Coefficient Sensitivity of Polynomial-Predictive FIR Differentiators: Design for Short Word Length

Jarno M. A. Tanskanen and Seppo J. Ovaska

Institute of Intelligent Power Electronics
Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, FINLAND
Tel. +358-9-451 2446, Fax: +358-9-460 244, E-mail: jarno.tanskanen@hut.fi

Abstract — **In this paper, a genetic algorithm for designing polynomial-predictive FIR differentiators (PPFD) for short word length fixed-point realizations is presented. Critical design aspects are pointed out, and a fitness function for genetic algorithms is designed to strive for desired prediction and differentiation properties with filter coefficients quantized to eight bits. The genetic algorithm is in many cases found to produce better quantized coefficient filters than the original quantized coefficient PPFDs.**

I. INTRODUCTION

Both predictive and derivative signal processing have been introduced into diverse range of control applications, like radio transmitter power control [1], and motion control, for examples, controlling motor drives of elevator cars [2,3]. In these examples, the main feature of interest is the physical near piece-wise polynomial nature of control input signal; Rayleigh fading radio channel power response, or position or velocity information of the elevator car.

The basics of polynomial-predictive filtering [4,5] as well as of polynomial-predictive differentiating filtering [2,3] are well established. While many implementation of the filters are done with general purpose processors or digital signal processors, still many applications require cost effective mass production of application specific integrated circuits (ASIC). To reduce ASIC cost, silicon area, and power consumption, short word length implementations are naturally highly desirable. Realizing a filter with a short coefficient word length, e.g. 8 bits, naturally introduces errors to the filter coefficients. Effects of coefficient quantization to PPFDs have been analyzed in [6] where also guidelines for word length robust direct, lattice structure, and sectioned implementations are proposed. In this paper, a genetic algorithm with a fitness function is proposed for designing PPFDs for short word length applications. Genetic algorithms are used in two ways; to produce new filters from a diverse initial population without specifying the filter length or, to find a short word length solution from a smaller search space for a given filter length.

The paper is organized as follows: In the Section II, PPFDs are shortly reviewed, along with a short review on the coefficient quantization effects, elaborated upon in [6]. In Section III, a genetic algorithm is proposed to produce quantized coefficient PPFDs, and some genetic filter design results are given in Section IV. Section V presents the conclusions.

Work of J. M. A. Tanskanen has been supported by Jenny ja Antti Wihuri Foundation, Finland, Walter Ahlström Foundation, Finland, and by The Finnish Society of Electronics Engineers.

II. POLYNOMIAL-PREDICTIVE FIR DIFFERENTIATORS AND COEFFICIENT QUANTIZATION EFFECTS

General predictive filtering theory has been well established [7]. Here we concentrate on polynomial-predictive FIR differentiators (PPFD), derived much like Heinonen-Neuvo polynomial predictors in [4], except that the filter output is defined as [2]

$$\sum_{k=0}^{N-1} h(k)x(n-k) = \dot{x}(n+m) \tag{1}$$

where $h(k)$ are filter coefficients, N is filter length, $x(n)$ are filter input samples, m is a prediction step, and the dot denotes time derivative. From (1) a set of linear constraints on filter coefficients can be derived [2]:

$$g_0 = \sum_{k=0}^{N-1} h(k) = 0\,, \tag{2}$$

$$g_1 = \sum_{k=0}^{N-1} (N-k-1)h(k) = 1\,, \tag{3}$$

$$g_2 = \sum_{k=0}^{N-1} (N-k-1)^2 h(k) = 2(N-1+p)\,, \tag{4}$$

$$\vdots$$

$$g_M = \sum_{k=0}^{N-1} (N-k-1)^M h(k) = M(N-1+p)^{M-1}\,, \tag{5}$$

providing for the prediction and differentiation of the polynomials of degrees 0, …, M. From the constraints (2)-(5) can closed form solutions for the PPFD coefficients for low-degree polynomial input signals be found by the method of Lagrange multipliers [2]. The rest of the degrees of freedom are used to minimize the noise gain

$$NG = \sum_{k=-\infty}^{\infty} |h(k)|^2\,. \tag{6}$$

The coefficients for the one-step-ahead second degree PPFDs, M = 2, m = 1, are given by [2]

$$h(k)=\frac{6\left[(30N-30)k^2+(-32N^2+38)k+6N^3-11N^2-9N+14\right]}{(N-2)(N-1)N(N+1)(n+2)},$$
$$k=0,1,\ldots,N-1\,. \tag{7}$$

Fig. 1 provides a typical example of exact and degraded frequency responses of a one-step-ahead PPFD. In Fig. 1 a) the frequency response of the second degree one-step-ahead PPFD of length N = 11 is shown with exact coefficients (7) and with coefficients quantized to 8 bits, along with the ideal PPFD frequency response. The corresponding group delays are shown in Fig. 1 b). One-step-ahead prediction, m = 1, can be identified as the negative unity group delay in Fig. 1 b), and the differentiation property is set by the zero magnitude response at zero frequency along with the ramp-shaped response near zero frequency, Fig. 1 a). As also seen in Fig. 1, differentiation property is generally more robust to coefficient quantization than prediction property which can be lost already with the coefficient word length of 16 bits. Finite word length effects are analyzed more closely in [6] where also some short word length robust designs are proposed. An exhaustive search method for finding the same type of quantized coefficient filters as in this paper, but satisfying exactly the constraints (2)-(5), is proposed in [8], and a fast search algorithm yielding exact fixed-point solutions for a related FIR family is proposed in [9].

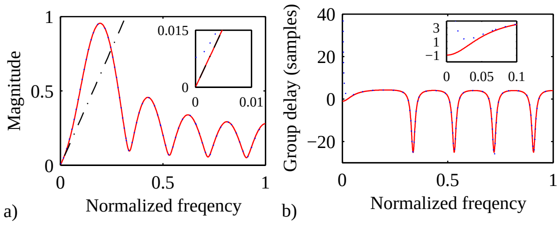

Fig. 1. a) Frequency responses of the second degree one-step-ahead PPFD of length N = 11 with coefficient word length of 8 bits (dotted) and with exact coefficients (solid), along with the frequency response of the ideal PPFD (dash-dot). b) Group delay of the same filter with quantized (dotted) and exact coefficients (solid).

III. GENETIC ALGORITHM FOR PPFD DESIGN

The genetic algorithm [5,10] applied in this paper is a simple and straight forward design that is aimed at producing quantized coefficient PPFDs that exhibit closer to ideal magnitude response and group delay performances than the corresponding filters obtained from (7) after coefficient quantization. In this paper, we consider only second degree, M = 2, one-step-ahead, m = 1, PPFDs. For fixed-point number representation, two's complement presentation with magnitude truncation is used. Matlab's full computational precision is considered the exact (infinite precision) number presentation.

A. Genetic Algorithm in General

For a genetic algorithm, solutions of a problem are to be formulated in a form of strings of bits, or numbers, which evolve through generations by mutation, crossover and reproduction [10]. For each solution in each generation, a fitness function is evaluated. Generally, the fittest solutions are preferred, and the population hopefully includes more and more fitter solutions as the generations pass. Here, the solutions are vectors whose elements are filter coefficients presented with eight bits.

To preserve the differentiating property under coefficient quantization would require that the quantized coefficients fulfilled the constraints (2)-(5) exactly. It is clear that analytically finding quantized coefficient that fulfilled (2)-(5) would be a very tedious task. Here we apply a genetic algorithm in search of quantized coefficient filters that behave as much as possible like the ideal second degree one-step-ahead PPFD in the sense of the frequency response and group delay at a low frequency range.

B. Fitness Function

Our fitness function (error) e (8) that is to be minimized, consists of the mean squared errors (MSE) of the frequency response and group delay of a candidate. The errors are calculated with respect to the ideal differentiator frequency response, c.f. the ideal ramp-like PPFD frequency response in Fig. 1 a), and the ideal one-step-ahead predictor group delay of negative unity. The fitness function to minimize is given by

$$e=e_F+e_G \tag{8}$$

with squared magnitude response and group delay errors

$$e_F=\sum_{f_F}\left(F_{quant}(f_F)-F(f_F)\right)^2, \tag{9}$$
$$e_G=\sum_{f_G}\left(G_{quant}(f_G)-G(f_G)\right)^2, \tag{10}$$

respectively. In (9) and (10), F and G are values of the frequency response and group delay of the ideal filters calculated at sets of frequency points f_F and f_G, respectively, and the subscript *quant* denotes the corresponding responses of candidate solutions. Here we have used frequency point sets f_F and f_G with normalized frequencies

$$f_F=\{0.0001, 0.001\}\ \text{and}\ f_G=\{0.0001, 0.001\} \tag{11}$$

to stress the frequency domain behavior in the low frequency band where these filters are to exhibit prediction and differentiation properties.

Another possibility would be to create a fitness function directly based on the constraints (2)-(5). This would speed up the algorithm, and provide clear indication of the found solutions that exactly fulfill the constraints (2)-(5) but nonzero fitness (error) values would require extra interpretation, since such a fitness function value could not directly indicate how good the frequency response and group delay characteristics of a candidate solution were.

C. Initial Population and Mutation

Searching the whole quantized coefficient space with totally random initial populations would be a prohibitively large task. Here we unfortunately have excellent initial guesses available, namely the filters whose coefficients given by (7). Some experiments were conducted with initial populations of 500 filters consisting of five randomly mutated copies of the filters of lengths $N = 3, \ldots, 102$, hoping that the algorithm would converge faster since it had a diverse initial population and also the filter length could vary. With different parameters values for the genetic algorithm, this approach never produced a filter that had both satisfactory frequency response and group delay properties.

Instead, like successfully applied in [8], we here restrict each initial solution within $\pm 2/2^8$ from the coefficients given by (7) after quantization. Also coefficient changes in mutations are restricted to $\pm 1/2^8$, $\pm 2/2^8$. This means that each mutation may affect the two least significant bits. Since location of the binary point is set for each coefficient according to its magnitude, larger coefficients are not affected. In this approach, it is the designer's task to select the filter length, and the genetic algorithm is used to find a better behaving quantized coefficient solution. Mutation probabilities applied are $P_M = 0.1/N$, $1/N$, $4/N$. Mutation probability over each run is fixed.

Initial populations of 500 filters are formed from the filter of the selected length, provided by (7), after coefficient quantization to eight bits. Each initial population consists of 10 unmutated copies of this quantized coefficient filter, 290 filters with coefficients mutated with the probability of $1/N$ randomly by $\pm 1/2^8$, and 200 filters with coefficients mutated with the probability of $1/N$ randomly by $\pm 1/2^8$ or $\pm 2/2^8$.

D. Crossover, Elitism, and Reproduction

With a fixed probability over each run, pairs of filters are selected for crossover with probability $P_{CO} = 0.1$, 0.4, 0.8, and crossover is performed at a random location. Note that crossover is done between coefficients, i.e., each coefficient value is preserved in crossover. Mutation and crossover probabilities for each run are set in pairs as

$$(P_M, P_{CO}) = \{(0.1/N, 0.8), (1/N, 0.4), (4/N, 0.1)\}. \quad (12)$$

Also, elitist strategy is employed; two copies of the 10 fittest filters, i.e., of the 10 filters with minimum e (8), are passed unaltered to the next generation, and also the 40 next fittest filters are copied directly to the next generation. The rest 450 filters are killed. The 50 fittest filters reproduce 440 new filters by roulette wheel selection with probabilities that are inversely proportional to their noise gains (6). This way we can strive for minimizing both frequency response and group delay errors, and also noise gains, without having to create one single fitness function that would take into account all these aspects.

IV. Design Results

The genetic algorithm was run for the filter lengths $N = 40$, and $N = 80$. The algorithm was run for 300 generations, and the best filters and their fitnesses were recorded for each generation. For each parameter pair (12) and filter length, results of three runs are summarized in Table 1. For each of these cases, fitness (8) of the best candidate solution in the initial population, and the fitness of the best solution after a number of generations are shown in Table 1, along with the noise gains (6) of the latter. For the filter length $N = 40$, with the parameters $(P_M, P_{CO}) = (0.1/N, 0.8)$ the algorithm did not converge to anything usefull. For the parameters $(P_M, P_{CO}) = (1/N, 0.4)$, two out of the three runs produced quite nice filters in 165 and 214 generations, and with $(P_M, P_{CO}) = (4/N, 0.1)$ all the runs produced filters with desired qualities in 4, 116, and 136 generations. Similarly, the results for the filter length $N = 80$ are listed in Table 1. From Table 1 it is seen that for filter length $N = 40$, convergence gets better as the mutation probability P_M is increased and crossover probability P_{CO} is decreased. Though the statistics shown in Table 1 are not sufficient for removing the effect of the fitness of the initial population for drawing conclusions, this conclusion can be drawn from the number of converged runs with each parameter pair.

TABLE 1.
RESULTS OF THREE RUNS OF THE GENETIC ALGORITHM FOR THE THREE PARAMETER PAIRS (P_M, P_{CO}) (12), AND FILTER LENGTHS $N = 40$ AND $N = 80$. FOR EACH RUN, THE FITNESS (ERROR) (8) OF THE BEST CANDIDATE IN THE INITIAL POPULATION I, AND THE FITNESS OF THE BEST CANDIDATE F AFTER G GENERATIONS, ARE SHOWN, ALONG WITH THE NOISE GAINS NG OF THE FOUND SOLUTIONS. NOISE GAINS OF THE EXACT COEFFICIENT FILTERS OF LENGTHS $N = 40$ AND $N = 80$ ARE $3.2\cdot10^{-3}$ AND $3.8\cdot10^{-4}$, RESPECTIVELY. THE CASES MARKED N/A DID NOT CONVERGE TO ANY APPROPRIATE SOLUTION.

		$(P_M,P_{CO}) = (0.1/N,0.8)$			$(P_M,P_{CO}) = (1/N,0.4)$			$(P_M,P_{CO}) = (4/N,0.1)$		
$N =$		run 1	run 2	run 3	run 1	run 2	run 3	run 1	run 2	run 3
40	I	0.25	11.76	0.021	11.27	6.94	16.26	0.034	1.47	0.065
	G	N/A	N/A	N/A	214	165	N/A	116	4	136
	F	N/A	N/A	N/A	$8.6\cdot10^{-6}$	$1.6\cdot10^{-6}$	N/A	$1.9\cdot10^{-6}$	$4.2\cdot10^{-5}$	$4.3\cdot10^{-6}$
	NG	N/A	N/A	N/A	$2.6\cdot10^{-3}$	$3.9\cdot10^{-3}$	N/A	$4.3\cdot10^{-3}$	$3.6\cdot10^{-3}$	$3.6\cdot10^{-3}$
80	I	1.96	$7.4\cdot10^{-3}$	2.88	3.21	2.34	0.18	68.65	14.11	32.91
	G	100	100	181	6	20	12	164	100	2
	F	$2.6\cdot10^{-5}$	$5.3\cdot10^{-5}$	$6.9\cdot10^{-5}$	$4.2\cdot10^{-3}$	$2.4\cdot10^{-3}$	$2.2\cdot10^{-3}$	$3.9\cdot10^{-5}$	$2\cdot10^{-4}$	$2.1\cdot10^{-3}$
	NG	$6.7\cdot10^{-4}$	$6.1\cdot10^{-4}$	$8.9\cdot10^{-4}$	$6.7\cdot10^{-4}$	$1.0\cdot10^{-3}$	$7.6\cdot10^{-4}$	$2.7\cdot10^{-3}$	$1.4\cdot10^{-3}$	$7.0\cdot10^{-4}$

It was noted that not having the zero frequency included in the frequency point sets (11) for calculating the fitness function (8), and also calculating the errors (9) and (10) with respect to the ideal frequency response and group delay, and not with respect to the frequency response and group delay of the filters given by (7), caused some quite good initial solutions to be discarded. Also because of these reasons, fitness values do not provide full knowledge of the properties of the solutions, and the solutions listed in Table 1 are thus not selected strictly according to the values of the fitness function, but also according to a visual inspection of frequency responses and group delays near the zero frequency.

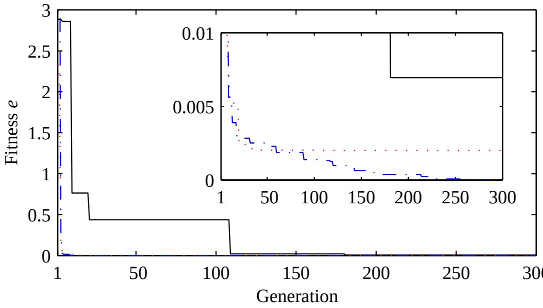

Fig. 2. Fitness (8) of the best solution in each generation over one run for filter length $N = 80$ with each of the parameters (12); $(P_M, P_{CO}) = (0.1/N, 0.8)$, run 3 (solid), $(P_M, P_{CO}) = (1/N, 0.4)$, run 2 (dotted), and $(P_M, P_{CO}) = (4/N, 0.1)$, run 1 (dash-dot).

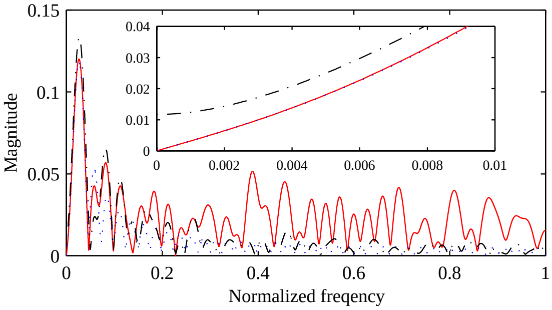

Fig. 3. Frequency response of the solution found for $N = 80$ with $(P_M, P_{CO}) = (0.1/N, 0.8)$ after 181 generations (run 3 in Table 1) (solid). Also shown are the frequency response of the exact coefficient filter given by (7) for $N = 80$ (dotted), and of the same filter with the coefficients quantized to 8 bits (dash-dot).

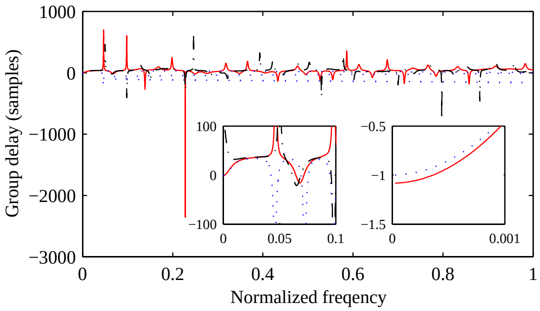

Fig. 4. Group delays of the filters shown in Fig. 3; genetic solution (solid), exact coefficient filter (dotted), and the exact coefficient filter after coefficient quantization to 8 bits (dash-dot).

In Fig. 2, a representative fitness value history of the best solution in each generation for one run with each parameter pair (12) is shown for the filter length $N = 80$. Figs. 3 and 4 show a representative solution found by the algorithm for $N = 80$ and $(P_M, P_{CO}) = (0.1/N, 0.8)$ after 181 generations, run 3 in Table 1. From Figs. 3 and 4, it can be seen that the algorithm has found a PPFD whose properties near zero frequency closely resemble those of the exact coefficient filter obtained from (7). The found solution is also much better than the corresponding ordinary 8-bit coefficient filter shown in Figs. 3 and 4.

V. Conclusions

We have demonstrated that genetic algorithms can find quantized-coefficient PPFDs with close to desired magnitude response and group delay properties, though the filters produced by genetic algorithms have to be screened carefully. The most difficult coupling between zero locations and quantized coefficients makes quantized-coefficient filter design a lucrative challenge to genetic optimization. Genetic algorithms are well applicable to the problem since the exact infinite precision filter coefficients are known, and can be used in forming the initial populations, and search space can be limited to a close vicinity of the initial solutions. Genetic algorithms are found capable of finding quantized coefficient PPFDs which are much better than the original quantized coefficient filters. Also, one important aspect of programming and running genetic filter design algorithms is that a filter designer can gain much useful insight into the meaning of different filter quality measures by observing the evolution of filter populations with different fitness functions.

References

[1] P. T. Harju, T. I. Laakso, and S. J. Ovaska, “Applying IIR predictors on Rayleigh fading signal,” *Signal Processing*, vol. 48, pp. 91–96, Jan. 1996.

[2] S. Väliviita and O. Vainio, “Delayless differentiation algorithm and its efficient implementation for motion control applications,” in *Proc. IEEE Instrumentation and Measurement Technology Conf.*, St. Paul, MN, USA, May 1998, pp. 881–886.

[3] S. Väliviita and S. J. Ovaska, “Delayless recursive differentiator with efficient noise attenuation for control instrumentation,” *Signal Processing*, vol. 69, pp. 267–280, Sept. 1998.

[4] P. Heinonen and Y. Neuvo, “FIR-median hybrid filters with predictive FIR substructures,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, pp. 892–899, June 1988.

[5] P. T. Harju and S. J. Ovaska, “Optimization of polynomial predictive IIR filters using genetic algorithms,” in *Proc. 3rd Int. Conf. on Signal Processing*, Beijing, China, Oct. 1996, pp. 87–90.

[6] J. M. A. Tanskanen and S. J. Ovaska, “Coefficient sensitivity of polynomial-predictive FIR differentiators: Analysis”, in *Proc. 42nd Midwest Symposium on Circuits and Systems*, Las Cruces, NM, USA, Aug. 1999, in press.

[7] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. New York, NY, USA: Macmillan Publishing Company, 1992.

[8] V. S. Dimitrov and J. M. A. Tanskanen, “Round-off error free fixed-point design of polynomial-predictive FIR differentiators,” in *Proc. IASTED Int. Conf. on Intelligent Systems and Control*, Santa Barbara, CA, USA, Oct. 1999, in press.

[9] J. M. A. Tanskanen and V. S. Dimitrov, “Round-off error free fixed-point design of polynomial FIR predictors,” in *Proc. 33rd Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Oct. 1999, in press.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.