Radio Science, Volume 32, Number 1, Pages 9–18, January–February 1997

Long alternating codes
2. Practical search method

Markku Markkanen
Geophysical Observatory, Sodankylä, Finland

Tuomo Nygrén
Department of Physical Sciences, University of Oulu, Oulu, Finland

Abstract. This paper is the second one in a series explaining a new search method of long alternating codes for incoherent scatter radars. The first paper explains the general idea of the method in terms of a special game of dominoes. This second paper gives an alternative mathematical formalism suitable for computer search. It consists of three rules and a mathematical analysis leading to a formula which can be used in practical search. Although the rules were originally experimental, a mathematical proof of their sufficiency is also given. The method has been used to make a complete search up to a length of 1,048,576 bits. Even longer codes have been found; the longest one known at the moment contains 4,194,304 bits. For demonstration, complete tables of 8-, 16-, 32-, and 64-bit codes and examples of 128- and 256-bit codes are presented.

1. Introduction

A search of alternating codes could, in principle, be done by going through all possible sign combinations in a sign matrix and testing which make phase patterns satisfying the condition of an alternating code. Because of the large number of alternatives, however, such a method is very time consuming, and it is easy to see that not even short codes could be found in a reasonable time. In the case of a strong 8-bit code, for instance, the number of alternatives to be studied would be $2^{8\times16} = 2^{128} \approx 3.4 \times 10^{38}$. The search method discovered by *Lehtinen* [1986] was based on testing so-called weak or strong conditions for strings of small integer numbers. Although this method is enormously faster than a straightforward test of sign matrices, it has allowed finding codes only up to 32 bits. A new, still much faster method has now been invented, and it is presented here in a series of two papers.

The new method was originally discovered by studying regular patterns in the strong codes found by *Lehtinen and Häggström* [1987]. These patterns could be used in developing a special game of dominoes which has a property that playing the game successfully to the very end is equivalent to finding a strong alternating code. The game is described in detail in the first paper [*Nygrén and Markkanen,* 1996] (hereafter referred to as paper 1).

The game of dominoes is not practical in a true code search. For this purpose it is useful to transform it into a mathematical framework so that the search can be more easily programmed for a computer. Such a formalism is also more suitable for mathematical analysis. The mathematical formulation is presented in this paper together with tables of several long alternating codes which are published here for the first time. The first 64-bit strong code found by this method was shown in a preliminary paper by *Markkanen and Nygrén* [1996]. Although it is advisable to read paper 1, which contains a description of the method in an easy visual form, first, the method can be understood by reading the present paper only.

Paper number 96RS03117.
0048-6604/97/96RS-03117$11.00

2. Alternating Codes as Walsh Sequences

The alternating codes are usually presented in terms of their Walsh sequences which are strings

of small integer numbers [see paper 1 and *Lehtinen and Häggström,* 1987]. A sequence is converted to the sign matrix of a code by taking the respective columns from the Walsh matrix and inserting them one after another in this specific order. In the case of an n-bit weak or strong code the phase patterns of the code sequence are then found from the n or $2n$ uppermost rows of the resulting matrix, respectively.

The length n of a full-length alternating code is always a power of 2 so that it can be put in the form $n = 2^m$, where m is a positive integer. *Lehtinen and Häggström* [1987] have pointed out that all Walsh indices in a strong code sequence are smaller than $2n-1 = 2^{m+1}-1$, and the same index can appear in a sequence only once. This means that we can make a strong code by selecting 2^m integer numbers in a proper order from the set

$$G = \{0, 1, \ldots, 2^{m+1} - 1\}. \tag{1}$$

This gives $(2n)!/n!$ different choices to be tested. Already in the case $n = 4$ the number of alternatives is 1680, and it grows steeply with increasing code length so that for $n = 32$ the number is about 5×10^{53}.

Although the number of alternatives in choosing sequences of n numbers from G is immensely smaller than the number of possible sign combinations in the corresponding sign matrix, it is still so great that testing all possibilities for a 32-bit strong code, for instance, cannot be done by any computer. Luckily enough, this is not even necessary. Because every piece of an alternating code also makes an alternating code, one can try to build a code step by step starting from each element in G. Whenever a number joined after a possible piece of a code makes a sequence which does not satisfy the strong condition, there is no need to try to extend this piece any longer. This procedure leads to studies of tree-like structures, and a majority of the $(2n)!/n!$ alternatives will be automatically rejected without testing. In spite of this, codes longer than 32 bits have not been found by this method.

3. Regular Patterns in Strong Codes

The new search method deals with strong codes only. This is not a limitation since a weak code can always be obtained from a strong one by dividing its Walsh indices by 2 and neglecting the remainder.

The method is based on studies of the strong codes found by *Lehtinen and Häggström* [1987]. It was observed that all codes up to the length of 32 bits show a similar regular behavior. Therefore it seems reasonable to assume that a similar behavior will be met in longer codes also. The regularities can be written in terms of three rules. By taking the rules as axioms it is then possible to derive further results which are useful in practical search. The rules and the derivation of some of their consequences are given in this section. In each case the idea is demonstrated in terms of 8-bit codes. The reader can then easily check that the statements are valid for the longer codes also.

In the following, we deal with the elements of G, which are treated as binary numbers. For later use we divide G into two subsets

$$G_l = \{0, 1, \ldots, 2^m - 1\} \tag{2}$$

$$G_u = \{2^m, 2^m + 1, \ldots, 2^{m+1} - 1\}. \tag{3}$$

A binary exclusive or operation of numbers a and b is denoted by $a \oplus b$, and it means a binary number which has a 1 only at such digits where either a or b (but not both) have a 1; the contents of all the other digits are "zeros".

The parity of a binary number is determined by the number of its nonzero digits. If a has an even number of nonzero digits, its parity is $\mathrm{par}(a) = 1$ (even), otherwise $\mathrm{par}(a) = -1$ (odd). In the continuation we say for brevity that a is even or odd if it has an even or odd parity, respectively. By a definition

$$a^o = 2a \oplus \max[\mathrm{par}(a), 0] \tag{4}$$

we obtain a function o which shifts the bit pattern of a by one step toward higher digits and puts the least significant digit to zero if a is odd and to unity if a is even. The value of this function is always odd. The function will be used in constructing number sequences which are pieces of alternating codes.

A careful inspection of the known strong alternating codes shows that they are built from fixed patterns of the elements of the set G. Each number of G can only appear in one of these patterns, and each pattern can appear only once in the same code. The following experimental rule is valid.

Rule 1. If P is the set of all possible number sequences of the form $p = \{a, a^o, a^{oo}, \ldots, a^{o\cdots o}\}$ constructed in such a way that a is even and the last and only the last number in the sequence is larger than $2^m - 1$, then n-bit strong alternating codes can be built by selecting proper elements of P and inserting

them in a string in a proper order. The first element in the code is $\{0, 2^0, 2^1, \ldots, 2^m\}$.

In the case of 8-bit codes the binary presentation of $2^{m+1} - 1$ is 1111 so that

$$G = \{0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\}.$$

The even elements of G are 0, 11, 101, 110, 1001, 1010, 1100, and 1111, which are the first elements of all $p \in P$. Each p sequence is stopped after passing the threshold $2^m - 1 = 111$. Therefore

$$P = \{\{0, 1, 10, 100, 1000\}, \{11, 111, 1110\}, \{101, 1011\}, \{110, 1101\}, \{1001\}, \{1010\}, \{1100\}, \{1111\}\}.$$

In the conventional octal presentation we can write

$$P = \{\{0, 1, 2, 4, 10\}, \{3, 7, 16\}, \{5, 13\}, \{6, 15\}, \{11\}, \{12\}, \{14\}, \{17\}\}.$$

The two strong 8-bit codes given by *Lehtinen and Häggström* [1987] are

$$C^{(8a)} = \{00, 01, 02, 04, 10, 03, 07, 16\}$$
$$C^{(8b)} = \{00, 01, 02, 04, 10, 14, 05, 13\},$$

which can indeed be constructed from the elements of P. Note that only a part of the elements of P are used in each code and some of the elements are not used at all.

The set P has some properties which immediately follow from the definition in rule 1. Obviously, each number in G belongs to one and only one element of P. This is because all even numbers are used as start points of elements of P, and each odd number belongs to an element starting with a certain even number. The latter is true because every odd number can be traced back to some even number by shifting its bit pattern downward until its least significant 1 drops out from the lower end. Since G always contains n numbers with even and n with odd parity and the element strings of P are started from all even numbers, P must obviously contain n elements. For the same reason, $n/2$ of them start from a number smaller than 2^m and are therefore longer than one so that the remaining $n/2$ elements consist of a single number. The validity of all these results for the case $n = 8$ is directly seen from the presentation of the corresponding set P given above.

Although rule 1 is already of a great help in code search, there are still quite many alternatives to be studied. The number of possibilities for joining the elements of P together can be reduced by the next rule, which is as follows.

Rule 2. If S is a set with its elements $s = \{a \oplus a^o, a^o \oplus a^{oo}, \ldots\}$ obtained from all those $n/2$ elements $p = \{a, a^o, \ldots\} \in P$ which are longer than 1 and $C = \{a_1, a_2, \ldots, a_n\}$ is a strong alternating code, the cyclic sequence

$$C_{c1} = \{a_1 \oplus a_2, a_2 \oplus a_3, \ldots, a_{n-1} \oplus a_n, a_n \oplus a_1\}$$

consists of all elements of S in some order.

In the case of 8 bits the first four elements of P lead to

$$S = \{\{1, 3, 6, 14\}, \{4, 11\}, \{16\}, \{13\}\},$$

and the two codes $C^{(8a)}$ and $C^{(8b)}$ give

$$C_{c1}^{(8a)} = \{1, 3, 6, 14, 13, 4, 11, 16\}$$
$$C_{c1}^{(8b)} = \{1, 3, 6, 14, 4, 11, 16, 13\}.$$

We see that both $C_{c1}^{(8a)}$ and $C_{c1}^{(8b)}$ contain all elements of S.

From rule 2 it is clear that even if some $p \in P$, which is longer than one, is not present in code C, the corresponding element $s \in S$ is in C_{c1}. From this, one can derive an important consequence. Assume that

$$p = \{b_1, b_2, \ldots, b_j\}, \tag{5}$$

where $j > 1$, is not in a code. The corresponding element in S is

$$s = \{b_1 \oplus b_2, b_2 \oplus b_3, \ldots, b_{j-1} \oplus b_j\}. \tag{6}$$

Then, according to rule 2, for some i,

$$s = \{a_i \oplus a_{i+1}, a_{i+1} \oplus a_{i+2}, \ldots, a_{i+j-2} \oplus a_{i+j-1}\}, \tag{7}$$

where it is understood that the code makes a cycle so that after n is reached the indices are started from 1 all over again. The expressions of s in equations (4) and (5) must be equal. This is only possible if

$$\{a_i, a_{i+1}, \ldots, a_{i+j-1}\} = \{b_1 \oplus k, b_2 \oplus k, \ldots, b_{j-1} \oplus k\}, \tag{8}$$

where $k = a_i \oplus b_1$. This means that each element $p \in P$, which is longer than 1, appears in C either directly or in a transformed form $\tilde{p} = p \oplus k$, where k is some constant. An element may start at the end

of C and continue in the beginning so that the code must be treated here as a cycle.

Going back to the 8-bit codes, we observe that the elements of P which are not present in $C^{(8a)}$ are $\{5,13\}$ and $\{6,15\}$. By taking $k = 13$ we obtain $\{5,13\} \oplus 13 = \{16,0\}$, which are the last and first elements of the code sequence. Hence $\{5,13\}$ appears in the (cyclic) code sequence in a transformed form. On the other hand, if $k = 16$, the transformation of $\{6,15\}$ is $\{6,15\} \oplus 16 = \{10,3\}$. This also appears in the sequence.

The corresponding check for $C^{(8b)}$ is as follows. The elements of P which are not present in $C^{(8b)}$ are $\{6,15\}$ and $\{3,7,16\}$. If $k = 15$, $\{6,15\} \oplus 15 = \{13,0\}$, and if $k = 13$, $\{3,7,16\} \oplus 13 = \{10,14,15\}$. Again, we observe that both transformed elements appear in the code.

Hence we have derived from rule 2 a condition which indicates that a code can be constructed only from those elements of P which are longer than 1. For this purpose some of the elements must be transformed in a proper way. This property has even more important consequences which contain a means of putting the original and transformed elements of P together to make a code.

Starting from equations (5)–(8), where $s \in S$ corresponds to some $p \in P$ which is not in a code, we see that all elements of the string $\{a_i, a_{i+1}, \ldots, a_{i+j-1}\} \in C$ corresponding to s must belong to different elements of P. Namely, if two successive elements $\{a_l, a_{l+1}\}$ would belong to a same element of P, this element should be a part of the code. This is not possible because $a_l \oplus a_{l+1} \in s$, and this s is associated with a p which is not a part of the code. Especially, this means that $a_i = b_1 \oplus k$ must be the last element of some p' in the code and $a_{i+j-1} = b_j \oplus k$ the first element of some p'' in the code. Note here that the length of p' or p'' may be equal to 1. Opposite statements are also true. If p is longer than 1 and belongs to a code and its last element is a_l, then $a_{l-1} \oplus a_l \in C_{c1}$ is the last element of the corresponding s. The next element in C_{c1} is $a_l \oplus a_{l+1}$, and it must be the first element of some s'. Since a_l and a_{l+1} do not belong to the same $p \in P$, s' must be associated with some p' such that a_l is the first element of $\tilde{p}'$. Hence the last element of p is equal to the first element of some $\tilde{p}'$ which is in the code. Similarly, one can show that the first element of some p which is longer than 1 and is in the code is equal to the last element of some $\tilde{p}'$.

These results mean that a code can be constructed from those elements of $p \in P$ which are longer than 1 and their transformations $\tilde{p} = p \oplus k$ by making a string in which the last number of each element is equal to the first number in the next one. In this string every second member must be an original element of P and every second must be a transformed one. Because each element appears in a code either in the original or transformed form but not in both ways, the number of elements in the complete cyclic code sequence is $n/2$. Finally, since every second element is original and every second is transformed, there will be $n/4$ elements of both kinds. Thus a cyclic code sequence can be written in the form

$$\begin{aligned} C_c &= \{p_1, p_2 \oplus k_1, p_3, p_4 \oplus k_2, \ldots, p_{n/2-1}, \\ &\quad p_{n/2} \oplus k_{n/4}, p_1\}, \end{aligned} \tag{9}$$

where the last and first numbers in the successive elements are always the same. The final code C is obtained by removing the duplicate numbers at the joints of the elements and the end part $\{p_{n/2} \oplus k_{n/4}, p_1\}$ which closes the cycle.

The transformed elements for 8-bit codes were already calculated above. The codes can now be constructed by joining the elements according to the principle in equation (9). This gives

$$\begin{aligned} C_c^{(8a)} &= \{\{0,1,2,4,10\}, \{6,15\} \oplus 16, \{3,7,16\}, \\ &\quad \{5,13\} \oplus 13, \{0,1,2,4,10\}\} \\ &= \{\{0,1,2,4,10\}, \{10,3\}, \{3,7,16\}, \\ &\quad \{16,0\}, \{0,1,2,4,10\}\} \end{aligned}$$

$$\begin{aligned} C_c^{(8b)} &= \{\{0,1,2,4,10\}, \{3,7,16\} \oplus 13, \{5,13\}, \\ &\quad \{6,15\} \oplus 15, \{0,1,2,4,10\}\} \\ &= \{\{0,1,2,4,10\}, \{10,14,5\}, \{5,13\}, \\ &\quad \{13,0\}, \{0,1,2,4,10\}\}. \end{aligned}$$

The correct codes $C^{(8a)}$ and $C^{(8b)}$ are reconstructed from these sequences by removing the duplicates of elements 10, 3, and 5 and the two last strings closing the cycles.

The remaining problem in constructing the code is to find the numbers k_i in equation (9) which are used in transforming the elements of P. By studying the known codes it was observed that the k constants in equation (9), except the last one, are always identical. This is the final rule of the regular behavior of strong alternating codes.

Rule 3. If a cyclic sequence constructed from all elements of P according to equation (9) is a code, then

$$k_1 = k_2 = \ldots = k_{n/4-1} = k,$$

where $k = a \oplus a^o$ and $\{a, a^o\} \in P$.

This rule does not say anything about the constant $k_{n/4}$, but its value is not really needed since it only closes the code into a circle. Furthermore, the closing element $p_{n/2}$ is also not needed so that a complete code is formed from $n/2-1$ elements of P which are longer than 1.

The 8-bit codes are so short that they cannot be used for demonstrating rule 3. It is, however, easy to check the validity of this rule using the 16- and 32-bit codes.

In conclusion, the rules imply that the codes satisfy the equation

$$\begin{aligned} C_c &= \{p_1, p_2 \oplus k, p_3, p_4 \oplus k, \ldots, p_{n/2-2} \oplus k, \\ &\quad p_{n/2-1}, p_{n/2} \oplus k_{n/4}, p_1\}, \end{aligned} \tag{10}$$

where each $p_i \in P$ is longer than 1, the constant k is constructed from some element $\{a, a^o\} \in P$ according to the rule $k = a \oplus a^o$, and the last number of each element in the sequence is equal to the first number of the next element. Note that the rules do not imply that a code could be found for each choice $\{a, a^o\} \in P$. In some cases the transformed elements are such that no sequence C_c can be constructed which satisfies the joining condition.

4. Sufficiency of the Three Rules

In section 3 it was assumed that the three rules are valid for strong alternating codes. It was then shown that if this is true, strong codes must satisfy equation (10). In this section it will be proved that all sequences built according to the three rules do make a strong code. Therefore rules 1–3 make sufficient conditions, and it is not necessary to check separately that the codes found by the new method satisfy the strong condition. What has not been shown is that all strong codes would actually satisfy rules 1–3 or equation (10). Therefore we cannot claim that all strong alternating codes could be found using the new search method.

In analogy with equation (4) we first define a function e by the formula

$$a^e = a^o \oplus 1. \tag{11}$$

This means that the result of a^e is always even, and it is obtained from the corresponding a^o by changing the least significant bit to create even parity. Obviously, $a \oplus b$ is even if both a and b are even or odd, and $a \oplus b$ is odd if a is even and b odd or vice versa. Then one can easily show that

$$(a \oplus b)^e = a^e \oplus b^e = a^o \oplus b^o. \tag{12}$$

Let H be a set of all elements of G which belong to some of the sequences $s \in S$. Therefore the elements of H are of the form $a \oplus a^o$, where $a \in G_l$ so that H consists of n elements. If K is a set of all the other elements of G, these can obviously be presented in the form $a \oplus a^e$, where $a \in G_l$. Then K also consists of n elements. In the case of $n = 8$, for example,

$$\begin{aligned} H &= \{1, 3, 4, 6, 11, 13, 14, 16\} \\ K &= \{0, 2, 5, 7, 10, 12, 15, 17\}. \end{aligned}$$

The sets H and K have certain properties which we need in the continuation. Starting from elements $c = a \oplus a^o \in H$ and $d = b \oplus b^o \in H$ and using equation (12), we readily obtain

$$\begin{aligned} c \oplus d &= (a \oplus b) \oplus (a^o \oplus b^o) \\ &= (a \oplus b) \oplus (a \oplus b)^e \in K. \end{aligned} \tag{13}$$

In a similar manner one can show that $c \oplus d \in K$ if $c \in K$ and $d \in K$, and also that $c \oplus d \in H$ if $c \in H$ and $d \in K$ or vice versa.

We next reformulate the weak and strong conditions of alternating codes given by *Lehtinen* [1986] and *Lehtinen and Häggström* [1987]:

If $C = \{a_1, a_2, \ldots, a_n\}$ is a sequence of elements of G and

$$C_i = \{c_{i1}, c_{i2}, \ldots\} = \{a_1 \oplus a_{1+i}, a_2 \oplus a_{2+i}, \ldots\} \tag{14}$$

for $i = 1, 2, \ldots, n-1$, C satisfies the weak condition if the elements of C_i are different and the strong condition if the elements of C_i and C_{i+1} are different for all values of i.

When C is constructed from the elements of P according to rule 1, rule 2 says that all elements of C_1 belong to some $s \in S$, that is, $C_1 \subset H$. The elements of C_2 are of the form $a_j \oplus a_{j+2} = (a_j \oplus a_{j+1}) \oplus (a_{j+1} \oplus a_{j+2})$. According to equation (13), this means that they all belong to K so that $C_2 \subset K$. Continuing in this manner, it is easily seen that $C_i \subset H$, when $i = 1, 3, 5, \ldots$, and $C_i \subset K$, when $i = 2, 4, 6, \ldots$. Since $H \cap K = \emptyset$, also $C_i \cap C_{i+1} = \emptyset$. Therefore

rules 1 and 2 imply that whenever the weak condition is valid, the strong condition is valid also. Thus it remains to be shown that the weak condition follows from the three rules.

When k is selected according to rule 3, it is obtained from some $p_j = \{a, a^o\} \in P$ such that $a \in G_l$ and $a^o \in G_u$. Therefore $k = a \oplus a^o \in G_u$, and because k is odd, it must be the last element of some $p_i \in P$ which is longer than 1. When $i < n/2 - 1$, p_i must be in the code either in its original or transformed form $\tilde{p}_i = p_i \oplus k$. In the former case a transformed p_{i+1} appears in the code so that $k = b \oplus k$, where b is the first element of p_{i+1}. Then $b = k \oplus k = 0$, which is impossible. The latter case is also impossible because the last element of $\tilde{p}_i$ is $k \oplus k = 0$. According to rule 2, p_j or $\tilde{p}_j$ must also be in the code. In the same way as above it is then seen that if $j < n/2 - 1$, the first element of p_{j+1} must be equal to a. This cannot be true, and therefore $i, j \in \{n/2 - 1, n/2\}$. If $i = n/2$, $k \oplus k_{n/4} = 0$ (i.e., equal to the first element of p_1), which gives $k_{n/4} = k$. Then, on the same argument as above, p_j cannot be equal to $p_{n/2-1}$. Therefore the only possibility is that $i = n/2 - 1$ and $j = n/2$. This means that k is the last number in the code, and $p_{n/2} = \{a, a^o\}$. The above analysis also makes it clear that necessarily $k_{n/4} \neq k$.

The next step is to express each code element a_{j+1} in terms of a_j. There are two cases to be studied. In the former one, $a_j \in G_l$, which simply gives $a_{j+1} = a_j^o$. If, on the other hand, $a_j \in G_u$, there is always a $b \in G_l$ such that $a_j = b \oplus k$ and $a_{j+1} = b^o \oplus k$. Therefore $a_{j+1} = (a_j \oplus k)^o \oplus k$. In conclusion,

$$a_{j+1} = \begin{cases} a_j^o & \text{if } a_j \in G_l \\ (a_j \oplus k)^o \oplus k & \text{if } a_j \in G_u \end{cases}. \qquad (15)$$

This result is used in proving that the elements of C_i are all different. For this purpose we first show that each $c_{i,j+1}$ can be expressed in terms of c_{ij}. If both $a_j \in G_l$ and $a_{j+i} \in G_l$ or $a_j \in G_u$ and $a_{j+i} \in G_u$, then $a_j \oplus a_{j+i} \in G_l$, and, using equations (11), (12), and (15), one can easily show that $c_{i,j+1} = c_{ij}^e$. If, on the other hand, $a_j \in G_l$ and $a_{j+i} \in G_u$ or vice versa, then $a_j \oplus a_{j+i} \in G_u$, and one can similarly show that $c_{i,j+1} = (c_{ij} \oplus k)^e \oplus k$. Therefore each c_{ij} can be constructed recursively from c_{i1} using the function

$$f(a) = \begin{cases} a^e & \text{if } a \in G_l \\ (a \oplus k)^e \oplus k & \text{if } a \in G_u, \end{cases} \qquad (16)$$

that is, $c_{i2} = f(c_{i1})$, $c_{i3} = f(c_{i2}) = f^2(c_{i1})$, etc.

Since a^e and $(a \oplus k)^e \oplus k$ are injective and the former expression always gives an even result and the latter one an odd result, $f(a)$ is an injection. Furthermore, because G is finite and $f(a) \in G$ when $a \in G$, $f(a)$ is also a bijection. Therefore there is a smallest n_a such that $f^{n_a}(a) = a$, which means that the numbers $a, f(a), \ldots, f^{n_a-1}(a)$ are different, but after this the cycle starts all over again. Hence $n_b = n_a$ if $b = f^i(a)$ for some i. The definition in equation (16) also directly leads to the condition $f^i(a \oplus k) = f^i(a) \oplus k$ which gives $n_{a \oplus k} = n_a$.

Since always $a_n = k$ (the last element in the code), the last element of C_{c1} is $a_n \oplus a_1 = k \oplus 0 = k$. Therefore none of the elements of C_1 are k. Because the $n-1$ elements of C_1 are all different and they can be constructed recursively from c_{11} using the function f, $n_{c11} \geq n-1$. This means that for any $a \in H \backslash \{k\}$, $n_a \geq n - 1$ (notice that the elements of C_{c1} and H are the same; the difference is that C_{c1} is ordered).

From equation (13) we immediately see that $K = H \oplus k$ because $k \in H$. It is also clear that $K \backslash \{0\} = (H \backslash \{k\}) \oplus k$, since $k \oplus k = 0$. Then the result $n_{a \oplus k} = n_a$ means that $n_a \geq n - 1$ also for each $a \in K \backslash \{0\}$.

Combining the above results we observe that since $G = H \cup K$, the condition $n_a \geq n - 1$ is true for any $a \in G \backslash \{0, k\}$. Clearly, $c_{i1} \in G \backslash \{0, k\}$ for every $i \leq n - 1$, and the number of elements in each C_i is $n - i \leq n - 1$. Thus, when the elements of C_i are constructed by applying the formula $c_{ij} = f^{j-1}(c_{i1})$, also $j - 1 < n_{ci1}$ so that the cycle of f is never closed. Therefore all elements of C_i are different for all values of i. This completes the proof that the weak condition is valid.

In conclusion we first showed that a complete sequence built according to equation (10) satisfies the strong condition if it satisfies the weak condition. Rules 1 and 2 were sufficient to give this proof. Then it was possible to use the additional information in rule 3 to show that the weak condition is valid. Hence the codes built using the three rules are necessarily strong alternating codes.

5. Search of Long Codes

A method of code search can now be easily built from the results in section 3. The first thing to do is to construct the elements of P using rule 1 and the definition in equation (2). It is not necessary to have all elements of P, but it is sufficient to make only those which are longer than 1. This is achieved by using start elements smaller than 2^m.

The next step is to make a set of possible constants

Table 1. Number of Strong Alternating Codes of Different Lengths

Code Length	Number of Codes
$2^{03} = 0000008$	00002
$2^{04} = 0000016$	00002
$2^{05} = 0000032$	00006
$2^{06} = 0000064$	00006
$2^{07} = 0000128$	00018
$2^{08} = 0000256$	00016
$2^{09} = 0000512$	00048
$2^{10} = 0001024$	00060
$2^{11} = 0002048$	00176
$2^{12} = 0004096$	00144
$2^{13} = 0008192$	00630
$2^{14} = 0016384$	00756
$2^{15} = 0032768$	01800
$2^{16} = 0065536$	02048
$2^{17} = 0131072$	07710
$2^{18} = 0262144$	07776
$2^{19} = 0524288$	27594
$2^{20} = 1048576$	24000

k by calculating the exclusive or operation of the two numbers in every two-element $p \in P$. As a result, $n/4$ constants are obtained which are then tested one by one. The transformation $\tilde{p} = p \oplus k$ is made for each p which is longer than 1. However, it is not necessary to transform the element $\{0, 2^1, 2^2, \ldots, 2^m\}$ (which is used as the first element of the code) nor the element which was used in calculating the constant itself.

When the original elements p and their transformations $\tilde{p}$ are available, a construction of a code is tried according to equation (10) by starting from the element $\{0, 2^1, 2^2, \ldots, 2^m\}$ and fitting, alternately, a transformed and an original element in a string. The string can only be continued by an element with its first number equal to the last number in the string. If this can be continued until each element is used either in its original or transformed form, a new strong alternating code has been found. This is successful only for some values of k. In other cases the process comes to a dead end so that an element matching the string can no more be found.

Following these guidelines, a complete search has been made up to the length of $2^{20} = 1{,}048{,}576$ bits. Even longer codes have been found, the longest one consisting of $2^{22} = 4{,}194{,}304$ bits. The number of codes found for the lengths 2^3–2^{20} is given in Table 1. Table 2 shows results of a complete search of 8-, 16-, 32-, and 64-bit codes. Here M is the length of the code cycle, i.e., the number of different phase patterns to be transmitted, and it is always twice the code length. Three dots in the first column indicate that the code is continued from the previous row. Complete results of longer codes cannot be shown because this would be too space consuming. Therefore only samples of 128- and 256-bit codes are presented in Table 3.

6. Discussion

The new codes presented in this paper are long enough to satisfy all practical needs of incoherent scatter radars which one can imagine at the moment. If even longer codes are needed for some other specific reason or just for a general interest, they can be easily calculated along the given guidelines.

The present search method is based on three rules, which were originally experimental. The mathematical analysis of these rules led to a practical scheme for constructing the codes. It was also possible to give a mathematical proof which ensures that every sequence constructed by the present method is indeed a strong alternating code. After this the rules are no more experimental, but they have a firm mathematical basis. Therefore there is no need for a separate check that each new code satisfies the strong condition. However, there is no mathematical proof that all alternating codes of a given length could be found in this way.

It is possible to show that the codes constructed by the present method are closely connected to the divisibility of certain polynomials and to so-called m sequences [see *Sarwate and Pursley,* 1980]. Furthermore, it can be proved that the number of n-bit strong codes built in this way is $\phi(2^m - 1)/m$. Here $\phi(i)$ is the Euler ϕ function, which is the number of integers smaller than i which do not have a common factor with i. This implies that arbitrarily long strong alternating codes exist. A more detailed description of these results, however, lies beyond the scope of the present paper.

The connection of this paper and paper 1 is quite straightforward. Each column in the domino game corresponds to a single Walsh index, and, if the topmost boxes are wiped off from each column, the dot pattern is simply the binary presentation of the Walsh index with the least significant bit on the

Table 2. Complete Walsh Sequences of 8-, 16-, 32-, and 64-Bit Strong Alternating Codes

M	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
...	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{27}	a_{29}	a_{30}	a_{31}
16	000	001	002	004	010	003	007	016								
16	000	001	002	004	010	014	005	013								
32	000	001	002	004	010	020	017	037	021	014	031	035	024	006	015	032
32	000	001	002	004	010	020	036	003	007	016	034	006	015	032	012	025
64	000	001	002	004	010	020	040	006	015	032	064	056	033	067	050	027
...	057	030	061	044	017	037	076	072	063	041	005	013	026	054	036	075
64	000	001	002	004	010	020	040	030	061	072	055	003	007	016	034	070
...	050	011	023	046	024	051	012	025	052	014	031	062	074	041	033	067
64	000	001	002	004	010	020	040	011	023	046	005	013	026	054	021	043
...	017	037	076	065	042	014	031	062	055	022	045	003	007	016	034	070
64	000	001	002	004	010	020	040	044	055	077	033	067	012	025	052	060
...	005	013	026	054	074	035	073	022	045	056	071	027	057	072	021	043
64	000	001	002	004	010	020	040	053	074	022	045	041	050	072	036	075
...	021	043	055	060	012	025	052	077	024	051	071	030	061	011	023	046
64	000	001	002	004	010	020	040	065	036	075	017	037	076	011	023	046
...	071	006	015	032	064	035	073	003	007	016	034	070	005	013	026	054
128	000	001	002	004	010	020	040	100	137	140	036	075	172	053	127	161
...	074	171	055	133	151	014	031	062	144	027	057	136	143	030	061	142
...	033	067	156	003	007	016	034	070	160	077	177	041	103	131	154	006
...	015	032	064	150	017	037	076	174	047	117	101	134	146	022	045	112
128	000	001	002	004	010	020	040	100	162	027	057	136	116	157	055	133
...	104	173	005	013	026	054	130	102	167	035	073	166	036	075	172	006
...	015	032	064	150	042	105	170	003	007	016	034	070	160	022	045	112
...	146	077	177	014	031	062	144	072	165	030	061	142	066	155	050	121
128	000	001	002	004	010	020	040	100	175	006	015	032	064	150	055	133
...	113	152	050	121	137	102	170	014	031	062	144	065	153	053	127	123
...	132	110	154	044	111	157	042	105	167	022	045	112	151	056	135	107
...	162	030	061	142	071	163	033	067	156	041	103	173	012	025	052	124
128	000	001	002	004	010	020	040	100	154	065	153	072	165	006	015	032
...	064	150	074	171	036	075	172	030	061	142	050	121	116	161	017	037
...	076	174	024	051	122	110	175	027	057	136	120	115	167	003	007	016
...	034	070	160	014	031	062	144	044	111	176	021	043	106	140	055	133
128	000	001	002	004	010	020	040	100	047	117	071	163	101	044	111	065
...	153	161	104	056	135	035	073	166	113	060	141	145	154	176	132	022
...	045	112	063	147	151	164	116	072	165	115	074	171	125	014	031	062
...	144	157	170	126	012	025	052	124	017	037	076	174	137	030	061	142
128	000	001	002	004	010	020	040	100	033	067	156	107	024	051	122	077
...	177	145	120	072	165	161	170	152	116	006	015	032	064	150	113	014
...	031	062	144	123	074	171	151	110	012	025	052	124	063	147	125	060
...	141	131	050	121	071	163	175	140	132	056	135	041	103	035	073	166

Table 3. Walsh Sequences of a 128- and a 256-Bit Strong Alternating Code

M	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
...																
256	000	001	002	004	010	020	040	100	200	204	215	237	273	363	143	307
...	012	025	052	124	250	324	055	133	266	350	125	253	322	041	103	206
...	210	225	257	333	063	147	316	030	061	142	304	014	031	062	144	310
...	024	051	122	244	314	035	073	166	354	134	271	366	151	323	042	105
...	212	220	245	317	033	067	156	334	074	171	362	140	301	006	015	032
...	064	150	320	044	111	222	240	305	017	037	076	174	370	164	351	126
...	255	336	071	163	346	110	221	246	311	027	057	136	274	374	175	373
...	162	345	116	235	276	371	167	357	132	265	356	131	263	342	101	203
512	000	001	002	004	010	020	040	100	200	400	420	461	563	767	377	777
...	356	735	252	525	672	145	313	626	074	171	362	744	330	661	162	345
...	712	204	411	402	425	473	547	737	257	537	656	115	233	466	574	751
...	303	607	036	075	172	364	750	300	601	022	045	112	224	450	500	621
...	063	147	316	634	050	121	242	504	630	041	103	206	414	410	401	423
...	467	577	757	317	637	056	135	272	564	770	341	703	226	455	512	605
...	033	067	156	334	670	140	301	602	024	051	122	244	510	600	021	043
...	106	214	430	440	521	663	167	357	736	254	531	642	125	253	526	674
...	151	323	646	134	271	562	764	371	763	366	755	312	625	072	165	352
...	724	270	561	762	365	753	306	615	012	025	052	124	250	520	660	161
...	343	706	234	471	542	725	273	567	776	355	733	246	515	612	005	013
...	026	054	130	260	540	720	261	543	726	275	573	746	335	673	146	315
...	632	044	111	222	444	530	641	123	247	516	614	011	023	046	114	230
...	460	560	761	363	747	336	675	152	325	652	104	211	422	464	571	743
...	327	657	116	235	472	544	731	243	507	636	055	133	266	554	710	201
...	403	426	475	553	707	237	477	556	715	213	427	476	555	713	207	417

top and the most significant bit on the bottom. The transformer corresponds to k, and, just like $k = a \oplus a^o$, it is obtained as a result of exclusive or operation of the two columns of a generator. The transformer is used in modifying the other bones by a columnwise exclusive or operation, which corresponds to the operation $\tilde{p} = p \oplus k$. The principle of playing the game is the same as that in inserting the elements in a string according to equation (10).

In comparison with the second type of alternating codes presented by *Sulzer* [1989, 1993], the strong alternating codes have had, already before this new discovery, a benefit that they are much longer and therefore allow a higher range resolution. The strong alternating codes are also not sensitive to systematic changes in radar power during the transmission of a single pulse, which would be disastrous to Sulzer-type codes. The drawback in strong alternating codes is that the cycle length may be quite long. In the case of 128-bit codes, for instance, the transmission and reception of the whole cycle could take more than 1 s. This means that extremely high temporal resolutions cannot be achieved and also that the code does not necessarily work properly in situations where the parameters of the ionospheric plasma change in a timescale of a second. Such cases are not unusual in the high-latitude ionosphere.

Until now the random codes by *Sulzer* [1986] have been the principal modern modulation method in incoherent scatter experiments with high range resolution. This has been possible in radars with a special random code generator. The present discovery will probably change the situation completely because codes of all lengths are now available with no range ambiguities at all. Therefore it seems advisable to use alternating codes instead of random codes.

In comparing random and alternating codes there is still one more point to consider because in random

codes one can reduce the time resolution at the cost of increase in the level of range ambiguities. It seems probable that the same thing can actually be done with alternating codes also. If a higher time resolution is needed than required by the cycle length, one can use the code as a random code and accept the appearance of range ambiguities, provided that the range ambiguities of a fraction of a code cycle are indeed small. This still remains to be studied, but it can probably be arranged by transmitting the modulation patterns in a proper order and/or randomizing the basic alternating code (a randomized code is obtained from a basic code by changing the signs of some randomly chosen columns in the sign matrix; the resulting matrix still satisfies the strong condition). Then the code would normally work as an alternating code, but, whenever an extremely high time resolution is needed, it could also be used like a random code.

References

Lehtinen, M. S., Statistical theory of incoherent scatter measurements, *EISCAT Tech. Note 86/45,* 97 pp., EISCAT Sci. Assoc., Kiruna, Sweden, 1986.

Lehtinen, M. S., and I. Häggström, A new modulation principle for incoherent scatter measurements, *Radio Sci., 22,* 625–634, 1987.

Markkanen, M., and T. Nygrén, A 64-bit strong alternating code discovered, *Radio Sci., 31,* 241–243, 1996.

Nygrén, T., and M. Markkanen, Long alternating codes, 1, Search by playing dominoes, *Radio Sci.,* this issue.

Sarwate, D. V., and M. B. Pursley, Crosscorrelation properties of pseudorandom and related sequences, *Proc. IEEE, 68,* 593–619, 1980.

Sulzer, M. P., A radar technique for high range resolution incoherent scatter autocorrelation function measurements utilizing the full average power of klystron radars, *Radio Sci., 21,* 1035–1040, 1986.

Sulzer, M. P., Recent incoherent scatter techniques, *Adv. Space Res., 9,* 153–162, 1989.

Sulzer, M. P., A new type of alternating code for incoherent scatter measurements, *Radio Sci., 28,* 995–1001, 1993.

M. Markkanen, Geophysical Observatory, FIN-99600 Sodankylä, Finland. (e-mail: mm@skynet.oulu.fi)

T. Nygrén, Department of Physical Sciences, University of Oulu, FIN-90570 Oulu, Finland. (e-mail: tuomo@skynet.oulu.fi or Tuomo.Nygren@oulu.fi)

(Received April 18, 1996; revised October 1, 1996; accepted October 14, 1996.)