

Evolution of Digital Filters Using a Gate Array Model

Julian F. Miller

School of Computing, Napier University, 219 Colinton Road, Edinburgh, EH14 1DJ, UK
j.miller@dcs.napier.ac.uk

Abstract. The traditional paradigm for digital filter design is based on 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 small rectangular array of logic gates to perform low pass FIR filtering. The circuit is evolved by assessing its response to digitised pure sine waves. The evolved circuit is demonstrated to possess nearly linear properties, which means that it is capable of filtering composite signals which it has never seen before.

1 Introduction

The difference equation is a fundamental concept employed in the construction and analysis of digital filters [8]. Formally this is represented in the following way. The output of the filter at time n , $y(n)$, may be a function of N samples of the signal $x(n-i)$ at earlier times, and may also, if feedback is present, involve earlier outputs $y(n-i)$ given by the following equation:

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

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. 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. There are many traditional approaches which have been developed to address these issues [8]. In particular one popular method for reducing the complexity of implementation is to restrict the filter coefficients to integer coefficients, see [4] and references therein. Recently, researchers have started to explore the application of evolutionary algorithms to filter design [1] [2] [3] [5] [6] [15] [17] [18] [19]. 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 [1] [6] for IIR filter design, or it may be in an adaptive context [5][18]. Other workers have employed evolutionary algorithms to optimise coefficients together with add and shift operations in so-called multiplier-less designs [15] [17] [19]. In [3] a genetic algorithm was used to design an efficient non-linear 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.

The main idea of the work presented in this paper is to explore for the first time at a logic gate level whether it is possible to evolve networks of logic gates to carry out filtering tasks. 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 [7]. In [13] the authors 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. Their method was not rigidly fixed to be linear in operation, it could be carried out very quickly, and relatively inexpensively in hardware. 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 that 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 evolved gate arrays do appear to be quasi-linear.

The actual method employed here to evolve a gate array (section 2) is developed from earlier work in [9][10][11][12] and has some similarity to a method called Parallel Distributed Genetic Programming (PDGP) [14]. In earlier work [10][11][12], 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 [9] that the real applications for gate-array evolution probably lie in real number mapping problems, where the 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 only a simple low pass FIR filter is considered. The details of this are explained in section 3. In section 4 the evolved filtering characteristics of the gate array are examined,

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. In Fig. 1 is shown 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). The circuit in question actually arose in an earlier experiment reported elsewhere [12] and is quite novel in its own right. A, B, and Cin denote 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. Thus, 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. 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 feed-forward nature of the circuit and removes any time dependent behaviour. Actually the connectivity is further constrained by the presence of a parameter denoted l , which dictates the number of columns on the left (including the primary inputs at column zero) to which the inputs of cells in column l may be connected. The purpose of this is to constrain the fan-out of signals and thereby improve the ease with which the circuit may be routed when it is physically implemented.

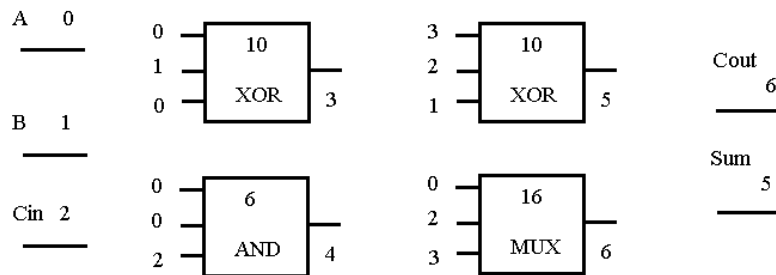


Fig. 1. One-bit adder (with carry-in) implemented as a feed-forward gate array

The chromosome representing the gate array shown in Fig. 1 is given below:

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

where the emboldened integers are the cell functions. 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 , \wedge represents the exclusive-OR operation and $|$ the OR operation.

Table 1. Allowed gate functions

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
0	1	a	b	\bar{a}	\bar{b}	ab	$\bar{a}\bar{b}$	$a\bar{b}$	$\bar{a}b$	$\bar{a}\bar{b}$	$a\bar{b}$	$a\bar{b}$	$a b$	$a b$	$\bar{a} b$	$\bar{a} b$	$\bar{a} b$	$\bar{a} b$	$\bar{a} b$	$\bar{a} b$	$a\wedge(bc)$

Functions 16-19 describe various multiplexers and 20 describes a Reed-Muller ULM. The last five functions prove to be very effective components in assisting the evolutionary process, this is probably due to their flexibility in that they are all universal logic modules and allow the synthesis of any logic function of one or two variables. The genetic algorithm employed random mutation which respected the feed-forward nature of the circuits and also the different alphabets associated with connections and functions. Uniform crossover was employed with a 50% genetic exchange. Elitism was always used as it is markedly beneficial [11]. 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).

3 Evolving a filter response with a gate array

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 r bits. In a FIR filter of order n one therefore must collect nr bits at each sampling time. These nr bits for the s samples are collected and represent the input conditions to the gate array. For each nr input bits the gate array must produce r 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 [8] 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, f_n (half sampling frequency) minus 1. 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 Fig. 2. In this figure an input sine wave is shown on the left which is digitised to binary numbers with wordlength 4 and filter order 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. The desired filter response is characterised by a low pass cutoff point f_p . 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.

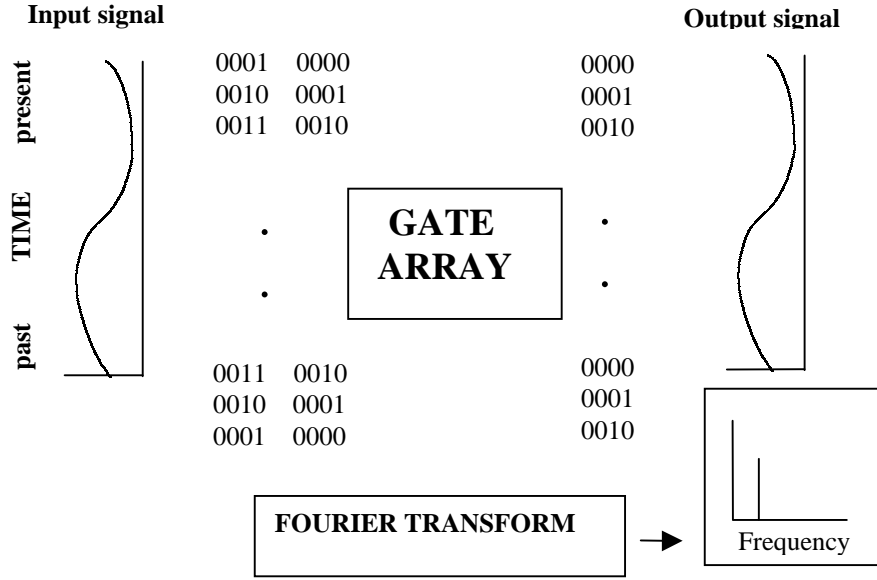


Fig. 2. The training scenario for evolving a gate array with filtering properties

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. The d.c. component of the output is ignored. The fitness x_i of the gate array for a sine wave of frequency f_i is calculated in the following way:

$$x_i = W(f_i) - \max\{W(f_j), \forall j: j \neq i, f_1 \leq f_j \leq f_n - 1\}, f \leq f_p \quad (2)$$

$$x_i = 1.0 - \max\{W(f_j), \forall j: f_1 \leq f_j \leq f_n - 1\}, f > f_p$$

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 $f_n - 1$. Thus if the maximum power for a sine wave with frequency greater than the cutoff point is zero, the fitness contribution will be 1, if the maximum power is not zero then the fitness component will be lower. This definition of fitness means that one is trying to suppress sine waves with frequencies above the cutoff point, and trying to enhance only the *pure* frequencies below the cutoff point. Thus the degree to which the actual shape of the outgoing sine wave conforms to a pure sine wave is being rewarded for frequencies below the cutoff point.

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 .

- number of samples $s=128$, wordlength $r=8$, filter order = 4,
- normalised passband cutoff = 0.08 (10.24 un-normalised)
- population_size is 10, breeding rate is 100%, mutation probability is 0.005
- num_generations is 5000, number of runs is 2, elitism,
- tournament selection (size 2) acceptance probability is 0.7
- number of rows in gate array is 9, number of columns in gate array is 9
- connectivity parameter $l=9$. The only gate type allowed was the multiplexer (type 16).

The results shown in this paper are for the best of two runs of the genetic algorithm under the above conditions. Investigation of the most suitable parameter settings lies outside the scope of this paper. A small population size was chosen purely for speed of execution. The frequency response of the evolved filter is shown in Fig. 3.

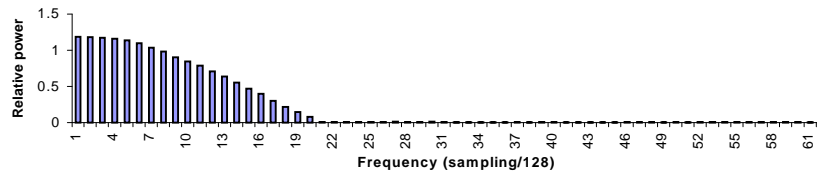


Fig. 3. Frequency response of the evolved filter

4.1 Filter response to pure sine signals in the passband

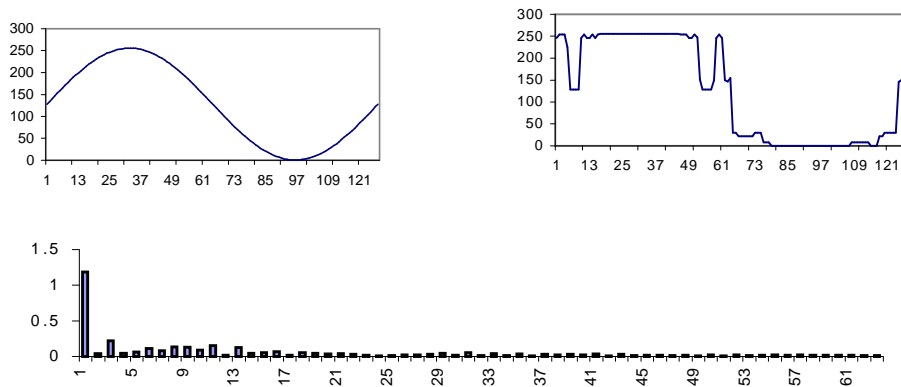


Fig. 4. Incident signal f_I , output response and frequency response

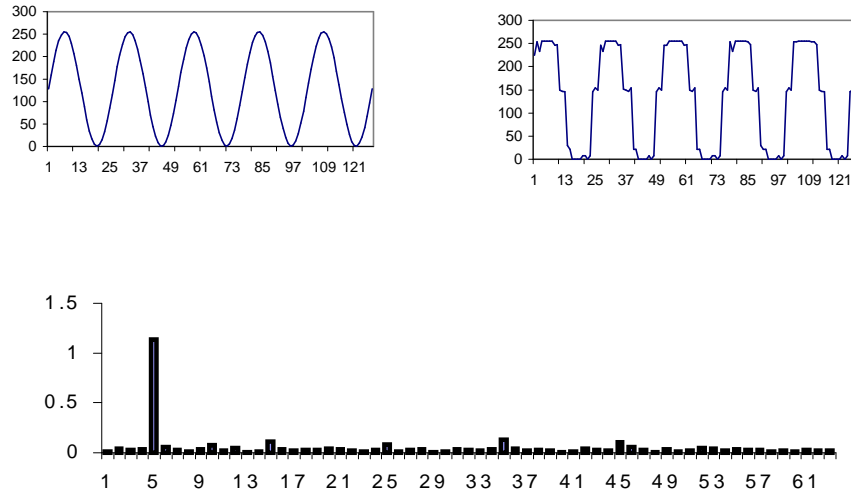


Fig. 5. Incident signal f_5 , output response and frequency response

4.2 Filter response to pure sine signals in the stopband

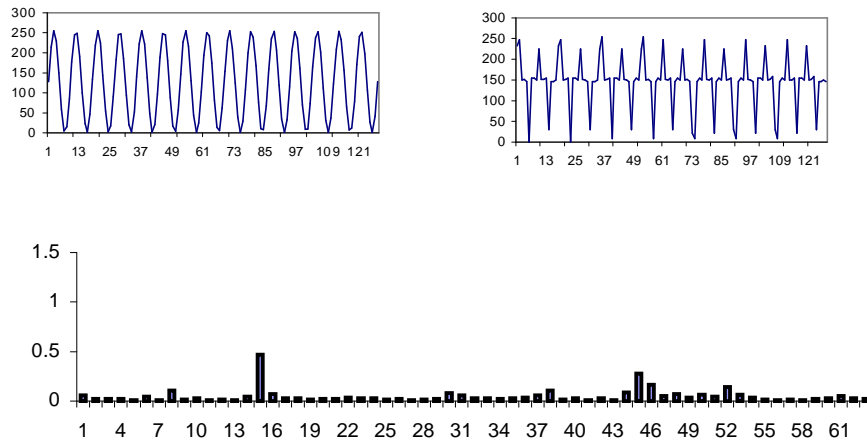


Fig. 6. Incident signal f_{15} , output response and frequency response

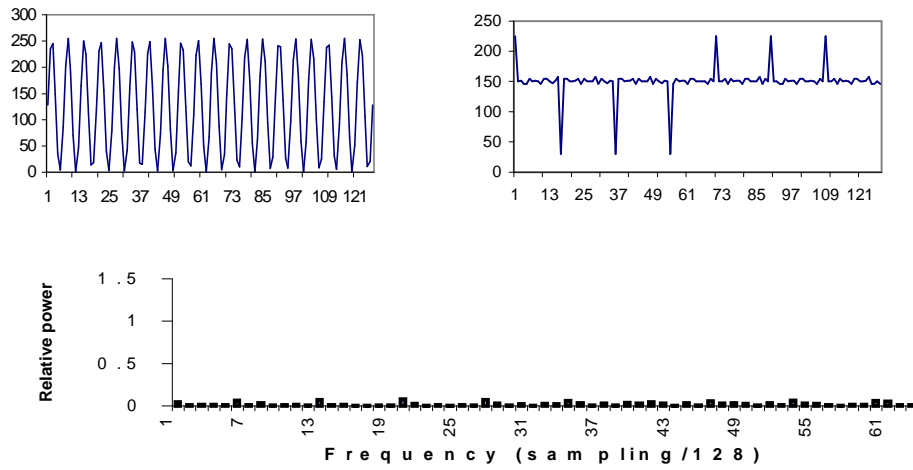


Fig. 7. Incident signal f_{20} , output response and frequency response

4.3 Filter response to signals which are a sum of two sine waves

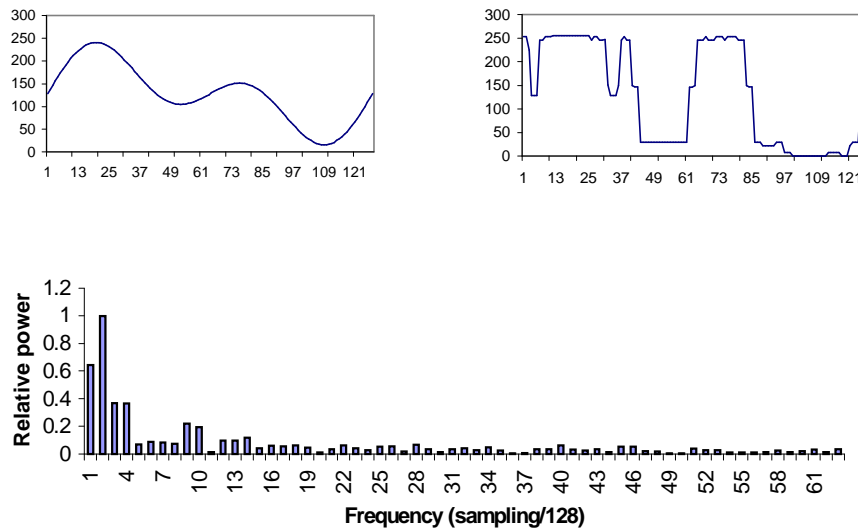


Fig. 8. Incident signal $0.5(f_1 + f_2)$, output response and corresponding frequency response

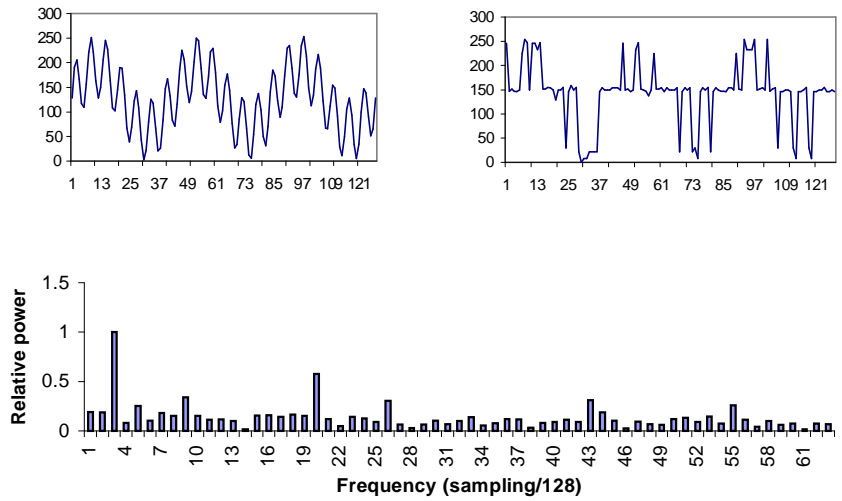


Fig. 9. Incident signal $0.5(f_3 + f_{20})$, output response and corresponding frequency response

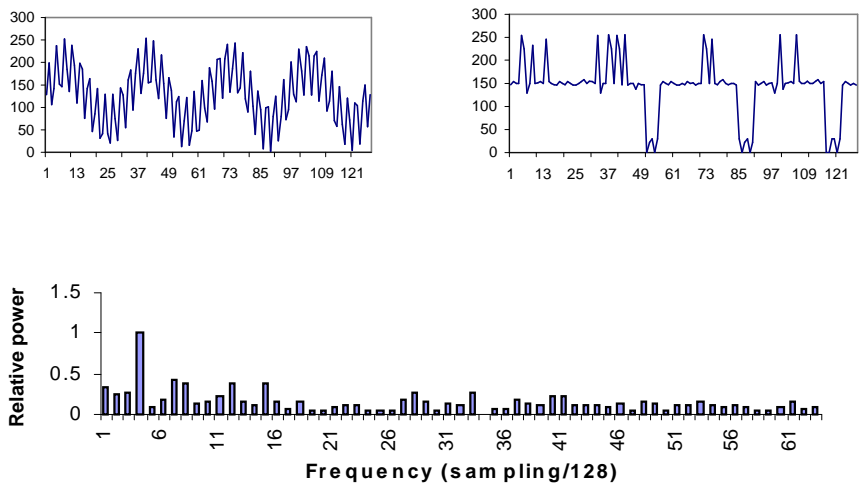


Fig. 10. Incident signal $0.5(f_4 + f_{40})$, output response and corresponding frequency response

4.4 Filter response to signals which are a sum of three sine waves

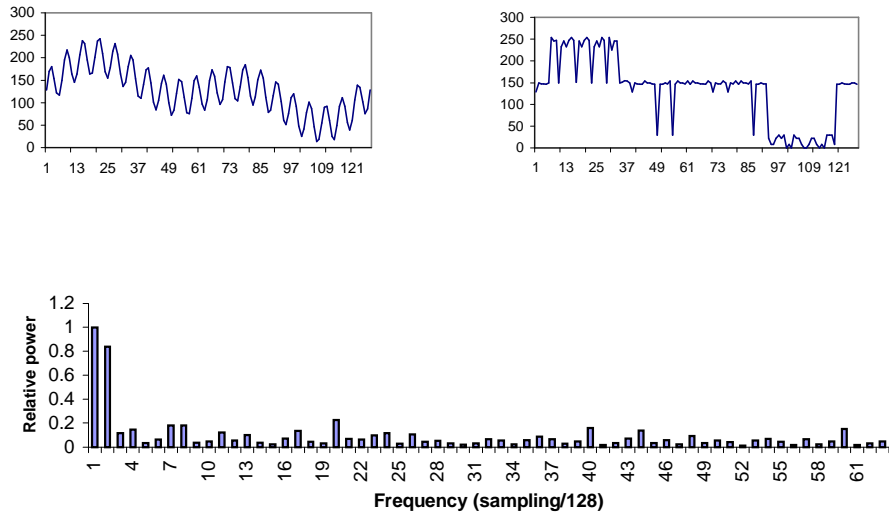


Fig. 11. Incident signal $0.33(f_1 + f_2 + f_{20})$, output response and corresponding frequency response

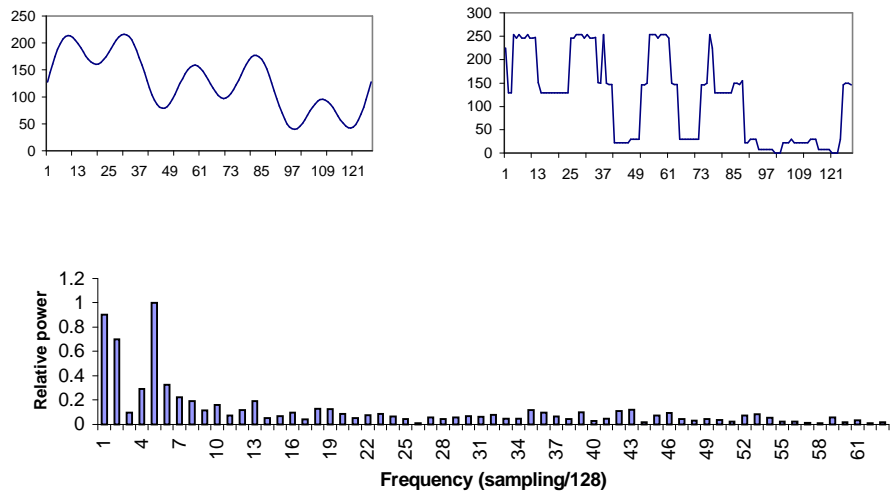


Fig. 12. Incident signal $0.33(f_1 + f_2 + f_5)$, output response and corresponding frequency response

5 Discussion of results

5.1 Filter characteristics

In Fig. 3, the filter response is shown, there is still a noticeable tail which extends past the cutoff frequency of $10 f_i$ from $11 f_i$ to about $20 f_i$. However it should be noted that the gate array is tiny and the work is still at a preliminary phase. The quality of the frequency response in meeting the specification is encouraging. In section 4.1 are shown the output responses of the filter to incident pure sine signals and also the output response in the frequency domain. Sine waves in the passband are being passed with little attenuation, however it can be seen especially in the case of the lowest frequency sine wave (Fig. 4) that there is the largest distortion of the signal. In Figures 6 and 7 the incident sine signals have frequencies in the stopband so they should be highly attenuated. One can see that there is a marked drop in signal amplitude as the signal is converging to a d.c. component. As the frequency of the incident signals are increased the off d.c. spikes become more and more sparse. Actually there is something a little puzzling here as the fitness function is designed to suppress frequencies in the stopband with uniform probability so that the attenuation of those frequencies should show no frequency dependent behaviour. The reason for this is not currently understood but it may be due to a frequency dependent distortion in the incident sine signals. In Figs. 8-12 are shown the output responses of the filter to various sums of sine waves. All these signals have never been seen by the filter before. In Fig. 8 it can be seen that the filter is exaggerating the changes in amplitude of the incident signal. The frequency response shows the dominant frequencies to be the same as the incident. The filter is displaying a nearly or quasi-linear response. In Figs. 9 and 10 the higher frequency lies in the stopband thus for ideal filter behaviour one would expect the higher frequency component to be highly attenuated. The evolved filter appears to be doing this as it is responding to the slower changes in the signal. This is confirmed by the frequency responses. In Figs. 10 and 11 more complex signals were presented to the filter. These were sums of three sine waves. In the first case (Fig. 11) two components were in the passband. Again the filter is still trying to follow the slower changes and the frequency response is dominated by the lower frequencies. In Fig. 12 all the frequencies lay in the passband, again it is seen that the filter is trying to follow all the changes in the incident signal. However once again it is exaggerating the changes.

5.2 Hardware requirements and speed of evolved filter compared with conventional

When the evolved filter circuit 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 adder and multiplier 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. The output would be delayed by a number of clock cycles to accumulate the response (see equation 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. The gate-array filter is produced without many of the conventional assumptions in that it does not employ coefficients or any explicit arithmetic operations. The evolved filter has a quasi-linear response that has emerged 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 almost as many questions as it answers. 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? What would the filtering action of cascades of these smaller filters be like? How would the filter response to changes in phase of the incident sine waves? 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 [16].

Acknowledgements

Especial thanks to Gary Robertson for many helpful discussions, and also, thanks to George Rae, Mohammed Yaminysharif, and Jay Hoy of the EE&CE Department, Napier University.

References

1. Arslan T., and Horrocks D. H., "A Genetic Algorithm for the Design of Finite Word Length Arbitrary Response Cascaded IIR Digital Filters", : Proceedings 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.
2. Chellapilla K., Fogel D. B., and Rao S. S., "Gaining Insight into Evolutionary Programming Through Landscape Visualization: An Investigation into IIR Filtering", Evolutionary Programming 97, pp. 407-417, 1997.
3. Delibasis K. K., Undrill P. E., and Cameron G. G., "Genetic algorithm implementation of stack filter design for image restoration", ", IEE Proceedings in Vision, Image and Signal Processing, Vol. 143, No. 3, pp. 177-183, 1996.
4. 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
5. Esparcia Alcazar A. I., and Sharman K. C., "Some Applications of Genetic Programming in Digital Signal Processing", in Late Breaking Papers at Genetic Programming 96, Stanford, pp. 24-31, 1996.

6. 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.
7. Iba H., Iwata M., and Higuchi T., Machine Learning Approach to Gate-Level Evolvable Hardware, in Higuchi T., Iwata M., and Liu W., (Editors), 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.
8. Ifeachor E. C., and Jervis B. W., "Digital Signal Processing: A Practical Approach", Addison-Wesley, 1993.
9. Miller J. F., and Thomson P., "Evolving Digital Electronic Circuits for Real-Valued Function Generation using a Genetic Algorithm". Koza, John R. et al, (Editors). 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, 1998.
10. Miller J. F., Thomson P., "Aspects of Digital Evolution: Evolvability and Architecture", in (Editors), 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, 1998.
11. Miller J. F., Thomson P., "Aspects of Digital Evolution: Geometry and Learning", in Sipper M., Mange D., and Perez-Urbe A. (Editors), *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.
12. Miller J. F., Thomson P., and Fogarty T. C., "Designing Electronic Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study", in Genetic Algorithms and Evolution Strategies in Engineering and Computer Science: D. Quagliarella, J. Periaux, C. Poloni and G. Winter (eds), Wiley, 1997.
13. Murakawa M., Yoshizawa S., and Higuchi T., "Adaptive Equalisation of Digital Communication Channels Using Evolvable Hardware", in Higuchi T., Iwata M., and Liu W., (Editors), 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. 379 – 389, 1996.
14. Poli R., "Evolution of graph-like programs with parallel distributed genetic programming", in Bäck T. (Editor), Genetic Algorithms: Proceedings of the Seventh International Conference, Morgan Kaufmann, pp. 346-353, 1997.
15. Redmill D. W., and Bull D. R., "Design of Low Complexity FIR Filters using Genetic Algorithms and Directed Graphs", in Proceedings of the Second IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'97), No. 446, IEE, London, 1997.
16. Sipper M., Sanchez E., Mange D., Tomassini M., Perez-Urbe 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.
17. 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.

18. Sundaralingam S., and Sharman K. C., "Genetic Evolution of Adaptive Filters", in Proceedings of DSP, London UK, pp. 47-53, 1997.
19. 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.