ATR

AUSTRALIAN TELECOMMUNICATION RESEARCH

VOL. 17, No. 1, 1983

ATR is published twice a year (in May and November) by the Telecommunication Society of Australia. In addition special issues may be published.
ATR publishes papers relating to research into telecommunications in Australia.

CONTRIBUTIONS: The editors will be pleased to consider papers for publication. Contributions should be addressed to the Secretary, ATR, c/- Telecom Australia Research Laboratories, 770 Blackburn Rd., Clayton, Vic., 3168.

SUBSCRIPTIONS: Subscriptions for ATR may be placed with the General Secretary, Telecommunication Society of Australia, Box 4050, G.P.O., Melbourne, Victoria, Australia, 3001. The subscription rates are detailed below. All rates are post free. Remittances should be made payable to the Telecommunication Society of Australia, in Australian currency and should yield the full amount free of any bank charges.

The Telecommunication Society of Australia publishes the following journals:

1. **The Telecommunication Journal of Australia** (3 issues per year)
   Subscription – Free to Members of the Society* resident in Australia
   Non-members of Australia $12.00
   Non-members or Members Overseas $20.00

2. **ATR** (2 issues per year)
   Subscription – To Members of the Society* resident in Australia $9.00
   Non-members in Australia $18.00
   Non-members or Members Overseas $24.00
   Single Copies – To Members of the Society resident in Australia $6.00
   Non-members within Australia $12.00
   Non-members or Members Overseas $16.00

*Membership of the Society $7.00

All overseas copies are sent post-free by surface mail.
Prices are for 1983.

- Enquiries and Subscriptions for all publications may be addressed to:
  The General Secretary, Telecommunication Society of Australia, Box 4050, G.P.O. Melbourne, Victoria, Australia, 3001.

ATR

AUSTRALIAN TELECOMMUNICATION RESEARCH

ISSN 0001-2777

VOLUME 17, NUMBER 1, 1983

Contents

Eric Ramsay Craig

In January 1983, those working in the Telecommunications field were saddened by the death of Mr Eric Craig, following a short illness. Mr Craig was Head of the Telecommunications Technology Branch, Research Laboratories, Telecom Australia.

Eric Craig graduated from St Andrew's University, Scotland, in 1945 with a BSc(Eng) honours degree in electrical engineering. He joined the Australian Post Office (APO) in 1949, and spent the first five years of his career working with the South Australian Administration on various radio projects such as the conversion of the Adelaide–Kangaroo Island system to multi-channel operation.

On being promoted to the APO Research Laboratories in 1955, Eric continued his career in Radio communications, with involvements in radio propagation measurements on systems employing tropospheric and ionospheric scatter techniques. It was at this time that he became aware of the potential of satellites in communications, and in 1960/61, he initiated and carried out a state-of-the-art review of this field.

From 1961 to 1964, Mr Craig worked with the British Post Office on secondment from the APO. He was responsible for the technical direction and management of the Goonhilly satellite earth station, and during this period, he first became involved in the activities of the International Telecommunications Union (ITU) in the satellite field, which later became one of his major interests.

Eric's main ITU involvement was with the CCIR Study Group 4, which is concerned with the characteristics of fixed communications services using satellite systems. He travelled extensively overseas in the course of his work. Eric became Vice-Chairman of this Study Group in 1970 and, at his death, he was its Chairman. He was also the Chairman of the Australian National Study Group 4, which developed a national viewpoint on matters to be put before CCIR Study Group 4.

In his work in Telecom Australia's Research Laboratories from 1964 until 1977, he was in charge of groups working on both transmission measurements and satellite communication studies. In 1977, he became Head of the Telecommunications Technology Branch of the Laboratories, a position which he held until his untimely death.

Challenge . . .

The first years of this decade have witnessed a major inquiry into telecommunications — one with the potential to rival the Vernon Inquiry of the seventies in its impact. The Davidson Inquiry was set up by the Government to inquire into all aspects of telecommunications and make recommendations for a structure to meet future needs.

Since federation, telecommunications have been provided in Australia by the former Post Master-General's Department and, since 1975 by Telecom Australia as a Commonwealth Statutory Authority. In this regard Australia has followed the general style of most nations by having a central governmental responsibility for the provision of its telecommunications. This has enabled a number of aspects to be taken into consideration which could not have been handled as practical propositions in a commercial environment, such as provision of service in very high cost areas at affordable prices, reduced prices to pensioners, etc. The ability to be able to give effect to social needs of this kind is one of the reasons advanced by the proponents of the system of government-provided telecommunications.

In its report, the Davidson Committee recommended the establishment of a commercially oriented, price-to-cost system, devoid of social considerations, which were to be the responsibility of government. In this regard, the Committee envisaged a very radical departure from the system which has been in use since federation. In evaluating the report technically, it is necessary to recognise that there are now very few technical constraints and that economic or cost constraints are progressively reducing due to advances in solid state technology and convergence of the telecommunications and computer technologies.

The stakeholders in the future telecommunications arena are the residential and business customers, industry, government and Telecom. The challenge now facing those responsible for Australia's telecommunications in the future lies in balancing the needs of the stakeholders in the future technological environment so that world parity telecommunications can be provided nationwide at affordable prices in an information based economy.

H.S. WRAGGE.

Demand-Assignment Schemes For SCPC FDMA Satellite Systems With Contiguous Spot Beams

E.S. SEUMAHU
La Trobe University

Mobile and remote-area satellite services usually operate in an environment which involves a multitude of simple ground stations using single-channel-per-carrier (SCPC) frequency division multiple access (FDMA). To increase spectrum utilisation it has been proposed that future satellites in these services be equipped with multiple contiguous spot beams.

This paper discusses the grade of service, traffic capacity, frequency allocation and possible number of channels for contiguous multi beam operation under various demand-assignment schemes. Fixed-, dynamic- and mixed-assignment schemes are considered, with and without frequency reuse. Simple examples are given to highlight the particular features of these schemes and analytical expressions for their performance are deduced where appropriate. The treatment is extended to practical systems such as the proposed North American mobile satellite and the Australian domestic satellite systems.

1. INTRODUCTION

Many communication satellite systems operate in environments which involve a large number of small ground stations, with each station carrying a very low volume of traffic from as few as one subscriber. Single-channel-per-carrier frequency division multiple access is often used in this type of environment. Typical examples are the MARISAT and INMARSAT systems and the remote area services associated with INTELSAT and domestic satellite systems. Dedicated assignment of individual frequencies to each ground station is a very inefficient way to utilise the bandwidth of the satellite in these applications. Hence various demand-assigned multiple-access (DAMA) schemes have been used in existing systems to increase spectrum utilisation. Further economy of spectrum has been achieved by the use of non-contiguous multiple spot beams with the same frequencies being reused in several independent geographic locations.

Recently a high degree of interest has been expressed in the use of satellites for land mobile applications (Refs. 1 and 2), especially in the UHF frequencies where the technology of demand assignment for terrestrial mobile services is well established but where frequency allocation is very meagre. The great progress in the development of the "cellular" method for increasing spectrum utilisation in terrestrial mobile services (Ref. 3) suggests that the same principle could also be applied to mobile satellite services which use SCPC FDMA. In the case of the satellite the "cells" would be replaced by contiguous multiple spot-beams. There are future satellites being proposed to exploit this technique.

This paper investigates various demand-assignment schemes applicable to SCPC FDMA satellite systems where multiple contiguous spot beams are used.

2. DEFINITIONS AND ASSUMPTIONS

The basic terms and definitions used in this paper are those well known in traffic theory literature (eg. Ref. 4) and high capacity terrestrial mobile services (eg. Ref. 5). Some of the terminology needs to be clarified in the context of this paper.

We shall use the term "allocation" to mean the reservation of certain frequencies or channels for a certain purpose and the term "assignment" to mean the actual handing over of one frequency or channel to a particular ground station on demand. The assignment only lasts for the duration of the holding time of that ground station.

Although each channel uses a single frequency, we make a distinction between "frequency" and "channel" because a particular frequency may be associated with more than one channel. A channel may be assigned to only one ground station at any one time whereas a frequency may be assigned to several ground stations simultaneously provided the ground stations are not covered by the same beam or by adjacent beams.

The total number of spot beams in a system will be called L. If a group of frequencies is allocated specifically to beam ℓ, the total number of frequencies in that group will be called N_ℓ. In this case the total number of

channels specifically allocated to that beam, called M_ℓ will also be equal to N_ℓ. If the same arrangement applies to all beams, the total number of channels possible for the system is given by

$$M_T = \sum_{\ell=1}^{L} M_\ell$$

In general the total number of frequencies N_T allocated to the system is less than M_T because of frequency reuse. With dynamic- and mixed-assignment schemes a certain number of frequencies are allocated in a "pool". The actual number of frequencies thus allocated is N_P. The possible number of channels associated with these frequencies will generally be larger than N_P because of reuse.

It is common in traffic theory literature to equate the grade of service with blocking probability. In our case we make a distinction between the two, using blocking probability in association with individual states for which blocking occurs and reserving the grade of service for the sum total of all such probabilities within a beam. The grade of service, G, is the probability that a call is lost because no usable channel can be found when the call is attempted.

To simplify the analysis we shall assume Erlang-B behaviour by the subscribers, i.e. that calls are terminated when unsuccessful, that the calls arrive randomly according to a Poisson distribution of arrival times, and that successful calls have a random holding time with negative exponential probability density function. If the mean number of call arrivals per second is λ and the mean holding time is μ, then the amount of traffic offered is:

$$E = \lambda\mu \qquad (1)$$

where E is measured in Erlang and defined for the busiest hour. In Australian terrestrial mobile services G is held at 0.05 and E is assumed to be between 0.01 and 0.03 Erlangs per subscriber (Ref. 6).

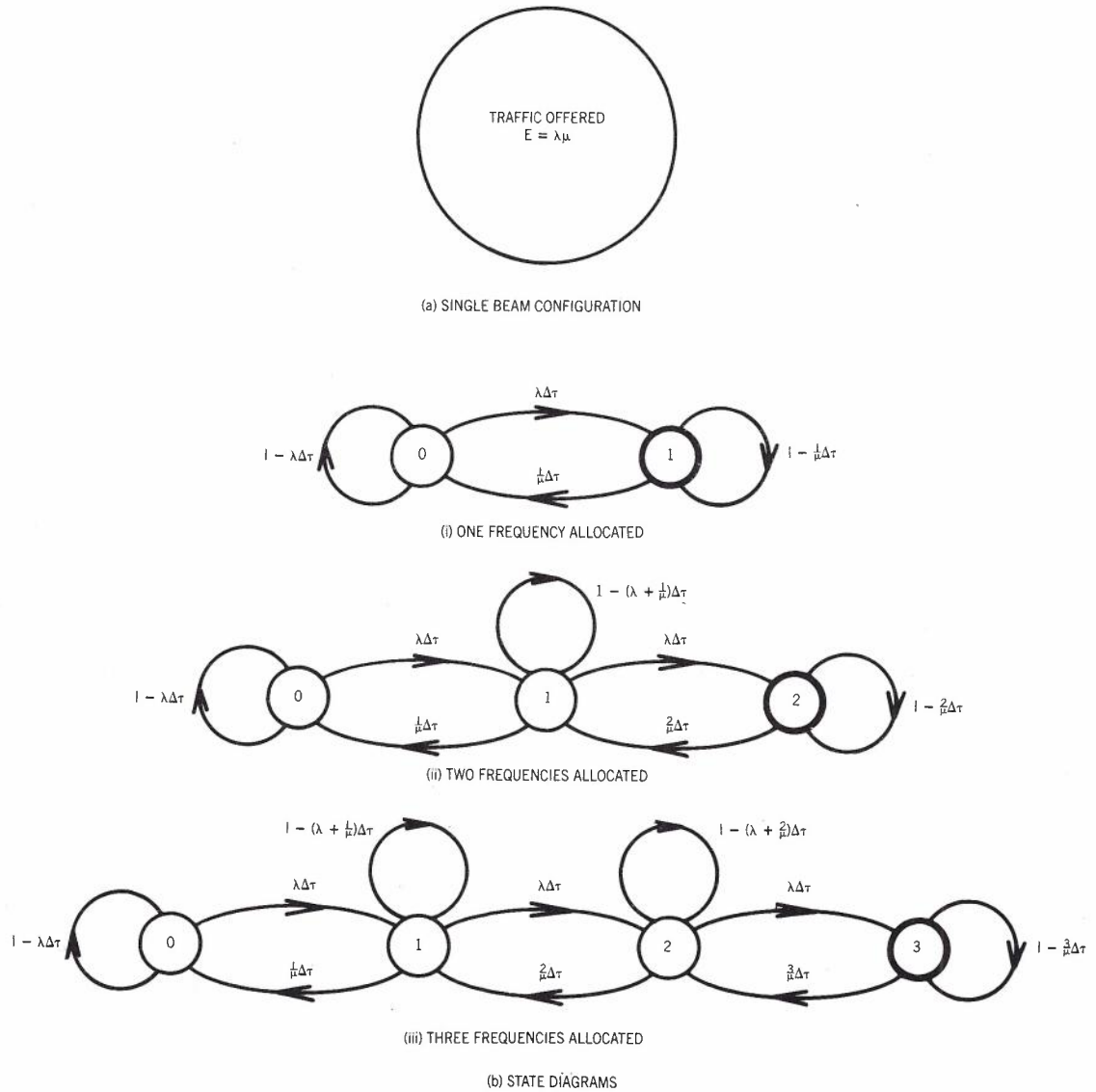


Fig. 1 - Examples of Fixed Assignment in Single Beam

We shall assume that calls are generated independently of one another throughout the whole service area of the satellite although the volumes of traffic may vary from beam to beam. The topology of the beams has a very strong influence on the performance when frequency reuse is employed. Thus certain results in this paper are valid only for those particular topologies being considered.

The discussions shall be confined to the case where the satellite transponder is bandwidth-limited only, with no impairment due to power limitation and non-linearities. Appropriate modifications need to be made before the results can be applied to power-limited satellites.

Single polarisation is assumed. If dual polarisation were employed the spectrum utilisation would be much higher.

3. FIXED ASSIGNMENT

By fixed assignment we mean assignment from groups of frequencies which have been allocated exclusively to each beam on a fixed basis. A total of N_ℓ frequencies is allocated to beam ℓ, $\ell=1,2 \ldots L$. For this beam there is exactly $M_\ell=N_\ell$ number of channels possible. The total number of frequencies for the whole system is

$$N_T = \sum_{\ell=1}^{L} N_\ell$$

which is the same as the total number of channels for the system.

3.1 Single Beam

As a starting point let us consider the case when only one beam is involved as shown in Fig.1a. If a total of N frequencies are allocated to this beam, then at any one time the number of frequencies in use will be given by

$$n = 0,1,2, \ldots\ldots, N$$

n cannot be greater than N. There are no potential users waiting at any time because of the Erlang-B assumption.

This situation is completely specified by N+1 number of states. The probability of being in state n is denoted by p_n which satisfies:

$$0 \leq p_n \leq 1$$

$$\sum_{n=0}^{N} p_n = 1 \qquad (2)$$

The N+1 states may be illustrated by a state diagram, examples of which are shown in Fig.1b.

When n=N a blocking condition exists because no further user can be accommodated at this point in time. The blocking states are marked by heavy circles in Fig.1b.

The probability of being in the blocking state N is simply p_N. Since there is only one blocking state in this single beam sample, the grade of service is simply given by

$$G = p_N$$

The examples in the following sub-sections will show that for any state n the probability p_n can always be expressed as a function of the probability p_0, i.e.

$$p_n = p_n(p_0)$$

The sole blocking state probability can be expressed as

$$p_B(p_0) = p_N(p_0)$$

If we call the sum of all valid-state probabilities p_T then p_T can also be expressed as:

$$p_T = p_T(p_0) = \sum_{n=0}^{N} p_n(p_0) \equiv 1$$

The grade of service can now be expressed as

$$G = \frac{p_B(p_0)}{p_T(p_0)} \qquad (3)$$

3.1.1 One Frequency Allocated to the Beam. In this example we have two states, i.e. n=0 and n=1 as shown in Fig.1b part i). State 0 occurs when there are no calls in progress, i.e. any call arriving at this time will be serviced immediately. Once a call is accepted the state changes to 1 which is a blocking state since no further call can be accepted while the beam is in this state.

The transitional probability for changing from state 0 to state 1 in the infinitesimal time interval $\Delta\tau$ is $\lambda\Delta\tau$ when λ is the average number of arrivals of calls per unit time. The probability of the beam remaining in state 0 during the same time interval is obviously given by $1-\lambda\Delta\tau$. Once in state 1, the probability of returning to state 0 in the time $\Delta\tau$ is $1/\mu\ \Delta\tau$ because μ is the average length of the holding time of the call currently being serviced. The probability of remaining in state 1 during $\Delta\tau$ is just $1-1/\mu\ \Delta\tau$.

To find p_1 in terms of p_0 we note that state 1 may be reached either as a result of a change from state 0 or as a result of the beam remaining in state 1. Therefore we have:

$$p_1 = p_0\lambda\Delta\tau + (1 - \frac{1}{\mu}\Delta\tau)p_1 \qquad (4)$$

from which

$$p_1 = \lambda\mu p_0 = Ep_0 \qquad (5)$$

From equation (2) we get:

$$p_0 + p_1 = (1+E)p_0 = 1 \qquad (6)$$

hence

$$p_0 = \frac{1}{1+E}$$

and

$$p_1 = \frac{E}{1+E}$$

The grade of service is simply p_1. Another way of finding the grade of service is to use equations (3) with

$$p_B(p_0) = p_1(p_0) = Ep_0$$

and

$$p_T(p_0) = p_0+Ep_0$$

to give

$$G = \frac{Ep_0}{p_0+Ep_0} = \frac{E}{1+E} \qquad (7)$$

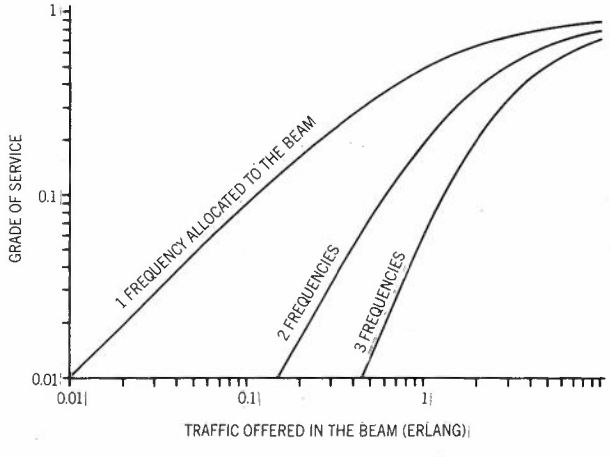


Fig. 2 - Grade of Service for Fixed Assignment in a Particular Beam

This is a general approach and can be readily applied to more complicated cases. The plot of G as a function of E is shown by the left-most curve in Fig.2.

3.1.2 Two Frequencies Allocated to the Beam. When two frequencies are allocated we have three valid states, viz 0, 1 and 2. Fig.1b part ii) shows these three states with all the relevant transitional probabilities in time $\Delta\tau$. We note that transition from state 2 to state 1 is $2/\mu\ \Delta\tau$ since there are 2 calls in progress in this state and each may terminate with probability $1/\mu\ \Delta\tau$. State 2 is the only blocking state.

It can be easily shown by solving two simultaneous linear equations involving p_0, p_1 and p_2 that

$$p_1 = Ep_0$$

as in equation (5), whereas

$$p_2 = \frac{E^2}{2}\ p_0 \qquad (8)$$

The total probability of blocking is simply $p_B(p_0) = E^2/2\ p_0$ and the total probability of all valid states is

$$p_T(p_0) = p_0+p_1+p_2 = (1+E+E^2/2)p_0 \qquad (9)$$

The grade of service is found from equation (3) thus:

$$G = \frac{p_B(p_0)}{p_T(p_0)} = \frac{E^2/2}{1+E+E^2/2} \qquad (10)$$

G is also plotted as the middle curve in Fig.2.

3.1.3 The General Case. For the general case of N frequencies being allocated to the beam, we have N+1 valid states and the possibility of the nth state is given by:

$$p_n = \frac{E^n}{n!}\ p_0\ , \quad 0 \leq n \leq N \qquad (11)$$

The only blocking probability is the one associated with state N, i.e. the total blocking probability is

$$p_B = \frac{E^N}{N!}\ p_0$$

whilst the total probability of all valid states is given by

$$p_T = \sum_{n=0}^{N} \frac{E^n}{n!}\ p_0$$

Hence the grade of service is:

$$G = \frac{E^N/N!}{\sum_{n=0}^{N} E^n/n!} \qquad (12)$$

This is the well known Erlang-B result of traffic theory (Ref. 4).

3.2 Two Contiguous Beams

The case of two contiguous beams is illustrated in Fig.3a. Let the beams be called A and B with individual offered traffic given by $E_A=\lambda_A\mu_A$ and $E_B=\lambda_B\mu_B$ respectively according to equation (1). If N_A frequencies are allocated to beam A and N_B frequencies to beam B, then the total number of valid states is given by $(N_A+1)(N_B+1)$. Each state can be denoted by

$$n_An_B, \quad \begin{matrix} 0 \le n_A \le N_A \\ 0 \le n_B \le N_B \end{matrix}$$

All the states can be represented by a two-dimensional state diagram such as the one in Fig.3b.

The state probabilities $p_{n_An_B}$ obey the relationships

$$\left.\begin{matrix} 0 \le p_{n_An_B} \le 1 \\ \sum_{n_A=0}^{N_A}\sum_{n_B=0}^{N_B} p_{n_An_B} = 1 \end{matrix}\right\} \quad (13)$$

Blocking occurs in beam A for all the states N_An_B, $0 \le n_B \le N_B$ and in beam B for all the states n_AN_B, $0 \le n_A \le N_A$. These states are marked by heavy circles in Fig.3b. We define two grades of service for the two beams:

$$G_A = \sum_{n_B=0}^{N_B} p_{N_An_B}$$

and

$$G_B = \sum_{n_A=0}^{N_A} p_{n_AN_B}$$

Each state probability can be expressed as functions of p_{00}, i.e.

$$p_{n_An_B} = p_{n_An_B}(p_{00})$$

The sum of all valid probabilities is

$$p_T(p_{00}) = \sum_{n_A=0}^{N_A}\sum_{n_B=0}^{N_B} p_{n_An_B}(p_{00}) \equiv 1$$

and the sums of all the blocking probabilities are given by:

$$p_{BA}(p_{00}) = \sum_{n_B=0}^{N_B} p_{N_An_B}(p_{00})$$

$$p_{BB}(p_{00}) = \sum_{n_A=0}^{N_A} p_{n_AN_B}(p_{00})$$

The grades of service may therefore be given as

$$G_A = \frac{p_{BA}(p_{00})}{p_T(p_{00})}$$

and (14)

$$G_B = \frac{p_{BB}(p_{00})}{p_T(p_{00})}$$

3.2.1 Examples: Two Frequencies Allocated per Beam. The case when $N_A=N_B=2$ is illustrated in Fig.3b together with the appropriate transitional probabilities in time $\Delta\tau$. Again, by solving the appropriate number of simultaneous linear equations, we can find all the state probabilities which can be arranged in matrix form thus:

$$\begin{bmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} 1 & E_B & \frac{E_B^2}{2} \\ E_A & E_AE_B & \frac{E_AE_B^2}{2} \\ \frac{E_A^2}{2} & \frac{E_A^2E_B}{2} & \frac{E_A^2E_B^2}{4} \end{bmatrix} p_{00} \quad (15)$$

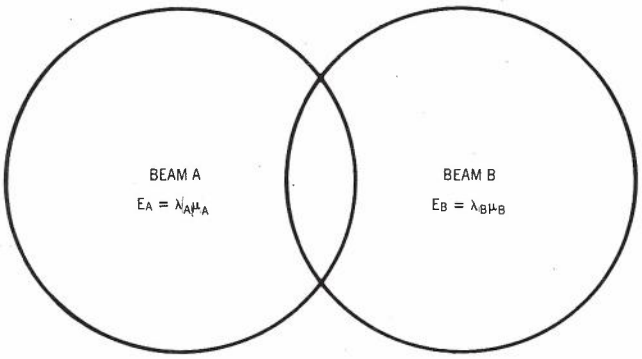
Applying equation (14) using the values of state probabilities found in (15) gives:

$$G_A = \frac{E_A^2/2}{1+E_A+E_A^2/2} \quad (16)$$

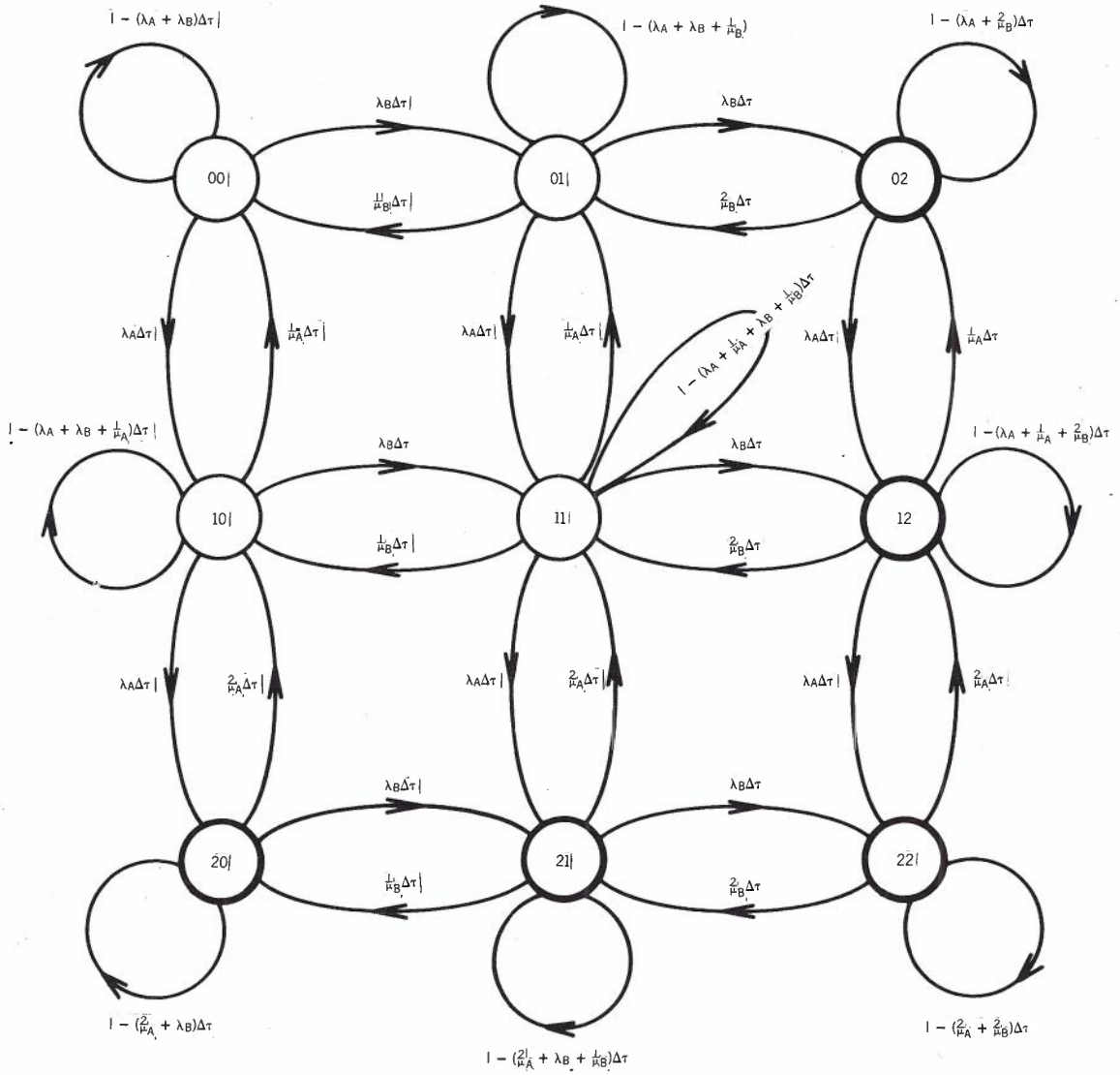
and

$$G_B = \frac{E_B^2/2}{1+E_B+E_B^2/2} \quad (17)$$

We see that G_A is dependent only on the traffic in beam A and G_B is dependent only on the traffic in beam B as should be expected. When $E_A = E_B = E$ the plots of G_A and G_B are the same as that of Fig.2 for the case when 2 frequencies are allocated to a beam.



(a) TWO-BEAM CONFIGURATION



(b) STATE DIAGRAM, TWO FREQUENCIES ALLOCATED PER BEAM

Fig. 3 - Example of Fixed Assignment in Two Beams

3.2.2 The General Case. A generalisation of (15) shows that

$$p_{n_A n_B} = \frac{E_A^{n_A}}{n_A!} \times \frac{E_B^{n_B}}{n_B!} \times p_{00} \qquad (18)$$

Using this in equation (14) gives:

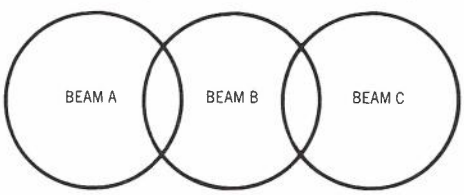
$$G_A = \frac{E_A^{N_A}/N_A!}{\sum_{n_A=0}^{N_A} E_A^{n_A}/n_A!} \qquad (19)$$

and

$$G_B = \frac{E_B^{N_B}/N_B!}{\sum_{n_B=0}^{N_B} E_B^{n_B}/n_B!} \qquad (20)$$

3.3 Three Contiguous Beams

Fig.4a shows the three-beam configuration with beams A, B and C. The state diagram for this case would be very difficult to represent in a two-dimensional diagram. It is better that all the possible combinations of demand in each beam be listed in a table, together with indications of the valid states, the blocking states, and the probabilities $p_{n_A n_B n_C}$ as functions of p_{000}. This has been done in Fig.4b for the case when one frequency is allocated to each beam.



(a) THREE-BEAM CONFIGURATION

COMBINATION NUMBER	STATE A	STATE B	STATE C	PROBABILITY	REMARKS
1	0	0	0	p_{000}	
2	0	0	①	$E_C \cdot p_{000}$	Blocking in C
3	0	①	0	$E_B \cdot p_{000}$	Blocking in B
4	0	①	①	$E_B E_C \cdot p_{000}$	Blocking in B&C
5	①	0	0	$E_A \cdot p_{000}$	Blocking in A
6	①	0	①	$E_A E_C \cdot p_{000}$	Blocking in A&C
7	①	①	0	$E_A E_B \cdot p_{000}$	Blocking in A&B
8	①	①	①	$E_A E_B E_C \cdot p_{000}$	Blocking in A,B&C

Fig. 4 - Example of Fixed Assignment in Three Beams

When N_A, N_B and N_C number of frequencies are allocated on a fixed basis to beams A, B and C respectively, we have the state probabilities:

$$p_{n_A n_B n_C} = \frac{E_A^{n_A}}{n_A!} \cdot \frac{E_B^{n_B}}{n_B!} \cdot \frac{E_C^{n_C}}{n_C!} \cdot p_{000} \qquad (21)$$

The grades of service for A and B are as given in equations (19) and (20) respectively whilst the grade of service for beam C is given by:

$$G_C = \frac{E_C^{N_C}/N_C!}{\sum_{n_C=0}^{N_C} E_C^{n_C}/n_C!} \qquad (22)$$

3.4 General Expressions for Fixed Assignment

In the most general case of L contiguous beams with the numbers of allocated frequencies

$$N_1, N_2, \ldots, N_\ell, \ldots, N_L$$

and offered traffic of

$$E_1, E_2, \ldots, E_\ell, \ldots, E_L$$

we have a total of

$$\prod_{\ell=1}^{L} (N_\ell + 1) \qquad (23)$$

number of valid states, each having the probability:

$$p_{n_1 n_2 \ldots n_\ell \ldots n_L} = \left(\prod_{\ell=1}^{L} \frac{E_\ell^{n_\ell}}{n_\ell!} \right) p_{00\ldots0\ldots0} \qquad (24)$$

The total probability of all valid states is given by the general expression:

$$p_T = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \cdots \sum_{n_\ell=0}^{N_\ell} \cdots \sum_{n_L=0}^{N_L} \left(\prod_{\ell=1}^{L} \frac{E_\ell^{n_\ell}}{n_\ell!} \right) p_{00\ldots0\ldots0} \qquad (25)$$

which reduces to:

$$p_T = \prod_{\ell=1}^{L} \left(\sum_{n_\ell=0}^{N_\ell} \frac{E_\ell^{n_\ell}}{n_\ell!} \right) p_{00\ldots0\ldots0} \qquad (26)$$

and can also be expressed as:

$$p_T = \left(\sum_{n_k=0}^{N_k} \frac{E_k^{n_k}}{n_k!} \right) \prod_{\substack{\ell=1 \\ \ell \neq k}}^{L} \left(\sum_{n_\ell=0}^{N_\ell} \frac{E_\ell^{n_\ell}}{n_\ell!} \right) p_{00\,..\,0\,..\,0} \qquad (27)$$

The total probability of all blocking states in a particular beam k is found by performing a similar summation as in (25) but only over the states for which $n_k = N_k$. The result reduces to:

$$p_{Bk} = \frac{E_k^{N_k}}{N_k!} \prod_{\substack{\ell=1 \\ \ell \neq k}}^{L} \left(\sum_{n_\ell=0}^{N_\ell} \frac{E_\ell^{n_\ell}}{n_\ell!} \right) p_{00\,..\,0\,..\,0} \qquad (28)$$

The grade of service in the k^{th} beam is therefore given by the ratio of (28) over (27), i.e.

$$G_k = \frac{E_k^{N_k}/N_k!}{\sum_{n_k=0}^{N_k} E_k^{n_k}/n_k!} \qquad (29)$$

4. FIXED ASSIGNMENT WITH FREQUENCY REUSE

Frequency reuse is possible with contiguous spot beams under certain conditions. A frequency cannot be reused within a single beam (unless dual polarization is used). Because it is not possible to produce completely non-overlapping cont guous spot beams from the satellite, it is also not advisable for frequencies to be reused in two adjacent beams. We shall assume that the contours of the spot beams are sufficiently sharp for interference to be negligible when a frequency is reused in any non-adjacent beams.

The blocking probabilities and hence the traffic capacity for fixed allocation with reuse is the same as those for pure fixed allocation without reuse when the same number of channels are present. However, the number of frequencies needed to maintain this performance is less with reuse. Alternatively, for the same number of allocated frequencies, the number of possible channels and hence the performance are increased by reuse.

For example, in the case of the three beams shown in Fig.4a we may use 3 channels (one for each beam) but require only 2 frequencies since the frequency allocated to beam A can also be used by the channel in beam C. If the number of frequencies allocated to beam A and B are N_A and N_B respectively the total number of frequencies allocated is:

$$N_T = N_A + N_B$$

The number of channels possible in beam A is $M_A = N_A$. Similarly for beam B we have $M_B = N_B$. In channel C the number of channels possible is $M_C = N_A$ so we have for the total number of channels:

$$M_T = M_A + M_B + M_C = 2N_A + N_B \qquad (30)$$

We define reuse factor as the total number of channels over the total number of frequencies. For this three-beam example we have:

$$\text{Reuse factor} = \frac{2N_A + N_B}{N_A + N_B} \qquad (31)$$

Because of the adjacency restriction, the topology of the collection of spot beams has a very strong influence on the reuse factor and thus also on the overall performance of the system.

5. DYNAMIC ASSIGNMENT

With fixed assignment a number of frequencies are allocated to a particular beam and these frequencies cannot be used anywhere else even when they are idle during the times of low traffic demand. To improve frequency utilisation, these idle frequencies should be employed somewhere else. This is the basis of dynamic assignment schemes.

In pure dynamic assignment, a certain number of frequencies N_P are allocated in a pool for use by the whole system. These frequencies are assigned on demand to users in any beam on a first-come-first-served basis. Once a frequency has been assigned to a user it cannot be used again during the holding time of that particular user. Frequencies which are no longer required are returned to the pool. There are the same number of channels as there are frequencies. Blocking occurs when all the allocated frequencies have been used up. Clearly, blocking occurs simultaneously in all beams and hence each beam has the same grade of service.

The technology to implement dynamic assignment is well established in terrestrial cellular mobile systems. Each ground station is supplied with synthesized transmitting and receiving equipment. A centralised processor decides on the frequency to be assigned to a particular ground station and, by means of a control channel, instructs the station to synthesize that frequency for use during the duration of a call.

5.1 Two-Beam Example

The state diagram of a two-beam system with $N_P = 2$ is shown in Fig.5. We see that, although there are 9 possible combinations of demand, only 6 states are valid. The other states (12,21,22) represent demands which exceed the number of available channels and are therefore

not allowed under the Erlang-B assumption.

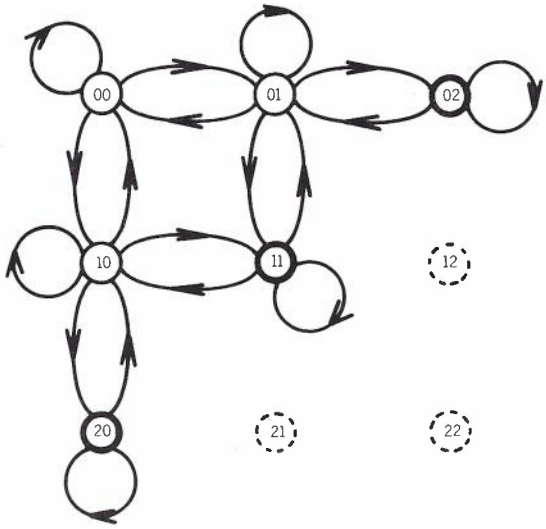


Fig. 5 - Two-Beam Example of Dynamic Assignment

Blocking occurs in the three states 02, 11 and 20 where the total usage in both beams exactly equals the total number of available channels. Following the procedures of Section 3 it can be shown that the state probabilities are given by:

$$\begin{bmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} 1 & E_B & \frac{E_B^2}{2} \\ E_A & E_A E_B & 0 \\ \frac{E_A^2}{2} & 0 & 0 \end{bmatrix} p_{00} \qquad (32)$$

The total probability of the six valid states is:

$$p_T(p_{00}) = (1+E_A+E_B+E_A E_B+ \frac{E_A^2}{2} + \frac{E_B^2}{2})\ p_{00} \qquad (33)$$

which can be rewritten as:

$$p_T(p_{00}) = [\,1+(E_A+E_B) + (E_A+E_B)^2/2\,]\ p_{00} \qquad (34)$$

The total probability of the three blocking states is:

$$p_B(p_{00}) = (\frac{E_A^2}{2} + E_A E_B + \frac{E_B^2}{2})\ p_{00}$$

which can be written as:

$$p_B(p_{00}) = \frac{(E_A + E_B)^2}{2}\ p_{00} \qquad (35)$$

So the grade of service is just:

$$G = \frac{(E_A+E_B)^2/2}{1+(E_A+E_B)+(E_A+E_B)^2/2} \qquad (36)$$

We see that this has just the same form as equation (10) with a total offered traffic of E_A+E_B. Hence the plot of G is the same as that in Fig.2 for two frequencies, but with the traffic axis replaced by $E_T=E_A+E_B$.

5.2 Three-Beam Example

When three beams are used, the states are better represented by a state table. This is illustrated in Table 1 for the case when $N_P = 1$ with 8 possible combinations of demand. Combinations numbered 4, 6, 7 and 8 are invalid because they require more than 1 channel to be provided. Blocking occurs simultaneously in all beams during the states corresponding to combinations number 2, 3 and 5.

TABLE 1 - Three-Beam Example of Dynamic Assignment

COMBINATION NUMBER	STATE A	STATE B	STATE C	PROBABILITY ($\times p_{000}$)	REMARKS
1	0	0	0	1	
2	(0)	(0)	(1)	E_C	Blocking in A,B&C
3	(0)	(1)	(0)	E_B	Blocking in A,B&C
4	~~0~~	~~1~~	~~1~~	$E_B E_C$	Invalid
5	(1)	(0)	(0)	E_A	Blocking in A,B&C
6	~~1~~	~~0~~	~~1~~	$E_A E_C$	Invalid
7	~~1~~	~~1~~	~~0~~	$E_A E_B$	Invalid
8	~~1~~	~~1~~	~~1~~	$E_A E_B E_C$	Invalid

Note: Encircled states indicate blocking.
States crossed out are invalid.

The total probability of valid states is given by:

$$p_T(p_{000}) = (1+E_C+E_B+E_A)\ p_{000} \qquad (37)$$

whereas the total probability for all blocking states is given by:

$$p_B(p_{000}) = (E_A+E_B+E_C)\ p_{000} \qquad (38)$$

making the overall grade of service:

$$G = \frac{(E_A+E_B+E_C)}{1+(E_A+E_B+E_C)} \qquad (39)$$

Again we see that this has the same form as for fixed assignment with a single beam given in equation (7) but with a total offered traffic of

$E_T = E_A+E_B+E_C$. The corresponding curve for one frequency in Fig.2 may therefore be suitably modified for this example.

5.3 General Expressions for Dynamic Assignment

In contrast with fixed assignment where we have a fixed number of frequencies and channels allocated to each beam, here we have a total of N_P frequencies and N_P channels available to be shared among the beams according to demand. The total number of possible combinations of demand is

$$\prod_{\ell=1}^{L} (N_P + 1) = (N_P + 1)^L \qquad (40)$$

although not all of these represent valid states. The probabilities associated with the valid states are as given in equation (24).

Of the total number of possible combinations, only the states where

$$n_1+n_2+ \ldots n_\ell+ \ldots n_L \leq N_P \qquad (41)$$

are actually valid. The total probabilities of all these valid states is given by the general expression:

$$p_T = \sum_{n_1=0}^{N_P} \sum_{n_2=0}^{N_P} \ldots \sum_{n_\ell=0}^{N_P} \ldots \sum_{n_L=0}^{N_P} \left(\prod_{\ell=1}^{L} \frac{E_\ell^{n_\ell}}{n_\ell!} \right) p_{00 \ldots 0 \ldots 0} \qquad (42)$$

$$\text{for } n_\ell\text{'s where } \sum_{\ell=1}^{L} n_\ell \leq N_P$$

which reduces to:

$$p_T = \sum_{n=0}^{N_P} \frac{\left(\sum_{\ell=1}^{L} E_\ell \right)^n}{n!} p_{00 \ldots 0 \ldots 0} \qquad (43)$$

Blocking occurs in all beams simultaneously when

$$n_1+n_2+ \ldots n_\ell+ \ldots n_L = N_P \qquad (44)$$

The sum of the probabilities of all blocking states is given by:

$$p_B = \sum_{n_1=0}^{N_P} \sum_{n_2=0}^{N_P} \ldots \sum_{n_\ell=0}^{N_P} \ldots \sum_{n_L=0}^{N_P} \left(\prod_{\ell=1}^{L} \frac{E_\ell^{n_\ell}}{n_\ell!} \right) p_{00 \ldots 0 \ldots 0}$$

$$\text{for } n_\ell\text{'s where } \sum_{\ell=1}^{L} n_\ell = N_P \qquad (45)$$

which reduces to

$$p_B = \frac{\left(\sum_{\ell=1}^{L} E_\ell \right)^{N_P}}{N_P!} p_{00 \ldots 0 \ldots 0} \qquad (46)$$

Taking the ratio of (46) over (43) gives the grade of service:

$$G = \frac{\left(\sum_{\ell=1}^{L} E_\ell \right)^{N_P} / N_P!}{\sum_{n=0}^{N_P} \left(\sum_{\ell=1}^{L} E_\ell \right)^n / n!} \qquad (47)$$

6. DYNAMIC ASSIGNMENT WITH FREQUENCY REUSE

In pure dynamic assignment we have the same number of frequencies as the number of channels. The nature of contiguous spot beams is such that frequencies can be reused in non-adjacent beams, resulting in an increase of available channels for the same frequency allocation or a reduction in the required frequency allocation for the same number of channels. This is the principle of dynamic assignment plus frequency reuse.

In addition to its spectral economy, dynamic assignment with reuse also results in improved performance. This scheme has the effect of relieving some of the blocking conditions which exist in the pure dynamic assignment.

As in the case of fixed assignment with reuse, the topology of the collection of spot beams has a very strong influence on the overall performance of the system. The topology also determines the algorithm for computing the performance of dynamic assignment with reuse; analytical expressions are usually not very helpful in this case.

6.1 Three-Beam Example

The state table for the three-beam example of Fig.4a with N_P=1 is shown in Table 2. It is clear for this topology that the number of possible channels is now 2 because the same frequency can be assigned simultaneously in beam A as in beam C.

TABLE 2 - Three-Beam Example of Dynamic Assignment Plus Frequency Reuse

COMBINATION NUMBER	STATE A	STATE B	STATE C	PROBABILITY ($\times p_{000}$)	REMARKS
1	0	0	0	1	
2	0	(0	1)	E_C	Blocking in B&C
3	(0	(1)	0)	E_B	Blocking in A,B&C
4	~~0~~	~~1~~	~~1~~	E_BE_C	Invalid
5	(1	0)	0	E_A	Blocking in A&B
6	(1	(0)	1)	E_AE_C	Blocking in A,B&C
7	~~1~~	~~1~~	~~0~~	E_AE_B	Invalid
8	~~1~~	~~1~~	~~1~~	$E_AE_BE_C$	Invalid

Three effects may be noted when comparing Table 2 with Table 1. Firstly in combination number 2 there is a release of blocking in beam A. This is due to the fact that an additional channel is now available for this beam using the same frequency already assigned in C. The same effect is observed on beam C in combination number 5. Secondly, combination number 6 now corresponds to a valid state because the sum of demands in beam A and C does not exceed the possible number of channels although the same sum does exceed the allocated number of frequencies. Thirdly, the blocking conditions exist in pairs of adjacent beams rather than simultaneously in all beams. All these effects contribute to an improvement in performance.

With these new alterations we now have for the total probabilities of all valid states:

$$P_T = (1+E_C+E_B+E_A+E_AE_C)\ P_{000} \tag{48}$$

The total probabilities of the blocking states in each of the three beams are now given by:

$$P_{BA} = (E_B+E_A+E_AE_C)\ P_{000} \tag{49}$$

$$P_{BB} = (E_C+E_B+E_A+E_AE_C)\ P_{000} \tag{50}$$

$$P_{BC} = (E_C+E_B+E_AE_C)\ P_{000} \tag{51}$$

Hence the grades of service are now:

$$G_A = \frac{E_A+E_B+E_AE_C}{1+E_A+E_B+E_C+E_AE_C} \tag{52}$$

$$G_B = \frac{E_A+E_B+E_C+E_AE_C}{1+E_A+E_B+E_C+E_AE_C} \tag{53}$$

$$G_C = \frac{E_B+E_C+E_AE_C}{1+E_A+E_B+E_C+E_AE_C} \tag{54}$$

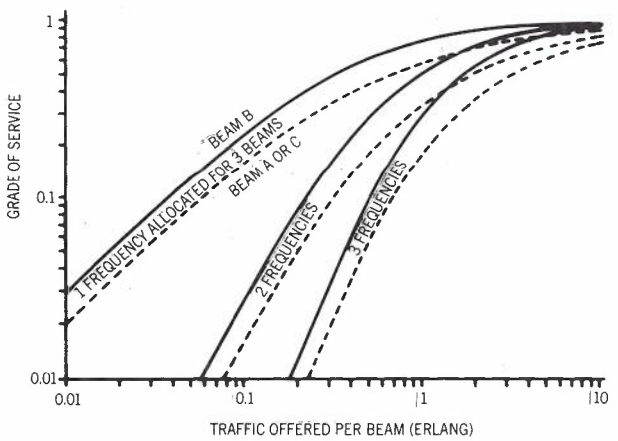


Fig. 6 - Examples of Dynamic Assignment Plus Reuse

It is not always possible to reduce equations (52), (53) and (54) in the simple manner as with fixed and pure dynamic assignment. We note that G_B is greater than either G_A or G_C. This is to be expected because the topology of the beams is such that there is a minimum of freedom in assigning channels to beam B as compared to beams A and C. The grades of service G_A, G_B and G_C are plotted in the left-most curves of Fig.6 for $E_A=E_B=E_C=E$.

6.2 General Expressions for Dynamic Assignment With Reuse

As with pure dynamic assignment, for Np total frequencies the number of possible combinations of demand is as given by equation (40) with the associated probabilities given in equation (24). Of this number of combinations only the states with

$$n_1+n_2+ \ldots n_\ell+ \ldots n_L \leq M_P \tag{55}$$

are actually valid, where M_P is the total number of channels possible because of reuse.

The total probability of all valid states is given by the general expression:

$$P_T = \sum_{n_1=0}^{N_P} \sum_{n_2=0}^{N_P} \ldots \sum_{n_\ell=0}^{N_P} \ldots \sum_{n_L=0}^{N_P} \left(\prod_{\ell=1}^{L} \frac{E_\ell^{n_\ell}}{n_\ell!} \right) P_{00\ldots0\ldots0} \tag{56}$$

$$\text{for } n_\ell\text{'s where } \sum_{\ell=1}^{L} n_\ell \leq M_P$$

This expression is generally not reducible.

Blocking occurs in a beam when the sum of the frequencies being used in that beam and the frequencies being used in all beams adjacent to it is equal to N_P. Mutually adjacent beams form a group. Blocking occurs simultaneously in members of such groups. A general expression for the total blocking probability in beam k would be:

$$P_{Bk} = \sum_{n_1=0}^{N_P} \sum_{n_2=0}^{N_P} \ldots \sum_{n_\ell=0}^{N_P} \ldots \sum_{n_L=0}^{N_P} \left(\prod_{\ell=1}^{L} \frac{E_\ell^{n_\ell}}{n_\ell!} \right) P_{00\ldots0\ldots0} \tag{57}$$

$$\text{for } n_\ell\text{'s where } \sum n_\ell = N_P \quad \text{(all } \ell \text{ adjacent to and including k)}$$

This expression may not be analytically very convenient.

The grade of service in beam k would be given by

$$G_k = \frac{P_{Bk}}{P_T} \tag{58}$$

where P_{Bk} and P_T may be found by a suitable algorithm which enumerates all the appropriate probabilities. This has been done for the three beam configuration of Fig.4. The results are also shown in Fig.6 for $N_P=2$ and $N_P=3$ respectively with $E_A=E_B=E_C=E$.

7. COMPARISON OF ASSIGNMENT SCHEMES

There are many ways the performance of the different assignment schemes may be compared with each other. For example, they may be compared on the basis of equal number of total frequencies allocated, on the basis of equal number of possible channels, on the basis of equal size of state tables, etc.

7.1 State Tables

The way that the various assignment schemes differ from each other may be illustrated by comparing their state tables. This is done in Table 3 for the case of the three-beam configuration of Fig.4 when the state tables are 27 entries long, i.e. when each beam may have access of up to 2 channels.

The effect of spectrum reuse with dynamic assignment is to reduce the interdependence of blocking between the beams and to reduce the number of invalid combinations. Compared to pure dynamic assignment, the effect of reuse is to increase the total number of channels for the same total number of frequencies, with a consequent reduction in blocking and grade of service.

TABLE 3 - Sample State-Table Comparison of Different Assignment Methods

METHOD OF ASSIGNMENT	FIXED	FIXED WITH REUSE	DYNAMIC	DYNAMIC WITH REUSE
TOTAL FREQUENCIES ALLOCATED	6	4	2	2
TOTAL CHANNELS POSSIBLE	6(=2+2+2)	6(=2+2+2)	2	4
Combination No. 1	0 0 0	0 0 0	0 0 0	0 0 0
2	0 0 1	0 0 1	0 0 1	0 0 1
3	0 0 ②	0 0 ②	0 0 2	0 0 2
4	0 1 0	0 1 0	0 1 0	0 1 0
5	0 1 1	0 1 1	0 1 1	0 1 1
6	0 1 ②	0 1 ②	~~0 1 2~~	~~0 1 2~~
7	0 ② 0	0 ② 0	0 2 0	0 2 0
8	0 ② 1	0 ② 1	~~0 2 1~~	~~0 2 1~~
9	0 ② ②	0 ② ②	~~0 2 2~~	~~0 2 2~~
10	1 0 0	1 0 0	1 0 0	1 0 0
11	1 0 1	1 0 1	1 0 1	1 0 1
12	1 0 ②	1 0 ②	~~1 0 2~~	1 0 2
13	1 1 0	1 1 0	1 1 0	1 1 0
14	1 1 1	1 1 1	~~1 1 1~~	1 1 1
15	1 1 ②	1 1 ②	~~1 1 2~~	~~1 1 2~~
16	1 ② 0	1 ② 0	~~1 2 0~~	~~1 2 0~~
17	1 ② 1	1 ② 1	~~1 2 1~~	~~1 2 1~~
18	1 ② ②	1 ② ②	~~1 2 2~~	~~1 2 2~~
19	② 0 0	② 0 0	2 0 0	2 0 0
20	② 0 1	② 0 1	~~2 0 1~~	2 0 1
21	② 0 ②	② 0 ②	~~2 0 2~~	2 0 2
22	② 1 0	② 1 0	~~2 1 0~~	~~2 1 0~~
23	② 1 1	② 1 1	~~2 1 1~~	~~2 1 1~~
24	② 1 ②	② 1 ②	~~2 1 2~~	~~2 1 2~~
25	② ② 0	② ② 0	~~2 2 0~~	~~2 2 0~~
26	② ② 1	② ② 1	~~2 2 1~~	~~2 2 1~~
27	② ② ②	② ② ②	~~2 2 2~~	~~2 2 2~~

We see from Table 3 that the blocking conditions for fixed assignment and for fixed assignment plus reuse are identical. In both cases each beam is blocked independently and there are no invalid combinations of demand. The same total number of channels are used by the system. However, the required number of frequencies to support this is less with spectrum reuse.

Table 3 also shows the contrast between pure dynamic assignment and fixed assignment. Blocking in the former is highly dependent from beam to beam, in fact blocking always occurs simultaneously in all beams. There are a large number of invalid combinations of demands. The total number of frequencies equals the total number of channels, both much less than the numbers required in either of the fixed assignment schemes.

7.2 Grades of Service

For the three-beam topology of Fig.4 and a total number of 3 frequencies allocated to the whole system the resulting grades of service can be readily computed. In the case of fixed assignment we would allocate one frequency per beam and use equation (29) to calculate the grade of service. The result is plotted as the left-most curve in Fig.7 when the offered traffic is identical in each beam. The same result would be obtained for fixed assignment plus reuse, but here the total frequencies required would only be 2.

With dynamic assignment we would use equation (47) to calculate the grade of service. The result is also plotted in Fig.7 as the second curve from the left for the case when the offered traffic is identical for each beam. We note that dynamic assignment gives a large amount of improvement in grade of service when

the volume of traffic is low. The advantage decreases as traffic increases.

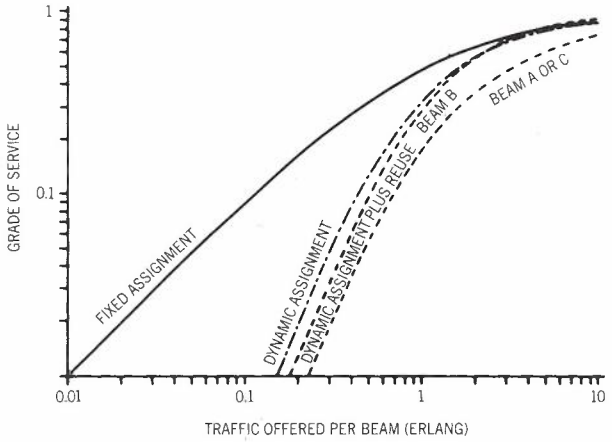


Fig. 7 - Comparison of Different Assignment Methods - 3 Frequencies Allocated to 3 Beams

When frequency reuse is added to dynamic assignment we may use equations (56), (57) or (58) to compute the grade of service. The result for equal values of traffic in each beam is again shown in Fig.7 where a distinction is apparent between the grades of service in beams A and C on the one hand and in beam B on the other. Further improvement to dynamic allocation due to frequency reuse is evident. The improvement is quite significant. However, in beam B the grade of service may actually be worse than the fixed or pure dynamic assignments at very high volumes of traffic. This effect is similar to the one well known in terrestrial high-capacity cellular mobile systems (Refs. 5, 7 and 8).

7.3 Traffic Capacity

One important consideration in the design and dimensioning of SCPC FDMA satellite systems is the traffic capacity and the total number of individual ground terminals which may be serviced by the satellite. The capacity definition depends on the allowable grade of service. The total number of ground terminals serviceable is found by dividing the traffic capacity (in Erlang) by the individual traffic per terminal. In the Australian mobile services context the grade of service is kept at 0.05 and the traffic offered per mobile terminal is about 0.025 Erlang. For this case there are some 40 terminals serviceable for every Erlang of capacity on the satellite.

With the grade of service fixed at 0.05, we can calculate the traffic capacities for the various assignment schemes with the help of equations (29), (47) and (58) for a given number of N_T or N_P of allocated frequencies. Assuming equal traffic in all beams and the three-beam topology of Fig.4, the results are shown in Fig.8. We note here the general improvement from pure fixed assignment right through to dynamic assignment with spectrum reuse. We also note the cross-over effect between dynamic assignment and fixed assignment with reuse.

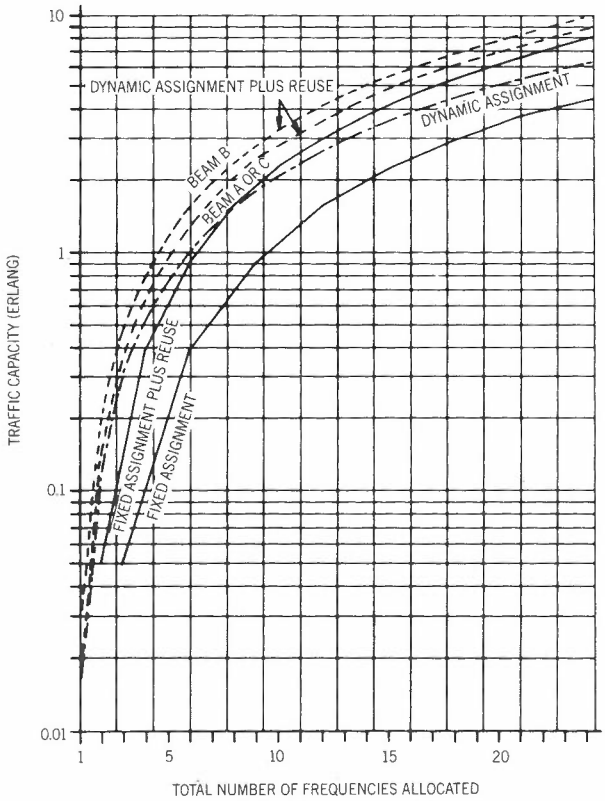


Fig. 8 - Comparison of Different Assignment Methods at 5% Grade of Service

7.4 Comparison with Single Global Beam

A realistic basis of comparison is to consider the case of a single, appropriately shaped, beam serving the entire area with all the allocated frequencies available in straight DAMA mode. In this case the grade of service is as given in equation (12), with $N = N_T$ being the total number of frequencies allocated to the whole service area. The result is identical to the case of pure dynamic assignment where N_T is the total number of frequencies available in the pool and each frequency is allocated according to demand irrespective of the beam where the demand originates.

8. SAMPLE APPLICATION

8.1 Reuse Patterns

Beam topology does not have any effect on pure fixed or pure dynamic assignments. On the other hand the topology has a very strong influence on the performance whenever frequencies are being reused. This is due to the fact that frequencies cannot be reused in adjacent beams so that blocking conditions vary according to the way that the beams are positioned relative to each other.

In terrestrial cellular reuse schemes, the aim is to maximise the reuse factor by making the cells very small and packing them as tightly as possible. This results in the adoption of "honeycomb" patterns of reuse (Ref. 3). The counterpart of this for satellite systems is shown in Fig.9.

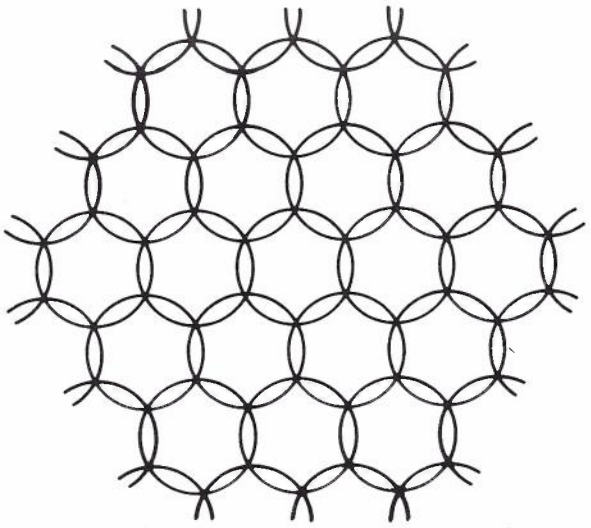


Fig. 9 - High Capacity Reuse Pattern

Unfortunately, it is much more difficult in satellite systems to shape the beam into very small tightly-packed spots. Furthermore, the design of spot beam "footprints" is often governed by geographical and political considerations. As a result the number of spot beams is usually much less than the number of cells in terrestrial systems and the reuse factor is also smaller for the satellite systems. The following examples illustrate typical reuse patterns being contemplated for future satellites and the resulting performance.

8.2 North American Example

One of the reuse patterns which has been considered for the North American continent is shown in Fig.10, (Ref. 9). In this proposal there are 4 groups of frequencies labelled A, B, C and D. From Fig.10 it can be seen that group A and group C frequencies can be reused 3 times whilst group B and group D frequencies can be reused 6 times. If the number of frequencies allocated to each group is equal and given by N, then the total number of frequencies allocated is $N_T = 4N$. The total number of channels possible is $M_T = 3N + 6N + 3N + 6N = 18N$ so the average reuse factor is 4.5.

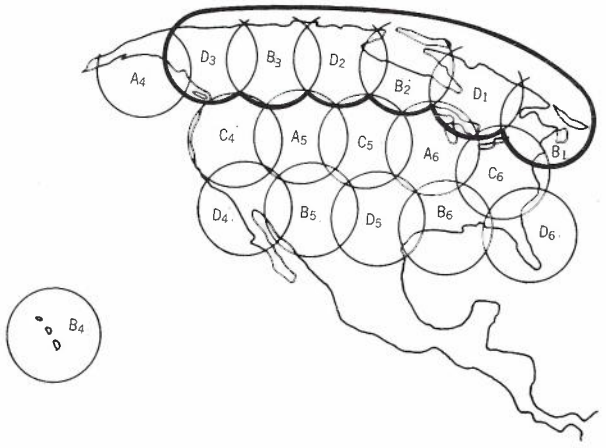


Fig. 10 - Reuse Pattern for North American Example

The Canadian beams can be treated in isolation because of the particular topology of Fig.10 (Ref. 10). Here there is a system of 6 contiguous beams with 2 groups of frequencies labelled B and D. Each group is reused 3 times. For this isolated topology we can calculate the performance under various assignment schemes.

Suppose that 6 frequencies are allocated to the system as a whole. Under a pure fixed allocation scheme and assuming equal traffic in all beams, we would allocate 1 frequency per beam using a total of 6 channels. The resulting grade of service is shown as the left-most curve in Fig.11. At the other extreme when using dynamic allocation plus frequency reuse we would allocate all 6 frequencies in a pool with a total of 18 possible channels to be shared among the beams according to a proper algorithm. The grades of service are computed using equation (58) and shown in the group at the rightmost end of Fig.11. As expected, different blocking conditions exist in different beams. Because of symmetry there are only 3 different grades of service, one for beam B_1 and D_3, one for beam B_3 and D_1 and one for beam B_2 and D_2.

The reason for the three differing grades of service is the peculiar adjacency configurations relative to a beam. Each of beams B_1 and D_3 has 5 beams arranged linearly to one side and no beams on the other side. Beams B_3 and D_1 have 4 beams on one side and 1 beam on the other whereas B_2 and D_2 have 3 beams on one side and 2 beams on the other.

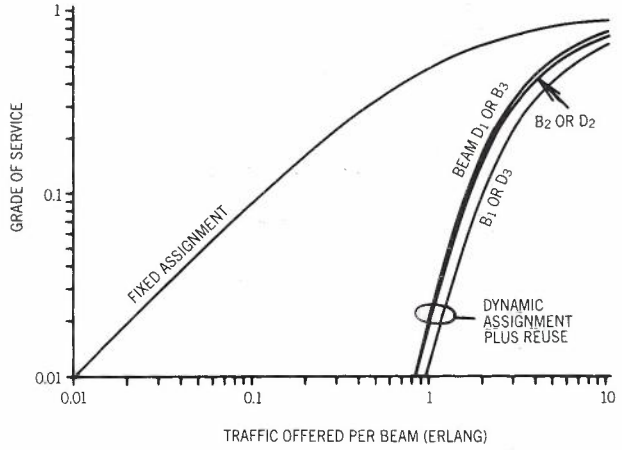


Fig. 11 - Comparison of Assignment Methods for Canadian Example - 6 Frequencies Allocated to 6 Beams

An examination of Fig.11 shows the very extensive improvement possible by means of dynamic assignment plus reuse. For example, at 10% grade of service there is up to 18 fold improvement in traffic capacity. At 5% grade of service the improvement is as much as 28 fold.

8.3 Australian Example

The proposed Australian domestic satellite system has provision for 4 contiguous spot beams (Ref. 11). If these beams were to be used for SCPC FDMA applications, we could apply the theoretical results to calculate the performance of the system using the beam topology shown in Fig.12. The four beams are labelled WA

(Western Australia), CA (Central Australia), NE (North-East) and SE (South-East) beams respectively.

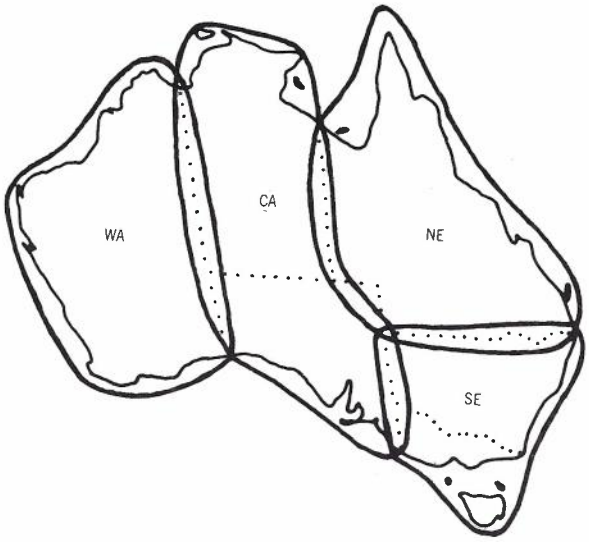


Fig. 12 - Reuse Pattern for Australian Example

We note that any frequencies assigned in the WA beam may also be assigned simultaneously in either the NE or the SE beam but not in both. Any frequencies assigned in the NE or SE beams may be reused in the WA beam. Assuming equal traffic in all beams, the minimum reuse factor is 4/3 because for every 3 frequencies allocated we can achieve a minimum of 4 channels by reuse.

The grades of service for the Australian example are shown in Fig.13 for the cases where 1 and 4 frequencies are allocated to the whole system under dynamic assignment plus reuse. We note here the great discrepancies between beams carrying the same traffic. The WA beam has the best grade of service because it has the least constraint in the assignment algorithm due to its position relative to the other beams. The NE and SE beams have equal performance, both constrained by two adjacent beams. The worst performance is displayed in the CA beam which is constrained by a total of 3 adjacent neighbours.

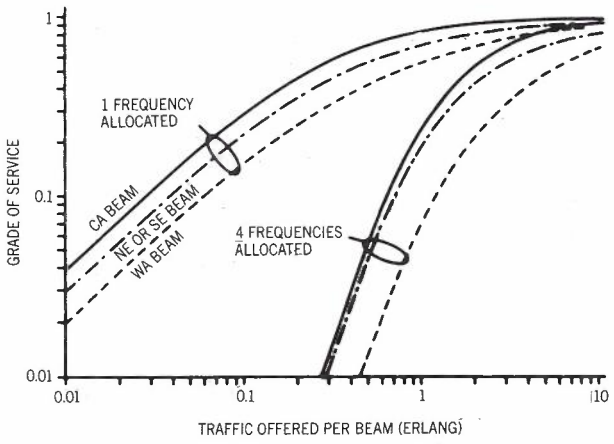


Fig. 13 - Australian Example with Dynamic Assignment Plus Reuse

To calculate the traffic capacities for the various assignment schemes, we fix the grade of service at the Australian practice of 5% and apply the same procedure as in Section 7.3. The results are shown in Fig.14 where we note the extremes of performance with a 17 fold possible improvement when the total number of frequencies allocated is 4. Bearing in mind that every Erlang of traffic capacity corresponds to about 40 terminals in this context, the improvement is quite substantial.

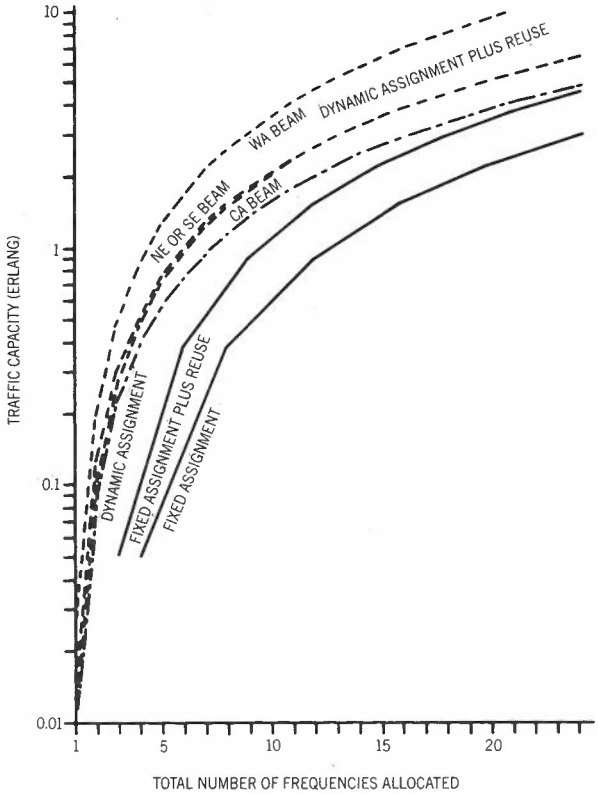


Fig. 14 - Comparison of Assignment Methods for Australian Example

9. MIXED ASSIGNMENT

Fixed assignments are the simplest to implement, both on the ground and at the satellite. Dynamic assignments require switching either on the satellite or at a central point on the ground. Obviously dynamic assignment is more costly than fixed assignment. However, the increased capacity and more efficient use of the spectrum makes dynamic assignment preferable and even mandatory in some cases.

From the previous discussions it is clear that fixed assignment is very inefficient for low volumes of traffic whereas dynamic assignment gives very little advantage and may even be worse at very high volumes of traffic. This suggests a compromise which may improve the performance at reasonable cost, i.e. a mixed assignment method.

In mixed assignment a portion N_F of the total allocated frequencies is used in fixed assignment and the rest of the frequencies are reserved in a pool for dynamic assignment. The total number of the dynamic-assignment frequencies is N_D and the total allocation is

N_F+N_D. The calculation of performance in this case is best done on the computer by enumerating all valid states and their probabilities, all blocking states and their probabilities, and then taking the ratio of the two.

As an example, let us consider the case of the two contiguous beams of Fig.3a with $N_F=2$ (1 for each beam) and $N_D=1$. There is a total of 3 frequencies and 3 channels available in the system. The state diagram for this example is shown in Fig.15. In a pure fixed assignment with one frequency for each beam we would have had blocking conditions in states 01, 11 and 10. The addition of one frequency in dynamic assignment has relieved these blocking conditions. In pure dynamic assignment with a total of 2 frequencies we would have had blocking conditions in states 02, 11 and 20. The blocking in state 11 has been relieved by the mixed assignment.

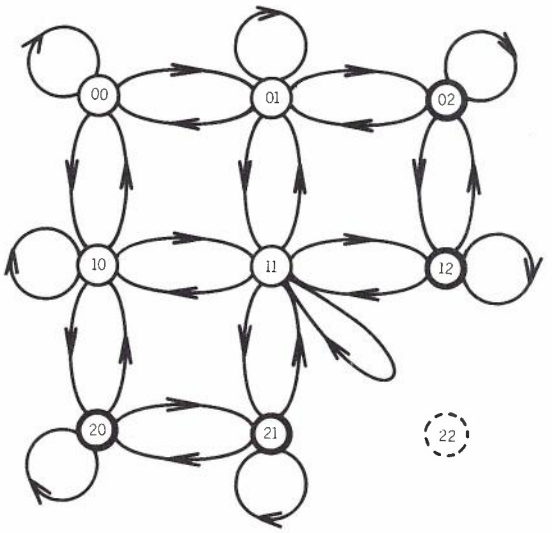


Fig. 15 - Two-Beam Example of Mixed Assignment

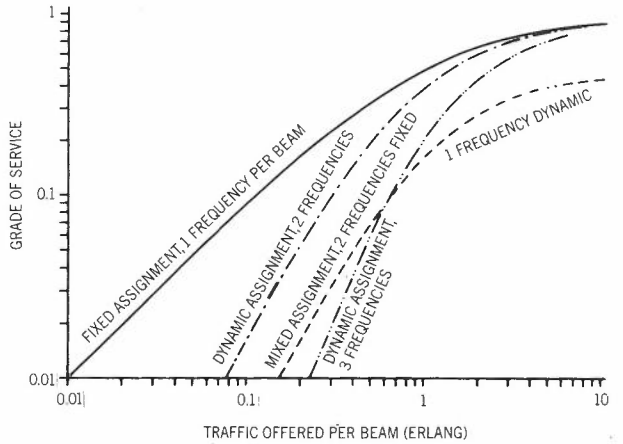


Fig. 16 - Comparison of Different Assignment Methods in Two Beams

Assuming equal traffic in each beam, the grade of service performance of this example is shown in Fig.16 together with those of fixed and dynamic assignments where a total of 2 frequencies have been allocated in each assignment. We note the great improvement over both types of assignment when mixed assignment is used. The additional frequency required by the mixed assignment has accounted for this improvement. If all three frequencies allocated to the mixed system are used instead for a three-frequency dynamic assignment, the result is a better performance at low traffic levels but a worse situation at high traffic. This effect is also well illustrated in Fig.16. All the curves in Fig.16 assume equal traffic in both beams.

10. MIXED ASSIGNMENT WITH FREQUENCY REUSE

With reference to the three-beam dynamic assignment with reuse, equations (52), (53) and (54) show that the grades of service are different for each beam, and that the grade of service is worst in the centre beam. This is accounted for when we examine the state tables such as Table 2 where we see that there is always a blocking condition in beam B whenever there is blocking in either beam A or beam C or both. One way to reduce the blocking in the centre beam B is to provide one or more additional frequencies for the sole use by this beam in fixed assignment mode. That is, we now have mixed assignment plus frequency reuse.

Let us consider the addition of one fixed-assigned frequency to beam B while one frequency is being used as in the original demand assignment with reuse. The state table now has additional entries as shown in Table 4.

TABLE 4 - Three-Beam Example of Mixed Assignment Plus Reuse

COMBINATION NUMBER	STATE A	STATE B	STATE C	PROBABILITY ($\times P_{000}$)	REMARKS
1	0	0	0	1	
2	0	0	(1)	E_C	Blocking in C
3	0	1	0	E_B	
4	0	(1	1)	E_BE_C	Blocking in B&C
	(0	(2)	0)	$E_B^2/2$	Blocking in A,B&C
	~~0~~	~~2~~	~~1~~	$E_B^2E_C/2$	Invalid
5	(1)	0	0	E_A	Blocking in A
6	(1)	0	(1)	E_AE_C	Blocking in A&C
7	(1	1)	0	E_AE_B	Blocking in A&B
8	(1	(1)	1)	$E_AE_BE_C$	Blocking in A,B&C
	~~1~~	~~2~~	~~0~~	$E_AE_B^2/2$	Invalid
	~~1~~	~~2~~	~~1~~	$E_AE_B^2E_C/2$	Invalid

Comparing Table 4 with Table 2 we see that the blocking conditions in beam B in combination numbers 2 and 5 have now been relieved without affecting the blocking conditions in beam A and C. In addition, combinations 4, 7 and 8 which were previously invalid can now be used to increase traffic capacity. The resulting

grades of service are shown by the two leftmost curves in Fig.17 for the case of equal traffic in each beam. We see that the performance in beam B is now very much improved relative to beams A and C. In fact we now have "overcompensated" for the original lack of performance in beam B.

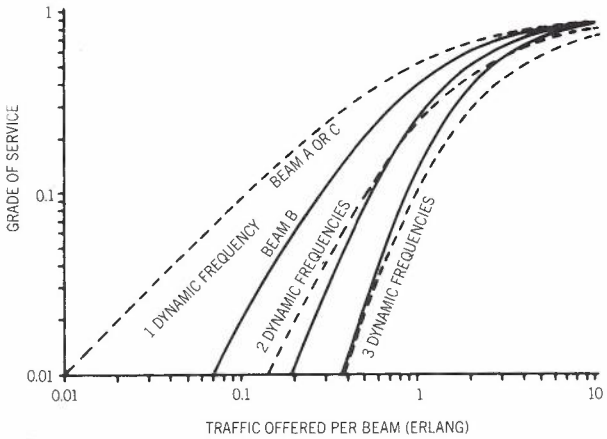


Fig. 17 - Mixed Assignment Plus Reuse in Three Beams - 1 Fixed Frequency in Central Beam

Various mixes of fixed and dynamic assignment can be tried with spectrum reuse. Certain mixes will be optimum in the sense of equalising the grades of service in the beams. For example, we may try 2 dynamic frequencies and 1 fixed frequency in beam B. The results are again shown in Fig.17 where we see that the grades of service are much more even among the beams (see the two centre curves). The use of 3 dynamic frequencies and 1 fixed frequency gives the rightmost curves in Fig.17. Here we have a case of "undercompensation" because the grade of service in beam B, although much improved relative to that of unmixed assignment, is still not sufficient to equalise the performance among the beams.

11. CONCLUSIONS

In this paper we have discussed fixed assignment, fixed assignment with frequency reuse, dynamic assignment and dynamic assignment with frequency reuse as applied to SCPC FDMA satellite systems. Simple examples have shown how these assignment schemes affect the grades of service, the traffic capacity, frequency allocation and the number of channels. There is a general order of improvement in performance from the simple fixed assignment scheme right through to dynamic assignment with reuse. The improvement can be quite spectacular when applied to practical systems such as the proposed North American and the Australian satellite systems. Although analytical expressions can be derived for the first three assignment schemes, the treatment of dynamic assignment plus reuse is done most easily by computer enumeration technique.

Further examples show that a mix of fixed and dynamic assignment with and without frequency reuse may result in improved performance over simple fixed and dynamic schemes. However, over- and undercompensation can result for certain mixes.

The choice of assignment scheme depends on the particular application. There is a general increase in satellite and ground equipment costs as we move from fixed assignment through to dynamic assignment with reuse. The benefits of using spot beams is off-set by the need for more transponders, hence the increase in system cost. Whether the increase in cost is compensated by the increase in traffic capacity is a determining factor. The scarcity of radio frequency spectrum may well force the decision in favour of the more sophisticated schemes.

When there is a large number of channels, the computer algorithm needs to be modified to avoid excessive computing times and overflow problems. Further study is being done in this direction to find analytical limits and bounds which can be used instead of the enumeration technique.

Work is also proceeding on new and old demand-assignment algorithms. It is hoped that optimum algorithms will result in the best utilisation of the spectrum at the lowest cost.

12. ACKNOWLEDGEMENTS

The author wishes to acknowledge the financial support by the Australian Radio Research Board, the International PEACESAT Consortium, the Canadian Department of Communications and the Department of Electronic and Communication Science at La Trobe University.

The work had its beginning at the Space Systems Directorate of the Communications Research Centre in Ottawa. The help and cooperation of Dr J.L. Pearce of the Military Satellites Section and W.D. Hindson of the MSAT Project is gratefully appreciated.

The reviewers offered many helpful suggestions and constructive criticism which have been included in this paper.

13. REFERENCES

1. Anderson, R.E. and Milton, R.T., "Satellite-Aided Mobile Radio Concept Study", Final Report, NASA Contract No. NAS5-15134.

2. Nowland, W.L. and Pearce, J.L., "Government of Canada Multipurpose Satellite (MUSAT) Communications Concept", CRC Internal Report July 1979.

3. Bell Laboratory Staff, "Advanced Mobile Phone Services", BSTJ, Vol. 58, No. 1, Special Issue, January 1979.

4. Bear, D., Principles of Telecommunication - Traffic Engineering, IEE Telecommunication Series 2, 1976.

5. Jakes, W.C. Jr., (Ed.), Microwave Mobile Communications, John Wiley & Sons, 1974.

6. Simpson, K.J., Sivyer, M., Sargeant, V., Boland, J.E., de Jong, H., "Telecom's Mobile Telephone Service", Telecommunication Journal of Australia, Vol. 31, No. 3, 1981, pp. 163-196.

7. Elnoubi, S., Singh, R., Gupta, S.C., "A New Channel Assignment Scheme in Land Mobile Radio Communications", Nat. Telecom. Conf., 1979, pp. 9.1.1-9.1.4.

8. Nehme, G., Mousseau, G., Michaud, A., Georganas, N.D., "A Simulation Study of High-Capacity Cellular Land-Mobile Radiocommunication Systems", Canadian Comm. & Power Conf., October 1980, pp. 421-429.

9. Hindson, W.D., Butterworth, J., Heal, J., "Communication System Concept for the 800 MHz Service on the Demonstration MSAT", CRC Internal Report, December 1980.

10. Seumahu, E.S., "Some Aspects of Demand-Assigned Multiple-Access (DAMA) for MSAT", CRC Internal Report, June 1981.

11. Satellite Project Office, Australian Postal and Telecommunications Department, "Planning for a National Communications Satellite System", Information Paper, July, 1980.

LIST OF SYMBOLS

E	- Traffic volume
E_A, E_B, E_C	- Traffic volume in beams A,B,C
E_ℓ	- Traffic volume in the ℓ^{th} beam
G	- Grade of service
G_A, G_B, G_C	- Grade of service in beams A,B,C
G_k	- Grade of service in the k^{th} beam
L	- Total number of spot beams
ℓ	- Beam number
M	- Number of channels available
M_A, M_B, M_C	- Number of channels available to beams A,B,C
M_ℓ	- Number of channels available to the ℓ^{th} beam
M_T	- Total number of channels available to the whole system
N	- Number of frequencies available
N_A, N_B	- Number of frequencies available to beams A,B
N_ℓ	- Number of frequencies available to the ℓ^{th} beam

N_T - Total number of frequencies available to the whole system

N_P - Number of frequencies available in a pool

n - Number of frequencies in actual use at a particular time

n_A, n_B - Number of frequencies in use in beams A,B

n_ℓ - Number of frequencies in use in the ℓ^{th} beam

P_B - Total blocking probability

P_{BA}, P_{BB} - Total blocking probability in beams A,B

P_{Bk} - Total blocking probability in the k^{th} beam

P_T - Total of all valid probabilities

P_n - Probability of n frequencies being used at a particular time

P_{n_A}, P_{n_B}, P_{n_C} - Probability of n_A, n_B, n_C frequencies being used in beams A,B,C

$P_{n_A\, n_B\, n_C}$ - Probability of n_A frequencies being used in beam A, at the same time that n_B frequencies are being used in beam B and n_C frequencies are being used in beam C

P_{n_ℓ} - Probability of n_ℓ frequencies being used in the ℓ^{th} beam

$P_{n_1, n_2 \ldots n_\ell \ldots n_L}$ - Probability of n_ℓ frequencies being used in the ℓ^{th} beam simultaneously with n_1 frequencies being used in the 1st beam, n_2 frequencies being used in the 2nd beam, etc.

λ - Mean number of call arrivals per second

μ - Average holding time of a call

BIOGRAPHY

E. STEFANUS SEUMAHU was born in Indonesia. He completed a special Leaving Honours course at Adelaide Boys High School in 1957. In 1961 he graduated from the University of Adelaide and the S.A.Institute of Technology with distinctions in the B.Tech. course. From 1961-63 he studied at the University of Melbourne, graduating with a B.E. degree and a M.Eng.Sc. degree with honours. Dr Seumahu obtained his Ph.D degree from Monash University in 1970. He undertook Post-Doctoral work as a Fellow of Advanced Engineering Studies at Massachusetts Institute of Technology during 1973-74. His fields of interest are Communication Theory, Communication Systems, Random Signals and Satellite Communication.

Dr Seumahu worked for a total of 3 years between 1960 and 1966 with the P.M.G. Research Laboratories in Melbourne in the Radio Propagation and Radio Systems Sections. He also worked for Garuda Indonesian Airways from 1964-65. Between 1969 and the present he has taught widely as Lecturer, Senior Lecturer and Reader at S.A.I.T., La Trobe University and the P.N.G. University of Technology. He spent his 1980/81 study leave at the Space Systems Directorate of the Communications Research Centre in Ottawa, working on the DAMA aspects of the proposed MSAT mobile satellite.

Dr Seumahu has been involved with the PEACESAT project since its beginnings in 1971. He is currently the national co-ordinator of the PEACESAT AUSTRALIA project and responsible to NASA for the use of the ATS-1 satellite in Australia. During WARC-79 he was an advisor/observer on the Australian delegation. He is on the councils of the International Peacesat Consortium and the Satellite and Telecommunications Users Association and is a member of the CCIR National Study Group 4 on Fixed Satellite Services.

Interference To Satellite Earth Stations Due To Scatter of Terrestrial Transmissions By Aircraft

J.V. MURPHY
Telecom Australia Research Laboratories

Scattering by aircraft of radiation from terrestrial transmitters is a possible source of interference to a satellite earth station (ES) sharing the same frequency band. The interference consists of a short pulse of noise. For an earth station antenna diameter of 13 m the pulse duration varies from 60-90 ms. A study to assess the likelihood of interference to a television receive-only ES at a typical commercial site in the Northern Sydney surburbs showed that burst levels of interference much greater than the thermal noise will occur due to several 4 GHz Telecom radio-relay transmitters.

The interference outage time will depend on the precise siting of the earth station. Calculations for locations near a minor commercial aircraft route, and near a reporting point for general aviation, result in interference times of 0.13 min/yr and 0.05 min/yr respectively. Interference to ES via aircraft scatter should not, therefore, be a problem unless the earth station antenna main beam intersects a final approach path of a major airport.

1. INTRODUCTION

In order to reduce the length and cost of the terrestrial circuits carrying traffic to population centres there is a developing trend of locating earth stations (ES) of the fixed satellite service in, or close to, city centres. Similar motivation for reducing the travelling time from airports leads to the placement of airports as close to cities as environmental and planning constraints allow. There exists, consequently, the increasing likelihood of interference between terrestrial radio transmitters and satellite earth stations sharing the same frequency bands. The interference mechanism concerned is the scattering, by aircraft, of power from the terrestrial transmitter into the ES receiver (Fig.1). Should the aircraft fly into or near the main beam of the ES receiver then the possibility arises of unacceptable levels of interference to the ES.

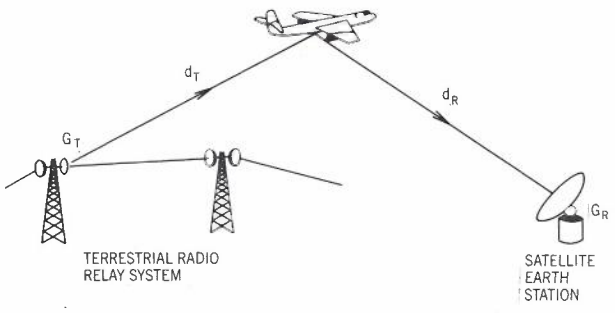


Fig. 1 - Mechanism of interference by aircraft scatter

This paper considers the specific problem of determining the interference at 4 GHz from repeaters of the Telecom microwave radio-relay network, via aircraft scatter, into a television receive-only ES located in the Sydney suburbs. In this area two earth stations for television reception are currently being constructed. The results of this study are also relevant to the case of a transmitting/receiving ES of the fixed satellite service because, in view of the high sensitivity of the ES, the dominant mode of interference is usually that from terrestrial transmitter to ES receiver.

In the calculations below imperial units are used for aircraft locations in accordance with standard aviation practice. SI units are used elsewhere. Aircraft height and flight statistics were supplied by the Department of Aviation, Sydney.

2. CRITERION OF INTERFERENCE

Interference via aircraft scatter consists of short bursts of noise which occur for small percentages of the time. The duration of each burst is the time of transit of an aircraft through the ES antenna beam. Assuming an ES antenna diameter of 13 m and a height and speed of the aircraft of 2,000 ft and 200 knots respectively, the burst duration will vary between 60 and 90 ms depending on the aircraft azimuth bearing. Each noise burst will therefore affect only a few frames of the received television signal.

For the case of international television transmission the CCIR specifies a permissible short-term baseband signal-to-noise ratio of 25 dB (Ref. 1). In this paper we therefore determine the ratio of wanted to unwanted signals, i.e. the carrier-to-interference ratio, assuming a wanted signal level of -129 dBW.

The time for which interference is permitted to occur may be estimated, in the absence of appropriate CCIR Recommendations, by consideration of the case of an ES carrying telephony traffic.

In this case high-level interference is permitted for 0.03% of the time (Ref. 2). For co-ordination purposes, the allowed time percentage per entry is 0.01% (Ref. 3). This latter outage time percentage is equivalent to 53 min/yr.

The impulse noise interference time which is permitted for television transmission should exceed that specified for telephony since the frame stores used in television reception reduce the interfering effects of noise bursts. On detecting loss of synch, the frame store repeatedly outputs the previous uncorrupted frame. The perceptible effect to the observer is a freezing of any motion in the picture during the noise burst. Since bursts of interference due to aircraft scatter last for only 2-3 frames, the perceptible degradation will not be serious.

3. PROPAGATION OF INTERFERENCE

Detailed calculations show that, of all possible aircraft positions, significant interference arises only when the aircraft flies through the main lobe of the ES antenna pattern. Interference due to scatter from aircraft in the main beams of the nearer terrestrial transmitters cannot occur because their main beams are assumed to be below the minimum permissible aircraft height in the vicinity of the ES.

Because of the orientation of the radio-relay and ES antenna beams, the scattering aircraft will not in general be on the great circle between the transmitter and receiver. The microwave signal scattered from the aircraft therefore consists of the sum of many components arising from the major surfaces and edges of the aircraft. The interfering signal level is highly variable, depending both on frequency and on aircraft orientation relative to the transmitter and receiver.

At high altitudes, because of the incoherence of the scattered signal, the received interference power, P_I, is proportional to the radar cross-section, σ, of the aircraft and is determined using the bistatic radar equation, thus:

$$P_I = P_T + G_T + G_R - L_b(d_T) - L_b(d_R) + 10 \log_{10} \frac{4\pi\sigma}{\lambda^2} - B \text{ dBW} \quad (1)$$

where P_T = transmitter power (+10 dBW assumed),

G_T, G_R = transmitter and receiver (ES) antenna gains, respectively, relative to isotropic (dB),

d_T, d_R = ranges from terrestrial radio transmitter to aircraft and from aircraft to ES, respectively,

$L_b(d_T)$, $L_b(d_R)$ = basic transmission losses (dB),

λ = wavelength (= 0.075 m at 4 GHz) and

B = factor (dB) to allow for blocking by the terrain of the transmitter beam, which is assumed to be horizontal.

Values of radar cross-section for various types of aircraft are given in Ref. 4. A value of 60 m^2 at S-band is typical of the larger aircraft and is consistent with the value of median geometrical area at 19 GHz measured in Ref. 5. For the assumed ES antenna diameter at 60% efficiency the gain G_R = 52 dB. Substituting these values into equation (1) the ratio of wanted to interference power is given by

$$(P_W/P_I)_1 = 20 \log (d_T d_R) - G_T - 30 \text{ dB} \quad (2)$$

where a value of 3 dB for B has been assumed.

For low flying aircraft (e.g. the general aviation category to be discussed below) the aircraft may completely fill the ES antenna beam. The received interference power can be calculated by considering the reciprocal situation where the ES transmits power $\eta_R P_T$, η_R being the ES antenna efficiency. This power is radiated isotropically by the aircraft, i.e. the aircraft appears as an isotropic antenna whose input power is $\eta_R P_T$. The received power is then:

$$P_I = 10 \log_{10} \eta_R + P_T + G_T - L_b(d_T) - B \text{ dBW} \quad (3)$$

and the ratio of wanted to interference power in the near scatterer case can be found as

$$(P_W/P_I)_2 = 20 \log d_T - G_T - 29 \text{ dB}. \quad (4)$$

For an ES diameter of 13 m, this formula applies for $d_R < 1.25$ km.

4. GEOMETRY OF BEAM INTERSECTIONS

The interference potential of a particular repeater will depend on the separation (approximately equal to d_T) between it and the ES and the angular separation between its boresight direction and the azimuthal bearing to the ES. For indicative calculations we assume that the height relative to the ES and the angle of elevation of the terrestrial transmitter antenna are zero. Under conditions of normal refraction the sidelobes of the terrestrial antenna then illuminate the ES main beam above a certain height, h, where

$$h = d_T^2/2a,$$

and a = modified earth's radius (8500 km). The distance, d_R, from the scattering aircraft to the ES is then

$$d_R = h'/\sin \phi$$

where h' is the greater of h and the minimum permitted aircraft height, taken as 1000 ft, in the area; ϕ is the angle of elevation of the

earth station antenna beam. For the two satellite positions 174°E and 179°E, the average elevation angle is 42.2°.

5. CALCULATION OF INTERFERENCE LEVELS

Because Sydney is a node of the Telecom radio-relay network, earth stations in the area are likely to suffer interference from a relatively large number of radio-relay transmitters on routes converging on the node. The actual location of the ES will determine the potential interfering sources as those 4 GHz transmitters in the national Telecom radio-relay network whose azimuth line-of-shoot passes sufficiently close to the earth station to render interference likely.

In this section we calculate the interference levels for a typical ES in the North Sydney area. The transmitters likely to cause interference are shown in Fig.2. For each of the interfering paths listed in Table 1, the angle, ψ, between the Telecom transmitter boresight direction and the azimuth at which, looking from the transmitter, the ES main beam is visible above the radio horizon is estimated from the site co-ordinates. This angle determines the interfering transmitter antenna gain, G_T, assuming the average power off main beam is 3 dB below the radiation pattern envelope. A 10 ft transmitter antenna is assumed. The ratio of wanted to interfering signals is calculated from equation (1) if $d_R > 1.25$ km and from equation (4) otherwise. The results are given in Table 1.

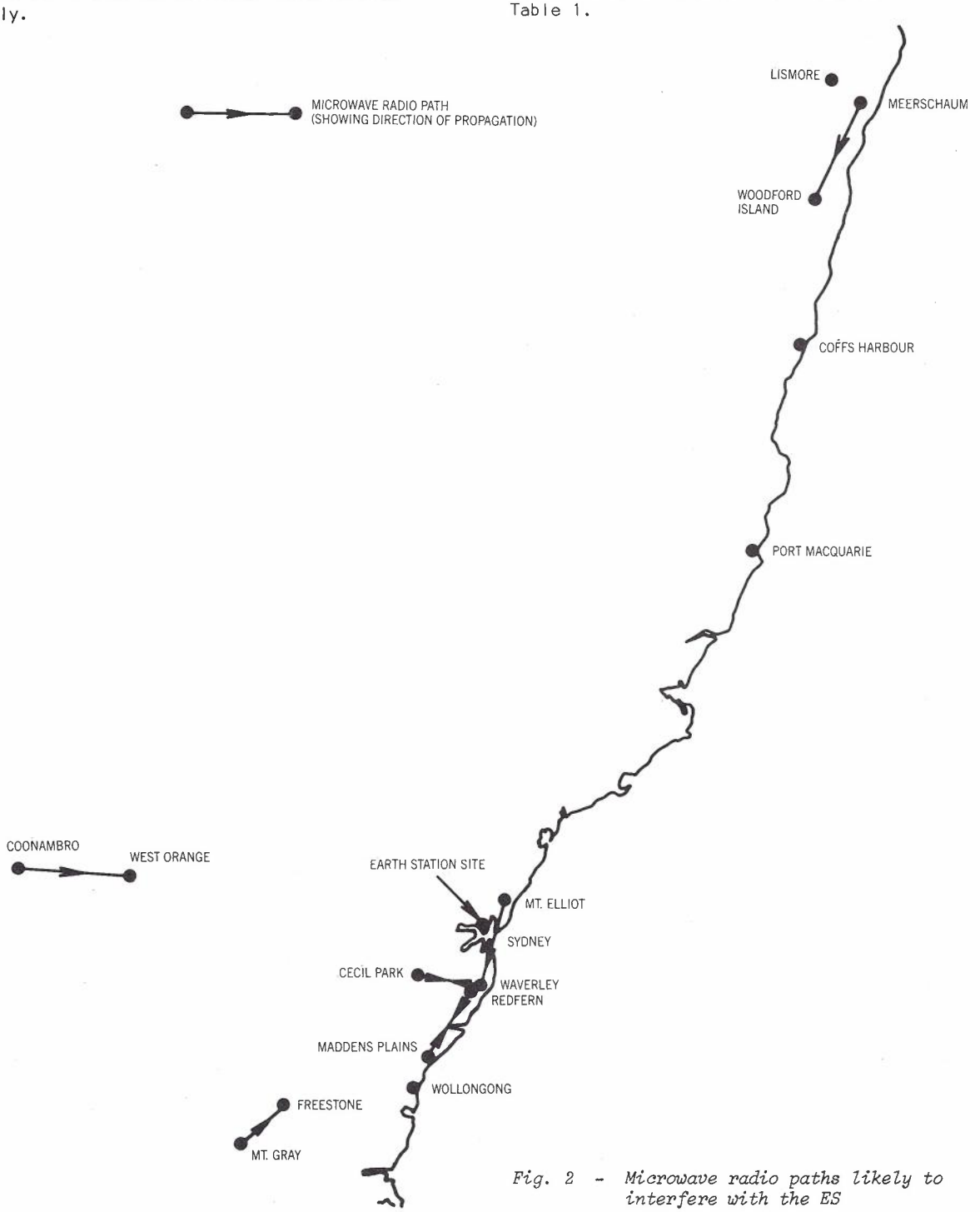


Fig. 2 - Microwave radio paths likely to interfere with the ES

TABLE 1 - Ratio of Wanted to Interfering Signals for Predominant Interfering Repeaters

Path	d_T (km)	h (km)	d_R (km)	ψ (deg)	G_T (dB)	$(P_W/P_I)_1$ (dB)	$(P_W/P_I)_2$ (dB)
Mt. Elliot - Waverley	49	0.14	0.45	8.5	8	-	-3
Meerschaum - Woodford Is.	581	20	29.5	0.8	37	18	-
Coonambro - West Orange	273	4.4	6.5	3.3	14	21	-
Mt Gray - Freestone	169	1.7	2.5	18.6	0	23	-
Cecil Park - Waverley	35	.072	0.45	16	0	-	2
Waverley - Cecil Park	11	-	0.45	80	-8	-	0
Maddens Plains - Redfern	55	0.18	0.45	5	12	-	-6
Redfern - Maddens Plains	9	-	0.45	160	-26	-	16

The low values of wanted signal to interference ratio show that severe interference bursts will be caused by the passage of an aircraft through the ES main beam.

6. FLIGHT GEOMETRY AND STATISTICS

In this section we calculate the fraction of the time that an aircraft is expected to take in passing through the main lobe of the ES antenna, and therefore the total fraction of the time that the interference exceeds the permissible level. This time depends strongly on the location of the ES relative to the main air routes of regular commercial aircraft and to concentration points of general aviation aircraft. Unfavourable cases are considered in order to derive pessimistic estimates of the outage time for typical sites in the suburban area. These include the cases of intersection of the ES beam with a commercial aircraft track and the location of the ES near a general aviation reporting point where the traffic density is higher than in the overall airspace.

Two other categories (miscellaneous aircraft including helicopters, and military aircraft) are also considered.

6.1 Commercial Aircraft

Commercial aircraft on take-off tend to head in particular directions depending on the runway used and the destination. Each departure bearing has a tolerance of $\pm 10^{o}$ ±1 nautical mile (nmi), but in practice aircraft headings are held to a few degrees. The aircraft is therefore assumed to pass through a rectangular window whose dimensions are determined by the height and azimuth tolerances of the aircraft trajectory (see Fig.3 and Appendix I).

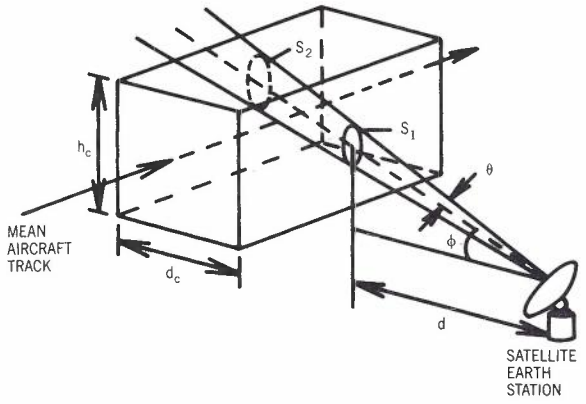


Fig. 3 - Intersection of earth station beam and trajectory cell for commercial flights

A typical unfavourable situation involves the intersection of the ES beam and a track flown by commercial aircraft on a minor route. In this case the total interference time per annum is given by equation (A3), where the tolerances d_c and h_c in direction and height are assumed to be 7500 ft and 15,000 ft respectively. The former dimension corresponds to an error in heading of $\pm 3^{o}$ at a radius of 12 nmi from the airport. The vertical tolerance is the difference between the maximum attainable ceiling and the minimum allowed height (5000') at the same radius. For the ES antenna diameter of 13 m the semi-beamwidth, θ, is 0.2^{o}. An ES situated 1 nmi from the track would then experience an average outage of 2.5×10^{-5} minutes per flight for an aircraft speed of 250 kt. If the flight frequency is 100 flights/week the total outage time per annum is calculated as 0.13 min.

6.2 General Aviation

In contrast to commercial aircraft, general aviation aircraft tend to be randomly distributed within a particular broadly defined area. Concentrations tend to occur around reporting points and near prominent terrain features, landmarks etc. The probability of interference is calculated using the random track geometry discussed in Appendix II.

Apart from the direct runway approaches, the greatest aircraft density in the suburban area occurs near reporting points. This concentration is modelled by assuming that aircraft are uniformly distributed throughout a high-density cylindrical cell around the reporting point (Fig.4). In this high-density cell aircraft are restrained between heights of h_ℓ = 1500 ft and h_u = 2000 ft. Taking the radius, R, of the cell as 2 nmi the fraction of this volume intersected by the ES antenna beam is obtained from equation (A6) as 8.4×10^{-7}. The average path length through the high-density cell is easily shown to be $\pi R/2$. At an average speed of 150 kt, each aircraft takes 1.3 min to pass through the cell. The number of flights per month from Bankstown airport is about 25,000, of which 4000 may be assumed to pass through the vicinity of the reporting point. Using equation (A7) the total time per annum for which interference occurs is calculated as t_2 = 0.05 min.

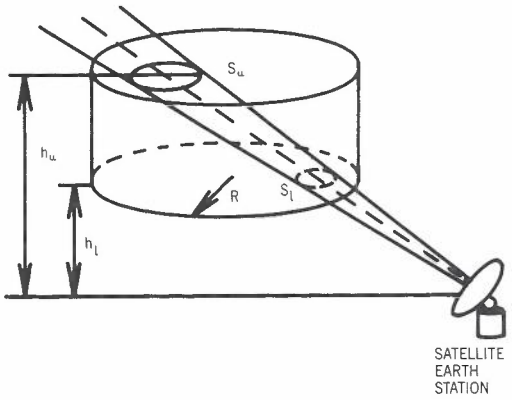


Fig. 4 - Intersection of ES beam and high-density cell for random flights

6.3 Miscellaneous Light Aircraft

This category includes helicopters (used either by the media, by the police, or for joy flights), float planes etc. The statistics of these flights are particularly ill-defined and the tracks are more random than those of light aircraft. They tend to operate at low altitudes between 1000 and 2000 ft over a cylindrical cell whose area is comparable with that of the suburban area.

The geometry applicable to the calculation of interference time is that shown in Fig.4; the method of Appendix II can be used to determine the fraction of this volume contained in the ES antenna beam (see equation (A6)). The flight frequency is 3350/mth and the duration would be comparable with the general aviation category. The radius of the high-density cell is, however, much larger than the value assumed for a reporting point for general aviation; the outage time due to this category can therefore be neglected in comparison.

6.4 Military Aircraft

The situation with military aircraft is broadly similar to that described in the previous sub-section. These aircraft tend to cover a large area of assumed radius 100 nmi at high altitudes - between 24,000 and 55,000 ft. The statistics of flight occurrences tend to vary with the time of year and to depend on political and strategic factors. If an average flying time of 2 h/week for each of 14 fighter aircraft is assumed, the large value of R in equation (A7) results in negligible total outage time.

7. DISCUSSION

Calculations show that significant levels of interference power are received due to aircraft scatter only when the aircraft flies through the main beam of the ES antenna. Although the levels are high, the duration of these events is sufficiently short to cause insignificant picture degradation if frame stores are used by the ES operators. The overall outage time is found to be much less than that permitted in the case of telephony for typical unfavourable situations.

Only on the approach path to major airports would the traffic density be sufficiently high to result in unacceptable interference times should intersection with the ES antenna beam occur. This situation does not occur for the 4 GHz earth stations being implemented in the Sydney area at the present time. The method of Appendix I can be used to evaluate this situation, which can be easily identified in the preliminary stages of planning the ES location.

The increasing tendency for outbound commercial aircraft to be directed onto various tracks according to traffic density and meteorological conditions is an advantage in that it reduces the concentration of aircraft on particular routes which may be intersected by the ES main beam.

The general locations being considered for the new airport at Sydney are sufficiently distant from the earth stations currently being constructed to render intersection of the antenna beams and the approach path unlikely.

8. ACKNOWLEDGEMENTS

The author wishes to thank Mr M. McNaught for providing details of the site geometry and the estimate of wanted signal level and Mr C. McCurley for providing the data on aircraft heights and flight statistics which have been quoted in this paper.

9. REFERENCES

1. CCIR Recommendation 567, Section D.3.2.4.

2. CCIR Recommendation 356.

3. CCIR Recommendation 382.

4. "Pulse Radar Range", Selenia Technical Monograph.

5. "Reflections from aircraft observed on a great-circle transhorizon path", CCIR Doc. 5/305, (FRG), 26.5.1981.

APPENDIX I

PROBABILITY OF SCATTERING BY AIRCRAFT ON REGULAR TRACKS

For these types of flights, the highest aircraft density will occur in a box-shaped cell of arbitrary length ℓ (Fig.3). The width, d_c, is dependent on the variation in aircraft heading. The height of the cell, h_c, depends on the altitude range in which the aircraft is expected to be located.

In practice the aircraft trajectories tend to be concentrated about a mean trajectory with particular distributions in the vertical and horizontal dimensions. Because of the uncertainty in the cell dimensions and the extra computational complexity in evaluating the exact distributions, we shall assume for simplicity that aircraft are uniformly distributed within the cell.

The interference probability depends on the likelihood of an aircraft passing through the ES beam. If the azimuths of the ES beam and the aircraft are perpendicular and the distance from the ES to the aircraft track is d, then the volume, V, of the beam which intersects the trajectory box is the difference between the two cones whose bases are S_1 and S_2. The volume of the former is given by

$$V_1 = \frac{1}{3} S_1 d$$

where

$$S_1 = \pi \left(\frac{d}{\cos \phi}\right)^2 \theta^2/\cos \phi$$

and θ is the semi-beamwidth of the ES antenna. We have therefore:

$$V_1 = \frac{\pi}{3} \frac{d^3\theta^2}{\cos^3\phi}$$

similarly

$$V_2 = \frac{\pi}{3} \frac{(d+d_c)^3\theta^2}{\cos^3\phi} .$$

Hence the required volume is

$$V = V_2 - V_1,$$

i.e.,

$$V = \frac{\pi}{3} \frac{\theta^2}{\cos^3\phi} \left[(d+d_c)^3 - d^3 \right] . \qquad (A1)$$

Let the aircraft speed be v kt. Then for a trajectory box of length ℓ nmi, the time spent in the box will be ℓ/v h. The time spent in the ES antenna beam will then be

$$t_1 = \rho_1 \frac{\ell}{v} h \qquad (A2)$$

where

$$\rho_1 = \frac{\pi}{3} \frac{\theta^2}{\cos^3\phi} \frac{(d+d_c)^3 - d^3}{\ell\, d_c h_c}$$

is the ratio of the volume of the beam segment to that of the box. Substituting ρ_1 into (A2) we obtain the following expression for the interference time:

$$t_1 = 20\pi \frac{\theta^2}{\cos^3\phi} \frac{(d+d_c)^3 - d^3}{v d_c h_c} \text{ min} . \qquad (A3)$$

APPENDIX II

PROBABILITY OF SCATTERING BY AIRCRAFT ON RANDOM TRACKS

Aircraft in uncontrolled areas are assumed to be uniformly distributed throughout a defined cell taken as a right cylinder of radius R whose axial dimension is the difference in the minimum and maximum heights, h_ℓ and h_u respectively. The geometry is then as shown in Fig.4.

The volume in which scattering is important is the intersection of the conical ES beam of semi-beamwidth θ, with the cylindrical cell. This volume is the difference between the volumes of the two inverted slant cones whose bases are the areas S_u and S_ℓ. Considering first the upper cone, the distance, r, from the ES to the top of the cylinder is given by

$$r = h_u/\sin \phi \qquad (A4)$$

and the area of the base of the cone is

$$S_u = \pi r^2 \theta^2/\sin \phi . \qquad (A5)$$

Substituting (A4) into (A5) we obtain the volume of the larger cone as

$$V_u = \frac{\pi}{3} \frac{\theta^2}{\sin^3\phi} h_u^3 .$$

A similar expression holds for the volume, V_ℓ, of the cone based on S_ℓ, so that the difference in the two volumes is

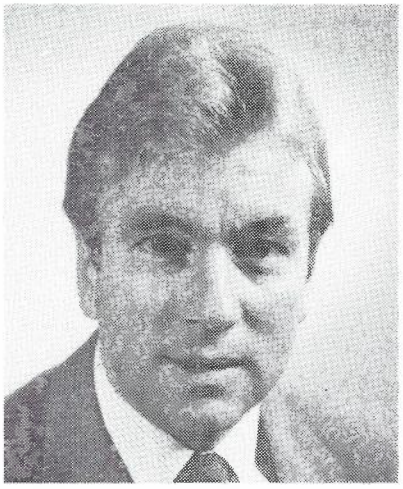
$$V = \frac{\pi}{3} \frac{\theta^2}{\sin^3\phi} (h_u^3 - h_\ell^3)$$

and the ratio of the intersection to the cell volume is

$$\rho_2 = \frac{\theta^2}{3R^2 \sin^3\phi} \cdot \frac{h_u^3 - h_\ell^3}{h_u - h_\ell} \quad . \qquad (A6)$$

If the average aircraft time in the cell is T, the average time, t_2, spent in the ES main beam will be $\rho_2 T$, thus

$$t_2 = \frac{\theta^2 T}{3R^2 \sin^3\phi} \cdot \frac{h_u^3 - h_\ell^3}{h_u - h_\ell} \quad . \qquad (A7)$$



BIOGRAPHY

JOHN VINCENT MURPHY was born in Sydney in 1941. He obtained the B.E. (Elec.)(Hons.) degree in 1961 and the B.A. degree in 1966, from the University of Melbourne. After graduation, Mr Murphy joined the Telecom Australia Research Laboratories, where he has been engaged in the development of high speed solid-state equipment for the statistical analysis of television signals, the measurement of the frequency of noisy signals using phase-controlled oscillators and the investigation of microwave radio. He was a consultant to the Systems Analysis Group of Telettra S.p.A., Milan and was also employed with the British Telecom Research Department where he was engaged in the planning of a large-scale wideband millimeter wave terrestrial propagation experiment. The study included the effects of anomalous propagation mechanisms on high-capacity digital radio systems.

After a period on the staff at LaTrobe University, he is now responsible for the microwave radio propagation research activities of the Antennas and Propagation Section.

The Concurrent Processing Features Of The CCITT Language CHILL

J.L. KEEDY

Department of Computer Science
Monash University

With the increasing use of stored program control (SPC) exchanges, programming is becoming an important part of telephone technology. Software systems for SPC have become very large and complex. The High Level Programming Language, CHILL, has been designed by the International Telegraph and Telephone Consultative Committee (CCITT) as a tool for building such systems. This paper examines the features of the language which enable the user to specify co-operating processes which can execute concurrently. These features are evaluated from the viewpoints of both their potential use in applications programs and the likely complexity of an operating system to support their implementation.

1. INTRODUCTION

The International Telegraph and Telephone Consultative Committee (CCITT) has developed a high level programming language called CHILL, which is defined in CCITT Recommendation Z.200, published by CCITT as "CHILL Language Definition" in May 1980 and adopted at the Plenary Assembly in November 1980.

CHILL is intended primarily for use in SPC telephone exchanges, with possible applications for other switching and systems programming such as packet switching. The requirements set when designing the language were to:

- enhance reliability by allowing for extensive compile-time checking
- permit the generation of highly efficient object code
- be flexible and powerful in order to cover the required range of applications and to exploit various kinds of hardware
- be easy to learn and use.

To these ends, CHILL provides a number of basic constructs with alternative mechanisms provided for particular purposes. This paper examines the various constructs provided in CHILL for handling concurrent processing i.e. the parallelism of execution of various processes in a computer system. The paper refers to the CHILL Language Definition, in particular to sections 1.8, 3.8, 3.9, 5.2.17, 5.2.18, 5.2.19, 6.12, 6.13, 6.14, 6.15, 6.16, 6.17, 6.18, 6.19, 7.5, 7.7, 8 (all), Appendix D examples 13-16, Appendix E, and appropriate other sections cross-referenced from these sections. Reference has also been made to the CCITT document "Introduction to CHILL", May 1980, in particular section 4.5, but not as a definitive reference for the CHILL language. References in the text to the Language Definition appear in the form L.D.1.8, L.D.3.8, etc.

The paper has the objective of evaluating the concurrent processing features of CHILL from the viewpoints both of their potential use in applications programs and of the likely complexity of an operating system to support their implementation. In preparation for the subsequent analysis and discussion, section 1 briefly reviews the main features of CHILL which support concurrent processing, viz. processes, regions, events, signals and buffers.

Section 2 examines CHILL processes in more detail and points out that for a proper appreciation of synchronisation facilities it is necessary to distinguish between two different models for decomposing a system into processes. Other issues discussed are the absence of a notion of process priority in CHILL and the relationship between the CHILL 'imaginary outermost process' and other processes.

Section 3 outlines the main areas of interest from the viewpoint of synchronisation, viz. exclusion, resource allocation and inter-process communication. Guided by the examples in Appendix D of the CHILL Language Definition, the report relates these three major areas of synchronisation to the concurrency mechanisms provided in CHILL.

Section 4 isolates some of the major problems, including syntactic inconsistencies, deadlock risks and arbitrary limitations on the level of parallelism that can be achieved.

Section 5 discusses the cost of and problems associated with the implementation (in either an operating system or a run-time language package) of the CHILL synchronisation mechanisms. The various hardware-level synchronisation mechanisms found in typical computer systems are described and their use as a basis for implementing the CHILL features is discussed. Specific implementation problems, especially associated with regions, are then described.

Section 6 considers some areas where expansion of the CHILL synchronisation features might be undertaken, in particular to support input-output and synchronisation with time, as well as for distributed processing environments. Some comments are then offered on selecting which facilities should be considered as reasonable base mechanisms for such expansions.

Section 7 suggests guidelines for using CHILL, arguing that a uniform philosophy should be adopted in the decomposition of a system or program into processes. Specific guidelines are then recommended for each of the two process models described in section 2. The use of regions and of events is entirely avoided because of their associated problems, and it is demonstrated how signals will simply and effectively support the needs of one model, while buffers are sufficient to support the second model.

1.1 Overview of CHILL Concurrency Features

"Concurrency" in the context of programming languages and operating systems, refers to the parallelism which can actually or notionally occur among computations executing in the same computer system or in linked computer systems. To achieve actual parallelism requires that the system contains two or more processors which can execute simultaneously. In the restricted sense that input-output devices (e.g. disc units, printers) are independent processors, almost all modern systems are concurrent; there are also some systems which include several central processing units (CPUs), thus allowing calculations to be carried out simultaneously with each other as well as with input-output operations. But even where there is only one CPU, it is usually beneficial to execute several tasks concurrently in order to maximise the use of this CPU. In this case notional parallelism or quasi-parallelism is achieved by multiplexing the single CPU among the various tasks. The algorithm used for determining when tasks are actually executed by the CPU (or CPUs) will be known in this report as the 'CPU scheduling algorithm'.

In a computer system which supports concurrent execution of tasks, these tasks must co-operate with each other in various ways which cannot be precisely defined nor predicted (with respect to time) when the programs for the tasks are developed. At the very least co-operation will be needed regarding the allocation of scarce resources (e.g. a printer), and some form of synchronisation mechanism will be needed to allow implicit or explicit inter-process communication. The definition and implementation of mechanisms to achieve harmonious co-operation is a non-trivial task. It has been the subject of (literally) hundreds of academic papers and books; even now, after 15 years of sustained research effort, there is no universally accepted set of mechanisms. Appropriate solutions must be found at three different levels in any system; in the hardware, in the operating system and in the programming language(s).

In evaluating the concurrency features of the CHILL programming language it is therefore necessary to consider not only their inherent suitability as constructs for developing programs, but also the ease with which they can be mapped onto more primitive synchronisation mechanisms which can be expected to be found in the hardware and in the operating system. In fact the CHILL documents make no reference to the requirements of the underlying hardware or operating system, and indeed it is not clear whether its designers regard CHILL as a suitable language for developing operating systems (or for application programs which run without an operating system - in which case they must be able to solve those synchronisation problems typically handled by the operating system).

In the following subsections the main concurrent processing features of CHILL are informally introduced as a preparation for more detailed analysis in the following sections.

1.1.1 The PROCESS Concept

CHILL introduces the possibility of concurrency into the definition of a program by means of the "process" concept. A process is a unit of concurrency, in the sense that several processes may be executing different (or even the same) statements in a CHILL program concurrently. The parallelism may be notional or actual, i.e. the programmer is unaware of the actual number of CPUs in the system. The program consists of one 'imaginary outermost process' known hereafter as the 'outer process', which is started implicitly when the program starts, and as many additional processes as the programmer explicitly causes to be created. Several processes may use the same PROCESS definition. A process terminates either when it reaches the end of its definition or when it executes a stop statement. Each process is identified by an INSTANCE value which is supplied when the process is created and which can be later obtained by the THIS operator.

1.1.2 The REGION Construct

When concurrent processes can access a set of common variables (variables are called 'locations' in CHILL), it may be necessary to ensure that they do so in a controlled manner, to ensure that they do not produce inconsistent values nor create other undesirable side-effects. CHILL provides a construct known as a REGION to assist in this situation. A region is a form of module which is defined such that only one process at a time can be active within it (and therefore access to its variables is mutually exclusive). Synchronisation to support this is provided by the language implementation (or operating system), not by the programmer, except that he declares the module to be a region.

1.1.3 The EVENT Construct

CHILL processes can be explicitly synchronised by the use of EVENT variables. An event is declared in a similar manner to

other variables, and can be thought of as a process queue. However the programmer does not need to explicitly program the queue operations; instead he uses the pair of actions DELAY and CONTINUE, the implementation of which is supported at run time by the language (or operating system).

When declaring an event the programmer can optionally specify the maximum number of processes which can be concurrently* delayed on the event. If more than this number of processes attempt to delay, an exception condition is raised at run time.

When a process delays itself it may optionally specify a relative priority which will be used when a CONTINUE call occurs, to determine whether it or some other process should be reactivated. Where several processes of the same highest priority are delayed, the implementation determines which process will be continued.

A CONTINUE operation wakes up at most one process. If it occurs when no process is delayed, it is equivalent to a non-operation, i.e. it is not "remembered".

A DELAY CASE statement allows a process to delay on the union of several events. A CONTINUE for any one of the specified events can cause the delayed process to be re-activated. A mechanism is available to allow a process issuing a DELAY CASE to determine by which process it was reactivated.

Notice that no explicit messages can be passed between processes via event calls.

1.1.4 The SIGNAL Construct

Signals provide a means of communication between the processes of a CHILL program. The definition of a signal optionally includes the specification of a message which will be sent with each actual transmission of a signal, as well as an optional name of the process type which can receive the signal. But a signal can be defined with no message and/or with an unspecified receiving process type.

To transmit a signal the calling process executes a SEND statement specifying the appropriate signal and if necessary supplies the values of the associated message. It can also optionally specify which process instance can receive the signal and it can specify a priority.

A signal and its message are received via a RECEIVE CASE action which can specify that any one of several signals can be accepted. The recipient must belong to the process type (if any) specified in the signal declaration and must be the process instance (if any) specified by the sender. The recipient can optionally also receive the instance name of the sender.

If more than one appropriate signal is outstanding the signal with the highest priority (as supplied with the SEND operation) will be selected. If no signal is outstanding the recipient will be delayed for the receipt of a signal unless the ELSE option is specified, in which case the process will continue at the ELSE action list.

1.1.5 The BUFFER Construct

As with signals, the CHILL buffer mode allows the processes of a CHILL program to synchronise and communicate with each other. The declaration of a buffer includes a definition of the mode of a buffer element and optionally the number of elements in the buffer. (The default is not defined; L.D.8.3 suggests that it is infinity.)

The actions which can be performed on a buffer variable are SEND, RECEIVE and RECEIVE CASE. The SEND action is semantically and syntactically similar to the signal SEND, except that a value (i.e. message) must be supplied, and the sender cannot designate the instance of the receiver. (Likewise in declaring a buffer the programmer cannot specify the type of the receiving process(es).) The sender can optionally supply a priority for the message. Unlike the signal case, if no space is available to insert the message into a buffer element, the sender will be delayed.

RECEIVE CASE is similar to that used for signals. Several buffers can be specified with different action lists, and the receiver of a message can learn the identity of the sender. If more than one message can be received the one with the highest priority will be selected. If no messages are available the receiver will be delayed pending the arrival of a suitable message unless an ELSE clause was supplied, in which case processing will continue at the ELSE action list.

A simpler method of receiving a buffer message is the RECEIVE expression, which specifies a buffer, and returns a message from the buffer. If the buffer is empty the process is delayed pending the arrival of a message. (There is no corresponding construct for receiving a signal.)

1.1.6 Summary

CHILL provides the basic features necessary to support concurrent processing - a notion of processes and a set of synchronisation mechanisms to enable them to co-operate with each other. In the next section the notion of a process is examined in more detail, followed in section 3 by an analysis of the various synchronisation mechanisms. Subsequent sections consider the implementation and use of these features in greater detail.

---

* Actually L.D.6.16 does not unambiguously refer to a concurrent number of delayed processes. However, the interpretation that only a specified number of processes can be delayed throughout the lifetime of the event is rejected as improbable.

2. CHILL PROCESSES

The concept 'process' is well known and appears widely in the literature on concurrency and on operating systems. This section examines the CHILL view of processes with reference to other notions of processes. Some standard definitions of processes are given, followed by a discussion of the two basic models used in decomposing a system or program into processes. Then follow discussions of the significance of the notion of process priority and its absence from CHILL, and of the relationship between a CHILL program's outer process and its dynamically created processes.

2.1 Some Definitions of Processes

There is no widely agreed formal definition of the word process as applied to computer software. The following is a representative sample of informal definitions:

> "a program in execution by a pseudo-processor" where "a pseudo-processor is not necessarily one implemented in hardware, but rather a composite processor made up of the hardware and the programs of the system in which the process is executed" (Ref. 1)
>
> "the locus of control within an instruction sequence" (Ref. 2)
>
> "a sequence of program states" (Ref. 3)
>
> "a computation in which the operations are carried out strictly one at a time" (Ref. 4)
>
> "A program or an algorithm is defined as 'rules of behaviour'; equipment able to follow such rules is called a computer; what happens during such process execution is called a 'process'" (Ref. 5).

Each of these definitions tries to characterise a process in terms of the execution of a program, the instructions of which are carried out strictly sequentially. CHILL processes fall within these definitions: they are entities corresponding to the sequential execution by a pseudo-processor of an instruction sequence or algorithm.

2.2 Two Process Structuring Models

Much of the literature on processes is concerned with the worm's eye view of synchronisation, i.e. the mechanisms needed to support processes as defined in section 2.1. Only very recently has the more fundamental question been raised of how problems are mapped onto processes. It has now become evident that there are two separate basic models which can be used to decompose a concurrent program or system into processes and that these two models use different techniques for synchronising processes with each other. Most programs and systems in the past have tended to combine features of both models in ad hoc ways, with the result that parallel programs are usually obscure and difficult to understand, using a multiplicity of synchronising mechanisms in apparently arbitrary ways. The concurrent processing constructs of CHILL are representative of this confusion.

In order to introduce clarity and consistency into the following discussion of CHILL's concurrency features (and hopefully into programs written in CHILL), it is first necessary to understand these two process structuring models.

The differences between the models revolve largely around their different views of the relationship between processes and program modules (i.e. textually separate program sub-units which may or may not be compiled independently). How does a process executing in one module enlist the services supplied by a second module?

In the in-process or procedure-oriented model, a process requests the services of some other module by making a procedure call. In this case the CPU scheduling algorithm is not invoked, so that the services of the called module can be described as being supplied in the calling process (hence the term 'in-process'). Provided that a switch of protection domain can accompany a procedure call this technique, which is commonly used to provide library services, can be extended to include the provision of operating system services. An operating system designed in this way need in principle have no separate system processes although in practice a few 'independent' activities such as job scheduling and spooling are usually implemented as processes (Refs. 6,7).

In the out-of-process or message-oriented model, each module is a separate process, so that if a module needs the services of a second module it sends a message to the second module's process with a request for the service. In this case provision of the service involves a switch of process and therefore activation of the CPU scheduler. Because the service is provided not in, but out of, the calling process, the term 'out-of-process' is used to describe this technique. It was widely used in the design of older operating systems, where each module or function of the system was implemented as an independent process.

Lauer and Needham (Ref. 8) have demonstrated that these different techniques can be regarded as duals of each other, in the sense that in principle a system designed in one way can with relatively minor transformations be converted into an equivalent system of the other type. (In practice this will not always be possible if certain conventions assumed by Lauer and Needham are not followed.)

Keedy (Ref. 9) has shown that despite their functional duality the models have different dynamic properties and that in general the in-process model will be more efficient (e.g. because of the reduced

number of process switches) and can be programmed to achieve greater parallelism.

The two models rely on different techniques to synchronise processes. Because procedure calls are not normally synchronised but allow the calling process to proceed directly into the called module, it is possible in an in-process system that two or more processes can be active concurrently in one module. Since their concurrent access to shared variables in many cases can produce inconsistent results an exclusion mechanism must be available to ensure harmonious co-operation in each module. The classical method of achieving this is binary semaphores (Refs. 5,10), which can be applied exactly at the critical sections in the code. Hoare (Ref. 11) and Brinch Hansen (Ref. 12) have proposed as an alternative that exclusion is applied as part of the procedure calling mechanism, naming modules with this property 'monitors'. CHILL regions appear to have been based on the monitor concept, which unfortunately has many shortcomings (Refs. 13,14). In addition to an exclusion mechanism a general mechanism for suspending and activating processes is needed to support more complex resource allocation requirements. Notice that an explicit message-passing system is unnecessary since processes can communicate implicitly via shared variables.

With the out-of-process technique an exclusion mechanism is unnecessary provided that data structures (other than parameters/messages) are not shared between modules. For resource allocation, which should be organised within a separate module or modules, some mechanism for suspending and activating processes is required. But the main synchronisation mechanism of an out-of-process system is that for passing messages between processes. The CHILL buffer construct can be considered as such a mechanism.

2.3 Process Priorities

In systems where several processes share one or more real CPUs in a multiplexed manner the CPU scheduling algorithm of the operating system must make decisions about when processes are allowed to execute. In timesharing systems it is usual to use a 'round-robin' algorithm which allows all the processes to run in turn for a limited time; this ensures a fair response to all the terminal users. But in most environments it is desirable to allocate relative priorities to different processes, and to take these priorities into account in the CPU scheduling algorithm. The reason for this is not only to reflect the users' views of the urgency of their jobs, but more importantly it can contribute significantly to the balanced use of system resources.

This can be illustrated by the simple example of a system with one CPU, a console, a card reader and a line printer. Suppose that there are two jobs, an 'urgent' job which does a very significant amount of calculation and occasionally outputs a result to the console, and a normal batch job which consists of a loop, in each iteration of which a card is read, the data is trivially transformed and the result is printed. If the CPU-bound job is given higher priority the effect will be that the IO-bound batch job will rarely be able to read, process and print a card since the other job will dominate the use of the CPU. As a result the card reader and line printer will be severely underutilised and the batch job will take an inordinately long time to finish. On the other hand if the batch job is given higher priority (although it is less urgent), the result will be that it frequently uses the CPU for very short periods sufficient for it to drive the peripherals at full speed, but for the very long periods (relative to CPU speeds) when the program is waiting for IO operations to terminate, the other job will have control of the CPU. The effect on the calculating job will be negligible, but the effect on the batch job and on peripheral usage will be dramatic.

Similar considerations could apply to any type of peripheral device usage, and the effect on real-time devices which must be serviced in a short time could be disastrous. It could therefore be regarded as a very serious defect of CHILL that no facility exists to allocate relative priorities to different CHILL processes. (The reader should not be deceived into thinking that the priority parameter associated with some of the synchronisation mechanisms will solve the problem.) Note, however, that L.D.5.2.17 allows a START expression to include "additional actual parameters ... with an implementation defined meaning"; perhaps this is the way to include a priority parameter, where necessary. Even so, there is apparently no method of changing a process's priority dynamically, e.g. as a process changes from an IO-bound phase to a CPU-bound phase.

One final point about process priorities. It is now almost universally accepted that solutions to synchronisation problems should not depend on the relative priorities of the processes involved in the solution, i.e. knowledge of the relative speeds of processes should not normally be assumed in designing algorithms. The danger is not only that such a solution is more difficult to understand and to modify, but also that the program will be less portable. For example a solution which assumes that process A will execute before process B (in a single CPU system) may not be portable to a dual CPU system.

2.4 Process Homogeneity

All processes created in an executing CHILL program are homogeneous in the sense that they are all subject to the same rules and behave in similar ways. However, it is unfortunate that this homogeneity does not extend to the outer process, i.e. the process which begins executing when a CHILL program is started. In fact the outer process is an elusive entity which has not been well thought out by the CHILL designers. It is difficult to get a complete or coherent view of its place in the language from either the Language Definition or the Introduction.

Other processes are started by executing a START action which specifies their starting-point (as a PROCESS name) and returns an INSTANCE value, i.e. a process number. A process can use the THIS operator to discover its process number dynamically. It terminates on completing its process definition or on executing a STOP action.

By comparison, the outer-process is 'considered to be created by a start expression executed by the system under whose control the program is executing' (L.D.8.1). It is not entirely clear where the outer-process starts executing. L.D.7.8 speaks of 'an imaginary process definition ... (which) ... is considered to contain in its reach a standard CHILL prelude module. This module contains the definitions of the CHILL pre-defined names and the implementation pre-defined built-in routines, modes and register names'. Since a PROCESS definition includes a <body> which ends in an <action statement> one might guess that the imaginary outermost process begins executing the <action statement> of the imaginary process definition. Unfortunately the fragments of example programs do not confirm this and the definition of <program> appears to make it improbable. In this case we can only conclude that the starting point for the outer process is undefined.

It is not clear whether the outer process has an INSTANCE number. If so it can only be obtained via the THIS operator. It is however clear (from L.D.8.1) that the outer process can execute a STOP, but when it does so it will only be terminated after all other processes have terminated.

It is not clear whether the outer process can synchronise with other processes using the normal mechanisms; for example there appears to be no way of naming its class of process. L.D.8.2.1 suggests that it can break the usual REGION rules by entering the REGION. (This is presumably to initialise REGION data, but there is no restriction that entry can occur only once.)

In summary, the imaginary outermost process appears to be a poorly thought-out and incompletely defined concept, which does not harmonise with the CHILL view of dynamically created processes. For this reason it is recommended that in a concurrent program its use should be confined to initialisation of data and to the creation of other processes.

2.5 Summary

CHILL includes concurrent processing features which superficially appear suitable to support either of two basic models for decomposing concurrent programs into processes. However, it does not directly include any means of controlling process priorities and its concept of an imaginary outermost process is both elusive in its definition and lacks homogeneity with dynamically created processes.

3. CHILL SYNCHRONISATION MECHANISMS

This section presents an analysis of the three main synchronisation requirements for concurrent programs: exclusion, resource allocation and inter-process communication. Attention is drawn to two different forms of exclusion (mutual exclusion and reader/writer exclusion), to a variety of resource allocation problems, and to the desirability of commutative operations for achieving effective inter-process communication. The relationships between these requirements and the CHILL synchronisation mechanisms are noted, although a detailed discussion of problems is left to subsequent sections.

3.1 Exclusion

When several processes need to access the same variable or variables, precautions must be taken to ensure that they do not interfere with each other and produce incorrect results. This problem arises only if one or more processes attempt to modify variables; the reading of variables is quite safe (provided that they are not at the same time being modified). Consequently we can recognise two main classes of exclusion:

(a) Mutual Exclusion: access to a variable or group of variables by one process must exclude accesses by all others. The regions of code which contain critical accesses are known as (mutually exclusive) critical regions. Only one process at a time may be active in any of the critical regions associated with a particular variable or variables. A simple example of this would be the updating of a variable used for counting the number of occurrences of an event, where different occurrences might be detected by different processes.

(b) Reader/Writer Exclusion: this case arises where one group of processes (the 'readers') wishes to read the variable and another group wishes to modify the variable (the 'writers'). The critical regions fall into two groups: reader regions and writer regions. Only one process may enter a writer region at a time, excluding both readers and other writers. But any number of readers may be concurrently active in a reader region, provided that writers are excluded.

Reader/writer exclusion is probably the most common requirement. It applies frequently to the accessing of tables, indices and files. Properly implemented it achieves a high degree of parallelism, especially in cases where a variable is frequently read but only occasionally updated. However it has proved difficult to implement using commonly available techniques (cf. (Ref. 15) for a semaphore-based implementation) and in consequence most practical systems reduce the problem to that of mutual exclusion, which is simpler to implement but which also reduces the amount of parallelism that can be achieved. This can be particularly unfortunate in high throughput real-time

systems. A new mechanism has recently been proposed to support the efficient implementation of reader/writer exclusion in microcode (Ref. 16).

In CHILL, processes can share data declared in a REGION and data declared in a MODULE. Data declared in a PROCESS appears to be local to each instance of a process (except if declared as STATIC, see L.D.7.9). Data declared within a module is not protected from multiple accesses by different processes. Therefore unless further precautions are taken it can only be shared by a community of processes which includes only readers but no writers.

Data declared within a region can be accessed only by entering the region (from the outer process) or by other processes calling its critical procedures (L.D.8.2.2). The language implementation has the responsibility of ensuring that a process executing within the region has mutual exclusion. Notice that a process in a region may call another region or module. In either case the region remains locked, and in the first case the called region is also locked. If a process is delayed in a region (in the sense of L.D.8.3) the mutual exclusion of that region is released (and regained when the process is reactivated).

Thus CHILL directly supports mutual exclusion; it does not support reader/writer exclusion.

3.2 Resource Allocation

In most concurrent systems processes have to compete for and cooperate in the use of scarce resources. For some resources this is made invisible to the applications programmer by facilities in the operating system (e.g. CPU allocation, memory management) but in other cases the programmer must provide explicit control of resource allocation. The exclusion problem discussed in Section 3.1 is actually a special case of resource allocation, where only one resource is available. Other situations include:

(a) a process requiring any one of several identical resources;

(b) a process requiring several different resources together.

The CHILL synchronisation features do not directly support these cases, but they are sufficiently flexible to allow the programmer to write a section of code to control allocation. This lack of direct support is unfortunate, at least with respect to (a), since that situation can be controlled very efficiently if appropriate hardware or microcode is available in the form of microcoded general semaphores (Ref. 5) or set semaphores (Ref. 17). Such operations could not be used from a CHILL program.

One method of programming these cases is by the use of regions, as is illustrated in L.D. Appendix D, example 13. Examples 15 and 16 illustrate resource allocation by the use of processes and signals or buffers.

An ever present problem with resource allocation is the risk of deadlock. In general it will be the programmer's responsibility to ensure that deadlocks are avoided, and the wise programmer will prefer some simple technique such as:

(a) allocating together all resources needed by a process;

(b) arranging resources into a hierarchy and enforcing the rule that once a process has a resource classified at level l in the hierarchy only claims for resources at levels higher than l will be considered.

While such rules are restrictive they have the advantage of simplicity, and do not incur the heavy run-time overheads associated with more ambitious deadlock prevention schemes.

Of more importance to this report is the question whether the CHILL language definition creates any hidden risks of deadlocks. Some examples of this will be described in later sections.

3.3 Interprocess Communication

Processes may communicate with each other for several reasons. In a message-oriented system (see Section 2.2) they may pass work to each other in the form of messages. In a procedure-oriented system they may issue signals to indicate that the signalling process has freed a resource which thus becomes available for use by another process waiting for the resource.

In both these cases there is a one-to-one relationship between the sending of one message and its receipt, or between one signal and its receipt. This relationship should be commutative in the sense that it does not matter whether the signal/send precedes or follows the receive in time; the nett effect should be the same in either case. Both the signal and the buffer modes of CHILL have this property and can be used in a similar way (see L.D. Appendix D examples 15 and 16). Notice that the event mode does not have this property, since the continue action is 'forgotten' if no process is waiting. Notice also that a message can be passed with each signal or buffer 'send', but not with the event 'continue'.

The problems of non-commutativity and lack of message passing with events can be illustrated by the 'allocate' procedure of example 13 (L.D. Appendix D). In that example if a resource is not available the claimant delays until one is freed, then has to search the whole list to find which one has been freed (because a message could not be passed to the waiting process with the continue call).* In fact the claimant even needs a 'do for ever' loop to achieve this. This inefficiency should be contrasted with the set semaphore solution (Ref. 17) which in a commutative way efficiently achieves the equivalent of example 13.

---

* See section 5.4(d) for a more detailed analysis of this example.

The CHILL buffer mode is more powerful than signals both in the sense that it provides an automatic mechanism for allocating buffer slots for passing messages between processes and in that it provides additional synchronisation by delaying a sender if there are no free slots in a buffer as well as delaying a receiver if there are no full slots. Thus buffers provide a direct solution of the classical 'producer/consumer' or 'bounded buffer' problem. Dijkstra (Ref. 5) has presented an elegant solution of this problem using general semaphores, and it can be solved even more simply and efficiently with set semaphores (Ref. 17). Hoare (Ref. 11) also presented a solution using monitors. Since CHILL regions are based on monitors, they could be used as a basis for providing bounded buffers; however the CHILL buffer mode provides a much more direct and attractive technique for producer/consumer relationships, which are the main basis for inter-process communication in 'out-of-process' systems (section 2.2).

3.4 Summary

CHILL provides direct support for mutual exclusion in the form of regions, but it does not provide direct support for reader/writer exclusion. No direct support is provided for resource allocation although regions and events, buffers and signals can be used for this purpose. An important requirement of most interprocess communication situations, commutative operations, is found to exist for the signal and buffer constructs, but not with events.

4. PROBLEMS WITH CHILL CONCURRENCY FEATURES

This section examines a variety of problems which are associated with the CHILL definition of concurrency features, ranging from lack of syntactic consistency, through process related problems, to the more serious problems of deadlocks and reduced parallelism which are created by the simplistic view of mutual exclusion adopted in the region construct.

4.1 Syntactic Inconsistency

It is an important programming language design principle that similar things are handled in a consistent and uniform manner. (This more than anything else explains why PASCAL has found increasing favour during the last decade). Among the advantages of such an approach are that such a language is easier to learn than one with an ad hoc syntax, and that it will tend to need a smaller compiler and supporting run-time package. Because it is easier to learn, programmers will make less errors and will not need to consult the manual so often. Because of the smaller compiler/run-time package programs can be more efficiently compiled and executed; there is also less chance of errors in the compiler itself.

CHILL, unfortunately, is not a good language in this respect. This is reflected not only in its sequential features but also in its concurrency provisions, as the following examples illustrate.

(a) Receive Statements As already noted, signals and buffers have similar properties, yet signals can only be received with a <receive case> statement while buffer messages can be received using either a <receive case> or a <receive> expression. Since a <receive case> can be specified with only one signal type there is no semantic loss. However, as the ALLOCATE-RESOURCES example in Fig. 3 (to be discussed in section 7.2) illustrates, the lack of a receive expression for signals results in syntactic inelegance, and can also be expected to lead to compilation errors for inexperienced CHILL programmers.

(b) Delay/Continue and Receive/Send Signals and buffers have receive and send statements, whereas events have delay and continue statements for performing parallel operations. Either a single set of pairs or three different pairs would be more appropriate. Probably the latter would be more suitable because (a) the detailed syntax of signal and buffer 'sends' differs, and (b) with the present syntax a programmer might be tempted to mix buffers and signals in a single 'receive' statement.

(c) Signal Declarations Buffers and events are regarded as modes, and in consequence follow the mode declaration syntax whereas a signal follows a different definition syntax, not being a mode.

(d) Delay and Delay Case If a <delay case> is used the reactivated process can discover, by the 'SET <instance>' option, which process activated it. But if a simple <delay> statement is used the activator's identity cannot be discovered. (Notice that the <continue> action for an event can be executed by any process).

(e) Event and Signal Direction The sender of a signal can specify a destination process, but the execution of an event <continue> cannot include a destination process.

4.2 Problems with PROCESS Constructs

The following problems were discovered in considering the apparently trivial problem of how to organise a set of operations to allow processes to suspend or activate each other in arbitrary ways.

(a) Arrays of Process Instances. One method that can in principle be used to solve the above problem is to declare an array of events (or signals) with one element in the array corresponding to each process instance. A process would then suspend itself by the operation

```
DELAY PROCESSARRAY(THIS)
```

which uses the THIS operator to select the correct event in the array. Later the process would be activated by another process executing the operation

```
CONTINUE PROCESSARRAY(X)
```

where X specifies the instance value of a delayed process. Unfortunately this technique cannot be used in CHILL programs because array indices must be of a discrete mode, and process instance values are not classified as discrete. (In practice most systems avoid this problem by treating process identifiers as an integer subrange.)

(b) PROCESS Declarations and Deadlocks. Solution (a) having failed, a second solution was considered. The aim of the first solution was to implement a private event for each process. Why not achieve this by declaring an event in each PROCESS definition? The solution partly works, in the sense that such an event per process can be declared, and the process can delay on its own event. However, the event variable is not in the addressing scope of other processes, unless it is explicitly passed as a LOC parameter. (Notice that GRANT and SEIZE do not solve this problem of visibility). Therefore, except in this case, it is impossible for any other process to continue such a delayed process, and the process is in a deadlock. There is one further possible exception, which is that the outer process may be able to access such an event (see L.D.7.5) - although this (a) does not solve the initial problem, and (b) once again demonstrates the awkward relation between the outer process and other processes. A careful language designer would have avoided this potential deadlock risk by prohibiting events (and signals) from PROCESS definitions. With the CHILL definition, the onus is unnecessarily placed on programmers to avoid this deadlock.

4.3 Regions and Deadlocks

In CHILL a process executing in a REGION can call a critical procedure of another region. In this case each region is locked when it is called and unlocked when its critical procedure returns. Thus if a process P calls region A and from A calls region B both regions will have mutual exclusion locks set. A process can be delayed within a region (e.g. by executing a delay or receive operation), in which case that region is unlocked, but not the regions from which it was called (see L.D.8.2.1). Thus if P calls A, then from A calls B and then delays, A will remain locked but B will be unlocked.

Suppose now that A controls a set of resources which, to be useful, need to be combined with a resource controlled by B. Process P_1 calls A and B, and having succeeded in acquiring its resources returns from both regions. Process P_2 attempts to do likewise but fails to get B's resource, so delays until it becomes available. A is now locked but B is free. Eventually P_1 decides to release the resources, so it calls A - and waits because A is locked. This is a classic deadlock situation. The programmer has apparently applied good structuring techniques (layers of abstraction, information-hiding), but the language construct has caused him to create an unexpected deadlock. The basic problem is that the language designers have oversimplified mutual exclusion requirements by implicitly associating critical sections with region procedures and removing control from the reach of the programmer. Unfortunately the situation as described is frequently desirable in the design of real-time systems and operating systems.

4.4 Regions and Parallelism

As was noted above, regions provide full mutual exclusion for access to their encapsulated data structures. That this reduces the level of parallelism which can actually be achieved in some situations has already become evident from the discussion of the reader/writer synchronisation requirements in section 3.1. In the following we describe some further examples of this problem.

(a) Parameter Checking. It is usually considered good practice, especially in a large system developed by different programmers, to check parameters in a called procedure before performing the main function of the call. Since parameters are normally local to the process several processes can in principle carry out checks in parallel within the same module (see Fig. 1). The exclusion rules make this impossible in critical procedures of CHILL regions.

```
SHARED_INFO:
   MODULE
     GRANT THIS_PROC, THAT_PROC;
     DCL SHARED_DATA  {some data definitions};
     SIGNAL MUTEX;    {mutual exclusion mechanism}

   THIS_PROC:
     PROC (--- params ---) (INT);
     IF param1 not valid THEN RETURN 1 FI;
     IF param2 not valid THEN RETURN 2 FI;
     ..... other parameter checks .....
     RECEIVE CASE (MUTEX):

       {mutually exclusive code accessing
        the shared data}

     ESAC; SEND MUTEX;
   END THIS_PROC;

   THAT_PROC:
     PROC (--- params ---) (INT);
     IF param1 not valid THEN RETURN 3 FI;

           ..... etc. .....

     RECEIVE CASE (MUTEX):

           ..... etc. .....

     ESAC; SEND MUTEX;
   END THAT_PROC;

   ..... shared data initialisation .....

   SEND MUTEX;
END SHARED_INFO;
```

Fig. 1 - A Framework for Modules in an In-Process System

(b) The Bounded Buffer Problem. Although CHILL supports bounded buffers directly in its syntax, it is instructive to use this situation as an example of how varying degrees of parallelism can be achieved in a synchronisation problem. (It would also be interesting to discover the level of parallelism achieved in CHILL compiler impiementations of bounded buffers.) We illustrate this by the use of semaphores, showing a series of solutions with increased parallelism. In each case FULL and EMPTY are general semaphores, counting the number of full and empty buffers respectively. For a buffer with N slots, the initial values are FULL = 0, EMPTY = N.

Solution 1 The Classical Solution (Ref. 5).

MUTEX is a binary semaphore initialised to 1.

```
proc producer (item)          proc consumer (item)
 begin                         begin
    P(EMPTY);                     P(FULL);
    P(MUTEX);                     P(MUTEX);
    fill a slot                   empty a slot
    V(MUTEX);                     V(MUTEX);
    V(FULL);                      V(EMPTY);
 end;                          end;
```

This solution allows any number of producer processes and any number of consumer processes to work concurrently. However MUTEX ensures that only one process at a time can access the buffer. (N.B. Because the sequence of P operations is critical - the reverse order can produce deadlocks - a CHILL programmer should not attempt to implement this algorithm using a combination of region procedures and signals.)

Solution 2 Parallel Producers and Consumers

PROD and CONS are binary semaphores initialised to 1;

NEXTFULL and NEXTEMPTY are integers initialised to 0;

```
proc producer (item)
 begin
    P(EMPTY);
    P(PROD);
    BUFFER[NEXTEMPTY]:=item;
    NEXTEMPTY:=NEXTEMPTY+1 mod N
    V(PROD);
    V(FULL);
 end;

proc consumer (item)
 begin
    P(FULL);
    P(CONS);
    item:=BUFFER[NEXTFULL];
    NEXTFULL:=NEXTFULL+1 mod N;
    V(CONS);
    V(EMPTY);
 end;
```

In this solution producers exclude each other and consumers exclude each other, but one consumer and one producer can work in parallel on different buffer slots.

Solution 3 One Consumer and Several Producers

```
proc producer (item);
 begin
    P(EMPTY);
    P(PROD);
    BUFFER[NEXTEMPTY]:=item;
    NEXTEMPTY:=NEXTEMPTY+1 mod N;
    V(PROD);
    V(FULL);
 end;

proc consumer (item);
 begin
    P(FULL);

    item:=BUFFER[NEXTFULL];
    NEXTFULL:=NEXTFULL+1 mod N;

    V(EMPTY);
 end;
```

In this solution the CONS semaphore is dropped because only one process consumes items. (Mutatis mutandis the case of one producer and several consumers would retain the CONS semaphore and remove the PROD semaphore.)

Solution 4 One Producer and One Consumer

```
proc producer (item);
 begin
    P(EMPTY);
    BUFFER[NEXTEMPTY]:=item;
    NEXTEMPTY:=NEXTEMPTY+1 mod N;
    V(FULL);

proc consumer (item)
 begin
    P(FULL);
    item:=BUFFER[NEXTFULL];
    NEXTFULL:=NEXTFULL+1 mod N;
    V(EMPTY);
```

Where there is only one producer process and one consumer process solution 4 applies. Notice there is no mutual exclusion semaphore whatsoever. Provided there is an empty slot the producer can be active; while there is a full slot the consumer can be active.

This example illustrates vividly that mutual exclusion is an overkill approach to many synchronisation problems.

It is evident from these examples that the use of regions in CHILL may well reduce the amount of parallelism which would otherwise be achieved in some cases.

4.5 Summary

There is a lack of consistency and uniformity in the way the CHILL concurrency features are defined. This is likely to result both in compilation inefficiencies and programmer errors. There are at least two aspects of the language definition which increase the risk of deadlocks. The second of these, a defect in the region construct, is particularly unfortunate since it arises in many situations and it limits the use of good program

structuring techniques. Finally, the region construct is found also to limit the level of parallelism which can otherwise be achieved in solving some problems.

5. IMPLEMENTATION OF THE CHILL SYNCHRONISATION CONSTRUCTS

This section reviews the basic hardware mechanisms which can be used to implement concurrent features of programming languages. After considering how to overcome the problems inherent in some mechanisms, the problems associated with CHILL regions, events, signals and buffers are examined.

5.1 Basic Implementation Techniques

Synchronisation constructs provided by a programming language can only be implemented if some synchronising mechanism is provided on the base machine. The ease with which the constructs can be implemented will depend on the nature of the constructs, the nature of the basic primitives and how well the two are matched. This section briefly reviews the properties of the sorts of primitives likely to be available for implementing CHILL constructs on typical computers.

(a) Memory Interlock. All modern hardware ensures that the storing of a value into a single word memory will be carried out indivisibly, i.e. if two store operations into a word of memory are attempted at exactly the same time the hardware will arbitrate, forcing the stores to proceed in sequence. Likewise a single word cannot be simultaneously read and updated.

Memory interlock is a sufficient condition for implementing synchronisation, through the use of Dekker's algorithm, (Ref. 18). However this is of little practical interest since it requires that the number of processes be known in advance, its complexity increases with the number of processes and it relies on 'busy waiting'. It would be a horrendous task to implement CHILL constructs using memory interlock alone.

(b) Inhibiting Interrupts. The basic pre-requisite for synchronisation is that a process be guaranteed indivisible access to some variable. In a single CPU system this can usually be achieved by turning off interrupts, performing the required action, then turning them on again. With all interrupts turned off (inciuding clock interrupts) the process scheduler cannot regain control and unexpectedly switch to another process. The three problems with this technique are (i) if interrupts are inhibited for long periods other side-effects can arise (e.g. data can be lost on reading a card); (ii) it is not suitable for synchronising user programs which might contain errors (e.g. an infinite loop with interrupts off destroys the system), and (iii) it cannot be used in a system with multiple CPUs (which implies that the system cannot be upgraded later).

For these reasons inhibiting interrupts is not a recommended technique for implementing CHILL synchronisation constructs. However, if it is the only available mechanism, the designer is recommended to follow the technique described in section 5.2.

(c) Test and Set Instructions. This is the basic mechanism supported by many third generation computers. Mutual exclusion is achieved as follows:

```
entry:  repeat test-and-set (common, local)
          until local = 0;
        {mutual exclusion section}

exit:   common:=0;
```

The test and set instruction indivisibly copies the value of common into local then sets common to 1. Common is a shared variable; common = 1 implies mutual exclusion is set, and common = 0 implies mutual exclusion is not set. Each process has its own copy of local which it examines to see if it has been granted exclusion.

The advantages of the test and set instruction are that it is straightforward, it does not inhibit interrupts and it works for any number of CPUs and any number of concurrent processes.

The main problem is that it uses busy waiting. This not only consumes CPU time wastefully, but it can also lead to deadlocks if the CPU scheduling algorithm is priority-based, as follows. A high priority process in a single CPU system might busy wait for ever while a lower priority process which was earlier granted exclusion is in the critical region but has lost use of the processor in favour of the higher priority (busy waiting) process (Ref. 19).

(d) Semaphore-like Instructions. Busy waiting occurs when instructions are not commutative. Some modern computers remedy this by providing pairs of instructions which correspond to the (non-queueing aspects of) P and V operations on semaphores, as follows:

```
entry:  decrement-and-test (common, local);
        if local < 0 then suspend (semqueue);
        {mutual exclusion section}

exit:   increment-and-test (common, local);
        if local ≤ 0 then activate (semqueue);
```

The instructions 'decrement-and-test' and 'increment-and-test' are indivisible. They each operate on the shared integer variable common, decrementing or incrementing it by 1 then copying the result into local, a separate copy of which exists for each process (e.g. the ICL2900 variant of these instructions uses the condition code register (Ref. 20). 'Suspend' and 'activate' are queueing routines of the process scheduler which suspend the calling process or activate another process in a queue associated with the semaphore. These queueing routines are also commutative (i.e. an 'activate' on an empty queue causes the first process suspending thereafter to continue without being suspended).

The advantages of semaphore-like instructions are that they work for any number of processes and/or CPUs, they do not 'inhibit' interrupts, and they do not involve busy waiting.

(e) Extensions to Semaphore-like Instructions. Two further types of improvement can be made:

(i) special-purpose extensions such as set semaphores (Ref. 17) to simplify resource allocation, and reader/writer semaphores (Ref. 16); and

(ii) direct hardware or microcode support for manipulating the queues (cf. VAX11-780 (Ref. 21))or reducing contention on queues (Ref. 22).

5.2 Layers of Implementation

Semaphore-like instructions are sufficiently powerful to be placed in user programs (which will call operating system routines to suspend and activate processes where necessary). This can result in a very simple set of operating system functions, upon which a compiler can build more powerful facilities.

However, mechanisms such as inhibiting interrupts and test and set are too dangerous and too primitive to be used directly in user programs. For this reason most operating systems supply higher level facilities which themselves use the primitives in a safer way.

For example instead of using control of interrupts or test and set directly to implement entry and exit from critical sections, they can be used to control shorter critical sections which in turn implement the entry and exit protocols. Thus P and V operations could be implemented along the following lines as operating system routines:

```
P(SEM):
   turn-off-interrupts;
   SEM:=SEM-1
   if SEM < 0 then suspend-caller(semqueue);
   reschedule;
   continue selected process with interrupts on;

V(SEM):
   turn-off-interrupts;
   SEM:=SEM+1;
   if SEM ≤ 0 then activate-a-process(semqueue);
   reschedule;
   continue selected process with interrupts on;
```

A similar technique can be used with test and set instructions. P(SEM) and V(SEM) are made available as supervisor calls and the more primitive mechanisms are used only by the process scheduler.

This has several advantages:

(a) user processes cannot crash the system;

(b) user programs/compilers are unaware of the basic synchronisation mechanism; hence it can be changed (e.g. from inhibiting interrupts to test and set if a second CPU is added) without the need to modify or recompile them;

(c) interrupts are off only for short periods;

(d) any busy waiting is reduced to a minimum.

5.3 Implementing Higher Level Constructs

So far we have concentrated on mutual exclusion in the discussion of implementation techniques. This is because almost all other synchronisation constructs can be implemented only if a mutual exclusion mechanism is available. All the constructs of CHILL fall into this category.

There is no firm nor clear line regarding the method of structuring the implementation of higher level synchronisation constructs. It is possible to provide direct support for all the desired mechanisms in an operating system, or, provided that the operating system supports basic facilities for suspending and activating processes, the higher level can be built as run-time routines of the language. Compromises between these two extremes are also possible. Any particular case will depend on questions such as:

> Is the operating system designed to support several languages with different synchronisation requirements?
>
> Will assembler programs also use the system?
>
> Were the language constructs defined or known when the operating system was developed?
>
> Should the system be extensible to support other languages later?

For these reasons the following discussion of the problems associated with implementing CHILL constructs could apply either to the development of an operating system or to the development of a run-time package. For convenience it is easier to refer to them as operating system problems, but this should be interpreted as a reference to either. In practice it does not seriously affect the nature of the problems, whichever approach is taken.

5.4 Problems with Implementing Regions

At first sight regions appear to map directly onto a mutual exclusion primitive. While it is true that entry to a region's critical procedures will involve the claiming of a mutual exclusion semaphore and returns from a critical procedure will involve the release of the semaphore, the situation is much more complicated, as the following discussion shows.

(a) The Outer Process. This process can enter and exit from a region without calling its critical procedures. All entry points will have to be determined and protected in a manner

similar to the calls of critical procedures. Note that while a created process cannot jump into a region, the definition of the 'goto' action (L.D.6.9) does not preclude the outer process from jumping into a region.

(b) Nested Regions. As noted in section 4.3, a process executing in a region may call another region. To implement the correct exclusion it will be necessary to maintain a stack of semaphores to be released.

(c) Delays in Regions. If a process is delayed within a region (in the sense of L.D.8.3), exclusion must be released for the region in which the delay occurs. Since delays can also occur outside regions the compiler must compile different code (for each of six different constructs, as described in L.D.8.3) depending on the context. Likewise a delayed process must regain exclusive control when it restarts (with four different constructs involved) if it is in a region. Notice the implementation complexity of 'send buffer', which if called from within a region may cause the calling process to delay (releasing region exclusion if necessary) or it may cause another process to be activated (perhaps needing to reclaim exclusion for the same or some other region): 'receive buffer' creates similar complexity.

(d) Resuming a Delayed Process. When a process has been reactivated after a delay in a region, it is possible that the activating process may also be in the same region. Unlike Hoare's solution to the equivalent monitor problem (Ref. 11), the CHILL designers have defined that the "reactivating process will remain active, i.e. it will not release the region at that point" (L.D.8.4). In this situation the reactivated process is not allowed simply to proceed, but it must regain mutual exclusion. It appears to be undefined whether such a process is guaranteed to regain exclusion as soon as its activation leaves the region or whether it must compete with other contenders for use of the region. Both alternatives have problems.

If the reactivated process is guaranteed to be next to run after the activator releases the region, then the implementation becomes very messy, since it could conflict with the normal algorithm of the CPU scheduler, and it becomes a very special case from the implementation viewpoint. (The problem becomes even more confused if the activating process reactivates several processes!)

If on the other hand a reactivated process competes again for mutual exclusion in the normal way, the programmer has to be very careful to ensure that his code does as he intends. This can be illustrated by the case of a region used to allocate a single resource, as follows:

```
ALLOCATE_RESOURCE:
   REGION
     GRANT ALLOCATE, DEALLOCATE;
     DCL ALLOCATED BOOL:=FALSE;
     DCL RESOURCE_FREED EVENT;
     ALLOCATE:
       PROC( );
          IF ALLOCATED
            THEN DELAY RESOURCE_FREED;
          FI;
          ALLOCATED:=TRUE;
     END ALLOCATE;
     DEALLOCATE:
       PROC( );
          ALLOCATED:=FALSE;
          CONTINUE RESOURCE_FREED;
     END DEALLOCATE;
```

This code naively assumes that if the resource was allocated the process will be delayed until it becomes free; the next statement declares it reallocated and the caller now has the resource. This code would be correct if the delayed process is guaranteed to be the next process to enter the region - and this may be the intention of the CHILL designers, though it is not defined. But if a different process can intervene it may allocate the resource, with the result that two processes think they have the same resource. (This is just another problem of non-commutative synchronisation mechanisms.)

To guarantee a correct solution we have to build in busy waiting, as follows:

```
ALLOCATE:
   PROC( );
     DO FOR EVER;
       IF ALLOCATED
         THEN DELAY RESOURCE_FREED;
         ELSE ALLOCATED:=TRUE;
              RETURN;
       FI;
     OD;
END ALLOCATE;
```

This is based on the method used in L.D. Appendix D example 13. A neater alternative would be:

```
ALLOCATE:
   PROC( );
     DO WHILE ALLOCATED;
            DELAY RESOURCE_FREED;
     OD;
     ALLOCATED:=TRUE;
END ALLOCATE;
```

It is evident that delays in regions are either difficult to implement or are dangerous to use. The Language Definition does not specify which of these two conditions is true.

(e) Regionality Restrictions. CHILL defines a set of static context conditions which place restrictions on various features such as region variables and procedures (see L.D.8.2.2). Thus regions produce secondary effects on a very wide range of other CHILL constructs; this can only result in considerably increasing the size and error-proneness of a CHILL compiler and in reducing compilation speed.

5.5 Problems with Implementing Events, Signals and Buffers

In general these constructs and their operations appear to be relatively straightforward to implement, given a basic mutual exclusion mechanism, despite the inconsistencies in their definitions. The main problem is their multiplicity, i.e. different mechanisms for achieving basically similar results. This will result in a large amount of implementation code.

The only other point worth noting is the 'delay case' and 'receive case' actions will result in an excessive amount of queueing code and/or queue searching at run time. This is worsened by the facility for specifying individual processes or process classes for activation and the facility for specifying activation priorities.

5.6 Summary

A variety of basic mechanisms for supporting concurrent constructs is found in modern computers. While commutative semaphore-like operations can be used directly in the object code of user programs, mechanisms such as test-and-set and inhibiting interrupts are too dangerous. A technique was described for using these mechanisms as a basis in an operating system for providing safer higher level operations. Then followed an examination of regions, which were found to suffer from a substantial number of non-trivial implementation problems. The subsequent review of events, signals and buffers suggests that while not trivial to implement, they are much more straightforward than regions.

6. EXPANDING THE CONCURRENT PROCESSING FEATURES OF CHILL

CHILL already provides such a wide variety of concurrent processing features, that the addition of new general purpose synchronisation mechanisms is hardly necessary. The only significant omission is a facility for reader/writer exclusion. While a construct of this nature could be supported (based for example on the module and signal facilities) the complexity of the operations, as demonstrated by the reader/writer semaphore solutions given by Courtois, Heymans and Parnas (Ref. 15), argues against such an extension, given that mutual exclusion is available as an alternative. There are, however, three areas where expansion of the present facilities might become necessary, for input-output operations, for timing operations and for distributed processing systems. The discussion of these areas is followed by a recommendation on how expansion should proceed.

6.1 Synchronising Input-Output Operations

CHILL does not provide input and output routines, on the grounds that such routines can be written in CHILL (Introduction 2.3.2). While it is not a function of this report to define these facilities or their implementation, the following points should be considered by the designer of such facilities.

(a) Synchronisation between programs and input-output operations can be defined in two basic ways, corresponding to the two process structuring models described in section 2.2. In an in-process system each task executes as a separate process and an entire task executes as one process. Input-output operations are therefore considered in principle as procedure calls, so that the process sees the operation as one in a sequence. This leads to waiting-I/O in the sense that the process issuing the I/O waits for completion before attempting any further instructions. Synchronisation with the I/O devices can therefore be viewed as requesting a resource and waiting for it to become available. In accordance with the guidelines recommended in section 7 this can be implemented as a special case of signals, where RECEIVE corresponds to an I/O request and SEND corresponds to a completion signal.

In an out-of-process system each subtask is a separate process, and input-output can be considered as a separate concurrent operation, which leads to non-waiting I/O, often implemented in practice as double-buffering. With the I/O device considered as a separate process, each requesting process sends a message, using a message buffer and in return the I/O device sends a message to inform of completion.

(b) The mapping of these operations onto real processors is not trivial and at some point in the routines implementing I/O (either in the operating system or in CHILL run-time routines) real input-output operations have to be synchronised with the hardware. Some techniques for achieving this are discussed in Keedy (Ref. 23) and in Keedy and Rosenberg (Ref. 24). The former discusses problems with monitor implementations, which will be of particular interest to a designer contemplating implementing I/O with CHILL regions.

6.2 Synchronising with Time

CHILL has no provisions for synchronising with time. L.D. Appendix D example 14 contains a procedure(?)WAIT. Presumably this also has to be supplied in a CHILL routine. The designer of such a routine might be interested to note that time can be treated like I/O devices and the recommendations in section 6.1 can therefore be followed. In practice the use of signals (in-process design) or buffers (out-of-process design), rather than a special WAIT routine, is recommended because the 'receive case' provides the opportunity for a process to suspend waiting for some other signal, but timing out after a defined interval.

6.3 Distributed Systems

CHILL does not address the area of synchronising distributed processing systems (e.g. systems built as networks of micro-processors). Recently there have been some attempts to define suitable high level language constructs for such systems, based on the notion of input-output operations (Refs. 25, 26). These proposals are as yet largely untried in real systems, and it is not certain that they

represent a good solution. At this stage it seems premature to suggest how CHILL should be used or extended to meet this possible future need.

6.4 Guidelines for Expanding CHILL

If expanded facilities are required for CHILL, it is recommended that they are based on 'modules' and 'signals' (for in-process systems) or on 'processes' and 'buffers' (for out-of-process systems), in accordance with the more detailed guidelines and examples provided in section 7. Events are rejected because they are not commutative; experience shows that the absence of this property results in busy waiting as well as difficult and error-prone programming (cf. sections 3.3 and 5.4(d)). Regions are rejected as a basis for expansion because of their implementation difficulties (section 5.4) as well as their potential for introducing deadlocks (section 4.3) and for reducing parallelism (section 4.4).

6.5 Summary

Expansion of CHILL to provide I/O facilities and timing facilities may be necessary. Guidelines regarding the synchronisation aspects of both features for both in-process and out-of-process systems are suggested. No proposals are made for the possible extension of CHILL to cover distributed systems. As a general principle it is recommended that either 'modules' and 'signals' or 'processes' and 'buffers' are used as a basis for expanding CHILL, with events and regions rejected because of their inherent problems.

7. PROPOSED GUIDELINES FOR THE USE OF CHILL CONCURRENT PROCESSING FACILITIES

CHILL provides an unnecessarily and confusingly wide range of synchronisation facilities. This section proposes a set of guidelines to be followed in the development of CHILL programs. They are based on

(i) the need to develop a uniform philosophy for developing concurrent programs;

(ii) the selection of supporting facilities which are clean and easy to use, and which will result in programs which can be easily implemented and easily understood;

(iii) the avoidance of problematic constructs.

As noted in section 6.4, regions and events are avoided because of their many hidden difficulties both of implementation and of use.

7.1 Uniformity of Approach in Developing Concurrent Programs

Perhaps the biggest weakness of CHILL is that it tries to do all things for all philosophies of concurrent programming; hence the wide variety of constructs.

In section 2.2 attention was drawn to the fact that two quite distinct philosophies can be adopted in determining how a concurrent program should be decomposed into processes. With the in-process or procedure-oriented approach many similar processes will work in parallel, performing parallel tasks and communicating with each other indirectly via shared variables (e.g. to determine the allocation of resources amongst themselves). The static decomposition of a program into modules will not affect the definition of a process, since communication between modules will take the form of procedure calls.

With the out-of-process or message-oriented approach a task is decomposed into separate sub-tasks which are implemented in separate modules. Each module is executed by a separate process, and communication between processes (i.e. between modules) takes the form of message passing.

Lauer and Needham (Ref. 8) have argued that statically these two techniques are duals, and that any system which can be implemented using one technique can, in principle, be transformed into the other in a systematic way. Keedy (Ref. 9) has argued that the dynamic properties of the two approaches are not equivalent, and that the in-process or procedure-oriented technique is superior in this respect.

It is of course possible to mix the two approaches, and this is what has happened in many, if not most, concurrent programs and systems (without conscious decision on the part of the designer). The problem with this is that such programs become inconsistent, they lack coherence and they are more difficult to understand. Furthermore two different classes of synchronisation techniques have to be supported.

It is therefore recommended that the user should carefully study the documents discussing the two philosophies and make a choice between them to be used as a standard for the development of concurrent programs in CHILL. This will have at least two benefits. First, programmers will need to learn only a standard subset of the bewildering variety of facilities in the language. Second, programmers will find it much easier to understand (and maintain) the work of other programmers.

The following subsections recommend alternative guidelines which could be adopted depending on the decision made.

7.2 Use of CHILL for In-Process Designs

When decomposing a CHILL program into processes according to the in-process philosophy, the designer will identify the basic concurrent activities of the system and for each type of activity thus identified will declare a PROCESS. For example in a program to control calls at a switchboard, the actions involved in handling a single call will be defined as a PROCESS. (Of course common sub-actions could be declared in other modules, to be called as procedures, from the PROCESS definition.) If different basic activities can be identified (e.g. incoming

calls, outgoing calls, operator actions), each type of activity should be given a separate PROCESS definition.

After initialising all module data, etc. the outer process should either create a set of process instances (as many as necessary for each PROCESS definition) and suspend them awaiting the arrival of an associated activity, or should dynamically create process instances as activities arrive (the decision between these two options being based on considerations such as the maximum number of concurrent processes, the overheads of creating processes, etc.).

Although sharing code, these processes work independently of, but in parallel with, each other. Their basic synchronisation needs are

(a) exclusion in accessing common variables, and

(b) serialisation of activities when waiting for resources (e.g. access to the operator).

(a) Exclusion. While the CHILL designers offer regions as the exclusion mechanism, this report recommends that they be avoided because of the many problems of implementation and of use which they create. Instead mutual exclusion can be implemented straightforwardly by the combined use of MODULEs and SIGNALs, as the framework in Fig. 1 illustrates. There are several points to note about this framework.

(1) It uses encapsulated data (i.e. the information-hiding principle, cf. (Ref. 27)), as does the region construct. This is an important software engineering technique (Ref. 28). Notice that only procedures, not shared data, should be GRANTed.

(2) Initialisation (at the end of the module, performed by the outer process) includes a SEND MUTEX call. This makes the shared data initially available.

(3) Only the necessary minimum of code is mutually exclusive. In the example parameter checking proceeds in parallel (cf. section 4.4).

(4) RECEIVE CASE (MUTEX): is equivalent to a P operation on a Binary Semaphore. It is unfortunate that there is no receive without a case for signals. This leads to ESAC; SEND MUTEX, which can be regarded as a V operation.

(5) If the data is decomposable into different access groups, these could be protected by separate signal semaphores.

(6) Nested module calls are permitted. The programmer now has the flexibility to control when mutual exclusion is on and can therefore avoid the deadlock situation described in section 4.3. (There is no single solution to this problem; the appropriate remedy depends on the circumstances, cf. (Ref. 29)).

The above example achieves mutual exclusion but not reader/writer exclusion, for which CHILL offers no straightforward technique. But because signals can act like binary semaphores, one of the solutions presented by Courtois, Heymans and Parnas (Ref. 15) could be implemented, although in practice it will be easier to treat reader/writer cases as mutual exclusion cases.

(b) Resource Allocation. Consider first the case of a module which controls the allocation of a single resource to processes. Such a module is shown in Fig. 2. Notice that this is just another form of the mutual exclusion problem. Consequently the body of each procedure simply contains the code equivalent to the P or V operation in the previous example. (As usual the requirement to use CASE ... ESAC is irritating.) Although the operations can obviously be inserted in-line it will sometimes be useful to implement a module with this framework but including ancillary operations such as checking the caller's right to use the resource.

```
SINGLE_RESOURCE:
    MODULE
        GRANT ALLOCATE, DEALLOCATE;
        SIGNAL RESOURCE;
        ALLOCATE:
          PROC( );
          RECEIVE CASE (RESOURCE): ESAC;
        END ALLOCATE;
        DEALLOCATE:
          PROC( );
          SEND RESOURCE;
        END DEALLOCATE
        SEND RESOURCE;
END SINGLE_RESOURCE;
```

Fig. 2 - A Module for Allocating a Single Resource

A second common resource allocation problem occurs when a module allocates any one of several identical resources to a caller. Fig. 3 provides a framework for this case, which can be seen to be a slight extension of Fig. 2. Callers are advised which resource

```
ALLOCATE_RESOURCES:
    MODULE
        GRANT ALLOCATE, DEALLOCATE;
        NEWMODE  RESOURCE_SET = INT (0:9);
        SIGNAL RESOURCE = (RESOURCE_SET);
        ALLOCATE:
          PROC( )(INT);
          RECEIVE CASE
          (RESOURCE IN I): RETURN I;
          ESAC;
        END ALLOCATE;
        DEALLOCATE:
          PROC(I INT);
          SEND RESOURCE(I);
        END DEALLOCATE;

    DO FOR NEXT IN RESOURCE_SET;
        SEND RESOURCE (NEXT);
    OD;
END ALLOCATE_RESOURCES;
```

Fig. 3 - A Module for Allocating Equivalent Resources

(identified in the example by an integer in the range 0 to 9) has been allocated to them, and they are trusted to pass back the identifier when deallocating the resource. If callers are not trustworthy then extra code should be added to validate the information; if this involves the use of shared data mutual exclusion signals should be used. The module shown in Fig. 3 simulates a set semaphore (Ref. 17).

More powerful resource allocators might have to be developed to handle such problems as requests for multiple resources, deadlock avoidance, etc. In some such cases the resource allocator would need to organise its own process queues, which implies the need to suspend and activate processes in an arbitrary way. Two simple primitives 'suspend (me)' and 'active (him)' are needed for this purpose. Fig. 4 shows a module which provides these functions. This uses the signal option of specifying a destination process instance on SEND. Users of the module have to store in their own variables (e.g. queues) the instance numbers of suspended processes, obtained by using the THIS operator.

```
SCHEDULER:
   MODULE
      GRANT SUSPEND, ACTIVATE;
      SIGNAL SCHEDULE;
      SUSPEND:
         PROC( );
         RECEIVE CASE (SCHEDULE): ESAC;
      END SUSPEND;
      ACTIVATE:
         PROC (HIM INSTANCE);
         IF HIM=NULL THEN {error action}
                     ELSE SEND SCHEDULE TO HIM;
         FI;
      END ACTIVATE;
END SCHEDULER;
```

Fig. 4 - A Module for Suspending and Activating Processes

7.3 Use of CHILL for Out-of-Process Designs

When decomposing a CHILL program into processes according to the out-of-process philosophy, the designer will break the program activities into discrete subactivities which can be carried out in parallel, rather like the way a manufacturing activity is broken into subactivities implemented at different parts of a factory assembly line. In general each stage will correspond to a separate module with its own process.

Notice that in CHILL terms all modular program units will be declared as PROCESSES; there will be no MODULEs or REGIONs in a truly out-of-process system. The only exception, deriving from the poorly formulated program structure of CHILL, is that the outer process will operate in a MODULE! In other words, because facilities for separate compilation and linking are not discussed, it is not clear whether each service module, e.g. of an operating system, will operate in a separate imaginary outermost process.

(a) Exclusion. Provided that PROCESSes are formulated as information-hiding modules, such that they do not share common data (Ref. 27), which is in any case a desirable software engineering practice (Ref. 28), the problem of exclusion does not exist: each process accesses separate variables (ignoring messages).

(b) Resource Allocation. In an out-of-process system it is often feasible to allocate resources statically to processes since the subdivision of the activity may have been governed by resource usage. In cases where this is not true it is usual to provide a separate PROCESS which allocates resources to other processes. This process will receive messages requesting the allocation and de-allocation of its resources. If resources are not immediately available the process may need to suspend the caller and reactivate it later, i.e. ideally it needs to pass messages to a scheduler along the lines 'suspend (him)' and 'activate (him)'. Notice that this suspend call differs from that for in-process systems insofar as one process designates another for suspension. CHILL provides no facility onto which this can be mapped directly. Therefore the sensible alternative is for the resource allocator process to pass back a message to the calling process advising it to suspend itself; the allocator will then wake it up when the resources become available.

(c) Inter-process Communication. In an out-of-process system processes communicate by sending messages to each other. In their discussion of the duality of the two process structuring models Lauer and Needham (Ref. 8) compare message-passing with procedure calls. This analogy is only correct if each message is followed by a corresponding reply, for which the sending process may wait. They define message-passing protocols to achieve this.

The buffer facilities of CHILL are concerned with the passing of messages, but not replies. Fig. 5 provides a framework showing how these facilities would typically

```
FILE: MODULE

   NEWMODE  OPENPARAMS    = .........,
            CLOSEPARAMS   = .........,
            READPARAMS    = .........,
            WRITEPARAMS   = .........;

   DCL OPEN BUFFER OPENPARAMS,
       CLOSE BUFFER CLOSEPARAMS,
       READ BUFFER READPARAMS,
       WRITE BUFFER WRITEPARAMS;

   P: PROCESS (.........);
   DO FOR EVER;
   RECEIVE CASE
       (OPEN) :.... code for opening a file ....
       (CLOSE):.... code for closing a file ....
       (READ) :.... code for reading a record ....
       (WRITE):.... code for writing a record ....
   ESAC;
   OD;
   END P;
   END FILE;
```

Fig. 5 - A Framework for Implementing Processes

be used to implement an out-of-process module which offers multiple related services to other processes using the sample of a file manager. (Note that with the in-process dual, 'open', 'close', 'read', and 'write' would be separate procedures of a MODULE.)

For a message with reply system the present CHILL facilities are inadequate because as part of a send operation the sender must receive a message identifier with which he can identify his reply (see (Ref. 8)). However a variety of alternatives could be devised to achieve an almost equivalent mechanism. For example each process could have its own input reply buffer; since recipients of messages can discover their sender, they can direct replies to his reply buffer. Such a solution is safer than attempting to use one buffer for messages and their replies without the knowledge of the buffer implementation mechanism, because in that situation deadlocks can arise when buffers get full.

7.4 Summary

The designer of concurrent CHILL programs is urged to adopt a uniform approach based either on the in-process or the out-of-process model for achieving concurrency. If he selects the in-process model all his synchronisation requirements (mutual exclusion and resource allocation) can be achieved in a straightforward manner by using only SIGNALs and MODULEs. For the out-of-process model all synchronisation requirements (resource allocation and interprocess communication) can be met by using BUFFERs and PROCESSes. Resort to the use of the problem-ridden REGIONs and EVENTs should be completely avoided.

8. REFERENCES

1. Saltzer, J.H., "Traffic Control in a Multiplexed Computer System", Massachusetts Institute of Technology report MAC-TR-30, 1966.

2. Dennis, J.B. and Van Horn, E.C., "Programming Semantics for Multiprogrammed Computations", Communications of the A.C.M., Vol. 9, No. 3, pp 143-155, 1966.

3. Horning, J.J. and Randell, B., "Process Structuring", A.C.M. Computing Surveys, Vol. 5, No. 1, pp 5-30, 1973.

4. Brinch Hansen, P., "Operating System Principles", Prentice Hall, Englewood Cliffs, 1973.

5. Dijkstra, E.W., "Cooperating Sequential Processes" in Programming Languages, ed. F. Genuys, Academic Press, N.Y. pp 43-112, 1968.

6. Keedy, J.L. and Ramamohanarao, K., "A Job Management Model for In-Process Systems", Dept. of Computer Science, Monash University, 1979.

7. Ramamohanarao, K., "A New Model for Job Management Systems", Ph.D. Thesis, Dept. of Computer Science, Monash University, 1980.

8. Lauer, H.C. and Needham, R.M., "On the Duality of Operating System Structures", A.C.M. Operating Systems Review, Vol. 13, No. 2, pp 3-19, 1979.

9. Keedy, J.L., "A Comparison of Two Process Structuring Methods", Dept. of Computer Science, Monash University, 1979a.

10. Dijkstra, E.W., "Hierarchical Ordering of Sequential Processes", Acta Informatica, Vol. 1, pp 115-138, 1971.

11. Hoare, C.A.R., "Monitors: An Operating System Structuring Concept", Communications of the A.C.M., Vol. 17, No. 10, pp 549-557, 1974.

12. Brinch Hansen, P., "The Purpose of Concurrent Pascal", Proceedings of the International Conference on Reliable Software, in A.C.M. Sigplan Notices, Vol. 10, No. 6, pp 305-309, 1975.

13. Lister, A.M. and Maynard, K.J., "An Implementation of Monitors", Software-Practice and Experience, Vol. 6, No. 3, pp 377-385, 1976.

14. Keedy, J.L., "On Structuring Operating Systems with Monitors", Australian Computer Journal, Vol. 10, No. 1, pp 23-27, 1978.

15. Courtois, P.J., Heymans, F. and Parnas, D.L., "Concurrent Control with 'Readers' and 'Writers' ", Communications of the A.C.M., Vol. 14, No. 10, pp 667-668, 1971.

16. Keedy, J.L., Rosenberg, J. and Ramamohanarao, K., "On Synchronising Readers and Writers with Semaphores", Dept. of Computer Science, Monash University, 1981.

17. Keedy, J.L., Ramamohanarao, K. and Rosenberg, J., "On Implementing Semaphores with Sets", The Computer Journal, Vol. 22, No. 2, pp 146-150, 1979.

18. Tsichritzis, D.C. and Bernstein, P.A., "Operating Systems", Academic Press, N.Y., 1974.

19. Keedy, J.L., "A Problem with the Test and Set Instruction", A.C.M. Operating Systems Review, Vol. 13, No. 4, p. 1, 1979c.

20. Keedy, J.L., "An Outline of the ICL2900 Series System Architecture", Australian Computer Journal, Vol. 9, No. 2, pp 53-62, 1977.

21. Digital Equipment Corporation, "VAX 11-780 Architecture Handbook", Digital Equipment Corporation, 1977.

22. Denning, P.J. and Dennis, T.D., "On Minimizing Contention at Semaphores", A.C.M. Operating Systems Review, Vol. 14, No. 2, pp 9-16, 1980.

23. Keedy, J.L., "On the Programming of Device Drivers for In-Process Systems", Dept. of Computer Science, Monash University, 1979b.

24. Keedy, J.L. and Rosenberg, J., "On the Handling of Low Level System Processes", Dept. of Computer Science, Monash University, 1979.

25. Hoare, C.A.R., "Communicating Sequential Processes", Communications of the A.C.M., Vol. 21, No. 8, pp 666-677, 1978.

26. Brinch Hansen, P., "Distributed Processes: A Concurrent Programming Concept", Communications of the A.C.M., Vol. 21, No. 11, pp 934-941, 1978.

27. Parnas, D.L., "On the Criteria to be Used in Decomposing Systems into Modules", Communications of the A.C.M., Vol. 15, No. 12, pp 1053-1058, 1972.

28. Keedy, J.L., "Software Engineering", Australian Computer Society, 1980.

29. Parnas, D.L., "The Non-problem of Nested Monitor Calls", A.C.M. Operating Systems Review, Vol. 12, No. 1, pp 12-14, 1978.

BIOGRAPHY

LESLIE KEEDY was born in 1940 in Leeds, England. After undergraduate study at Kings College, London and graduate study at Mainz University in West Germany, he obtained a D.Phil. at Trinity College, Oxford in 1968. He then worked for International Computers Limited in the United Kingdom on the design of various operating systems and in Germany as a technical consultant. In 1974 he moved to the Department of Computer Science, Monash University where he was a Senior Lecturer and a chief investigator of the MONADS projects. Currently he is Professor and head of the research group for operating systems and software engineering at Technical University of Darmstadt, West Germany. Dr Keedy is a Fellow of the Australian Computer Society.

Phase-Conjugate Wavefront Generation In Four-Wave Mixing with Photorefractive $Bi_{12}GeO_{20}$ (BGO) Crystals

Y.H. JA

Telecom Australia Research Laboratories

Phase-conjugate wavefront generation by degenerate four-wave mixing experiments in a reflection geometry with photorefractive $Bi_{12}GeO_{20}$ crystals is reported. The phase volume hologram responsible for the phase-conjugate wavefront generation is a reflection hologram with a very small fringe spacing of about 0.26 μm. No external electric field is applied and electron diffusion is the main mechanism responsible for the formation of the hologram. The theoretical and experimental results for wavefront reflectivity as a function of the intensity ratio of the writing beams are given. The dependence of the modulation depth of the reflection hologram and its average value over the crystal on the absorption and the intensity ratio of the writing beams is investigated also.

1. INTRODUCTION

The newly-emerged field of so-called phase-conjugate optics has received much attention in recent years and numerous papers have been published (Ref. 1). The main feature of this field is the generation of an electro-magnetic wave with a spatial phase distribution which is, at every point in space, the exact opposite of that of an arbitrary incoming monochromatic wave. In other words, the generated wave is the phase-conjugate of the incoming wave.

Of all the different techniques used to generate a phase-conjugate wave, degenerate four-wave mixing (FWM) is considered to be the most advantageous, since the phase-matching condition is automatically satisfied in FWM. At present there are a great number of nonlinear optical materials which have been used in FWM to generate a phase-conjugate wave, such as vapour sodium, CS_2, nitrobenzine, $Bi_{12}SiO_{20}$(BSO), $Bi_{12}GeO_{20}$(BGO), $LiNbO_3$ and $BaTiO_3$ etc. The first three of these, i.e. atomic vapour and anisotropic molecular liquids are hazardous to handle while the photorefractive crystals BSO, BGO, $LiNbO_3$ and $BaTiO_3$ are not. Therefore the latter four have greater potentials for practical applications. In this paper we deal with one of them, i.e. BGO, which so far has not attracted the same attention as BSO, its isomorph.

Peltier and Micheron (Ref. 2) have found that the photosensitivity of BGO is about four times lower than that of BSO in conventional volume hologram recording at the initial stage where diffraction efficiency increases with exposure (= incident light power x exposure time). However, BGO may not behave similarly at the saturation stage where the diffraction efficiency reaches the maximum and is independent of the exposure, because photosensitivity depends on the response time of the recording medium, whereas wavefront reflectivity (Equation 8) at the saturation stage does not. It is now known that the response time of BGO is longer than that of BSO (Ref. 3).

In nearly all of the FWM experiments using photorefractive crystals to generate phase-conjugate wavefronts, holograms formed inside the crystals are transmission holograms, i.e. the reference and the object beams are incident from the same side of the recording medium (see Fig. 1a). In most cases (Refs. 2, 4-7), the angle 2θ between the reference and object beams is small, because the corresponding low spatial frequency makes the drift of photo-carriers under an applied electric field dominant and this significantly enhances the wavefront reflectivity. Only at very high spatial frequencies does the diffusion of photocarriers become dominant. In this paper, we report the theoretical and experimental results of a FWM experiment in a BGO crystal, using the same FWM configuration (Fig. 1a) as other workers (Refs. 4-5 and 8-9). However, it is found that a reflection hologram, where the reference and object beams are incident from opposite sides of the medium, is responsible for the phase-conjugate wavefront generation.

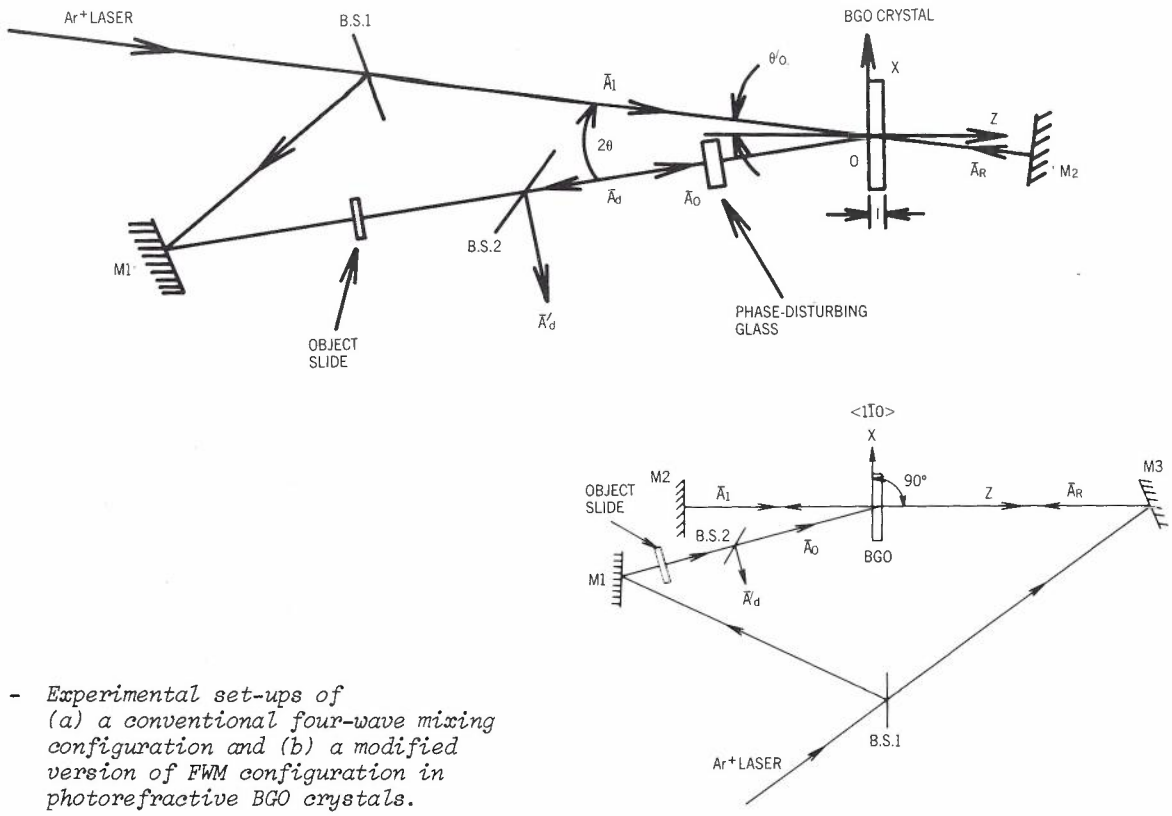


Fig. 1 - Experimental set-ups of (a) a conventional four-wave mixing configuration and (b) a modified version of FWM configuration in photorefractive BGO crystals.

The FWM technique to generate a phase-conjugate wave is very versatile and has potential applications to:

(a) the removal of aberrations caused by phase distorting media or in communication channels such as optical fibres and other optical waveguides,

(b) optical signal beam amplification,

(c) real-time optical data processing (e.g. spatial convolution or correlation, object edge enhancement, image subtraction and addition, etc.),

(d) self-referencing interferometry and

(e) real-time holography including holographic interferometry.

Several of the above applications namely, self-referencing interferometry (Ref. 10), object edge enhancement (Ref. 11), real-time image subtraction and real-time double-exposure holographic interferometry (Ref. 3), which have been performed using FWM in a reflection geometry with BGO crystals for the first time, will not be discussed and only the correction of aberrations caused by phase distorting media will be discussed in some detail in the latter part of this paper.

2. THEORY

Fig. 1a shows the conventional FWM configuration, where the reference beam $\bar{A}_1$ and the object beam $\bar{A}_0$ impinge on the crystal from the left-hand side, while the retroreflective reading beam $\bar{A}_R$ obtained from the mirror M_2 illuminates the crystal from the other side. In our reflection type, the roles of $\bar{A}_1$ and $\bar{A}_R$ are exchanged, i.e. $\bar{A}_R$ becomes the reference beam and $\bar{A}_1$ the reading beam.

For low beam coupling, low diffraction efficiency and H-mode incident beams, whose polarization vectors are all perpendicular to the incident plane, and ignoring multiple internal reflections, the total light power inside the crystal can be written as follows:

$$I_+ = |\bar{A}'_c + \bar{A}'_1 + \bar{A}'_R|^2$$

$$= I_1 (1-R)e^{-\alpha Z} + I_1 (1-R)\beta e^{-\alpha Z} +$$

$$I_R (1-R)e^{-\alpha(\ell-Z)} + 2|\bar{A}'_0| |\bar{A}'_1| \cos K_1 X +$$

$$2|\bar{A}'_0| |\bar{A}'_R| \cos K_2 Z + JJ \qquad (1)$$

where $\bar{A}'_0$, $\bar{A}'_1$ and $\bar{A}'_R$ are the complex amplitudes of the three incident beams inside the crystal respectively, R and α the power reflectance and the intensity absorption coefficient of the

crystals respectively, ℓ the crystal thickness, β ($= I_0/I_1$) the intensity ratio of the object and reading beams and K_1 and K_2, the amplitude of the grating vectors. Equation (1) assumes that these three beams add coherently to each other and that the angle between the bisector of the beams $\bar{A}_1$ and $\bar{A}_0$ and the Z axis (Fig. 1a) is very small. The last term JJ in Equation (1), representing the interaction of the beams $\bar{A}_1$ and $\bar{A}_R$, is of no interest to us since it produces no spatial modulation inside the crystal (Ref. 12).

When no external electric field is applied, grating K_2 (a reflection hologram grating) has a much higher spatial frequency than that of the transmission hologram grating K_1, consequently the diffusion field of the former will be much larger than that of the latter (Ref. 5) and so is the wavefront reflectivity. Therefore, of these two gratings, K_2 is dominant and is the main concern and will be studied in this paper.

Retaining the terms of interest and using the relation

$$I_R = (1-R)^2 e^{-\alpha\ell} I_1 \qquad (2)$$

Equation (1) becomes

$$I_+ = I_1(1-R)[e^{-\alpha Z} + \beta e^{-\alpha Z} + (1-R)^2 e^{-\alpha(2\ell-Z)}](1+M\cos K_2 Z) \qquad (3)$$

where

$$M = 2\sqrt{\beta}\,(1-R)e^{-\alpha\ell} / [e^{-\alpha Z} + \beta e^{-\alpha Z} + (1-R)^2 e^{-\alpha(2\ell-Z)}] \qquad (4)$$

is the modulation depth of the hologram grating K_2 and a function of Z.

The average value of M over the crystal thickness is easily obtained from (4) as follows

$$\bar{M} = \frac{1}{\ell}\int_0^{\ell} M\,dZ = \frac{2}{\alpha\ell}\left(\frac{\beta}{1+\beta}\right)^{\frac{1}{2}}\left[\tan^{-1}\left[\frac{1-R}{(1+\beta)^{\frac{1}{2}}}\right] - \tan^{-1}\left[\frac{(1-R)e^{-\alpha\ell}}{(1+\beta)^{\frac{1}{2}}}\right]\right] \qquad (5)$$

For a small $\alpha\ell$, $\bar{M}$ is only slightly different from the modulation depth

$$M_2 = 2\sqrt{\beta}\,(1-R)e^{-\frac{1}{2}\alpha\ell} / [1+\beta+(1-R)^2 e^{-\alpha\ell}] \qquad (6)$$

which was obtained previously (Ref. 13) by ignoring the absorption of the incident beams inside the crystal (see Fig. 2). Fig. 2 shows the dependence of $\bar{M}$ and $\bar{M}/M_2$ on β and $\alpha\ell$, computed from Equations 5 and 6. It can be seen from

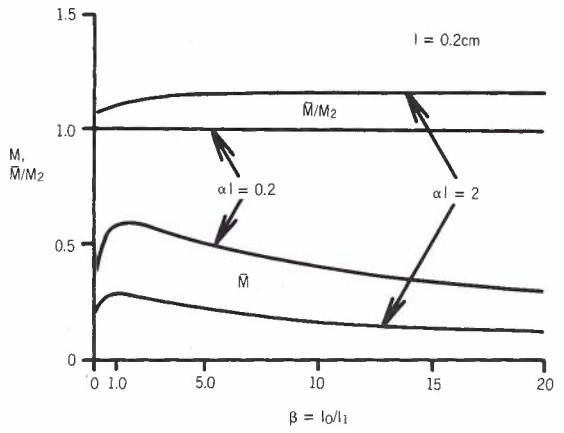


Fig. 2 - The average modulation depth of the phase volume reflection hologram $\bar{M}$, and the ratio of $\bar{M}$ to M_2, the modulation depth when ignoring the absorption of the incident beams, as functions of the intensity ratio β, with $\alpha\ell$ as the parameter.

Fig. 2 that $\bar{M}$ reaches the maximum at $\beta \simeq 1$ (but not at $\beta=1$) and the position of $\bar{M}_{max}$ changes with $\alpha\ell$ and that $\bar{M}$ decreases with an increasing $\alpha\ell$. For a large $\alpha\ell$ (say 2), the difference between $\bar{M}$ and M_2 may not be negligible. Fig. 3 shows the

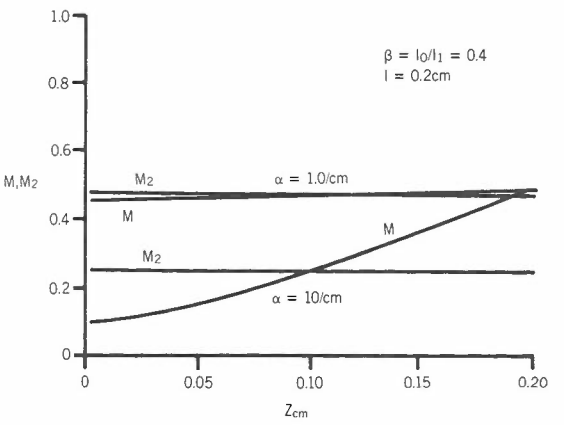


Fig. 3 - M and M_2 as functions of Z, the depth inside the crystal.

difference between M and M_2 at different depths inside the crystal for $\beta=0.4$ and $\ell=0.2$ cm with α as the parameter. From Fig. 3 it is obvious that M is a nonlinear function of Z, therefore $\bar{M}$, its average value over Z can be larger than M_2 (Fig. 2). Note that when θ_0, the incident angle of the reading beam $\bar{A}_1$ inside the crystal, is not small, α should be replaced by $\alpha/\cos\theta_0$ in Equations 1-6, assuming the incident beams $\bar{A}_0$ and $\bar{A}_1$ impinge symmetrically on the crystal.

In Fig. 1a, the reading beam $\bar{A}_1$ is diffracted by the reflection grating K_2 and a fourth wave $\bar{A}_d$, which is the phase-conjugate of $\bar{A}_0$, is generated and travels along the opposite direction to $\bar{A}_0$ with an intensity

$$I_d = n_r' \; I_1 \; (1-R)^2 \qquad (7)$$

where n_r' is the diffraction efficiency of the phase grating. The phase-conjugate wavefront reflectivity is defined as (Ref. 4).

$$w = \frac{I_d'}{I_0} = R_{BS} \frac{I_d}{I_0} = R_{BS} \frac{(1-R)^2}{\beta} \; n_r' \qquad (8)$$

where R_{BS} is the reflectivity of beam splitter B.S.2 in Fig. 1a.

The diffraction efficiency of a lossy phase volume grating of the reflection type has been given by Kogelnik (Ref. 14) as:

$$n_r = \left[\frac{\xi}{\nu} + \left(1 + \frac{\xi^2}{\nu^2}\right)^{\frac{1}{2}} \coth \left(\xi^2 + \nu^2\right)^{\frac{1}{2}} \right]^{-2} \qquad (9)$$

where

$$\nu = \frac{j\pi n_1 \ell}{\lambda \sqrt{C_R C_S}}$$

$$\xi = \frac{\alpha \ell}{4 \cos\theta_0} \left(1 - \frac{C_R}{C_S}\right)$$

$$j = \sqrt{-1}$$

$$C_R = \cos\theta_0$$

$$C_S = \cos\theta_0 - \frac{K\lambda}{2\pi n} \cos\phi$$

λ is the light wavelength in free space, n_1 the amplitude of the spatial modulation of the refractive index, n the average refractive index of the crystal, C_R and C_S the slant factors, ϕ the angle between the grating vector $K(|K| = 2\pi/\Lambda)$ and the axis Z which is the normal to the crystal surface (Fig. 1a) and Λ the fringe-spacing.

Equation (9) is obtained from a static theory in which the writing process is not considered (Ref. 14). Considering the writing process and for low efficiency, the diffraction efficiency n_r' of a reflection grating is also proportional to the square of the average modulation depth $\bar{M}$ (or M_2 if $\alpha\ell \ll 1$) as in the case of a transmission grating (Ref. 4).

In Equation (9), n_1 can be expressed as:

$$n_1 = \frac{n^3 r E_e}{2} \qquad (10)$$

where

$$E_e = \frac{E_T}{1 + E_T/E_q} \qquad (11)$$

and for linear recombination of the photo-carriers (F=1)

$$E_T = \frac{kT}{e} \frac{2\pi}{\Lambda}, \quad E_q = \frac{e}{\varepsilon_0 \varepsilon_r 2\pi} N\Lambda = B\Lambda \qquad (12)$$

and $B = 7 \times 10^7$ V/cm^2 for BSO (Ref. 4). In Equations (10-12), ε ($= \varepsilon_0 \varepsilon_r$) is the static dielectric constant of the crystal, N the concentration of trapping centres, e the electronic charge, k the Boltzmann constant, T the temperature, r the appropriate electro-optic coefficient (in our case $r = r_{41}$ (see Section 3)), E_e the effective field, E_T the diffusion field and E_q the maximal field of the volume space charge (Refs. 4, 15). Equations (10-11) are obtained by inferring from the formulae in Refs. 4 and 15 in which the dynamic theory of Kukhtarev et al. (Ref. 16) has been adapted and verified by experiments.

Using Equations (8-9) and the relation of $n_r' \propto \bar{M}^2$ finally we have

$$w = R_{BS} \frac{F^2 (1-R)^2}{\beta} \left[\frac{\xi}{\nu} + \left(1 + \frac{\xi^2}{\nu^2}\right)^{\frac{1}{2}} \coth \left(\xi^2 + \nu^2\right)^{\frac{1}{2}} \right]^{-2} \bar{M}^2 \qquad (13)$$

If $\nu \ll \xi \ll 1$ (as applies in our case), (13) becomes

$$w \propto \frac{1}{\beta} \; n_1^2 \; M^2 \qquad (14)$$

which has the same form as that in a transmission geometry (Ref. 5). However, (14) may not be valid for a large θ_0.

3. EXPERIMENTAL RESULTS AND APPLICATIONS

The experimental set up is also shown in Fig. 1a. In our experiment, the object and reading beams are expanded laser beams with small divergence obtained from an Argon-ion laser (λ = 0.5145 μm) by using the beam splitter B.S.1 and the mirror M_1. Their powers are 3.9 and 9.2 mW, respectively. Their polarization vectors are parallel to the crystallographic direction <110> (the Y axis in Fig. 1a). The optical beam directions and the bisector of the beams $\bar{A}_1$ and $\bar{A}_0$ are very close to the crystallographic direction <001> (the Z axis in Fig. 1a), i.e., the angle 2θ (11.9°) between $\bar{A}_1$ and $\bar{A}_0$, and the angle between the grating vector $\bar{K}_2$ and the Z axis (=4.1°) are all small. In other words, the angle between the object beam $\bar{A}_0$ and the reference beam $\bar{A}_R$ is large and close to 180°. Referring to Equation (13) and the argument in Section 2, the diffusion field generated by grating K_2 will be much higher than that of K_1, consequently the wavefront reflectivity of grating K_2 will be much larger than that of K_1.

The generation of the phase-conjugate wavefront of an object wave can be confirmed experimentally by inserting phase-disturbing glass between the BGO crystal and the beam

splitter B.S.2 as shown in Fig. 1a. The object (character E) slide is placed between the mirror M_1 and the beam splitter B.S.2. Since the generated phase-conjugate wavefront travels back through the phase-disturbing glass, phase distortion can then be removed. This can be demonstrated clearly as follows. On entering the disturbing glass the object wave can be expressed as

$$\bar{A}_0 = \bar{A}(x,y)e^{-jkZ} = |\bar{A}_0(x,y)|e^{-j[\psi(x,y)+kZ]} \quad (15)$$

where a factor of $e^{-j\omega t}$ has been dropped. On emerging from the glass, the object wave becomes

$$\bar{A}_0' = |\bar{A}_0(x,y)|e^{-j[\psi(x,y)+\Delta(x,y)+kZ]} \quad (16)$$

where $\Delta(x,y)$ denotes the phase distortion caused by the glass. Assuming all beams are paraxial beams and $|\bar{A}_1|^2, |\bar{A}_R|^2 \gg |\bar{A}_0|^2, |\bar{A}_{pc}|^2$, then the generated phase-conjugate wave on entering the glass is

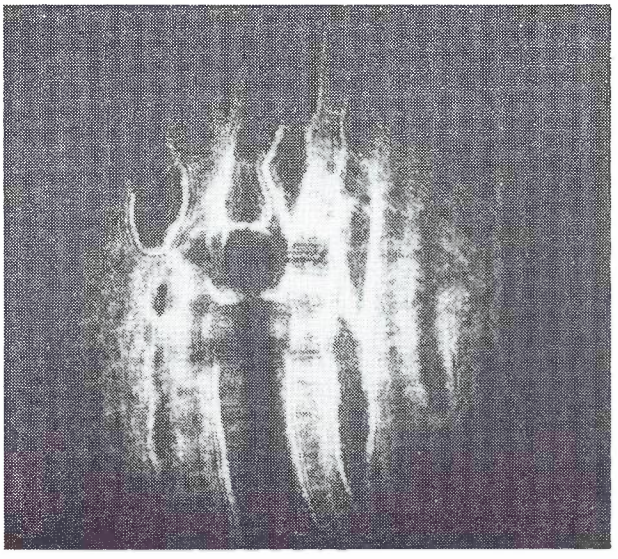
$$\bar{A}_{pc} = \bar{A}_0'^{*}C \propto |\bar{A}_0(x,y)C|e^{+j[\psi(x,y)+\Delta(x,y)+kZ]} \quad (17)$$

where * denotes complex conjugation and C is a complex constant.

Finally, the phase-conjugate wave emerging from the glass can be expressed as

$$\bar{A}_d \propto |\bar{A}_0(x,y)C|e^{+j[\psi(x,y)+\Delta(x,y)-\Delta(x,y)+kZ]} \propto |\bar{A}_0(x,y)C|e^{+j[\psi(x,y)+kZ]} \quad (18)$$

Comparing (18) with (15), it is obvious that the original phase $\psi(x,y)$ is restored. In other words, the phase distortion $\Delta(x,y)$ is removed. Some experimental results are shown in Fig. 4. Note that in Fig. 4 phase distortion is not completely removed because of the severity of the phase disturbance.



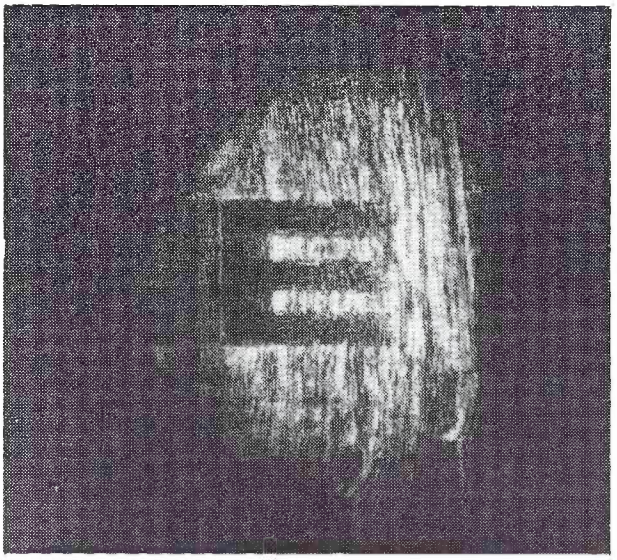


Fig. 4 - Reconstructed images (a) image after travelling through phase-disturbing glass (b) image with phase-distortion corrected by real-time phase-conjugate wavefront generation.

From Equation (13), it is easy to see that the wavefront reflectivity is a function of β (or $\beta' = I_0/I_R$), the intensity ratio of the object and reading beams. Fig. 5 shows the measured results of the relation between w and β^{-1}, and theoretical results by using Equation (13) are also shown for comparison where different values of the constant B in Equation (12) have been used. A good agreement between theory and experiment is obtained for $B=0.82 \times 10^8 V/cm^2$. This value of B is somewhat different from that obtained in Ref. 13 ($B = 0.90 \times 10^8 V/cm^2$), because here the measured intensity absorption coefficient $\alpha = 1.44/cm$ of the BGO sample at the operating wavelength $\lambda = 0.5145\ \mu m$ is used, instead of adopting the value of $\alpha = 2.1/cm$ from Ref. 2, as was done elsewhere (Ref. 13). It is believed that the new value of B is closer to the

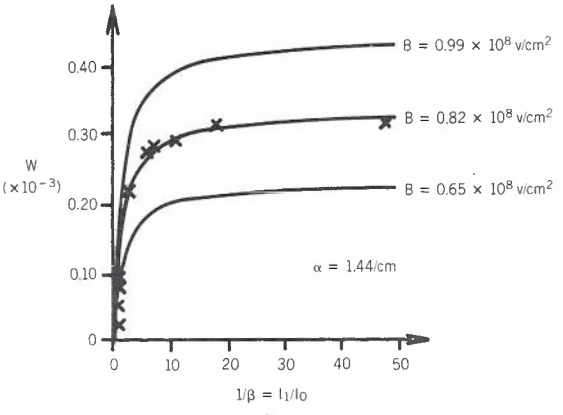


Fig. 5 - Relationship of wavefront reflectivity $w = I_d'/I_0$ with reading-to-object beam intensity ratio $I/\beta=I_1/I_0$. Crystal thickness $\ell=1.52$ mm and fringe-spacing $\Lambda=0.259\ \mu m$

Solid curves - theoretical results
Crosses - experimental results

true value than that obtained in Ref. 13. It can be seen from Fig. 5 that the wavefront reflectivity increases with B. The curve of w versus β^{-1} in Fig. 5 is quite similar to that in BSO crystals, e.g. Fig. 4 in Ref. 4. The wavefront reflectivity in our case also rapidly saturates for $I_1 > 10 I_0$. Comparing these two figures, one can see that the wavefront reflectivity of a reflection hologram in BGO (with no applied electric field) is comparable to that of a transmission hologram in BSO with an applied electric field of 7kV/cm (Note that our BGO crystal (ℓ = 1.52mm) is much thinner than the BSO crystal used in Ref. 4 (ℓ = 3.2mm)). One possible explanation is as follows: n_1, the saturation value of index modulation is directly related to N, the concentration of trapping centers in the crystal (Equations 10-12). The concentration of trapping centers in the BGO crystal we used ($N \approx 1.1 \times 10^{16}/cm^3$, obtained from the best fit between theory and experiment in Fig. 5, is close to that in the BSO crystal used in Ref. 4 ($N \approx 1.2 \times 10^{16}/cm^3$). It is known that N may differ from one sample of the BGO (or BSO) crystal to another (Ref. 17).

The BGO crystal we used measured 76.2 x 15.2 x 1.52 mm (initially intended for use as a surface acoustic wave device). Since no external electric field is applied, any part of the crystal surface can be illuminated. This is quite different from the case where the area between the electrodes should be uniformly illuminated in order to prevent the generation of a large scale space-charge field (Refs. 5, 18) when an external electric field is applied. As the edges of the crystal are not illuminated in our case, the signal to noise ratio SNR can increase considerably, therefore it may not be necessary to use a polarizer to suppress the unwanted noise (Ref. 19).

We have also used a modified version of the usual FWM configuration, shown in Fig. 1b, where the reference and object beams impinge on the crystal from opposite sides directly (Ref. 20). As usual, the reading beam is obtained by a reflective mirror placed behind the crystal. However the measured wavefront reflectivity is smaller than that of the configuration in Fig. 1a. It is most likely due to:

(i) the different intensity ratio of the pump beam $\bar{A}_1$ to $\bar{A}_R$: the recent work of Fisher *et al.* (Ref. 21) indicates that wavefront reflectivity increases with the ratio I_1/I_R. Referring to Figs. 1a and 1b, it is evident that I_1/I_R in Fig. 1a is larger than that in Fig. 1b, because of the absorption by the BGO crystal and the reflection at its surfaces.

(ii) the optical activity (OA) in the BGO crystal: BGO is a crystal with high OA (rotation of the polarization vector $\simeq 38^{\circ}$/mm at λ = 0.5145 μm)(Ref. 22). In the first configuration (Fig. 1a) the polarization vectors of $\bar{A}_0$ and $\bar{A}_R$ are always parallel to each other inside the crystal due to the compensation of OA. However in the second configuration (Fig. 1b) they are not parallel to each other inside the crystal, but make an angle of about 58° (crystal thickness ℓ = 1.52mm). Thus in the latter case w will be smaller.

The reflection hologram requires high optical and mechanical stability. Since the Argon-ion laser we use is without an etalon, its coherent length is quite short, and the optical path-length difference between the object beam $\bar{A}_0$ and the reference beam $\bar{A}_R$ should be minimal in order to obtain high wavefront reflectivity.

4. DISCUSSION

Equation (13) is only an approximate formula, due to the following factors:

(a) Equation (13) is obtained on the basis of the dynamic theory of volume holography recently developed by Kukhtarev *et al.* (Refs. 16, 23). However, a strict derivation from the complete set of material equations and nonlinear wave equations (Ref. 16), taking into account the specific boundary conditions of a reflection hologram is yet to be completed. When taking into account the absorption inside the crystal, it is difficult to obtain an analytical solution for the nonlinear wave equation involving four interacting waves. Even for two interacting waves, in general, numerical methods have to be used to obtain the desired solution (Ref. 24).

(b) Multiple Internal Reflections (MIR) inside the crystal have been ignored in our theory. Kogelnik (Ref. 25) and Ctyroky (Ref. 26) have pointed out that the diffraction efficiency can be affected by MIR. In fact, MIR has been observed in our experiment and it can be used as the reading beam (Ref. 20). Fig. 6 shows the fringe pattern caused by the interference between the phase-conjugate wave generated by the reading beam $\bar{A}_1$ and that generated by the first-order internal reflection (Ref. 20). Although the interference between these two generated phase-conjugate waves may be of little use, it demonstrates the existence of MIR and its role as the reading beam.

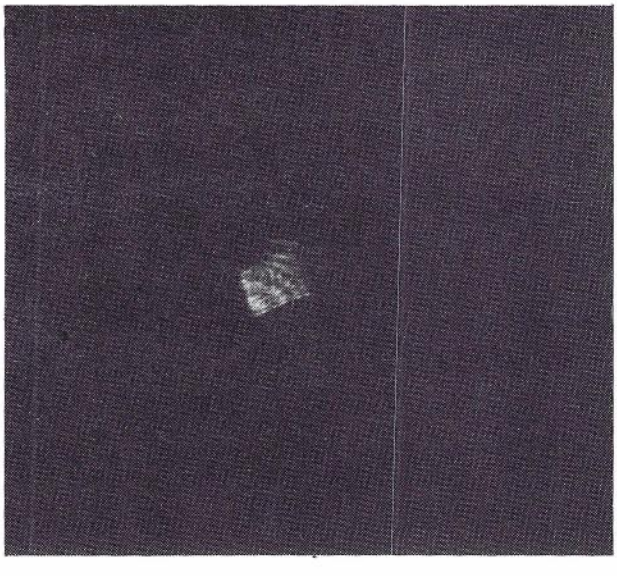


Fig. 6 - Observed fringe pattern caused by the interference between the two phase-conjugate waves generated by the reading beam $\bar{A}_1$ and the first-order internal reflection, respectively.

The following effects have also been neglected:

(1) the optical activity in the BGO crystal

(2) the increase of the temperature inside the crystal due to light absorption and

(3) the non-planar characteristic of the incident beam (Gaussian beam with small divergence).

The extremely small fringe-spacing ($\Lambda \simeq 0.259\ \mu m$) of the reflection hologram grating K_2 means that it can be accurately described as a volume grating. Hence higher diffraction-orders (Ref. 27) have not been observed and the Bragg (volume) effect is strictly observed.

5. CONCLUSIONS

The generation of phase-conjugate wavefronts using the conventional FWM configuration of a reflection hologram type has been investigated and both theoretical and experimental results have been presented.

Electron diffusion (Ref. 28) is the main mechanism responsible for the generation of the phase volume hologram in a BGO crystal when no external electric field is applied. The reasonably high wavefront reflectivity of the reflection hologram type (comparable to that of BSO or BGO with a high intensity applied electric field) is mainly due to the high spatial frequency of the formed fringes ($\Lambda = 0.259\ \mu m$) and the relatively high concentration of trapping centres in the BGO crystal used in our experiment.

6. ACKNOWLEDGEMENTS

The author wishes to thank Dr W.J. Williamson and Mr G.E. Rosman for their stimulating and useful discussions, and Dr P.V.H. Sabine for providing the BGO crystals used in this experiment. Thanks are also due to Mr H. Alleaume for taking the photographs shown in this paper.

7. REFERENCES

1. Yariv, A., "Phase Conjugate Optics and Real-Time Holography", IEEE J. Quan. Electron., Vol. QE-14, 1978, pp. 650-660.
Pepper, D.M., "Nonlinear Optical Phase Conjugation", Opt, Eng., Vol. 21, 1982, pp. 156-183. Excellent and up-to-date review papers in which many important references are given.

2. Peltier, M. and Micheron, F., "Volume Hologram Recording and Charge Transfer Process in $Bi_{12}SiO_{20}$ and $Bi_{12}GeO_{20}$", J. Appl. Phys., Vol. 48 1977, pp. 3683-3690.

3. Ja, Y.H., "Real-Time Double-Exposure Holographic Interferometry in Four-Wave Mixing with Photorefractive $Bi_{12}GeO_{20}$ Crystals", Appl. Opt. Vol. 21, 15 Sept. 1982, pp. 3230-3231, and "Real-time Image Subtraction in Four-Wave Mixing with Photorefractive $Bi_{12}GeO_{20}$ Crystals", Opt. Commun. Vol. 42, 1982, pp. 377-380.

4. Huignard, J.P., Herriau, J.P., Aubourg, P. and Spitz, E., "Phase-Conjugate Wavefront Generation via Real-Time Holography in $Bi_{12}SiO_{20}$", Optics Lett., Vol. 4, 1979, pp. 21-23.

5. Huignard, J.P., Herriau, J.P., Rivet, G. and Gunter, P., "Phase-Configuration and Spatial-Frequency Dependence of Wavefront Reflectivity in $Bi_{12}SiO_{20}$ Crystals", Optics Lett., Vol. 5, 1980, pp. 102-104.

6. Marrakchi, A., Huignard, J.P. and Herriau, J.P. "Application of Phase Conjugation in $Bi_{12}SiO_{20}$ Crystals to Mode Pattern Visualisation of Diffuse Vibrating Structures" Opt. Comm., Vol. 34, 1980, pp. 15-18.

7. White, J.O. and Yariv, A., "Real-Time Image Processing via Four-Wave Mixing in Photorefractive Medium", Appl. Phys. Lett., Vol. 37, 1980, pp. 5-7.

8. Woerdman, J.P., "Formation of a Transient Free Carrier Hologram in Si", Opt. Comm., Vol. 2, 1970, pp. 212-214.

9. Huignard, J.P., Herriau, J.P. and Valentine, T. "Time Average Holographic Interferometry with Photoconductive Electrooptic $Bi_{12}SiO_{20}$ Crystals", Appl. Opt., Vol. 16, 1977, pp. 2796-2798.

10. Ja, Y.H., "Observation of Interference between a Signal and its Conjugate in a Four-Wave Mixing Experiment Using $Bi_{12}GeO_{20}$ Crystals", Opt. & Quan. Electron. Vol. 14, 1982, pp. 367-369.

11. Ja, Y.H., "Real-Time Edge Enhancement in Four-Wave Mixing with Photorefractive BGO Crystals", Opt. & Quan. Electron. Vol. 15, 1983 (in press).

12. Giuliano, C.R., "Applications of Optical Phase Conjugation", Phys. Today, Vol. 27, April 1981, pp. 27-35.

13. Ja, Y.H., "Phase-Conjugate Wavefront Generation via Four-Wave Mixing in $Bi_{12}GeO_{20}$ Crystals-Reflection Hologram Type", Opt. Commun., Vol. 41, 1982, pp. 159-163.

14. Kogelnik, H., "Coupled Wave Theory for Thick Hologram Gratings", Bell Syst. Tech. J., Vol. 48, 1969, pp. 2909-2947.

15. Krumins, A.E. and Gunter, P., "Photorefractive Effect and Photoconductivity in Reduced Potassium Niobate Crystals", Appl. Phys., Vol. 19, 1979, pp. 153-158.

16. Kukhtarev, N.V., Markov, V.B., Odulov, S.G., Soskin, M.S. and Vinetskii, V.L., "Holographic Storage in Electrooptic Crystals. I. Steady State:, Ferroelectrics, Vol. 22, 1979, pp. 949-960.

17. Ja, Y.H., "The Significance of Trapping Center Concentration in Photorefractive Crystals Used for Four-Wave Mixing" (submitted for publication).

18. Cornish, W.D., Moharam, M.G. and Young, L., "Effects of Applied Voltage on Hologram Writing in Lithium Niobate", J. Appl. Phys., Vol. 47, 1976, pp. 1479-1484.

19. Huignard, J.P., Herriau, J.P. and Aubourg, P., "Some Polarization Properties of Volume Holograms in $Bi_{12}SiO_{20}$ Crystals and Applications", Appl. Opt., Vol. 17, 1978, pp. 1851-1853.

20. Ja, Y.H., "Utilizing Internal Reflection as the Reading Beam in Four-Wave Mixing in Photorefractive $Bi_{12}GeO_{20}$ Crystals", (submitted for publication).

21. Fischer, B., Cronin-Golomb, M., White, J.O. and Yariv, A., "Amplified Reflection, Transmission and Self-Oscillation in Real-Time Holography", Opt. Lett., Vol. 6, 1981, pp. 519-521.

22. Lenzo, P.V., Spencer, E.G. and Ballman, A.A., "Optical Activity and Electrooptic Effect in Bismuth Germanium Oxide ($Bi_{12}GeO_{20}$)", Appl. Opt., Vol. 5, 1966, pp. 1688-1689.

23. Vinetskii, V.L. and Kukhtarev, N.V., "Theory of the Conductivity Induced by Recording Holographic Gratings in Nonmetallic Crystals", Sov. Phys. - Solid State Phys., Vol. 16, 1975, pp. 2414-2415.

24. Ja, Y.H., "Energy Transfer between Two Beams in Writing a Reflection Volume Hologram in a Dynamic Medium", Opt. & Quan. Electron., Vol. 14, 1982, pp. 547-556.

25. Kogelnik, H., "Bragg Diffraction in Hologram Gratings with Multiple Internal Reflections", J. Opt. Soc. Am., Vol. 57, 1967, pp. 431-433.

26. Ctyroky, J., "Coupled-Mode Theory of Bragg Diffraction in the Presence of Multiple Internal Reflections", Opt. Comm. Vol. 16, 1976, pp. 259-261.

27. Ja, Y.H., "Observation of Higher-Order Diffraction Components in Degenerate Four-Wave Mixing Experiments in $Bi_{12}GeO_{20}$ Crystals", Elec. Lett., Vol. 17, 1981, pp. 488-489.

28. Amodei, J.J., "Electron Diffusion Effects during Hologram Recording in Crystals", Appl. Phys. Lett., Vol. 18, 1971, pp. 22-24.

BIOGRAPHY

YU HONG JA graduated from the Department of Radio and Electronics of Peking University, Peking, China. After graduating, he spent a few years with Shangtung Institute of Marine Sciences, Tsingtao, Shangtung, China, teaching electronics and physics and researching into marine physics and electronics. He received the degree of Doctor of Philosophy in 1971 for his research work on Electron Paramagnetic Resonance in gem stones at the School of Electrical Engineering of the University of Sydney. Since then, he has joined the Australian Telecommunications Commission, Research Laboratories, Melbourne. He is the author of more than 26 published research papers, scattered over quite diverse fields such as electron paramagnetic resonance, optical and microwave holography, antennas and wave propagation, and phase-conjugate optics. His current research interests are optical holography, nonlinear optics, and optical image processing.

A Tutorial Paper On Medium Bit Rate Speech Coding Techniques

R.A. SEIDL

Telecom Australia Research Laboratories

This paper reviews waveform speech coding techniques that inherently are applicable at bit rates from 64 kbit/s down to 7.2 kbit/s. Topics covered include uniform, non-uniform and adaptive quantization, differential and adaptive differential coding systems which include delta modulation and differential PCM (DPCM). Prediction and adaptive prediction for DPCM and adaptive predictive coders are discussed. Two "frequency domain" coders, namely sub-band and adaptive transform coders have been included. Finally a brief summary of coder complexity and expected speech quality at various bit rates is presented.

1. INTRODUCTION

Currently, public telephone networks are becoming more digital in nature with the introduction of digital transmission and switching techniques which will lead initially to a telephony-based integrated digital network (IDN). The currently adopted speech coding standard algorithm (CCITT Recommendations G711, G712 (Ref. 1, Ref. 2)) is A-law 64 kbit/s log-PCM. Note that the above CCITT recommendations also include μ-law 64 kbit/s log-PCM (with μ = 255), but digital paths between countries which have adopted different encoding laws should carry signals encoded in accordance with the A-law.

Many telecommunications administrations are examining speech coding techniques with a view to providing digital speech channels of similar quality to 64 kbit/s log-PCM (as assessed subjectively) but at lower bit-rates. For the purposes of this paper medium bit-rate encompasses the range 32 to 9.6 kbit/s. Channels employing the lower bit-rate encoding techniques would find application in long haul trunk systems, in digital mobile telephony and in satellite communications where channel capacity is expensive. The adoption of lower bit-rate techniques could also be employed for the alleviation of network congestion during peak periods. These coding techniques will also facilitate the integration of voice and non-voice services over a 64 kbit/s channel.

Speech coding techniques may be classified into two different types. The first class of coding techniques is referred to as waveform coding. These techniques attempt to represent (via some suitable code) the speech signal waveform. The second class of speech coding techniques is referred to as source coding and these, in general, exploit the physical properties of the information source, that is the speech production mechanism. This paper only considers waveform coding techniques, since this class of technique is used for medium bit-rate coding schemes and produces encoded speech whose decoded perceived quality is superior to the latter class of coding technique (albeit that source coding techniques use bit-rates below 9.6 kbit/s in general).

2. DIGITAL REPRESENTATION OF SPEECH

2.1 Sampling

The general procedure for producing digital waveform representations is shown in Fig.1. The continuous time signal $x_a(t)$ is sampled periodically in time (with period T) to produce a sequence of samples $x(n) = x_a(nT)$. These samples can take on a continuum of values, and it is therefore necessary to quantize them to a finite set of values in order to obtain a digital representation, i.e. one that is discrete in both time and amplitude. The sampling frequency must be at least twice the highest frequency of the band limited analog input signal. For speech signals, this minimum sampling frequency is the Nyquist rate.

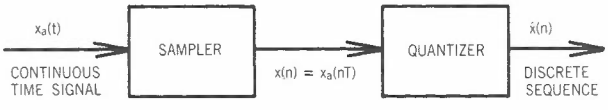


Fig.1 - General Digital Waveform Representations

2.2 Quantization

Quantization produces a sequence of discrete samples. An encoding procedure is subsequently carried out to represent each quantized sample by a code word (Fig. 2). A decoder would be used to transform these code words back into a sequence of quantized samples.

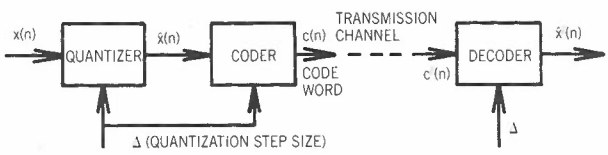


Fig.2 - Quantization and Coding

In most cases it is convenient to use binary numbers to represent the quantized samples. With B-bit code words it is possible to represent 2^B quantization levels. The information capacity required to transmit or store the digital representation is therefore:

$$I = B.F_s \text{ bit/s},$$

where F_s is the sampling frequency.

For a given speech bandwidth, F_s is fixed by the sampling theorem. The only way to reduce the bit-rate is to reduce the number of bits per sample. It is generally desirable to maintain the bit rate as low as possible whilst maintaining a required level of quality.

It is generally assumed that the samples $\{x(n)\}$ will fall in a finite range of amplitudes such that $|x(n)| \leq X_{MAX}$ for all n. The amplitudes of the samples are quantized by dividing the entire amplitude range into a finite set of amplitude ranges and assigning the same amplitude value (and hence code word) to all samples falling in a given range. The input/output relationships for two different uniform quantization step-size quantizers (3 bit) are depicted in Fig.3.

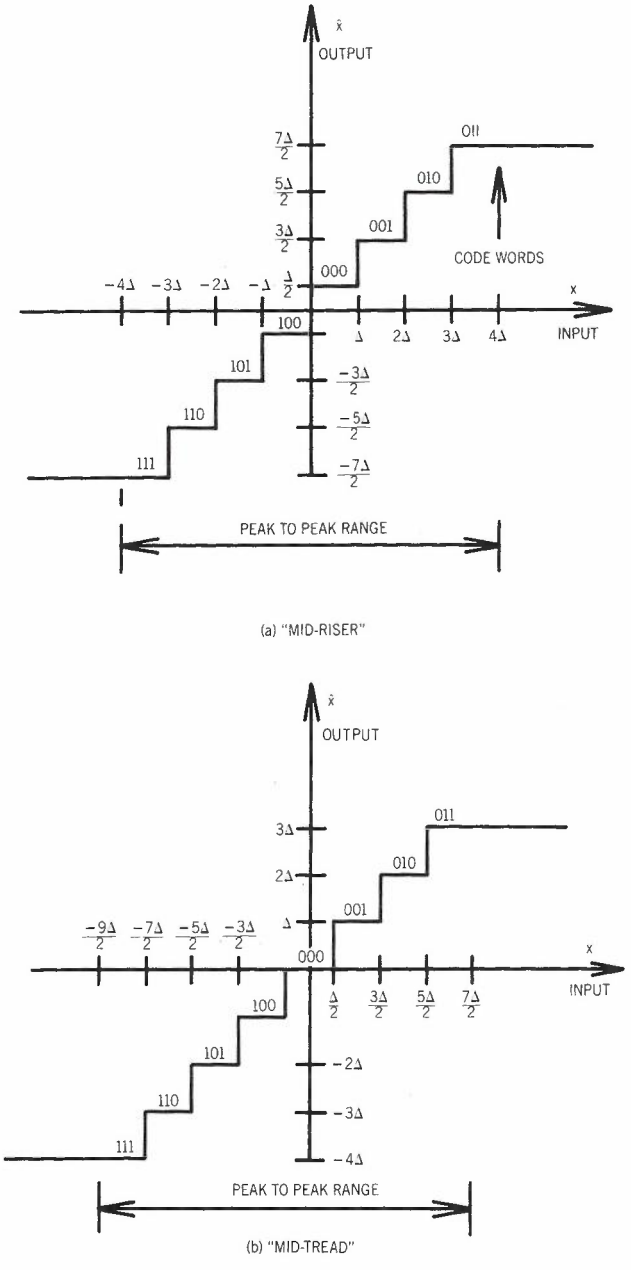


Fig.3 - 3-bit Uniform Quantizer Characteristics

Figure 3(a) is referred to as a "mid-riser" characteristic. This is a symmetric characteristic with the same number of positive and negative codes but no true zero representation. The "mid-tread" characteristic (Fig.3b) does have a "zero" code but it is asymmetric in that there is one more negative code than positive.

For uniform quantizers the fidelity of representation of the input signal is level dependent and is only optimum when the signal amplitude achieves the maximum possible excursion, otherwise it is equivalent to using a coder with fewer bits. Therefore to maintain a high quality representation of speech signals with a uniform quantizer it is generally necessary to use more bits than might be implied by any signal-to-noise ratio (SNR) analysis. At least 11 bits are required for speech signals.

2.3 Non-Uniform Quantization

For the above reasons, it would be very desirable to have a quantizer for which the SNR was independent of signal level. That is, rather than the quantization error being of constant variance independent of signal amplitude as for a uniform quantizer, it would be desirable to have a constant percentage error. The quantization error is the difference between the sample value and its quantized value; i.e. $e(n) = x(n) - \hat{x}(n)$). This can be achieved by using a non-uniform quantizer whose quantization levels are logarithmically spaced (Ref. 3).

Log-PCM coders approximate such a logarithmic compression characteristic by segmenting the input signal range into a number of intervals. Within each interval the quantization step size is uniform but it changes logarithmically between segments. These quantizers attempt to achieve constant SNR over a wide range of signal levels.

In cases where the signal variance is known, it is possible to choose the quantizer levels so as to minimize the quantization error variance and thus minimize the SNR. This problem is discussed by Max (Ref. 4) and Paez and Glisson (Ref. 5). Non-uniform quantizers using Gaussian and Laplacian statistics to model speech amplitude densities are derived in the latter. Although optimum quantizers yield minimum mean-square-error when matched to the amplitude distribution of the signal, the non-stationary nature of the speech production process yields less than satisfactory results.

2.4 Adaptive Quantization

Adaptive quantizers allow the step size Δ to vary so as to match the time dependent variance of the input signal. This is usually referred to as adaptive PCM (Fig.4a). An

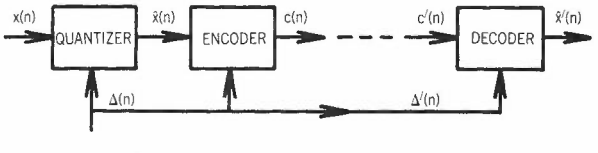


Fig.4(a) - Adaptive Quantization - Variable Step Size

alternative approach is to use a fixed quantizer characteristic in conjunction with a time varying gain which attempts to keep the variance of the input to the quantizer constant (Fig. 4b).

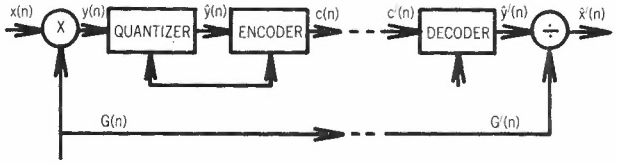


Fig.4(b) - Adaptive Quantization - Variable Gain

For variable step-size quantizers, the step size should increase and decrease with the variance of the input. If the quantizer were non-uniform, the quantization levels and ranges would be scaled linearly. In the variable gain case, the gain should change inversely as the variance of the input thus keeping the variance of the quantizer input constant.

There are two classes of adaptation for adaptive quantizers. In the first case, the step size or gain changes are determined from the input signal itself (x(n)). This is referred to as feed-forward adaptation. The other class of quantizers determines its adaptation on the basis of the output of the quantizer ($\hat{x}(n)$). This is referred to as feedback adaptation.

Schemes employing feed-forward adaptation require the step size or gain to be transmitted along the channel to the decoder in addition to the codeword. Feedback adaptation schemes do not require the additional information to be transmitted. However, they suffer from increased sensitivity to errors in code words.

Adaptation may occur on a sample by sample basis, in which case it is referred to as instantaneous adaptation, or at a slowly varying rate (e.g. on a phonemic - the basic unit of speech sound - basis) where it is referred to as syllabic adaptation.

Optimum step-size multipliers for adaptive quantization are discussed by Jayant (Ref. 6). Adaptive differential pulse code modulation (ADPCM) coding of speech is discussed by Cumminskey *et al.* (Ref. 7), with particular emphasis upon adaptive quantization.

3. DIFFERENTIAL CODING

3.1 Differential Quantization

Differential coding schemes exploit the correlation between adjacent signal samples. The meaning of this high correlation is that, in an average sense, the signal does not vary rapidly between samples, and therefore the variance of the difference signal should be lower than that of the signal itself.

Simply, in differential coding the differences between adjacent input signal samples are quantized, and the reconstructed estimate of the input is obtained by essentially integrating these quantized difference samples.

A differential coder is depicted in Fig.5.

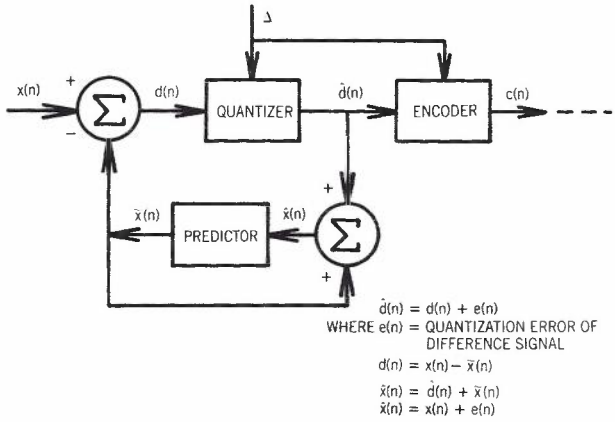


Fig.5(a) - Differential Quantization Encoder

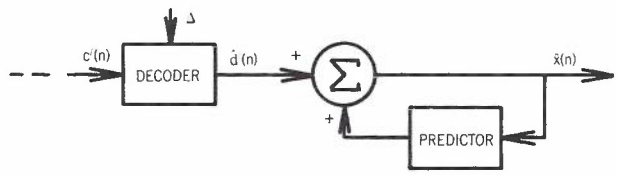


Fig.5(b) - Differential Quantization Decoder

The difference signal d(n) is given by:

$$d(n) = x(n) - \tilde{x}(n),$$

and is the difference between the unquantized input sample x(n) and a prediction of the input sample $\tilde{x}(n)$. This is also referred to as the prediction error signal.

It can easily be shown that the predictor input $\hat{x}(n)$ is a quantized version of the input signal x(n), and, independently of the predictor characteristics, it differs from the input only by the quantization error of the difference signal (see Fig.5a). This means that if the prediction process is good, the variance of the difference signal d(n) will be smaller than that of the input signal x(n), and consequently a quantizer with a given number of levels will give a smaller quantization error than would be possible by direct quantization. Alternatively, a quantizer with a lower number of bits could be used to yield the same quality.

The amount of improvement of differential quantization over direct quantization is dependent upon the amount of correlation between signal samples. Differential coding schemes may use a fixed or adaptive quantizer to encode the difference signal. For differential coders a predictor of the form:

$$\tilde{x}(n) = \sum_{k=1}^{p} \alpha_k \hat{x}(n-k)$$

is used. This is referred to as a linear predictor since the signal estimate $\tilde{x}(n)$ is a linear combination of previous predictor inputs and p is the order of the predictor. The predictor coefficients $\{\alpha_k$'s$\}$ are chosen,

in general, to minimise the variance of the difference signal with respect to each of the coefficients (Ref. 9, Ref. 8).

Differential coding schemes may use fixed or adaptive predictors (where the predictor coefficients are updated at a syllabic rate), and may use either fixed or adaptive quantizers.

3.2 Delta Modulation (DM)

Delta modulation systems are special examples of differential encoding schemes. For such schemes the sampling rate is many times the Nyquist rate for the input signal so that the correlation between adjacent samples is very high and, in general, the difference signal is very small. A one-bit quantizer is used in conjunction with a fixed first order predictor (α = constant). Fig.6 illustrates a delta modulator.

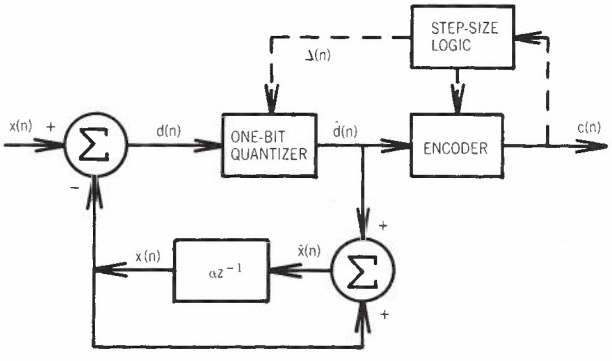


Fig.6(a) - Delta Modulation Encoder

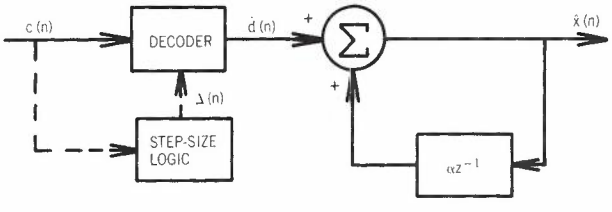


Fig.6(b) - Delta Modulator Decoder

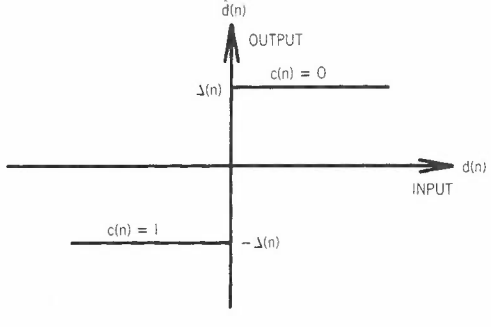


Fig.6(c) - Delta Modulator Quantizer Characteristics

Note that with $\alpha = 1$, the predictor forms the digital equivalent to integration. If α is less than one, it is sometimes referred to as a "leaky" integrator.

A linear delta modulator (i.e. one without step-size adaptation to accurately track the input signal) requires that the difference signal should not exceed the step-size in the region of maximum slope of the input signal, as otherwise the reconstructed signal will fall behind the input signal. This condition is called "slope overload" and the resulting quantization error is called slope overload distortion. This problem may be circumvented by adopting a step adaptation quantizer. Slope overload is depicted in Fig.7.

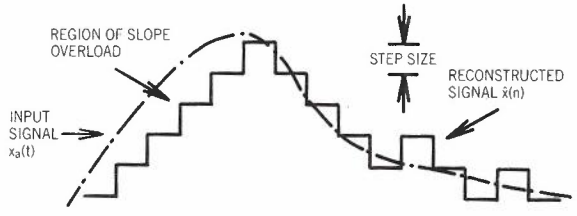


Fig.7(a) - Delta Modulator - Fixed Step Size

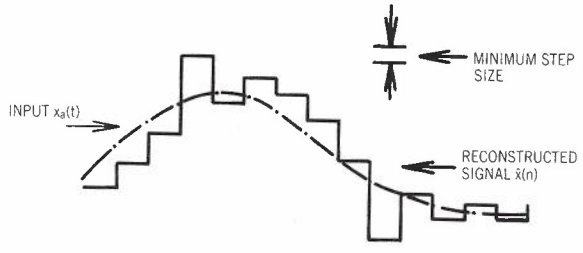


Fig.7(b) - Delta Modulator - Adaptive Step Size

For delta modulation systems, when the input is zero (idle channel condition) the output of the quantizer is an alternating sequence of 1's and 0's, and the resultant reconstructed signal will alternate about zero with a peak-to-peak variation equivalent to the step size for linear DM systems or the minimum step size for adaptive DM systems. This latter type of quantization error is referred to as granular noise. A detailed study of DM can be found in reference 10.

Jayant (Ref. 11) discusses an Adaptive Delta Modulation (ADM) system where the step-size adaptation algorithm is given by:

$$\Delta(n) = M\Delta(n-1) \quad : \quad \Delta_{min} \leq \Delta(n) \leq \Delta_{max}$$

where

$M = P > 1$ if $c(n) = c(n-1)$

$M = Q < 1$ if $c(n) \neq c(n-1)$

To minimize granular noise Δ_{min} should be small, and $\Delta_{max}/\Delta_{min}$ should be large enough to maintain a high SNR over a desired range of input signal levels. Jayant's simulation studies (Ref. 11) showed that for stability the product of P and Q should not exceed 1, and that an optimal value for P is 1.5. Since the peak around the optimal value of P is relatively broad, for ease of implementation practical systems adopt values of $P = 2$ and $Q = \frac{1}{2}$.

The ratio of the sampling frequency used by DM systems to twice the Nyquist frequency is referred to as the oversampling index. The oversampling index is equivalent to the role played by the number of bits/sample for a multi-bit quantizer employing sampling at the Nyquist rate. Practical values of the oversampling index lie in the range 2 to 32. ADM coders attain about 9 dB improvement in SNR for each doubling of the oversampling index (Ref. 10, Ref. 11). At bit rates of 32 kbit/s and below, ADM coders provide SNR and perceptual advantages over PCM.

Another example of adaptive quantization in DM systems, finding common use, is continuously variable slope delta modulation (CVSD) and was first proposed by Greefkes (Ref. 12). In this case the step-size adaptation is given by:

$$\Delta(n) = \beta\Delta(n-1) + D_2 \text{ , if } c(n) = c(n-1) = c(n-2)$$

$$= \beta\Delta(n-1) + D_1 \text{ , otherwise}$$

where

$$0<\beta<1, \text{ and } D_2>>D_1>0$$

The minimum and maximum step sizes are inherent in the recurrence relationships for $\Delta(n)$. The parameter β controls the speed of adaptation. If β is close to 1 the adaptation rate is slow, and increases as the value of β decreases.

CVSD has been used in situations requiring low sensitivity to channel errors with speech quality below those required for commercial communications (e.g. military situations). In this case the adaptation is at a syllabic rate, and the predictor coefficient (α)(see Fig.6a) is much less than one so that the effect of channel errors dies out quickly. The price paid for this insensitivity to channel errors is decreased quality (in the reconstructed speech) when there is no error (Ref. 9, Ref. 31). A major advantage of ADM systems is this flexibility to be able to provide an effective tradeoff between quality and robustness.

3.3 Differential PCM (DPCM)

Delta modulators could be called 1-bit DPCM systems. However, the term differential PCM is generally reserved for differential quantization systems in which the quantizer has more than two levels.

DPCM systems with fixed predictors can provide improvement in SNR over direct quantization. The greatest improvement occurs in going from no prediction to first order prediction with somewhat smaller additional gains resulting from increasing the predictor order up to 4 or 5, after which little additional gain results (Ref. 1, Ref. 14).

Fig.8 illustrates a DPCM system with feed-forward adaptive quantization, and Fig.9 depicts a DPCM system with feedback adaptive quantization. Normally a DPCM system with adaptive quantization is referred to as an adaptive DPCM (ADPCM) system and would have a

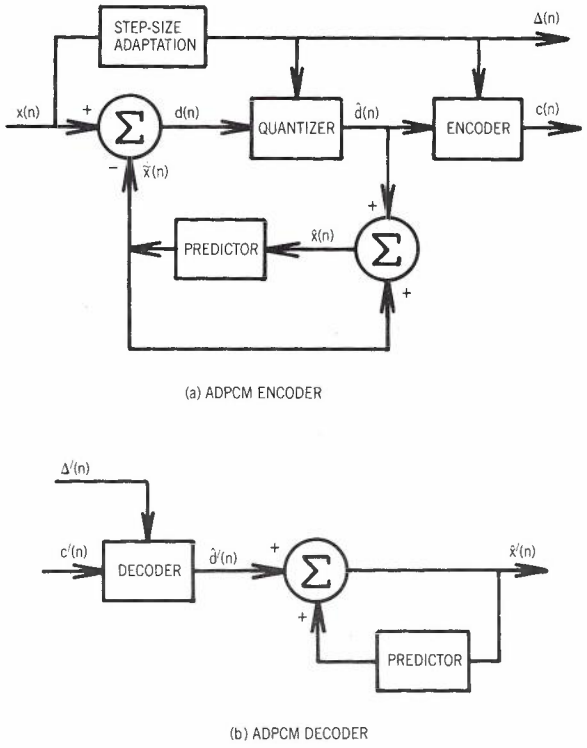


Fig.8 - ADPCM System With Feed Forward Adaptive Quantization

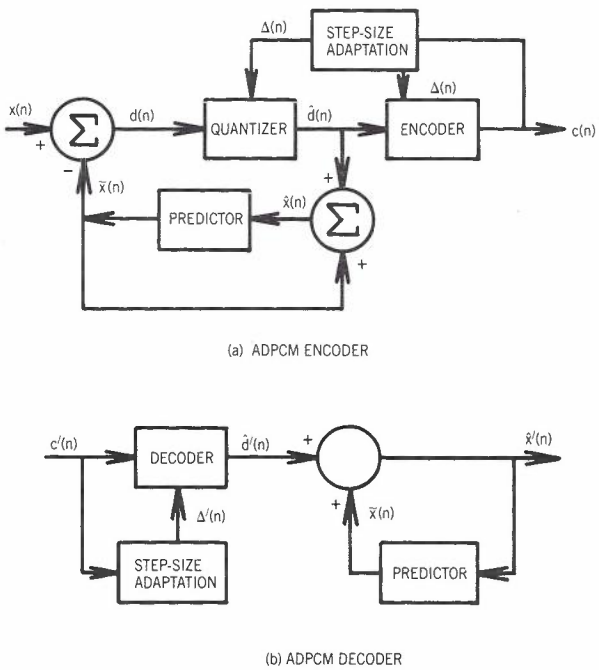


Fig.9 - ADPCM System With Feed Back Adaptive Quantization

fixed predictor. DPCM systems with adaptive predictors are more usually referred to as adaptive predictive coders but there is some inconsistency in the nomenclature appearing in the literature.

As noted previously, systems using feed-forward adaptation require the step size to be transmitted with the code word. This is

not the case with feedback adaptation since the step size may be determined from the code word sequence, however the quality of the reconstructed output is more sensitive to errors in transmission (Ref. 9). Differential encoding improves the SNR performance of the coding system. In addition adaptive quantization provides improved dynamic range as well as improved SNR.

There exist many other sophisticated waveform coders based upon similar underlying principles of waveform representation as DPCM and DM systems (Ref. 15 - 19). These include such schemes as tree encoding, aperture coding and gradient search coding. These techniques are beyond the scope of this paper except to say that the advantages of these various techniques depend upon greater memory and more extensive processing power in the encoder than the schemes already discussed.

4. ADAPTIVE PREDICTION

With DPCM systems, the expected SNR improvement over direct quantization is a function of the speaker and speech material. To cope with the nonstationarity of the speech communication process, an adaptive predictor (as well as the quantizer) to match the temporal variations of the speech signal could be employed. Although much appears in the literature on DPCM systems, relatively little is concerned with adaptation of the predictor. This appears to be because predictor adaptation is generally felt to be too complex computationally, and based on the results of McDonald (Ref. 13) performance improvement due to predictor adaptation is not felt to be substantial. A comprehensive review of adaptive prediction in speech differential encoding systems is given by Gibson (Ref. 20).

The linear predictor makes use of the correlation between consecutive samples of speech, and by regular updating of the filter coefficients the non-stationarity of the speech signal may be accounted for. Atal and Schroeder (Ref. 21) have adopted another approach which tries to remove another redundancy in the speech signal due to the correlation inherent in the quasi-periodic nature of speech. Fig. 10 outlines the coder structure.

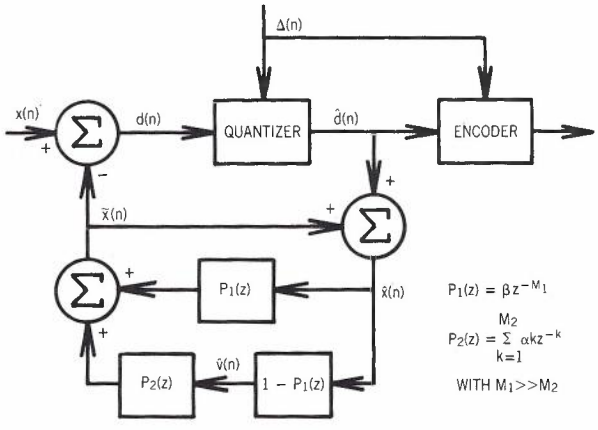


Fig.10 - Adaptive Predictive Coder

In this case the predictors P_1 and P_2 are forward adaptive and are referred to as the pitch predictor and spectral predictor, respectively. If we neglect the effects of quantization (i.e. assume $\hat{x}(n) = x(n)$), then the prediction error (difference) signal can be expressed as:

$$d(n) = x(n) - \beta x(n-M_1) - \sum_{k=1}^{M_2} \alpha_k \left[x(n-k) - \beta x(n-k-M_1) \right]$$

which can be expressed as

$$d(n) = v(n) - \sum_{k=1}^{M_2} \alpha_k v(n-k),$$

where

$$v(n) = x(n) - \beta x(n-M_1)$$

The computation of β, M_1, and $\{\alpha_k;\ k = 1,2, \ldots, M_2\}$ to minimise the variance of d(n) is not straightforward. A sub-optimal solution adopted by Atal and Schroeder is to minimise the variance of v(n), and then the variance of d(n) subject to fixed β and M_1. The predictor coefficient β was chosen to be the value of the peak of the short term normalized auto correlation function of the input signal x(n) and M_1 was selected to be the position of the peak. Thus β accounts for the variability of amplitude between consecutive periods, while M_1 is the pitch period. Given β and M_1 the sequence v(n) can then be determined and then the corresponding α_k's (for a given order (M_2) of predictor) to minimize the variance of v(n). For Atal and Schroeder's implementation $M_2 = 8$ and a 2-level quantizer was used.

Since this particular scheme derived the predictor coefficients from the input signal it is therefore a forward adaptation system. Note that this also requires the input signal to be stored to allow determination of the parameter values, prior to performing the encoding. Also it is necessary to transmit, as well as the quantized difference signal, the quantizer step-size (if feed-forward adaptive quantization is used) and the (quantized) predictor coefficients. It is possible to estimate the predictor parameters via a feedback adaptation mechanism based on the auto-correlation function of the quantized signal $\hat{x}(n)$, however this mechanism has not been widely used due to the inherent sensitivity to errors and the inferior performance that results from basing the adaptation upon a noisy input.

For voiced speech, at the instance of glottal excitation of the vocal tract, the speech signal undergoes its most rapid rate of change. Around this interval significant quantization error may be introduced (clipping errors) which manifest themselves as "pops" or "clicks" in the reconstituted speech. The pitch predictor in an adaptive predictive coder (APC) system reduces the large residual values due to "pitch pulses", and hence the clipping problem.

It has been found (Ref. 22) that for bit rates of 16 kbit/s or less a single coefficient pitch predictor was inadequate. A modified pitch predictor of the following form has been suggested (Ref. 23):

$$P_1(z) = \beta_1 z^{-(M-1)} + \beta_2 z^{-M} + \beta_3 z^{-(M+1)}$$

Alternatively a pitch adaptive (or pitch predictive) quantizer has been suggested where the number of quantizer levels is increased in the regions with large residual (prediction error) amplitudes. Information regarding the positions of these regions must be transmitted.

In order to avoid clipping entirely, Makhoul and Berouti (Ref. 24) have suggested a uniform quantizer with an indefinite number of levels, and to maintain a transmission rate of at most 16 kbit/s they use variable-length entropy coding (Ref. 38) In practice they have found a 19-level quantizer to be sufficient. With variable-length entropy coding, quantizer levels occurring most frequently are represented by short code words, whilst those occurring with low probability are represented by longer code words. In this manner the average code word length is kept as small as possible.

Adaptive predictive coders represent the extreme of complexity of digital waveform coding systems. Indeed, some schemes are on the borderline between waveform and source coding techniques since they make use of properties of the speech production process.

4.1 Noise Feedback Coding

All coders discussed so far have considered the minimization of quantization noise power to achieve optimum performance. To ensure that the distortion in the reconstructed speech signal is perceptually small, it is necessary to consider the spectrum of the quantization noise and its relation to the speech spectrum.

The theory of auditory masking suggests that noise in the frequency regions where speech energy is concentrated (the formants) would be partially or totally masked by the speech signal (Ref. 37). Thus, a large part of the perceived noise in a coder comes from those frequency regions where the signal level is low. Furthermore, what needs to be minimized is perhaps not the power of the quantization noise, but its subjective loudness.

Noise spectral shaping may be used in conjunction with any differential coding scheme. A feedback structure to perform the noise shaping is depicted in Fig.11 (Ref. 23).

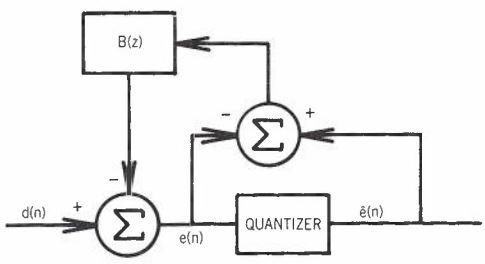


Fig.11 - Noise Feedback Coder Configuration

The purpose of the noise feedback coder is to modify the quantization noise spectrum to produce a perceptually more pleasing output. To accomplish this, the difference between the quantizer input and output (i.e. the quantization noise) is fed back via filter B(z) which determines the noise spectral shape.

5. FREQUENCY DOMAIN CODERS

The coders already discussed can be classed as "time domain" coding schemes, as the speech signal is treated as a single full-band signal. As has been observed, the main differences in the various schemes are determined by the degree and adaptation of the prediction and whether or not the quantization step size is adaptive.

Another class of coding algorithm currently undergoing investigation has been referred to as "frequency domain" coding. This class of coder divides the speech signal into a number of separate frequency bands and encodes each component separately. These techniques have the additional advantage that the number of bits used to encode each frequency component can be raised dynamically, so that encoding accuracy is placed where required in the frequency domain. Bands with little or no energy need not be encoded at all. Frequency domain coding techniques are fully discussed in (Ref. 27).

5.1 Sub-band Coding (SBC)

This coding technique has been shown to be an efficient way to exploit the short-time correlations in speech (Ref. 25, Ref. 26). In the sub-band coder the speech band is divided into four to eight sub-bands by a bank of bandpass filters. Each sub-band is low pass translated to zero frequency and then sampled at its Nyquist rate, and finally digitally encoded with an adaptive step-size PCM coder (APCM). The speech signal is reconstructed by decoding the sub-band signals, modulating each band back to its original frequencies and then summing the result. This process is depicted in Fig.12.

By encoding in sub-bands quantization noise can be contained within bands thus preventing masking of one frequency band by quantizing noise from another. Also, separate adaptive quantizer step-sizes can be used and hence bands with lower energy can have smaller step-sizes thus contributing less quantization noise.

With appropriate allocation of bits among the bands, the frequency spectrum of the quantization noise may be controlled. In lower frequency bands where pitch and formant structure must be accurately preserved, a larger number of bits/sample are used. In upper frequency bands where fricative and noise-like sounds occur in speech a lower number of bits/sample are used.

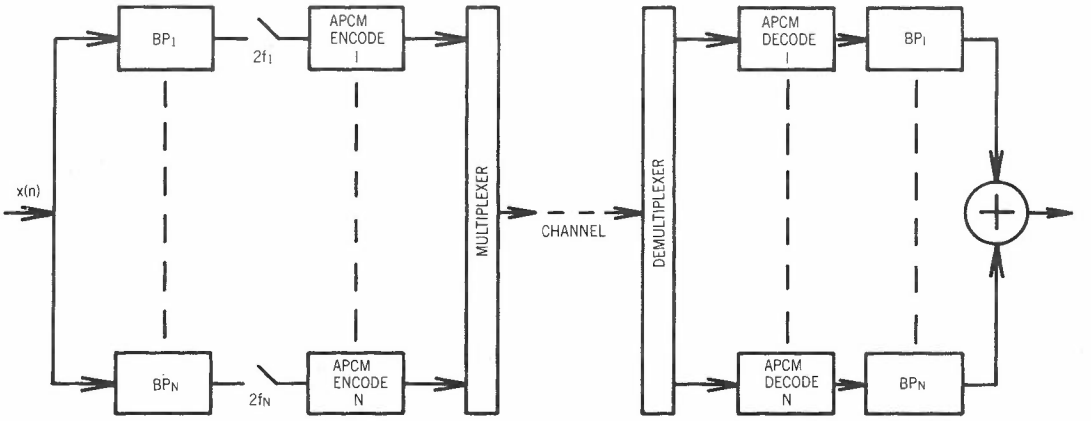


Fig.12 - Sub Band Coder

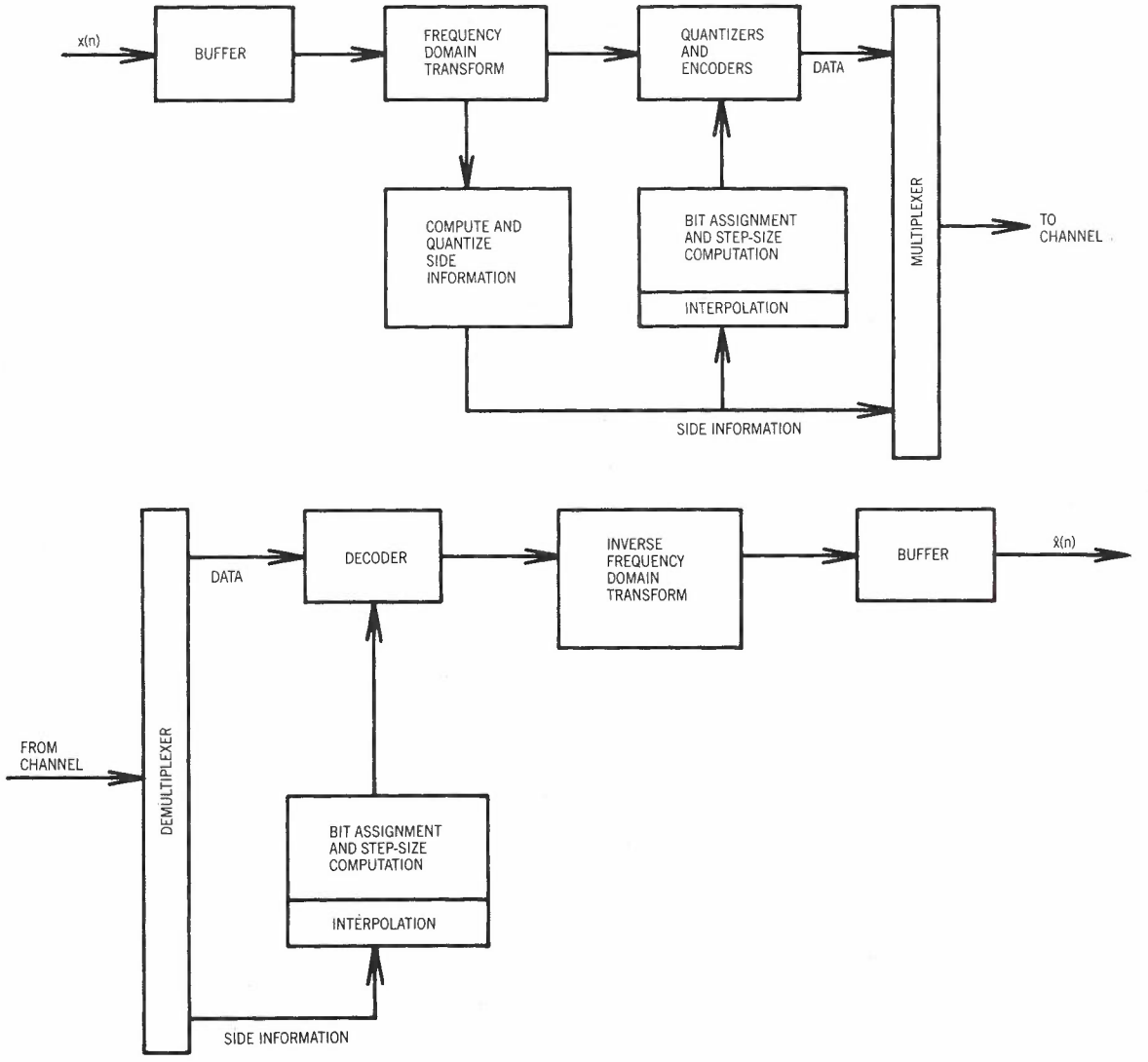


Fig.13 - Adaptive Transform Coder

TABLE 1 - An Example of Bit Assignment in Sub-Band Coding

Band No.	1	2	3	4	5
Freq. Range [Hz]	3200-1600	1600-800	800-400	400-200	200-100
Bit Allocation	2	2	4	5	5

For example, a five band sub-band coder for 16 kbit/s encoding (sampling rate 6400 Hz) used the bit assignment (Ref. 39) shown in Table 1.

5.2 Adaptive Transform Coding

Adaptive transform coding as originally proposed by Zelinski and Noll (Ref. 28) is illustrated in Fig.13.

The input speech is buffered into short time blocks (or frames) of data, and then transformed using a "frequency domain" transform such as the discrete Fourier transform (DFT), the discrete cosine transform (DCT), or a closely related symmetric discrete Fourier transform (SDFT), or the Karhunen-Loeve transform (KLT) (Ref. 29).

In practice the DCT or SDFT are usually chosen because they have fast computational algorithms, are signal independent, produce a better SNR performance than the DFT, reduce frame end effects (which can lead to undesirable "clicks" and "burbling noises"), and finally they also asymptotically approach the theoretically optimum performance of the KLT. The KLT is generally not used because of its computational complexity. A further advantage of the DCT and SDFT is that these transforms lead to a set of N real coefficients (where N is the block size) instead of N/2 complex coefficients as obtained with the DFT.

The transform coefficients are then each assigned a number of bits for transmission and then quantized using a scheme designed to minimize the average distortion. This process involves making an estimate of the expected spectral levels of the transform coefficients (referred to as the "basis spectrum") and the quantizer step-size and bit allocation are derived from this spectrum estimate. These are determined on a frame by frame basis and a smoothed down-sampled version of the basis spectrum is transmitted as "side information" along with the quantized coefficients. The basis spectrum is reconstructed at the receiver via "geometric" interpolation (i.e. linear interpolation of the logarithmic values of the transmitted spectrum estimate).

The choice of bit allocation determines the accuracy with which the individual transform coefficients are encoded, and thus controls the distribution of the quantizing noise in the frequency domain. The only constraints with regard to the bit allocation are that the number of bits for any one quantizer should not exceed some predetermined maximum value and that the sum of all the bits should be less than a predetermined number of bits per block.

Studies by Tribolet *et al.* (Ref. 26) show that frequency domain coders can match and exceed the quality of their time domain counterparts.

6. CODER QUALITY AND COMPLEXITY

The spectrum of speech coding transmission rates and associated quality is shown in Fig.14 below (Ref. 30).

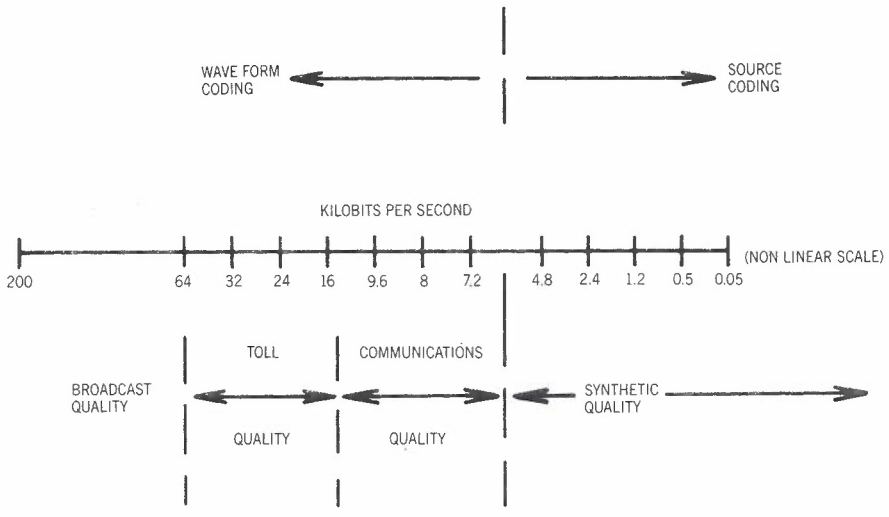


Fig.14 - Digital Speech Coding and Associated Quality

Toll quality has been loosely defined (Flanagan (Ref. 30)) to be quality comparable to that of an analog speech signal having the properties:

Frequency range - 200 to 3200 Hz
Signal-to-noise ratio ≥30 dB
Harmonic distortion - ≤2 to 3%

Currently digital coding algorithms exist capable of producing toll quality speech at coding rates down to about 16 kbit/s, and research is aimed at attaining this quality down to 9.6 kbit/s. At bit rates above 64 kbit/s, it is possible to obtain quality characteristics similar to toll quality for wider bandwidth (0 to 7 kHz) broadcast signals.

At bit rates below 16 kbit/s waveform coders can provide communications quality speech. In this case the signal is still highly intelligible but has noticeable quality reduction and some detectable distortion.

Coders in the source coding range produce speech of a synthetic nature which has lost substantial naturalness and may sound automaton-like. Speaker recognition may be difficult or impossible and the coder performance is speaker dependent.

The following tables (also from Flanagan (Ref. 30) indicate the relative complexity of various source coders and their achievable quality. The relative complexity figures are based on a relative count of logic gates required for implementation and are only approximate, being dependent upon circuit architecture. On this scale log PCM falls in the range 1 - 5.

TABLE 2 - Coder Complexity

Relative Complexity	Coder
1	ADM : adaptive delta modulator
1	ADPCM : adaptive differential PCM
5	Sub-band : sub-band coder (with CCD filters)
5	P-P ADPCM : Pitch predictive ADPCM
50	APC : Adaptive Predictive Coder
50	ATC : Adaptive Transform Coder

TABLE 3 - Coder Quality

Coder	Toll-Quality kbit/s	Communications-Quality kbit/s
Log PCM	64	36
ADM	40	24
ADPCM	32	16
Sub-band	24	9.6
APC, ATC	16	7.2

7. CONCLUSIONS

The standardization of speech coding algorithms for bit rates below 64 kbit/s is currently under discussion within CCITT study group XVIII.

The first of these will be an algorithm for 32 kbit/s speech transmission and then at some later time one for 16 kbit/s. As soon as a standard algorithm has been agreed it is expected, as happened with 64 kbit/s log PCM, that integrated circuits will be developed by many of the IC manufacturers.

The transmission aspects of the coding schemes discussed in this paper were not mentioned. More work will need to be done to examine the impact of coder technique as a function of network environment. In a purely digital network this would mean purely the susceptibility to bit errors in transmission. However a hybrid network, in which the speech signal may be converted from analog to digital to analog several times, would introduce added quality degradation at each recoding and this factor needs consideration.

Waveform coders in principle are designed to be signal independent and therefore can code a variety of signals such as speech, music, tones and voiceband data. This makes them robust for a wide range of speaker characteristics and for noisy environments. However, with more complex coders, to achieve lower bit rates and greater efficiency signal redundancies must be eliminated and therefore these coders are more signal specific. It therefore follows that those lower bit-rate (and therefore more complex) coders which are optimised for speech may not be effective for other types of signals such as tones or voiceband data.

Finally, the possibility of several standard transmission rates in the network raises the question of the effects of digital code conversion upon speech quality.

This paper has attempted to give an overview of medium bit-rate speech coding techniques. For further detailed reading the reader is referred in the first instance to Refs. 9, 20, 30 - 36.

8. REFERENCES

1. CCITT Recommendation "Pulse Code Modulation (PCM) of Voice Frequencies", Vol. III, Fascicle III.3, Rec. G711.

2. CCITT Recommendation "Performance Characteristics of PCM Channels at Audio Frequencies", Vol. III, Fascicle III.3, Rec. G712.

3. Smith, B., "Instantaneous Companding of Quantized Signals", Bell Syst. Tech. J., Vol. 36, No. 3, May 1957, pp. 653-709.

4. Max, J., "Quantizing for Minimum Distortion", IRE Trans. Info. Theory, Vol. IT-6, March 1960, pp. 7-12.

5. Paez, M.D. and Glisson, T.A., "Minimum Mean-Squared-Error Quantization in Speech", IEEE Trans. Commun. Vol. COM-20, April 1972, pp. 225-230.

6. Jayant, N.S., "Adaptive Quantization With a One Word Memory", Bell Syst. Tech. J., Vol.52, No. 7, September 1973, pp. 1119-1144.

7. Cumminskey, P. *et al.*, "Adaptive Quantization in Differential PCM Coding of Speech", Bell Syst. Tech. J., Vol.52, No. 7, September 1973, pp. 1105-1118.

8. Markel, J.D. and Gray, A.H., "Linear Prediction of Speech", Springer-Verlag, New York, 1976.

9. Rabiner, L.R. and Schafer, R.W., "Digital Processing of Speech Signals", Prentice-Hall, New Jersey, 1978.

10. Abate, J.E., "Linear and Adaptive Delta Modulation Systems", Proc. IEEE, Vol.55, March 1967, pp. 298-308.

11. Jayant, N.S., "Adaptive Delta Modulation with a One-Bit Memory", Bell Syst. Tech. J., Vol.49, No. 3, March 1970, pp. 321-342.

12. Greefkes, J.A., "A Digitally Companded Delta Modulation Modem for Speech Transmission", Proc. IEEE International Conference on Communications, June 1970, pp. 7-33 to 7-48.

13. McDonald, R.A., "Signal-to-noise and ideal channel performance of differential pulse code modulation systems - Particular applications to voice signals", Bell Syst. Tech. J., Vol.45, September 1966, pp. 1123-1151.

14. Noll, P., "A Comparative Study of Various Quantization Schemes for Speech Encoding", Bell Syst. Tech. J., Vol.54, No. 9, November 1975, pp. 1597-1614.

15. Anderson, J.B. and Bodie, J.B., "Tree Encoding of Speech", IEEE Trans. Inf. Theory, Vol. IT-21, July 1975, pp. 379-387.

16. Davis, C.R. and Hellman, M.E., "On Tree Coding with a Fidelity Criterion", *ibid*, pp. 373-378.

17. Jayant, N.S. and Christensen, S.A., "Tree Encoding of Speech using the (M,L) - algorithm and adaptive quantization", IEEE Trans. Commun. Vol. COM-26, September 1978, pp. 1376-1379.

18. Huang, J. and Schultheiss, P., "Block quantization of correlated Gaussian random variables", IEEE Trans. Commun. Syst., Vol. CS-11, September 1963, pp. 289-296.

19. Gray, R.M., "Sliding-block source coding", IEEE Trans. Inf. Theory, Vol. IT-21, July 1975, pp. 357-368.

20. Gibson, J.D., "Adaptive Prediction in Speech Differential Encoding Systems", Proc. IEEE, Vol.68, No. 4, April 1980, pp. 488-525.

21. Atal, B.S. and Schroeder, M.R., "Adaptive Predictive Coding of Speech Signals", Bell Syst. Tech. J., Vol.49, No. 8, October 1970, pp. 1973-1986.

22. Makhoul, J. and Berouti, M., "Adaptive Predictive Speech Coding", J. Acoust. Soc. Am., Vol.66, No. 6, December 1979, pp. 1633-1641.

23. Atal, B.S. and Schroeder, M.R., "Predictive Coding of Speech Signals and Subjective Error Criteria", IEEE Int. Conf. Acoust. Speech Signal Processing, Tulsa, OK, 1978 pp. 573-576.

24. Makhoul, J. and Berouti, M., "Adaptive Noise Spectral Shaping and Entropy Coding in Predictive Coding of Speech", IEEE Trans. Acoust. Speech, Signal Process. Vol. ASSP-27, 1979, pp. 63-73.

25. Crochiere, R.E., *et al.*, "Digital Coding of Speech in Sub-bands", Bell Syst. Tech. J., Vol.55, October 1976, pp. 1069-1085.

26. Tribolet, J.M. *et al.*, "A Comparison of the Performance of Four Low Bit-rate Speech Waveform Coders", Bell Syst. Tech. J., Vol.58, March 1979, pp. 699-712.

27. Tribolet, J.M. and Crochiere, R.E., "Frequency Domain Coding of Speech", IEEE Trans. Acoust. Speech Signal Processing, Vol. ASSP-27, No. 5, October 1979, pp. 512-530.

28. Zelinski, R. and Noll, P., "Adaptive Transform Coding of Speech", IEEE Trans. Acoust. Speech, Signal Processing, Vol. ASSP-25, August 1977, pp. 299-309.

29. Ahmed, N. and Rao, K.R., "Orthogonal Transforms for Digital Signal Processing", Springer-Verlag, N.Y. 1975.

30. Flanagan, J.L., *et al.*, "Speech Coding", IEEE Trans. on Commun., Vol. COM-27, No. 4, April 1979, pp. 710-737.

31. Jayant, N.S., "Digital Coding of Speech Waveforms: PCM, DPCM and DM Quantizers", Proc. IEEE, Vol.62, May 1974, pp. 611-32.

32. Gold, B., "Digital Speech Networks", Proc. IEEE, Vol.63, April 1975, pp. 561-80.

33. Flanagan, J.L., "Opportunities and Issues in Digitized Voice", Proc. EASCON, September 1978, pp. 709-12.

34. Steele, R., "Delta Modulation Systems", London: Pentech Press, 1975.

35. Bayless, J.W., *et al.*, "Voice Signals: Bit by Bit", IEEE Spectrum, Vol. 10, October 1973, pp. 28-34.

36. Jayant, N.S., Ed. "Waveform Quantization and Coding", New York, IEEE Press, 1976.

37. Fletcher, H., "Speech and Hearing in Communication", New York, D. Van Nostrand, 1953.

38. Huffman, D.A., "A Method for the construction of Minimum Redundancy Codes", Proc. IRE, Vol.40, September 1952, pp. 1098-1101.

39. Malah, D., Crochiere, R.E. and Cox, R.V., "Performance of Transform and Sub-band Coding Systems contained with Harmonic Scaling of Speech", IEEE Trans. Acoust. Speech and Signal Processing, Vol. ASSP-29, April 1981, pp. 273-283.

BIOGRAPHY

ROLAND SEIDL graduated with a B.E. (Hons.) in 1971 from the University of Adelaide. In 1975 he received a Ph.D from the University of Newcastle for a thesis dealing with a theory of structure and encoding of visual patterns with application to character recognition.

He joined the Postmaster-General's Department in Adelaide in 1968 as a cadet engineer, and commenced with the Research Department in June 1974 upon completion of postgraduate studies. He is currently employed as a senior engineer in the Voice Services Section where he is engaged in speech processing research. This includes the investigation of speech coding, speech synthesis and speech recognition techniques and their application to new and enhanced voice services.

Book Review

New Concepts In Multi-User Communication

J.K. SKWIRZYNSKI

Sijthoff and Noordhoff, Alphen van der Rijn,
The Netherlands

The third NATO Advanced Study Institute was entitled "New Concepts in Multi-User Communication". It was held at Norwich, United Kingdom from August 4 to 16, 1980. The proceedings have been published in a book of the same name by Sijthoff and Noordhoff of The Netherlands in 1981.

Multi-user communication refers to that process by which a communication system is shared between many users by virtue of the stochastic nature of their demands; the capacity of the systems is greater than needed by any individual user but not so great that all users can be accommodated simultaneously. This is an old problem, queueing being an obvious model. What makes the modern problem so interesting and difficult are the dynamics - very small lulls in the flow of communication from an individual user provide communication capability for other users. Individual communication patterns are stochastic and the task requires control of stochastic flows from many sources. Since the aim is fast, accurate communication, multi-user communication systems must detect errors in the message and perhaps provide correction; they must have means for assigning communication channels or time slots. The concentration on data networks (especially packet switching) and satellite systems is not surprising.

Mr J.K. Skwirzynski of GEC-Marconi Electronics is the editor of the proceedings. As Director of the Institute, Mr Skwirzynski assembled an international collection of well known researchers working in the fields covered by the proceedings. The aim of the Institute was "... to give a series of co-ordinated tutorial presentations covering the main contributory areas, namely the theory and mathematical background of signal and channel coding, information, statistics, network topology, traffic routing and algorithmic complexity." This was to be an endeavour to bring together diverse aspects of the theory of multi-user communication as a start to building the foundations of an integrated theory.

Most conferences attract a great spread of topics allegedly related to the conference theme; even invited papers will approach a target from all points of the compass. The Institute has done better than most other conferences in co-ordinating the presentations - authors have obviously been given the opportunity to read drafts of other papers prior to their presentation, and some authors have relied on others to cover particular material. Unlike most textbooks, there is no easy flow from one topic to the next, but this is not necessarily a disadvantage. Each paper does deal with an aspect of the theme of multi-user communication; every one of the aspects may be important in the planning, design, development and operation of communication systems (and aren't all networks really multi-user communication systems), and every one of the facets needs to be considered in developing an 'integrated theory'.

This is a book for the post-graduate student and researcher in the field of technical communications. The practising engineer who is faced with the complexities of planning new communication systems to cater for the range of traffic requirements of many and varied users, may well benefit from reading it. But be warned, it is not a book that a neophyte can read lightly. Many of the papers assume some background knowledge in at least one of the communication system types considered. This assumption is not unreasonable given that the purpose is to bring together theories that apply to the same type of system but deal with different aspects. Indeed the range of knowledge that would be required to learn from all the presentations is considerable. Prior reading of a more elementary book such as "Computer Networks" by Andrew Tannenbaum (Prentice Hall, 1981), would be a decided advantage. The sum total of the information broadcast at the Institute is probably indigestible *en masse*, and for that reason the printed form of the proceedings is invaluable. In particular, the references are extensive and provide a well-filled reservoir of knowledge for research and study. There is one minor point to note about the material: some of it no longer reflects the forefront of thinking in the rapidly changing views on protocols and the open systems model for interconnection of communication networks.

The proceedings are in six parts. These are preceded by an historical survey of computer communication and data networking by Professor F.L.N.M. Stumpers, formerly of Philips Research Laboratories. Part 1 is entitled "Extension of Information Theory to Multi-user Systems". The papers of part 1 deal with coding and noise in the so-called "broadcast channel" and layering for efficient link control (R.G. Gallagher's paper). The paper by J.K. Wolf provides a useful set of definitions for those readers who are not familiar with information theory - e.g. 'capacity region of multiple access channels', 'mutual information'. This brings out the point that the book would have been improved by a glossary of terms, which by its nature would not have satisfied everybody, but would provide some help.

Part 2 deals with "Signal and Channel Coding Aspects of Multi-User Systems". P.G. Farrell's paper discussed error detection and correction (EDC) codes in multi-user channels (in the information theoretic sense) emphasising "EDC aspects which in some way involve collaboration between the users", i.e. co-operative use of a single channel by means of encoder/decoders. Channel codes for a range of different channel types are compared with respect to their realisability and the achievable information rate. This paper is an excellent survey of the results achieved and provides some suggestions for further study.

Part 3 is called simply "Communication Networks". Some well established traffic and queueing theory is sketched. One paper by Professor Mischa Schwartz is a gem. Called "Routing and Flow Control in Data Networks", the paper provides an excellent statement of the strategic problems of routing and flow control procedures at the transport level. F.J. Symon's "Presentation, Analysis and Verification of Communication Protocols" provides a useful survey of the problems of verifying correct operation of complex communication protocols over the complete set of conditions possible in a communication network.

"Algorithms and Computational Complexity" is the title of Part 4. Three connected but different classes of paper are presented. "Communication Network Design and Control Algorithms" by H. Kobayashi is an excellent survey of algorithms for various models of centralised and distributed networks. The complexity and order of cost (in computational steps) is discussed for network design algorithms that optimise network 'cost' over such design variables as topological structure, route selection and link capacities. The other papers discuss the theory of computational complexity in relation to the control of communication networks and network reliability.

Part 5, "Multi-user Application Areas" seems to be the 'final gathering of what's left' section. It contains an interesting model of "An Engineering Discipline for Distributed Protocol Systems" by T.F. Piatkowski. In his paper he lists the "characteristics of a mature engineering discipline" as he sees such and relates these to the need for computer aids for protocol engineering.

The last set of papers are from the Comsat Communication Laboratory and deal with the means for increasing the capacity of satellites - bandwidth efficient modulators and compandors - and crytographic techniques for satellites.

Finally, I wish to criticise the production of the book and the quality of the typescript. The publishers have used, quite reasonably, the modern technique of photo-typesetting from type written pages prepared by the authors. Many different type fonts have been used. Some papers have obviously been typed using word processors - they are right justified. One paper is double spaced. These type variations between papers can be distracting. The obvious virtue is a reduction in the cost of producing the book. However, I feel that some better attempt at uniformity would enhance the book's value and emphasise the attempted integration of the subject matter, even though the information content would be unchanged ; perhaps this is only the complaint of a bibliophile. I also find amusement in noting the number of typographical errors; I counted at least 300, which, in a book of approximately 3 million characters infers an error rate of greater than 1 in 10^4, about what one could expect from transmission over the analogue public switched telephone network.

D.W. Clark
Telecom Australia
Research Laboratories

Information for Authors

ATR invites the submission of technical manuscripts on topics relating to research into telecommunications in Australia. Original work and tutorials of lasting reference value are welcome.

Manuscripts should be written clearly in English. They must be typed using double spacing with each page numbered sequentially in the top right hand corner. The title, not exceeding two lines, should be typed in capital letters at the top of page 1. Name(s) of author(s), in capitals, with affiliation(s) in lower case underneath, should be inserted on the left side of the page below the title. An abstract, not exceeding 150 words and indicating the aim, scope and conclusions of the paper, should follow below the affiliation(s).

ATR permits three orders of headings to be used in the manuscript. First-order headings should be typed in capitals and underlined. Each first-order heading should be prefixed by a number which indicates its sequence in the text, followed by a full stop. Second-order headings should be underlined and typed in lower case letters except for the first letter of each word in the heading, which should be typed as a capital. They should be prefixed by numbers separated by a full stop to indicate their hierarchical dependence on the first-order heading. Third-order headings are typed as for second-order headings, underlined and followed by a full stop. The text should continue on the same line as the heading. Numbering of third-order headings is optional, but when used, should indicate its hierarchical dependence on the second-order heading (i.e. two full stops should be used as separators).

Tables may be included in the manuscript and sequentially numbered in the order in which they are called up in the text. The table heading should appear above the table. Figures may be supplied as clear unambiguous freehand sketches. Figures should be sequentially numbered in the order in which they are called in the text using the form: Fig. 1. A separate list of figure captions is to be provided with the manuscript. Equations are to be numbered consecutively with Arabic numerals in parenthesis, placed at the right hand margin.

A list of references should be given at the end of the manuscript, typed in close spacing with a line between each reference cited. References must be sequentially numbered in the order in which they are called in the text. They should appear in the text as (Ref. 1). The format for references is shown in the following examples.

> Wilkinson, R.I., "Theories for Toll Traffic Engineering in the U.S.A.", BSTJ, Vol. 35, No. 2, March 1956, pp.421-514.
>
> Abramowitz, M. and Stegun, I.A., (Eds), Handbook of Mathematical Functions, Dover, New York, 1965.

Three copies of the paper, together with a biography and clear photograph of each of the authors should be submitted for consideration to the secretary (see inside front cover). All submissions are reviewed by referees who will recommend acceptance, modification or rejection of the material for publication. After acceptance and publication of a manuscript, authors of each paper will receive 50 free reprints of the paper and a complimentary copy of the journal.

Benefits of Authorship for ATR.

- ATR is a rigorously reviewed journal with international distribution and abstracting.
- Contact with workers in your field in the major telecommunication laboratories in Australia can improve interaction.
- Contributions relevant to the Australian context can contribute to the viability and vitality of future Australian research and industry.
- Publication is facilitated because ATR arranges drafting of figures at no cost to the author, and with no need for the author to spend time on learning journal format standards.

ATR AUSTRALIAN TELECOMMUNICATION RESEARCH
ISSN 0001-2777

VOLUME 17, NUMBER 1,
1983

Titles (Abbreviated)