Maximising information transfer through nonlinear noisy devices

Mark D. McDonnell[a], Nigel G. Stocks[b], Charles E. M. Pearce[c] and Derek Abbott[a]

[a]Centre for Biomedical Engineering (CBME) and
Department of Electrical & Electronic Engineering,
The University of Adelaide, SA 5005, Australia

[b]School of Engineering, The University of Warwick, Coventry CV4 7AL, UK

[c]Department of Applied Mathematics
The University of Adelaide, SA 5005, Australia

ABSTRACT

Consider an array of parallel comparators (threshold devices) receiving the same input signal, but subject to independent noise, where the output from each device is summed to give an overall output. Such an array is a good model of a number of nonlinear systems including flash analogue to digital converters, sonar arrays and parallel neurons. Recently, this system was analysed by Stocks in terms of information theory, who showed that under certain conditions the transmitted information through the array is maximised for non-zero noise. This phenomenon was termed Suprathreshold Stochastic Resonance (SSR). In this paper we give further results related to the maximisation of the transmitted information in this system.

Keywords: optimal quantisation, stochastic resonance, ADC

1. INTRODUCTION

The problem we examine in this paper is to maximise the information flow through the array of N comparators (threshold devices) shown in Figure 1. All comparators receive the same input signal, x and the i–th device is subject to independent continuously valued additive noise, η_i $(i = 1, .., N)$. The output from each comparator is unity if the input signal plus the noise is greater than the threshold, θ_i, of that device and zero otherwise. The outputs from each comparator are summed to give the overall output signal, y. Hence, y is a discrete signal taking on integer values from 0 to N and can be considered as the number of devices that are currently "on".

Such arrays can model various devices such as flash analog to digital converters (ADCs)[1] (when the thresholds are uniformly distributed across the signal space), DIMUS (Digital Multibeam Steering) sonar arrays, in the "on target" position[2,3] or a summing network of N FitzHugh-Nagumo neurons.[4]

In recent work, Stocks analysed this system using Shannon information theory. For the case of all thresholds set equal to the mean, it was shown that the maximum transmitted information (also known as mutual information, i.e. the information, in bits per sample, about the input contained in the output) has a maximum for nonzero noise. This phenomenon was termed Suprathreshold Stochastic Resonance (SSR).[5–7] Conventional Stochastic Resonance (SR) occurs when a nonlinear system is optimised by a nonzero value of noise.[8] For a single threshold SR only occurs for subthreshold signals. By contrast, SSR occurs for any magnitude of signal, due to the presence of more than one threshold. More recently, the principle of SSR has been applied to cochlear implants.[9,10]

Further author information: (Send correspondence to Mark D McDonnell.)
Mark D. McDonnell.: E-mail: mmcdonne@eleceng.adelaide.edu.au, Telephone: +61 8 83036296 Fax: +61 8 8303 4360
Nigel G. Stocks.: E-mail: es2003@eng.warwick.ac.uk
Charles E. M. Pearce.: E-mail: cpearce@maths.adelaide.edu.au
Derek Abbott.: E-mail: dabbott@eleceng.adelaide.edu.au, Telephone: +61 8 83035748

Biomedical Applications of Micro- and Nanoengineering, Dan V. Nicolau, Abraham P. Lee, Editors, Proceedings of SPIE Vol. 4937 (2002) © 2002 SPIE · 0277-786X/02/$15.00

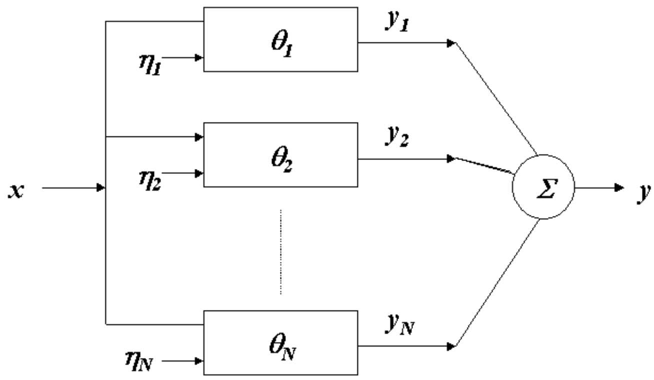

Figure 1. Array of N summing comparators. Each comparator receives the same input signal, x, and is subject to independent additive noise, η_i. The output from comparator i is zero if the sum of the signal and noise is greater than that comparator's thresholds, θ_i. The overall output, y, is the sum of the individual comparator outputs.

However, although there is a maximal nonzero noise value for the case of all thresholds equal to the signal mean, such a threshold setting does not necessarily maximise the transmitted information. For example, in the noiseless case, the transmitted information is maximised when all output states are equally probable, and is equal to $\log_2(N+1)$ bits per sample. However in the noiseless case with all thresholds equal to the signal mean, the transmitted information is only one bit per sample. In this paper we use a genetic algorithm to solve the problem of finding the thresholds that maximise the transmitted information for a given noise value.

2. CALCULATING INFORMATION TRANSMITTED THROUGH THE ARRAY

The output of device i is given by

$$y_i = \begin{cases} 1 & \text{if } x + \eta_i > \theta_i, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, the output of the array is $y = \sum_{i=1}^{N} y_i$. We consider the array to be an information channel. The transmitted information I through a channel is given by the entropy $H(y)$ of the output less the conditional entropy $H(y|x)$ of the output given the input as

$$I = H(y) - H(y|x). \tag{1}$$

As noted by Stocks,[5] $H(y|x)$ can be interpreted as the amount of *encoded* information about the input signal lost through the channel. Since the input to the array is continuously valued and the output is discretely valued, we can consider the channel to be semi–continuous.[11] The transmitted information through such a channel is given by

$$I = -\sum_{n=0}^{N} Q(n)\log_2 Q(n) - \left(-\int_{-\infty}^{\infty} P(x) \sum_{n=0}^{N} P(n|x)\log_2 P(n|x)dx \right), \tag{2}$$

where $P(x)$ is the probability density of the input signal x, $Q(n)$ is the probability of the output signal y being equal to n $(n = 0, 1, \ldots, N)$ and $P(n|x)$ the conditional probability that the output is n given the input is x.[5–7]

We have also the equation

$$Q(n) = \int_{-\infty}^{\infty} P(n|x)P(x)dx \tag{3}$$

relating $Q(n)$ and $P(n|x)$. Hence, the transmitted information can be expressed in terms of only $P(x)$ and $P(n|x)$. In turn, $P(n|x)$ is determined by $P(x)$ and the channel characteristics, that is, the number N of threshold devices, the values θ_i of the thresholds and the noise probability density $R(\eta)$.

Following the notation of Stocks,[5] let $P_{1|x,i}$ be the probability of device i being "on" (that is, signal plus noise exceeding the threshold θ_i), given the input signal x. Then

$$P_{1|x,i} = \int_{\theta_i - x}^{\infty} R(\eta) d\eta = 1 - F_R(\theta_i - x) \quad (i = 1, .., N). \tag{4}$$

where F_R is the cumulative distribution function of the noise.

2.1. Numerical calculation of I

We will obtain results for the transmitted information plotted against noise intensity, $\sigma = \sigma_r/\sigma_p$, where σ_r is the noise standard deviation and σ_p is the signal standard deviation.

To obtain I numerically, it is necessary to perform some numerical integration. The simplest form of numerical integration is to approximate the signal density function by a discrete version, with resolution $\Delta x \ll 1/N$. Hence, if in the case of the continuous density function $P(x)$ we have $a \leq x \leq b$, then discretisation with resolution Δx gives discrete values $x = a + i\Delta x, i = 0, 1, .., (b-a)/\Delta x$.

Then the above equations then become

$$I = -\sum_{n=0}^{N} Q(n) \log_2 Q(n) - \left(-\Delta x \sum_x P(x) \sum_{n=0}^{N} P(n|x) \log_2 P(n|x) \right), \tag{5}$$

where

$$Q(n) = \Delta x \sum_x P(n|x) P(x). \tag{6}$$

For an input distribution such as the uniform distribution, where $x \in [a, b]$, we only need to define the resolution. However for a distribution which has infinite bounds, such as a Gaussian, we need to restrict the upper and lower bounds of x. We will set these to be a multiple, w, of the standard deviation, σ_p, that is $x \in [-w\sigma_p, w\sigma_p]$, and then discretise to a resolution of Δx.

Given a noise density and threshold value, $P_{1|x,i}$ can be calculated numerically for any value of x from (4).[12] Assuming $P_{1|x,i}$ has been calculated for desired values of x, a convenient way of numerically calculating the probabilities $P(n|x)$ for a given number N of devices is as follows. Let $T^k_{n|x}$ denote the probability that n of the devices ($n = 1, \ldots, k$) are "on", given x. Then $T^1_{0|x} = 1 - P_{1|x,1}$ and $T^1_{1|x} = P_{1|x,1}$ and we have the recursive formulæ

$$\begin{aligned} T^{k+1}_{0|x} &= (1 - P_{1|x,k+1}) T^k_{0|x}, \\ T^{k+1}_{n|x} &= P_{1|x,k+1} T^k_{n-1|x} + (1 - P_{1|x,k+1}) T^k_{n|x} \quad (n = 1, .., k), \\ T^{k+1}_{k+1|x} &= P_{1|x,k+1} T^k_{k|x}. \end{aligned} \tag{7}$$

We have $P(n|x)$ given by $T^N_{n|x}$. An alternative evaluation is the coefficient of z^n in the power series expansion of

$$\prod_{i=1}^{N} \left[1 - P_{1|x,i} + z P_{1|x,i} \right].$$

In particular, when the thresholds all have the same value, then each $P_{1|x,i}$ has the same value $P_{1|x}$ and we have the binomial distribution

$$P(n|x) = \binom{N}{n} (P_{1|x})^n (1 - P_{1|x})^{N-n} \quad (0 \leq n \leq N).$$

Thus, for any arbitrary threshold settings and signal and noise probability distributions, $P(n|x)$ can be easily calculated from (4) and (7) and therefore the transmitted information can be calculated from (5) and (6).

3. FINDING THE OPTIMAL THRESHOLD CONFIGURATION

The problem of interest here is to find the threshold settings that optimise the transmitted information through the array. This problem is an optimisation problem and can be described in the notation of optimisation as an unconstrained nonlinear program:

$$\begin{aligned}
\text{Maximise} \quad & I = -\sum_{n=0}^{N} Q(n)\log_2 Q(n) - \left(-\int_{-\infty}^{\infty} P(x) \sum_{n=0}^{N} P(n|x)\log_2 P(n|x)dx \right), \\
\text{where} \quad & Q(n) \text{ and } P(n|x) \text{ are functions of } P_{1|x,i}, \\
\text{and} \quad & P_{1|x,i} = \int_{\theta_i - x}^{\infty} R(\eta)d\eta = 1 - F_R(\theta_i - x) \quad (i = 1,..,N), \\
\text{subject to:} \quad & \theta_i \in \mathbf{R}.
\end{aligned}$$

Note that the order of the thresholds is unimportant, that is there are no constraints of the form $\theta_j \leq \theta_k$.

3.1. The noiseless case

For the case where all comparators are noiseless $H(y|x)$ is zero, since the output of the array is completely deterministic given the input. Therefore, from (1), the transmitted information is simply the entropy $H(y)$ of the output signal. Maximizing the output entropy is achieved by ensuring all output states are equally probable, that is, $Q(n) = 1/(N+1)$ for all n.[13] In this case, from (2), the transmitted information is given by $\log_2(N+1)$ bits per sample and $\theta_i = F_P^{-1}\left(\frac{i}{N+1}\right), (i = 0,..,N)$, where F_P^{-1} is the inverse cumulative distribution function of the input signal.[12]

3.2. Nonzero noise

When there is a finite noise power added to the signal at each comparator, the problem of optimising the transmitted information becomes difficult. There are now N independent variables (each threshold, θ_i) and a highly nonlinear cost function, I. If we were to constrain each value of θ_i to be a multiple of some small value, say 0.01, and let the upper and lower bounds of a threshold be less than a certain multiple of the standard deviation of the input signal, then the solution space becomes finite, and the problem becomes a constrained integer program. In such a case it is theoretically possible to enumerate every possible combination of N thresholds in the set of possible thresholds, and record the combination that maximises the information. However such a technique is impractical, as the number of combinations to check increases combinatorially with increasing N since if there are m possible individual threshold settings, then the number of possible combinations of N thresholds is $\binom{m}{N}$.

Fortunately, unconstrained optimisation problems are easily amenable to heuristic algorithms, that is, algorithms which do not guarantee optimal solutions but should return a range of good, near optimal, solutions.

The simplest way to attempt to optimise the transmitted information is to choose all N thresholds randomly, and then calculate the transmitted information. Performing such a random selection of thresholds a large number of times and then recording the thresholds that give the highest transmitted information will hopefully provide a good solution.

However it is possible to improve on this, by assuming that a better solution can be obtained by restricting the search of possible thresholds to a local neighbourhood of the threshold settings in an already good solution. Two examples of such *random search algorithms* are simulated annealing, and genetic algorithms.

4. GENETIC ALGORITHMS

A technique that has been widely used in many sorts of combinatorial optimisation problems is that of genetic algorithms. Genetic algorithms are a method based on an analogy with genetic evolution. In general, an initial population of different feasible solutions is generated. Then at each time step of the algorithm, the cost function associated with each solution is calculated, which we will call the solutions score. Some of the lowest scoring solutions are discarded, and their place taken by newly created solutions, formed by merging some random combination of two or more of the higher scoring solutions. Some of these newly created solutions are mutated by random changes, and a new generation is thus formed. The idea is that after a large number of generations, the population of solutions should evolve towards the optimal solution. Genetic algorithms exist in many forms, and have been applied successfully to a wide range of problems despite the non-existence of a rigorous mathematical proof for their convergence.[14]

4.1. Applying a genetic algorithm to the optimal quantisation problem

We assume that we are only interested in signal and noise distributions which have zero-mean, even probability densities, so that by symmetry the maximum transmitted information must occur when half the thresholds are the negative of the other half. We will consider only the positive threshold values, which we will label as being from 1 to $\bar{N}$.. There are three main parts to applying a genetic algorithm to this problem:

- an algorithm for calculating the cost function (in this case the transmitted information), for arbitrary signal and noise probability density functions, and arbitrary threshold settings,
- a method for discarding poor solutions, and combining good solutions for the next generation, and
- a method for mutating solutions for the next generation.

We already have an algorithm for the first part. Given signal and noise densities, and threshold settings, $P_{1|x,i}$ can be calculated for a sufficiently small resolution of x and we then make use of the iterative formula (7) to calculate $P(n|x)$ for those values of x. Then the cost function (transmitted information), I, can be calculated from (5) and (6).

Given a population of S solutions, the method we have used for the second part is to take the top t percent of solutions, and discard the rest. These t solutions are then combined to form a new population of S solutions as follows. Recall that we are only interested in the positive thresholds, due to symmetry. Two randomly selected solution from the t are selected and their thresholds are sorted into ascending order. Then we randomly select j of the thresholds for the new solution to come from the first randomly selected solution, and the other $\bar{N}-j$ come from the second randomly selected solution. This is repeated S times to form a new population of S solutions.

The third part of the algorithm is to mutate the new S solutions. We have chosen to mutate all $\bar{N}$ thresholds for each of the S solutions. This is done by selecting a normally distributed random number and adding it to the threshold value. The variance of this number is set to be quite small, so that generally only small mutations occur. Intuitively, it is expected that the mean of the cost function increases with each generation. Also intuitively, decreasing the variance of the mutation of the thresholds with each generation will help the algorithm converge towards an optimum solution. We also ensure that no mutation allows a threshold to become negative.

To obtain a graph of maximum information against noise intensity, σ, the algorithm can be run for Y values of σ with G generations at each value. The order of complexity of calculating the transmitted information in this case is $O(\bar{N}/\Delta x)$.

Hence the overall algorithm has an order of complexity of $O(Y.G.\bar{N}/\Delta x)$. Increasing the accuracy of the algorithm can be achieved by increasing G, or by decreasing Δx. In either case, the runtime of the algorithm increases linearly. If $\bar{N}$ or Y is increased, the algorithm runtime also increases linearly. Hence, good solutions can be expected in a finite time, unlike an enumeration algorithm, where the runtime increases combinatorially with $\bar{N}$.

5. GENETIC ALGORITHM RESULTS

All the results presented in this section use Gaussian signal and noise distributions. It has been found that for Gaussian signal and noise, a resolution of integration (Δx) of 0.01, with the Gaussian signal bounded by $[-3\sigma_p, 3\sigma_p]$, gives sufficient accuracy.

5.1. Benchmark threshold settings

A useful benchmark is to set the thresholds to the optimal noiseless thresholds for all values of σ. The transmitted information for this case is shown by the solid line in Figure 2 (where all plots are for $N = 5$). Note that for $\sigma = 0$, $I = \log_2(N + 1)$ as expected. A second benchmark is to set all thresholds equal to the signal mean. This is shown by the dotted line in Figure 2. Note that for $\sigma = 0$, $I = 1$ as expected. For the thresholds set to the optimal noiseless values, the transmitted information decreases monotonically with increasing σ, and when all thresholds are set to zero, the SSR effect occurs, with a convex curve, and a maximum I for non zero noise.

As a means of verification of our genetic algorithm, we first applied an algorithm where all N thresholds were chosen completely randomly 10000 times, for values of σ between 0.001 and 1.4. The results for $N = 5$ can be seen in Figure 2 (indicated by an "o"). The highest value of I obtained for each value of sigma was very close to the I for the optimal noiseless thresholds, until a point near $\sigma = 0.55$. At this point the highest value of I obtained for the random thresholds was very close to that of I for all thresholds equal to zero. It is difficult to see from this plot, but I for all thresholds equal to zero is never exceeded by I for random thresholds once σ increases past the cross-over point near $\sigma = 0.55$.

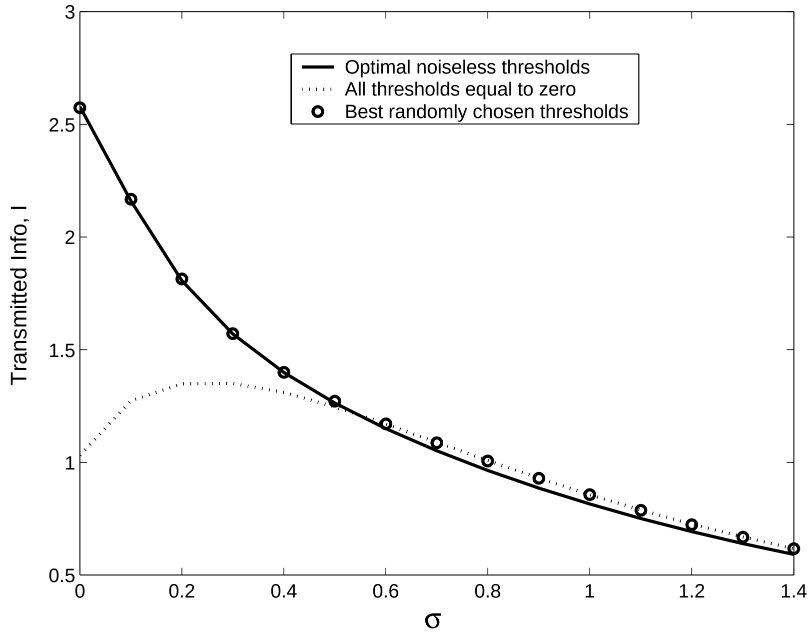

Figure 2. Noise intensity, σ, against transmitted information, I, for $N = 5$. The solid line is for thresholds distributed optimally for zero noise, the dotted line is for all thresholds equal to zero, and the circles show the maximum transmitted information found from 10000 random threshold selections.

Figure 3 shows the values of the thresholds that gave the maximum value of I, for each value of σ. Note the general trend of two positive thresholds and two negative thresholds, with a central threshold close to zero, until σ increases past about 0.7, in which case all thresholds become close to zero.

A graph of the value of N against the value of σ where the cross over point occurs (that is, where the curve for I with the thresholds set to the optimal noiseless values decreases below the curve for I obtained with all thresholds equal to zero) is shown in Figure 4. It can be seen that the cross over value of σ increases with increasing N.

Note from Figure 4 that for $N = 5$, the cross over point is between $\sigma = 0.53$ and $\sigma = 0.54$. This is verified in Figure 2.

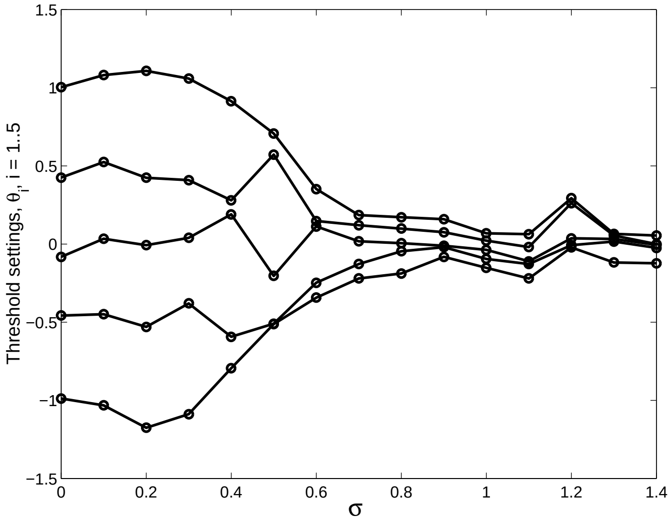

Figure 3. Noise intensity, σ, against the values of the randomly chosen threshold settings that gave the maximum transmitted information from 10000 random selections, for $N = 5$.

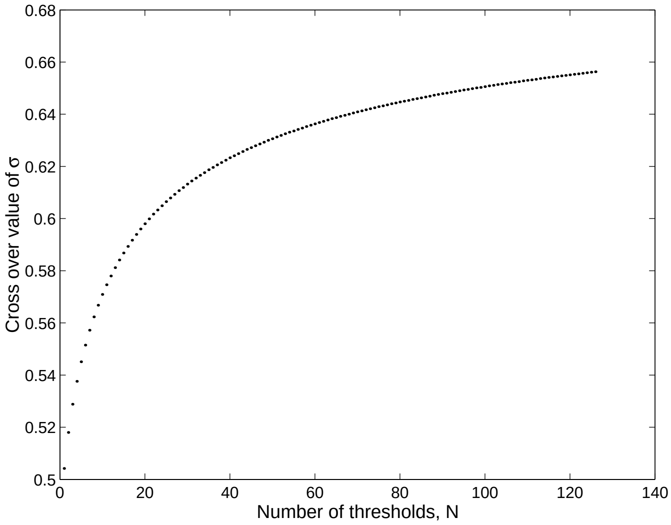

Figure 4. Number of thresholds, N, against the cross-over value of σ, that is, the value of σ where the curve for the transmitted information obtained with thresholds distributed optimally for zero noise decreases below the curve for transmitted information obtained with all thresholds set equal to zero. As N increases, the cross-over value of σ also increases.

5.2. Genetic algorithm results

The results of the random selection of thresholds verify our assumption that only positive valued thresholds need to be selected. One threshold can be set to zero, and half the rest to be the negative of the other half. This means that mutations should only be allowed on the positive thresholds, and we cannot allow a threshold to become negative. Also, since there is a smooth curve in the thresholds with increasing σ, a good initial population for the next value of σ should be the best threshold settings found for the previous value of σ. Also, once it is established that σ is large enough such that the optimum thresholds have moved into the regime where all positive thresholds are equal, the algorithm should then decide that this is the case, and only one positive threshold needs be selected. Furthermore, once it is established that σ is large enough such that all thresholds at zero is the optimal setting, the algorithm can stop.

The genetic algorithm results presented here were obtained for $\Delta x = 0.01$, $G = 10$ generations, a population size of $S = 1000$, where the top $t = 1.0$ percent (i.e. 10) of solutions in each generation are kept. The variance of the mutations was set to be 0.1, divided by the generation number, so that the average mutation size became smaller with each generation.

5.2.1. N=5

For each value of σ, the genetic algorithm found threshold settings that increased the maximum value of I over value of I from the benchmark where the thresholds were set to optimise the noiseless information. These threshold settings are shown in Figure 5. Note that the curve is much smoother than that for the randomly chosen thresholds in Figure 3. However, although there was an improvement it was fairly negligible; in the order of 10^{-4} bits per sample. The improvement is shown in Figure 6.

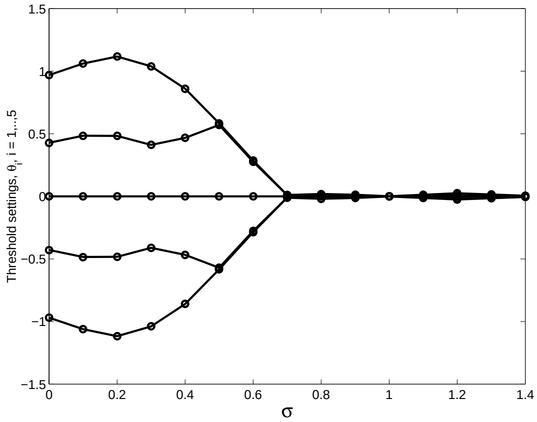

Figure 5. Noise intensity, σ, against the values of the optimal threshold settings found by the genetic algorithm for $N = 5$.

We were able to verify these results for the case of $N = 5$. Since N is small, and we only need to find two positive thresholds, it is feasible to find the optimal solution by quantising the possible threshold values and iterating all possible combinations of these to find the maximum value of I for each value of sigma. If we allow m possible threshold values, then such an algorithm requires $\binom{m}{2}$ calculations of the transmitted information. Performing this algorithm for the case $N = 5$ showed that the solutions obtained by the genetic algorithm given in Figure 5 are extremely close to the optimal solutions.

5.2.2. $N = 15$

The optimal thresholds found by the genetic algorithm are shown in Figure 7. Only the positive thresholds are shown, as the rest of the thresholds is a mirror image. Note how above a certain value of σ, some thresholds tend to converge to certain values. At $\sigma = 0.4$, approximately half the positive thresholds converge to zero and approximately half to unity. At $\sigma = 0.5$, all the positive thresholds converge to a value near 0.6. In the latter case, this means the array is acting as if it had only three thresholds with values $+0.6, 0$ and -0.6.

The region where the optimal setting is all thresholds equal to zero starts at a value between $\sigma = 0.6$ and 0.7. For σ smaller than this value, the maximum amount of information gained by optimising the thresholds instead of using the thresholds that maximise the noiseless information, is about 0.03 bits per sample. The maximum information gain in the region where the optimal setting is all thresholds zero is about 0.07 bits per sample.

5.2.3. $N = 31$

The results of the genetic algorithm for $N = 31$ showed similar behaviour to the case of $N = 15$. A larger increase in I can be obtained, but this value is still very small; about 0.05 bits per sample in the first region, and about 0.08 bits per sample in the second region.

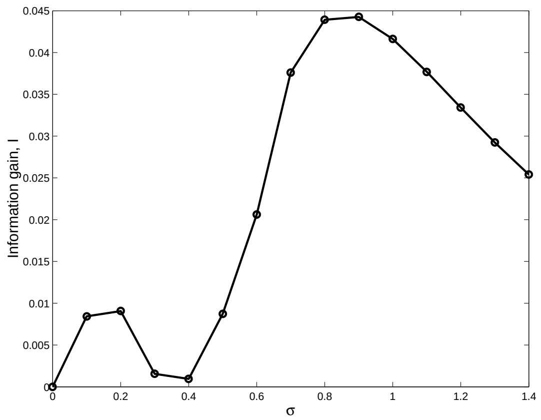

Figure 6. Noise intensity, σ, against the increase in transmitted information, I, gained by using the optimal threshold settings for each value of σ found by the genetic algorithm, instead of the threshold settings that optimise the noiseless transmitted information, for $N = 5$.

6. CONCLUSIONS

It is evident from the results given in this paper that there are two or three distinct regions in the curve of maximum information against noise intensity. In the first region, for low values of σ, only very small improvements in I can be gained by adjusting the thresholds from those that maximise the noiseless information. The second main region is that where for large enough values of σ, all thresholds set to zero maximises I, although this does not give a substantially large increase over the thresholds that maximise the noiseless information. In a small intermediate region, either threshold setting will suffice.

We have also shown that a genetic algorithm approach to solving the optimal quantisation problem should be applicable under general conditions for this problem.

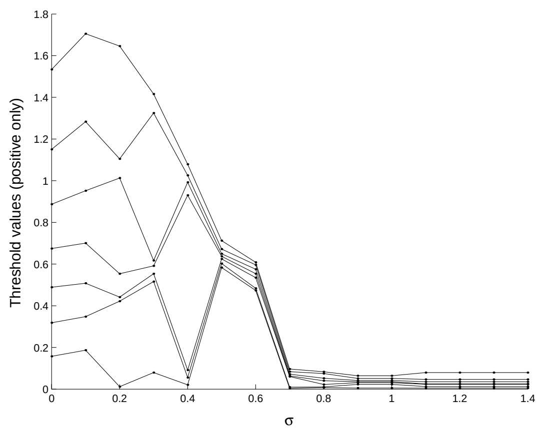

Figure 7. Noise intensity, σ, against the values of the positive optimal threshold settings found by the genetic algorithm for $N = 15$.

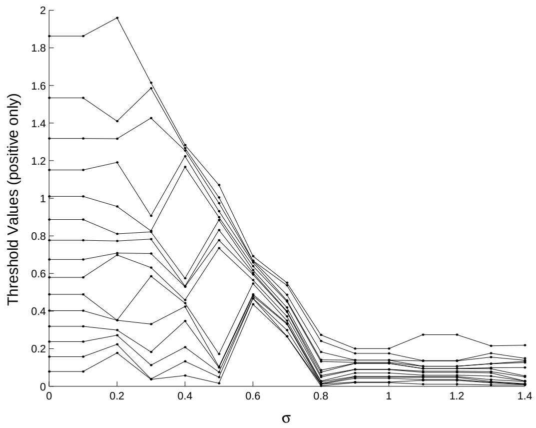

Figure 8. Noise intensity, σ, against the values of the positive optimal threshold settings found by the genetic algorithm for $N = 31$.

REFERENCES

1. V.K. Madisetti and D.B. Williams, *The Digital Signal Processing Handbook*, CRC Press, 1999.
2. V.C. Anderson, "Digital array phasing," *The Journal of the Acoustical Society of America* **32**(7), pp. 867–870, 1960.
3. P. Rudnick, "Small signal detection in the DIMUS array," *The Journal of the Acoustical Society of America* **32**(7), pp. 871–877, 1960.
4. N.G. Stocks and R. Manella, "Generic noise-enhanced coding in neuronal arrays," *Physical Review E* **64**, p. 030902(R), 2001.
5. N.G. Stocks, "Suprathreshold stochastic resonance in mutlilevel threshold systems," *Physical Review Letters* **84**(11), pp. 2310–2313, 2000.
6. N.G. Stocks, "Suprathreshold stochastic resonance: an exact result for uniformly distributed signal and noise," *Physics Letters A* **279**(5-6), pp. 308–312, 2001.
7. N.G. Stocks, "Information transmission in parallel threshold arrays: Suprathreshold stochastic resonance," *Physical Review E* **63**(4), 2001.
8. L. Gammaitoni, P. Hänggi and P. Jung, "Stochastic resonance," *Reviews of Modern Physics* **70**(1), pp. 223–287, 1998.
9. N.G. Stocks, D. Allingham, R.P. Morse and A.P. Nikitin, "The possible role of suprathreshold stochastic resonance in neural coding with applications to cochlear implants," in *Third International Conference on Unsolved Problems of Noise and Fluctuations, NIH, Washington DC*, In Press, 2002.
10. R.P. Morse, N.G. Stocks and D. Allingham, "Enhanced cochlear implant coding using noise," in *Third International Conference on Unsolved Problems of Noise and Fluctuations, NIH, Washington DC*, In Press, 2002.
11. A. Feinstein, *Foundations of Information Theory*, McGraw-Hill, 1958.
12. M. McDonnell, C. Pearce, and D. Abbott, "Neural information transfer in a noisy environment," in *SPIE's International Symposium on Microelectronics and Micro-Elecro-Mechanical Systems, 17-19 December 2001, Adelaide, Australia*, *Proceedings of SPIE* **4591**, pp. 59–69, 2001.
13. T.M. Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley and Sons, 1991.
14. D. Beasley, D.R. Bull and R.R. Martin, "An overview of genetic algorithms: Part I, fundamentals," *University Computing* **15**, pp. 58–69, 1993.