University of
Waterloo Research Institute

Code Division for Spread Spectrum Multiple Access

Final Report
Project No. 808-01-03

Prepared for
The Department of Communications
under DSS Contract No. OSU 80-00117

by

Ian F. Blake and Jon W. Mark
Department of Electrical Engineering
University of Waterloo

Scientific Authority
J.L. Pearce
Communications Research Centre
Ottawa.

Rec'd from J. Parvis 8/7/81

OFFICE OF RESEARCH ADMINISTRATION

UNIVERSITY OF WATERLOO

INCORPORATING THE

WATERLOO RESEARCH INSTITUTE

Project No. 808-01-03

Code Division for Spread Spectrum Multiple Access

Final Report

March 16, 1981.

Table of Contents

0. INTRODUCTION

Various aspects of code division multiple access spread spectrum have been studied during the period of this contract and are reported on here. Certain problems associated with the generation and state estimation of maximum length shift register (pseudo random) sequences are first considered. This includes an efficient algorithm to generate primitive polynomials of degree 89 and 127. In addition an algorithm to generate the state of the generating shift register given an initial state and the elapsed number of clock cycles is outlined in section 2. The problem of determining the number of clock cycles between two given states is also considered and shown to be equivalent to the problem of finding logarithms in a finite field. The third section extends previous work on the difficult problem of acquiring and tracking the spreading sequence. The performance of the proposed scheme is compared to previous schemes. An acquisition method based on a reduced state trellis search algorithm is also proposed and discussed. The final section considers three aspects of coded spread spectrum systems.

1. APPLICATION OF FINITE FIELD THEORY

The following two problems are considered;

1. The generation of primitive polynomials of high degree.
2. Two problems of linear feedback shift registers.

Both problems are straight forward applications of finite field theory, except for one of the problems of 2. The purpose of this report is to present the results in a coherent fashion to ease their application to the spread spectrum systems under consideration.

1.1. The Generation of Primitive Polynomials of High Degree

Let N be a prime such that 2^N-1 is a Mersenne prime. While many of the results and techniques do not require these assumptions there are a few valuable simplifications that can be made because of them. In addition these appear to be the values of interest in the applications.

Let α be a root of the primitive trinomial $x^N + x^K + 1$ and thus a primitive element of $GF(2^N)$. Since N is assumed prime, the only subfield of $GF(2^N)$ is GF(2) and every element of $G = GF(2^N)\setminus GF(2)$ is primitive. Every irreducible polynomial of degree N over GF(2) is, in fact, primitive and the minimal polynomial of some element of G. The number of such polynomials is

$$(2^N-2)/N.$$

To generate primitive polynomials of degree N it is sufficient to find the minimal polynomials of elements in G. Each such minimal polynomial corresponds to a cyclotomic coset of the integers modulo 2^N-1 and every cyclotomic coset has order N (a simplification of the general case afforded by the assumption 2^N-1 is a Mersenne prime). Furthermore, all cosets containing the elements $1,3,5,\ldots, \ell$, $\ell < 2^{(N+1)/2}-1$ are distinct ([1], p.262), which is true even when 2^N-1 is not a Mersenne prime. To generate primitive polynomials it then suffices to find minimal polynomials of elements α^J, $J < 2^{(N+1)/2}-1$, J odd, and for distinct J, distinct primitive polynomials result. This is a significant saving since in general it will be tedious to check whether or not two elements are in the same cyclotomic coset. The remainder of the section describes an algorithm to find the minimal polynomial of a given element.

Let α be a root of the primitive trinomial $x^N + x^K + 1$ and thus a primitive element of $GF(2^N)$, and let $Q = \{1,\alpha,\alpha^2,\ldots,\alpha^{N-1}$, which will be used as a basis for $GF(2^N)$ over $GF(2)$. Corresponding to each element of $GF(2^N)$, α^J, there is a representation in term of binary N-tuples:

$$\alpha^J = \sum_{i=0}^{N-1} a_i^{(j)} \alpha^i \equiv (a_0^{(J)}, a_1^{(J)}, \ldots, a_{N-1}^{(J)}) = \underline{a}^{(J)},\ a_i^{(J)} \in GF(2)$$

Define the (N-1) X N binary array $\underline{A}$ whose ith row is $\underline{a}^{(i+N-1)}$, $1 \le i \le N-1$ (i.e. the (N-1) rows of the array are the binary representations of α^i, $N \le i \le 2N-2$ with respect to the basis Q).

Consider now the problem of multiplying two elements of $GF(2^N)$, say $\alpha^J \equiv \underline{a}^{(J)}$ and $\alpha^K \equiv \underline{a}^{(K)}$. Clearly $\alpha^J \cdot \alpha^K = \alpha^{J+K}$ but only the vectors $\underline{a}^{(J)}$ and $\underline{a}^{(K)}$ are available. Notice that

$$\alpha^J \cdot \alpha^K = \left(\sum_{i=0}^{N-1} a_i^{(J)} \alpha^i\right) \cdot \left(\sum_{j=0}^{N-1} a_j^{(K)} \alpha^j\right) = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} a_i^{(J)} a_j^{(K)} \alpha^{i+j}$$

$$= \sum_{\ell=0}^{2N-2} \left(\sum_{i=r}^{s} a_i^{(J)} a_{\ell-i}^{(K)}\right) \alpha^\ell \quad , \quad \begin{array}{l} r = \max\ (\ell-(N-1),0) \\ s = \min\ (\ell,N-1) \end{array}$$

The term in the inner bracket is in GF(2) (either 0 or 1). To find the binary N-tuple representing α^{J+K} we can replace each power of α by its corresponding N-tuple and add those for which the coefficient is nonzero. Since the highest power of α appearing in this term is 2N-2 this is easily achieved by using the rows of the array $\underline{A}$. It will be convenient to refer to this operation of multiplication as

$$\alpha^J \cdot \alpha^K \equiv \underline{a}^{(J)} \dot{\times} \underline{a}^{(K)}$$

A routine to achieve this multiplication of binary N-tuples is quite simple to program (at least in APL). It is actually achieving multiplication of the polynomials

$$f^{(J)}(x) = \sum_{i=0}^{N-1} a_i^{(J)} x^i \quad \text{and} \quad f^{(K)}(x) = \sum_{i=0}^{N-1} a_i^{(K)} x^i$$

modulo the trinomial x^N+x^K+1 and expressing the result as a binary N-tuple.

Define also the N x N array B such that the i^{th} row is the binary representation of $\alpha^{2^{i-1}}$, $i=1,2,\ldots,N$. Notice that this is easily achieved using the above definition of multiplication since if $\underline{b}^{(J)}$ is the J^{th} row then the $(J+1)^{st}$ row is simply

$$\underline{b}^{(J+1)} = \underline{b}^{(J)} \dot{\times} \underline{b}^{(J)}$$

and a recursive construction of the array is straight forward.

<u>The Minimal Polynomial of α^J</u>. The first step in the algorithm is to determine the representation of α^J with respect to the basis Q. This is fairly easy to do with the multiplication routine $\dot{\times}$ and the array B. If

$$J = \sum_{i=0}^{N-1} j_i 2^i \qquad j_i \in \{0,1\}$$

then

$$\alpha^J \equiv \underline{a}^{2^{i_1}} \dot{x} \; \underline{a}^{2^{i_2}} \dot{x} \dots \dot{x} \; \underline{a}^{2^{i_n}}$$

where $\{j_{i_1}, j_{i_2}, \dots, j_{i_n}\}$ are the nonzero coefficients in the binary expansion of J. All of the required vectors are stored as rows in the array B. There are other methods of computing the representation of α^J with respect to the basis Q, but the above method appears to be quite economical and efficient in terms of speed, complexity and memory requirements.

The second step in the algorithm is to construct the N x (N+1) array whose kth column is the binary representation of $\alpha^{(K-1)J}$ with exponents reduced modulo 2^N-1 as necessary, with respect to the basis Q, $1 \le K \le (N+1)$. Notice that each column in this array is α^J times the previous column, and the array is easily constructed recursively. Row reducing this matrix (interchange of rows allowed) will always put it in the form

$$\left[\begin{array}{c|c} & C_0^{(J)} \\ & C_1^{(J)} \\ I_N & \cdot \\ & \cdot \\ & \cdot \\ & C_{N-1}^{(J)} \end{array}\right]$$

and the minimum polynomial of α^J is

$$f^{(J)}(x) = C_o^{(J)} + C_1^{(J)}x + C_2^{(J)}x^2 + \dots + C_{N-1}^{(J)}\, x^{N-1} + X^N .$$

<u>Example</u>. Let N=5 and let α be a root of the primitive polynomial x^5+x^2+1. The basis Q of $GF(2^5)$ over GF(2) is $\{1,\alpha,\alpha^2,\alpha^3,\alpha^4\}$, and the array $\underline{A}$ is

$$\underline{A} = \begin{bmatrix} \underline{a}^{(5)} \\ \underline{a}^{(6)} \\ \underline{a}^{(7)} \\ \underline{a}^{(8)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

while

$$B = \begin{bmatrix} \underline{a}^{(1)} \\ \underline{a}^{(2)} \\ \underline{a}^{(4)} \\ \underline{a}^{(8)} \\ \underline{a}^{(16)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

To find the minimal polynomial of, say, α^{19}, notice first that

$$19 = 1.2^4 + 0.2^3 + 0.2^2 + 1.2 + 1$$

and so

$$\alpha^{19} = \underline{a}^{(16)} \dot{x}\, \underline{a}^{(2)} \dot{x}\, \underline{a}^{(1)}$$

One way of viewing the multiplication algorithm for $\dot{x}$ is as follows: to determine $\underline{a}^{(16)} \dot{x}\, \underline{a}^{(2)}$ write

$$\begin{array}{rcl} \underline{a}^{(16)} & = & 1\ 1\ 0\ 1\ 1 \\ \underline{a}^{(2)} & = & \underline{0\ 0\ 1\ 0\ 0\quad\ \ } \\ & & 0\ 0\ 1\ 1\ 0\ 1\ 1 \\ \text{position} & & 0\ 1\ 2\ 3\ 4\ 5\ 6\ . \end{array}$$

By the array $\underline{A}$ a 1 in position 5 "corresponds" to 1 0 1 0 0 and a 1 in position 6 to 0 1 0 1 0. Thus

$$
\begin{array}{rcl}
\underline{a}^{(16)} \dot{\times} \underline{a}^{(2)} & = & 0\ 0\ 1\ 1\ 0 \\
 & & 1\ 0\ 1\ 0\ 0 \\
 & & \underline{0\ 1\ 0\ 1\ 0} \\
 & & 1\ 1\ 0\ 0\ 0 \quad = \underline{a}^{(18)}
\end{array}
$$

By the same technique

$$
\begin{array}{rcl}
\underline{a}^{(18)} \dot{\times} \underline{a}^{(1)} & = & 1\ 1\ 0\ 0\ 0 \\
 & & \underline{0\ 1\ 0\ 0\ 0\ \ } \\
 & & 0\ 1\ 1\ 0\ 0\ 0 \quad = \quad 0\ 1\ 1\ 0\ 0 \quad = \underline{a}^{(19)}
\end{array}
$$

For the second step of the algorithm we form the 5 x 6 array as follows:

$$
\begin{array}{cccccc}
1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
\underline{a}^{(0)} & \underline{a}^{(19)} & \underline{a}^{(7)} & \underline{a}^{(26)} & \underline{a}^{(14)} & \underline{a}^{(2)}
\end{array}
$$

and row reducing this matrix places it in the form

$$
\begin{array}{cccccc}
1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

and the minimum polynomial of α^{19} is $x^5 + x^3 + x^2 + x + 1$. This same polynomial also has α^7, α^{14}, α^{28}, α^{25} as roots (i.e. the set of elements {7,14,28,25,19} is a cyclotomic coset - it is closed under multiplication by 2 mod 31).

The following two tables present a listing of minimal primitive polynomials.

Table I contains two hundred primitive polynomials of degree 89 and Table II has two hundred of degree 127. The polynomials are given in octal representation (0 = 000, 1 = 001, 2 = 010, 3 = 011, ..., 7 = 111) with the coefficients of the highest degree on the left and the constant term on the right. The polynomial opposite J in the table is the minimum polynomial of α^J where α is a root of the first polynomial in the table ($x^{89} + x^{38} + 1$ for N = 89, Table I, and $x^{127} + x + 1$ for N = 127, Table II). Thus for J = 37 and N = 89 the primitive polynomial in octal representation is

40010 00000 00015 00600 02000 00001

and the actual polynomial is

$$x^{89} + x^{77} + x^{48} + x^{47} + x^{45} + x^{38} + x^{37} + x^{25} + 1$$

The octal digits are grouped in sets of 5 for ease of reading only. The "initial" polynomials $x^{89} + x^{38} + 1$ and $x^{127} + x + 1$ were obtained from [2].

Each entry in each table was tested for irreducibility, as a check, according to the following method ([2], [3]). If f(x) is the minimal polynomial under test, form the (N-1) x N array L whose ith row consists of the coefficients of $x^{2i}-x^{i}$, i = 1,2,..., N-1, each row reduced modulo f(x). Then f(x) is irreducible (and hence in our case also primitive), if and only if, the rank of L is N-1.

J	MINIMAL POLYNOMIAL OF α TO JTH POWER
1	40000 00000 00000 00400 00000 00001
3	40000 10000 02000 00400 00000 00001
5	40000 00002 00000 00410 00000 00001
7	40000 01000 00000 00400 40000 00001
9	40000 10000 02000 00402 01000 00001
11	40000 00200 00200 00401 00001 00001
13	40000 00000 00000 63400 00000 00001
15	40000 10000 02040 40400 00200 04001
17	44000 00000 00000 04400 00000 00001
19	40000 00000 00000 00524 00000 00125
21	40000 10000 02022 00444 40400 00001
23	40000 00010 00000 20600 10102 00001
25	40000 00014 00000 00651 00000 00001
27	40000 10020 02010 00402 00402 00401
29	40000 04001 00020 00444 00100 40201
31	40000 20000 40001 04400 10000 01001
33	40000 10200 12205 00400 02001 00041
35	42004 21000 10400 00400 02100 00001
37	40010 00000 00015 00600 02000 00001
39	40000 50000 56000 12400 70000 20001
41	40000 04004 10210 42402 00204 00001
43	40100 00441 10004 00432 04010 20001
45	40002 50002 12040 42612 41000 00001
47	40000 20011 00000 66432 11040 02001
49	40002 00000 16050 60641 40301 40001
51	74000 36000 17000 07400 00000 00001
53	40404 06000 40205 05440 20200 04001
55	40000 02000 03112 01500 00001 10111
57	40000 12001 02500 04534 00000 00125
59	40000 00220 41002 33664 00440 40001
61	40001 00002 02044 00540 12103 02041
63	40000 14200 03040 00610 00000 00001
65	40060 03004 40360 12416 07400 00401
67	40000 40014 22116 72410 42000 40201
69	40000 10040 06512 35725 10002 41201
71	40205 02402 01426 12400 00000 00041
73	40010 00114 06000 41546 42140 10401
75	40040 10004 02001 04467 03340 06001
77	40000 04520 00454 26444 40000 60001
79	41020 01000 04122 52747 34414 60001

TABLE I PRIMITIVE POLYNOMIALS OF DEGREE 89

J	MINIMAL POLYNOMIAL OF α TO JTH POWER					
81	40001	10006	42015	10634	01000	20001
83	40000	02020	24615	52602	40000	04001
85	44000	02002	00200	16452	05000	00001
87	40404	12121	62131	04721	60141	20201
89	60140	30060	00000	00400	03006	01403
91	40203	01007	00402	34513	00220	14001
93	40020	10010	03406	40201	10010	00111
95	40010	10202	22241	50676	40100	00125
97	40002	04540	31651	41045	36020	44001
99	40003	10114	22032	25072	14163	00001
101	40003	10005	30000	53620	00010	00001
103	40020	00040	01307	43700	02416	01401
105	40100	10000	36140	36420	26420	40201
107	40000	11203	47024	52102	25002	11201
109	40200	01006	03030	36530	50240	00041
111	40011	10222	16757	31115	65103	00001
113	40100	50260	73013	53760	72370	01001
115	40000	00100	22042	71404	30360	60001
117	41010	51636	16440	12100	40004	21001
119	44001	01001	01030	54431	03400	00001
121	52504	05002	20121	10400	10000	24001
123	40444	50514	53250	73301	41044	20001
125	40000	10600	06146	31630	71261	40301
127	40000	00010	00000	00402	00001	00213
129	40200	50742	17652	27361	44614	14101
131	40220	05062	67232	51314	16400	21111
133	40000	20041	70672	15673	76500	20125
135	41000	55134	64720	55576	34062	04401
137	42102	30410	06203	60765	14573	43001
139	40005	00657	27577	10472	44701	00001
141	40001	10102	01224	03456	05634	05401
143	40040	04110	43657	23154	76532	40621
145	40102	21412	55770	16707	70200	01001
147	40010	10001	04460	13030	32643	10241
149	40021	21156	60562	25156	66742	10401
151	40100	40152	54507	77334	15340	06001
153	74000	36000	56600	27402	51030	20001
155	40404	26565	76115	47614	53214	03001
157	44044	40361	24143	56732	12410	60001
159	50020	12004	02447	00410	42401	00001

J	MINIMAL POLYNOMIAL OF α TO JTH POWER					
161	40010	00015	06236	41040	00645	12111
163	41001	42205	46657	27762	72021	10101
165	40000	15070	02335	00414	00460	14233
167	40020	00161	44523	34531	03024	34301
169	40201	42570	57145	73425	26052	33111
171	40000	12043	37010	04044	00041	24125
173	41224	11406	62357	33462	06412	20001
175	40110	62315	66204	51524	26556	00041
177	40020	14177	44173	31547	40220	20001
179	40003	00330	42372	55564	47112	06401
181	41000	15027	42132	44366	65011	40601
183	40003	34005	14071	75503	47221	01041
185	40000	13200	03230	02735	02421	01041
187	44022	02302	62374	07364	63501	00001
189	40404	14201	42602	01712	43210	04001
191	73160	35400	16634	52520	00000	20001
193	40400	14103	02131	53460	41215	41041
195	44420	57312	26347	54371	64410	00001
197	40000	10000	23021	23457	44605	00405
199	40012	04351	30652	42265	70512	02101
201	41061	14161	54056	55433	36513	14001
203	40002	16565	22030	10577	02006	01023
205	40030	10646	50400	70301	45064	54401
207	40114	13743	25717	11210	03462	10111
209	41000	70624	35054	57521	57201	25125
211	40041	01100	12231	24140	45641	02401
213	42311	51434	46220	56672	50327	40041
215	40170	24502	37267	52621	20100	00001
217	40000	56010	10557	70051	06420	40401
219	40040	72766	54147	00213	22704	54221
221	44110	33311	53376	70665	12060	03001
223	50002	14012	50040	12640	13062	15241
225	44400	73604	76464	14161	73130	52401
227	40014	14157	70443	40444	27660	40201
229	66214	03113	54004	00222	73310	60001
231	40602	10576	46322	20110	25405	21001
233	40201	06272	10625	31024	67130	10001
235	40000	22531	62321	31130	66666	16025
237	40030	50506	06661	06047	04541	50101
239	42044	14432	02713	34320	21516	64341

J	MINIMAL POLYNOMIAL OF α TO JTH POWER					
241	40001	15244	30672	72410	45502	21043
243	40061	17340	22703	62451	04104	64401
245	40000	20205	35757	74643	54077	64111
247	40102	01153	42174	37416	57665	05325
249	40015	52375	74052	01235	34501	04001
251	40202	66735	26246	57511	10512	22401
253	40342	61260	23664	14053	25314	42001
255	74000	31400	13242	46601	76622	44401
257	40404	62667	72401	55630	14152	00201
259	40400	06006	57403	67222	02520	53001
261	50400	30302	26744	12405	32005	20041
263	40415	13302	13325	62221	31543	60201
265	41001	53066	73342	17574	56467	07201
267	60140	21040	26254	10622	67104	21403
269	40221	57214	43464	70530	70654	20101
271	40003	42235	73750	12306	33315	24101
273	40002	12254	74274	00000	33111	44421
275	40225	51455	07412	21245	52530	50111
277	40051	22114	02536	14736	03661	20201
279	40061	05157	04262	77057	42770	05453
281	40020	13422	60010	07110	44312	04101
283	41124	13704	04632	34141	24323	41311
285	40001	13417	14350	25756	03717	75325
287	40203	13372	26646	53130	56616	42401
289	44223	36416	45667	20717	72222	40001
291	40705	06335	75547	00550	20020	01001
293	62000	01000	12200	24543	61720	36401
295	41632	45452	31567	30730	37305	64621
297	40025	16372	21532	26253	35574	17241
299	52111	55514	13672	60072	57242	70241
301	44050	13402	41425	71270	43403	30401
303	40022	03341	42306	12761	12070	70101
305	40003	12767	64014	70027	70074	74537
307	40000	43252	65200	35462	24423	16141
309	40121	17262	64351	61415	17035	71011
311	40020	52276	11677	72735	71046	72021
313	40022	47066	74773	47127	00215	32401
315	40100	32606	63174	77560	22241	46001
317	40144	42625	36152	11621	34575	24473
319	40000	00003	00504	07403	17740	22301

J	MINIMAL POLYNOMIAL OF α TO JTH POWER					
321	40032	31342	57447	64544	73262	61331
323	45133	10036	65576	54072	16535	25125
325	52423	72057	36356	30345	02535	40041
327	40200	15335	61451	76205	00152	16041
329	40704	25423	52145	27427	52504	44001
331	57262	51457	47574	64443	32602	00401
333	40010	31444	31527	63276	70441	74641
335	44045	51126	70403	62041	37776	11241
337	52565	05142	70206	63504	74363	05041
339	41021	52435	00123	77305	62324	70211
341	42131	47044	65476	03727	47500	52001
343	40014	00666	76042	67410	26452	65675
345	40002	54320	70343	74311	66454	70201
347	40250	25241	00372	54715	43164	21001
349	40140	75536	22120	32522	65614	74425
351	40024	55760	40645	72247	65550	13111
353	40022	05703	01361	04614	67031	76141
355	40300	05775	77162	56101	45473	50463
357	74001	46206	25351	53712	16532	46401
359	41436	05432	73521	00322	54251	10111
361	44440	04717	73373	50736	36041	27125
363	50537	21532	32603	03002	65615	65401
365	40041	17347	50500	36637	00400	16241
367	40604	03662	22642	02250	60311	20001
369	55153	03122	23536	55313	50244	14401
371	40025	00417	15443	51325	14227	03261
373	44207	12212	05231	63265	56152	01201
375	40210	10116	21001	16040	20210	02565
377	41225	46102	45437	10612	07711	55511
379	41132	27345	63030	76074	32722	47001
381	40000	10012	02505	76777	21453	62005
383	40051	07333	74063	72563	73612	70401
385	41001	34025	40424	52676	24372	27201
387	41011	75247	33026	25077	27550	52005
389	40040	07513	63563	65031	36115	53451
391	46235	33722	31124	52244	70023	14201
393	40636	10100	77412	11060	00000	00403
395	71522	71622	70056	62310	64766	27401
397	41001	51231	16656	47167	43404	46511
399	40126	57217	32100	60776	75225	37365

J	*MINIMAL POLYNOMIAL OF* α *TO JTH POWER*								
1	00200	00000	00000	00000	00000	00000	00000	00000	00003
3	00200	00000	00000	02000	00000	00000	20000	00000	00003
5	00200	00000	00000	00000	00000	00100	00000	04000	00003
7	00200	00020	00002	00000	20000	02000	00200	00020	00003
9	00200	00400	00000	00000	00000	00000	00000	00001	00003
11	00200	00000	40000	00000	00000	20000	00040	00000	10003
13	00200	00000	00000	20000	00000	40020	00001	00040	02003
15	00200	00000	00000	02000	00400	00000	20020	00004	01003
17	00200	00000	00000	00001	00000	00001	00400	00401	00403
19	00200	00004	00000	10000	00000	01000	00020	00000	40203
21	00202	00000	20200	00000	00000	00000	00002	02000	00203
23	00200	04000	00020	00000	00000	00040	01000	20000	00103
25	00200	00000	00000	00000	00002	00102	00000	00100	04103
27	00200	00400	00000	00040	00000	00000	00400	00001	02043
29	00200	00000	00004	00000	00100	10000	02000	10200	40043
31	00200	00000	00000	00200	40100	00000	04210	00104	01043
33	00200	00000	00000	02104	00000	00000	20042	00042	00423
35	00200	00020	00402	00040	20004	02000	00200	00220	04023
37	00200	00000	20000	00000	00000	02001	00442	20010	44223
39	00200	01000	00000	02000	10004	00002	20001	00400	02023
41	00200	00000	00010	00000	00000	40404	40002	20020	22223
43	00200	00000	00000	00440	00440	00000	10000	11010	00113
45	00200	00400	00100	00000	00400	01010	00000	04401	10013
47	00200	00000	02000	01002	00440	10002	04410	02004	44113
49	00200	00020	00002	00000	22040	02000	00200	05020	20013
51	00200	00000	00000	02040	00000	00040	20001	00000	22453
53	00200	00000	00100	00004	10040	00402	04000	01040	12013
55	00200	00000	00200	00100	50240	00100	50200	20000	42253
57	00200	00000	00000	02010	00240	24004	20520	40010	50013
59	00200	00200	00200	01040	01000	01200	00000	24050	00253
61	00200	00000	02500	00000	25200	00000	50420	02024	01013
63	00252	52525	25252	52525	25252	52525	25252	52525	25253
65	00200	00000	00000	00000	00000	00401	20004	01202	10527
67	00200	00000	00012	00000	00000	12012	02400	02402	02407
69	00200	00001	00000	02100	02010	40200	25005	20000	22127
71	00200	00000	00000	00004	02010	04201	20122	00051	00007
73	00200	00010	04000	10000	02412	04020	00204	00500	01327
75	00200	00001	02000	02400	20012	04410	24022	01124	44207
77	00200	00020	00002	00002	20000	22050	00654	02265	40267
79	00200	00100	20040	10200	04111	02200	51000	64000	00007

TABLE II PRIMITIVE POLYNOMIALS OF DEGREE 127

J	MINIMAL POLYNOMIAL OF α TO JTH POWER								
81	00200	00400	10002	00004	41111	00202	40044	00111	22067
83	00200	00000	04004	04001	00101	01021	02002	20202	66207
85	00200	00000	00000	01100	00000	00110	00554	00100	15417
87	00200	00000	00444	02200	20100	14044	62202	01030	50007
89	00200	00040	00011	00402	02001	40442	03101	46043	22117
91	00200	00020	00002	01000	20310	06211	14221	10562	30307
93	00200	00200	00000	42200	40000	06104	20300	65014	64217
95	00210	40104	00042	10000	00014	43006	20000	00000	00007
97	00200	01000	04010	00043	00010	43001	00614	02503	16117
99	00200	00400	00001	04106	10610	01410	41141	00145	20307
101	00204	10200	10004	10600	00404	20004	10020	40410	50217
103	00200	00000	00040	40010	60004	30002	12363	06100	32007
105	00202	00000	20600	04001	41010	14042	43416	06340	41657
107	00200	00000	04020	00001	41410	10010	34703	23726	14047
109	00200	00000	00003	00004	00000	04400	07040	30156	00317
111	00200	40100	00060	06010	00011	02000	60400	00120	00007
113	00200	00000	00400	00100	01003	00300	00004	04505	45157
115	00200	00001	00010	00600	46015	00220	15700	23071	43047
117	00200	01400	04000	04002	40012	20040	00546	00367	05317
119	00200	00160	00046	00004	60010	06001	00600	10060	01007
121	00200	00001	30000	00144	00005	01200	06106	70120	22157
123	00200	00000	00060	16000	00010	00204	60000	00100	02047
125	00200	00000	00000	00000	00014	00000	00003	00001	40317
127	00377	77777	77777	77777	77777	77777	77777	77777	77775
129	00200	00000	00000	03340	71560	00000	24250	00124	05051
131	00200	00000	02300	00001	20240	00141	40764	06034	23255
133	00200	00022	00003	26000	32540	03776	40273	30024	11071
135	00200	00600	00200	00000	01200	00010	04030	00043	25375
137	00200	00000	40000	40024	01500	07003	00003	00066	05451
139	00200	00006	00000	24006	00420	53417	41361	17027	71655
141	00200	20060	00010	03600	20540	60106	20005	44246	01471
143	00200	00000	00000	20001	70000	04101	40630	23505	52375
145	00201	00406	01004	12010	14040	71504	61200	42400	03451
147	00202	00010	20300	01400	06120	61707	17033	52104	40655
149	00200	04000	40010	10406	00121	61210	04217	31042	51471
151	00200	00000	04001	06101	01023	40120	01424	13731	06275
153	00200	00400	00000	60060	00043	41061	50115	34442	15251
155	00200	20021	00102	10410	60021	50154	36441	76107	11555
157	00200	00000	10001	42104	13561	04610	26322	63252	22271
159	00200	00000	00000	03106	00000	00200	24043	02022	13515

J	MINIMAL POLYNOMIAL OF α TO JTH POWER								
161	00200	00020	00003	04400	34440	23467	32271	33320	53671
163	00200	00000	21000	06011	04007	23447	46642	76573	30075
165	00200	10000	00000	13011	50406	03022	25750	02331	70251
167	00200	00001	00010	00444	04200	27127	40306	41405	07315
169	00222	22020	02200	02422	20440	00220	55040	10264	24071
171	00200	00404	40101	01022	00206	05144	55346	02511	42175
173	00200	04001	00004	04001	00662	75104	25714	47641	77241
175	00200	00022	00403	02044	30046	47523	46713	24242	61345
177	00204	00200	00024	02240	51042	03246	63641	73507	42571
179	00200	00040	00012	20004	41040	10030	12451	51502	74075
181	00200	00002	01000	22450	30244	52542	63301	26677	57701
183	00200	20000	02012	02200	20060	46156	60252	43703	61245
185	00200	00000	04204	25040	05101	04066	26134	05155	12071
187	00200	00012	00100	10105	03425	05734	01630	14300	27175
189	00252	52525	25252	53777	77777	77777	60000	00000	00001
191	00200	00000	00000	00000	00000	00401	30004	01302	12561
193	00200	00012	10012	50504	42104	77137	66354	51415	06535
195	00200	00000	01005	06001	04026	22126	23503	15531	77211
197	00200	20000	02012	05204	25135	16166	42012	11502	20305
199	00200	00410	04024	02213	06004	01232	44151	01327	56061
201	00200	02000	02403	26010	25236	11772	34313	53721	37635
203	00204	00024	50022	44202	64014	26405	52600	47160	01711
205	00200	00100	24041	32004	54117	53452	14162	02440	22005
207	00200	00411	10020	04140	40327	61432	54374	73717	27721
209	00200	00040	00410	44121	33225	70553	64751	56246	12535
211	00202	22202	02200	21332	22115	30640	22127	24123	47451
213	00200	00010	11140	02642	44762	17351	70435	35452	52445
215	00200	00044	02030	00046	64013	41304	35604	51756	40771
217	00200	00420	01146	00300	60413	16445	20630	06331	12725
219	00200	00200	00201	02412	12052	36244	15002	35717	42611
221	00210	40100	23102	30001	31414	24266	71600	15540	00005
223	00200	01000	04010	06041	11000	17207	30172	75276	05531
225	00200	20401	02143	04116	60616	45071	46661	63650	17165
227	00204	10200	00030	61630	10003	02055	45404	02573	44651
229	00200	00000	04040	06001	21144	11404	65107	01577	34605
231	00202	00040	60600	00102	07140	50770	14544	56371	41771
233	00200	01000	00100	61406	41531	36135	25750	22017	42065
235	00200	00000	01011	02442	04505	76320	15127	00402	06251
237	00200	40300	10240	32254	25043	00212	43737	36202	25005
239	00200	00060	00404	00141	61013	20303	11064	46152	53071

J	MINIMAL POLYNOMIAL OF α TO JTH POWER								
241	00200	14006	00300	06001	70625	34175	26760	24171	17465
243	00200	03400	16002	04072	01513	77574	06135	51203	43451
245	00200	00320	00452	00575	63265	34302	53415	53244	13405
247	00200	00000	30001	00544	00544	70407	22412	53732	46671
249	00200	00000	04140	32003	27372	53453	10471	12155	55065
251	00200	00000	00000	00720	64350	00000	01242	00121	00251
253	00346	01400	00600	00000	00140	00000	00000	00000	00005
255	00200	00000	00000	03340	71560	00000	32164	00072	06437
257	00200	00000	04700	62002	35735	57114	60635	44614	45751
259	00200	00030	50003	07164	30470	55450	43135	66666	75163
261	00200	00620	00160	00110	03110	42511	47040	11003	26225
263	00200	01400	43003	26026	75414	41552	41535	75407	26617
265	00200	14002	40000	11510	34066	13566	66620	45761	75071
267	00200	00070	01407	22221	41523	70727	01373	45174	57643
269	00200	00200	00000	44001	51410	15620	20641	12234	37405
271	00201	00405	01403	12005	37517	01023	42515	51263	65437
273	00202	00004	30340	01701	07737	74654	25675	20264	14351
275	00202	06060	40204	12547	55307	44346	31050	67560	10163
277	00200	04001	00020	06002	01040	21005	77134	15531	34625
279	00204	10400	10014	00655	13435	44245	40635	15603	30017
281	00200	00420	02041	04646	33544	32572	34720	75430	10571
283	00200	01000	10004	67021	66315	01314	16601	71145	20643
285	00200	02100	62004	02157	22512	27747	26400	00176	00005
287	00200	00030	40003	04414	36564	01043	61105	22341	07357
289	00201	00002	20500	45135	06021	20717	57625	20402	06611
291	00200	00044	26013	37166	33222	75360	77701	34350	27703
293	00200	01000	14011	40022	30545	54636	36113	44632	67065
295	00222	02000	00213	10500	31115	10024	60151	30211	02577
297	00200	00400	04100	44001	12053	26025	11045	62400	11771
299	00200	00000	12000	41454	40464	76115	32601	72230	35313
301	00200	01020	02142	00057	00617	70063	11126	15042	40605
303	00204	00200	41202	56146	07772	45030	60353	34445	07057
305	00200	00001	00002	00562	41040	73655	63710	60162	25451
307	00200	00010	44100	63343	37110	35751	43435	24515	70533
309	00200	00020	40302	66263	66472	53237	62372	54536	47465
311	00200	05001	24305	07671	62342	13061	14773	71721	70777
313	00200	00200	50500	10521	45340	05233	02240	21224	20371
315	00252	52525	25252	52527	77777	77777	77777	76525	25253
317	00240	40120	20071	05024	00010	51200	00016	34000	00005
319	00200	00012	10013	54500	47306	47607	11636	31203	57543

J	MINIMAL POLYNOMIAL OF α TO JTH POWER								
321	00200	04001	00012	03474	61715	54757	34005	41573	71471
323	00200	04024	10530	41547	53457	32316	06534	25147	44067
325	00200	00012	50000	12130	06015	57373	57334	03206	00465
327	00200	02000	20021	13101	30211	35355	40451	41525	70763
329	00200	00025	00223	53137	67516	35104	41063	50473	67151
331	00200	00122	20051	30354	74343	33145	67404	30200	40407
333	00200	04500	32226	55367	47635	76403	03164	40203	40605
335	00200	00000	00412	44360	04024	02024	65722	25577	40143
337	00220	22202	06402	07742	60552	23530	61031	53174	23631
339	00200	00001	00044	02744	27041	72412	10374	07310	45517
341	00200	10040	00251	02012	00252	20006	62151	50503	33435
343	00200	00020	01012	06441	20755	34513	27122	17225	65123
345	00200	00210	00000	46150	44256	40201	11513	55162	71241
347	00210	40100	63110	31125	23635	01544	25201	74003	50007
349	00210	03102	15060	64606	34031	00310	00540	07000	30005
351	00200	20441	06041	25120	43034	03361	50162	56111	51773
353	00204	10004	51020	33363	20523	00346	36620	51777	52131
355	00200	10200	12241	46463	56232	13572	41553	23261	20017
357	00202	04041	21254	54505	41414	04347	55457	50110	22735
359	00200	01010	24041	62007	51133	66462	05115	35124	24163
361	00200	00001	01025	03604	37123	57242	64363	72316	17441
363	00200	40101	10040	22434	06112	32751	01524	17236	30307
365	00200	60050	21445	17360	41030	14704	03040	00410	20305
367	00200	14000	00341	52457	22070	50571	67461	71216	72573
369	00200	04401	32026	45173	20673	61603	40171	07727	57431
371	00200	00120	04056	03404	60414	32767	62715	61674	00217
373	00200	00126	30001	62524	01101	24204	51120	42255	50135
375	00200	00000	00040	13000	20054	10627	70000	00140	00063
377	00200	00000	00040	13000	00010	00204	64001	00300	41141
379	00342	60560	00270	54000	00042	50021	20000	00000	00007
381	00377	77777	77777	74000	00000	00000	20000	00000	00003
383	00200	00000	04700	62000	74655	57062	00302	74723	24137
385	00200	00022	00002	02000	21300	02000	20374	10420	04243
387	00200	00622	00367	05330	25222	52140	31367	46530	13707
389	00200	00000	44100	24150	22755	37352	60167	15610	25303
391	00200	10001	10154	40066	62626	71320	33174	20470	63177
393	00200	30000	03510	33413	26324	47774	21712	31170	02603
395	00200	20240	20246	45221	10171	04116	16356	64355	42607
397	00200	00204	00004	10200	34004	71120	46006	76365	27403
399	00203	01006	24324	51503	51464	75253	23341	72770	42737

1.2. Two Problems of Linear Feedback Shift Registers

A linear feedback shift register of length N is started in state $S^{(0)} = (x_{N-1}, x_{N-2}, \ldots, x_0)$ and j clock cycles later is in state $S^{(j)} = (x_{j+N-1}, x_{j+N-2}, \ldots, x_{j+1}, x_j)$. The feedback connections of the shift register are assumed known and the two problems of interest are:

i) given $S^{(0)}$ and M, the companion or "next state" matrix of the shift register, determine an efficient algorithm to compute $S^{(j)}$ for any j, $1 \leq j \leq 2^N-1$.

ii) given $S^{(0)}$, M and $S^{(j)}$, determine an efficient algorithm to compute j, the number of clock cycles between the two states.

The first problem is relatively easy and the only interest is in making sure the algorithm is an efficient as possible. The theory behind the second problem is also quite straight forward but it is shown easily to be equivalent to the problem of finding logs in a finite field, a problem known to be difficult. A "quasi-algorithm" has recently been found for this and its application will be discussed here.

1.2.1 State Generation for Linear Feedback Shift Registers.

The maximum length feedback shift register corresponding to the binary primitive polynomial $f(x) = x^N + a_{N-1}x^{N-1} + \ldots + a_1x + a_o$ is shown in Figure 1.

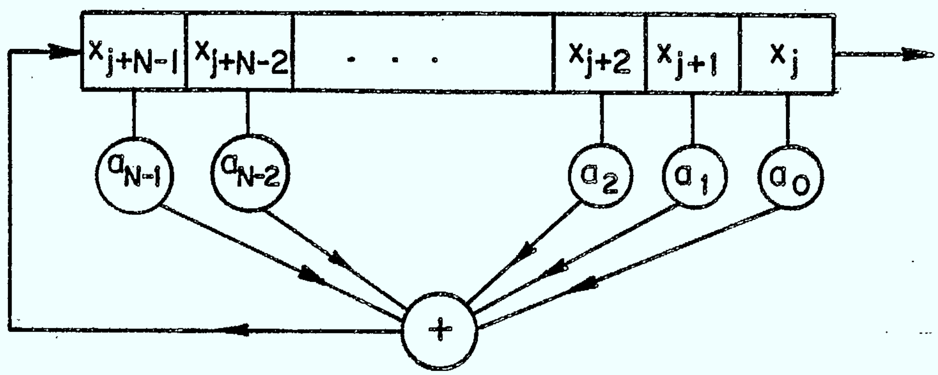

Figure 1.

The NxN companion matrix associated with the shift register is

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & & & \vdots & \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ a_o & a_1 & a_2 & a_3 & & a_{N-2} & a_{N-1} \end{bmatrix}$$

Define the state of the shift register at time j to be $S^{(j)} = (x_{j+N-1}, x_{j+N-2}, \ldots, x_{j+1}, x_j)$. For any given initial state $S^{(o)} = (x_{N-1}, x_{N-2}, \ldots, x_1, x_o)$, the state j clock cycles later is given by $S^{(j)} = M^j S^{(o)}$ or

$$S^{(j)} = \begin{bmatrix} x_j \\ x_{j+1} \\ \cdot \\ \cdot \\ \cdot \\ x_{j+N-1} \end{bmatrix} = \begin{bmatrix} \\ \\ M^j \\ \\ \\ \end{bmatrix} \begin{bmatrix} x_o \\ x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_{N-1} \end{bmatrix}$$

A convenient method for calculating the state j clock cycles later is as follows.

Let $\underline{j} = (j_{N-1}, j_{N-2}, \ldots, j_1, j_o)$ be the binary representation of j i.e.

$$j = \sum_{i=0}^{N-1} j_i 2^i \quad , \quad 0 \le j \le 2^N - 1$$

The quantity M^j can be expressed in the form

$$M^j = (M^{2^{N-1}})^{j_{N-1}} \cdot (M^{2^{N-2}})^{j_{N-2}} \ldots (M^2)^{j_1} \cdot (M)^{j_o}$$

which can be realized in at most $2(N-1)$ matrix multiplications. For large N, say of the order 100, the space requirements for such a computation would be prohibitive. In such a case the following procedure could be used.

Load the matrix M into register A and the intial state $S^{(o)}$ into register B. Replace the contents of B by $M^{j_o} S^{(o)} = S^{(j_o)}$ (Note $M^o = I_N$, the $N X N$ identity matrix). Multiply the matrix in A

by itself to give M^2. Multiply the vector in B by $(M^2)^{j_1}$ to give $S^{(j_o+j_1.2)}$. Multiply the matrix in A by itself to give $M^{2^2} = M^4$. Multiply the vector in B by $(M^{2^2})^{j_2}$ to give $S^{(j_o+j_1.2+j_2.2^2)}$ and so on until the desired state $S^{(j)}$ is reached. The memory storage requirements are thus kept to N^2+N bits and all arithmetic is binary (mod 2). Additional memory of the order of at least N^2 bits will be required for intermediate results. The algorithm is easily formalized into a procedure. The following detailed example illustrates the technique.

Example. Let N=5 and consider the feedback shift register shown in Fig.2 governed by the polynomial $f(x) = x^5 + x^2 + 1$. Choose as the initial state $S^{(0)} = 1\ 0\ 1\ 0\ 0 = (x_4, x_3, x_2, x_1, x_o)$. For illustration

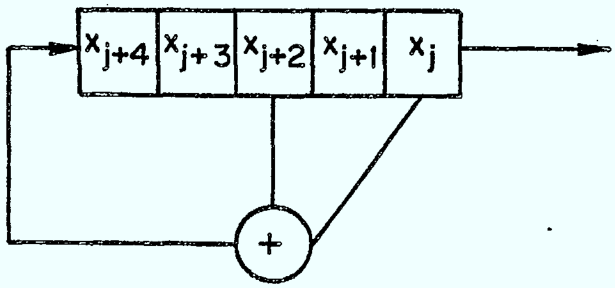

Figure 2.

purposes only, the complete set of states the register cycles through is:

State $S^{(j)}$	x_{j+4}	x_{j+3}	x_{j+2}	x_{j+1}	x_j
0	1	0	1	0	0
1	1	1	0	1	0
2	0	1	1	0	1
3	0	0	1	1	0
4	1	0	0	1	1
5	1	1	0	0	1
6	1	1	1	0	0
7	1	1	1	1	0
8	1	1	1	1	1
9	0	1	1	1	1
10	0	0	1	1	1
11	0	0	0	1	1
12	1	0	0	0	1
13	1	1	0	0	0
14	0	1	1	0	0
15	1	0	1	1	0
16	1	1	0	1	1
17	1	1	1	0	1
18	0	1	1	1	0
19	1	0	1	1	1
20	0	1	0	1	1
21	1	0	1	0	1
22	0	1	0	1	0
23	0	0	1	0	1
24	0	0	0	1	0
25	0	0	0	0	1
26	1	0	0	0	0
27	0	1	0	0	0
28	0	0	1	0	0
29	1	0	0	1	0
30	0	1	0	0	1
31 = 0	1	0	1	0	0

The companion matrix of the register is

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

To compute the state 11 clock cycles later, $S^{(11)}$ note that $11 \sim (j_4\ j_3\ j_2\ j_1\ j_0)$. Load the initial state $S^{(o)} = (1\ \ 0\ \ 1\ \ 0\ \ 0)$ into register B and the companion matrix M into register A . Since $j_o = 1$ replace the contents of B by $M^{j_o}S^{(o)} = MS^{(o)} = S^{(1)}$ where

$$S^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Multiply the matrix (in register A) M, by itself to give

$$M^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Replace the contents of B by $(M^2)^{j_1} S^{(1)} = M^{2 \cdot j_1} S^{(1)} = S^{(3)} =$ $(0\ \ 0\ \ 1\ \ 1\ \ 0)$. Multiply the matrix of A by itself to give $M^{2^2} = M^4$. Since $j_2 = 0$, the state vector in B is not multiplied. Multiply the matrix of A by itself to give $M^{2^3} = M^8$

$$M^8 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

and since $j_3 = 1$ the vector in B is multiplied by M^8 to give

$$S^{(11)} = S^{(1+1.2 + 0.2^2 + 1.2^3)} \quad \text{or}$$

$$= S^{(0.2^4 + 1.2^3 + 0.2^2 + 1.2 + 1.2^0)}$$

$$S^{(11)} = (0 \ \ 0 \ \ 0 \ \ 1 \ \ 1) \ .$$

1.2.2 Determining the Number of Clock Cycles between Given States

Given two states $S^{(0)}$ and $S^{(j)}$ of a linear feedback shift register, it is required to find the number of elapsed clock cycles between them. It is first observed that this problem is essentially equivalent to finding logarithms in a finite field. Using $S^{(0)}$, $S^{(j)}$ and M the states $S^{(1)}$, ..., $S^{(N-1)}$ and $S^{(j+1)}$, ..., $S^{(j+N-1)}$ are found and it is noted that

$$\begin{bmatrix} \updownarrow & \updownarrow & & \updownarrow \\ S^{(j)} & S^{(j+1)} & \ldots & S^{(j+N-1)} \\ \updownarrow & \updownarrow & & \updownarrow \end{bmatrix} = \begin{bmatrix} \\ M^j \\ \\ \end{bmatrix} \begin{bmatrix} \updownarrow & \updownarrow & & \updownarrow \\ S^{(0)} & S^{(1)} & \ldots & S^{(N-1)} \\ \updownarrow & \updownarrow & & \updownarrow \end{bmatrix}$$

or

$$\underline{S}_{(0)} = M^j \ \underline{S}_{(j)}.$$

Each of these matrices can be shown to be invertible and

$$M^j = \underline{S}_{(j)}^{-1} \ \underline{S}_{(0)} \ .$$

The first row of M^j is simply the binary N-tuple which expresses α^j with respect to the basis $\{1,\alpha,\ldots,\alpha^{N-1}\}$. Thus given $S^{(0)}$, $S^{(j)}$ and M, this representation of α^j is found and the problem is to find j, i.e. to find the logarithm of α^j.

This problem has arisen in a variety of contexts, mainly to do with shift register computations but also for computing a finite field and for compromising the security of certain recent public key crypto-systems. An algorithm for this in $GF(2^N)$, 2^N-1 a Mersenne prime, was recently proposed by Herlestam and Johannesson [4]. Little analysis of the running time of this algorithm was given there and it is possible that for many elements in the field $GF(2^N)$, the running time will be unreasonably large. Nonetheless it worked very effectively on all logarithms attempted for all Mersenne primes up to $2^{31} - 1$ and it is a very significant development. It has particularly devestating implications for public key systems implemented which assume this problem could not be solved in a reasonable time. Armed with the existence of such an algorithm it is clear that further research is called for. For the remainder of the section a description of the algorithm is given.

Denote by V the vector space of dimension N over GF(2), 2^N-1 a Mersenne prime and let S be the squaring transformation:

$$S: \; V \longrightarrow V$$
$$x \longmapsto x^2$$

Notice that since the field has characteristic 2, this is a linear transformation on V, $S(x+y) = S(x) + S(y)$. Also note that I, S, S^2, ..., S^{N-1} are distinct and linearly independent (in L(V,V), the vector space of linear transformations of V to itself). Each element of $GF(2^N)$ is identified with a binary N-tuple $(a_0, a_1, \ldots, a_{N-1})$ which is equated with a binary polynomial $a_0 + a_1 x + \ldots + a_{N-1} x^{N-1}$ of degree at most N-1. Multiplication is modulo f(x), a fixed primitive polynomial of degree N with α a root. With this terminology define the linear transformation T on $GF(2^N)$ as:

$$T: \quad GF(2^N) \longrightarrow GF(2^N)$$
$$a(x) \longmapsto xa(x) \quad \text{mod } f(x).$$

If $f(x) = 1 + xg(x)$ then $g(x) \equiv x^{-1} \bmod f(x)$ and the inverse transformation of T is

$$T^{-1}: \quad GF(2^N) \longrightarrow GF(2^N)$$
$$a(x) \longmapsto g(x)a(x) \quad \text{mod } f(x).$$

Since f(x) is primitive it is readily verified that $T^{-1}, T^{-2}, T^{-2^2}, \ldots, T^{-2^{N-1}}$ are distinct as are the transformations

$$T^{-2^r} S^s, \; 0 \le r,s \le N-1.$$

where

$$T^{-2^r} S^s: \quad GF(2^N) \longrightarrow GF(2^N)$$
$$a(x) \longmapsto x^{-2^r} (a(x))^{2^s}.$$

Notice that

$$\log (T^{-2^r} S^s a(x)) = -2^r + 2^s \log (a(x)). \tag{1}$$

These transformations are used in the following manner to find the logarithm of a(x).

1). Set $v = 0$, $a^{(0)}(x) = a(x)$.

2). Set $a_{rs}^{(v)}(x) = T^{-2^r} S^s a^{(v)}(x) \quad 0 \le r,s \le N-1$ and let $a^{(v+1)}(x)$ be any of the polynomials $a_{rs}^{(v)}(s)$ of lowest Hamming weight (fewest nonzero coefficients).

3). If the Hamming weight of $a^{(v+1)}(x) > 1$, increment v by 1 and go to step 2, otherwise stop.

Once a polynomial of Hamming weight 1 has been found, the logarithm of a(x) is easily found by retracing the values of r and s used at each stage and applying equation (1). It has apparently not been shown that the algorithm at step 2 always produces a polynomial of lower weight after the iteration. In addition there is considerable arbitrariness in choosing the next polynomial. Refinements of the procedure are suggested to assist with these problems. One saving can be realized by terminating the procedure at step 3 when the Hamming weight of the polynomial is either 1 or 2 and storing the logs of the N-1 elements of the form $1 + x^i$, $1 \le i \le N-1$.

Further research on either this algorithm or formulating another algorithm should prove useful.

References

[1] F.J. MacWilliams and N.J. Sloane, The Theory of Error Correcting Codes, North Holland Publishing Co., Amsterdam, (1977).

[2] N. Zierler and J. Brillhart, On Primitive Polynomials (mod 2), Information and Control, 13 (1968), 541-554.

[3] E.R. Berlekamp, Algebraic Coding Theory, McGraw-Hill Book Company, New York, 1968.

[4] T. Herlestam and R. Johanneson, On Computing Logarithms over GF(2^p), presented at the 1981 IEEE International Symposium on Information Theory, Santa Monica.

2. COMPLEX SPREAD SPECTRUM SEQUENCES

In most spread spectrum applications binary ± 1 sequences are used as spreading functions and these are invariably chosen to be pseudonoise random sequences or Gold sequences. It is possible however that system performance could be improved by not restricting the sequence alphabet to be binary and an alternative that has received attention in the literature is sequences defined over the complex nth roots of unity, for some integer n. In this section a new construction for such sequences is given.

Let $Q = \{(a_0^j, a_1^j, \ldots, a_{N-1}^j), j = 1,2,\ldots,M\}$ denote a set of M complex sequences of length N, $\sum_{i=1}^{N} |a_i^j|^2 = 1$, $j = 1,2,\ldots,M$. Define the correlations

$$c_{ij}(k) = \sum_{\ell=0}^{N-1} a_\ell^i \, \bar{a}_{\ell+k}^j$$

where either the sequences are to be viewed as periodic or the subscripts are to be reduced modulo N and let

$$C = \max_{\substack{i\neq j \\ \text{or} \\ k\neq 0}} |C_{ij}(k)|$$

Welch [1] has shown that

$$C^{2k} \geq \frac{1}{MN-1}\left[\frac{MN}{\binom{N+k-1}{k}} - 1\right]$$

and as a special case, by choosing k = 1,

$$C \geq \sqrt{\frac{M-1}{MN-1}}$$

These two bounds have proved to be remarkably effective in evaluating sequences and several sets exist which, for a given set of parameters, are either close to or meet these bounds. For large M and N however the bounds become

$$C^{2k} \geq \frac{k!}{(N+k+1)(N+k-2)\ldots(N-1)}$$

and

$$C \geq 1/\sqrt{N}\ .$$

In particular it has been shown [2] that $C \geq 1/\sqrt{N}$ if $M \geq (N+1)^2$. It would appear from this that for $M >> N$ the bound is probably quite weak since it depends only on N and not M. Further work on this bound, which is used, surprisingly, to evaluate both binary and complex sequences, might repair this apparent deficiency. The work of Sarwate [2] uses the Welch bounding technique to investigate the trade-off between the maximum off peak auto-correlation and the maximum cross correlation.

Recently Alltop [3] has determined three sets of sequences of the type of interest here and their construction and properties are briefly reviewed.

<u>i) Quadric phase sequences</u>: Let ω_N be a primitive nth root of unity and define the jth sequence in a set of (p-1) sequences of length N, N an odd integer greater than two and p the smallest prime divisor of it, by

$$a_\ell^j = N^{-1/2}\ \omega_N^{j\ell^2} \qquad \ell = 0,1,\ldots,\ N-1.$$

It can then be shown that

$$C_{ij}(k) = \begin{cases} 1 & \text{if } i=j,\ k=0 \\ 0 & \text{if } i\neq j,\ k=0 \\ \sqrt{p} & \text{otherwise} \end{cases}$$

Establishing these properties requires the use of some involved, but straight forward manipulations of sums of complex exponentials.

ii) Cubic phase sequences: Let p be an odd prime greater than or equal to 5 and define the jth sequence in a set of p sequences of length p, by

$$a_\ell^j = p^{-1/2}\ \omega_p^{\ell^3+j\ell} \qquad \ell = 0,1,\dots,p\ .$$

As before it can be shown that

$$C_{ij}(k) = \begin{cases} 1 & \text{if } i=j,\ k=0 \\ 0 & \text{if } i=j,\ k\neq 0 \\ p^{-1/2} & \text{otherwise}\ . \end{cases}$$

iii) Power residue sequences: Let p be a prime of the form $p = MN+1$ and γ a primitive root of GF(p) and hence a primitive (p-1)th root of unity. If $\beta = \gamma^M$ then β is a primitive Nth root of unity. The elements $\{1,\beta,\beta^2,\dots,\beta^{N-1}\}$ form a subgroup of the multiplicative group of GF(p), GF(p)*, and the elements $\{1,\gamma,\gamma^2,\dots,\gamma^{M-1}\}$ are coset representatives. Thus every nonzero element of GF(p)* can be represented by a product of the form $\gamma^i\beta^j$, $0 \le i \le M-1$, $0 \le j \le N-1$. Define the jth sequence in a set of M sequences of length N by

$$a_\ell^j = N^{-1/2}\ \omega_p^{\gamma^j\beta^\ell}\ .$$

It can be shown that for this set of sequences

$$C_{ij}(k) = \begin{cases} 1 & \text{if } i=j,\ k=0 \\ S_\eta & \text{otherwise} \end{cases}$$

where

$$S_\eta = N^{-1/2} \sum_{\ell=0}^{N-1} a_\ell^\eta\ .$$

In the case where $1, \beta, \beta^2, \ldots, \beta^{N-1}$ forms a cyclic difference set in GF(p) it is straight forward to show that

$$|S_\eta| = N^{-1/2} \left(1 - \frac{1}{M} + \frac{1}{MN}\right)^{1/2} .$$

Unfortunately the only infinite family of such difference sets has $p = 2N + 1$ which produces only a pair of sequences.

The new construction which is proposed here appears to bear some relation to the quadric and cubic phase sequences. Let $f(x) = \sum_{i=0}^{r} a_i x^i$ be a polynomial of degree r over GF(p) for some odd prime p and let

$$S = \sum_{k=1}^{p-1} e(f(k)) \tag{1}$$

where

$$e(x) = e^{\frac{2\pi i}{p} x} .$$

Such sums have been studied in the mathematical literature and applied to certain problems of coding theory. In particular it has been shown [4] that

$$|S| \leq (r-1)p^{1/2} .$$

In fact this result has been generalized considerably [4] to the following situation: let $F(x) = \sum_{i=0}^{r} a_i x^i$ be a polynomial over GF(q), $q = p^n$ and define

$$e'(x) = e^{\frac{2\pi i}{p} t(x)} , \quad t(x) = \sum_{i=0}^{n-1} x^{2^i}$$

i.e. $t(\cdot)$ is a trace function from $GF(p^n)$ to GF(p). Then ([4]) if

$$S' = \sum_{\alpha \varepsilon GF(q)} e'(F(\alpha))$$

it can be shown that

$$|S'| \leq (r-1)p^{1/2}$$

provided that F(x) is not a polynomial of the form $C(x)^p - C(x) + b$, $b \in GF(q)$,

$C(x) \in GF(q)[x]$. For convenience however only the restricted form of equation (1) will be used.

Let α be a primitive element of GF(p) and "label" the co-ordinate positions of the sequences to be constructed by $1 = \alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{p-2}$ consecutively - the sequences will be of length (p-1). To each polynomial f(x) of degree r over GF(p) is associated the complex sequence

$$a^{(f)} = (e_f(\alpha^0),\ e_f(\alpha^1), \ldots,\ e_f(\alpha^{p-2}))\ ,\ e_f(\alpha) = \exp(2\pi i\ \frac{f(\alpha)}{p})\ .$$

The sequences here will not be scaled to have unity length, as were the Alltop sequences mentioned earlier. There are p^{r+1} such sequences and we wish to use the bound of equation (1) to determine a set with good auto-correlation and cross correlation properties. For this purpose, not all polynomials can be used and we investigate those that can be used. Let

$$C_{ff'}(k) = \sum_{=0}^{p-2} e_f(\alpha^{\ell})\ \overline{e}_{f'}(\alpha^{\ell+k})\ .$$

The following simple lemma will assist with the task.

<u>Lemma</u>. i) If $f'(x) = f(x)+b$, $b \in GF(p)$, f, f' of degree $r < (p-1)$ over GF(p) then $|C_{ff'}(k)| = p-1$.

ii) If $f_1(x) = \sum_{i=0}^{r} a_j^{(1)} x^i$, $f_2(x) = \sum_{i=0}^{r} a_j^{(2)} x^i$, then the sequence $a^{(f_2)}$ is a cyclic shift of $a^{(f_1)}$ if and only if $a_j^{(1)} = a_j^{(2)} \alpha^j \quad j = 0,1,2,\ldots,r$

<u>Proof</u> Only part ii) need be proved and it is only necessary to show that $f_1(\alpha^i) = f_2(\alpha^{i+1})$ iff $a_j^{(1)} = a_j^{(2)} \alpha^j$. Since $f_1(\alpha^i) = a_r^{(1)} a^{ir} + a_{r-1}^{(1)} \alpha^{i(r-1)} + \ldots + a_1^{(1)} \alpha^i + a_0^{(1)}$

and

$$f_2(\alpha^{i+1}) = a_r^{(2)}\ \alpha^{(i+1)r} + a_{r-1}^{(2)}\ \alpha^{(i+1)(r-1)} + \dots + a_1^{(2)}\alpha^{i+1} + a_0^{(2)}$$

$$= (a_r^{(2)}\alpha^r)\ \alpha^{ir} + (a_{r-1}^{(2)}\alpha^{r-1})\ \alpha^{i(r-1)} + \dots + a_1^{(2)}\alpha\cdot\alpha^i + a_0^{(2)}$$

and the condition is trivially true if $a_j^{(1)} = a_j^{(2)}\alpha^j$, $j=0,1,\dots,r$. The converse is shown by observing that a polynomial of degree r is uniquely determined by its values at r + 1 points.

The implications of this lemma are clear: if the sequence $a^{(f)}$ is to be included in the set, then no sequence of the form $a^{(f+b)}$, $b\neq 0$, $b\epsilon GF(p)$, is to be included or a correlation of magnitude (p-1) will result. Similarly if the sequence $a^{(f)}$ is to be included in the set, $f(x) = a_r x^r + a_{r-1}x^{r-1} + \dots + a_1 x + a_0$, then no polynomial of the form

$$f'(x) = (a_r\alpha^{ir})\ x^r + (a_{r-1}\alpha^{i(r-1)})\ x^{r-1} + \dots + (a_1\alpha^{r-1})\ x + a_0$$

$$i = 1,2,\dots,p-2.$$

should be included since the corresponding sequence will be the ith shift of the original sequence. Thus the set of all sequences can be divided into "equivalence classes". We consider only the case where all the polynomial coefficients $a_1, a_2, \dots, a_r$ are nonzero since complications in determining the size of each class can occur when some coefficients are allowed to be zero. It is assumed that all sequences in the set have corresponding polynomials with zero constant term to accommodate part i) of the lemma. Of the $(p-1)^r$ such polynomials, only $(p-1)^{r-1}$ of them correspond to cyclically distinct sequences. The cross correlation of any two of these sequences, or any cyclic shifts of them, will then be of the form

$$C_{ff'}(k) = \sum_{\ell=0}^{p-2} e_f(\alpha^\ell)\, \overline{e}_{f'}(\alpha^{\ell+k})$$

$$= \sum_{\ell=0}^{p-2} e^{\frac{2\pi i}{p}(f(\alpha^\ell) - f''(\alpha^{\ell+k}))} \quad .$$

Since $f(\alpha^\ell) - f'(\alpha^{\ell+k})$ can be evaluated as $f''(\alpha^\ell)$ for some appropriately chosen polynomial f'' of degree at most r, the result of equation (1) can be applied to yield

$$|C_{ff'}(k)| \leq (r-1)\sqrt{p} \ .$$

We conclude there exists at leat $(p-1)^{r-1}$ sequences of length $(p-1)$ such that

$$C \leq (r-1)\sqrt{p} \ .$$

Before giving an example of this procedure, notice that the form of this bound is intuitively appealing since the bound increases with the number of sequences in the set. The earlier comment on the Welch bound, in that asymptotically it does not depend on M is now recalled. It is interesting to conjecture that a modified version of the Welch bound, similar in appearance to the above bound, should be possible for a lower bound on C, implying that the present bound is asymptotically "good".

Example. The procedure is illustrated for p=5, r=2. The following polynomials and their corresponding sequences are cyclically distinct and have a correlation, both auto and cross, between any two of them or between any cyclic shifts of them, of at most $\sqrt{5}$. Although there are only $(p-1)^{r-1} = 4$ such sequences, in fact three more sequences could be added to the set without violating the bound, these sequences corresponding to polynomials with some coefficients zero.

polynomial coefficients			sequences
a_2	a_1	a_0	
1	1	0	$(\omega^2, \omega^1, \omega^0, \omega^2)$
1	2	0	$(\omega^3, \omega^3, \omega^4, \omega^0)$
2	1	0	$(\omega^3, \omega^0, \omega^1, \omega^1)$
2	2	0	$(\omega^4, \omega^2, \omega^0, \omega^4)$

References

[1]. L.R. Welch, Lower Bounds on the Maximum Correlation of Signals, IEEE Transactions on Information Theory, Vol. IT-20, 1974, 397-399.

[2]. D.V. Sarwate, Bounds on Crosscorrelation and Autocorrelation of Sequences, IEEE Transactions on Information Theory, Vol. IT-25, 1979, 720-724.

[3]. W.O. Alltop, Complex Sequences with Low Periodic Correlations, IEEE Transactions on Information Theory, Vol, IT-26, 1980, 350-354.

[4]. L. Carlitz and S. Uchiyama Bounds for Exponential Sums, Duke Mathematics Journal, Vol. 24, 1957, 37-41.

3. SYNCHRONIZATION IN CODED SPREAD SPECTRUM SYSTEMS

Spread spectrum communication has a number of applications. By far the most pertinent ones are multiple access, ranging, and anti-jamming. In all cases signal acquisition is an important feature. The acquisition techniques for these cases are somewhat different depending on the interference patterns, the spreading sequences employed and the type of code division multiple access used. Ranging (a single user case) and multiple access (a multi-user case) may employ code sequences that are short compared to anti-jamming. In the latter case it is desirable (and may be imperative) to use a spreading code which is long and difficult to identify by an intruder. A long spreading sequence implies that the receiver may have to acquire the signal when the transmission has been taking place for some time and hence the initial state of the code generator would be unknown. In this case the matched filtering technique proposed in [5] for signal acquisition in a multiple access spread spectrum communications situation would be infeasible for anti-jamming purposes.

In this section we consider the signal acquisition problem in a multi-user situation (multiple access spread spectrum) and in a single user situation (ranging). Acquisition for the latter case when pn sequences are employed as spreading sequences has been considered by Ward [1], Kilgus [2], Ward and Yiu [3] and Hopkins [4]. The anti-jamming case requires further study on the design of "secured" sequences. Signal acquisition in a jamming environment is left for a future study.

3.1 Acquisition of Pseudonoise Signals in a Multi-user Environment

The subsequence matched filtering method for signal acquisition proposed in [5] assumes that the received signal is (time) continuously shifted through the subsequence matched filters (SMFs) so that the maximum output response would correspond to the situation when the matched filter is in exact synchronism (in frequency and phase) with the signal imbedded in the received signal. Thus there is no need to search for the code phase (delay). In [5] it was assumed that one period of the pn sequence corresponds exactly to one message data symbol, so that the proposed acquisition method is a "one" shot per SMF per data symbol strategy. Since there are L SMFs, the method offers L shots at synch per data symbol interval. Also, since each data symbol interval contains one complete period of the pn sequence, the SMFs serve as a cyclic redundancy check for subsequent data symbol intervals. In [5] a cumulative correlator was used to ascertain the correctness of signal acquisition. A drawback in using a cumulative correlator in this manner is that it may be falsely triggered and engaged by a false alarm. In the present study we consider the subsequence matched filtering method with feedback control for signal acquisition in the manner shown in Fig. 3.1.

3.1.1. Continuous-Time Subsequence Matched Filtering for Signal Acquisition

Here we consider the acquisition of pn signals in a direct sequence (DS) CDMA spread spectrum system. The method is readily extendable to an FH/DS hybrid system.

In what follows we assume user #1 to be the desired user so that $\{a_k^{(1)}\}$ is the pn sequence (associated with user #1) which we wish to synchronize. The acquisition model shown in Fig. 3.1 operates as follows: Each SMF is of length $m > n$ bits, where n is the length of the pn sequence generator. Adjacent subsequence matched filters are separated from each

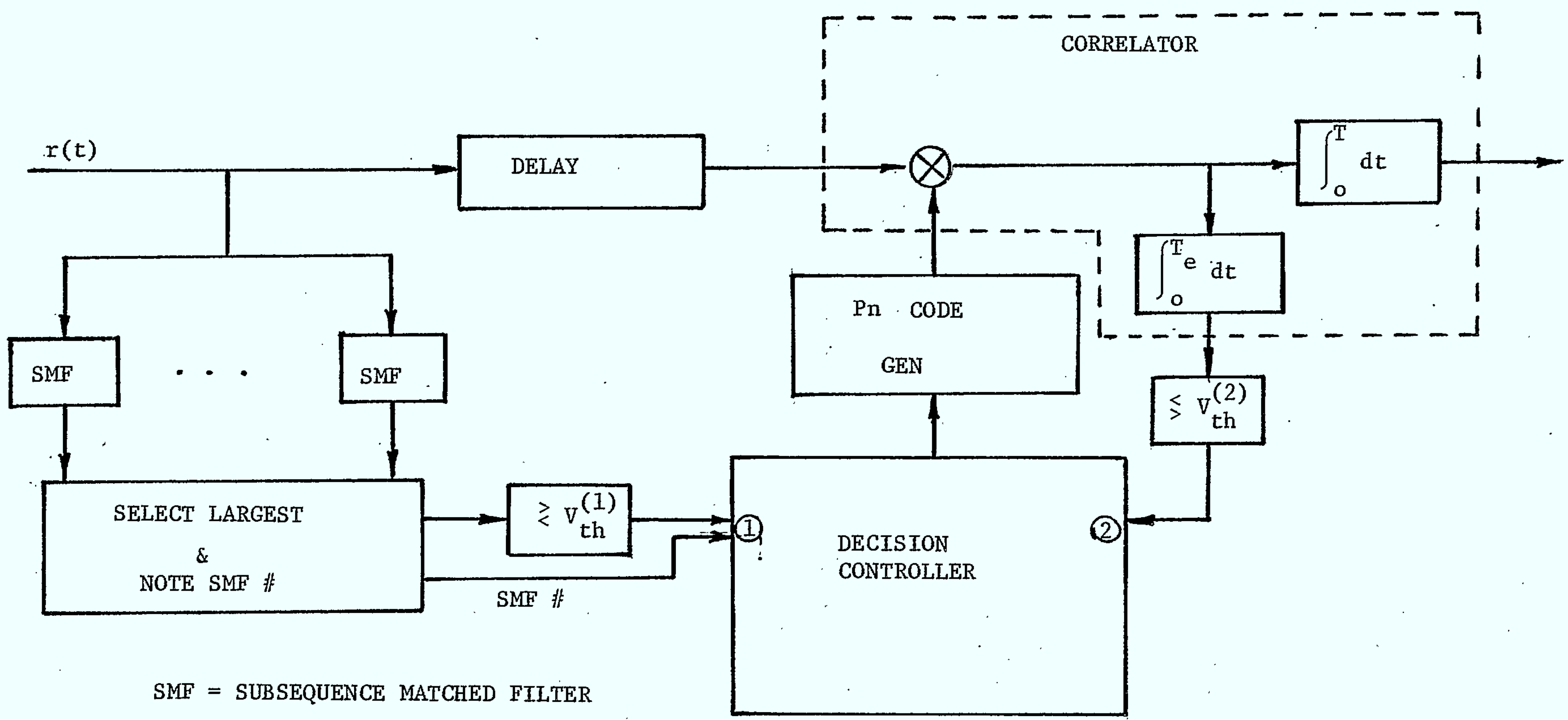

Fig. 3.1 Acquisition Using Subsequence Matched Filtering with Feedback (Baseband Model)

other by B digits, where $m \le B \le \lfloor \frac{N}{L} \rfloor$, $N = 2^n-1$ is the length of the pn sequence, L is the number of SMFs and the symbol $\lfloor \cdot \rfloor$ denotes the integer smaller than or equal to the argument. As the received signal r(t) traverses through the SMFs, the outputs are compared and the maximum selected. Let $\{\alpha_\ell\}_{\ell=1}^{L}$ be the outputs of the SMFs and let $\alpha = \max(\alpha_1, \alpha_2, \ldots, \alpha_L)$. If α exceeds the threshold $V_{th}^{(1)}$, then a synch is declared. Suppose $\alpha = \alpha_\ell > V_{th}^{(1)}$, i.e., the ℓth SMF is in synchronism with the received signal. Then the n-tuple $\underline{a}_\ell^{(1)} = (a_{(\ell-1)B+1}^{(1)}\ a_{(\ell-1)B+2}^{(1)}, \ldots, a_{(\ell-1)B+n}^{(1)})$ is loaded into the local pn sequence generator and the clock started at the indicated phase.

The delay in the main path preceding the correlator accounts for any delay time in the generation of the local pn sequence. The received signal r(t) and the locally generated pn signal $a_1(t)\cos(\omega_c(t-\tau_1) + \theta_1)$, where

$$a_1(t) = \sum_{k=-\infty}^{\infty} a_k^{(1)}\, P_{T_c}(t-k\, T_c), \tag{3.1}$$

$$P_{T_c}(t) = \begin{cases} 1 & 0 \le t \le T_c \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

τ_1 and θ_1 are the delay and phase of the signal component in the received signal and T_c is the chip time, are correlated. The correlation (integration) in the main path carries on for T sec, where T is the data symbol duration, while that in the feedback synch control path carries on for $M\, T_c$ secs $(M \ge m)$. If the synch indication is correct, the correlator (in the feedback path) output will coherently accumulate and is expected to exceed the threshold $V_{th}^{(2)}$ before time $M\, T_c$ sec from the start. If it were a false synch the feedback signal to the Decision Controller in Fig. 3.1 would be zero. Whatever the case may be, based on the two inputs at positions ① and ②, the Decision Controller takes the following actions:

a) If position ② input is a "1", ignore input ① and allow the pn sequence generation and the correlation processes to continue.

b) If input ② is a "0" and input ① is a "1" load pn sequence generator with new state estimate from ①.

c) If inputs ① and ② are both "0", take no action.

Intuitively, the above described actions of the Decision Controller (with feedforward signal from the SMFs and feedback signal from the correlator) are necessary and sufficient to acquire and to maintain synchronization for a bi-phase coded multiple access spread spectrum communication system.

Let the data sequence from user #1 be $\{d_\ell^{(1)}\}_{\ell=-\infty}^{\infty}$ and the corresponding spreading sequence be $\{a_\ell^{(1)}\}$. Consider the acquisition of the $\{a_\ell^{(1)}\}$ sequence and hence reception of the data sequence $\{d_\ell^{(1)}\}$ in the presence of other unwanted signals and additive white Gaussian noise. With binary phase shift keying and since binary phase modulation corresponds to double sideband suppressed carrier modulation, the transmitted signal from user i is

$$s_i(t) = \sqrt{2P} \cdot a_i(t) \cdot d_i(t) \cdot \cos(\omega_c t + \theta_i) \tag{3.3}$$

where $\sqrt{2P}$ is the signal amplitude, $a_i(t)$ is of the form given in (3.1), ω_c is the carrier frequency, θ_i is the phase offset and $d_i(t)$ is given by

$$d_i(t) = \sum_{\ell=-\infty}^{\infty} d_\ell^{(i)} \, P_T(t-\ell\, T), \tag{3.4}$$

$\{d_\ell^{(i)}\}$ is the ith user's data sequence, $P_T(t)$ is of the form given by (3.2) with duration T sec. The received signal is given by

$$r(t) = \sqrt{2P}\, a_1(t-\tau_1)\, d_1(t-\tau_1)\, \cos(\omega_c(t-\tau_1) + \theta_1)$$
$$+ \sqrt{2P} \sum_{i=2}^{K} a_i(t-\tau_i) \cdot d_i(t-\tau_i) \cdot \cos(\omega_c(t-\tau_i) + \theta_i)$$
$$+ n(t) \tag{3.5}$$

The first term on the right-hand side is the desired signal, the second term is the composite of (K-1) unwanted signals and n(t) is a zero mean additive white Gaussian noise with two-sided power spectral density $N_o/2$ watts/Hz. The parameter τ_i is the propagation delay. The impulse response of the ℓth SMF is characterized by

$$f_\ell^{(1)}(t) = \sum_{k=(\ell-1)B+1}^{(\ell-1)B+m} a_k^{(1)}\, P_{T_c}(t-kT_c-\tau_1+\theta_1) \cdot \cos(\omega_c(t-\tau_1)+\theta_1) \tag{3.6}$$

In (3.6) we have anticipated that the (passive) SMFs will eventually be matched with the corresponding subsequence in the desired signal component of r(t). Without loss in generality we can set $\tau_1 = 0$ and $\theta_1 = 0$. The response of the ℓth SMF to r(t) at time m T_c is representable by

$$y_\ell^{(1)} = \int_0^{m\,T_c} r(t)\, f_\ell^{(1)}(t)\, dt$$

Assuming $\omega_c >> 2\pi(m\,T_c)^{-1}$ (this is true for any reasonable choice of m), $y_\ell^{(1)}$ can be shown to be [6].

$$y_\ell^{(1)} = \begin{cases} \sqrt{P/2}\; m\, T_c\, d_j^{(1)} + Z_\ell + n_\ell & \text{, if in synch} \\ z_\ell + Z_\ell + n_\ell & \text{, if not in synch} \end{cases} \tag{3.7}$$

where

$$Z_\ell = \sqrt{P/2} \cdot \sum_{h=2}^{K} [d_{j-1}^{(k)}\, R_{k,1}(\tau_k) + d_j^{(k)}\, \hat{R}_{k,1}(\tau_k)]\cos\phi_k,$$

$$\left.\begin{aligned} R_{k,1}(\tau) &= \int_o^{\tau} a_k(t-\tau) \cdot a_1(t)\, dt, \\ R_{k,1}(\tau) &= \int_{\tau}^{m\,T_c} a_k(t-\tau) \cdot a_1(t)\, dt, \end{aligned}\right\} \quad 0 \le \tau \le m\,T_c$$

$\phi_k = \theta_k - \omega_c \tau_k,$

$$n_\ell = \int_0^{m T_c} n(t)\, f_\ell^{(1)}(t)\, dt,$$

and

$$z_\ell = \sqrt{P/2} \cdot d_j^{(1)}\, r_\ell(i)\ , \qquad i \neq 0$$

$$r_\ell(i) = \sum_{p=(\ell-1)B+1}^{(\ell-1)B+m} a_p^{(1)}\ a_{p+i}^{(1)}\ , \qquad i \neq 0$$

In (3.7) it is assumed that the observation interval is contained in the jth data symbol.

Without loss in generality we can consider $d_j^{(1)} = 1$. When the ℓth SMF is in synch, the signal power is $P\ m^2 T_c^2/2$. In [5] it was shown that z_ℓ is Gaussian with mean and variance given by

$$\mu_o = -\frac{(K-1)\sqrt{P/2}}{N}\, m^2 T_c$$

$$\sigma_o^2 \approx \frac{(K-1)}{N}\, m^2\, T_c^2\, \frac{P}{2}$$

Also, the variance of n_ℓ is given by

$$\sigma_n^2 = \frac{1}{4} N_o\, m\, T_c$$

The non-zero mean μ_o contributes to a d.c. power which is negligibly small ($\frac{m}{N} << 1$) compared to σ_o^2. For all intents and purposes the signal-to-noise power ratio at any one of the SMF outputs under the synchronism condition is given by

$$SNR_{SMF} \approx \left[\frac{K-1}{N} + \frac{N_o}{2PmT_c}\right]^{-1} = \left[\frac{K-1}{N} + \frac{N}{m.\, SNR_o}\right]^{-1} \qquad (3.8)$$

where the approximate symbol is used by virtue of the approximation made for σ_o^2 above, and $SNR_o = 2E/N_o$ is the signal-to-noise ratio in the band-width of the message signal. The energy per data symbol is $E = PT = PNT_c$.

The probability of error associated with synch acquisition using any one of the SMFs is thus approximately given by

$$P_{e/SMF} \simeq Q(SNR_{SMF}^{1/2}) \tag{3.9}$$

where the function $Q(\alpha)$ is defined by

$$Q(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-x^2/2}\, dx$$

The probability of error in acquiring the signal within a data symbol time is then

$$P_{e/symbol} \simeq [Q\ (SNR_{SMF}^{1/2})]^{L} \tag{3.10}$$

The probability of acquisition within m chip times of a data symbol time is then

$$P_{ac/SMF/symbol} \simeq 1 - Q(SNR_{SMF}^{1/2}) \tag{3.11}$$

and the probability of acquisition per symbol is

$$P_{ac/symbol} \simeq 1 - [Q(SNR_{SMF}^{1/2})]^{L} \tag{3.12}$$

In the performance evaluation of signal acquisition the interpretation of signal-to-noise ratio is of critical importance. In his calculation of correctness probability Ward [1,3] defines the signal-to-noise ratio as in the bandwidth of one chip time, which is $\frac{1}{N}$ of SNR_o defined above. On the other hand, Pursley's [6] definition of signal-to-noise ratio is SNR_o defined above. However, Pursley was concerned with post-detection performance and not signal acquisition.

For our purpose, since m chip time coherent accumulation is considered to provide a sufficiently large dynamic range between the coherent peak and the incoherent accumulation of (K-1) unwanted signals plus additive noise, we define the signal-to-noise ratio as in the bandwidth of m chip times, so that

$$SNR_m \triangleq \frac{2\ E_m}{N_o}$$

where $E_m = P\ m\ T_c$. Comparing with Ward's [1] definition $SNR_m = m\ SNR_{Ward}$. From (8) we have

$$SNR_{SMF} \simeq \left[\frac{K-1}{N} + \frac{1}{SNR_m}\right]^{-1}$$

For K = 1 user, $SNR_{SMF} = SNR_m$.

The probability of acquiring the signal within m chip times given the signal is present is given by (3.11). Hence the average acquisition time, $T_{a,SMF}$, for the SMF method is

$$T_{a,SMF} = \frac{m\ T_c + T_e}{[1 - Q(SNR_{SMF}^{1/2})](1 - P_{fd})(1 - P_{fa})} \tag{3.13}$$

where P_{fd} and P_{fa} are the probability of false dismissal and the probability of false alarm.

$$P_{fd} = Q(SNR_{SMF}^{1/2})$$

$$P_{fa} = P_r\ [y(T_e) > V_{th}^{(2)}\ |\ \text{signal absent}]$$

where $V_{th}^{(2)}$ is the threshold in the feedback path and T_e is the examination interval. Choosing $V_{th}^{(2)} = \rho \cdot$ (coherent peak after $M\ T_c$ sec.)

$$= \rho \cdot \sqrt{P/2}\ M\ T_c \qquad , \qquad \rho \le 1$$

then

$$P_{fa} \simeq Q\left[\rho\ \left(\frac{M}{m}\ SNR_m\right)^{1/2}\right]$$

where the approximation is due to neglecting the variance of the unwanted signals. The acquisition is minimized when P_{fa} is made equal to P_{fd} [1].

In this case

$$\rho = \sqrt{\frac{m}{M}}$$

and the threshold becomes

$$V_{th}^{(2)} = \sqrt{m\ M\ P/2}\ \ T_c$$

Under this condition the average time to acquisition, $T_{a,SMF}$, becomes

$$T_{a,SMF} = \frac{m\ T_c + T_e}{[1-Q(SNR_{SMF}{}^{1/2})](1-P_{fa})^2}$$

For small probability of false alarm, the $(1-P_{fa})^2$ term is approximately 1. The probability of synch acquisition and the average acquisition time for m = 1000, n = 16, $N = 2^n-1 = 32,767$, are plotted in Figs. 3.2 and 3.3, respectively. (We have assumed $T_e = m\ T_c$).

The average acquisition time for the RASE [1] and RARASE [3] methods, for small probability of false alarm, are given by

$$T_{a,RASE} = \frac{T_e}{p^n}$$

and

$$T_{a,RARASE} = \frac{T_e}{p^n}\,(p^2 + 3(1-p)^2)^k$$

where T_e is the examination period with minimum value equal to $n\ T_c$, k is the number of 3-input mod-2 adders in the RARASE method [3], and p is related to the signal-to-noise ratio SNR_m by

$$p = 1 - Q[(\frac{1}{m}\ SNR_m)^{1/2}]$$

For K ≃ 100, $N = 2^{16}-1$ and $SNR_m \leq 10$, $SNR_{SMF} \simeq SNR_m$. Then

$$P_{ac/SMF/symbol} \simeq 1 - Q[(SNR_m)^{1/2}]$$

Then, we have

$$T_{a,SMF} = \frac{m\ T_c + T_e}{1 - Q[(SNR_m)^{1/2}]} \tag{3.14}$$

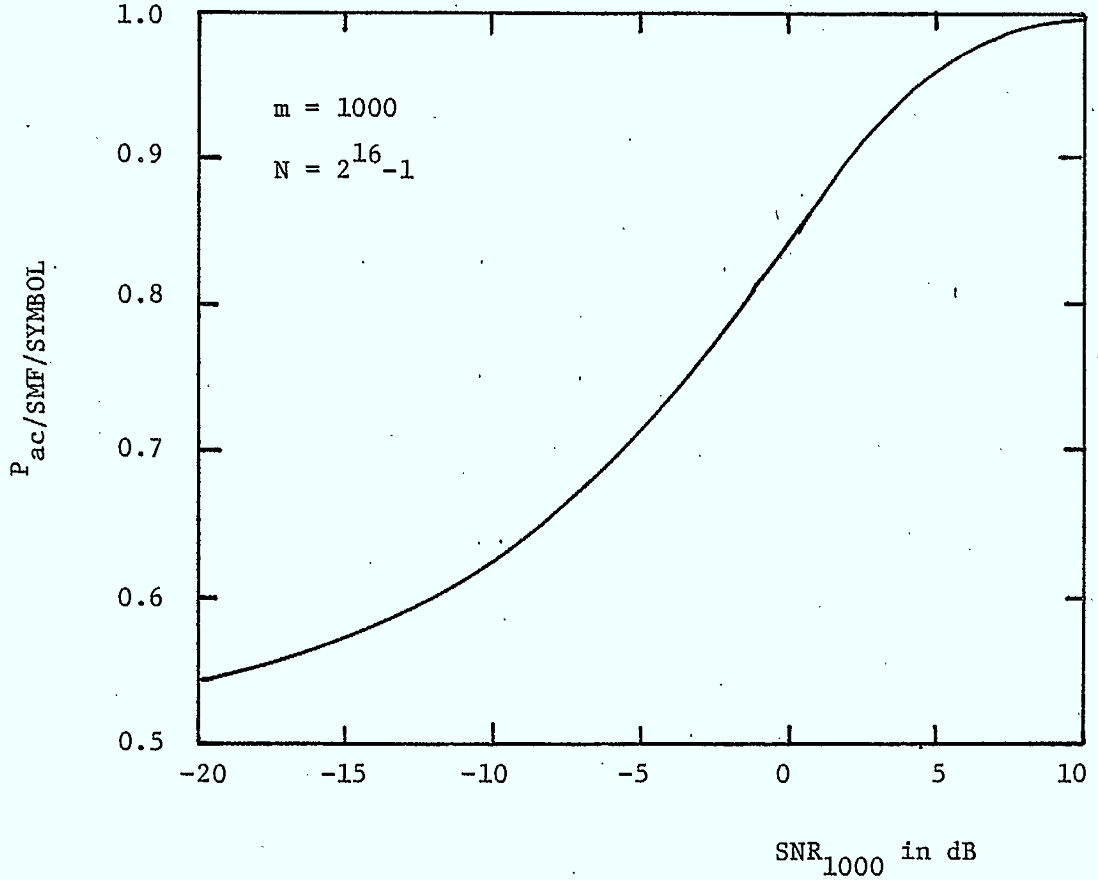

Fig. 3.2 Probability of Acquisition as a Function of SNR_{1000}

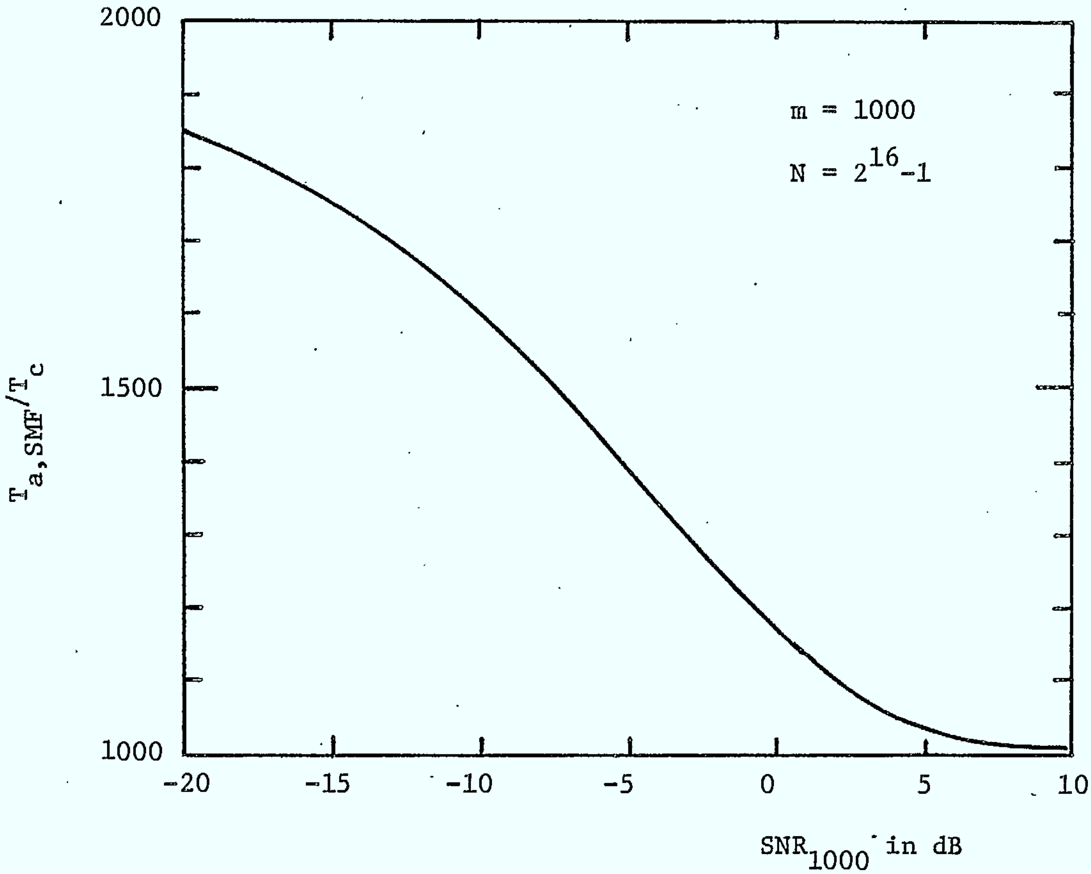

Fig. 3.3 Average Acquisition in Units of T_c as a Function of Signal-to-Noise Ratio

For the case $m = 100$, $n = 16$, $T_e = m\, T_c$, curves for $T_{a,\,RASE}$, $T_{a,\,RARASE}$ and $T_{a,SMF}$ are plotted in Fig. 3.4 as a function of SNR_m. Similar plots for $m = 300$ and 500 are shown in Figs. 3.5 and 3.6. It is observed that the SMF method offers faster acquisition than either RASE [1] or RARASE [3].

It is appropriate to remark here that for the SNR_m's considered the additive noise swamps out the combined interference from unwanted users for $K \leq 100$ in the SMF method. This desirable feature is a consequence of the despreading property of the SMF method. Since the RASE and RARASE methods are designed for acquisition of a single pn sequence in additive noise, they cannot cope with a multi-user environment, as neither of these two methods has the despreading feature during the acquisition mode.

It appears that smaller values of m offers faster acquisition. For despreading purposes however, m should be at least 5 times the shift register length n.

Once the signal has been acquired the correlation in the main path will despread the received signal and reject noise components lying outside of the bandwidth of the message signal. Sampling the correlator output signal at a rate $1/T$ and threshold detecting the samples will produce a "good" estimate of the message sequence $\{d_\ell^{(1)}\}$. Intuitively, encoding the message sequence by a convolutional code (or any other form of error correcting code) will add protection against additive noise at the post-detection stage. A comparison of performance of systems with and without convolutional code is given in section 4.

3.1.2 Discrete-Time Subsequence Matched Filtering for Signal Acquisition

In the preceding section it was assumed that incoming signal r(t) is continuously propagated through the SMFs so that the maximum

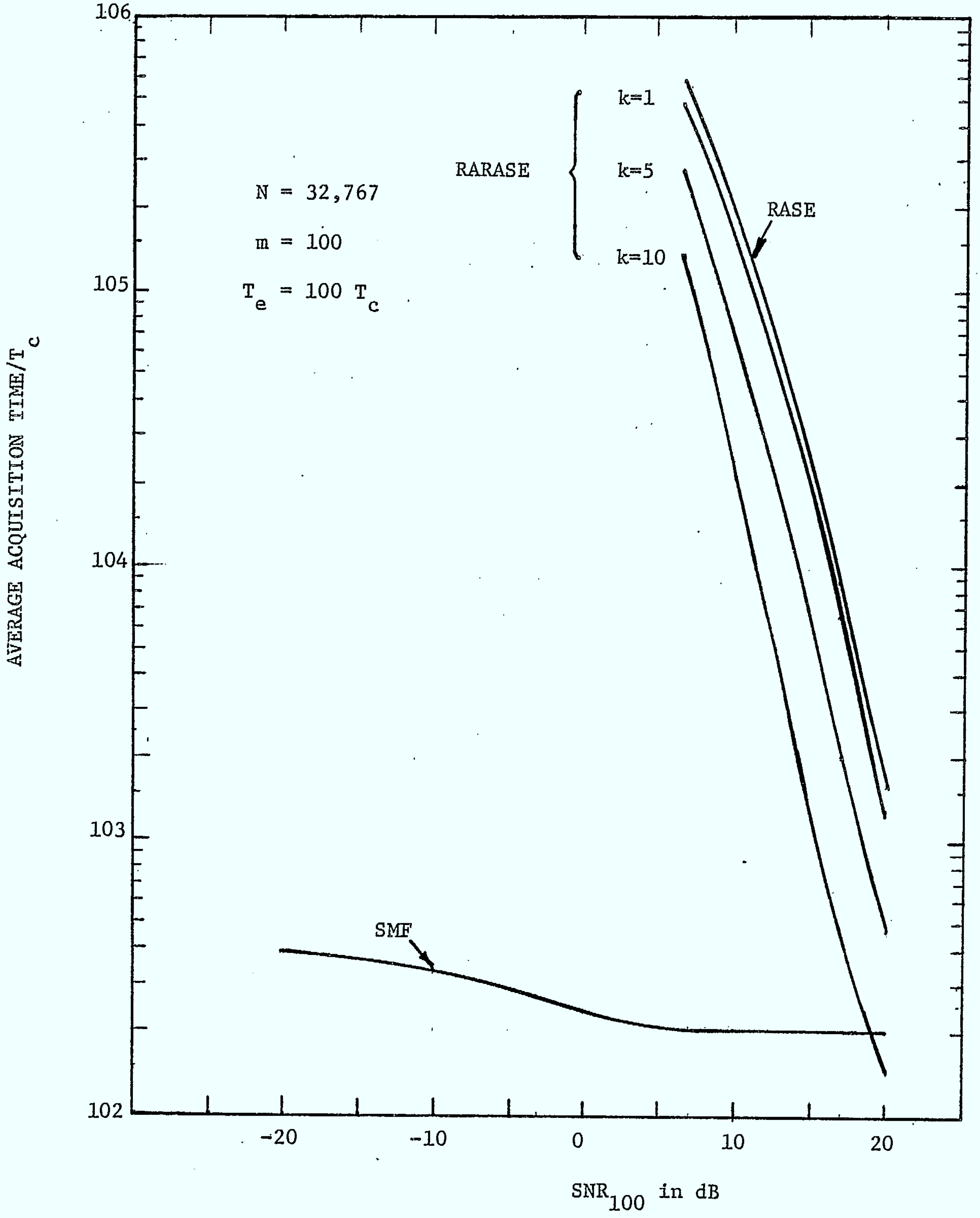

Fig. 3.4 Average Acquisition Time vs Signal-to-Noise Ratio

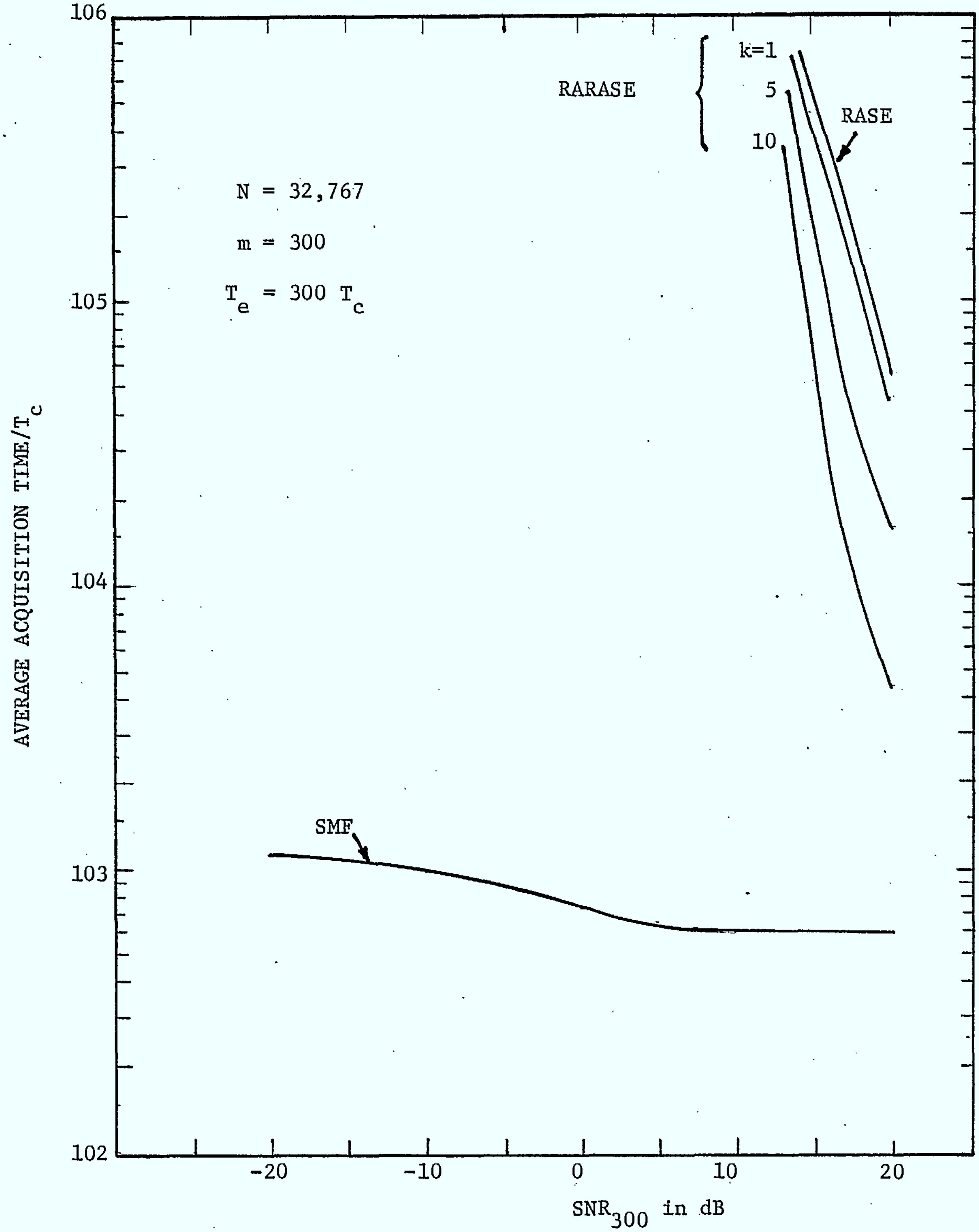

Fig. 3.5 Average Acquisition Time vs Signal-to-Noise Ratio

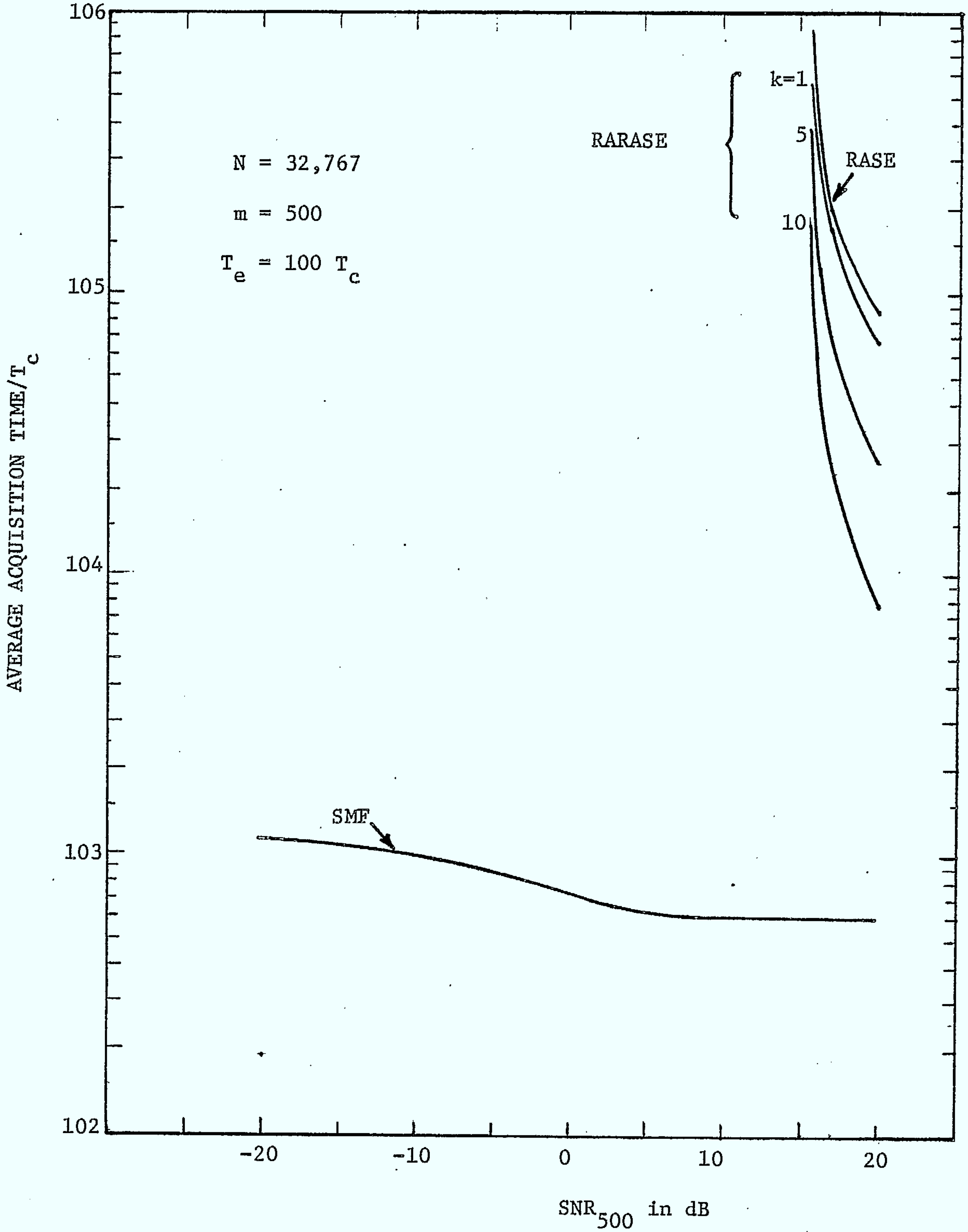

Fig. 3.6 Average Acquisition Time vs SNR_{500}

response occurs at the instant when the SMF impulse response and the incoming signal are exactly matched in frequency and phase. If the SMF's are sampled-data systems, the input signal has to be sampled and clocked into the SMFs. Then sampling the received signal at the correct phase is of critical importance, particularly when the received pulse shape is non-rectangular. In the preceding section we assumed a rectangular pulse $P_{T_c}(t)$ for each chip and that the received pulse shape remains rectangular. In practice the received pulse would be the convolution of $P_{T_c}(t)$ with the impluse response of the communication channel. The received pulse shape g(t) will not be rectangular and of duration greater than T_c. The latter characteristic is due to pulse stretching. Adjacent pulses will overlap If the overlapping is severe, i.e., complete or almost complete overlap, a distortion known as intersymbol interference (ISI) takes place. If the overlap is only partial then the main part of the pulse remains undistorted. It is important that the pulse be sampled at the undistorted part, usually the maximum point. For discrete time matched filtering, knowledge of the initial sampling offset can be critical. In the RASE [1] and RARASE [3] schemes, the received signal is low-pass filtered and hard-limited before sampling. Hard-limiting produces a constant envelope; it also has an inherent weak signal suppression effect. In a multi-user environment in which the combined effect of unwanted signals may dominate, hard-limiting prior to despreading by the SMF is undesirable.

Ignoring for the moment the information data and consider the reception of the pn sequence $\{a_k^{(1)}\}$. The model for the received composite signal can be represented by

$$r(t) = \sqrt{2P}\, \hat{a}_1(t-\tau_1) \cos(\omega_c(t-\tau_1) + \theta_1) + \sqrt{2P} \sum_{i=2}^{K} \hat{a}_i(t-\tau_i) \cdot \cos(\omega_c(t-\tau_i) + \theta_i) + n(t)$$

where

$$\hat{a}_1(t)\ \sum_{k=-\infty}^{\infty} a_k^{(1)}\ g(t-kT),$$

τ_i is the delay offset and g(t) is the received pulse shape. The received signal r(t) is sampled at the input to the SMFs as shown in Fig. 3.7. It is desired to make an accurate determination of τ_1 from operations performed on r(t). Assume that the best sampling instants are at $t = kT + \tau_1$, $k = 0, \pm 1, \pm 2, \ldots$. The objective is to recover $\{a_k^{(1)}\}$ from operations on $r(kT + \hat{\tau}_1)$, where $\hat{\tau}_1$ is an estimate of the parameter τ_1.

Except for the sampler at the input to the SMFs the acquisition model of Fig. 3.7 is similar to that shown in Fig. 3.1. An additional pn sequence generator and correlator combination is used to facilitate estimation of the offset τ_1. The error in the estimate is within one chip time, i.e., $|\tau_1-\hat{\tau}_1| < T_c$. Operationally, the Decision Controller performs the same actions stated previously plus the following: When an SMF indicates synch, the Decision Controller initiates pn sequence generator #1 in the indicated state and phase, i.e., at offset $\hat{\tau}_1$, and the other pn sequence generator in the same state but at offset $\hat{\tau}_1 + \delta$. After one examination interval (an examination interval is approximately m T_c) the outputs of the correlators are compared. If the output of correlator #2 is larger than that of correlator #1, then $\hat{\tau}_1(\text{new}) = \hat{\tau}_1(\text{old}) + \delta$ and pn sequence generator #1 is re-clocked at $\hat{\tau}_1(\text{new})$. At the same time the sampler is also re-clocked to reflect the new estimate $\hat{\tau}(\text{new})$. The procedure is repeatedly executed, always in a direction to improve the estimate $\hat{\tau}_1$ based on comparison of the correlator outputs. This is done on a per examination interval basis to upgrade and to track the estimate $\hat{\tau}_1$. The algorithm adjusts the offset estimate by a fixed amount δ. A

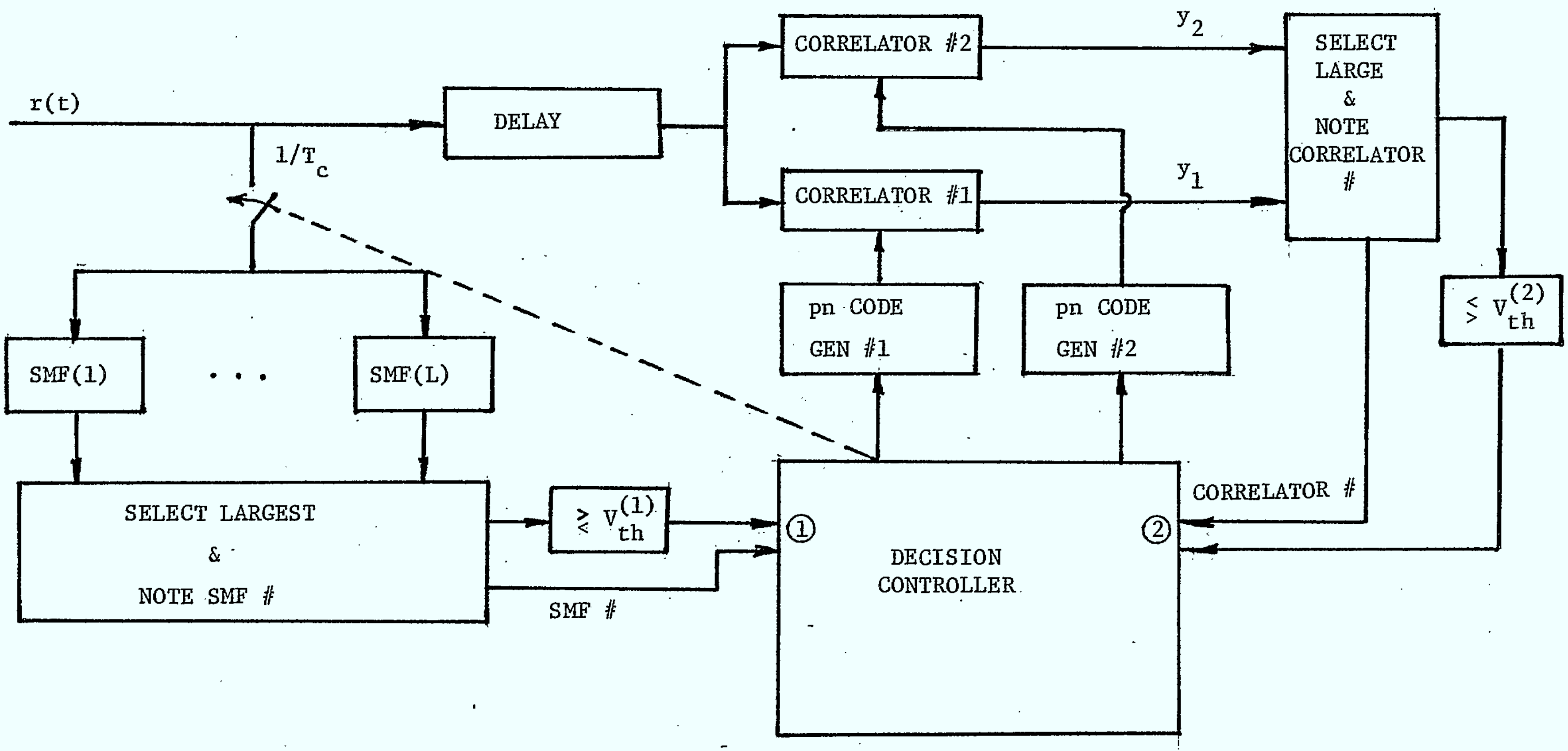

Fig. 3.7 Discrete-Time Subsequence Matched Filtering For Pseudonoise Signal Acquisition (Baseband Model)

flowchart for implementing the algorithm is shown in Fig. 3.8. Since $| \hat{\tau}_1 - \tau_1 | < T_c$, a reasonable choice of δ is $T_c/4$. Pn sequence generator #1 is selected as the main local reference and generator #2 is used to adjust the offset estimate. A "good" estimate is made within 2 examination intervals. In fact the estimation time is zero if the initial estimate is correct, equal to one examination interval T_e if the first δ adjust is in the correct direction, or equal to two examination intervals if the first δ adjust is in the wrong direction. Finer adjustments can be made by executing the algorithm more than once.

For the digital subsequence matched filtering method the output of the ℓth SMF is given by (3.7) modified by a gain factor $g(\hat{\tau}_1 - \tau_1)$. The desired signal component is thus given by

$$x_\ell = \sqrt{P/2}\ m\ T_c\ g(\hat{\tau}_1 - \tau_1)\ d_j^{(1)}$$

where $g(\hat{\tau}_1 - \tau_1) \leq g(o) = 1$. The signal power at the output of the matched filter under synchronism is $g^2(\hat{\tau}_1 - \tau_1)\ Pm^2 T_c^2/2$. The $g^2(\hat{\tau}_1 - \tau_1)$ will also appear in the interference and noise terms, so that the signal-to-noise ratio at the output of the SMF remains the same as in the previous section and the average time to acquisition, $T_{a,SMF}$ is given by Eqn. (3.13) or Eqn. (3.14) plus the estimation time mentioned above.

3.2 Acquisition of Psuedonoise Signals By Suboptimum State Estimation

The communication situation being considered in this section is a single user transmitting over a noisy channel in which a pn code is used for performance enhancement. Such a system is normally used for "ranging" applications. Signal acquisition for such a system has been considered

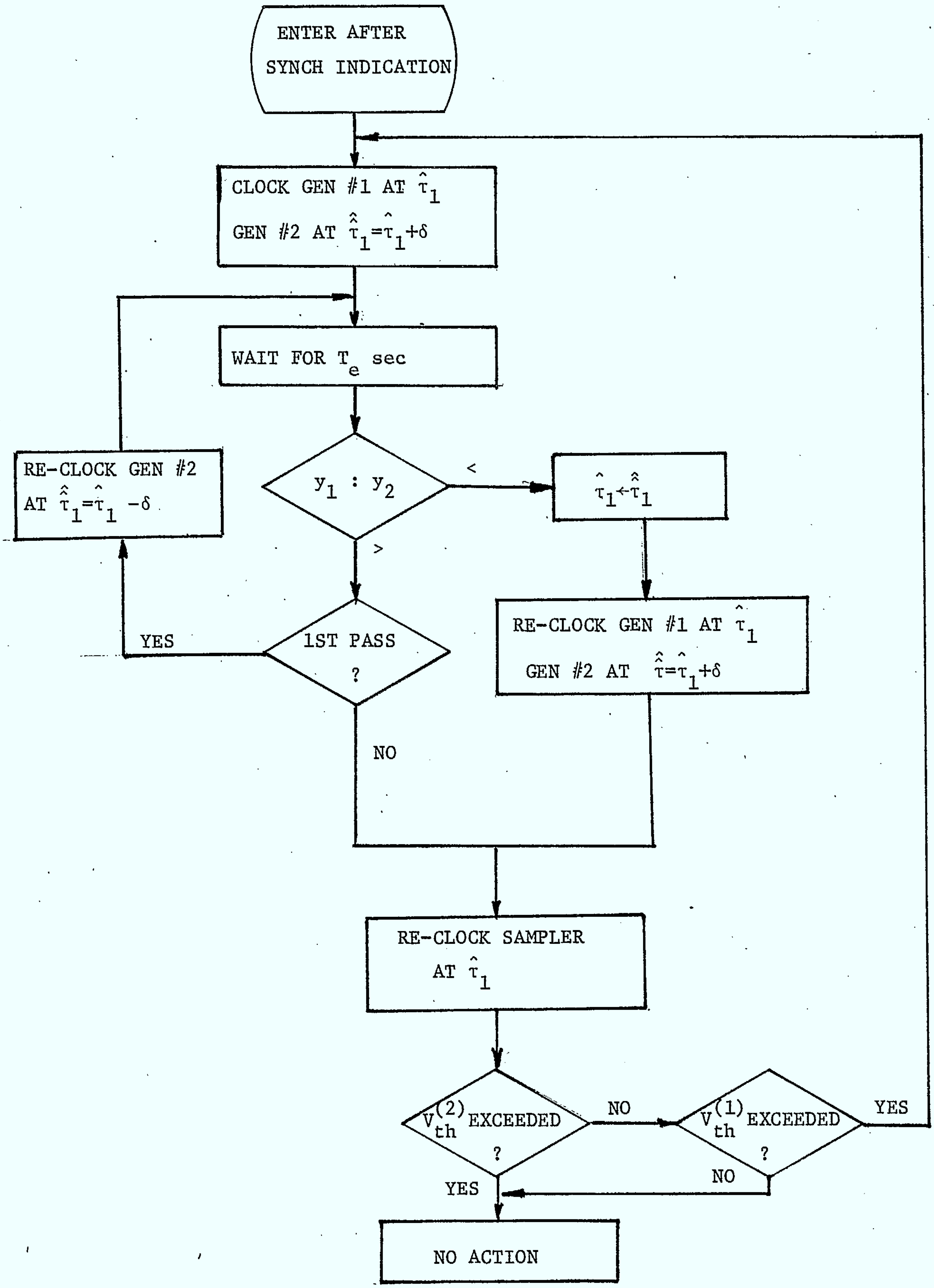

Fig. 3.8 Flowchart for Decision Controller

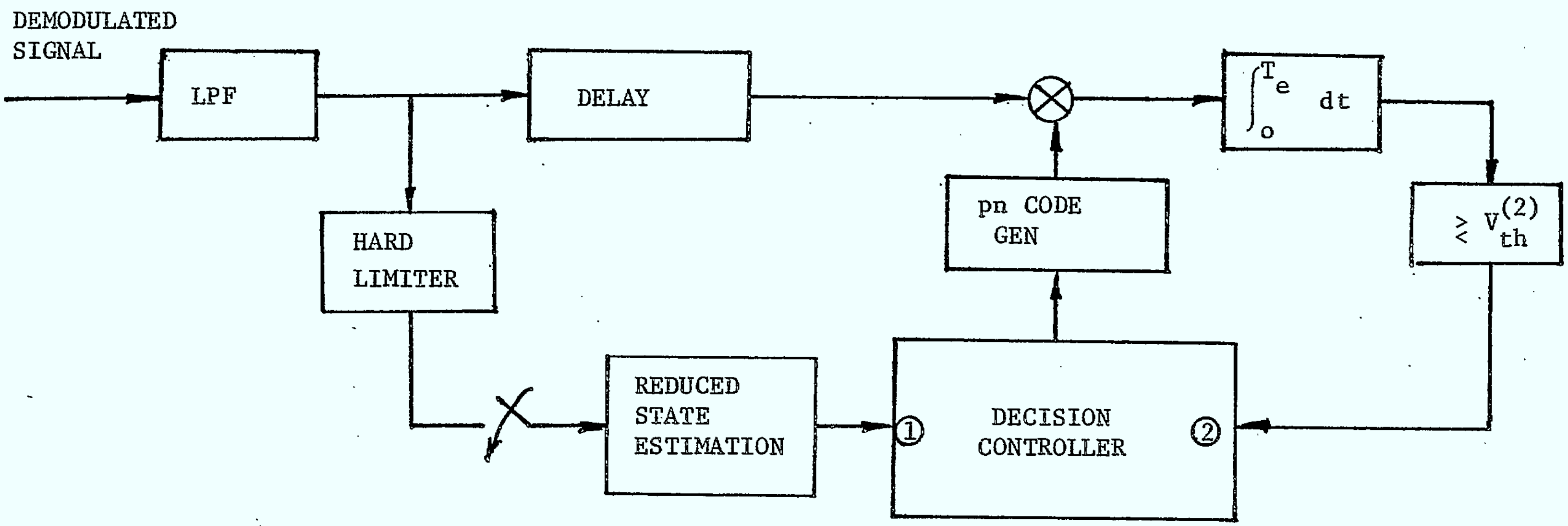

Fig. 3.9 Acquisition of Pseudonoise Signals by Reduced State Estimation

by Ward [1], Kilgus [2], Ward and Yiu [3] and Hopkins [4]. We are interested in the acquisition of a pn sequence using a suboptimum state (reduced state) estimation with feedback control as shown in Fig. 3.9.

The demodulated signal is low-pass filtered, hard-limited and then sampled at a rate $1/T_c$. The sampled sequence is then searched in a reduced trellis to estimate a correct state, which is then loaded into the pn sequence generator. The generated pn sequence is then correlated with the low-pass filtered signal to provide a feedback control signal. If the correlator output indicates synch, the Decision Controller ignores information from the state estimator. If synch is not indicated at the end of an examination interval, a new state estimate is reloaded to restart the pn sequence generation. In this section we are primarily concerned with describing the suboptimum reduced state estimation technique.

Let the state vector of an n-stage shift register generator be $\underline{s} = (s_1, s_2, \ldots, s_n)$. Let $\hat{s}_1 = s_1$ and $\hat{s}_2 = s_n$, $\underline{\hat{s}} = (\hat{s}_1, \hat{s}_2)$ is a state vector of the reduced state space. $\underline{\hat{s}}$ is a superstate which can assume the 2-tuple 00, 01, 10, or 11. That is, any state vector $(0, s_2, \ldots, s_{n-1}, 0)$ will be mapped into the superstate (0, 0), $(0, s_2, \ldots, s_{n-1}, 1)$ into superstate (0,1), $(1, s_2, \ldots, s_{n-1}, 0)$ into superstate (1,0) and $(1, s_2, \ldots, s_{n-1}, 1)$ into superstate (1, 1). The first superstate variable is the pn generator autonomously generated input and the second superstate variable is the pn generator output symbol.

In general each superstate has 4 predecessors and 4 successors. The trellis diagram for the 4-superstate space is shown in Fig. 3.10. The branch values in the trellis diagram of Fig. 3.10 is $x_i(y_i)(L_{\rho_i}(y_i))$, where x_i represents the pn generator input and y_i the possible generator output

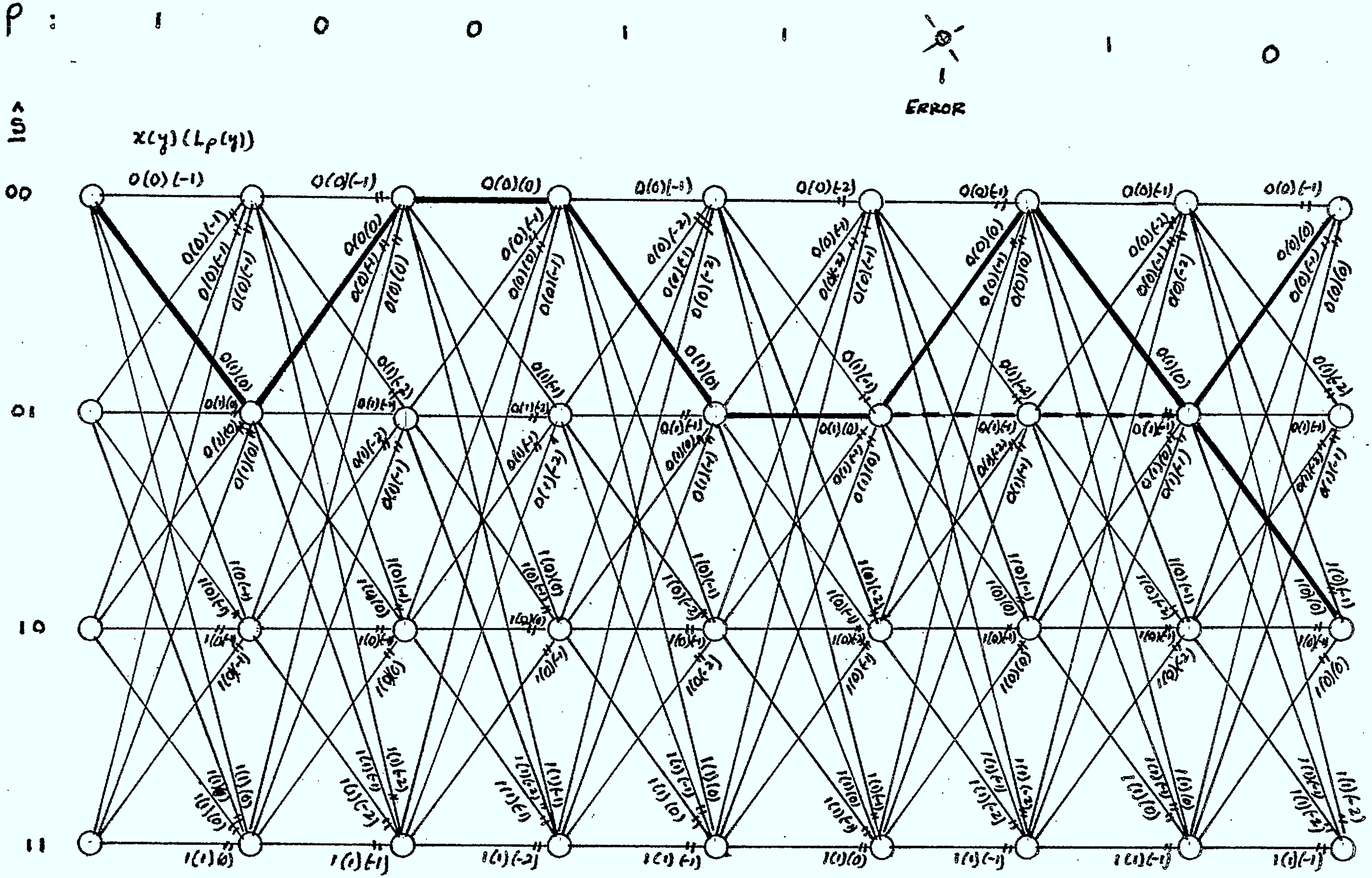

Fig. 3.10 Superstate Trellis Search (Heavy Line Indicates Correct Path, Dashed Line Indicates Decision Due to Single Error).

at the ith iteration or depth in the trellis. Thus the 2-tuple (x_i, y_i) represents the next state in the reduced trellis. $L_{\rho_i}(y_i)$ represents the cumulative path metric between the received bit ρ_i and the possible generator output y_i at depth i, where

$$L_{\rho_I}(y_I) = -\sum_{i=1}^{I} | \rho_i - y_i |, \; I = 1, 2, \ldots$$

The quantity $-| \rho_i - y_i |$ is the branch metric at the ith depth. As illustrated in Fig. 3.10, $L_{\rho_I}(y_I)$ can take on one of the values 0, -1, -2 only. At each depth the trellis is pruned. There are 4 inputs to each superstate. The pruning process cuts off 3 of the 4 branch inputs to each state which have smaller cumulative path metrics. In the case of a tie, the lower branch (or branches) is cut arbitrarily. At any depth only 4 paths survived. One of these 4 paths will exhibit the maximum cumulative path metric. The above reduced trellis search procedure has been described in [5] and is refered to as a maximum likelihood search. The cumulative path metric $L_{\rho_I}(y_I)$ is equivalent to the likelihood function. n consecutive branch values y_i of the surviving path (the heavy path in Fig. 3.10) in the reduced trellis constitute a state vector.

Any n consecutive bits of a pn sequence represent a state of the pn generator. A one bit error can thus affect n consecutive states. Our study in reduced trellis search indicates that a one bit error causes an error in one superstate (dotted heavy line in Fig. 3.10). In fact the erroneous state is not a logical successor to the preceding correct state. Based on this fact it is proposed to use a majority decision strategy in the reduced state space search to provide an estimate of the state vector for loading into the pn generator shown in Fig. 3.9. By a majority decision strategy it is meant that after searching I levels into the reduced state trellis, a majority of consecutive superstates satisfying the pn generator state transitions is taken to give rise to a "good" state estimate. That is, the branch values y_i of n consecutive logical state transitions in which the majority are correct is taken as the good estimate for loading into the local pn sequence generator.

Define the signal-to-noise ratio by $SNR_{T_c} = \frac{2\,PT_c}{N_o}$, where $\sqrt{2P}$ is the amplitude of the transmitted signal, as in the preceding section, $N_o/2$ is the two-sided noise power spectral density and T_c is the chip time. The channel can be viewed as a binary symmetric channel (BSC) with cross-over probability $\varepsilon = Q(SNR_{T_c}^{1/2})$. The probability of correctly determining a state from the reduced state trellis search is $p = 1-\varepsilon = 1-Q(SNR_{T_c}^{1/2})$ (p is the same as Ward's correctness probability [1]). The average time to acquisition, $T_{a,RTS}$, where RTS stands for reduced trellis search, is given by

$$T_{a,RTS} = \frac{I\ T_c + T_e}{p} ,$$

where I is the search interval before making a premature decision and T_e is the examination interval in the feedback path. In the decoding of convolutional codes using a full trellis search, a premature decision can be made after 5 constraint lengths. On this basis, we can let I = 5n.

On the average p/ε consecutive superstates are correct. The maximum number of correct consecutive superstates is $B_{Max} \geq p/\varepsilon$. Over the 5n depth of the reduced trellis, it is reasonable to expect that at least one run of correct consecutive superstates = B_{Max}. Comparing with the RASE method [1] and RARASE method [3], the improvement factor is given by

$$\frac{T_{a,RTS}}{T_{a,RASE}} = \frac{(I\ T_c+T_e)/p}{T_e/p^n}$$

$$= (1 + \frac{I\ T_c}{T_e})\ p^{n-1} \tag{3.15}$$

$$\frac{T_{a,RTS}}{T_{a,RARASE}} = \frac{(I\ T_c+T_e)/p}{T_e(p^2+3(1-p)^2)^k/p^n}$$

$$= (1 + \frac{I\ T_c}{T_e})\ p^{n-1}/(p^2+3(1-p)^2)^k \tag{3.16}$$

where k is the number of 3-input mod 2 adders used for decision making in the RARASE method [3].

It is noted that for a given ε the worst case for our method is when the εI erroneous superstates are uniformly distributed over the I superstates searched (this corresponds to εI channel error bits uniformly distributed over I received bits). A run of maximum number of correct consecutive superstates would be longer if some of the error bits are bunched together.

Consider for example, $\varepsilon = 0.15866$ (corresponding to a $SNR_{T_c} = 0$ d B), the average number of correct consecutive superstates is $p/\varepsilon = 5.6$. Over a search depth $I = 5n$, any run of consistent consecutive superstates exceeding 7 can be used to generate a state estimate, i.e., the branch values of the maximum run and its extensions to n consecutive superstates are taken to comprise the state estimate.

It is only necessary to search a depth $I = 5N/2$ for $p/\varepsilon \leq n/2$. For $p/\varepsilon > \frac{n}{2}$, the amount of search can be reduced since the average run of correct consecutive superstates exceeds half the number of stages in the pn generator and there is every likelihood that the maximum run would be longer. It is conjectured that a search depth of $I = 3\ p/\varepsilon$ may be sufficient.

To obtain a state estimate by majority decision and logical state extension it is necessary to execute a second pass search of the reduced state trellis. At this time we only conjecture that the reduced state space search with a majority decision strategy appears to be a viable acquisition scheme for pn signals. Further investigations, both analytically to prove that a single error only affects a single superstate in the reduced trellis search and by computer simulation, will be pursued.

REFERENCES

[1] R.B. Ward, "Acquisition of Pseudonoise Signals by Sequential Estimation", IEEE Trans. on Commun Technol., Vol. COM-13, No. 4, pp. 475-483, Dec. 1965.

[2] C.C. Kilgus, "Pseudonoise Code Acquisition Using Majority Logic Decoding", IEEE Trans. on Commun. Vol. COM-21, No. 6, pp. 772-3, June 1973.

[3] R.B. Ward and K. Yiu, "Acquisition of Pseudonoise Signals by Recursion-Aided Sequential Estimation", IEEE Trans. on Commun. Vol. COM-25, No. 8, pp. 784-794, Aug. 1977.

[4] P.M. Hopkins, "Pseudonoise Synchronization at Low Signal-to-Noise Ratio", National Telecommun. Conf. Paper 32.2.1, November 1976, Dallas, Texas.

[5] J.W. Mark and I.F. Blake, "Rapid Acquisition Techniques and the Sequence Design Problem for CDMA Spread Spectrum Communication Systems", Final Report Project No. 808-02, Dss Contract No. OSU 79-00082, University of Waterloo, March 1980.

[6] M.B. Pursley, "Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication - Part I: System Analysis", IEEE Trans. on Commun., Vol. COM-25, No. 8, pp. 795-799, Aug. 1977.

4. RECENT APPROACHES TO SPREAD SPECTRUM SYSTEMS

The subject of spread spectrum systems continues to receive considerable attention in the literature. It is perhaps characterized by a diversity of approaches and system configurations that are often only partially motivated, making a full comprehension difficult. This section of the report attempts to review certain aspects of this work with a view to future directions.

4.1. Review of the Performance of Coded Spread Spectrum Systems.

Despite the accepted importance of coded systems for spread spectrum applications, there have been relatively few contributions in the open literature in this area. Perhaps the most significant of these are the papers by Viterbi [1] and Viterbi and Jacobs [2]. Some of these results which may prove useful for future work are briefly reviewed here, along with some comments on a problem raised by Campbell [3].

Consider first the error bounds ([1], equations (4) and (5)) for bit error probabilities:

$$P_b < 2^{-K(\alpha-1)}\ , \quad \text{block code}$$

$$P_b < \frac{2^{-K\alpha}}{(1-2^{1-\alpha})^2}\ , \quad \text{convolutional code} \tag{4.1.1}$$

where $\alpha = r_o/r$, r the code rate and r_o the so called computational cut off rate. The parameter K is the number of information bits per block in the block code case and the constraint length of the convolutional code. The parameter r_o is perhaps more appropriately referred to as an error exponent break point for block and convolutional codes rather than the computational cut off point, which refers to the point for which the average number of computations for sequential decoding becomes unbounded.

For the binary input, additive white Gaussian noise channel this break point in the error exponent curve for block codes is at the point ([4], p.142)

$$E_o(1) = \max_{\underline{q}} \left\{ -\ln \sum_y \left\{ \sum_x q(x)\sqrt{p(y/x)} \right\}^2 \right\}$$

$$= 1 - \log_2(1 + Z)$$

where

$$Z = \sum_y (p(y/0)p(y/1))^{1/2} = \exp(-E_s/N_o)$$

and E_s is the energy per dimension. For rates $R < E_o(1)$ the error exponent is given by

$$P_b < 2^{-N(E_o(1)-R)} = 2^{-NR(\frac{E_o(1)}{R} - 1)}$$

$$= 2^{-K(\frac{R_o}{R} - 1)} = 2^{-K(\alpha-1)}$$

in the notation of Viterbi [1]. Thus the error exponent decreases linearly for rates R between $(0,R_o)$. Between R_o and capacity C the error exponent curve is more complicated. The situation is similar for convolutional codes ([5]) except that the curve is constant for rates between 0 and R_o, a fact which leads to the superiority of convolutional codes over block codes. For rates above R_o the curve is again more complicated. Thus the simple error bounds of (4.1.1) are valid only for rates between 0 and R_o.

A question of interest posed by Campbell [3] was to determine the tradeoff between pure direct sequence spread spectrum and a coded version of it. For the direct sequence system using coherent binary phase shift keying (BPSK) the probability of bit error is given by

$$P_b < Q\left(\frac{2E_b}{N_o}\right)$$

where Q(·) is the usual complementary error function (which here will be approximated by [6]

$$Q(x) \sim \frac{1}{\sqrt{2\pi(1+x^2)}} \; e^{-x^2/2} \quad).$$

In the pure direct sequence system the data is BPSK modulated and multiplied by the spectrum spreading direct sequence and E_b denotes the energy per data bit. For the coded/direct sequence version the data is first convolutionally encoded with a code of rate r so that $E_s = E_b r$ where E_s is the energy per encoded bit. The result is then multiplied with the same direct sequence as the first sequence.

The question arises as to what rate code should be used. It is assumed that the problem of the (pn) direct sequence tracking loop is the same for both systems and that the signal to noise ratio at the input to the tracking loop is not a limiting factor i.e. a sufficiently sophisticated despreading technique is available to operate at the expected SNR. The decoder however is assumed to require an input SNR of at least 1.5 db. ([3]).

To compare the effect of convolutional coding on system performance, assume that a convolutional code of constraint length 8 is used to allow for the possibility of Viterbi decoding and that the SNR at the decoder input is 1.5 db. For a given P_b the value of α can be determined (graphically, in this case) from equation (4.1.1). The value of E_s/N_o determines r_o and from these two values the code rate r can be determined and hence $E_b/N_o = E_s/rN_o$. The performance of the direct sequence system using this SNR can then be calculated and compared with the coded system. The results of this process are shown in Table 1 and it is concluded that, for the range of parameters used and under the assumptions stated, the performance of the coded system is superior. It would appear that as long as

P_b(conv. code)	10^{-4}	$5\text{x}10^{-5}$	$2\text{x}10^{-5}$	10^{-5}	$5\text{x}10^{-6}$	$2\text{x}10^{-6}$	10^{-6}
code rate	.356	.338	.317	.303	.289	.272	.261
P_b (uncoded)	$2.523\text{x}10^{-3}$	$1.985\text{x}10^{-3}$	$1.460\text{x}10^{-3}$	$1.164\text{x}10^{-3}$	$9.081\text{x}10^{-4}$	$6.482\text{x}10^{-4}$	$5.124\text{x}10^{-4}$

Table 1.

the SNR condition at the decoder input is met and as long as the SNR at the input to the despreader is not a factor, the coded system will offer superior performance.

It is shown in [1] that the effect on the direct sequence BPSK system of a pulse or non-uniform jammer is to reduce the probability of bit error from

$$P_b < Q\left(\sqrt{\frac{2E_b}{N_o}}\right) \sim e^{-E_b/N_o}$$

to

$$P_b < e^{-1}/(E_b/N_o)$$

for the optimal jammer who jams the fraction

$$\rho = 1/(E_b/N_o)$$

of the band or jams 100ρ % of the time with a noise density of N_o/ρ. It is also shown that the drop in performance from exponential dependence to inverse linear dependence can be recovered using coding. This section is concluded by elaborating on these results with some material from [2].

The aim of the paper is to treat three types of channel impairments; Rayleigh fading, partial band noise and unregulated multiple access interference. The Rayleigh fading is characterized by a received amplitude (in the binary signalling case) of aE_b where a is a random variable with probability density function

$$p(a) = 2a\, e^{-a^2}, \quad a > 0$$

normalized so that $E(a^2) = 1$. For partial band noise it is assumed that a fraction ρ of the available band is jammed with a spectral density N_o/ρ, to equate the total noise power with the unjammed case. In a frequency hopping system this partial band jamming has an equivalent effect of partial time jamming. The multiple access interference case actually follows, for frequency hopped systems, from results for the partial band jamming. The highlights of the interesting contribution of Viterbi and Jacobs [2] with the relevant equations having a number referring to the equation number in [2].

Recall first [7] that for binary antipodal signals with coherent detection, the probability of error is given by

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_o}}\right) .$$

For orthogonal signals with coherent detection it is

$$P_e = Q\left(\sqrt{\frac{E_b}{N_o}}\right)$$

while for noncoherent (uniform phase distribution of received carrier)

$$P_e = \frac{1}{2} \exp\,(-E_b/2N_o) .$$

Now consider the noncoherent case employing 2 orthogonal signals on a channel subject to Rayleigh fading. The probability of error, after averaging over the Rayleigh density, is

$$P_e = \frac{1}{2 + (E_b/N_o)} \tag{3}$$

and the catastrophic result that the probability of error is now inversely proportional to the SNR rather than exponentially, is noted. For partial band interference (noncoherent, 2 orthogonal signals) with a fraction ρ of the band jammed, the probability of error is

$$P_e(\rho) = (\rho/2)\ \exp(-\rho E_b/2N_o) \tag{4}$$

and maximizing this expression with respect to ρ, corresponding to the worst case partial band interference gives

$$P_e \geq e^{-1}/(E_b/N_o) \quad , \qquad \rho = 2N_o/E_b \leq 1 \tag{5}$$

with equality when $E_b/N_o \geq 2$. Again the inverse linear relationship results. For example, to achieve an error probability of 10^{-5} using antipodal signals and coherent detection requires E_b/N_o of 9.6 db while for noncoherent orthogonal signals 13.4 db is required. For the case of orthogonal signals, noncoherent detection on the fading channel, an E_b/N_o of 50 db is required while for the worst case partial band channel it is 45.7 db.

To improve upon these results some form of diversity transmission is required, where, say, L independent observations per bit are combined in some manner to yield a decision statistic yielding an improved performance. A common form of diversity transmission is that employing L centre frequencies spread uniformly across the available band. Each data bit interval is divided into L chip intervals and typically, the carrier is hopped to each frequency during one data bit. Such a scheme can be used to overcome frequency selective fading since the estimates during each chip interval can be assumed independent, contributing to an overall statistic. If the fading is not frequency selective then some form of time diversity can be employed, such as interleaving.

It is shown that for noncoherent reception of 2 orthogonal signals using L chip diversity, the probability of error can be upper bounded for:

$$\text{Rayleigh fading:} \quad P_e < \frac{1}{2}\,(p(1-p))^L \,, \qquad p = 1/(2+E_b/N_oL) \tag{8}$$

$$\text{worst case partial band:} \quad P_e < \frac{1}{2}\left[\frac{4e^{-1}L}{E_b/N_o}\right]^L, \text{ provided } E_b/N_oL \geq 3 \tag{10}$$

Minimizing these expressions over L yields:

$$\text{Rayleigh fading:} \quad P_e < \frac{1}{2}\exp(-0.149(E_b/N_o)) \,, \quad E_b/N_oL = 3 \tag{12}$$

$$\text{worst case partial band:} \quad P_e = \min_{L} \max_{0\leq\rho\leq1} P_e(\rho) < \frac{1}{2}\exp(-(E_b/4N_o)), \; E_b/N_oL=4 \tag{13}$$

To compare these results for the 2 orthogonal signal case, noncoherent reception with L chip diversity to achieve a probability of error of 10^{-5}, the SNR required for:

Rayleigh fading: 18.6 db , L = 24

worst case partial band: 16.4 db, L = 11

It is noted in [2] that these results can immediately be applied to the multiple access channel by assuming that when the centre frequency of two users fall on the same frequency during a given chip interval, one will appear as Gaussian noise to the other. If each user has rate R and the system bandwidth is W then the SNR of the jth user is

$$E_{b_j}/N_o = \left(\frac{W}{R}\right) \frac{E_{b_j}}{\sum\limits_{n \neq j} E_{b_m}}$$

From (13) it can be shown that to achieve $P_e = 10^{-5}$ a ratio W/R of approximately 40(N-1) is required where N is the total number of users on the system.

Coding can be used effectively to improve the SNR's derived so far and the effect of block orthogonal coding is first considered. Assume that one of $M=2^k$ signals can be transmitted during a given interval, giving a signal energy of kE_b and L chip/bit diversity is used. A straight forward application of the union bound then yields a probability of bit error

$$P < 2^{k-2} \exp(-\mu k E_b/N_o) \tag{16}$$

for optimum diversity where:

$$\text{Rayleigh fading:} \quad \mu = 0.149 \qquad E_b/N_o L = 3 \tag{17}$$

$$\text{worst case partial band:} \quad \mu = 1/4 \qquad E_b/N_o L = 4.$$

For k=3, $P_b < 10^{-5}$, an E_b/N_o of 14.4 db is required with L = 9 for fading and 12.1 db with L = 4 for worst case partial band. These results could be improved by using different block codes at the expense of increased receiver complexity and decreased signal distance.

Because of the ability of Viterbi decoders, as well as other convolutional code decoders, to use soft information in addition to a superior random coding exponent, superior performance can be expected from such codes. Three distinct classes of convolutional codes are considered here; dual k, orthogonal and semi orthogonal. The structure of these codes is considered in [2] but will be omitted here. For dual k codes, for example, the probability of bit error, using L chip diversity can be upper bounded as

$$P_b < \frac{2^{k-2}\ \exp(-2\mu k E_b/N_o)}{(1-\exp(-\mu E_b/N_o) - (2^k-2)\exp(-\mu(k-1)E_b/N_o))^2} \tag{19}$$

where

$$\begin{array}{lll} \text{Rayleigh fading:} & \mu = 0.149\ , & E_b/N_oL = 3 \\ \text{worst case partial band:} & \mu = 1/4 & E_b/N_oL = 4 \end{array} \tag{20}$$

The other classes of convolutional codes appear to yield even better performance.

The performance results of the various cases considered in [2] are given in Table 2, from which it can be seen that coding both reduces the SNR requirements and the amount of diversity required to achieve a given performance level. Applied to the multiple access channel these results can be shown to greatly reduce the number of users, each user enjoying a given level of performance.

The results and techniques developed in this paper ([2]) should prove applicable and very useful to analyzing the performance of any given spread spectrum system under a variety of assumptions on the type of coding (block, convolutional or a concatenated version of both). It is hoped to build on these results to achieve this purpose.

E_b/N_o, db	Channel	Comments
9.6	coherent	binary antipodal signals
13.4	noncoherent	binary orthogonal signals
50	fading	2 orthogonal signals
45.7	wcpb*	2 orthogonal signals
18.6	fading	2 orthogonal signals, $L = E_b/3N_o = 24$
16.4	wcpb	2 orthogonal signals, $L = E_b/4N_o = 11$
14.4	fading	2^3 orthogonal block code, $L = 9$
12.1	wcpb	2^3 orthogonal block code, $L = 4$
9.3	wcpb	dual 3 code, $L = 2$
8.3	wcpb	semiorthogonal, $K = 7$, $k = 3$, optimum diversity

Table 2. Required E_b/N_o for $P_e < 10^{-5}$.

4.2. The Multiple Access Problem.

The error analysis of direct sequence code division multiple access (CDMA) spread spectrum communication systems has been investigated in two recent papers ([8], [9]). These contributions are reviewed here with a view to establishing their significance to the literature of this area.

Let the spreading sequence of the ith user be denoted by the function

$$a_i(t) = \sum_{j=1}^{n} a_{ij}\ \phi_j(t) \quad , \qquad a_{ij} \text{ real}$$

where $\phi_j(t)$, $j=1,\ldots,n$ is a sequence of n orthonormal functions on $(0,T)$ and we assume that

$$E = \int_0^T a_i^2(t)\ dt = \sum_{j=1}^{n} a_{ij}^{\ 2} \ .$$

This sequence is modulated with a data stream $b_i(t)$, which is either +1 or -1 on $(0,T)$. Now suppose there are K users transmitting to distinct receivers, each with a random delay i.e. operation is asynchronous. For receiver 1 the received signal is despread by multiplying it by $a_1(t)$ which is assumed to be synchronized with the spreading sequence $a_1(t)$ contained in the signal. Thus the output of the matched filter of receiver 1 is [8]

$$y = \int_0^T a_1^2(t)\ b_1(t)\ dt \ + \ \sum_{i=2}^{K} \int_0^T a_i(t-\tau_i)b_i(t-\tau_i)a_i(t)dt$$

$$+ \int_0^T n(t)a_1(t)dt \qquad (4.2.1)$$

where $\tau_2, \tau_3, \ldots, \tau_K$ are independent, uniformly distributed random variables. The model of Yao [8] included carriers with random phases, which we delete here. Equation (4.2.1) can be expressed as

$$y = Eb + Z + n$$

where b is the data bit, n is a Gaussian random variable with mean 0 and variance σ^2 and Z is a random variable which is a complicated function of the K-1 random delays. The probability of error, assuming that b = +1, is P(Eb + Z + n < 0) and the overall probability of error is [8]

$$P_e = \frac{1}{2} P\,(E + Z + n < 0) + \frac{1}{2} P\,(-E + Z + n > 0)$$

$$= E\,(Q(\frac{h+Z}{\sigma}))$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-y^2/2)\,dy\ .$$

and the expectation is over all the delays $\tau_2,\ldots,\tau_K$.

Yao [8] derives upper and lower bounds to this probability of error using moment space methods. These are found to be quite tight for most situations of interest. An additional conclusion of the work is that the random variable Z is well approximated by a Gaussian random variable except for small K and small n. This is seen as justifying a useful and common assumption in spread spectrum work.

Mazo [9] was also concerned with the probability of error in CDMA/SS systems, but only in the synchronous case. Define the cross correlations of the spread functions as

$$E\rho_{ij} = \int_0^T a_i(t)a_j(t)\,dt = \sum_{k=1}^{n} a_{ik}a_{jk} = (\underline{a}_i, \underline{a}_j)$$

If $b_1(t) = 1$, the probability of error ignoring noise is

$$P_e(K) = P\left(1 + \sum_{i=2}^{K+1} b_i \rho_{1i} < 0\right)$$

where b_i is the data bit intended for user i and $b_1 = +1$ and there are K other users on the channel. It is first shown that if the a_{ik} are chosen independently and with an identical distribution to be ± 1, then an upper (random coding) bound on $P_e(K)$ is

$$P_e(K) \leq e^{-n/2K} \quad . \tag{4.2.2}$$

An alternative point of view is to consider the other user interference to be uniformly distributed over the channel bandwidth W. If each user contributes a power P the one sided power spectral density is then

$$N_o = \frac{KP}{W}$$

The problem now is to detect antipodal signals in Gaussian noise for which the result is

$$P_e(K) = Q\left(\sqrt{\frac{2E}{N_o}}\right) < e^{-E/N_o} = e^{-n/2K}$$

as before. The question arises as to what interpretation can be placed on these bounds. One interpretation is that there exists a code for (K+1) users so that each user has an error rate no larger than exp(-n/2K). The Gaussian approximation placed no restriction on the total number of possible users and, in particular, is unrelated to the maximum cross correlation. The interpretation here is that $P_e(K)$ is an average error rate, averaged over all sets of K+1 users and, presumably, some sets of (K+1) users will have very bad performance.

The question of interest [9] is the level of performance that could be guaranteed for (K+1) users chosen from a total set of M users. The parameter M now plays a more visible role in the proceedings and it is shown that

$$P_e(K) \le e^{-1/2K\rho_{max}^2} \tag{4.2.3}$$

where ρ_{max} is the maximum cross correlation between any two user sequences. The relationship between M, n and ρ_{max} has received some attention in the literature but the relationship is not well established. One result, due to Welch [10] is that

$$\rho^2{max} \ge \frac{1}{M-1}\left[\frac{M}{n} - 1\right]$$

and for M large compared to n, $\rho^2_{max} \ge 1/n$, for which value (4.2.3) reduces to the random coding bound. A variation on the problem of determining a lower bound on ρ_{max} for a given M and n is to consider the more restricted case where ρ_{max} is determined by the absolute value of the correlations is used rather than the correlations themselves. Some recent results on this problem [11] are considered, but many questions remain open.

4.3. New Approaches to Spread Sprectum Systems.

Spread spectrum systems offer considerable scope for the imaginative use of time and bandwidth to either combat intentional interference (jamming) or other user interference where the entire bandwidth is shared among all users on a non-assigned basis (CDMA). A recent system apparently proposed by Viterbi [12] (reference not yet available to the authors) is a good example of this and a brief description of this system and its performance is given here as described by Einarsson [13].

Let Q be either a prime or power of a prime (so that the finite field GF(Q) exists) and L be an integer, less than or equal to Q-1. During each interval of length T seconds a symbol from GF(Q) is transmitted (thus $\log_2 Q$ bits) by the following scheme. The interval is divided into L chips, each of length T/L seconds and during each chip one of Q frequencies is transmitted, each frequency corresponding to an element of GF(Q). Each user is assigned a codeword of frequencies of length L over GF(Q)

$$\underline{a}_m = (a_{m1}, a_{m2}, \ldots, a_{mL}) , \quad a_{mi} \varepsilon GF(Q) .$$

If during a particular data interval of length T it is desired to transmit the symbol x_m the following sequence of frequencies is transmitted:

$$\underline{y}_m = \underline{a}_m + x_m \cdot \underline{1} = (a_{m1}+x_m, a_{m2}+x_m, \ldots, a_{mL}+x_m) \quad a_{mi}, x_m \varepsilon GF(Q)$$

where $\underline{1}$ is the all ones vector and the additions are in GF(Q). To obtain x_m at the receiver the users address is subtracted from the sequence of received frequencies, leaving the vector $x_m \cdot \underline{1}$ in the absence of interference. Viewing each codeword of frequencies in a Q x L frequency-time array, after decoding in the absence of interference, a row of frequencies x_m appears and the received vector is decoded to the symbol x_m.

For such a scheme to work each user must be assigned a codeword so there is minimum interference, regardless of what data is being transmitted. Assume first that all users are synchronized at the data interval level and that user m is assigned the address

$$\underline{a}_m = (\gamma_m, \gamma_m\beta, \gamma_m\beta^2, \ldots, \gamma_m\beta^{L-1}) \ , \quad \gamma_m \varepsilon GF(Q)$$

where β is a fixed primitive element of GF(Q). Now viewing two user sequences

$$\underline{y}_1 = \underline{a}_1 + x_1 \cdot \underline{1} \qquad \underline{y}_2 = \underline{a}_2 + x_2 \cdot \underline{1}$$

it is easily shown that the same frequency can result in at most one chip interval. Thus after user 1 decodes there will be exactly one row in the array. If there are $M \leq L$ users, correct decoding will take place since the frequencies of the other M-1 users could, in the worst case combine to produce a row with at most M-1 entries. Clearly, at most Q users can be accommodated in this scheme.

The above situation can be modified to the nonsynchronous case by identifying user m with an element $\gamma_m \varepsilon GF(Q)$ and, for data x_m, transmit the frequency sequence

$$\underline{y} = x_m \underline{\beta} + \gamma_m \cdot \underline{1} \ , \qquad \underline{\beta} = (1, \beta, \beta^2, \ldots, \beta^{L-1})$$

Again it can be shown that the user 1 sequence $\underline{y}_1$ and any shift of user sequence $\underline{y}_2$ can interfere (have the same frequency) in at most one chip.

The error probability analysis of these schemes, in the presence of interference from (M-1) other users only can be easily bounded. In the synchronous case the following two bounds on the word error probability are given [13]

$$P_e \leq (Q-1) \prod_{i=1}^{L} (M-i)/Q^L$$

and

$$P_e \leq (Q-1) \prod_{i=0}^{L-1} \left[1 - (1 - \frac{1}{Q-i})^{M-i-1}\right]$$

For the nonsynchronous case it is shown that

$$P_e \leq (Q-1) \left\{ \prod_{i=0}^{\frac{L}{2}-1} \left[1 - (1 - \frac{1}{Q-i})^{M-i-1}\right] \right\}^2$$

The above frequency hopping scheme takes a novel approach to the use of the frequency hops for transmitting data, although the elements of a similar scheme were suggested in Sarwate and Pursley [14]. The possibilities for innovation with such schemes seem considerable. For example perhaps there would be advantages to considering a combined time-frequency-correlation approach, rather than just the time-frequency approach discussed above. Specifically, let $C_1(t), C_2(t), \ldots, C_N(t)$ be N binary (± 1) sequences of time of duration one chip time T/L (using the same notation as the previous discussion) with the property that

$$\int_0^{T^1} C_i(t) C_j(t) dt = \rho_{ij}$$

and ρ_{ij} takes on values in a restricted set $S = \{S_1, S_2, \ldots, S_K\}$. During each chip interval the signal transmitted is of the form $C_i(t)\mathrm{Cos}(w_j t)$ i.e. each tone is modulated by an additional spreading function $C_i(t)$ leading to a form of combined direct sequence/frequency hopping system. At the receiver, during each chip interval, the frequency is first determined and the tone removed. The correlation between the direct sequence $C_i(t)$ and a given fixed direct sequence is then determined, giving a value in S. Thus during each chip interval both a correlation and a frequency is determined. This would seem to allow various possibilities for coding to assist with synchronization and reduce the effects of other user interference. It would appear that this scheme, or similar ones, are worthy of further consideration.

References

[1] A.J. Viterbi, "Spread Spectrum Communications - Myths and Realities," IEEE Communications Magazine, May, 1979, 11-18.

[2] A.J. Viterbi and I.M. Jacobs, "Advances in Coding and Modulation for Noncoherent Channels affected by Fading, Partial Band, and Multiple-Access Interference," in Advances in Communications Systems, vol. 4, 1975, Academic Press, New York.

[3] R. Campbell, personal communication.

[4] A.J. Viterbi and J.K. Omura, Principles of Digital Communication and Coding, 1979, McGraw-Hill, New York.

[5] A.J. Viterbi, "Convolutional Codes and their Performance in Communication Systems," IEEE Transactions on Communications Technology, vol. COM-19, 1971, 751-771.

[6] P.O. Börjesson and C.W. Sundberg, "Simple Approximations of the Error Function Q(x) for Communications Applications," IEEE Transactions on Communications, vol. COM-27, 1979, 639-643.

[7] A.J. Viterbi, Principles of Coherent Communication, 1966, McGraw-Hill, New York.

[8] K. Yao, "Error Probability of Asynchronous Spread Spectrum Multiple Access Communications Systems," IEEE Transactions on Communications, vol, COM-25, 1977, 803-809.

[9] J.E. Mazo, "Some Theoretical Observations on Spread Spectrum Communications," Bell System Technical Journal, vol. 58, 1979, 2013-2023.

[10] L. Welch, "Lower Bounds on the Maximum Cross Correlation of Signals," IEEE Transactions on Information Theory, vol. IT-20, 1974, 397-399.

[11] G.A. Kabatyanskii and V.I. Levenshtein, "Bounds for Packings on a Sphere and in Space," Problems of Information Transmission, vol. 14, 1978, 1-17.

[12] A.J. Viterbi, "A Processing Satellite Transponder for Multiple Access by Low-Rate Mobile Users," Proc. Digital Satellite Communications Conference, Montreal, 1978.

[13] G. Einarsson, "Address Assignment for a Time-Frequency-Coded, Spread-Spectrum System," Bell System Technical Journal, vol. 59, 1980, 1241-1255.

[14] D.V. Sarwate and M.B. Pursley, "Hopping Patterns for Frequency-Hopped Multiple-Access Communication," Proceedings ICC, 1978.

DATE DUE
DATE DE RETOUR

NOV 23 1981			
DEC 29 1981			
JUN 19 1986			
FEV 23 1988			
JUN - 6 1988			

LOWE-MARTIN No. 1137