

ROUND-OFF ERROR FREE FIXED-POINT DESIGN OF POLYNOMIAL-PREDICTIVE FIR DIFFERENTIATORS

VASSIL S. DIMITROV¹ and JARNO M. A. TANSKANEN²

¹Laboratory of Signal Processing and Computer Technology
Institute of Radio Communications
Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, Finland
Tel. +358-9-451 2455, Fax: +358-9-460 224
E-mail: vdimitro@wooster.hut.fi

²Institute of Intelligent Power Electronics
Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, Finland
Tel. +358-9-451 2446, Fax: +358-9-460 224
E-mail: jarno.tanskanen@hut.fi

Keywords: Filtering, quantization error, Diophantine equations, integer programming, prediction, differentiation

Abstract: In this paper, we present a novel method for designing polynomial-predictive FIR differentiators for fixed-point environments. Our method yields filters that exactly fulfill the given constraints even with short coefficient word lengths. Under ordinary quantization, either by rounding or truncation, these filters in most cases lose their predictive and/or differentiating properties, making their fixed-point implementations useless. With the method proposed in this paper, the filters are designed so that the desired properties are exactly preserved in fixed-point implementations. The presented filter design method is based on integer programming (IP) and can be directly applied to any fixed-point FIR design specifications which can be stated in a form of linear constraints on filter coefficients.

I. INTRODUCTION

By their nature, digital devices handle numbers using a finite number of bits per digit [1]. On the other hand, digital filters are typically designed using general-purpose computers. When the target application has the same computation precision as the filter design environment, there are usually no implementation problems if the filter itself was appropriately designed. Many times this is not the case, however, but the filters are operating within inexpensive, fixed-point processors, or in embedded applications using highly optimized, small and less power-consuming application specific integrated circuit (ASIC) designs. In these cases, there might be a great difference between the calculation precisions of the filter design environment and the final operation platform. This obviously results in filter quality degradation and possibly even in a totally unintended kind of filtering operation. In this paper, we present a novel method for designing polynomial-predictive FIR differentiators [2] whose quantized coefficients exactly fulfill the set constraints.

In many engineering disciplines, accurate control of processes is absolutely necessary. In turn many of the real world physical process parameters exhibit more or less smooth transitions. Noisy measurements of these parameters are then used for process control after a delay. Our examples of closed loop control include motion control of an elevator car [2], and mobile phone power control [3]. In the latter, the inherent closed loop control delays make it a lucrative environment to apply polynomial predictive techniques since the received power fluctuations can in many cases be modeled as Rayleigh distributed signals which in turn can be accurately modeled as piece-wise low degree polynomials. Accurate control of an elevator car can effectively utilize, not only predicted position, but also predicted velocity and acceleration information. This information can be made available to the controller by a predictive differentiator. Here again, the position and velocity of the elevator car can be accurately modeled as piece-wise polynomial. Should these controllers be implemented in a fixed-point environment, which is definitely desirable in a mobile handset, the actual properties of the quantized-coefficient filters are crucial, thus making most of the filters inapplicable. As the method presented in this paper yields quantized-coefficient filters that exactly fulfill the given constraints, these filters are naturally safe for even critical applications in short word length fixed point environments.

In Section II, predictive FIR differentiators are shortly reviewed along with the constraints that are to be exactly fulfilled by the coefficients to provide for the desired filter properties. Also the coefficient quantization effects are shortly discussed in Section II. Integer programming interpretation of fixed-point polynomial-predictive FIR differentiator design and the proposed design method are given in Section III. Characteristics of the quantized-coefficient and ideally quantized-coefficient filter are illustrated in Section IV. Further research topics are discussed in Section V, and Section VI concludes the paper.

II. PREDICTIVE FIR DIFFERENTIATOR IN FIXED-POINT ENVIRONMENTS

A. Predictive FIR Differentiators

Predictive filtering theory has been well established [2,3,4,5]. Here we concentrate on polynomial-predictive FIR differentiators, whose applicability has suffered from practical constraint of finite coefficient precision. Polynomial-predictive FIR differentiators, derived in [2], assume a low-degree polynomial input signal contaminated by white Gaussian noise. Filter output is defined to be a p -step-ahead predicted time derivative of the input,

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

where $h(k)$ are filter coefficients, $x(n)$ are input samples, N is filter length, p is a prediction step, and the dot denotes time derivative. After providing for exact prediction and differentiation, the rest of the degrees of freedom are used to minimize the white noise gain, given by

$$NG = \sum_{k=0}^{N-1} |h(n)|^2. \quad (2)$$

In [2], a feedback extension to FIR differentiators is given to provide considerable noise attenuation while maintaining the prediction and differentiation properties set forth by the underlying predictive FIR differentiator. In order for the feedback extension to function properly, it is necessary that the underlying FIR basis filters are implemented exactly. Until now, this has been rarely possible in short word length fixed-point environments.

A set of constraints can be derived from the definition of the filter output (1) [2]:

$$g_0 = \sum_{k=0}^{N-1} h(k) = 0, \quad (3)$$

$$g_1 = \sum_{k=0}^{N-1} (N-k-1)h(k) = 1, \quad (4)$$

$$g_2 = \sum_{k=0}^{N-1} (N-k-1)^2 h(k) = 2(N-1+p), \quad (5)$$

⋮

$$g_M = \sum_{k=0}^{N-1} (N-k-1)^M h(k) = M(N-1+p)^{M-1}. \quad (6)$$

The constraints (3)-(6) give the prediction and differentiation for the polynomial degrees $0, \dots, M$, and from them can closed form solutions for the FIR coefficients for low-degree polynomial input signals be found by the method of Lagrange multipliers [6]. The closed form solution for FIR coefficients for the first degree polynomial

input signals is given in [5], and for the second degree in [2]. Since differentiation of a first degree polynomial input signal is, in a way, trivial from an application point of view, in this paper we use the case with the highest polynomial input signal component degree of two, $M=2$, as an example. In this case we have to fulfill the constraints (3), (4) and (5), and use the remaining degrees of freedom to minimize the noise gain (2). The exact, i.e., infinite precision, coefficients for the second degree polynomial-predictive one-step-ahead, $p=1$, FIR differentiators are given by [2]

$$\frac{6[(30N-30)k^2 + (-32N^2+38)k + 6N^3 - 11N^2 - 9N + 14]}{(N-2)(N-1)N(N+1)(n+2)}. \quad (7)$$

B. Coefficient Quantization Effects

For fixed-point presentation of filter coefficients, two's complement presentation is used, and in the direct fixed-point implementations of the filter coefficients (7), magnitude truncation is applied. Location of the fixed point is set so that maximum accuracy is achieved given the range of the filter coefficient values. In our calculations, 'infinite precision' means the computational precision of Matlab.

Quantization effects can be seen in section IV, in Figs. 3 through 5. It is clearly seen that as the coefficients are truncated, the prediction and/or differentiation properties of the filters are lost. As also seen comparing Figs. 4 and 5, the differentiation property is generally more robust to the coefficient quantization than the prediction property which can be lost already with the coefficient word length of 16 bits. Differentiation of an input signal consisting of polynomial signal components of 0th, 1st and 2nd degree, is set by zero magnitude response at zero frequency along with a ramp-shaped response within a desired differentiation band which are also given by the constraints (3)-(5), Figs. 1 a), 3 a), 4 a) and 5 a). The one-step-ahead prediction property can be seen as the negative unity group delay within a desired prediction band, c.f. Figs. 1 b), 3 b), 4 b) and 5 b).

It is worth noting that the coefficients (7) for the filter length $N=3$ are still exact if quantized to eight bits. The frequency response and group delay of this filter are shown in Figs. 1 a) and b), respectively. If the responses shown in Fig. 1 are adequate for the fixed-point application at hand, this is the filter to apply, otherwise the filter functioning has to be verified, or the method described in this paper is to be used to obtain exactly correctly functioning fixed-point coefficients filters.

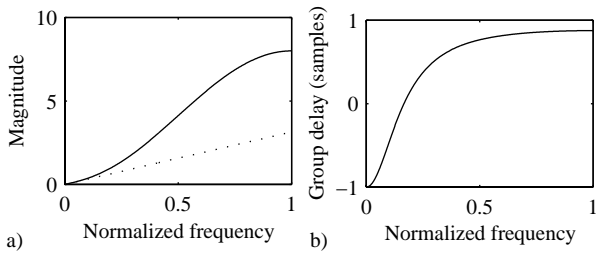


FIG. 1. a) Frequency response (solid) and b) group delay of the second degree one-step-ahead predictive differentiator of length $N = 3$. In a) also the ideal differentiator frequency response is shown (dotted).

III. POLYNOMIAL-PREDICTIVE FIR DIFFERENTIATOR DESIGN BY LINEAR DIOPHANTINE EQUATION BASED SOLUTION

The optimization problem that has to be solved can be reformulated as an integer programming problem. Suppose that all the coefficients of the filter, $h(k)$, are multiplied by 2^n . Then the optimization task can be defined as follows:

Input: Function (2)

$$NG = F(h(0), h(1), \dots, h(N-1)) = \sum_{k=0}^{N-1} h^2(k) \quad (8)$$

with variables, i.e., filter coefficients, $h(k)$. The constraint conditions (3)-(6) for the variables can be formulated as

$$g_0 = \sum_{k=0}^{N-1} h(k) = 0, \quad (9)$$

$$g_1 = \sum_{k=0}^{N-1} (N-k-1)h(k) = 2^n, \quad (10)$$

$$g_2 = \sum_{k=0}^{N-1} (N-k-1)^2 h(k) = 2^{n+1}(N-1-p), \quad (11)$$

⋮

$$g_M = \sum_{k=0}^{N-1} (N-k-1)^M h(k) = 2^n M(N-1-p)^{M-1}, \quad (12)$$

respectively.

Output: An integer vector, $\mathbf{h} = (h^*(0), h^*(1), \dots, h^*(N-1))$ that satisfies *exactly* the constraints (9)-(12) given above and minimizes NG (8).

The solution we offer is based on the following considerations:

1. The task in hand is a quadratic integer programming problem, which is well-known to be an NP-complete problem [7-9]; therefore it is unrealistic to find the best solution in a reasonable amount of time, especially for long filters. This state of affairs is in sharp

contrast to the quadratic real programming problem [8], which is solvable in polynomial time.

2. Without restricting the variables to be integers, we have a closed form solution of the problem, which is given by (7) for the case $M = 2$ and $p = 1$. Although the values computed by this formula are not integers, this expression gives us a very good initial approximation.
3. To make sure that the conditions (9)-(12) are met exactly, one has to solve the above system in integers. This problem has been a subject of very deep investigations in number theory and the theory of Diophantine equations. By eliminating the variables, one can reduce the problem to a single linear equation of the form

$$A_1 x_1 + A_2 x_2 + \dots + A_l x_l = B \quad (13)$$

where A_1, A_2, \dots, A_l and B are integers.

The solutions of (13) are usually obtained by multidimensional continued fraction algorithms [10,11], and the reader can find a large variety of methods aimed at solving this class of Diophantine equations. Here our approach is based on Clausen-Fortenbacher algorithm. The reasons why we chose this particular technique are: firstly, the algorithm succeeds in finding very fast the solutions of (9)-(12), from which the optimal one, that is, the one that would minimize the noise gain NG (8), can be quickly found; secondly, the program provided in [12] can be easily generalized to more than 16 variables (the largest case analyzed by Clausen and Fortenbacher); thirdly, we have a good initial approximation that significantly speeds up the algorithm.

In the following Tables 1 through 5, we show some results for the filter length $N = 16$ with different dynamic ranges for the variables $h(k)$ along with the real number form (infinite precision) solutions of (7) and the best integer solution obtained. It is worth noting that the most straightforward approximation of the infinite precision coefficients with the closest integers never produced a solution of the system of the Diophantine equations (9)-(11). This demonstrates the necessity of special techniques aimed at solving the integer optimization problem. Here the search for the ideal quantization has been conducted within ± 2 from the normally quantized coefficients presented in integer form. This search band is illustrated in Fig. 2 for the filter length $N = 16$, with coefficient precisions of 8, Fig 2 a) and 16 bits, Fig 2 b), which correspond to Tables 1 and 5, respectively. Table 6 lists the numbers of solutions that exactly satisfy the constraints (9)-(11) for coefficient precisions 6, 8, 10, 12, 14, and 16 bits for the filter lengths $N = 8$ and $N = 16$. To find the optimum solution, it is necessary of search all of the solutions and to select the one which minimizes noise gain (8).

For the filter length $N = 8$, this takes less than one second on a 166 MHz Pentium processor using exhaustive search programmed with C language. For many applications, also the first-found solution could most probably be adequate, which should be checked by comparing the noise against the noise gain of the corresponding infinite precision filter, and the application at hand.

Table 1 shows that with the filter length $N = 16$ and the coefficient precision of 8 bits, for five out of sixteen coefficients one has to approximate the real coefficient with an integer that is not the closest one. The 10, 12, 14 and 16-bit filter coefficients are shown in Tables 2, 3, 4 and 5, respectively.

TABLE 1. The infinite precision presentation (real number form) of the digital filter coefficients computed by (7) for the filter length $N = 16$ and their best integer approximations that guarantee the exact solution of (9)-(11) while also minimizing the noise gain (2) with the coefficient precision of 8 bits.

Coefficients	Real number form	Best integer appr.	Coefficients	Real number form	Best integer appr.
256 $h(0)$	32.313725	32	256 $h(8)$	-16.376471	-17*
256 $h(1)$	20.894118	20*	256 $h(9)$	-15.605602	-17**
256 $h(2)$	10.998319	12**	256 $h(10)$	-13.310924	-13
256 $h(3)$	2.626331	3	256 $h(11)$	-9.492437	-9
256 $h(4)$	-4.221849	-4	256 $h(12)$	-4.150140	-4
256 $h(5)$	-9.546218	-9*	256 $h(13)$	2.715966	3
256 $h(6)$	-13.346779	-13	256 $h(14)$	11.105882	11
256 $h(7)$	-15.623529	-16	256 $h(15)$	21.019608	21

* The best integer approximation is not the integer closest to the real (infinite precision) coefficient value.

** The best integer approximation is not an integer on either side of the real (infinite precision) coefficient value.

TABLE 2. The infinite precision presentation (real number form) of the digital filter coefficients computed by (7) for the filter length $N = 16$ and their best integer approximations that guarantee the exact solution of (9)-(11) while also minimizing the noise gain (2) with the coefficient precision of 10 bits.

Coefficients	Real number form	Best integer appr.	Coefficients	Real number form	Best integer appr.
1024 $h(0)$	129.254902	128*	1024 $h(8)$	-65.505882	-65*
1024 $h(1)$	83.576471	84	1024 $h(9)$	-62.422409	-64**
1024 $h(2)$	43.993277	45**	1024 $h(10)$	-53.243679	-53
1024 $h(3)$	10.505322	11	1024 $h(11)$	-37.969748	-38
1024 $h(4)$	-16.887395	-17	1024 $h(12)$	-16.600560	-17
1024 $h(5)$	-38.184874	-38	1024 $h(13)$	10.863866	11
1024 $h(6)$	-53.387115	-54*	1024 $h(14)$	44.423529	45*
1024 $h(7)$	-62.494118	-62	1024 $h(15)$	84.078431	84

* The best integer approximation is not the integer closest to the real (infinite precision) coefficient value.

** The best integer approximation is not an integer on either side of the real (infinite precision) coefficient value.

TABLE 3. The infinite precision presentation (real number form) of the digital filter coefficients computed by (7) for the filter length $N = 16$ and their best integer approximations that guarantee the exact solution of (9)-(11) while also minimizing the noise gain (2) with the coefficient precision of 12 bits.

Coefficients	Real number form	Best integer appr.	Coefficients	Real number form	Best integer appr.
4096 $h(0)$	517.019608	516**	4096 $h(8)$	-262.023529	-262
4096 $h(1)$	334.305882	335*	4096 $h(9)$	-249.689636	-251**
4096 $h(2)$	175.973109	176	4096 $h(10)$	-212.974790	-213
4096 $h(3)$	42.021289	42	4096 $h(11)$	-151.878992	-152
4096 $h(4)$	-67.549580	-67*	4096 $h(12)$	-66.402241	-67*
4096 $h(5)$	-152.739496	-153	4096 $h(13)$	43.455462	43
4096 $h(6)$	-213.548459	-213*	4096 $h(14)$	177.694118	178
4096 $h(7)$	-249.976471	-249*	4096 $h(15)$	336.313725	338**

* The best integer approximation is not the integer closest to the real (infinite precision) coefficient value.

** The best integer approximation is not an integer on either side of the real (infinite precision) coefficient value.

TABLE 4. The infinite precision presentation (real number form) of the digital filter coefficients computed by (7) for the filter length $N = 16$ and their best integer approximations that guarantee the exact solution of (9)-(11) while also minimizing the noise gain (2) with the coefficient precision of 14 bits.

Coefficients	Real number form	Best integer appr.	Coefficients	Real number form	Best integer appr.
16384 $h(0)$	2068.078431	2068	16384 $h(8)$	-1048.094118	-1048
16384 $h(1)$	1337.223529	1337	16384 $h(9)$	-998.758543	-1000**
16384 $h(2)$	703.892437	704	16384 $h(10)$	-851.899160	-852
16384 $h(3)$	168.085154	168	16384 $h(11)$	-607.515966	-607*
16384 $h(4)$	-270.198319	-270	16384 $h(12)$	-265.608964	-266
16384 $h(5)$	-610.957983	-611	16384 $h(13)$	173.821849	174
16384 $h(6)$	-854.193838	-854	16384 $h(14)$	710.776471	710*
16384 $h(7)$	-999.905882	-999*	16384 $h(15)$	1345.254902	1346*

* The best integer approximation is not the integer closest to the real (infinite precision) coefficient value.

** The best integer approximation is not an integer on either side of the real (infinite precision) coefficient value.

TABLE 5. The infinite precision presentation (real number form) of the digital filter coefficients computed by (7) for the filter length $N = 16$ and their best integer approximations that guarantee the exact solution of (9)-(11) while also minimizing the noise gain (2) with the coefficient precision of 16 bits.

Coefficients	Real number form	Best integer appr.	Coefficients	Real number form	Best integer appr.
65536 $h(0)$	8272.313725	8272	65536 $h(8)$	-4192.376471	-4193*
65536 $h(1)$	5348.894118	5348*	65536 $h(9)$	-3995.034174	-3997**
65536 $h(2)$	2815.569418	2816	65536 $h(10)$	-3407.596639	-3408
65536 $h(3)$	672.340616	673*	65536 $h(11)$	-2430.063866	-2430
65536 $h(4)$	-1080.793277	-1080*	65536 $h(12)$	-1062.435854	-1062
65536 $h(5)$	-2443.831933	-2444	65536 $h(13)$	695.287395	696*
65536 $h(6)$	-3416.775350	-3416*	65536 $h(14)$	2843.105882	2843
65536 $h(7)$	-3999.623529	-3999*	65536 $h(15)$	5381.019608	5381

* The best integer approximation is not the integer closest to the real (infinite precision) coefficient value.

** The best integer approximation is not an integer on either side of the real (infinite precision) coefficient value.

TABLE 6. The number of ideally quantized solutions N_{ios} that exactly satisfy constraints (3)-(5) for the filter lengths $N = 8$ and $N = 16$ as a function of coefficient precision (6, 8, 10, 12, 14 and 16 bits).

Coeff. prec. (bits)	6	8	10	12	14	16
$N_{ios}, N = 8$	21	14	14	21	14	14
$N_{ios}, N = 16$	56326	53633	58791	55027	58287	57341

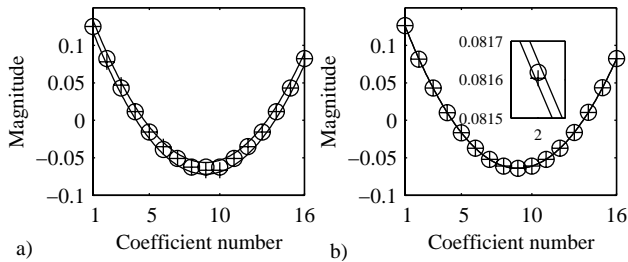


FIG. 2. Ideal quantization search bands (between solid lines) for the filters of length $N = 16$ with the coefficient precisions of a) 8 bits and b) 16 bits. Circles 'o' denote the quantized, and pluses '+' the ideally quantized coefficients.

IV. CHARACTERISTICS OF THE QUANTIZED AND IDEALLY QUANTIZED COEFFICIENT FILTERS

In this section, frequency response and group delay properties of the infinite precision, quantized-coefficient, and ideally quantized-coefficient filters are illustrated. In Fig. 3, second degree polynomial-predictive FIR differentiator of length $N = 8$ is shown for the three cases mentioned above with the coefficient precision of 8 bits. Same plots are given in Figs. 4 and 5 for the filter length $N = 16$ and coefficient precisions of 8 and 16 bits, respectively. If the ideal quantization yields several filters that exactly satisfy the constraints (3)-(5), the one which minimizes the noise gain (2) is shown in Figs. 3 through 5.

From the Figs. 3, 4 and 5 it can be seen that the filters with quantized coefficients are practically useless considering their prediction and/or differentiation properties whereas the filters with the ideally quantized coefficients behave like the filters with infinite precision coefficients, as they should, since they satisfy the constraints (3)-(5) exactly. The filters shown in Fig. 4 correspond to the coefficients listed in Table 1, and Fig. 5 to those listed in Table 5.

In Fig. 6, the noise gains of the second degree polynomial-predictive FIR differentiator of length $N = 16$ with coefficients optimally quantized to 8, 10, 12, 14 and 16 bits are plotted along with the noise gain of the corresponding infinite precision filter. From Fig. 6 it is seen that as the coefficient precision increases, the noise gain loss nicely diminishes. Should the noise gain of a quantized coefficient filter be less than that of the infinite coefficient counterpart, the constraints (3)-(6) were not exactly preserved since the infinite precision filters by definition satisfy (3)-(6) and minimize the noise gain (2).

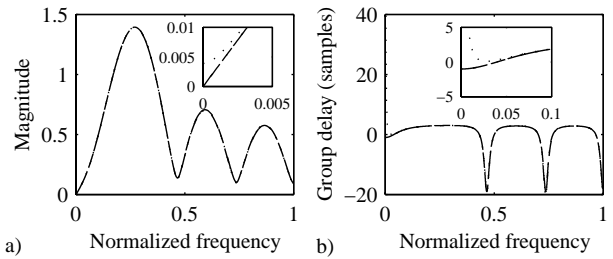


FIG. 3. Magnitude responses a) and group delays b) of the infinite precision (dashed), quantized coefficient (dotted) and optimally quantized coefficient (dash-dot) second degree polynomial-predictive FIR differentiators of length $N = 8$ with the coefficient accuracy of 8 bits.

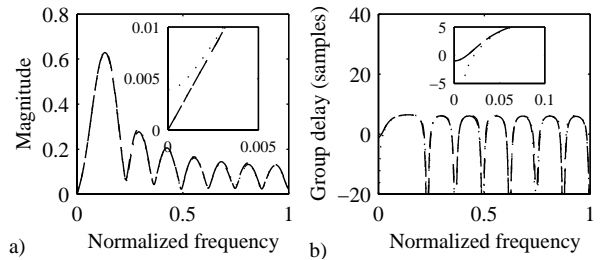


FIG. 4. Magnitude responses a) and group delays b) of the infinite precision (dashed), quantized coefficient (dotted), optimally quantized coefficient (dash-dot) second degree polynomial-predictive FIR differentiators of length $N = 16$ with the coefficient accuracy of 8 bits.

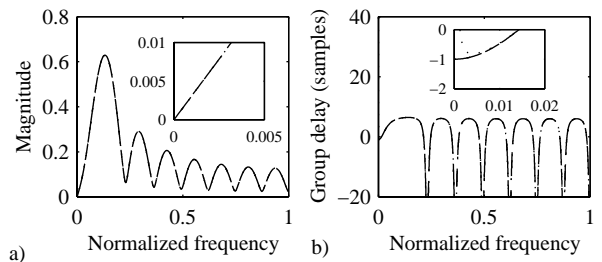


FIG. 5. Magnitude responses a) and group delays b) of the infinite precision (dashed), quantized coefficient (dotted), optimally quantized coefficient (dash-dot) second degree polynomial-predictive FIR differentiators of length $N = 16$ with the coefficient accuracy of 16 bits.

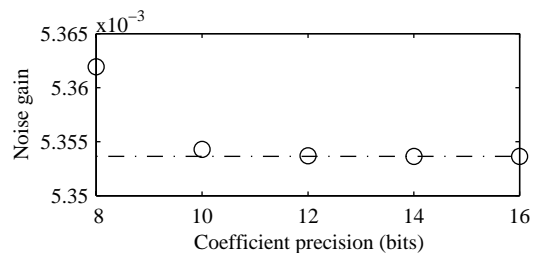


FIG. 6. Noise gains of the ideally quantized coefficient second degree polynomial-predictive FIR differentiator of length $N = 16$ as a function of the coefficient precision in bits (circles) along with the noise gain of the same filter with infinite precision coefficients.

V. FURTHER RESEARCH TOPICS

There is ample room for improvements of the method proposed. First of all, the idea is applicable to FIR and IIR filters if one aims at removing the influence of the round-off errors. Secondly, the problem gets more complicated for long filters, as it should. The NP-completeness of the optimization problem [6-9,13-16] involved, forces us to use heuristic solutions; the new one, proposed in this pa-

per, is based on number-theoretic methods. Summarizing, we could point out four directions for future research that seem particularly important:

1. The extension of the method for very long filters (length several hundred coefficients) would require more powerful technique for solution of the main Diophantine equation than Clausen-Fortenbacher algorithm. The precise comparison between their method and multidimensional continued fraction algorithm [10,11], to the best of our knowledge, has not yet been done.
2. It would be particularly beneficial to design a multiplierless architecture for this class of filters. In the language of number systems this means that one should look for a solution which consists of fairly small number of nonzero digits in a canonic signed-digit binary number system. That is coefficients $h(k)$ should be of the form $\pm 2^a \pm 2^b$ or $\pm 2^a \pm 2^b \pm 2^c$, where a , b and c are integers. Such a restriction normally results in extending the length of the filter and increasing the influence of round-off errors. Whether or not the method proposed can remove the later, that is, the influence of the round-off errors, remains to be seen.
3. Recursive extension [2] will be applied to the quantized-coefficient polynomial-predictive FIR differentiators designed in this paper to extend the concept of coefficient quantization by integer programming to IIR design.
4. For being able to utilize our integer programming solution for coefficient quantization of other filter types, the filter specifications should be expressed in a form similar to the constraints for the polynomial-predictive FIR differentiators, i.e., in the form of a set of linear equations with filter coefficients as variables. This concept will be further explored.

VI. CONCLUSIONS

A new technique for perfect digital filter coefficient quantization has been proposed. Our method uses number-theoretic tools. As it is demonstrated in the paper, the given filter design constraints giving the filters their polynomial signal prediction and differentiation properties, can be exactly satisfied, and thus, the influence of the round-off errors is eliminated. For the second degree polynomial-predictive FIR differentiators used in this paper, the conditions can be exactly satisfied with even as low as 6-bit coefficient precision, with still some degrees of freedom left to minimize the noise gain of the designed fixed-point coefficients filter. The proposed integer programming method for fixed-point filter design is well suited at least for all filter design tasks in which the design criteria can be formulated in a form of linear constraints on filter coefficients, like those for the polynomial-predictive FIR differentiators.

REFERENCES

- [1] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. New York, NY: Macmillan Publishing Company, 1992.
- [2] S. Väiliviita and S. J. Ovaska, "Delayless recursive differentiator with efficient noise attenuation for control instrumentation," *Signal Processing*, 69, Sept. 1998, pp. 267–280.
- [3] P. T. Harju, T. I. Laakso, and S. J. Ovaska, "Applying IIR predictors on Rayleigh fading signal," *Signal Processing*, 48, Jan. 1996, pp. 91–96.
- [4] P. Heinonen and Y. Neuvo, "FIR-median hybrid filters with predictive FIR substructures," *IEEE Trans. Acoustics, Speech, and Signal Processing*, 36, June 1988, pp. 892–899.
- [5] O. Vainio, M. Renfors, and T. Saramäki, "Recursive implementation of FIR differentiators with optimum noise attenuation," *IEEE Trans. Instrumentation and Measurement*, 46, Oct. 1997, pp. 1202–1207.
- [6] D. Bertsekas, *Constrained Optimization and Lagrange Multipliers Methods*. New York, NY: Academic Press, 1982.
- [7] E. L. Johnson, "Integer programming, facets, subadditivity and duality for group and semigroup problems," *CBMS-NSF Regional Conference Series in Applied Mathematics*, SIAM, 1990.
- [8] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley, 1986.
- [9] C. R. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice Hall, 1982.
- [10] G. Szekeres, "Multidimensional continued fraction algorithms", *Ann. Univ. Sci. Budapest Eotvos, Sect. Math.*, 13, 1970, pp. 113–140.
- [11] H. R. P. Ferguson and R. W. Forcade, "Generalization of the Euclidean algorithm for real numbers to all dimensions higher than two," *Bull. AMS*, Nov. 1979.
- [12] M. Clausen and A. Fortenbacher, "Efficient solution of linear Diophantine equations," *Journal of Symbolic Computation*, 8, July/Aug. 1989, pp. 201–216.
- [13] I. H. Dinwoodie, "Stochastic simulation on integer constraint set," *SIAM Journal on Optimization*, 9(1), Dec. 1998, pp. 53–61.
- [14] P. Diaconis and B. Strumfeld, *Algebraic algorithms for sampling from conditional distributions*. Technical Report No. 6, Department of Statistics, Stanford University, 1993.
- [15] J. More and G. Toraldo, "Algorithms for bound constrained quadratic programming problems," *Numerical Mathematics*, 55, 1989, pp. 377–400.
- [16] S. J. Wright, "Finite-precision effects on the local convergence of interior-point algorithms for nonlinear programming," *Preprint ANL/MCS P705-0198*, Mathematics and Computer Science Division, Argonne National Laboratories, Argonne, IL, 1998.

ACKNOWLEDGMENT

The work of V. S. Dimitrov has been funded by the Technology Development Centre of Finland, Nokia Corporation, Sonera Ltd., Finland, and Helsinki Telephone Company Ltd., Finland. The work of J. M. A. Tanskanen has been partially funded by the parties mentioned above, and his work has also been supported by Jenny ja Antti Wihuri Foundation, Finland, Walter Ahlström Foundation, Finland, and by The Finnish Society of Electronics Engineers.