Optimal Design of Multiplierless Second-Order Infinite Impulse Response Digital Filters

Rémi Garcia, Anastasia Volkova, Martin Kumm, Alexandre Goldsztejn, Jonas Kühle

Optimal Design of Multiplierless Second-Order Infinite Impulse Response Digital Filters

Rémi Garcia*, Alexandre Goldsztejn†, Jonas Kühle‡, Martin Kumm‡, Anastasia Volkova* *Universiteé de Nantes, CNRS, LS2N F-44000 Nantes, France
†CNRS, Universiteé de Nantes, LS2N F-44000 Nantes, France
‡Faculty of Applied Computer Science, Fulda University of Applied Sciences, Germany

Abstract

In this work we solve the problem of design of optimal multiplierless second-order Infinite Impulse Response (IIR) filters. Given a frequency specification, we design a filter with fixed-point coefficients yielding minimal number of adders when evaluated with Direct Form. The coefficient design and quantization steps, typically conducted independently, are now gathered into one global optimization problem, modeled through mixed integer linear programming and efficiently solved using generic solvers. Moreover, we guarantee the frequency-domain stability of the filter, increasing the confidence in the result. The generated filters are implemented and synthesized using our new operators within FloPoCo IP core generator for Field Programmable Gate Arrays (FPGA). With respect to state-of-the-art two-step filter design methods our one-step optimal design approach achieves, on average, 42% reduction in number of lookup tables (LUT) and 21% improvement in delay.

Index Terms

Digital filters, IIR, optimal design, multiplierless hardware, ILP

I. Introduction

Digital filters are essential components of modern technology, from medical equipment to scientific instruments. Filter design is a core topic in digital signal processing and control, and efficient filter implementation in software and hardware has received a significant research interest for half a century. Infinite impulse response (IIR) filters is class of widely-used recursive linear time-invariant filters. IIR filters can be relatively easily designed in software, but hardware implementation is essential for embedded systems, where performance/power constraints are critical. Some application domains, such as 5G/6G backbones and autonomous vehicles, rely on reconfigurable hardware using Field Programmable Gate Arrays (FPGA).

A designer of floating-point software filters rely on many powerful and easy-to-use approaches. Hardware design is confronted to numerous challenges, such as fixed-point arithmetic, which is more efficient in hardware. What is a constraint in software (e.g. using 32-bit coefficients) becomes a degree of freedom in hardware (e.g. finding the smallest fixed-point format).

Classic fixed-point filter design and implementation flow follows three separate steps:

- filter design consists in finding real (in practice, double precision) filter coefficients, adhering to the given frequency specification. IIR filters are defined by coefficients of a rational transfer function, for which a stability criteria must be also satisfied. In general, a large amount of different filter coefficient sets can realize a given frequency specification;
- quantization converts the coefficients to a fixed-point representation in such a way that they still respect the given frequency response;
- implementation consists in generating, using quantized coefficients, a valid hardware description. This step exposes a high number of parameters, e.g. the type of multipliers used.

The combination of filter design and quantization steps has been studied extensively since 1960s, for IIR filters it can even be considered solved, but with respect to a probabilistic error model, not guaranteeing numerical safety. A large body of work exists for the implementation step. Hardware filters involve multiplications with constants, for which optimization techniques have been extensively explored. In the *shift-and-add*-based methods, constant multiplications are replaced by additions, subtractions and bit shifts. The associated optimization problem is known as the multiple constant multiplication (MCM) problem, for which heuristics [7], [8] as well as optimal approaches exist [4], [5], [9]. Another constant multiplication method, especially relevant for FPGAs, is based on precomputed tables and is called Ken Chapman multiplier (KCM), after its inventor [10]. It has also been successfully applied to digital filtering [2], [5], [11] and in [1] an approach for optimization of combined quantization and implementation steps has been proposed. Both KCM and shift-and-add reduce arithmetic resources by sharing intermediate results. However, this reduction strongly depends on the coefficient values, which are typically fixed in the previous FD and Q steps. This is a first major drawback of the state-of-the-art, since the obtained implementations are optimized only for one filter instance, or a small sub-set of the overall design space.

In this work, we address the question of modeling the filter design, quantization and implementation steps as one global optimization problem. This is a very difficult non-linear combinatorial problem, hence heuristics are often applied. In contrast, our ambition is to attempt optimal solution thanks to the advancement of mixed-integer linear programming (MILP) solvers. Hence, the objective is to model the combined problem using linear constraints over integer variables, and rely on

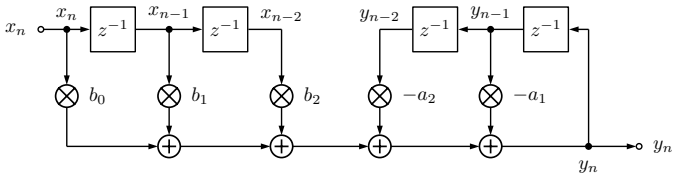

Fig. 1: Direct form realization of a second-order IIR filter.

existing solvers such as CPLEX and Gurobi. This problem is very difficult in general case, it has been partially solved for FIR filters in [6] but has not yet been attempted for IIR filters by previous work.

We restrict the design space to the second-order IIR filters, which are basic building blocks for higher-order filters [12]–[15] that are often decomposed into second-order cascades. We consider the multiplierless implementation using optimal MCM solution [4], [5], which is modeled as an MILP problem and can be easily incorporated for en overall global optimization problem. The constraints of the global optimization approach include the frequency-domain specification constraints that capture the combined filter design and quantization steps. They will define a (large) set of fixed-point coefficients achieving the given frequency specification. We linearize the filter design constraints exploiting an efficient frequency discretization. Additionally, for IIR filters, a stability constraint must be incorporated in the overall process. Classically, the stability criteria [16] is expressed w.r.t. locations of filter's poles, which is clearly non-linear but can be simplified in case of second-order filters. Moreover, we show in the present paper that the stability constraints divide the design space into symmetrical zones and that optimization time improvement can be achieved when these symmetries are broken. Finally, various cost functions can be considered. We target a metric based on the number of adders in the multiplierless implementation, which is high-level but representative enough [4]. An important consequence of the global optimization approach is that if a stable implementation with a given fixed-point wordlength is not possible, the proof is generated quickly. This allows the filter designer a faster implementation process. Our approach is implemented as an open-source Julia package[1], which takes the worldnegth and frequency specifications as inputs, and generates the fixed-point coefficients for the multiplierless implementation with the minimal number of adders.

Finally, we implement a second-order hardware code generator within the FloPoCo hardware IP core generator for FPGAs. Our implementations are numerically stable in the presence of rounding errors, guaranteeing that all output digits are correct. Extensive benchmarks show that our one-step design process achieves significant performance improvement: 42% improvement in number of LUTs and 21% in improvement in delay, on average.

Section II presents the proposed ILP model formulation, where we first detail the linear frequency constraints and provide several search space reductions, followed by a high-level description of the overall formulation, including the MCM formulations fromn [4]. In Section III, we show that our model combined with well-known solvers leads to the optimal design of multiple filters in a reasonable amount of time, and present a hardware code generator.

II. ILP FORMULATION FOR SECOND-ORDER IIR FILTER DESIGN

A. Problem definition

Direct Form II structure of second order digital filters is represented in Figure 1. Its output is computed as follows:

$$y_n = \sum_{k=0}^{2} b_k u_{n-k} - \sum_{k=1}^{2} a_k y_{n-k}, \tag{1}$$

where $a_k, b_k \in \mathbb{R}$ are the filter coefficients. The corresponding transfer function is

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \tag{2}$$

with $z \in \mathbb{C}$. Designing a filter means finding the coefficients a_k and b_k such that the filter is stable and given frequency constraints hold. These constraints can be expressed as bounds on the filter frequency response:

$$\underline{\beta}(\omega) \leq \left|H\left(e^{j\omega}\right)\right| \leq \overline{\beta}(\omega)\,, \quad \forall \omega \in [0, \pi]\,, \tag{3}$$

where $\underline{\beta}$ and $\overline{\beta}$ are the given frequency dependent lower and upper bounds. They typically encode constant bounds valid inside frequency intervals.

As we will search for coefficients a_k and b_k in a FxP representation, we introduce $a'_k, b'_k \in \mathbb{Z}$ their integer counterparts that are linked with these real numbers by

$$a_k = 2^{-w+1}\, 2^{g_a}\, a'_k, \qquad a'_k \in \left[-2^{w-1}+1, 2^{w-1}\right], \tag{4}$$

$$b_k = 2^{-w+1}\, 2^{g_b}\, b'_k, \qquad b'_k \in \left[-2^{w-1}+1, 2^{w-1}\right], \tag{5}$$

[1]available at

where w is the wordlength (including sign bit), g_a and g_b are the maximum of the position of the most significant bit (MSB) of the FxP representation (double braces represent intervals of integers). The MSB positions g_a and g_b together with the wordlength w define the FxP representation of the coefficients. It is typical to use the same wordlength but to allow different values for the MSB of the coefficients in the numerator and denominator of the transfert function. The optimal design process proposed here is cast into the integer programming framework by searching for the integer counterparts a'_k and b'_k of the FxP representation. The values of w, g_a and g_b are not known a priori but are critical since they actually fix the precision and the range of FxP representation used for the coefficient: on the one hand, a large wordlength increases the precision of the FxP representation but also lead to heavier hardware implementations. On the other hand, large values of the MSB enlarge the domain of representable coefficients but also lower the precision of the FxP representation. Resolving this typical tradeoff requires having tight bounds on the coefficients, which allow fixing the MSB of the representations. These bounds will have to be computed as a preprocess before actually searching for values in the chosen FxP representation. The wordlength is typically fixed independently to reach a user prescribed precision. The wordlength is supposed given here.

As it has been successfully done for FIR filters [17], we unify the filter coefficients design and their quantization by searching directly for coefficients satisfying the frequency constraints in their FxP representation: the overall objective is to find FxP representations of filter coefficients such that filter specifications are met and the hardware implementation is optimal with respect to the number of adders. Extending this approach from FIR to IIR filters requires several technical contributions: the frequency constraints (3) on the module of the rational transfert function (2) involve highly nonlinear operations, which are linearized in Section II-B. A stability criteria is added to the ILP model, which allows enforcing stability of the coefficient in FxP format. Bounds on the coefficients a_k and b_k are furthermore deduced in Section II-C. Search space reductions are presented in Section II-D. The connection of this filter specification part of the model to two standard MCM ILP models for the adder graphs formulation is presented in Section II-E.

B. Formulating linear frequency constraints

In order to use obtain an ILP model, we have to solve two issues related to the frequency constraints (3). First of all, (3) actually enforces one constraint for each $\omega \in [0, \pi]$. There are therefore infinitely many such constraints to be satisfied. The constraint (3) is called semi-infinite (see [18] for a survey on this class of constraints). The standard discretization approach to handling semi-infinite constraints consists in discretizing $[0, \pi]$ into a finite set of frequencies Ω_d, leading to a finite number of constraints. This discretization can be dynamically updated, but we consider here a fixed discretization dedicated to frequencies [17] and a *a posteriori* verification of the semi-infinite frequency constraint [19]. In the rare cases where the verification fails, the verification procedure proposed in [19] outputs a faulty frequency that can be added to the discretized set of frequencies for a new trial. Continuous variables generically lead to active constraints, which require some theory and accurate algorithms with some adaptive discretization to allow discovering active frequencies for optimal coefficients (see [18], [20] for details). Integer variables generically don't lead to active constraints, therefore allowing a finite number additional faulty frequency addition to the model before obtaining coefficients verifying the constraints associated to all frequencies. This easy reduction to a finite problem of the semi-infinite constraint is a positive side-effect of searching directly FxP representations of the coefficients.

After this discretization of the semi-infinite constraint, a finite number of nonlinear constraints need to be handled. For a fixed $\omega \in \Omega_d$, the constraint (3) includes a complex absolute value, which involves a square-root of squared terms, and a fraction. By incorporating (2) into (3), then multiplying with the denominator $\left|A\left(e^{j\omega}\right)\right|$, and finally squaring the result we obtain a constraint equivalent to (3):

$$\left|A\left(e^{j\omega}\right)\right|^2 \underline{\beta}(\omega)^2 \leq \left|B\left(e^{j\omega}\right)\right|^2 \leq \left|A\left(e^{j\omega}\right)\right|^2 \overline{\beta}(\omega)^2 , \tag{6}$$

where,

$$\left|B\left(e^{j\omega}\right)\right|^2 = \sum_{k=0}^{2}\sum_{l=0}^{2} b_k b_l \cos((k-l)\,\omega), \tag{7}$$

$$\left|A\left(e^{j\omega}\right)\right|^2 = \sum_{k=0}^{2}\sum_{l=0}^{2} a_k a_l \cos((k-l)\,\omega), \tag{8}$$

with $a_0 = 1$.

Yet, in (7) and (8), the filter coefficients are still involved in bilinear terms $b_k b_l$ and $a_k a_l$. Billionnet *et al.* [21] proposed a method that allows linearizing products of positive integers assuming that bounds on these numbers are known: consider the product $z = xy$ where $x, y, z \in \mathbb{N}$, with $x \leq \overline{x}$ and $y \leq \overline{y}$. The linearization [21] basically consists in first rewriting of one of the positive integers into its binary representation:

$$x = \sum_{k=0}^{\lceil \log_2 \overline{x} \rceil + 1} 2^k\, t_{x,k}, \tag{9}$$

where $t_{x,k}$ are binary auxiliary variables. This constraint ensures that the bits $t_{x,k}$ encode the value of x. Then, the product $z = xy$ becomes a sum of products between binary variables $t_{x,k}$ and a positive integer y. Finally, such a binary by integer product is common and its well known linearization involves indicator or big M constraints [22], [23].

Here however, x and y correspond to filter coefficients that have no sign restriction. We extend the linearization exposed above to signed integers by adding the auxiliary variables $x^+, y^+ \in \mathbb{N}$ and $x^{\text{sg}}, y^{\text{sg}} \in \{0, 1\}$, and link them by following constraints:

$$x^+ = |x|\,, \qquad y^+ = |y|\,, \tag{10}$$

$$x^{\text{sg}} = \text{sign}(x)\,, \qquad y^{\text{sg}} = \text{sign}(y)\,, \tag{11}$$

$$z^+ = x^+ y^+, \tag{12}$$

where the linearization of the absolute values (10) and the sign constraints (11) involve indicator or big M constraints [24], [25], and where $z^+ = x^+y^+$ is the positive case we already presented. Finally, $z = \pm z^+$ and the sign is determined by the values of x^{sg} and y^{sg} directly in the model. This whole linearization relies on the fact that bounds on x and y are known. This is addressed in the next subsection.

C. Stability and bounds on filter coefficients

Necessary and sufficient stability conditions for second-order filters are well-known [16, Section 16.8]:

$$-2 < a_1 < 2, \tag{13}$$

$$|a_1| - 1 < a_2 < 1. \tag{14}$$

As explained before, the absolute value is standardly linearized using indicator or big M constraints, thus (13)-(14) actually fit in an ILP model. From these constraints it is straightforward to derive bounds on a_k: $a_1 \in]-2, 2[$ and $a_2 \in]-1, 1[$. These bounds are independent of the frequency specification of the filter and yield an optimal value $g_a = 1$ for the MSB of the coefficients a_k.

Bounds on b_k, however, cannot be obtained independently of the filter specifications. Using the bounds (13)-(14) and the fact the cosine in (8) belongs to $[-1, 1]$, we deduce that $\left|A\left(e^{j\omega}\right)\right|^2 \leq 16$. This bound together with the frequency specification constraints (6) lead to the following constraint that needs to be satisfied by the coefficients b_k:

$$\left|B\left(e^{j\omega}\right)\right|^2 \leq 16\overline{\beta}(\omega)^2. \tag{15}$$

As can be seen from (7), $\left|B\left(e^{j\omega}\right)\right|^2$ is a quadratic form b^TQb with respect to the variables b_k. Its characteristic matrix Q, whose entries are $Q_{kl} = \cos((k-l)\omega)$, is symmetric and its spectrum is $\{0, 1 - \cos(2\omega), 2 + \cos(2\omega)\}$. Its eigenvalues being non-negative, the inequality constraint (15) is convex. As a consequence, some lower and upper bounds on the coefficients b_k can be computed by solving the convex quadratic problems consisting in minimizing or maximizing b_k subject to the convex quadratic constraints (15) for $\forall \omega \in \Omega_{d'}$ where $\Omega_{d'}$ is a discretization of Ω. The global minimum and maximum of these problems are required to be used as valid lower and upper bounds, such global extrema being easily computed by local solvers since the quadratic constraints are all convex and the cost is linear. In particular, common MILP solvers can solve this kind of nonlinear problem. In addition to allowing the linearization of frequency constraints, these bounds permit to decide a valid value g_b for the MSB of the coefficients b_k.

These bounds do not take into account the specificity of filter we are designing and use worst case and simplified models. Thus, it is possible to compute tighter these bounds in order to reduce the search space. This is explained in the next section.

D. Search space reduction

1) Linearized specifications projections: The bounds for a_k and b_k computed in the previous section allows implementing an ILP model by fixing an FxP format and linearizing the bilinear terms involved in the frequency specifications. This first incomplete ILP model does not include the geometry of the adder graph defined in Section II-E. Yet, in order to speed up the solving of the complete ILP model, similarly to [17], we tighten the bounds on the coefficients by solving these incomplete ILPs that are simpler than the complete ILP model in the sense that it does not include the geometry of the adder graph defined in Section II-E: they consist in minimizing or maximizing b'_k subject to the stability constraints and the linearized frequency constraints. These ILPs are obviously more difficult to solve than the continuous convex quadratic problems used in the previous section to obtain crude bounds, but still much easier to solve than the final comlpete model. As it has been shown in [17] in the context of the design of FIR filters, solving these simpler ILPs to obtain tighter bounds for the solving of the final ILP is worthwhile. However, this benefit has to be assessed experimentally in the context of second-order IIR filters.

2) Symmetry breaking: The model contains some symmetries that can be broken in order to reduce the search space: the first symmetry consists in simultaneously changing the sign of the values taken by b_0, b_1 and b_2, the second symmetry consists in exchanging the values taken by b_0 and b_2. These two symmetries leave the constraints on the coefficients b_k unchanged, as can be easily seen on the explicit expression

$$b_0^2 + b_1^2 + b_2^2 + 2b_0b_1\cos(\omega) + 2b_1b_2\cos(\omega) + 2b_0b_2\cos(2\omega), \tag{16}$$

of (6) involved in the constraint (15): indeed, (16) is insensitive to changing all coefficients sign simultaneously, and to exchanging b_0 and b_2. Both symmetries have no incidence one the MCM problem defined in Section II-E, the existence of symmetric adder graphs being obvious for opposed or exchanged coefficients.

As a consequence, from an arbitrary solution with

$$b_0 = b_0^*,\ b_1 = b_1^* \text{ and } b_2 = b_2^*, \tag{17}$$

we can build three new solutions by simply applying these symmetries:

$$b_0 = -b_0^*,\ b_1 = -b_1^* \text{ and } b_2 = -b_2^*, \tag{18}$$

$$b_0 = b_2^*,\quad b_1 = b_1^* \quad \text{and } b_2 = b_0^*, \tag{19}$$

$$b_0 = -b_2^*,\ b_1 = -b_1^* \text{ and } b_2 = -b_0^*. \tag{20}$$

The fourth solution (20) is obtained by applying the two symmetries consecutively, in any order. Note that some of these four symmetric solutions (17)-(20) may be equal in some special cases, *e. g.*, when $b_0 = b_1 = b_2 = 0$.

Breaking these symmetries means finding additional constraints, called symmetry breaking constraints (SBCs), that remove some symmetric solutions but keep at least one of them [26]. If necessary, symmetric solutions that have not been calculated due to the SBCs can be built afterward. In the best case, SBCs keep only one solution among all symmetric solutions, in that case the SBCs are called total. Therefore, we expect here to reduce the size of the search space of a ration 4 since solutions come within symmetry classes containing 4 symmetric solutions. Finding SBCs for general symmetry groups is difficult, *e. g.*, SBCs generation for symmetries consisting only of variable permutations rely on modern group theoretic algorithms [27]. In our case, the symmetry group is generated by one variable permutation and one central symmetry. Up to our knowledge, SBCs involving both variable permutations and central symmetries have not yet been investigated and no SBC defined for them.

The search space for b_0, b_1 and b_2, which is $\mathbb{R}^3$, can be divided into four areas:

$$\Sigma_1 = \left\{(b_0, b_1, b_2) \in \mathbb{R}^3 \mid b_0 \geq |b_2|\right\}, \tag{21}$$

$$\Sigma_2 = \left\{(b_0, b_1, b_2) \in \mathbb{R}^3 \mid -b_0 \geq |b_2|\right\}, \tag{22}$$

$$\Sigma_3 = \left\{(b_0, b_1, b_2) \in \mathbb{R}^3 \mid b_2 \geq |b_0|\right\}, \tag{23}$$

$$\Sigma_4 = \left\{(b_0, b_1, b_2) \in \mathbb{R}^3 \mid -b_2 \geq |b_0|\right\}. \tag{24}$$

One can verify that: a solution lying inside Σ_2 moves to Σ_1 applying the sign symmetry; a solution lying inside Σ_3 moves to Σ_1 applying the the exchange symmetry; a solution lying inside Σ_4 moves to Σ_1 applying both symmetries consecutively. As a consequence, one can restrict the search to Σ_1 and reconstruct all solutions using symmetries. This restriction to Σ_1 is achieved by adding the SBC

$$b_0 \geq |b_2| \tag{25}$$

to the model, which restricts the search to Σ_1. Restricting to another Σ_i would be lead to another SBC, with an equivalent improvement of the resolution process. Yet, on the frontier $\Sigma_k \cap \Sigma_l$ between areas two equivalent solutions might be kept despite the SBC, which is a typical situation that is counterproductive to resolve because the great majority of the search space lies in the interior the sets Σ_i and not on their boundaries ($\Sigma_k \cap \Sigma_l$ have zero volume in $\mathbb{R}^3$).

E. MCM for direct-form IIR filters

We stated that the last step of the filter design is the hardware implementation. This implementation can be optimized solving the MCM problem and we propose to incorporate this solving into the MILP model we are building. The idea is to link together an existing ILP model for MCM [28], [29] and the model used to search filter coefficients by adding a set of constraints that we will refer to as *glue*.

Before adding glue constraints, we note that two MCMs are involved in CCDE (1) since two sets of constants are multiplied with two variables. The objective function of each MCM-ILP model is meant to minimize the number of adders. We can sum the objective functions. To minimize the number of structural adders we can subtract to the objective function binary variables, one for each filter coefficient. Having these binary variables set to one if the associated coefficient is different to zero ensure that the total number of adders is minimized. As it can be seen in Figure 1, a filter coefficient fixed to zero implies removing an adder. After adding two MCM models, their output variables are linked to a'_k and b'_k.

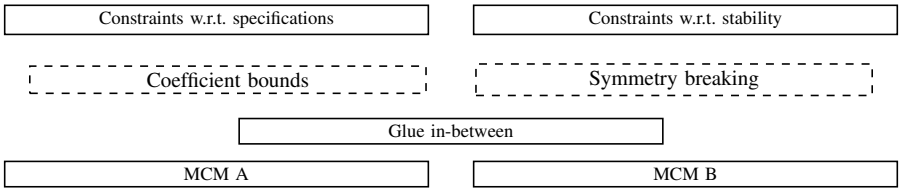

Fig. 2: High-level overview of second-order IIR filter ILP model. Search space reductions by the mean of coefficient bounds precomputation or symmetry breaking constraint are optional.

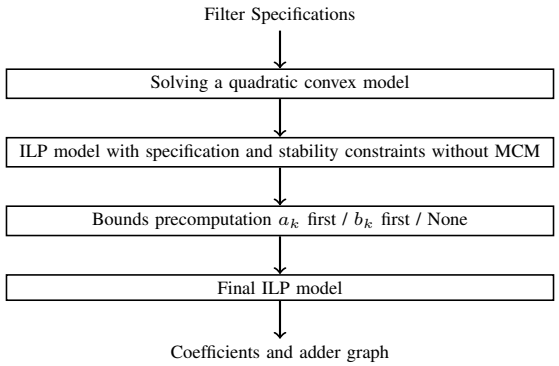

Fig. 3: Computation process that leads to an ILP model for the design of second-order IIR filter.

The model for MCM we use intrinsically bounds the AD. This limits the potential valid solutions of the theoretical MCM problem. However, in recursive filters it is a reasonable objective to bound the AD so the feedback stays controlled. Thus, in practice, this model would be preferred over a theoretical MCM.

F. Wrapping up

We propose a high-level representation of our model in Figure 2. Two blocks of constraints, represented with dashed lines, are optional. Activating the coefficient bounds block corresponds to the precomputation of a_k and b_k bounds. In any case, constraints with respect to specifications need a first bound on all the coefficients, these bounds are obtained by solving a quadratic convex model. The whole process that leads to the final model is represented by the Figure 3 in which we recognize the quadratic convex problem solving step and the optional reduction of the coefficient range. Doing or not the reduction of the coefficient range does not impact the optimality of the solution but might speed up the resolution time which we investigate in Section III.

Using a wide set of tools from optimization we obtained a linear model. To obtain this linear model we have solved a nonlinear model. Yet, we showed that the problem we face is quadratic convex, thus, easy to solve, even using linear specialized solvers. Finally, we have a model that, from filter specifications, permits us to directly obtain a hardware implementation. The traditional three step process can be tackled as a single step which ensures that we do not lose information at each step. In the following we test our linear model and verify that, using common MILP solvers, we are able to optimally solve practical second-order IIR filter designs.

III. Experimental results

Although the design of second-order IIR filters is an important part of the design of larger order filters, usually benchmarks are not intended to test their design efficiency but directly focus on large order filters. First, we propose a few sets of filter specifications of increasing difficulty specially for testing the design of second-order IIR filters. Second, we report our results and we compare them with other design methods in order to show that an optimal solution has a tangible impact on the hardware.

A. Set of benchmarks

We use three sets of lowpass filters with increasing difficulty. For each set, one of the parameters, ht magnitude specification δ or the width of the passband or stopband, is variable. These filters are given in Table I. To these sets we add another lowpass filter, lp_4, with a stopband in $[0.0, 0.5\pi]$, a passband in $[0.9\pi, \pi]$ and a δ of 0.1. Finally, we add a last filter, hp0, from the literature [30]–[33] that we approximate from its frequency response showing the versatility of our model.

TABLE I: Sets of lowpass filters used for the IIR experiments. First the set with decreasing δ, next the sets with increasing passband and stopband.

Name	lp1_k	lp2_k	lp3_k
k	$\{0, 1, \ldots, 9\}$	$\{0, 1, \ldots, 7\}$	$\{0, 1, \ldots, 7\}$
passband/π	$[0, 0.3]$	$[0, 0.3 + 0.05k]$	$[0, 0.3]$
stopband/π	$[0.7, 1]$	$[0.7, 1]$	$[0.7 - 0.05k, 1]$
δ	$0.1 - 0.01k$	0.1	0.1

It is important to note that for a given value of k and a given coefficients' wordlength, our benchmark can lead to an instance with filter specifications that are not realizable with a second-order IIR filter. Hence, the MILP solver will return "infeasible" and our method proved that no method could produce a second-order IIR filter that satisfies those specifications with a given coefficients' wordlength.

B. Solving optimization problem

For the design of second-order IIR filters we use the modified MCM model from [29]. Instances were solved using CPLEX on i7-10810U processor. The solving process has been tested with multiple parameter settings and based on unreported results we chose to use symmetry breaking constraint and reduction of the a_k coefficient range first and b_k coefficient range second with a time limit on this presolving. Note that even when the process stops because of that time limit, we still obtain a suboptimal coefficient range bound which permits to reduce the search space.

We report optimization results in Table II and show that our method can find, in a reasonable time, second-order filters for many different specifications using a reduced number of adders. In order to compare the hardware efficiency of our method to the state of the art, we used MATLAB to obtain double precision coefficients for our benchmarks. Due to the design methods, MATLAB was not able to return second-order IIR filter coefficients for a few specifications. By loosing up these, we obtained second-order IIR filter coefficients for every filter specification that we did not proved infeasible. Then we truncated to the nearest these coefficients for some wordlengths and compared hardware designs with our method. For that comparison we used KCM (2-step KCM) on the coefficients obtained with MATLAB and, on truncated coefficients, we used a generic method with (2-step Generic) and without (2-step Generic NoDSP) digital signal processors (DSPs). We also compare to a multiplierless method using MCM (2-step MCM) on truncated coefficients.

In the following, the error, ε, corresponds to the distance between the target specification and the obtained frequency response. As expected our method fits the specifications for every filter. In contrast, even double precision filter coefficients obtained from MATLAB lead to a small error. Depending on the rounding, truncated coefficients can produce a filter that has a big error to no error. Theses possibilities have been illustrated in Figure **??** for lp1_4. We can see on that figure that double precision coefficients produce a frequency response that is close to the passband and stopband while our coefficients lead to a frequency response that has a bigger margin from the bands.

Finally, hp0 has been defined with different sets of coefficients [31], [32]. To approximate it we used its poles and zeros from [31], plus a gain, as it seems to be the most precise coefficients available. Obtaining coefficients that permit to approximate this filter was not straightforward, we proved that for many sets of parameters, the coefficient wordlength and a approximation percentage, there is no second order filter that satisfy the specifications. The poles of this filter are close to the unit circle and especially to 1 thus it is not surprising that it is difficult to approximate it for frequencies around zero. However, by adding more slack in the approximation these frequencies, we were able to find a set of coefficients that leads to a frequency response close to the original. We represent both the original and our approximation in Figure **??**. Due to its pole position, the original filter is sensitive to rounding errors [33]. On the other hand, our coefficients ($b_0 = -101.5$, $b_1 = -51$, $b_2 = -50.75$, $a_1 = -0.46875$, $a_2 = -0.484375$) lead to a transfer function with poles that are further from the unit circle.

C. Hardware implementation and comparison

We report most results in Table II. The results were obtained using FloPoCo [34] for the VHDL generation and Vivado for their simulation. From Table II it can be noted that increasing the difficulty of specifications or the wordlength does not necessarily lead to more look-up tables (LUTs) or a bigger delay for our method. On the other hand, in Figure 4 it can be seen that, for the methods that rely on truncated coefficients, the number of LUTs and DSPs tends to increase with the wordlength while it is not clear that it permits to the error to decrease. Our method seems robust as the number of LUTs does not increase much with the wordlength and even decreases when a better solution in term of the number of adders is found.

Finally, in Figure 5 we note that other methods led to twice the number of LUTs in many cases or about the same number of LUTs plus DSPs. For wordlengths from 4 to 10 our method permits an improvement on the average number of LUTs of 42% compare to other methods and an improvement of 21% for the delay. This improvement was possible perfectly fitting target specifications while other methods led to some error.

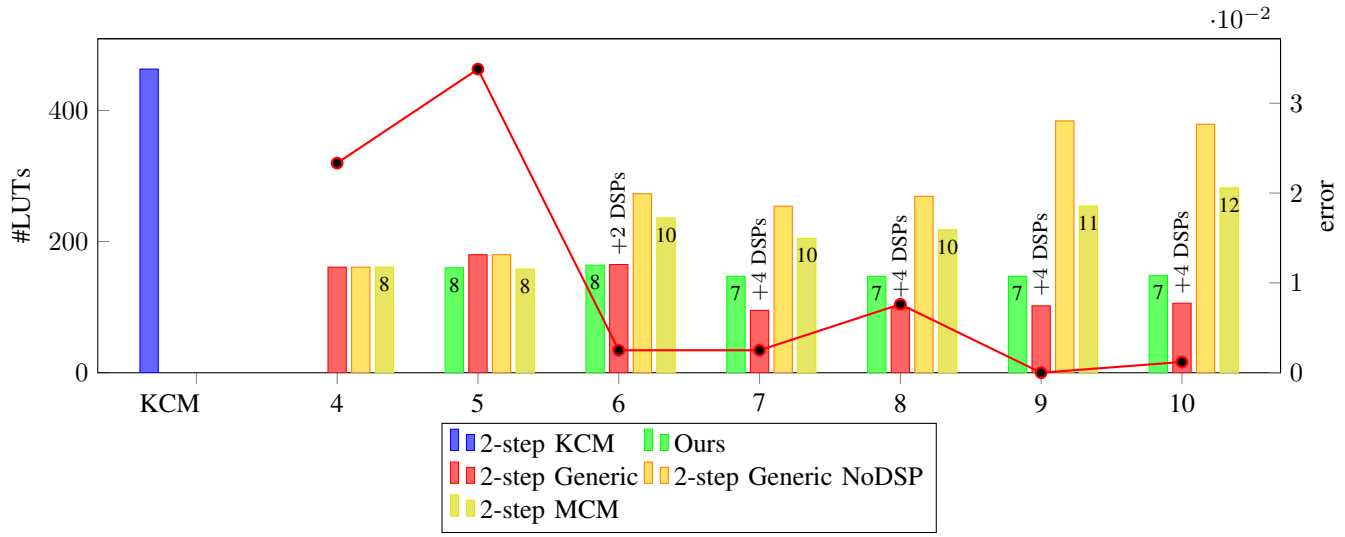

Fig. 4: Implementations for $\mathrm{lp1}_4$ benchmark with coefficient wordlengths from 4 to 10 bits. The 2-step generic and 2-step MCM approaches are based on quantized coefficients, hence yield a frequency response error (right axis).

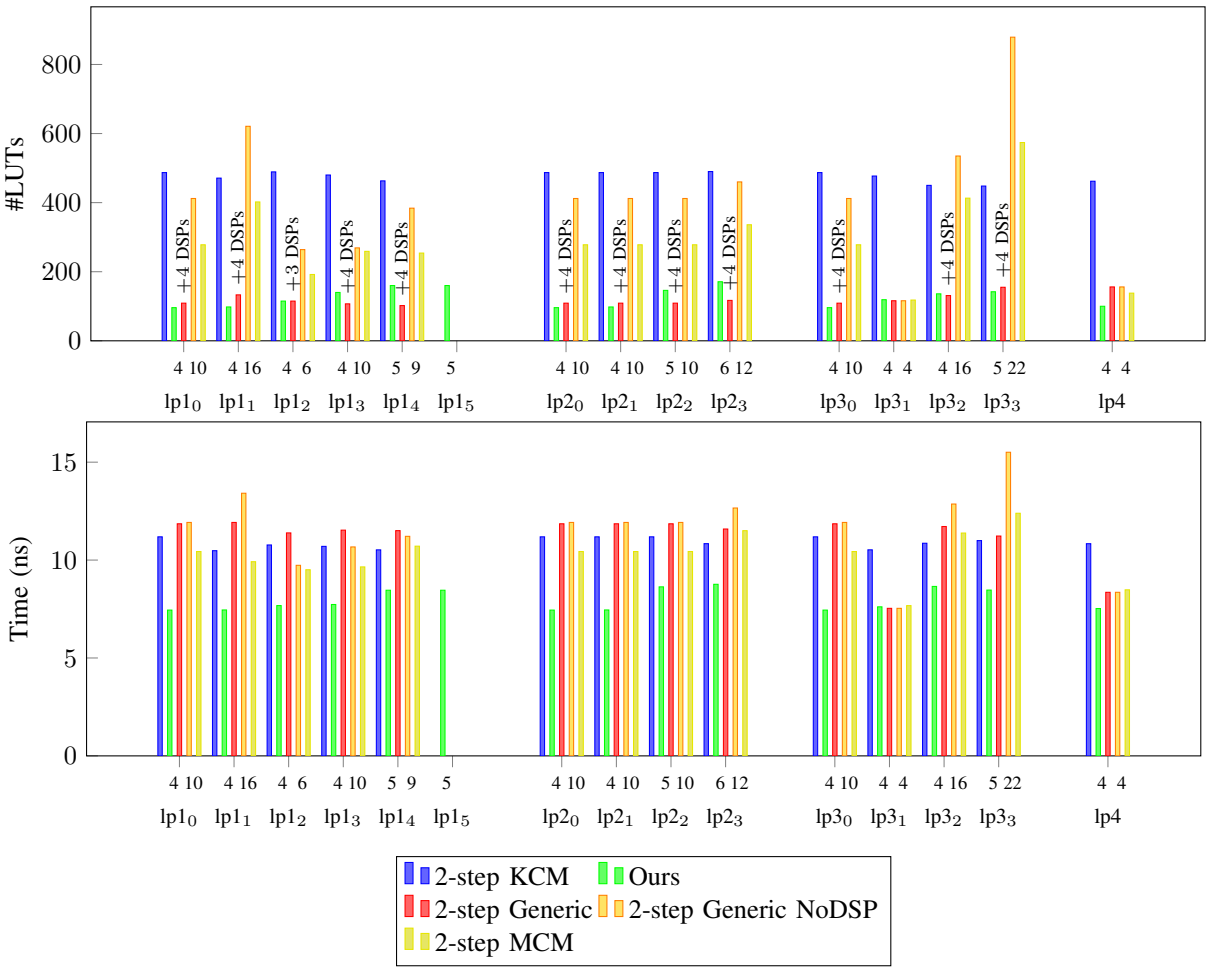

Fig. 5: Benchmark results. Here, the I/O worldength is 16 bits, and results for the lowest coefficient wordlength yielding no, for our approach, or acceptable ($< 10^{-7}$) frequency response error.

TABLE II: Optimization results for our method using ILP1 with CPLEX and synthesis results for a data wordlength of 16 – The time (in seconds) of the solving process is reported in the *Time* column and the number of adders are reported in three columns: A_M, A_S and A ($A = A_M + A_S$). Column w corresponds to the wordlength for which this was achieved. Columns *delay* and #LUTs are the synthesis results.

Names	W	ε	Time (s)	A_M	A_S	A	delay (ns)	#LUTs
$lp1_0$	4	0	2	1	4	5	7.446	96
$lp1_1$	4	0	2	1	4	5	7.452	98
$lp1_2$	4	0	2	2	4	6	7.676	115
$lp1_3$	4	0	2	3	4	7	7.731	140
$lp1_3$	7	0	13	2	4	6	7.754	123
$lp1_4$	5	0	2	4	4	8	8.460	160
$lp1_4$	7	0	14	3	4	7	8.439	147
$lp1_5$	5	0	2	4	4	8	8.460	160
$lp2_0$	4	0	2	1	4	5	7.446	96
$lp2_1$	4	0	2	1	4	5	7.452	98
$lp2_2$	5	0	3	3	4	7	8.632	146
$lp2_2$	10	0	320	3	4	7	7.617	138
$lp2_3$	6	0	2	4	4	8	8.763	171
$lp2_3$	7	0	7	4	4	8	8.924	163
$lp3_0$	4	0	2	1	4	5	7.446	96
$lp3_1$	4	0	2	2	4	6	7.612	119
$lp3_1$	10	0	59	2	4	6	7.829	116
$lp3_2$	4	0	2	3	4	7	8.654	136
$lp3_3$	5	0	2	3	4	7	8.468	142
lp4	4	0	1	1	4	5	7.525	74
lp4	7	0	7	1	3	4	7.570	100

IV. Conclusion and perspectives

- Our method works, we have certified optimal second-order IIR filters
- This probably can be extended to third and fourth order IIR filters as there exists explicit formulas for roots
- Is that important to do? (Numerical instability)
- Problem: Certified optimality is lost when using an external algorithm for the decomposition
- Can we overcome that with a better method?
- Extension to higher order IIR. Essentially needs to extend the stability constraints. They are nonlinear and can be included as constraints in an optimization problem, but they are not tractable (see, e.g., []). Sufficient conditions that can be linearized need to be investigated.

References

[1] A. Volkova, M. Istoan, F. de Dinechin, and T. Hilaire, "Towards Hardware IIR Filters Computing Just Right: Direct Form I Case Study," *IEEE Transactions on Computers*, pp. 1–1, 2019, https://doi.org/10.1109/TC.2018.2879432

[2] F. d. Dinechin, S.-I. Filip, L. Forget, and M. Kumm, "Table-Based versus Shift-And-Add Constant Multipliers for FPGAs," in *IEEE Symposium on Computer Arithmetic (ARITH)*, 2019, https://doi.org/10.1109/ARITH.2019.00037

[3] M. Kumm, O. Gustafsson, F. de Dinechin, J. Kappauf, and P. Zipf, "Karatsuba with Rectangular Multipliers for FPGAs," in *IEEE Symposium on Computer Arithmetic (ARITH)*, 2018, https://doi.org/10.1109/ARITH.2018.8464809. (achieved the best paper award)

[4] M. Kumm, "Optimal Constant Multiplication using Integer Linear Programming," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2018, https://doi.org/10.1109/TC.2018.2879432

[5] M. Kumm, D. Fanghänel, K. Möller, P. Zipf, and U. Meyer-Baese, "FIR Filter Optimization for Video Processing on FPGAs," *Springer EURASIP Journal on Advances in Signal Processing*, pp. 1–18, 2013.

[6] M. Kumm, A. Volkova, and S.-I. Filip, "Design of Optimal Multiplierless FIR Filters," *Submitted for publication, preprint available at https://arxiv.org/abs/1912.04210*, 2020.

[7] A. Dempster and M. Macleod, "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.

[8] Y. Voronenko and M. Püschel, "Multiplierless Multiple Constant Multiplication," *ACM Transactions on Algorithms*, vol. 3, no. 2, pp. 1–38, 2007.

[9] L. Aksoy, E. da Costa, P. Flores, and J. Monteiro, "Exact and Approximate Algorithms for the Optimization of Area and Delay in Multiple Constant Multiplications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 6, pp. 1013–1026, 2008.

[10] K. D. Chapman, "Fast Integer Multipliers Fit in FPGAs," *Electronic Design News*, 1994.

[11] M. Faust and C.-H. Chang, "Bit-parallel Multiple Constant Multiplication using Look-Up Tables on FPGA," *IEEE International Symposium of Circuits and Systems (ISCAS)*, pp. 657–660, 2011.

[12] Y. Wang, B. Li, and Z. Li, "Fixed-point digital IIR filter design using multi-objective optimization evolutionary algorithm," in *2010 IEEE Youth Conference on Information, Computing and Telecommunications*, Nov. 2010, pp. 174–177.

[13] H. Leich, "Toolbox for the design of IIR digital filters," in *Proceedings of 13th International Conference on Digital Signal Processing*, vol. 2, Jul. 1997, pp. 621–624.

[14] W. K. Jenkins and M. Nayeri, "Adaptive filters realized with second order sections," in *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Institute of Electrical and Electronics Engineers, 1986.

[15] A. G. Dempster and M. D. Macleod, "Comparison of IIR filter structure complexities using multiplier blocks," in *Proceedings of ISCAS'95 - International Symposium on Circuits and Systems*, vol. 2, Apr. 1995, pp. 858–861.

[16] A. Antoniou, *Digital Filters: Analysis, Design, and Signal Processing Applications*. New York: McGraw-Hill Education, 2018. [Online]. Available: https://www.mheducation.com/highered/product/digital-filters-analysis-design-signal-processing-applications-antoniou/9780071846035.html

[17] M. Kumm, A. Volkova, and S.-I. Filip, "Design of Optimal Multiplierless FIR Filters," 2019.
[18] R. Hettich and K. O. Kortanek, "Semi-infinite programming: Theory, methods, and applications," *SIAM Review*, vol. 35, no. 3, pp. 380–429, 1993.
[19] A. Volkova, C. Lauter, and T. Hilaire, "Reliable Verification of Digital Implemented Filters Against Frequency Specifications," in *2017 IEEE 24th Symposium on Computer Arithmetic (ARITH)*. IEEE, Jul. 2017.
[20] A. Marendet, A. Goldsztejn, G. Chabert, and C. Jermann, "A standard branch-and-bound approach for nonlinear semi-infinite problems," *European Journal of Operational Research*, vol. 282, no. 2, pp. 438 – 452, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221719308604
[21] A. Billionnet, S. Elloumi, and A. Lambert, "Linear Reformulations of Integer Quadratic Programs," in *Modelling, Computation and Optimization in Information Systems and Management Sciences*, H. A. Le Thi, P. Bouvry, and T. Pham Dinh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 43–51.
[22] F. Glover, "Improved Linear Integer Programming Formulations of Nonlinear Integer Problems," *Management Science*, vol. 22, no. 4, pp. 455–460, Dec. 1975.
[23] M. Oral and O. Kettani, "A Linearization Procedure for Quadratic and Cubic Mixed-Integer Problems," *Operations Research*, vol. 40, no. 1-supplement-1, pp. S109–S116, Feb. 1992.
[24] D. Bertsimas and J. Tsitsiklis, *Introduction to linear optimization*. Belmont, Mass: Athena Scientific, 1997. [Online]. Available: https://dl.acm.org/doi/book/10.5555/548834
[25] O. L. Mangasarian, "Absolute value programming," *Computational Optimization and Applications*, vol. 36, no. 1, pp. 43–53, Nov. 2006.
[26] T. Walsh, "General Symmetry Breaking Constraints," in *Principles and Practice of Constraint Programming - CP 2006*. Springer Berlin Heidelberg, 2006, pp. 650–664.
[27] A. Goldsztejn, C. Jermann, V. Ruiz de Angulo, and C. Torras, "Variable symmetry breaking in numerical constraint problems," *Artificial Intelligence*, vol. 229, pp. 105 – 125, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370215001216
[28] M. Kumm, *Multiple Constant Multiplication Optimizations for Field Programmable Gate Arrays*. Wiesbaden: Springer Fachmedien Wiesbaden, 2016.
[29] ——, "Optimal Constant Multiplication Using Integer Linear Programming," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 5, pp. 567–571, 2018.
[30] F. Krach, B. Frackelton, J. Carletta, and R. Veillette, "FPGA-based implementation of digital control for a magnetic bearing," in *Proceedings of the 2003 American Control Conference, 2003.* IEEE, 2003.
[31] J. Carletta, R. Veillette, F. Krach, and Z. Fang, "Determining appropriate precisions for signals in fixed-point IIR filters," in *Proceedings 2003. Design Automation Conference (IEEE Cat. No.03CH37451)*, Jun. 2003, pp. 656–661.
[32] O. Sarbishei, K. Radecka, and Z. Zilic, "Analytical Optimization of Bit-Widths in Fixed-Point LTI Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 3, pp. 343–355, Mar. 2012.
[33] A. Volkova, M. Istoan, F. De Dinechin, and T. Hilaire, "Towards Hardware IIR Filters Computing Just Right: Direct Form I Case Study," *IEEE Transactions on Computers*, vol. 68, no. 4, pp. 597–608, Apr. 2019.
[34] F. de Dinechin and B. Pasca, "Designing Custom Arithmetic Data Paths with FloPoCo," *IEEE Design & Test of Computers*, vol. 28, no. 4, pp. 18–27, Jul. 2011.