Digital Filter Design at Gate-level using Evolutionary Algorithms

Julian F. Miller
School of Computing,
Napier University,
219 Colinton Road,
Edinburgh, EH14 1DJ, UK

Abstract

Traditionally digital filters are designed using the concept of a linear difference equation with the output response being a weighted sum of signal samples with usually floating point coefficients. Unfortunately such a model is necessarily expensive in terms of hardware as it requires many large bit additions and multiplications. In this paper it is shown how it is possible to evolve a tiny feed-forward rectangular array of logic gates to perform various filtering tasks – lowpass, bandpass, and multiband. The circuit is evolved by assessing its response to digitised pure sine waves. Some of the evolved circuits possess almost linear properties, which means that they are capable of filtering composite signals which have not been encountered in training.

1 INTRODUCTION

The difference equation is a fundamental concept employed in the construction and analysis of digital filters (Ifeachor et al 1993).If the output of the filter at time *n*, *y(n)* is a function of *N* samples of the signal *x(n-i)* at earlier times it is referred to as of FIR type (finite impulse response), however if the output also involves earlier outputs *y(n-i)* then the filter is said to be of type IIR (infinite impulse response)). Formally this is represented in the following way.

$$y(n)=\sum_{i=0}^{N-1} a_i x(n-i)+\sum_{i=1}^{M} b_i y(n-i) \tag{1}$$

where the coefficients a_i and b_i are real valued floating-point numbers. The essential problem of filter design is the choice of $\{a_i\}$, $\{b_i\}$, N, and M, so that the filter has the desired behaviour (i.e. frequency response). In practice the coefficients $\{a_i\}$, $\{b_i\}$ are of finite precision. The practical requirements of implementing such a system in hardware consists of providing a number of shift registers, multipliers, and adders. This is shown in Fig. 1. Large bit multipliers are very costly in hardware terms. Three of the most important factors in the design of digital filters are quality of signal response, size (cost) of hardware implementation, and speed of operation.

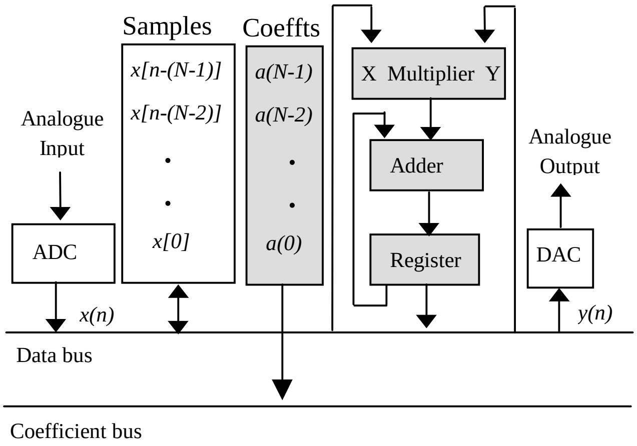


Figure 1: Conventional hardware schematic for digital filtering system. The evolved gate array carries out the function shown in grey (see section 3). ADC is the analogue to digital converter and DAC is the digital to analogue converter.

There are many traditional approaches which have been developed to address these issues (Ifeachor et al 1993). In particular one popular method for reducing the implementational complexity is to restrict the filter coefficients to integer coefficients (Dempster et al 1995 and references therein). Recently, researchers have started to explore the application of evolutionary algorithms to filter design (Arslan et al 1995, Chellapilla et al 1997, Delibasis et al 1996, Esparcia et al 1996, Harris et al 1995, Redmill et al 1997, Sriranganathan et al 1995, Wade et al 1994). The essential idea employed by most of these authors is to use an evolutionary algorithm to optimise the filter coefficients. This may be in combination with finite wordlength analysis (Arslan et al 1995, Harris et al 1995) for IIR filter design, or it may be in an adaptive context (Esparcia et al 1996, Sundaralingam et al 1997). Other workers have employed evolutionary algorithms to optimise coefficients together with add and shift operations in so-called multiplier-less designs (Redmill et al 1997, Sriranganathan et al 1995, Wade et al 1994). In (Delibasis et al 1996) a genetic algorithm was used to design an efficient non-linear filter, known as a stack filter, for signal noise reduction by finding a suitable positive boolean function (PBF). The PBF could be represented as a boolean sum of products, involving AND gates and OR gates. In (Beatriz et al 1998) the authors evolved stack filters using both GA and GP techniques. Evolutionary techniques have also been developed for signal processing in the analogue domain (Grimbleby 1995, Lohn et al 1998, Murakawa et al 1998, Zebulum et al 1998).

The work presented in this paper is an extension of recent work which looked at evolving low pass filters (Miller 1999). The objective of this work is to further extend that work and explore at a logic gate level whether it is possible to evolve networks of logic gates to carry out conventional filtering tasks such as low pass, band pass, and multi-band filters. This is an interesting thing to do for two main reasons. Firstly to explore the concept of digital filtering in a space of possibilities which is considerably larger and richer than the traditional human, top-down, difference equation method. Secondly to see how effective a microscopic number of logic gates might be in a filtering task. The pioneering concept of gate-level evolution of digital functions was developed in (Iba et al 1996). Murakawa et al (1996) generalised the concept of gate-level evolution to the so-called functional level, and they showed how it was possible to carry out adaptive equalisation on a communications channel with superior bit error rates to the conventional least mean squares method. These authors believed that it would not be possible to achieve real-world performance using a gate-level approach.

One of the objectives of the work presented here is to show that the possibilities afforded by gate-level evolution have been left largely unexplored, and that there remains much fundamental work to be done at this level. An additional motivation for attempting this work is the enormous potential for new knowledge discovery afforded by the simple nature of logic functions. In other words, can new principles be extracted from gate-level evolution which can inspire and contribute to new methodological paradigms? There are of course enormous questions which need to be addressed if such a filtering method is to become practicable. Foremost among these would be the question of linearity. If a gate array is to be trained to carry out a filtering task then can this be done in such a way that composite signals, which can be represented as weighted sums of sine waves, will also be filtered? This would imply that the circuit at least be weakly linear. The findings presented in this paper are encouraging in this regard, as in section 4 it is shown that the some of evolved gate arrays do appear to be quasi-linear (especially in the lowpass scenario).

The actual method employed here to evolve a gate array is developed from earlier work in (Miller et al 1997, 1998a, 1998b, 1998c) and has some similar features to a method called Parallel Distributed Genetic Programming (PDGP) (Poli 1997). This method is explained in section 2. In earlier work (Miller et al 1997, Miller et al 1998b, Miller et al 1998c) the objective was to synthesise an entire truth table. This becomes increasingly time consuming and difficult as the number of inputs grow. It is obvious that attempting to evolve truth tables of larger sizes will not be feasible. It was argued in (Miller et al 1998a) that the real applications for gate-array evolution probably lie in real number mapping problems, where digitised real numbers are presented to a circuit and a digitised real number output is desired. In such a scenario the number of input conditions is determined by the problem and is not necessarily an exponential function of the number of inputs. Such a scenario is ideally furnished by the digital filtering task. In this paper three filtering tasks are considered: lowpass, bandpass, and multiband. The details of this are explained in section 3. In section 4 the evolved filtering characteristics of the gate array are presented, including some results which show the quasi-linear behaviour. These are discussed in section 5, and conclusions are given in section 6.

2 GATE-LEVEL EVOLUTION OF DIGITAL CIRCUITS

The chromosome representation used is best explained with a simple example. Fig. 2 shows the genotype and phenotype for a small gate array consisting of four logic cells. The logic cells in this case have functions XOR, AND, or MUX (multiplexer). The gate array implements the one-bit adder (with carry-in).

Genotype 0 1 0 **10** 0 0 2 **6** 3 2 1 **10** 0 2 3 **16** 6 5

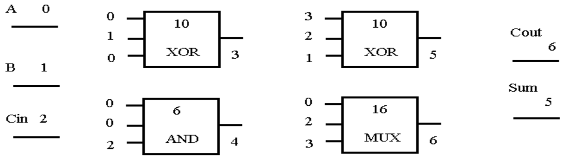


Figure 2: Genotype and phenotype for the gate array of logic cells which implement a one-bit adder

The circuit in question actually arose in an earlier experiment reported elsewhere (Miller et al 1997) and is quite novel in its own right. A, B, and Cin represent the primary inputs. Cout and Sum are the output bits of the adder. Each cell is assumed to possess three input connections. If the cell function does not require inputs then the corresponding genes are ignored. For example the upper right cell (output 5) below has input connections 3, 2, 1. This means that the first input is connected to the output of the cell with output label 3 (upper left), the second input is connected to the primary input Cin, and the third input is connected to primary input B. The function of each cell is expressed as the fourth gene associated with each cell (shown in bold typeface). The primary outputs of the gate array are also expressed as connections. For example Cout is connected to the output of the cell with output label 6. The gate array is envisaged as being divided into vertical columns of cells and the representation is so constrained that columns of cells may only have their inputs connected to connection points on their left. This ensures the atemporal, feed-forward nature of the circuit. Actually the connectivity is further constrained by the presence of a parameter called *levels-back*, and denoted by l, which dictates the number of columns (including the primary inputs at column zero) to which the inputs of cells in column l may be connected. The allowed cell functions can be chosen to be any subset of those shown in Table 1, where ab implies a AND b, $\bar{a}$ indicates NOT a, ⊕ represents the exclusive-OR operation and + the OR operation. Functions 0-15 are the basic binary functions of 0, 1 and two inputs. Functions 16-19 are all binary multiplexers with various inputs inverted. The multiplexer (MUX) implements a simple IF-THEN statement (i.e. IF c=0 THEN a ELSE b). It is important to note that one can consider multiplexers to be *atomic* both formally and from an implementational point of view. It is atomic in that it is a universal logic module so that it can be used to represent any logic function. Also some modern FPGAs now use a multiplexer based architecture so that all two input gates are synthesised with multiplexers.

Table 1: Allowed cell functions

0	1	2	3	4	5	6	7	8	9
0	1	a	b	$\bar{a}$	$\bar{b}$	ab	$a\bar{b}$	$\bar{a}b$	$\bar{a}\,\bar{b}$

10	11	12	13	14
$a \oplus b$	$a \oplus \bar{b}$	$a + b$	$a + \bar{b}$	$\bar{a} + b$

15	16	17	18	19
$\bar{a} + \bar{b}$	$a\bar{c} + bc$	$a\bar{c} + \bar{b}c$	$\bar{a}\bar{c} + bc$	$\bar{a}\bar{c} + \bar{b}c$

The genetic algorithm employed random mutation, which was defined as a percentage of genes in the population which were mutated. It respected the feed-forward nature of the circuits and also the different alphabets associated with connections and functions. When crossover was used it was of uniform type and employed a 50% genetic exchange. Elitism was always used as it is markedly beneficial (Miller et al 1998c). A probabilistic tournament selection method (size 2) was used in which the winner of the tournament was selected with a certain probability (between 0.5 and 1.0). In some cases a rudimentary $(1+\lambda)$ evolutionary strategy (ES) (Bäck et al 1991) was used to evolve the filter (with uniform mutation). In this case a population of random chromosomes is generated and the fittest chromosome selected. The new population is then filled with mutated versions of this. Rigorous experiments were not conducted to assess the relative effectiveness of the basic search algorithms chosen. The practical advantages of either the GA, or ES for filter evolution remain a topic for future research.

3 EVOLVING A FILTER RESPONSE WITH A GATE ARRAY

In digital signal processing an incoming analogue signal is sampled and the signal magnitude is represented as a binary number. Numerical manipulations of the digitised samples are carried out before the information is presented to a digital to analogue converter to produce an analogue output signal. This is the essential idea of digital signal processing. A fundamental theorem of DSP is called Nyquist's theorem, it states that one cannot reconstruct an incoming signal perfectly unless half the sampling frequency (this is defined as the Nyquist frequency f_n) is greater than the highest frequency component in the incoming signal. In the context of this paper the incoming analogue signals which are to be processed by the gate array are sampled at frequency f, with sampling period p. Thus the number of samples used, s, is given by $s=fp$. The samples are digitised and represented by a wordlength of w bits. In a filter of order n, one therefore must collect nw bits at each sampling time. These nw bits for the s samples are collected and represent the input conditions to the gate array. For each nw input bits the gate array must produce w output bits. In this way a set of input-output conditions are defined. When s samples have been collected the discrete fast fourier transform (DFFT) is taken. A program which was freely available in (Ifeachor et al 1993) was used to do this. In this way the frequency characteristics of the evolving gate array can be assessed for each input signal. The input signals chosen were pure sine waves with zero phase. They had frequencies which were integral multiples of the fundamental f_1 $(1/p)$ up to the Nyquist frequency minus one. It is important to note that a discrete Fourier transform differs from the familiar Fourier transform principally in that when it is applied to a digitised and sampled input signal the signal is resolved into a *finite* number of frequencies of the fundamental up to the Nyquist frequency. The sine waves were translated by the addition of a d.c. component so that they assumed only positive values, this removed the need for two's complement number representation. One can envisage this more clearly by noting that the fundamental corresponds to a single exact sine cycle fitting into the sampling window. The entire arrangement is shown in Figure 3. In this figure an input sine wave is shown on the left which is digitised to binary numbers with $w=4$, and $n=2$. An entire history of samples are collected for each sine wave. These are the input conditions presented to the gate array.

On the right of the gate array is shown the outputs of wordlength equal to 4 bits. To evaluate the fitness of a chromosome *each* digitised sine wave with frequency f is presented to the gate array and the DFFT of the output response is calculated. The power in the frequency domain $W(f)$, defined as the modulus of the output response in the complex frequency domain, is normalised by dividing by the maximum power associated with the DFFT of a pure sine wave.

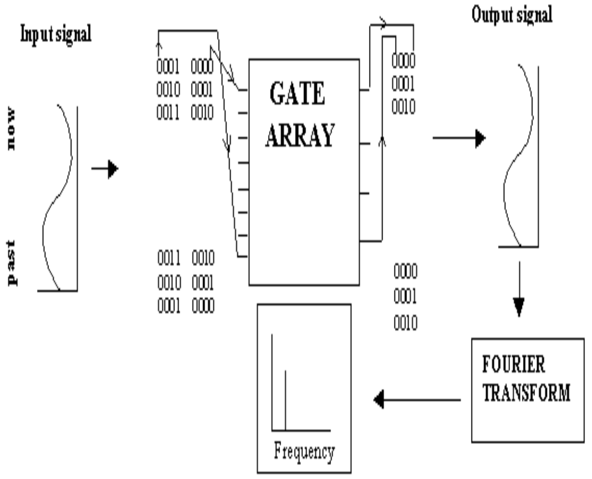


Figure 3: The training scenario for evolving a gate array with filtering properties

The d.c. component of the output is ignored. Two methods of fitness assignment were used. Define the maximum power over all frequencies , $W_{\max}$, and the maximum power over all frequencies excluding f_i , $W^i_{\max}$,

$$W_{\max} = \max\{W(f_j), \forall j : f_1 \le f_j \le f_n - 1\} \qquad (2)$$

$$W^i_{\max} = \max\{W(f_j), \forall j : j \ne i,\ f_1 \le f_j \le f_n - 1\} \qquad (3)$$

Define δ_i to be 1 if the frequency i is to be passed, and 0 if it is to be stopped. The elementary fitness contribution for the DFFT of the output signal, x_i , is given by,

$$x_i = \delta_i (W(f_i) - W^i_{\max}) + (1-\delta_i)(1-W_{\max}) \qquad (4)$$

In the first definition of fitness (passband experiments only) the fitness contributions were defined in (4). In the second method of fitness assignment (a later refinement), the fitness is calculated with a user defined set of frequency rewards r_i (fitness profile). The fitness is given by (5) below, where n_+ , and n_- , represent the number of frequencies to be passed, and stopped, respectively (over all frequencies up to f_n*–1).*

$$x_i = \delta_i (W(f_i) - W^i_{\max}) \frac{r_i}{n_+} + (1-\delta_i)(1-W_{\max}) \frac{r_i}{n_-} \qquad (5)$$

Note that the definition given in (5) is scale invariant and just assigns a fixed maximum fitness contribution for the pass regions and stop regions irrespective of their size. This is an important feature as if one were going to carry out filtering over a greater range of frequencies by using a higher sampling frequency the size of the pass region relative to the stop region could change drastically. Defining frequency rewards allows one to specify the *relative* importance of specific frequency behaviour. The total fitness x associated with a given chromosome is then given by the sum of the components x_i for all frequencies up to f_n*-1*. These definitions of fitness mean that one is trying to suppress all sine waves with frequencies in stop region, and trying to enhance only *pure* frequencies (uncorrupted sine waves) in the pass region.

4 RESULTS

The experimental parameters for this paper are given below, the nominal sampling period p was chosen to be 1 for convenience. Thus the sampling frequency f equals the number of samples s. All filters were evolved with the following parameters s=128, w =8, n = 4. On this occasion the only allowed function for all experiments was a multiplexer (type 16). The number of genes in the chromosome is equal to four times the number of gates used plus the number of output connections (see Fig. 2)

LOWPASS FILTER

The experimental parameters were: normalised passband cutoff = 0.08 (10.24 un-normalised), population size = 10, breeding rate = 100%, gene mutation probability =0.02, crossover rate = 50%, uniform crossover, number of generations = 10,000, elitism, size 2 tournament selection, acceptance probability is 0.7, geometry of gate array is 9 rows and 9 columns, levels-back l = 9. The results shown below are for the best of ten runs of the genetic algorithm under the above conditions. The frequency response of the evolved lowpass filter is shown in Fig. 4. The y-axis shows only the *maximum* power amplitudes of the filter responses to the digitised pure sinusoidal input signals. Other frequency components for particular incident signals can be seen in the frequency plots below (Figs. 5, 6, and 7). The plot of relative power against frequency for a pure unfiltered sine wave would have a single frequency spike at the given frequency of height one.

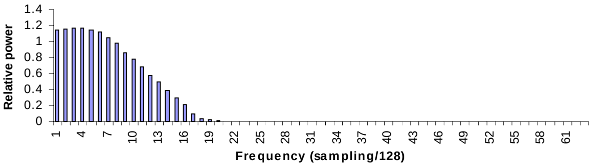


Figure 4**:** Frequency response of evolved lowpass filter

The response of the evolved filter to input signals of various frequencies both in the passband and the stopband are given below.

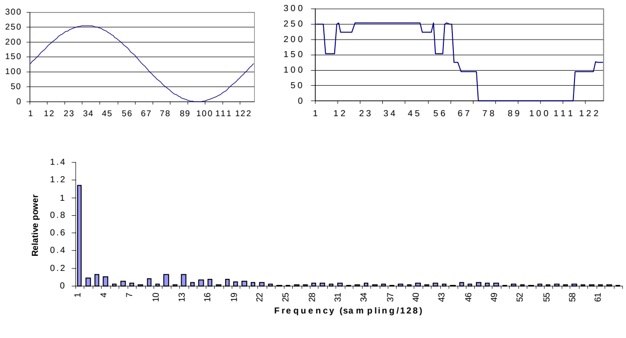


Figure 5: Incident signal f_1 , output response and frequency response

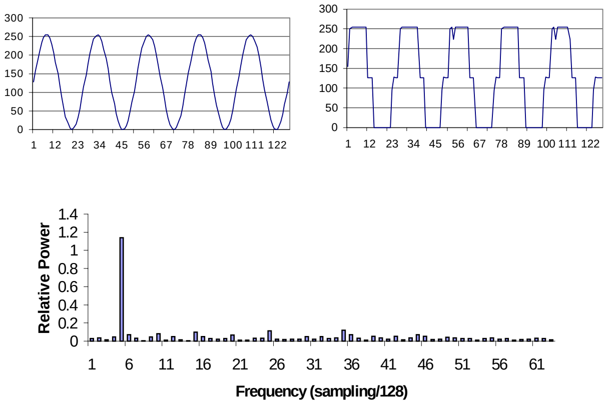


Figure 6: Incident signal f_5 , output response and frequency response

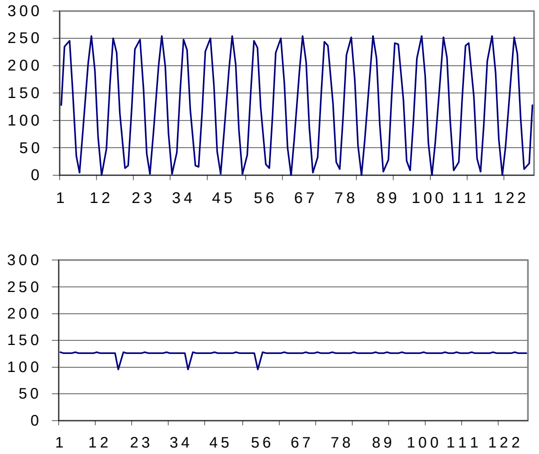


Figure 7: Incident signal f_{20} , and output response

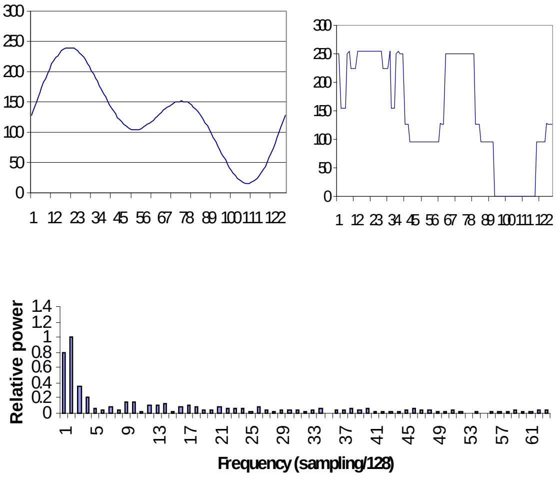


Figure 8: Incident signal $0.5(f_1 + f_2)$, output response and frequency response

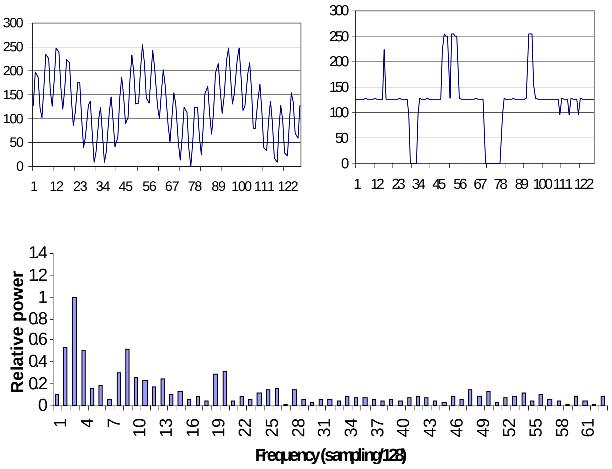


Figure 9: Incident signal $0.5(f_3 + f_{25})$, output response and frequency response

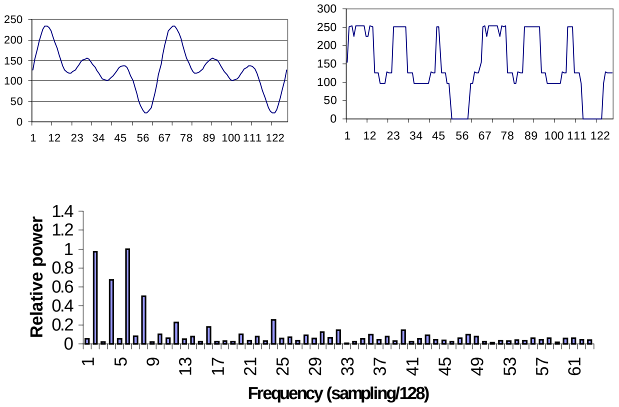


Figure 10: Incident signal $0.33(f_2 + f_4 + f_6)$, output response and frequency response

MULTI-BAND FILTER

In later experiments (which were not exhaustive) a form of $(1+\lambda)$ Evolutionary Strategy was used to find a multi-band filter with two pass regions having frequencies 1-8, and 25-32. Uniform mutation was used equal to 2% of the genes in a chromosome, which in this case equates to 8 genes per chromosome. In this case λ was set at 19. The number of generations was 10,000. The geometry was 10 rows by 10 columns. In this experiment a frequency dependent reward profile was defined as shown in Fig. 11, the fitness was calculated using equation 5. The evolved filter response is shown in Fig. 12. The reasons for choosing this particular reward profile are as follows. There appears to be a natural bias towards lowpass behaviour. Thus the incremental fitness rewards for transparency (2) in the frequency range 1-8 was chosen to be less than the reward for transparency in the second pass band (3), with frequency range 25-32. The reward for the stop region between the two bands was also higher to encourage opacity. The highest rewards for opacity (4) were awarded for a few frequencies either side of the pass regions. This was to encourage a sharp transition from transparency to opacity.

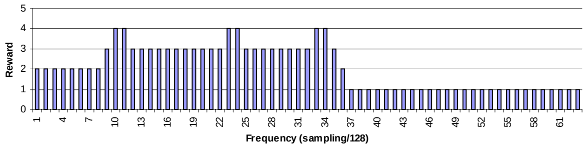


Figure 11: Reward profile for evolving the multi-band filter

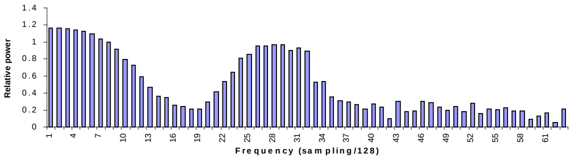


Figure 12: Frequency response of evolved multi-band filter

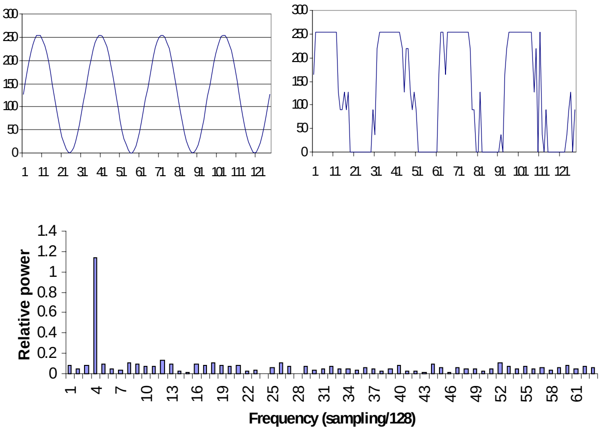


Figure 13: Incident signal f_4, output response and frequency response

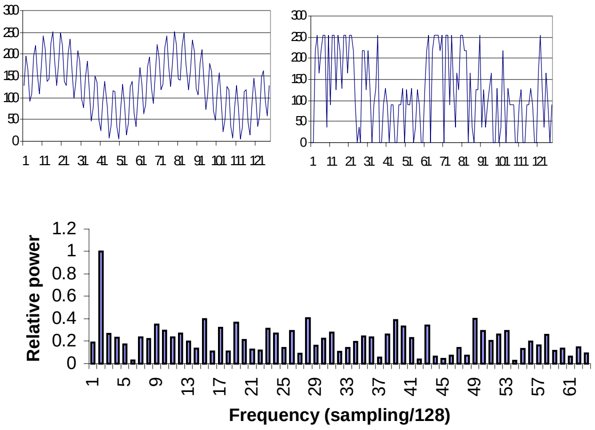


Figure 14: Incident signal $0.5(f_2 + f_{28})$, output response and frequency response

BANDPASS FILTER

A bandpass filter was evolved (see Fig. 15). The passband was 26-35. This proved to be the most difficult task of the three studied. Again a $(1+\lambda)$ Evolutionary Strategy was used with $\lambda = 49$. Mutation rate per chromosome = 2%, 10,000 generations, and levels-back $l = 5$. In this case the reward profile was uniformly set to 1.0.

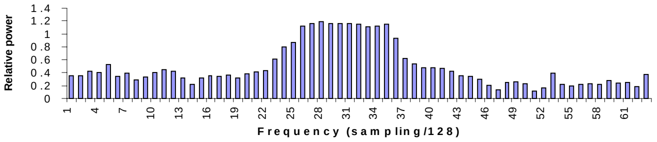
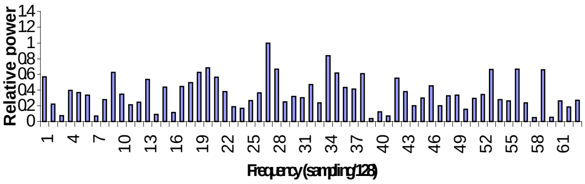
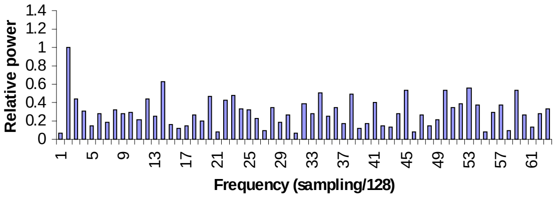


Figure 15: Frequency response of evolved bandpass filter



(a) $0.5(f_{27} + f_{34})$



(b) $0.5(f_2 + f_{30})$

Figure 16: Frequency responses of evolved bandpass filter to incident signals (a), and (b)

5 DISCUSSION OF RESULTS

For the lowpass filter it can be seen that signals in the pass region are transmitted fairly cleanly (in the frequency domain), though there is still noticeable distortion of the output signal (Figs. 5 and 6). Fig. 7 shows the almost d.c. response to a signal in the stop region. Figs. 8-10 show the response of the evolved filter to three composite signals, which have never been seen by the filter before. In Fig. 8 one can see that the dominant response frequencies are f_1 and f_2. If the response had been perfectly linear then the relative power for these would have been identical, with zero power in all other frequencies. Clearly the evolved filter is behaving in a nearly linear fashion. In Fig. 9 a composite signal consisting of a frequency in the pass band (f_3) and one in the stop region (f_{25}) is presented. Again the pass frequency dominates but with some leakage of power to adjacent frequencies. The stop frequency is heavily attenuated. Once again the filter is behaving in a quasi-linear fashion. Fig. 10 shows the near linear behaviour with a sum of three pass frequencies. Looking at the output responses it appears that the filter is exaggerating the changes in the incident signal. In the case of the multi-band problem the evolved filter transmits quite cleanly in the passband (Fig. 13) and attenuates in the stop regions, however the response to a composite signal $0.5(f_2+f_{28})$ is not so linear as with the lowpass filter. Ideally both frequencies would have been transmitted without attenuation. However it can be seen that the f_{28} is highly attenuated (though it is still the second largest transmitted frequency). This behaviour illustrates the better response of these evolved filters to low frequency. Perhaps if the reward profile has been more biased toward transmission of higher passband frequencies a better result might have been obtained. Finally examining the results for the bandpass filter (Figs. 15-16) it can be seen that for a composite signal of two passband (f_{27} and f_{34}), the filter does transmit, though with significant attenuation. Additionally, there is power leakage to other, spurious frequencies. If the composite signal has one frequency in the stopband (f_3) and the other (f_{30}) in the passband, then it should transmit the latter. However it transmits the stop frequency component f_3! This evolved filter is behaving in the least linear fashion of those evolved.

HARDWARE REQUIREMENTS AND SPEED Of EVOLVED FILTER VERSUS CONVENTIONAL DESIGN

When the evolved filter circuit corresponding to Fig. 4 was analysed it was found to require 29 multiplexers (equivalent to 87 two-input gates). In addition the filter would produce the filtered response very quickly as one only has to wait for the signals to propagate through the gate-array. A conventional filter of order 4 and wordlength 8 would require at least an eight-bit adder and multiplier as well as registers to store the coefficients. A conventional cellular multiplier alone of this size would require n^2 AND gates and $n(n-1)$ full adders (where $n=8$). Thus it would require 344 two-input gates. Additionally we would need the gates associated with the eight-bit adder (40 two-input gates) and the register. The output would be delayed by a number of clock cycles to accumulate the response (see equation 1). Thus the evolved circuits are much smaller and quicker than those which could be designed conventionally. However it should be noted that the filters evolved thus far are far from perfect. It is emphasised that the objective of this work was not explicitly to design more efficient circuits but to show for the first time that it is possible to evolve filters without many of the assumptions of conventional techniques, most notably, the absence of an imposed difference equation (eqn. 1).

6 CONCLUSIONS

In this paper it has been shown that it is possible to evolve filtering characteristics with a gate-array containing *very few* components. In some cases the evolved filter has a quasi-linear response which has emerged quite naturally. There is currently no mathematical framework for understanding how to design filters at this level. It is felt that the results presented here may encourage some thinking about a mathematical underpinning of this. There is still an enormous amount of further investigation to be undertaken. The work raises many questions. Why is the evolved filter quasi-linear? Can one evolve it in such a way as to enhance its linearity? Would this require greater gate resources? How would the filter response to changes in phase of the incident sine waves? What would happen if 'off-frequency' signals were presented. The results so far are preliminary, so that one can expect better with a more sophisticated evolutionary strategy. In conclusion it is felt that this work once again demonstrates the enormous capacity of a few gates to display complex behaviours, a fact which has become evident in much work in the field of evolvable hardware (Sipper et al 1997).

References

Arslan T., and Horrocks D. H., "A Genetic Algorithm for the Design of Finite Word Length Arbitrary Response Cascaded IIR Digital Filters", Proc. of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95), No. 414, IEE, London pp. 276-281, 1995.

Bäck T., Hoffmeister F., and Schwefel H.-P., "A Survey of Evolutionary Strategies", Belew R. and L. Booker L. (Eds.), Proceedings of Fourth Int. Conf. On Genetic Algorithms, San Mateo, CA, Morgan Kaufmann, pp. 2-9, 1991.

Beatriz Garmendia-Doval A., Mohan C. K., and Prasad M. K., "Evolving Tree Representations of Stack Filters", in J.R. Koza et al. (eds.), Genetic Programming: Proceedings of the Third Annual Genetic Programming Conference, Morgan Kaufmann, San Francisco, CA, pp. 103-108, 1998.

Chellapilla K., Fogel D. B., and Rao S. S., "Gaining Insight into Evolutionary Programming Through Landscape Visualization: An Investigation into IIR Filtering", Angeline P. et al (Eds.) Evolutionary Programming , Lecture Notes in Computer Science, Vol. 1213, pp. 407-417, 1997.

Delibasis K. K., Undrill P. E., and Cameron G. G., "Genetic algorithm implementation of stack filter design for image restoration", IEE Proceeedings in Vision, Image and Signal Processing, Vol. 143, No. 3, pp. 177-183, 1996.

Dempster A. G., and Macleod M. D., "Use of Minimum-Adder Multiplier Blocks in FIR Digital Filters", IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 42, No. 9, pp. 569-577, 1995

Esparcia Alcazar A. I., and Sharman K. C., "Some Applications of Genetic Programming in Digital Signal Processing", Late Breaking Papers at Genetic Programming 96, Stanford, pp. 24-31, 1996.

Grimbleby J. B., "Automatic Analogue Network Synthesis using Genetic Algorithms", Proceedings of the First

IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95), No. 414, IEE, London pp. 53-58, 1995.

Harris S. P., and Ifeachor E. C., "Automating IIR filter design by genetic algorithm", Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95), No. 414, IEE, London pp. 271-275, 1995.

Iba H., Iwata M., and Higuchi T., "Machine Learning Approach to Gate-Level Evolvable Hardware", Higuchi T. et al (Eds.), Proceedings of The First International Conference on Evolvable Systems: From Biology to Hardware (ICES96), Lecture Notes in Computer Science, Vol. 1259, Springer-Verlag, Heidelberg, pp. 327 – 343, 1997.

Ifeachor E. C., and Jervis B. W., "Digital Signal Processing: A Practical Approach", Addison-Wesley, 1993.

Lohn J. D. and Colombano S. P., "Automated Analog Circuit Synthesis using a Linear Representation", Sipper M. et al (Eds.), Proceedings of The Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98), Lecture Notes in Computer Science, Vol. 1478, Springer-Verlag, Heidelberg, pp. 125-133, 1998.

Miller J. F., and Thomson P., "Evolving Digital Electronic Circuits for Real-Valued Function Generation using a Genetic Algorithm". Koza, J. R. et al, (Eds.). Genetic Programming: Proceedings of the Third Annual Conference, July 22-25, 1998, University of Wisconsin, Madison, Wisconsin. San Francisco, CA: Morgan Kaufmann pp. 863-868, 1998a.

Miller J. F., Thomson P., "Aspects of Digital Evolution: Evolvability and Architecture", Eiben A. et al (Eds.), Proceedings of The Fifth International Conference on Parallel Problem Solving from Nature (PPSNV), Lecture Notes in Computer Science, Vol. 1498, Springer-Verlag, Heidelberg, pp. 927-936, 1998b.

Miller J. F., Thomson P., "Aspects of Digital Evolution: Geometry and Learning", Sipper M. et al (Eds.), Proceedings of The Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98), Lecture Notes in Computer Science, Vol. 1478, Springer-Verlag, Heidelberg, pp. 25-35, 1998c.

Miller J. F., Thomson P., and Fogarty T. C., "Designing Electronic Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study", Quagliarella D. et al (Eds.) Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, Wiley, 1997.

Miller J. F., "Evolution of Digital Filters Using a Gate Array Model", in Poli R, Voigt H-M, Cagnoni S., Smith G., Fogarty T. C, Proceedings of the First EvoIASP'99 Workshop on Image Analysis and Signal Processing, Lecture Notes in Computer Science Vol. 1596, Springer-Verlag, Heidelberg pp.17-30,1999.

Murakawa M., Yoshizawa S., and Higuchi T., "Adaptive Equalisation of Digital Communication Channels Using Evolvable Hardware", Higuchi T. et al (Eds.), Proceedings of The First International Conference on Evolvable Systems: From Biology to Hardware (ICES96), Lecture Notes in Computer Science, Vol. 1259, pp. 379 – 389, 1996.

Murakawa M., Yoshizawa S., Adachi T., Suzuki S., Takasuka K., Iwata M., and Higuchi T., "Analogue EHW Chip for Intermediate Frequency Filters", in Sipper M. et al (Eds.), Proceedings of The Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98), Lecture Notes in Computer Science, Vol. 1478, Springer-Verlag, Heidelberg, pp. 25-35, 1998.

Poli R., "Evolution of graph-like programs with parallel distributed genetic programming", Bäck T. (Ed.), Genetic Algorithms: Proceedings of the Seventh International Conference, Morgan Kaufmann, pp. 346-353, 1997.

Redmill D. W., and Bull D. R., "Design of Low Complexity FIR Filters using Genetic Algorithms and Directed Graphs", Proceedings of the Second IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'97), No. 446, IEE, London, 1997.

Sipper M., Sanchez E., Mange D., Tomassini M., Perez-Uribe A., and Stauffer A., "A Phylogenetic, Ontogenetic, and Epigenetic View of Bio-Inspired Hardware Systems", IEEE Transactions on Evolutionary Computation, Vol. 1, No 1., pp. 83-97, 1997.

Sriranganathan S., Bull D. R., and Redmill D. W., "Design of 2-D Multiplierless FIR Filters using Genetic Algorithms", Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95), No. 414, IEE, London pp. 282-286, 1995.

Sundaralingam S., and Sharman K. C., "Genetic Evolution of Adaptive Filters", Proceedings of DSP, London UK, pp. 47-53, 1997.

Wade G., Roberts A., and Williams G., "Multiplier-less FIR filter design using a genetic algorithm", IEE Proceedings in Vision, Image and Signal Processing, Vol. 141, No. 3, pp. 175-180, 1994.

Zebulum R. S., Pacheco M. A., and Vellasco M., "Comparison of Different Evolutionary Methodologies Applied to Filter Design", 1998 IEEE Int. Conf. On Evolutionary Computation, IEEE Press, Piscataway, NJ, pp. 434-439, 1998.