Successive Approximation FIR Filter Design

Alessandro J. S. Dutra
Programa de Engenharia
Elétrica – COPPE
Univ. Federal do Rio de Janeiro
a.dutra@ieee.org

Lisandro Lovisolo
Departamento de Eletrônica
e Telecomunicações
Univ. do Estado do Rio de Janeiro
lisandro@uerj.br

Eduardo A. B. da Silva and Paulo S. R. Diniz
Programa de Engenharia Elétrica – COPPE
Dept. de Eletrônica – POLI
Univ. Federal do Rio de Janeiro
eduardo@lps.ufrj.br, diniz@lps.ufrj.br

***Abstract*—A new method for the design of finite impulse response (FIR) filters whose discrete coefficient space is the power-of-two space is presented. We employ a vector successive approximation technique successfully used in data compression algorithms to produce a design method with a very low computational complexity that generates filters with implementation cost as low as those obtained by other, much more complex, optimization methods.**

I. Introduction

The design of FIR filters whose coefficients belong to a constrained class of numbers (as opposed to using the full-precision floating-point numbers) has long been a subject of great interest. In particular, it is desired to design filters whose frequency responses match a set of previously defined specifications using word lengths as short as possible.

A most useful outcome of such design philosophy is that of FIR filters whose coefficients are made of sums of powers-of-two (POTs). This type of design proves to be most beneficial when dealing with hardware implementations, in which the use of general multipliers is very costly [1]. In this case, the use of POTs leads to what is commonly referred to as a "multiplier-less implementation, as each coefficient can be generated from a combination of shifts and adds/subtracts.

In [2], [3], an integer programming approach is used to produce solutions to minimization problems using discrete constraints in the filter coefficient values for both the weighted minimax and weighted least-squares objective functions. In either case, the filter designs obtained are optimal for a given word length, but the design processes are extremely complex in computational terms. More recently, a method using linear programming to optimize the coefficients directly in the sub-expression space was proposed [4] that yields a considerable reduction in the number of required sums to attain a given specification.

A different optimization approach is used in [5], in which an infinite-precision prototype filter is designed to exceed the project specifications, thereby producing a margin of error for the coefficient quantization. A non-linear optimization is then performed to determine the final filter representation.

In this paper, we propose a design method also based on the approximation of a prototype filter, designed to match or exceed the project specifications, as needed. We employ a vector successive approximation technique initially developed for data compression applications, and define the representation dictionary in such a way that the design computational complexity is kept to a minimum.

The paper is organized as follows: Section II presents a brief review of vector representation techniques, including the *matching pursuits generalized bitplane* (MPGBP) algorithm [6], which will serve as the basis for our proposed approximation method. The successive approximation design of FIR filters is then described in Section III, with examples and performance comparison against other existing methods being discussed in Section IV. Our conclusions and future directions are then detailed in Section V.

1. Start with $\boldsymbol{w} = \boldsymbol{x}$, $m = 1$
2. Repeat until a stop criterion is met:
   (a) Choose $r_m \in \{1, \ldots, q\}$ such that
   $$\boldsymbol{w} \cdot \boldsymbol{v}_{r_m} = \max_{1 \le j \le q} \{\boldsymbol{w} \cdot \boldsymbol{v}_j\}$$
   (b) Choose
   $$k_m = \left\lceil \frac{\log(\boldsymbol{w} \cdot \boldsymbol{v}_{r_m})}{\log(\alpha)} \right\rceil$$
   where $\lceil \boldsymbol{y} \rceil$ is the smallest integer larger than or equal to $\boldsymbol{y}$.
   (c) Replace $\boldsymbol{w}$ with $\boldsymbol{w} - \alpha^{k_m} \boldsymbol{v}_{r_m}$
   (d) Increment m.
3. Stop.

Figure 1. The MPGBP Algorithm [6]

II. Successive Approximation of Vectors

Several methods for the approximation of vectors have been successfully proposed for use in the image and video coding fields. The existence of particular constraining factors in those applications, such as the ever present trade-off on permissible encoding rate/reproduction quality, have led us to conjecture that the use of those algorithms for the task of approximating the coefficients of FIR filters, where the main constraint lies on whether or not the filter's frequency response satisfies the specifications, should enjoy similar success.

One such approximation method is called *matching pursuits with generalized bitplanes* decomposition – the MPGBP algorithm [6]. It is a greedy algorithm in which a signal $\boldsymbol{x}$ is decomposed as

$$\boldsymbol{x} = \sum_{j=1}^{\infty} \alpha^j \sum_{l=1}^{L_j} \boldsymbol{g}_{i_{j,l}} \tag{1}$$

where $\boldsymbol{g}_n \in \mathcal{D} = \{\pm\boldsymbol{g}_1, \pm\boldsymbol{g}_2, \ldots, \pm\boldsymbol{g}_M\}$ and $0 < \alpha < 1$.

The MPGBP algorithm produces, after P passes, an approximation for $\boldsymbol{x}$ given by

$$\boldsymbol{x}^{(P)} = \sum_{m=1}^{P} \alpha^{k_m} \boldsymbol{v}_{r_m}, \tag{2}$$

where $\boldsymbol{v}_{r_m} \in \mathcal{D}$. The algorithm for performing such a decomposition is presented in Fig. 1.

Assuming that $\|\boldsymbol{x}\| < 1$, the error in the approximation is bounded by

$$\|\boldsymbol{x} - \boldsymbol{x}^{(P)}\|^2 \le \beta^P$$

where

$$\beta = \sqrt{1 - (2\alpha - \alpha^2)\cos^2(\Theta(\mathcal{D}))}$$

978-1-4244-9472-9/11/$26.00 ©2011 IEEE

and

$$\Theta(\mathcal{D}) = \cos^{-1}\left\{\min_{\boldsymbol{x}\in\mathbb{R}^n}\left\{\max_{\boldsymbol{g}_i\in\mathcal{D}}\left(\frac{\boldsymbol{x}\cdot\boldsymbol{g}_i}{\|\boldsymbol{x}\|\,\|\boldsymbol{g}_i\|}\right)\right\}\right\},$$

where $\boldsymbol{x}\cdot\boldsymbol{g}_i$ denotes the inner product of the vectors $\boldsymbol{x}$ and $\boldsymbol{g}_i$.

For $0 < \alpha < 1$ and $\Theta(\mathcal{D}) < \pi/2$, we have that $\beta < 1$ and therefore the representation error converges to zero as the number of approximation passes grows. Also, the smaller the value of $\Theta(\mathcal{D})$, the faster the convergence is.

We shall now propose a modified version of the MPGBP algorithm to be used in the design of FIR filters.

III. FIR Filter Approximation

Given a set of frequency response specifications $\{\omega_p, \omega_s, \delta_p, \delta_s\}$, we use the Parks-McClellan algorithm [7] to design a order N prototype filter $h(n)$, which will be represented as

$$\boldsymbol{h} = [h(0)\, h(1)\, \ldots\, h(N)]. \tag{3}$$

In order to approximate the coefficients of $\boldsymbol{h}$ using a modified MPGBP algorithm, we first define the approximation dictionary

$$\mathcal{D} = \{\pm\boldsymbol{g}_1, \pm\boldsymbol{g}_2, \ldots, \pm\boldsymbol{g}_M\}$$

in which the codewords $\boldsymbol{g}_i \in \mathbb{R}^{N+1}$ and each codeword has P components with magnitude equal to 1 and $(N+1-P)$ components equal to 0, i.e., they are permutations of the form

$$\boldsymbol{g}_i = (\pm 1^P, 0^{(N+1-P)})$$

The vector $\boldsymbol{w}$ is initialized with $\boldsymbol{h}$ and is then approximated using a version of the MPGBP algorithm (Fig. 1) in which the evaluation of k_m in step 2.b is now given by

$$k_m = \text{round}\left(\frac{\log(\boldsymbol{w}\cdot\boldsymbol{v}_{r_m})}{\log(\alpha)}\right). \tag{4}$$

The search for $\boldsymbol{v}_{r_m}$, which was an exhaustive one in the original method, can now be accomplished in a much faster way by taking into account the existing codeword structure. With that in mind, it suffices to sort the absolute values of the components of the current representation of $\boldsymbol{w}$ in decreasing order of magnitude and store the indices of the P largest ones. The approximation codeword $\boldsymbol{v}_{r_m}$ is then obtained by setting those P components whose indices were stored to ± 1 according to the sign of $\boldsymbol{w}$.

Amongst the possible stop criteria for use with the design method one may cite the approximation error and the number of adds/subtracts used in the approximation.

The search for the best approximation with the proposed method includes testing for prototypes of different orders N – starting with the minimum required by the Parks-McClellan algorithm to meet the filter specifications – and for different values of P, the number of non-zero codeword components, and look for the one that yields the lowest implementation complexity, in terms of number of adders used by the filter.

It is also worth noticing that the number of POTs used in the approximation of distinct components is not fixed, a constraint found in other optimization methods. For each design, a global output multiplier is determined to ensure that the average passband gain is kept at 0 dB.

IV. Experimental Results

In this section we present the performance of the proposed algorithm by comparing it against that of other methods in the literature.

Table I
LOWEST COMPLEXITY IMPLEMENTATION PARAMETERS OBTAINED WITH THE SA-DESIGN METHOD FOR SPECIFICATIONS OF EXAMPLE 1.

P	N	POTs	Adders
5	33	11	33
4	29	11	30
3	33	11	31
2	29	11	30
1	33	12	33

Table II
COEFFICIENTS FOR THE SA-DESIGNED FIR FILTER OF EXAMPLE 1. ($P = 4$, $N = 29, h(29-n) = h(n)$ for $n = 0, \ldots, 14$)

n	$h(n)$	POTs used in approximation
0	-0.00097656250	-2^{-10}
1	-0.00341796875	$-2^{-8} + 2^{-11}$
2	0.00000000000	0
3	0.00683593750	$2^{-7} - 2^{-10}$
4	0.00585937500	$2^{-7} - 2^{-9}$
5	-0.00830078125	$-2^{-7} - 2^{-11}$
6	-0.01757812500	$-2^{-6} - 2^{-9}$
7	0.00000000000	0
8	0.03173828125	$2^{-5} + 2^{-11}$
9	0.02539062500	$2^{-5} - 2^{-7} - 2^{-9}$
10	-0.03515625000	$-2^{-5} - 2^{-8}$
11	-0.07714843750	$-2^{-4} - 2^{-6} + 2^{-10}$
12	0.00048828125	2^{-11}
13	0.19824218750	$2^{-2} - 2^{-4} - 2^{-7} + 2^{-8} - 2^{-10}$
14	0.37304687500	$2^{-1} - 2^{-3} + 2^{-9}$

A. Example 1

The specifications are those of [5], Ex. 1. A low-pass filter design is desired in which the passband and stopband edges are located at $\boldsymbol{w}_p = 0.3\pi$ and $\boldsymbol{w}_s = 0.5\pi$ and the maximum allowed passband and stopband ripples are $\delta_p = \delta_s = 0.00316$, corresponding to a 50 dB stopband attenuation.

For the prototype design, the minimum order leading to a design attaining the specifications is determined to be $N_{\min} = 25$. We employ the design method for values of $P \in \{1, 2, 3, 4, 5\}$ and order up to $N = 35$, searching for the minimum complexity (in number of adders) implementation.

The best results presented in [5] indicate that the specifications can be attained by using a $N = 29$ order filter, with 37 adders. The number of adders, however, may be lowered to 30 by using a common sub-expression elimination post-design method. As can be seen from Table I, these numbers are matched with the design based on the dictionary with $P = 4$. The amplitude response for this filter, along with its passband details, can be seen in Fig. 2. The impulse response of this particular filter is described in Table II.

B. Example 2

The specifications are those of [5], Ex. 2. A low-pass filter design is desired in which the passband and stopband edges are located at $\boldsymbol{w}_p = 0.3\pi$ and $\boldsymbol{w}_s = 0.5\pi$ and the maximum allowed passband and stopband ripples are $\delta_p = \delta_s = 0.001$, corresponding to a 60 dB stopband attenuation.

For the prototype design, the minimum order leading to a design attaining the specifications is determined to be $N_{\min} = 32$. We employ the design method for values of $P \in \{1, 2, 3, 4, 5, 6\}$ and order up to $N = 42$, searching for the minimum complexity (in number of adders) implementation.

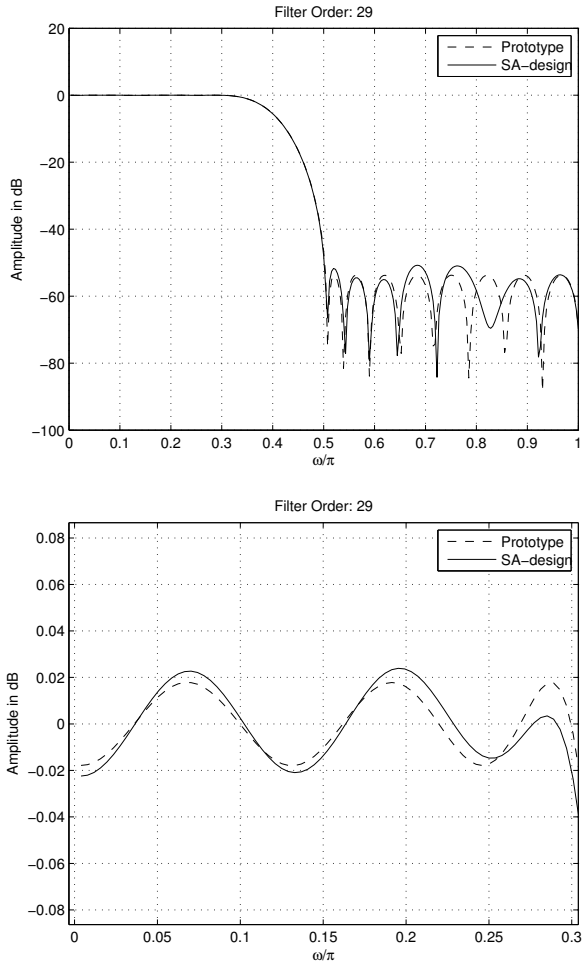

Figure 2. Magnitude responses for the prototype and the SA-designed FIR filters ($P = 4$) in Example 1. (Passband detail shown separately)

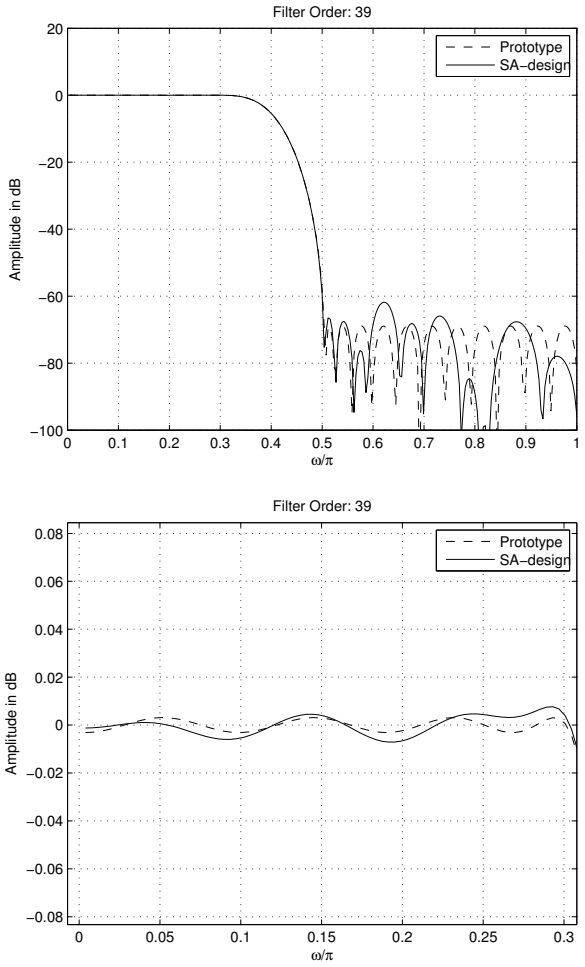

Figure 3. Magnitude responses for the prototype and the SA-designed FIR filters ($P = 3$) in Example 2. (Passband detail shown separately)

Table III
LOWEST COMPLEXITY IMPLEMENTATION PARAMETERS OBTAINED WITH THE SA-DESIGN METHOD FOR SPECIFICATIONS OF EXAMPLE 2.

P	N	POTs	Adders
6	37	13	49
5	37	13	50
4	38	13	50
3	39	13	48
2	37	13	48
1	37	13	48

In this more restrictive case, the best results presented in [5] indicate that the specifications can be attained by using a $N = 37$ order filter, with 48 adders. Once again the number of adders may be lowered to 39 by using a common sub-expression elimination post-design method. As can be seen from Table III, the numbers obtained with the SA-design based on the dictionary with $P = 4$ are comparable to those of [5] without the post-processing step. The amplitude response for this filter, along with its passband details, can be seen in Fig. 3.

C. Example 3

As a third example, we propose the design of a band-pass filter whose specifications are as follows:

- passband edges: $\boldsymbol{w}_{p1} = 0.3\pi$, $\boldsymbol{w}_{p2} = 0.6\pi$
- stopband edges: $\boldsymbol{w}_{s1} = 0.15\pi$, $\boldsymbol{w}_{s2} = 0.8\pi$

and the maximum allowed passband and stopband ripples are $\delta_p = \delta_{s1} = \delta_{s2} = 0.001$, corresponding to a 60 dB stopband attenuation.

For the prototype design, the minimum order leading to a design attaining the specifications is determined to be $N_{\min} = 43$. We employ the design method for values of $P \in \{1, 2, 3, 4, 5, 6, 7\}$ and order up to $N = 63$, searching for the minimum complexity (in number of adders) implementation.

In this case, the prototype design method determined that the minimum order so that the specifications may be attained is $N = 43$. The lowest-complexity implementation is obtained by setting $P = 1$ and designing an $N = 55$ order filter. For that design, whose coefficients are presented in Table V, the number of required adders is 67, with the use of 12 different power-of-two factors. The amplitude response for this filter, along with its passband details, can be seen in Fig. 4.

Table IV
LOWEST COMPLEXITY IMPLEMENTATION PARAMETERS OBTAINED WITH THE SA-DESIGN METHOD FOR SPECIFICATIONS OF EXAMPLE 3.

P	N	POTs	Adders
7	61	12	72
6	61	12	74
5	62	11	73
4	53	12	76
3	62	11	73
2	55	13	74
1	55	12	67

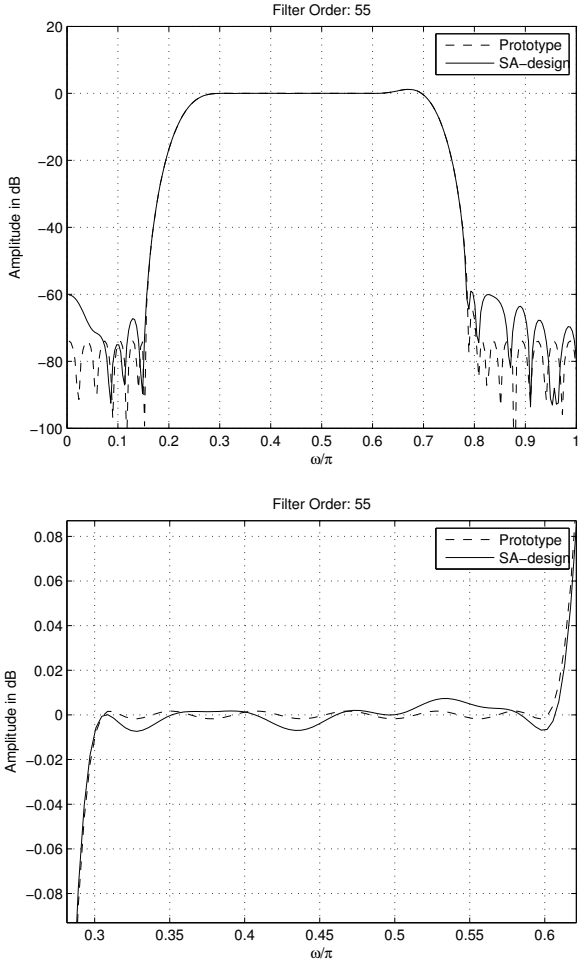

Figure 4. Magnitude responses for the prototype and the SA-designed FIR filters ($P = 1$) in Example 3. (Passband detail shown separately)

V. CONCLUSIONS

We have presented a novel method for the design of FIR digital filters based on the successive approximation of vectors in which the components are approximated by sums of powers-of-two. The proposed method produces approximations of the impulse response of a given filter with very low implementation complexity, without resorting to costly optimization techniques.

The obtained results show the potential of the proposed method, and indicate there may still be room left for improvement by, for instance, employing adaptive dictionaries in the approximation step.

Table V
COEFFICIENTS FOR THE SA-DESIGNED FIR FILTER OF EXAMPLE 3.
($P = 1$, $N = 55, h(55 - n) = h(n)$ for $n = 0, \ldots, 27$)

n	$h(n)$	POTs used in approximation
0	-0.000732421875	$-2^{-10} + 2^{-12}$
1	0.000488281250	2^{-11}
2	0.000732421875	$2^{-10} - 2^{-12}$
3	-0.000732421875	$-2^{-10} + 2^{-12}$
4	0.002929687500	$2^{-8} - 2^{-10}$
5	0.000488281250	2^{-11}
6	-0.004150390625	$-2^{-8} - 2^{-12}$
7	0.002441406250	$2^{-9} + 2^{-11}$
8	-0.005371093750	$-2^{-8} - 2^{-9} + 2^{-11}$
9	-0.007324218750	$-2^{-7} + 2^{-11}$
10	0.010742187500	$2^{-7} + 2^{-8} - 2^{-10}$
11	-0.001953125000	-2^{-9}
12	0.003906250000	2^{-8}
13	0.023437500000	$2^{-5} - 2^{-7}$
14	-0.014404296875	$-2^{-6} + 2^{-10} - 2^{-12}$
15	-0.006347656250	$-2^{-7} + 2^{-9} + 2^{-11}$
16	0.003662109375	$2^{-8} - 2^{-12}$
17	-0.047363281250	$-2^{-4} + 2^{-6} + 2^{-11}$
18	0.003906250000	2^{-8}
19	0.024414062500	$2^{-5} - 2^{-7} - 2^{-10}$
20	-0.014648437500	$-2^{-6} + 2^{-10}$
21	0.081542968750	$2^{-4} + 2^{-6} + 2^{-8} - 2^{-11}$
22	0.035156250000	$2^{-5} + 2^{-8}$
23	-0.052490234375	$-2^{-4} + 2^{-7} - 2^{-9} - 2^{-12}$
24	0.038574218750	$2^{-5} + 2^{-7} - 2^{-11}$
25	-0.189453125000	$-2^{-2} + 2^{-4} + 2^{-9}$
26	-0.247070312500	$-2^{-1} + 2^{-8} + 2^{-10}$
27	0.360107421875	$2^{-1} - 2^{-3} + 2^{-6} - 2^{-10} - 2^{-12}$

REFERENCES

[1] J. Yli-Kaakinen and T. Saramaki, "A systematic algorithm for the design of lattice wave digital filters with short-coefficient wordlength," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 54, pp. 1838–1851, aug. 2007.

[2] Y. C. Lim, S. Parker, and A. Constantinides, "Finite word length FIR filter design using integer programming over a discrete coefficient space," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 30, pp. 661–664, aug 1982.

[3] Y. C. Lim and S. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 31, pp. 583–591, jun 1983.

[4] Y. J. Yu and Y. C. Lim, "Design of linear phase FIR filters in subexpression space using mixed integer linear programming," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 54, pp. 2330–2338, oct. 2007.

[5] J. Yli-Kaakinen and T. Saramaki, "A systematic algorithm for the design of multiplierless FIR filters," in *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 2, pp. 185–188 vol. 2, 6-9 2001.

[6] R. Caetano, E. A. B. da Silva, and A. G. Ciancio, "Video coding using greedy decompositions on generalised bit-planes," *Electronics Letters*, vol. 38, pp. 507–508, 2002.

[7] P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, *Digital Signal Processing: System Analysis and Design*. Cambridge University Press, 2nd ed., 2010.