbol{a}, \boldsymbol{b}$ and any other two cycles $\boldsymbol{c}, \boldsymbol{d}$ are chosen, then there is at least one vertex that both $\boldsymbol{a}$ and $\boldsymbol{b}$ contain, but neither $\boldsymbol{c}$ nor $\boldsymbol{d}$ contain. We will use the term *cover-free* cycles for cycles with this property.
If graphs are to be used for constructing superimposed codes the number of edges needs

to be kept low, otherwise the search for cover-free cycles will be too difficult. How would one go about constructing a graph that contains cover-free cycles and has as few edges as possible? The goal is of course not to construct the graph as the superimposed code is already known. The goal is to use a graph to construct the superimposed code, not the other way around. So while it was rather easy to construct the graphs representing the $(14, 8, 2, 2)$ superimposed code, it is quite difficult to construct such a graph when the code is not yet known.
If we want to construct a graph that represents some other code than the $(14, 8, 2, 2)$ superimposed code, then how would we construct the graph? We see that the graphs representing the $(14, 8, 2, 2)$ superimposed code have $2^4 - 2 = 14$ vertices, so maybe this number can be used to generalize the number of vertices needed. Before proposing some of the properties of the general construction, let us look at some more properties of the $(14, 8, 2, 2)$ superimposed code.
One property that the graphs representing the $(14, 8, 2, 2)$ superimposed code have that seems to be important is that any two cover-free cycles have exactly 3 vertices in common. What this means for the $(14, 8, 2, 2)$ superimposed code is that any two columns have exactly 3 ones in common. This is of course easy to verify for the $(14, 8, 2, 2)$ superimposed code since we have its matrix. If something similar is true for a general graph construction as well, we would have enough information to derive the minimum distance of the code.
Also, since the $(14, 8, 2, 2)$ superimposed code was constructed by specifically choosing the codewords of weight 4 in the extended $[8, 4, 4]$ Hamming code, then each vertex in the graph representations of the $(14, 8, 2, 2)$ superimposed code is contained in exactly 4 cover-free cycles. Since this is such a deliberate restriction of the choice of codewords, maybe this is also something that will be important for a general graph construction. This makes the rows of the superimposed code a constant weight code as well. However, we do not have a derivation of the minimum distance of the constant weight code whose codewords are the rows of the superimposed code.
So let us propose some properties of the graph construction, that might be true.

Conjecture 6.4. *Let $n \geq 2$ be a natural number and let the graph have $2^{n+1} - 2$ vertices. Then it is possible to add edges to the graph in such a way that 2^n cover-free cycles of length $2^n - 1$ appear, all with the property that any given pair of cover-free cycles have exactly $2^{n-1} - 1$ vertices in common. Furthermore, each vertex is contained in exactly 2^{n-1} cover-free cycles. The cover-free cycles in this graph represent codewords in a $\left(2^{n+1} - 2, 2^n, 2, 2\right)$ superimposed code.*

This is of course true for $n = 3$, since that is the $(14, 8, 2, 2)$ superimposed code.
If Conjecture 6.4 is true, then the resulting code is of course not only a superimposed code, but also a constant weight code. The code has length $2^{n+1} - 2$, weight $2^n - 1$ and below we will derive its minimum distance.

Theorem 6.5. *The $\left(2^{n+1} - 2, 2^n, 2, 2\right)$ superimposed code conjectured to exist in Conjecture 6.4 has minimum distance 2^n.*

Proof. Consider any two codewords $\boldsymbol{a}$ and $\boldsymbol{b}$ of the $\left(2^{n+1}-2, 2^n, 2, 2\right)$ superimposed code. $\boldsymbol{a}$ and $\boldsymbol{b}$ have exactly $2^{n-1}-1$ ones in common. Since the weight of the code is $s = 2^n - 1$ we have that $\boldsymbol{a}$ has ones in $s - \left(2^{n-1}-1\right) = 2^{n-1}$ positions where $\boldsymbol{b}$ has zeros. Now $\boldsymbol{b}$ has $s - \left(2^{n-1}-1\right) = 2^{n-1}$ positions with ones left, but in all those 2^{n-1} positions, $\boldsymbol{a}$ has zeros because otherwise $\boldsymbol{a}$ and $\boldsymbol{b}$ would have more than $2^{n-1}-1$ ones in common. Remember that the length of the code is $N = 2^{n+1}-2$. On the remaining $N - s - 2^{n-1} = 2^{n+1} - 2 - (2^n - 1) - 2^{n-1} = 2^{n-1} - 1$ positions, both $\boldsymbol{a}$ and $\boldsymbol{b}$ have zeros. That is, $\boldsymbol{a}$ has ones in 2^{n-1} positions where $\boldsymbol{b}$ has zeros and $\boldsymbol{b}$ has ones in 2^{n-1} positions where $\boldsymbol{a}$ has zeros. Thus, we have the distance $\operatorname{dist}(\boldsymbol{a}, \boldsymbol{b}) = 2^n$. Since $\boldsymbol{a}$ and $\boldsymbol{b}$ were chosen arbitrarily, the distance is the same between any pair of codewords. Thus, the minimum distance is $d = 2^n$. Q.E.D.

We do not know how the edges should be added to the graph in order for this to be a valid construction of a superimposed code. But if there is a way to construct such a graph in general, then we have a construction of a $\left(2^{n+1}-2, 2^n, 2, 2\right)$ superimposed code that is also a constant weight code with length $2^{n+1}-2$, weight $2^n - 1$ and minimum distance 2^n.
Let us conclude this section with an example, showing that the Conjecture 6.4 is true for the parameter $n = 2$. Conjecture 6.4 was purely designed from the superimposed code constructed from the extended $[8, 4, 4]$ Hamming code, so it could strengthen our case a little if we could find another example where the conjecture is true.

Example 6.6. For $n = 2$ we should have a $(6, 4, 2, 2)$ superimposed code with constant weight 3, where any two codewords have exactly 1 one in common. Also, the rows of the matrix should be a constant weight code of weight 2. This sounds a lot like the trivial $(6, 4, 2, 2)$ superimposed code. Indeed, the trivial $(6, 4, 2, 2)$ superimposed code meets all those requirements. Below are two $(6, 4, 2, 2)$ superimposed codes. The code $\boldsymbol{H}$ corresponds to the graph in Figure 6.3 and $\boldsymbol{G}$ corresponds to the graph in Figure 6.4. $\boldsymbol{G}$ is only presented to show that also this time it is possible to manipulate the code a little to get a subjectively nicer looking graph.

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

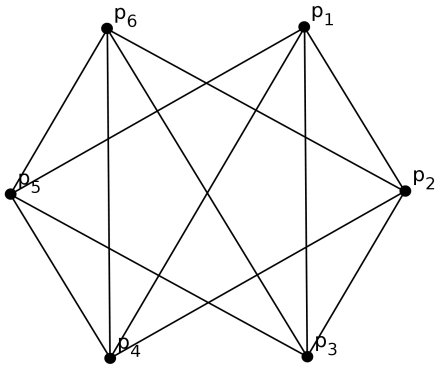

Figure 6.3: A graph represention of a $(6, 4, 2, 2)$ superimposed code.

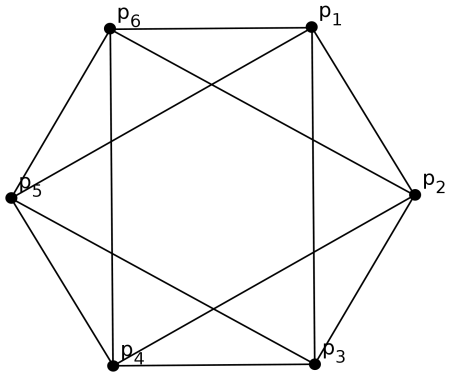

Figure 6.4: A subjectively nicer looking graph representation of a $(6, 4, 2, 2)$ superimposed code.

6.2 Construction Using Galois Fields

As we saw in Example 5.4, one superimposed code can be constructed from a Galois field. The construction in Example 5.4 was first presented in [7], however, no argument was given for *why* the construction gives rise to the super-simple $t-$design. It turns out that the construction in Example 5.4 can be used to construct other superimposed codes as well.
Recall from Example 5.4 the set $B = \left\{\left(\alpha^0, \alpha^2, \alpha^4, \alpha^6\right), \left(\alpha^1, \alpha^3, \alpha^5, \alpha^7\right)\right\}$. Note that the elements in $\left(\alpha^0, \alpha^2, \alpha^4, \alpha^6\right)$ are the elements of the subgroup $\left\langle \alpha^2 \right\rangle < GF^*\left(3^2\right)$ and the elements in $\left(\alpha^1, \alpha^3, \alpha^5, \alpha^7\right)$ are the elemnts of the coset $\alpha\left\langle \alpha^2 \right\rangle$. As we will soon see, this method for constructing superimposed codes is not restricted to the case with $GF\left(3^2\right)$. We will give a few more examples of codes that are constructed in this way, but using other fields than $GF\left(3^2\right)$, but still using the cosets of the subgroup $\left\langle \alpha^2 \right\rangle$.
Let us summarize what the proposed construction is.

Conjecture 6.7. *Let $GF(p^n)$ be a Galois field and α a primitive element. If the subgroup $\langle\alpha^2\rangle < GF^*(p^n)$ contains half the elements of $GF^*(p^n)$, then let $B = \{\langle\alpha^2\rangle, \alpha\langle\alpha^2\rangle\}$ and $\operatorname{dev} B = \bigcup_{b\in B}\{b+g : g \in GF(p^n)\}$. Now if $X = GF(p^n)$ is considered a set of points, then the pair $(X, \operatorname{dev} B)$ is a block design with $2|GF(p^n)|$ blocks and block size $|\langle\alpha^2\rangle|$. Sometimes the incidence matrix of $(X, \operatorname{dev} B)$ is a $(2p^n, p^n, 2, 2)$ superimposed code.*

In the following example we use $GF(5)$ to construct the trivial $(10, 5, 2, 2)$ superimposed code.

Example 6.8. Let $X = GF(5)$, $f(x) = x + 2$ and α be a root of f. Further, let $B = \{\langle\alpha^2\rangle, \alpha\langle\alpha^2\rangle\} = \{(\alpha^0, \alpha^2), (\alpha^1, \alpha^3)\}$ and $\operatorname{dev} B = \bigcup_{b\in B}\{b+g : g \in GF(5)\}$. In this case we get a $2-(5,2,1)$ design with the following blocks

$\langle\alpha^2\rangle$	$\alpha\langle\alpha^2\rangle$
(α^0, α^2)	(α^1, α^3)
$(\alpha^3, \alpha^{-\infty})$	(α^2, α^1)
(α^1, α^0)	$(\alpha^{-\infty}, \alpha^2)$
(α^2, α^3)	$(\alpha^0, \alpha^{-\infty})$
$(\alpha^{-\infty}, \alpha^1)$	(α^3, α^0)

and the incidence matrix

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

This matrix is so small that it is rather easy to verify by hand that this indeed is the trivial $(10, 5, 2, 2)$ superimposed code.

In the next two examples we will see that this construction does not work with $GF(7)$ and $GF(11)$. After those two examples, we will give another two examples that indeed do result in superimposed codes, and hopefully we will see a pattern emerging.

Example 6.9. Let $X = GF(7)$, $f(x) = x + 2$ and α be a root of f. Further, let $B = \{\langle\alpha^2\rangle, \alpha\langle\alpha^2\rangle\} = \{(\alpha^0, \alpha^2, \alpha^4), (\alpha^1, \alpha^3, \alpha^5)\}$ and $\operatorname{dev} B = \bigcup_{b\in B}\{b+g : g \in GF(7)\}$. The blocks are presented in the table below

$\langle\alpha^2\rangle$	$\alpha\langle\alpha^2\rangle$
$(\alpha^0, \alpha^2, \alpha^4)$	$(\alpha^1, \alpha^3, \alpha^5)$
$(\alpha^4, \alpha^1, \alpha^5)$	$(\alpha^3, \alpha^{-\infty}, \alpha^2)$
$(\alpha^5, \alpha^3, \alpha^2)$	$(\alpha^{-\infty}, \alpha^0, \alpha^1)$
$(\alpha^2, \alpha^{-\infty}, \alpha^1)$	$(\alpha^0, \alpha^4, \alpha^3)$
$(\alpha^1, \alpha^0, \alpha^3)$	$(\alpha^4, \alpha^5, \alpha^{-\infty})$
$(\alpha^3, \alpha^4, \alpha^{-\infty})$	$(\alpha^5, \alpha^2, \alpha^0)$
$(\alpha^{-\infty}, \alpha^5, \alpha^0)$	$(\alpha^2, \alpha^1, \alpha^4)$

and the incidence matrix is the following

$$C = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

If we let $W = \{1, 2\}$ and $R = \{3, 7\}$ we have $W \cap R = \varnothing$, but there is no $i \in \{1, 2, \ldots, 14\}$ such that $c_{i,1} = 1, c_{i,2} = 1, c_{i,3} = 0$ and $c_{i,7} = 0$. Thus, this is not a $(2, 2)$ superimposed code.

Example 6.10. Let $X = GF(11)$, $f(x) = x + 4$ and α be a root of f. Further, let $B = \left\{\langle\alpha^2\rangle, \alpha\langle\alpha^2\rangle\right\} = \left\{(\alpha^0, \alpha^2, \alpha^4, \alpha^6, \alpha^8), (\alpha^1, \alpha^3, \alpha^5, \alpha^7, \alpha^9)\right\}$ and $\operatorname{dev} B = \bigcup_{b \in B} \{b + g : g \in GF(11)\}$. The blocks are presented in the table below

$\langle \alpha^2 \rangle$	$\alpha \langle \alpha^2 \rangle$
$(\alpha^0, \alpha^2, \alpha^4, \alpha^6, \alpha^8)$	$(\alpha^1, \alpha^3, \alpha^5, \alpha^7, \alpha^9)$
$(\alpha^3, \alpha^7, \alpha^6, \alpha^2, \alpha^5)$	$(\alpha^9, \alpha^4, \alpha^{-\infty}, \alpha^1, \alpha^8)$
$(\alpha^4, \alpha^1, \alpha^2, \alpha^7, \alpha^{-\infty})$	$(\alpha^8, \alpha^6, \alpha^0, \alpha^9, \alpha^5)$
$(\alpha^6, \alpha^9, \alpha^7, \alpha^1, \alpha^0)$	$(\alpha^5, \alpha^2, \alpha^3, \alpha^8, \alpha^{-\infty})$
$(\alpha^2, \alpha^8, \alpha^1, \alpha^9, \alpha^3)$	$(\alpha^{-\infty}, \alpha^7, \alpha^4, \alpha^5, \alpha^0)$
$(\alpha^7, \alpha^5, \alpha^9, \alpha^8, \alpha^4)$	$(\alpha^0, \alpha^1, \alpha^6, \alpha^{-\infty}, \alpha^3)$
$(\alpha^1, \alpha^{-\infty}, \alpha^8, \alpha^5, \alpha^6)$	$(\alpha^3, \alpha^9, \alpha^2, \alpha^0, \alpha^4)$
$(\alpha^9, \alpha^0, \alpha^5, \alpha^{-\infty}, \alpha^2)$	$(\alpha^4, \alpha^8, \alpha^7, \alpha^3, \alpha^6)$
$(\alpha^8, \alpha^3, \alpha^{-\infty}, \alpha^0, \alpha^7)$	$(\alpha^6, \alpha^5, \alpha^1, \alpha^4, \alpha^2)$
$(\alpha^5, \alpha^4, \alpha^0, \alpha^3, \alpha^1)$	$(\alpha^2, \alpha^{-\infty}, \alpha^9, \alpha^6, \alpha^7)$
$(\alpha^{-\infty}, \alpha^6, \alpha^3, \alpha^4, \alpha^9)$	$(\alpha^7, \alpha^0, \alpha^8, \alpha^2, \alpha^1)$

and the incidence matrix is the following

$$C = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

If we let $W = \{1, 3\}$ and $R = \{6, 8\}$ we have $W \cap R = \varnothing$, but there is no $i \in \{1, 2, \ldots, 22\}$ such that $c_{i,1} = 1, c_{i,3} = 1, c_{i,6} = 0$ and $c_{i,8} = 0$. Thus, this is not a $(2, 2)$ superimposed code.

Unlike the above two examples, the next two examples will actually be valid constructions of superimposed codes. In the next example, we construct a $(26, 13, 2, 2)$ superimposed code.

Example 6.11. Let $X = GF(13)$, $f(x) = x + 6$ and α be a root of f. Further, let $B = \left\{\left\langle\alpha^2\right\rangle, \alpha\left\langle\alpha^2\right\rangle\right\} = \left\{\left(\alpha^0, \alpha^2, \alpha^4, \alpha^6, \alpha^8, \alpha^{10}\right), \left(\alpha^1, \alpha^3, \alpha^5, \alpha^7, \alpha^9, \alpha^{11}\right)\right\}$ and $\operatorname{dev} B = \bigcup_{b\in B} \{b + g : g \in GF(13)\}$. Below is the incidence matrix of this design, along with a table of the blocks

$$\begin{bmatrix}
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{bmatrix},$$

$\langle\alpha^2\rangle$	$\alpha\langle\alpha^2\rangle$
$(\alpha^0,\alpha^2,\alpha^4,\alpha^6,\alpha^8,\alpha^{10})$	$(\alpha^1,\alpha^3,\alpha^5,\alpha^7,\alpha^9,\alpha^{11})$
$(\alpha^{11},\alpha^5,\alpha^2,\alpha^{-\infty},\alpha^{10},\alpha^3)$	$(\alpha^9,\alpha^7,\alpha^6,\alpha^1,\alpha^4,\alpha^8)$
$(\alpha^8,\alpha^6,\alpha^5,\alpha^0,\alpha^3,\alpha^7)$	$(\alpha^4,\alpha^1,\alpha^{-\infty},\alpha^9,\alpha^2,\alpha^{10})$
$(\alpha^{10},\alpha^{-\infty},\alpha^6,\alpha^{11},\alpha^7,\alpha^1)$	$(\alpha^2,\alpha^9,\alpha^0,\alpha^4,\alpha^5,\alpha^3)$
$(\alpha^3,\alpha^0,\alpha^{-\infty},\alpha^8,\alpha^1,\alpha^9)$	$(\alpha^5,\alpha^4,\alpha^{11},\alpha^2,\alpha^6,\alpha^7)$
$(\alpha^7,\alpha^{11},\alpha^0,\alpha^{10},\alpha^9,\alpha^4)$	$(\alpha^6,\alpha^2,\alpha^8,\alpha^5,\alpha^{-\infty},\alpha^1)$
$(\alpha^1,\alpha^8,\alpha^{11},\alpha^3,\alpha^4,\alpha^2)$	$(\alpha^{-\infty},\alpha^5,\alpha^{10},\alpha^6,\alpha^0,\alpha^9)$
$(\alpha^9,\alpha^{10},\alpha^8,\alpha^7,\alpha^2,\alpha^5)$	$(\alpha^0,\alpha^6,\alpha^3,\alpha^{-\infty},\alpha^{11},\alpha^4)$
$(\alpha^4,\alpha^3,\alpha^{10},\alpha^1,\alpha^5,\alpha^6)$	$(\alpha^{11},\alpha^{-\infty},\alpha^7,\alpha^0,\alpha^8,\alpha^2)$
$(\alpha^2,\alpha^7,\alpha^3,\alpha^9,\alpha^6,\alpha^{-\infty})$	$(\alpha^8,\alpha^0,\alpha^1,\alpha^{11},\alpha^{10},\alpha^5)$
$(\alpha^5,\alpha^1,\alpha^7,\alpha^4,\alpha^{-\infty},\alpha^0)$	$(\alpha^{10},\alpha^{11},\alpha^9,\alpha^8,\alpha^3,\alpha^6)$
$(\alpha^6,\alpha^9,\alpha^1,\alpha^2,\alpha^0,\alpha^{11})$	$(\alpha^3,\alpha^8,\alpha^4,\alpha^{10},\alpha^7,\alpha^{-\infty})$
$(\alpha^{-\infty},\alpha^4,\alpha^9,\alpha^5,\alpha^{11},\alpha^8)$	$(\alpha^7,\alpha^{10},\alpha^2,\alpha^3,\alpha^1,\alpha^0)$

.

It is not easy to verify by hand that this indeed is a $(26,13,2,2)$ superimposed code. But if the above matrix is plugged into the program in Appendix A, it turns out that this indeed is a $(26,13,2,2)$ superimposed code.

Example 6.12. Let $X = GF(17)$, $f(x) = x+14$ and α be a root of f. Further, let $B = \{\langle\alpha^2\rangle, \alpha\langle\alpha^2\rangle\} = \{(\alpha^0,\alpha^2,\alpha^4,\alpha^6,\alpha^8,\alpha^{10},\alpha^{12},\alpha^{14}), (\alpha^1,\alpha^3,\alpha^5,\alpha^7,\alpha^9,\alpha^{11},\alpha^{13},\alpha^{15})\}$ and $\operatorname{dev} B = \bigcup_{b\in B}\{b+g : g\in GF(17)\}$. A table of the blocks of this design and its incidence matrix is printed below:

$\langle\alpha^2\rangle$	$\alpha\langle\alpha^2\rangle$
$(\alpha^0,\alpha^2,\alpha^4,\alpha^6,\alpha^8,\alpha^{10},\alpha^{12},\alpha^{14})$	$(\alpha^1,\alpha^3,\alpha^5,\alpha^7,\alpha^9,\alpha^{11},\alpha^{13},\alpha^{15})$
$(\alpha^{14},\alpha^3,\alpha^9,\alpha^8,\alpha^{-\infty},\alpha^2,\alpha^5,\alpha^1)$	$(\alpha^{12},\alpha^7,\alpha^{15},\alpha^{13},\alpha^6,\alpha^{10},\alpha^4,\alpha^{11})$
$(\alpha^1,\alpha^7,\alpha^6,\alpha^{-\infty},\alpha^0,\alpha^3,\alpha^{15},\alpha^{12})$	$(\alpha^5,\alpha^{13},\alpha^{11},\alpha^4,\alpha^8,\alpha^2,\alpha^9,\alpha^{10})$
$(\alpha^{12},\alpha^{13},\alpha^8,\alpha^0,\alpha^{14},\alpha^7,\alpha^{11},\alpha^5)$	$(\alpha^{15},\alpha^4,\alpha^{10},\alpha^9,\alpha^{-\infty},\alpha^3,\alpha^6,\alpha^2)$
$(\alpha^5,\alpha^4,\alpha^{-\infty},\alpha^{14},\alpha^1,\alpha^{13},\alpha^{10},\alpha^{15})$	$(\alpha^{11},\alpha^9,\alpha^2,\alpha^6,\alpha^0,\alpha^7,\alpha^8,\alpha^3)$
$(\alpha^{15},\alpha^9,\alpha^0,\alpha^1,\alpha^{12},\alpha^4,\alpha^2,\alpha^{11})$	$(\alpha^{10},\alpha^6,\alpha^3,\alpha^8,\alpha^{14},\alpha^{13},\alpha^{-\infty},\alpha^7)$
$(\alpha^{11},\alpha^6,\alpha^{14},\alpha^{12},\alpha^5,\alpha^9,\alpha^3,\alpha^{10})$	$(\alpha^2,\alpha^8,\alpha^7,\alpha^{-\infty},\alpha^1,\alpha^4,\alpha^0,\alpha^{13})$
$(\alpha^{10},\alpha^8,\alpha^1,\alpha^5,\alpha^{15},\alpha^6,\alpha^7,\alpha^2)$	$(\alpha^3,\alpha^{-\infty},\alpha^{13},\alpha^0,\alpha^{12},\alpha^9,\alpha^{14},\alpha^4)$
$(\alpha^2,\alpha^{-\infty},\alpha^{12},\alpha^{15},\alpha^{11},\alpha^8,\alpha^{13},\alpha^3)$	$(\alpha^7,\alpha^0,\alpha^4,\alpha^{14},\alpha^5,\alpha^6,\alpha^1,\alpha^9)$
$(\alpha^3,\alpha^0,\alpha^5,\alpha^{11},\alpha^{10},\alpha^{-\infty},\alpha^4,\alpha^7)$	$(\alpha^{13},\alpha^{14},\alpha^9,\alpha^1,\alpha^{15},\alpha^8,\alpha^{12},\alpha^6)$
$(\alpha^7,\alpha^{14},\alpha^{15},\alpha^{10},\alpha^2,\alpha^0,\alpha^9,\alpha^{13})$	$(\alpha^4,\alpha^1,\alpha^6,\alpha^{12},\alpha^{11},\alpha^{-\infty},\alpha^5,\alpha^8)$
$(\alpha^{13},\alpha^1,\alpha^{11},\alpha^2,\alpha^3,\alpha^{14},\alpha^6,\alpha^4)$	$(\alpha^9,\alpha^{12},\alpha^8,\alpha^5,\alpha^{10},\alpha^0,\alpha^{15},\alpha^{-\infty})$
$(\alpha^4,\alpha^{12},\alpha^{10},\alpha^3,\alpha^7,\alpha^1,\alpha^8,\alpha^9)$	$(\alpha^6,\alpha^5,\alpha^{-\infty},\alpha^{15},\alpha^2,\alpha^{14},\alpha^{11},\alpha^0)$
$(\alpha^9,\alpha^5,\alpha^2,\alpha^7,\alpha^{13},\alpha^{12},\alpha^{-\infty},\alpha^6)$	$(\alpha^8,\alpha^{15},\alpha^0,\alpha^{11},\alpha^3,\alpha^1,\alpha^{10},\alpha^{14})$
$(\alpha^6,\alpha^{15},\alpha^3,\alpha^{13},\alpha^4,\alpha^5,\alpha^0,\alpha^8)$	$(\alpha^{-\infty},\alpha^{11},\alpha^{14},\alpha^{10},\alpha^7,\alpha^{12},\alpha^2,\alpha^1)$
$(\alpha^8,\alpha^{11},\alpha^7,\alpha^4,\alpha^9,\alpha^{15},\alpha^{14},\alpha^{-\infty})$	$(\alpha^0,\alpha^{10},\alpha^1,\alpha^2,\alpha^{13},\alpha^5,\alpha^3,\alpha^{12})$
$(\alpha^{-\infty},\alpha^{10},\alpha^{13},\alpha^9,\alpha^6,\alpha^{11},\alpha^1,\alpha^0)$	$(\alpha^{14},\alpha^2,\alpha^{12},\alpha^3,\alpha^4,\alpha^{15},\alpha^7,\alpha^5)$

,

$$\begin{bmatrix}
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0\\
1&0&1&1&1&0&1&0&0&1&1&0&0&0&0&1&0\\
1&1&1&0&1&0&0&1&1&0&0&0&0&1&0&0&1\\
0&1&0&0&0&0&1&0&1&1&0&0&1&1&1&1&0\\
1&0&1&0&0&1&1&0&0&0&0&1&0&0&1&1&1\\
0&1&1&1&0&1&0&0&0&0&1&0&1&1&0&0&1\\
0&0&0&0&1&0&1&1&0&0&1&1&1&1&0&1&0\\
0&0&1&1&0&0&1&1&1&1&0&1&0&0&0&0&1\\
1&0&0&1&1&0&0&0&0&1&0&0&1&1&1&0&1\\
1&1&0&0&1&1&1&0&1&0&0&1&1&0&0&0&0\\
0&1&0&1&0&0&0&0&1&0&1&1&0&0&1&1&1\\
0&0&1&1&1&1&0&1&0&0&0&0&1&0&1&1&0\\
0&0&1&0&1&1&0&0&1&1&1&1&0&1&0&0&0\\
1&0&0&1&0&0&1&1&1&0&1&0&0&1&1&0&0\\
0&1&0&0&1&1&1&1&0&1&0&0&0&0&1&0&1\\
1&0&0&0&0&1&0&0&1&1&1&0&1&0&0&1&1\\
1&1&1&0&0&0&0&1&0&0&1&1&1&0&1&0&0\\
0&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\
0&0&0&0&0&1&0&1&1&0&0&1&1&1&1&0&1\\
0&0&0&1&0&1&1&0&0&1&1&1&1&0&1&0&0\\
1&0&0&1&1&1&0&1&0&0&1&1&0&0&0&0&1\\
0&1&0&1&1&0&0&1&1&1&1&0&1&0&0&0&0\\
1&0&0&0&1&0&0&1&1&1&0&1&0&0&1&1&0\\
1&1&1&1&0&1&0&0&1&1&0&0&0&0&1&0&0\\
1&1&0&0&1&1&0&0&0&0&1&0&0&1&1&1&0\\
0&1&1&0&0&1&1&1&1&0&1&0&0&0&0&1&0\\
0&0&1&0&0&0&0&1&0&1&1&0&0&1&1&1&1\\
1&0&1&0&0&1&1&1&0&1&0&0&1&1&0&0&0\\
1&1&0&0&0&0&1&0&0&1&1&1&0&1&0&0&1\\
1&1&0&1&0&0&1&1&0&0&0&0&1&0&0&1&1\\
0&1&1&0&1&0&0&0&0&1&0&1&1&0&0&1&1\\
1&0&1&1&0&0&0&0&1&0&0&1&1&1&0&1&0\\
0&1&1&1&1&0&1&0&0&0&0&1&0&1&1&0&0\\
0&0&0&1&1&1&1&0&1&0&0&0&0&1&0&1&1
\end{bmatrix}.$$

In this case it is even harder to verify by hand the validity of the code. But once again, if the matrix is plugged into the program in Appendix A, the code turns out to be a $(34, 17, 2, 2)$ superimposed code.

6.2.1 Concluding Remarks

Let us look at some properties of the superimposed codes constructed in this section. If we look back at the tables listing the blocks in the examples of this construction and

focus on the first position in the blocks derived from the subgroup $\langle\alpha^2\rangle$. Then we see that as we look through all the blocks corresponding to $\langle\alpha^2\rangle$ and look in the first position, every element of the corresponding Galois field is found exactly once. This is true for any position in the blocks for both cosets of the subgroup $\langle\alpha^2\rangle$. This is true since if α^k is the element in position i of either coset, in the corresponding blocks will be on position i elements of the form $g + \alpha^k$ where g belongs to the corresponding Galois field. If the same element is found on position i in two different blocks, then we have $g_1 + \alpha^k = g_2 + \alpha^k$, from which we get that $g_1 = g_2$ and thus we cannot have the same element on the same position in different blocks. Since the block size is $\left|\langle\alpha^2\rangle\right|$ then if we count over all blocks corresponding to both cosets of the subgroup, we see that every element in the Galois field is found exactly $2\left|\langle\alpha^2\rangle\right|$ times. Thus, the superimposed codes constructed in this way is a constant weight code of weight $2\left|\langle\alpha^2\rangle\right|$. It is, however, uncertain what the minimum distance would be.

Now one question to think about is whether or not this construction works in general. We saw examples where the construction does not work, so clearly the construction does not work for any arbitrary subgroup of the multiplicative group of any Galois field. But what is the property that the Galois field must have in order for this construction to work? Unfortunately, this question will not be answered in this thesis. As said earlier, however, we can see a pattern emerging in the examples above. In every case where the construction failed, the subgroup $\langle\alpha^2\rangle < GF^*(p^n)$ has odd size and in every case where the construction worked the subgroup $\langle\alpha^2\rangle$ has even size. However, the sample size is rather small so it is hard to say too much about this connection, and for that reason it might be something worth studying further.

But what if this construction goes further than superimposed codes? Clearly, the pair $(X, \operatorname{dev} B)$ as described in Conjecture 6.7 is some form of block design. But what are the properties of this block design? If $(X, \operatorname{dev} B)$ is a t-design, then what are the valid values of t and λ? Of course $v = p^n$, since the points are the elements of the Galois field, and in the examples we have seen, $k = \frac{p^n-1}{2}$. The 2 in the denominator of k comes from the fact that $|B| = 2$, which in turn comes from the fact that $\langle\alpha^2\rangle$ contains half the elements of the multiplicative group $GF^*(p^n)$. This suggests that Conjecture 6.7 could possibly be modified to work for other subgroups than $\langle\alpha^2\rangle$, with B being the set of cosets to this subgroup. If this modification of Conjecture 6.7 is possible, then the block size k would still be the size of the subgroup, which can also be expressed $k = \frac{p^n-1}{|B|}$. So we know which values we expect to possibly work for v and k, but still, what about t and λ? Valid values for t and λ will not be provided in this thesis, but maybe this could be the topic for further research.

Another way to continue working on this construction is to find sufficient conditions for this construction to also be a superimposed code. By the examples in this thesis it looks like the size of the subgroup $\langle\alpha^2\rangle$ could be important. Also, all the superimposed codes presented in this thesis have the parameters $w = r = 2$. But can this construction maybe be used to construct superimposed codes with the parameters w and r not both equal to 2, and if so, what exactly does w and r depend on? This could also be the topic for further research.

Appendix A

Code Verification Program

```
/* Compiled with gcc 5.4.0 */
/* Compiler flags used -std=c11 -Wall -Wextra -pedantic */
/* This program currently only works for w = 2 and r = 2 */
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

void print_array(long*, long);
long factorial(long);
long n_choose_k(long, long);

void print_array(long *x, long size){
  for(int i = 0; i < size; i++){i
    printf("%2ld", x[i]);
    if(i < size - 1) printf(", ");
  }
}

long factorial(long x){
  long i = 1;
  while(x > 1) i *= x--;
  return i;
}

long n_choose_k(long n, long k){
  return factorial(n) / (factorial(n - k) * factorial(k));
}

int main(int argc, char **argv){
  if(argc != 5){
```

```
  printf("USE: ./SIC_check N T w r\nQuitting\n");
  return 1;
}

FILE *fp = fopen("Code", "r");
const long N = atoi(argv[1]);
const long T = atoi(argv[2]);
const long w = atoi(argv[3]);
const long r = atoi(argv[4]);
long Code[N][T];
long num_of_subsets = n_choose_k(T, 2);
long subsets[num_of_subsets][w];
long is_ok, i, j, k;
char c;

printf("\n_____________________________________________________");
printf("\n Verifying (N, T, w, r) = (%ld, %ld, %ld, %ld) SIC", N, T, w, r);
printf("\n_____________________________________________________");
printf("\n The W, R sets");
printf("\n_____________________________________________________");

/* Initializes the first element in the subsets */
for(i = 2, j = 1, k = 0; k < num_of_subsets; k++){
  subsets[k][1] = (j < T ? ++j: (j = ++i));
}

/* Initializes the second element in the subsets */
for(j = 1, k = 0; k < num_of_subsets; k++){
  subsets[k][0] = (subsets[k][1] == T ? j++: j);
}
printf("\n");
for(i = j = 0; i < num_of_subsets; i++, j++){
  if(j == 5){
    j = 0;
    printf("\n");
  }
  printf(" {"); print_array((long*)subsets[i], 2); printf("}");
}

/* Reads the code into buffer */
if(!fp){
  printf("\nFile pointer error");
  return 1;
```

```
}
for(i = 0; i < N; i++){
  for(j = 0; j < T; j++){
    while(isspace(c = fgetc(fp)));
    Code[i][j] = c - '0';
  }
}
fclose(fp);

printf("\n_____________________________________________________");
printf("\n The code to be tested");
printf("\n_____________________________________________________");
for(i = 0; i < N; i++){
  printf("\n");
  for(j = 0; j < T; j++){
    printf("%ld ", Code[i][j]);
  }
}

printf("\n_____________________________________________________");
printf("\n The pairs of W, R that fail(if any)");
printf("\n_____________________________________________________");

/* Loops through all pairs of sets W, R */
for(i = 0; i < num_of_subsets; i++){
  for(j = 0; j < num_of_subsets; j++){
    is_ok = 0;

    /* The following condition checks if  */
    /* the current sets W, R are disjoint */
    if((subsets[i][0] != subsets[j][0]) &&
       (subsets[i][0] != subsets[j][1]) &&
       (subsets[i][1] != subsets[j][0]) &&
       (subsets[i][1] != subsets[j][1])){
      for(k = 0; k < N; k++){

        /* The following condition checks if */
        /* row k has 1 in columns in W and 0 */
        /* in columns in R */
        if((Code[k][subsets[i][0] - 1] == 1) &&
           (Code[k][subsets[i][1] - 1] == 1) &&
           (Code[k][subsets[j][0] - 1] == 0) &&
           (Code[k][subsets[j][1] - 1] == 0)){
```

```
            is_ok = 1;
          }
        }
        if(!is_ok){
          printf("\n W = {%ld, %ld}, R = {%ld, %ld}", subsets[i][0]
                                                    , subsets[i][1]
                                                    , subsets[j][0]
                                                    , subsets[j][1]);
        }
      }
    }
  }

  printf("\n");
  return 0;
}
```

Bibliography

[1] Haitao Cao, Kejun Chen, and Ruizhong Wei. "Super-simple balanced incomplete block designs with block size 4 and index 5". In: *Discrete Mathematics* 309.9 (2009), pp. 2808–2814.

[2] Kejun Chen, Zhenfu Cao, and Ruizhong Wei. "Super-simple balanced incomplete block designs with block size 4 and index 6". In: *Journal of statistical planning and inference* 133.2 (2005), pp. 537–554.

[3] Kejun Chen and Ruizhong Wei. "Super-simple (v, 5, 5) Designs". In: *Designs, Codes and Cryptography* 39.2 (2006), pp. 173–187.

[4] Kejun Chen and Ruizhong Wei. "Super-simple (v,5,4) designs". In: *Discrete Applied Mathematics* 155.8 (2007), pp. 904–913.

[5] Igor Gashkov, J AO Ekberg, and D Taub. "A geometric approach to finding new lower bounds of A (n, d, w)". In: *Designs, Codes and Cryptography* 43.2 (2007), pp. 85–91.

[6] Hans-Dietrich O.F. Gronau, Donald L. Kreher, and Alan C.H. Ling. "Super-simple (v,5,2)-designs". In: *Discrete Applied Mathematics* 138.1 (2004), pp. 65–77.

[7] Chen Kejun. "On the existence of super-simple (v, 4, 3)-BIBDs". In: *Journal of Combinatorial Mathematics and Combinatorial Computing* 17 (1995), pp. 149–159.

[8] Chen Kejun. "On the existence of super-simple (v,4,4)-BIBDs". In: *Journal of Statistical Planning and Inference* 51.3 (1996), pp. 339–350.

[9] Hyun Kwang Kim and Vladimir Lebedev. "On Optimal Superimposed Codes". In: *Journal of Combinatorial Designs* 12.2 (2004), pp. 79–91.

[10] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The Theory of Error-Correcting Codes*. Elsevier, 1977.

[11] Per-Anders Svensson. *Abstrakt algebra*. Studentlitteratur, 2007.