If it is not necessary to carry out the polynomial transform in-place, i.e., if there are two memories (one for input and the other for output of the butterfly), then the rotating of the polynomial $A(z) - B(z)$ can be carried out by appropriate addressing while storing the butterfly output, e.g., storing $a_0 - b_0$ in the location corresponding to b_k. If the algorithm is carried out in-place, the rotation of polynomial coefficients can be achieved by having a first-in–first-out buffer in the butterfly processor. Alternatively, this buffer memory can be avoided by having a pointer to indicate the location of z^0 in every row. Then the output $(a_0 - b_0)$ is stored in place of the input b_0 and the pointer associated with row B is decremented by k mod N, e.g., if the pointer was 0 before the butterfly it is set to $N - k$. A pointer update is then needed for every N add/subtract operations.

From the above discussion it is clear that this polynomial transform has all the advantages of the conventional FFT. It does not have any of the multiplications which makes the FFT computation so complicated. This polynomial transform butterfly operation can be implemented using off-the-shelf adders and subtractors or using application specific integrated circuit (ASIC) technology. In the latter case a relatively large number of butterflies can be included in the same device. Although only the radix-2 algorithm has been described here, considerable benefits can be gained by extending this to higher radices.

The 1-D skew-cyclic convolutions resulting from the polynomial transform can be implemented using an N-stage VLSI hardware convolver as shown in Fig. 2. To illustrate this

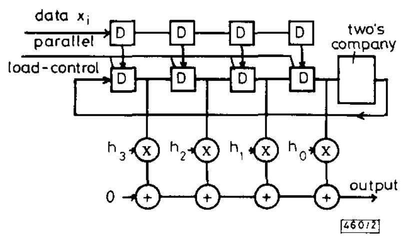

Fig. 2 *Implementing noncyclic convolution*
$N = 4$

approach, a four-tap hardware convolver implementing a polynomial multiplication modulo $(z^4 + 1)$ has been used. The data sequence is entered into the convolver in reverse order, i.e., x_3 entered in cycle one. After the four data samples have been loaded into the convolver the data is recycled through a two's complementer. The required four convolution results are obtained in cycles four to seven. The second data shift-register has been included to allow the next data sequence to be clocked-in when the results of the current convolution are being clocked-out. Hence an output sample is obtained for every input sample.

Since the polynomial transform consists of only additions and subtractions, it can easily be carried out with full precision. Most VLSI hardware convolution devices[4] are also designed to generate full precision results so the entire multidimensional convolution can be implemented without introducing any quantisation errors. The numbers can be scaled at any stage of the algorithm. The quantisation effects of the algorithm can be controlled. Unlike the FFT approach, there is no need here to use large wordlengths to reduce the quantisation effects. This low wordlength requirement in the polynomial transform architecture reduces the hardware complexity and memory size required and increases operating speed.

The architecture can be used for computing real or complex valued convolutions. In the case of real valued data the method presented, unlike the FFT approach, does not require any complex arithmetic. For complex convolutions, the polynomial transform can be carried out independently and in parallel on the real and imaginary parts of the input data array. In this case, the resulting 1-D skew-cyclic convolutions are complex-valued.

Number theoretic transforms (NTTs) can also be carried out without multiplications. The arithmetic of NTTs has to be implemented in an integer ring large enough to accommodate the largest possible output integer. Useful NTTs also have severe restrictions on the transform length imposed by the machine wordlength. The polynomial transform technique described here can be applied to compute convolutions of any size, using integer, fractional, block floating-point or full floating-point arithmetic.

To illustrate the practical significance of the method consider convolving an image of size (512 × 512) with a reference of size (256 × 256) using commercially available 20 MHz cascadable 64-stage convolvers.[4] To directly compute this convolution with 20 Mbyte/s throughput, an array of 1024 devices is needed. Using the polynomial transform architecture described here, the same approximate throughput can be achieved using a pipeline of four polynomial transform butterfly units, a linear array of eight 64-stage convolvers implementing the polynomial multiplication modulo $(z^{512} + 1)$, and four inverse polynomial transform butterfly units. Each polynomial transform/inverse transform butterfly unit consists of an adder and a subtractor.

Conclusion: An efficient architecture has been described for implementing a relatively high order 2-D convolution, using polynomial transforms and a few hardware convolution devices. This method can be extended to convolutions with more than one dimension. In all cases the polynomial transform is used to map the multidimensional convolution into a set of 1-D convolutions. Hardware efficiency increases with the dimensionality of the convolution. Since this transform consists only of additions and subtractions, it can be implemented without any complex number arithmetic, with little or no quantisation errors and using relatively low wordlengths.

B. ARAMBEPOLA
19th December 1989

Department of Electrical Engineering
Imperial College of Science, Technology and Medicine
Exhibition Road
London SW7 2BT, United Kingdom

References

1 ARAMBEPOLA, B., and RAYNER, P. J. W.: 'Efficient transforms for multidimensional convolutions', *Electron. Lett.*, 1979, **15**, pp. 189–190
2 ARAMBEPOLA, B., and RAYNER, P. J. W.: 'Discrete transforms over polynomial rings with applications in computing multidimensional convolutions', *IEEE Trans.*, 1980, **ASSP-28**, pp. 407–414
3 NUSSBAUMER, H. J., and QUANDALLE, P.: 'Computations of convolutions and discrete Fourier transforms by polynomial transforms', *IBM J. Res. Develop.*, 1978, **22**, pp. 134–144
4 RUETZ, P. A.: 'The architectures and design of a 20-MHz real-time DSP chip set', *IEEE J. Solid State Circuits*, 1989, **24**, pp. 338–348

SOME IMPROVEMENTS TO CONVENTIONAL IMPORTANCE SAMPLING TECHNIQUES FOR CODED SYSTEMS USING VITERBI DECODING

Indexing terms: Modelling, Monte Carlo methods, Sampled data systems

Importance sampling has been used to reduce simulation run-time. It is known that the effects of importance sampling can be reduced by system memory (e.g., systems employing a Viterbi decoder). Modifications to conventional importance sampling which give improvements for simulated Viterbi decoding are presented.

Introduction: In the evaluation of digital communication systems it is usually necessary to obtain an estimate of the bit error rate (BER) performance of the system. A common approach has been to use Monte Carlo simulation to estimate p_e (the BER). Monte Carlo runs can become prohibitively long for systems with low p_e values. A popular approach to reducing simulation time is to use importance sampling (IS).[1-3] IS offers the possibility of large savings on simulation run-time for systems without memories. However for systems with memory these improvements can become negligible as the memory length increases.[1-3]

We present some improvements to conventional IS techniques for use in simulating systems with memory. In particular we consider coded digital transmissions over an AWGN channel using Viterbi decoding. As an example consider the case of a 16-QAM system encoded to a 32-QAM system using a rate 1/2 (2, 1, 2) convolutional encoder.[4] The mapping of input symbols to the 32-QAM signal set is done using Ungerboeck's mapping by set partitioning.[5] The Viterbi decoder has been implemented with a truncation length of $\tau = 12$.

It is particularly difficult to use IS schemes in the presence of Viterbi decoding since the decoder decisions are based on an infinite stream of inputs. For IS schemes this infinite memory implies the need to use a weighting or debiasing procedure based on the infinite sequence of biased noise values. Hence in applying IS to Viterbi decoding it is crucial to avoid this problem of infinite memory by constructing a finite effective memory length. The effective memory length, M, can be defined as the number of input bits which are most likely to affect a single decoding decision. Using the notation of Herro and Nowack[2] the effective memory length of the decoder can be given by $M = 2n\tau + n$ for an (n, k, m) convolutional code operating on a Viterbi decoder with truncation length τ. Hence for the 32-QAM system studied $n = 2$, $\tau = 12$ and so $M = 50$. For two dimensional constellations $M/2$ is the effective memory length in terms of signal and noise pairs. Using this as the memory length Herro and Nowack[2] give an IS estimator of the form

$$\hat{p}_{IS} = \frac{1}{N} \sum_{i=1}^{N} D[g(\boldsymbol{s}_i + \boldsymbol{n}_i)] w(\boldsymbol{n}_i) \tag{1}$$

The following notation is used in eqn. 1. N denotes the length of the IS simulation. The decoder transfer function is given by $g(.)$. The function $D[g(.)]$ is the usual error indicator function taking value 1 if the input results in an error and 0 otherwise. The vectors $\boldsymbol{s}_i$ and $\boldsymbol{n}_i$ are M-dimensional vectors made up of $M/2$ two-dimensional pairs. The idea is that these vectors contain the effective memory of the decoder. The weight function $w(\boldsymbol{n}_i)$ has the product form

$$w(\boldsymbol{n}_i) = \prod_{j=1}^{M/2} w(n_{ij}^{(1)}, n_{ij}^{(2)}) \tag{2}$$

where

$$w(n_{ij}^{(1)}, n_{ij}^{(2)}) = \frac{f_n(n_{ij}^{(1)}, n_{ij}^{(2)})}{f_n^*(n_{ij}^{(1)}, n_{ij}^{(2)})} \tag{3}$$

$f_n(.,.)$ is the channel noise distribution and $f_n^*(.,.)$ is the biased noise distribution. This can be read as the weight of the jth input sample contributing to the ith output. Since the channel exhibits AWGN the channel noise distribution is simply a product of two zero mean Gaussian distributions with variance σ^2.

On the choice of biased distributions: In the conventional approach to IS described in References 1 and 2 the biased distribution considered is a zero mean Gaussian distribution with a distorted variance. Throughout the letter this biased distribution is referred to as the conventional distribution. There are two simple extensions to this which can be considered. Firstly varying the mean as well as the variance is of interest. This technique is intuitively effective since it has the desirable property of shifting the body of the noise distribution away from zero. Secondly there is no reason for the biased distribution to be Gaussian. In general few alternatives to the normal distribution have been investigated. Accordingly the following three biased distributions have been considered.

(A) Conventional distribution: The biased distribution is Gaussian in both dimensions, the same form as for the channel noise except a biased variance σ_*^2 is used where $\sigma_*^2 = r\sigma^2$.

(B) Star distribution: The biased distribution is basically a mixture of Gaussian distributions where the mean is zero in one dimension and $\pm\mu$ in the second dimension. The variance is also biased as above with $\sigma_*^2 = r\sigma^2$.

(C) Cross distribution: The distribution is basically a mixture of uniform distributions where the mean is zero in one dimension and $\pm 1/\sqrt{(2)}$ in the second dimension. The range of each uniform distribution is $2b$.

The use of the cross distribution results in a biased estimator. This property is not a serious problem since it can be shown that the bias can be made negligible by a suitable choice of the width parameter b. The three biased distributions are illustrated in Fig. 1 and further details are given in Reference 6. All three distributions are implemented in the partial weighting scheme discussed below. The relative merits of the distributions are shown in Table 2.

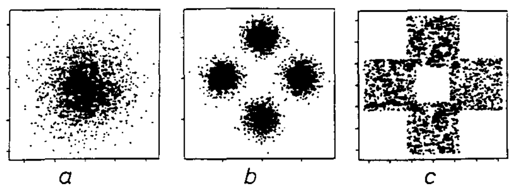

Fig. 1 *Three distributions*
a Conventional
b Star
c Cross

Partial weighting schemes: The effects of IS can be greatly reduced in the presence of system memory. Some approaches to this problem have been suggested by Davis.[3] The basic idea of that work is that a memory length of K can be reduced to a shorter effective memory length of $M < K$ were only M variables make significant contributions. Then biasing need only be applied to these M variables. The resulting IS estimator can be more effective in variance reduction since using a reduced effective memory regains some of the power of IS for memoryless systems.

For the case of Viterbi decoding Herro and Nowack[2] implemented these ideas to a certain extent by using the effective memory length of M in eqn. 1. This estimator suffered from many of the drawbacks of conventional IS as the effective memory length could still be substantial.

To avoid these problems our approach is to reduce the effective memory of the system further by biasing only a few noise terms. Hence in every $M = 2n\tau + n$ noise values only a few are biased. These are chosen to form a subsequence of the M values which is likely to cause an error in the Viterbi decoding. Hence the following burst weighting process is proposed. Every Lth signal is the beginning of a burst of $W \leq L$ biased noise pairs where $L = M + W$. Between bursts no biasing is applied. This scheme is illustrated in Fig. 2. Clearly every M input bits contains at most one burst of biased noise. The type of burst weighting used should be made to match typical noise configurations which cause errors. These can be found using *a priori* knowledge or pilot simulations.

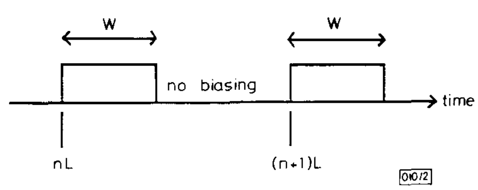

Fig. 2 *Bust weighting scheme*

With this burst weighting process the form of IS estimator is still given by eqn. 1 with a different weight function given by

$$w(\mathbf{n}_i) = \prod_{j \in A} w(n_{ij}^{(1)}, n_{ij}^{(2)}) \tag{4}$$

where A is the subset of $\{1, 2, \ldots, M/2\}$ containing integers of the form $nL, nL + 1, nL + 2, \ldots, nL + W - 1$.

The effectiveness of these burst weighting or BIS schemes can be evaluated in the form of a variance reduction ratio (VRR) relative to the conventional distribution implemented by Herro and Nowack.[2] Let the relative VRR be defined as

$$\text{relative } VRR = \frac{\sigma_{CON}^2}{\sigma_{IS}^2} \tag{5}$$

where σ_{IS}^2 is the variance of a general IS estimator and σ_{CON}^2 is the variance of the conventional IS estimator. This notation is used throughout and the square root of the relative VRR can be thought of as the multiplicative factor by which the IS method improves over the conventional method of Herro and Nowack.[2] The relative VRR can be estimated by the ratio of the sample variances produced by the simulations since no analytic results are available.

Results: The BIS approach has been simulated for the 32-QAM system discussed. The parameter W has to be chosen and the effects of the different biasing distributions is also considered. The conventional, star and cross distributions have been simulated, all implemented in burst mode. In the following discussion the notation burst CIS denotes the conventional biasing distribution operating in bursts. All simulations have length $N = 4 \times 10^5$ and each simulation is repeated 10 times. Optimal parameters were found by similation to give the smallest estimator variance and hence the greatest VRR. For each distribution values of W were varied and for each W value the distribution parameters were optimised by simulation. For the star distribution the variance distortion factor, r, was fixed at 0·9 to simplify optimisation and only the mean was varied.[6] The optimal parameters produced are given in Table 1.

Using these optimal parameters the distributions can be compared by studying their relative VRR as defined in eqn. 5. The estimated relative VRR values are given in Table 2. For the SNR values given the BER is of the order of 10^{-8}.

These results given in Table 2 show clearly that the BIS schemes give savings of up to 20 times the savings gained by conventional IS. Clearly the effect of using a partial weighting scheme is very important. The choice of biasing distribution is somewhat less critical since all three distributions give savings of the same order. There is evidence that the start and cross distributions are more effective than the conventional distribution. As usual the effects of the IS schemes increase with the SNR and the savings ratios become larger. For coded systems like the 32-QAM system these savings are very important since run times for 10 simulations of the BIS schemes are of the order of 160 min of CPU time on an IBM 4381 mainframe.

An important feature not shown by the Tables is the stability of the results over a variety of parameter values. The star and cross distributions yield VRR values which appear much less sensitive to changes than the conventional distribution. This property suggests that a BIS scheme using a star or cross distribution could be effectively used in practice.

Table 1 OPTIMAL PARAMETER VALUES FOR THE BIS SCHEME

	Biased distributions				
	Burst CIS		Burst Star IS		Burst Cross IS
	$W = 2$	$W = 3$	$W = 2$		$W = 2$
SNR	r	r	μ	r	b
21·18	4·166	3·846	0·72	0·90	0·24
20·21	3·448	2·941	0·73	0·90	0·27
19·42	3·030	2·632	0·74	0·90	0·31
18·75	2·500	2·439	0·77	0·90	0·35

Table 2 ESTIMATED RELATIVE VRR VALUES FOR THE BIS SCHEME

	Biased distributions			
	Burst CIS		Burst Star IS	Burst Cross IS
SNR	$W = 2$	$W = 3$	$W = 2$	$W = 2$
21·18	134·56	154·02	396·80	317·55
20·21	28·62	105·27	120·12	106·92
19·42	9·86	4·49	29·27	20·79
18·75	0·19	0·23	2·66	1·69

Conclusions: The search for the best IS scheme is clearly a difficult problem, especially for coded systems. Our results show that substantial improvements on conventional methods can be achieved. A study of biasing distributions has shown that the star and cross distributions are more effective in terms of variance reduction than the conventional distribution. For simulated Viterbi decoding, the most important result is that the problems caused by system memory can be greatly reduced by the use of a partial weighting scheme. In particular the weighting is applied in bursts and this reduces the effective memory of the system.

S. X. CHEN*
P. J. SMITH*
M. SHAFI†
D. VERE-JONES*

23rd March 1990

**Institute of Statistics and Operations Research*
Victoria University of Wellington, Wellington, New Zealand

†Telecom Corporation of New Zealand
Wellington, New Zealand

References

1 SHANMUGAN, K. S., and BALABAN, P.: 'A modified Monte Carlo simulation technique for the evaluation of error rate in digital communications systems', *IEEE Trans.*, 1980, **COM-28**
2 HERRO, M. A., and NOWACK, J. M.: 'Simulated Viterbi decoding using importance sampling', *IEE Proc. F*, 1988, **135**, (2), pp. 133–142
3 DAVIS, B. R.: 'An improved importance sampling method for digital communication system simulations', *IEEE Trans.*, 1986, **COM-34**, pp. 715–719
4 THAPAR, H. K.: 'Real-time applications of Trellis coding to high speed voiceband data transmission', *IEEE J. Sel. Areas Commun.*, 1984, **SAC-2**, pp. 648–658
5 UNGERBOECK, G.: 'Channel coding with multilevel/phase signals', *IEEE Trans.*, 1982, **IT-28**, pp. 55–67
6 CHEN, S.-X., SMITH, P. J., and SHAFI, M.: 'Some improvements to conventional importance sampling techniques for coded and uncoded systems'. Telecom NZ Engineering Report CS.TS 5035, October 1989

PLANAR InP/InGaAs AVALANCHE PHOTODETECTOR WITH INTEGRATED DIELECTRIC WAVELENGTH FILTER

Indexing terms: Semiconductor devices and materials, Photodetectors, Filters

Patterned bandpass dielectric interference filters with centre wavelengths around 1·3 μm have been applied to top-entry photodiodes to produce high-speed, high responsivity, photodetectors for wavelength division multiplexing. The measured responsivity exceeded 25 A/W with a full width half maximum of 19 nm and peak rejection of over 25 dB.

Introduction: Wavelength division multiplexing (WDM) has been shown to be a viable method of increasing the capacity of optical fibre communication systems. It is also of use in the