Radio Science, Volume 28, Number 6, Pages 995–1001, November–December 1993

A new type of alternating code for incoherent scatter measurements

Michael P. Sulzer

Arecibo Observatory, Arecibo, Puerto Rico

(Received August 16, 1991; revised July 13, 1993; accepted July 13, 1993.)

An alternating code set is employed as one of several possible techniques used in incoherent scatter radar transmissions to obtain ambiguity-free measurements of autocorrelation functions or spectra with good range resolution. An alternating code set consists of several codes; typically, each successive radar pulse is modulated with a different code in the sequence. This technique is useful in other types of radar transmissions when the target is overspread, assuming the targets have certain statistical properties. Code sets for a new type of alternating code are presented for code lengths 8-12. This new type of alternating code differs from the first kind in two ways: it is subject to a slightly different condition for the elimination of ambiguity, and it is not restricted to lengths that are powers of 2. The new lengths are useful because they allow greater freedom in designing a multipurpose radar waveform best utilizing the available duty cycle of the radar. The alternating code technique is described in detail sufficient to allow an understanding of the two types and to show that the new condition for ambiguity-free measurements is a useful one. A search program was used to find the new sets; the aspects of the program important for decreasing the size of the search space are described. The code sets are presented, and their significance and uses are discussed.

INTRODUCTION

Alternating codes provide a means of making radar measurements of autocorrelation functions or spectra free of range ambiguity when the target is overspread. The radar pulses are modulated by the codes, but this use of phase coding is different from pulse compression, its most common use in radar sounding. Pulse compression is a technique in which a long pulse is made to give nearly the same information about the target as a short pulse. The pulse is modulated with a suitable code, and so the bandwidth of the long pulse is approximately equal to that of the short pulse, allowing the recovery of high-resolution information when the phase of the Fourier transform of the received signal is modified in a manner determined by the code. The signal from the target must change slowly so that there is no significant change during the time interval equal to the transmitted pulse. Otherwise, the decoding cannot be done. If a spectrum of the target is required (or its equivalent, the autocorrelation function (ACF)), sometimes one can use the information from a sequence of properly spaced pulses. In atmospheric sounding, the spectrum of the scattered signal is narrow enough to apply this spectral technique at the lower altitudes but not at the higher ones. The simplest method for obtaining ACF measurements in the E and F regions of the ionosphere involves transmitting a pulse long enough so that all significant lags in the ACF are measured within a single pulse; then one must accept the range resolution set by the length of the pulse. This is inadequate in many cases, and so better techniques are required.

Obtaining autocorrelation functions from overspread scattering media such as the E and F regions of the ionosphere is possible with several techniques. The double-pulse and multiple-pulse techniques [Farley, 1969, 1972] have been used with incoherent scatter radars for more than twenty years. These techniques remove the range ambiguity associated with uncoded long-pulse measurements but cannot make full use of the available transmitter power, especially when the radar uses a modulated klystron circuit. Also, the functions are not complete; the double-pulse measures correlation at one spacing only, and so the spacing must be varied. A multiple-pulse sequence is much better but is missing one or two spacings in typical applications. More recently, techniques which involve phase coding of a long radar pulse have come into use. One of these techniques uses an indefinitely long sequence of random codes [Sulzer, 1986]. Another makes use of a finite-length sequence of predetermined codes [Lehti-

Paper number 93RS01918.
0048-6604/93/93RS-01918/$08.00

nen and Haggstrom, 1987, Lehtinen, 1986]. The two techniques have somewhat different but overlapping ranges of applicability described by Sulzer [1989].

The first technique, random codes, can be easily implemented for very long lengths once the capability to use it is established at a radar facility. As a practical matter, random codes must be generated during the experiment; these codes must be used in the on-line calculation of the correlation estimates or stored for use in off-line computation. The second technique is known as the alternating codes technique, and we shall quite arbitrarily call the alternating codes described by Lehtinen and Haggstrom [1987] type 1 alternating codes. Alternating codes are useful because the modulating waveform can be completely determined before the experiment, allowing implementation at some facilities where it is difficult to implement random codes. It is sometimes useful to use very short codes; this can be easily done with alternating codes but not with random codes. On the other hand, alternating code sequences become very difficult to find as the length increases; for lengths longer than 32 one must use random codes. Type 1 alternating codes have lengths which are a power of 2. We describe here a new type of alternating code, given the label type 2, which is not restricted to lengths which are powers of 2. These codes were mentioned by Sulzer [1989], but at that time the search technique for finding type 2 codes of useful lengths had not been developed. The codes resulting from the search technique are described here. Code sets for lengths 9-12 are presented; these code sets can be used for incoherent scatter radar measurements, possibly with a Barker subcode used on each baud.

Type 1 codes are ambiguity-free for each distinct lag product. Type 2 codes are ambiguity free for each lag of the autocorrelation function (ACF), but the individual lag products are not. It is necessary to add together all of the lag products which contribute to the ACF function at delay τ in order to eliminate the ambiguity. Analysis of incoherent scatter data is often done with a matrix of lag products rather than the ACF; the advantages of the lag product matrix are significant for an uncoded pulse, but less so when using alternating codes. The significance of the difference in the ambiguity of type 1 and type 2 codes will be discussed in the last section.

The primary motivation for finding the type 2 codes is to make available code lengths that are not limited to powers of 2 so that one can design experiments that make the most efficient use of the radar. For example, let us suppose that one were designing a radar pulse sequence in which a number of different types of measurements are to be performed in each pulse on different frequencies. For one of the measurements one might consider using a type 1 sequence in which the codes are 16 bauds long requiring 32 codes in the set, even though this code might be a bit longer than necessary. One might find that this cannot be implemented simultaneously with the other measurements due to limitations such as maximum allowed pulse length and maximum duty cycle, or limitations in the circuitry which generates the modulation waveform. A type 2 set with codes 12 bauds long and 16 codes in the set, as presented here, would leave additional room in the radar pulse for the other measurements and would use less than 40% as much digital memory in the device which generates code signals. The advantages and limitations of the two types of codes will be discussed in the last section.

We shall not describe the problem of ambiguity in radar observations beyond what is necessary to understand what alternating codes are and the differences between types 1 and 2. A very complete and rigorous description of the ambiguity problem is given by Lehtinen [1986]. We must mention, however, the differences between code sets meeting Lehtinen's weak and strong conditions. A set satisfying the weak condition cannot eliminate ambiguity caused by the filters in the receiving system. This can be accomplished by leaving spaces in the transmitted sequence (perhaps transmitting on a different frequency during the gaps), or by using an appropriate phase code on each baud of the main code. A more elegant method is to use a code set meeting the strong condition. A set satisfying the strong condition is good when the so-called square pulse matched filter is used, without additional coding or gaps in the sequence. All of the codes presented here meet only the weak condition. However, as was first shown by Lehtinen and Haggstrom [1987] and for the general case by Sulzer [1989] a weak condition set can be easily converted to a strong condition set. This transformation will be described later because it is very important for finding practical code sets.

The range ambiguity can be understood from the range-time diagram in Figure 1. A coded pulse is transmitted at the lower left corner of the figure. The pulse is long enough to adequately measure the ACF of the scattering medium from lag products spaced by an amount equal to or less than the width of the pulse. The signal from range h_1 is received near the right side of the figure. Signal from another range h_2 (dashed line) is also received during this interval. Can we distinguish the ACFs of the scatter from the two ranges, even though the signals mostly overlap? The zero lag is lost, but the other lags can be recovered. In the next section we shall describe how this is done in just enough detail to understand how alternating codes work and the differences between types 1 and 2. More complete explanations are given in the references listed above. Then the following two sections describe the search for the type 2 codes and how it was accomplished, emphasizing the methods for reducing the size of the search space. Finally, the results are presented and discussed.

We should also point out that it is possible to make code sets that eliminate nearly all ambiguity; they fall short of perfection but might be good enough for any practical application if the proper processing techniques are

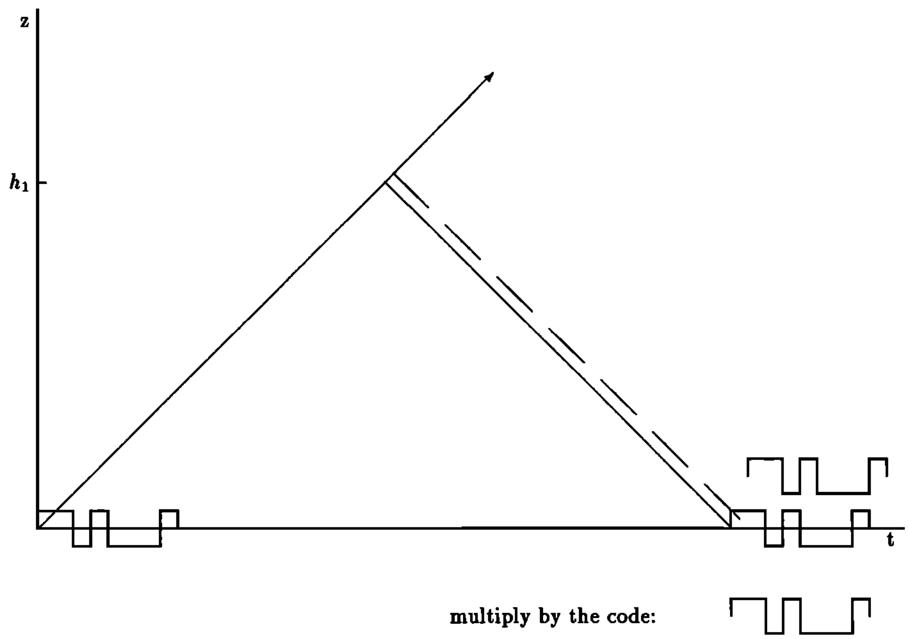

Fig. 1 Range time diagram for long pulse experiments.

used. These sets have the advantage that fewer codes are required in the set for a given code length than for the perfect codes. The techniques for the use of these codes have not yet evolved to the point of application.

DERIVATION OF THE REQUIRED PROPERTIES OF AN ALTERNATING CODE SET OF TYPE 2

Let us suppose that we transmit a binary phase-coded signal using a code which has transitions which occur on multiples of some basic time, the baud length. This means that the transmitted signal changes phase by π radians at various times. Let the code be the following:

$$- \quad - \quad + \quad - \quad - \quad + \quad - \quad + \quad +$$

Now let us suppose we are interested in obtaining the autocorrelation function (ACF) of the scattered signal from some particular range h_1. We identify the time interval in the received signal when signal from this range is present, and we line up the beginning of the code with the beginning of this interval and multiply the samples of the signal by the samples of the code. We then recover the signal from range h_1 as if we had used a uncoded pulse. We can now take the sample ACF for this radar pulse at range h_1 and accumulate it with the sample ACFs of other pulses, which used the same or other transmitted codes.

Now let us consider another range h_2 which is different from h_1 by an amount such that its received signal begins one baud later. When we multiply by the code beginning at range h_1, we "decode" the data from that range, but we further "encode" the data from range h_2, since it is now multiplied by the code resulting from this baud by baud multiplication operation:

$$\begin{array}{cccccccccc}
- & - & + & - & - & + & - & + & + & \\
 & - & - & + & - & - & + & - & + & + \\
\hline
 & + & - & - & + & - & - & - & + &
\end{array}$$

When we compute the ACF of the received signal, all of the lag products from h_1 have the same sign, but from h_2, some have positive signs and others have negative signs. Since the ACF at a particular lag is the sum of all of the lag products with a particular spacing, we merely have to arrange it so that as many products have positive signs as negative signs to make sure that the expected value of the sample ACF at a particular lag from h_2 will go to zero. Note that it is only the expected value that goes to zero; the statistical error from the power at the unwanted range is still present; it is called "clutter" and can be the dominant source of noise in some cases. Since the ACF is accumulated over many radar pulses, we can use different codes on different pulses if this will help in the cancellation. Let us say that we use M different codes and that over the time it takes to transmit the M pulses, the statistics of the scattering process remain the same. We also assume that the signals from the different ranges are uncorrelated.

We can define the multiplication operation we showed above as

$$t^l_{ij} = s_{ij}s_{i(j-l)} \tag{1}$$

where s_{ij} is the jth baud (out of N) of the ith of M codes, and l runs from 1 to $N-2$ and tells how many bauds the "unwanted" range is from h_1. In the operation shown above l is 1. In order that the signal from h_2 not affect the ACF of h_1, the set of sequences t_{ij}^1 must have ACFs that sum to zero for all nonzero lags. This is the same as saying that the sequences t_{ij}^1 must form a complementary code set.

It does us little good to remove the effect of the signal from just the adjacent range. We must do it for all ranges, which means that for each value of l the sequences t_{ij}^l must be a complementary code set. We start with the M codes s_i; from these we generate the codes $t_i^1, t_i^2, t_i^3, t_i^4, \cdots$. Each t_i^l is a set of M codes; each of these sets of codes must be a complementary set.

This is the definition of alternating codes of type 2. Alternating codes of type 1 meet a somewhat different condition. First let us restate the type 2 definition. The ACF of a sequence is given by

$$\rho_{t_i^l}(k) = \sum_{m=1}^{N-l-k} t_{im}^l t_{i(m+k)}^l. \tag{2}$$

The type 2 definition is that for nonzero k

$$\sum_{i=0}^{M} \rho_{t_i^l}(k) = 0. \tag{3}$$

For the type 1 codes we do not perform the summation in the ACF; each lag product, when summed across the M codes, must go to zero. That is, for $m = 1$ to $m = N-l-k$ and nonzero k,

$$\sum_{i=0}^{M} t_{im}^l t_{i(m+k)}^l = 0. \tag{4}$$

This is a much more difficult condition to meet, and it is not a necessary one except in certain circumstances which we will discuss later. It is remarkable that codes meeting this condition can be found, for N being a power of 2, by an extremely fast and elegant search technique described by Lehtinen [1986] and Lehtinen and Haggstrom [1987]. We describe here a slower and less elegant search technique which allows type 2 codes to be found for fairly small, but useful values of N, not restricted to a power of 2.

SIZE OF THE PROBLEM

Consider codes with K bauds, that is, consisting of a sequence of K symbols, where each symbol is a plus or minus one, indicating plus or minus phase in the radar transmission. The symbols used could also be "0" and "1," in which case the code can be labeled by the binary number which corresponds to that sequence, or more conveniently by the octal equivalent. The number of different codes is obviously 2^K, but it is the ACF that matters, and the number of different autocorrelation functions is much less. First, if we replace all ones with zeros, and vice versa, we obtain a different code but always get the same ACF since in the representation using plus and minus one symbols, we just get the negative sequence which has the same ACF. Second, we can flip the sequence end for end; this usually gives a new code but always gives the same ACF. If these two operations always gave a code different from the code operated on and never produced the same code when operating on the same given code, we would have 2^{K-2} different ACFs. However this is not the case, and it can be shown that the number of different ACFs is $2^{K-2} + 2^{\mathrm{int}[(K-2)/2]}$.

Let us suppose that we are working with a code set containing M codes each with N bauds. If we take one of these codes, shift by one baud and multiply we obtain the t^1 sequence which is $N-1$ bauds long. These sequences for each code in the set must form a complementary set, meaning that their ACFs must sum to zero for all nonzero lags. Thus it is the number of ACFs of t^1 sequences which is important, and so we substitute $N-1$ for K in the above equation to determine how many different ACFs we must search over. Table 1 shows how many different sets must be searched for various M and N. For $N = 7$ we can find sets with $M = 4$; for $N > 7$ we must use $M = 8$ since there is no set with $M = 4$ and we can easily show that the next possible number is $M = 8$. The number of sets is found by considering the number of different combinations of P things, where P is the number of different ACFs, taken M at a time, allowing each thing to be used more than once.

TABLE 1. Number of Sets to Search for Various Values of M, the Number of Codes in the Set, and N, the Number of Bauds in the Codes

M (Codes)	N Bauds	$N-1$	Different ACFs	Approximate Number of Sets
4	7	6	20	1×10^4
8	8	7	36	1×10^8
8	9	8	72	2×10^{10}
8	10	9	136	3×10^{12}
8	11	10	272	1×10^{15}
8	12	11	528	2×10^{17}

Note that the number of sets to search through increases by more than 100 times every time N is increased by 1.

MAKING THE SEARCH SIMPLER AND FASTER

Searching through so many sets is very time consuming. We describe here several ways to make the search faster than a simple search through all of the combinations. We begin with one method of making the program for the search simpler. Although by itself it does not make much difference in the search speed, it makes the program simpler, and thus implementation of the other improvements becomes easier.

Suppose that we want to find a type 2 set with M codes each with N bauds. Then from the set of codes s_i we must generate the sets of codes t_i^l and check to see if each is complementary, moving on to the next set s_i whenever we find that one of the sets t_i^l is not complementary. Thus we can think of the innermost operation as a check on whether or not a set is complementary and the first loop around this operation as a loop over l on the sets t_i^l. We shall now show that all of the iterations in this loop can be represented as the check of the complementarity of a single set.

Let us consider all possible sets t_i^1. For any t_i^1 we can find more than one s_i which generates it. Also, it is true that any s_i which generates a given t_i^1 also generates the same t_i^l for $l > 1$, since the value of any element of t_i^l just depends upon whether the elements of s_i in the multiplication have the same or different signs. Thus it makes sense to label the codes by the t_i^1 since they are fewer than the s_i. Now consider the ACFs associated with the codes in t_i^j. Since t_i^1 determines the other sequences, define an extended ACF which includes all of the nonzero lags of all of these ACFs and associate it with the t_i^1. That is, we have only one extended ACF for each value of i. We only have to check to see if the extended ACFs of the M codes of the particular set under consideration are complementary. In this way we can eliminate one loop from the search program. Since ACFs are computed once and stored, it is simpler to have one extended ACF instead of a number of smaller ACFs.

Let us now consider dividing the search into an inner and outer search. The inner search remains a complementary search but only over a small fraction of the possible sets. The outer search eliminates many possible sets very quickly in a manner which we shall show now and then starts the inner search when a group of possible complementary sets is found.

Take any lag of the extended ACF, say the first. It turns out that if we look at the values of this lag for each of the possible t_i^1, we see that many of the codes have the same value. That is, there are fewer different lag values than there are codes. We take each value and associate with it a list of codes which have this value.

A necessary condition for complementarity is that the M values of this lag associated with the M codes of the set sum to zero. Therefore we take a list of these values and search for combinations of these values which sum to zero; this is the outer search. Since there are fewer values than codes, it takes a lot less time to do the search over the list of values than over a list of all ACFs. Each time a combination summing to zero is found, we perform the inner search using the lists of codes associated with each of the values of the set which summed to zero. In this way a huge number of sets are eliminated in the outer search very quickly because we only have to do the inner search when the first lag is complementary.

Since all lags of the ACFs must sum to zero, then instead of using the value of a single lag in the outer search, we can use any linear combination of values of the different lags. If we use the value of a single lag, we do not eliminate enough combinations in the outer loop and spend too much time in the inner loop. On the other hand, if we add the values of the different lags with scale factors designed to produce as many different values as possible, we have too many different values and we save no time. It turns out that very near to the optimum number of different values is obtained if we simply add all of the lags of the extended ACF with unity scale factors. This condition was found experimentally.

Next let us consider the ACFs of codes with three bauds; the three different ACFs are shown in Table 2. The simplest way to make a complementary set uses four codes, and there is only one way: 0, 1, 1, 2. Obviously, one can make complementary sets using different codes but there is only one way to combine the ACFs to make a set with four codes. We also can make a set with eight codes by repeating each ACF, using any code which has that ACF. We can make sets using any multiple of four, but no other lengths are possible. This proves the statement made earlier that if we cannot find a type 2 set with $M = 4$, the next possible value of M is eight.

Let us consider our list of extended ACFs which we are using in the search. Two of the lags in each of these extended ACFs are associated with a code of length three and must be one of the three possible types. Thus we can divide our list of extended ACFs into three sublists, and both the inner and outer searches are set up so that we consider only combinations that use one of type 0, two of

TABLE 2. Codes of Length Three With Unique ACFs

Octal Number	Expansion			ACF		
0	−	−	−	3	2	1
1	−	−	+	3	0	-1
2	−	+	−	3	-2	1

type 1, and one of type 2. This saves time because many combinations are not used in the search.

RESULTS

A computer program was written in C to implement the search. Table 3 shows how many code sets were found for N (number of bauds in the code) between eight and twelve with $M = 8$ (number of codes in the set). The search for $N = 12$ found only one code and covered only a small fraction of the possible sets. It is apparent that if one can find one code set for a particular $N = I$, then in less computer time one can find all of the sets for all $N < I$. It is worthwhile to do these complete searches for the smaller N because one can see that the number of code sets is not a strong function of N. This indicates that there are probably sets to be found for larger N with the same M if one can make the program more efficient so that one could find them.

Each different code set for a given number of bauds represents a different complementary set; that is, not all the same ACFs are used to make the set. Since there are several ways to generate the same ACF with different codes, the one code set found for length $n = 12$ could be realized in many different ways.

Also shown in Table 3 is the average time it took to find a code set for each value of N. These times are for a SPARC station 1+. The search program ran in the background, and although the times given are elapsed time, one can assume that the search got nearly all of the CPU time.

One code set s_i for each length is shown in Table 4. Octal numbers are shown; to obtain the code for the transmitter, perform the binary expansion on each octal number and assign plus or minus phase to the zeros and ones. It does not matter how the assignment is made, and the assignment need not be consistent from code to code. The codes shown are those with the lowest numerical value of the several that would lead to the same ACF.

A set meeting Lehtinen's strong condition is made by using the given set as the first half of the new set. The second half of the set (the second subset of M codes of the necessary $2M$ codes) is generated by taking each code of the first subset expressed as a sequence of plus and minus ones and changing the sign of every other baud.

DISCUSSION

Alternating codes of the original type and those of type 2 have somewhat different advantages and restrictions. The primary advantage of type 1 is that ambiguity is removed on each lag product, while the primary restriction is that codes only exist for lengths that are a power of 2. The primary advantage of type 2 is that the length is not restricted, while the most important restriction is that ambiguity is eliminated only when all lag products contributing to a given delay in the ACF are summed. The following discussion describes the significance of these differences.

There are two reasons for carefully controlling the length of the radar pulse used for alternating code transmission. Both of these require one to avoid an excessively long pulse, first, in circumstances in which one needs other types of modulation within the same pulse and second, in circumstances in which the scatter is clutter-limited. Suppose one uses a pulse of length L and is considering doubling it to $2L$. The advantages of the longer pulse are that one gains both more independent estimates of the delay lengths obtained with the shorter pulse and also one gains new measurements at longer lags.

In situations in which the power from the unwanted ranges is small (not clutter-limited) the importance of the longer pulse depends upon correlation time of the scattering medium. If the longer pulse results in measurements at longer delays which have significant correlation, one would almost always choose the longer pulse because the nonlinear least squares fits to the ACFs would be much better. However, if no new significant delays in the ACF are measured, it might be best to keep the shorter pulse and use the available time on a different frequency for a different type of measurement. The availability of codes allowing lengths between L and $2L$ would assure that all significant delays are measured on the alternating code measurement, while leaving as much time as possible for the other measurement.

In situations in which the power from the unwanted ranges contributes most of the noise (clutter-limited), it is

TABLE 3. Number of Sets Found and Approximate Time It took for Each Length N

N Bauds	Number of Sets Found	Search Time Per Set, s
8	86	3×10^{-2}
9	78	3×10^{0}
10	150	1×10^{2}
11	85	3×10^{4}
12	1	1×10^{7}

TABLE 4. Codes for One Set of Each of the Lengths Searched

$N = 8$	$N = 9$	$N = 10$	$N = 11$	$N = 12$
0	161	7	7	167
41	17	306	621	1101
15	124	142	22	262
64	62	364	713	1663
11	13	11	274	524
33	46	121	326	236
23	64	73	206	554
26	21	64	243	371

even more important to choose the correct pulse length. If one doubles the pulse length in a situation in which one is already measuring all the significant lags, the quality of the measurement decreases. The noise level doubles, which by itself would cause the integration time for constant error to increase by a factor of 4. However, the number of significant independent products rises by about a factor of 2, and so the overall loss would be about a factor of 2 in integration time. The exact loss can only be computed in a specific situation where the ACF is known, but it is clear that it is important to set the pulse length fairly accurately, that is, to better than a factor of 2. In some circumstances it is necessary to put a second measurement on a different frequency within the same pulse in order to make full use of the duty cycle of the radar. This could be an alternating code measurement or another type.

It is a characteristic of the type 2 codes that one must receive and process all of the possible lag products containing information from a given range in order to obtain ambiguity-free performance. Normally, we have all of the lag products, but under certain circumstances the first lag products might be truncated by the receiver cutoff for the lowest ranges or samples might not have been obtained for the last lag products of the highest ranges. Ambiguity-free results are still obtained with type 1 sets because each lag product is free of ambiguity; for certain kinds of data this is significant, as was pointed out by Wannberg [1990]. One recent improvement in the analysis of ionospheric incoherent scatter uses an array of lag products as the data rather than the ACFS's which would result from summing all lags of a given delay. The technique is extremely useful for improving the results from uncoded long-pulse measurements which can suffer significantly from effects of range ambiguity. It also is useful for data made with coded measurements, although its use is usually less important. The technique can be used very simply with alternating codes of type 1 because each lag product is free of ambiguity. It can also be used with the type 2 codes if the ambiguity function is built into the experimental model. Results from ranges where all lags are measured will be nearly the same as if the lags had been added. Results from ranges with some missing lags will have lower signal-to-noise ratio than if all lags were present, but the ambiguity will be eliminated.

Alternating codes require memory space in the device which controls the radar modulation, and so codes which use less space are an advantage. Type 1 codes which meet the weak condition with length N require that the number of codes M is equal to N. On the other hand, type 2 codes usually have M less than N and so use less storage space for a given length of code.

It is clear that further improvements in the search program could be made so that longer sets can be found, although these improvements might not be so easy to implement. It was described above how the properties of the complementary set for codes of length three were used to reduce the size of the search space. It is in principle possible to do this for codes of any length and reduce the search space even more. This was easy to do for codes of length three because there was only one set. For other lengths, many possible sets exist, and it would probably require a computer program to set up the conditions on the search space.

Acknowledgments. The author thanks M. S. Lehtinen for useful discussions concerning the technique. The Arecibo Observatory is part of the National Astronomy and Ionosphere Center and is operated by Cornell University under cooperative agreement with the National Science Foundation.

REFERENCES

Farley, D. T., Incoherent scatter correlation measurements, *Radio Sci.*, *4*, 935-953, 1969.

Farley, D. T., Multiple pulse incoherent scatter correlation function measurements, *Radio Sci.*, *7*, 661-666, 1972.

Lehtinen, M. S., Statistical theory of incoherent scatter measurements, Ph. D. thesis, University of Helsinki, Helsinki, 1986 (EISCAT Tech. Note 86/45)

Lehtinen, M. S., and I. Haggstrom, A new modulation principle for incoherent scatter measurements, *Radio Sci.*, *22*, (4), 625-634, 1987.

Sulzer, M. P., A radar technique for high range resolution incoherent scatter autocorrelation function measurements utilizing the full power of klystron radars, *Radio Sci.*, *21*, (6), 1033-1040, 1986.

Sulzer, M. P., Recent incoherent techniques, *Adv. Space Res.*, *9*, (5), 153-162, 1989.

Wannberg, G., The EISCAT alternating-code algorithm library, paper presented at URSI General Assembly XXIII (session G1), Union Radio Scientifique Inernationale, Ave. Albert Lancaster, 32, B-1180, Brussels, Belgium, August 28-September 5, 1990

M. P. Sulzer, Arecibo Observatory, P.O. Box 995, Arecibo, PR 00613.