DESIGN OF DIGITAL FILTERS AND FILTER BANKS BY OPTIMIZATION: APPLICATIONS

Tapio Saramäki and Juha Yli-Kaakinen
Signal Processing Laboratory, Tampere University of Technology,
P.O.Box 553, Tampere, FINLAND
Tel: +358 3 365 2930; fax: +358 3 365 4333
e-mail: ts@cs.tut.fi; juha.yli-kaakinen@tut.fi

SUMMARY

DURING the last two decades, the role of digital signal processing (DSP) has changed drastically. Twenty years ago, DSP was mainly a branch of applied mathematics. At that time, the scientists were aware of how to replace continuous-time signal processing algorithms by their discrete-time counterparts providing many attractive properties. These include, among others, a higher accuracy, a higher reliability, a higher flexibility, and, most importantly, a lower cost and the ability to duplicate the product with exactly the same performance.

Thanks to dramatic advances in VLSI circuit technology as well as the development in signal processors, these benefits are now seen in the reality. More and more complicated algorithms can be implemented faster and faster as well as in a smaller and smaller silicon area and with a lower and lower power consumption.

Due to this fact, the role of DSP has changed from theory to a “tool”. Nowadays, the development of products requiring a small silicon area as well as a low power consumption in the case of integrated circuits is desired. The third important measure of the “goodness” of the DSP algorithm is the maximal sampling rate that can be used. In the case of signal processors, the code length is a crucial factor when evaluating the effectiveness of the DSP algorithm. These facts imply that the algorithms generated twenty years ago have to be re-optimized by taking into account the implementation constraints in order to generate optimized products.

Furthermore, when generating DSP products, all the subalgoritms should have the same quality. A typical example is a multirate analysis-synthesis filter bank for subband coding. If a lossy coding is used, then there is no need to use a perfect-reconstruction system due to the errors caused by coding. It is more beneficial to improve the filter bank performance in such a way that small errors are allowed in both the reconstruction and aliasing transfer functions. The goal is to make these errors smaller than those caused by coding and simultaneously either to improve the filter bank performance or to achieve a similar performance with a reduced overall delay.

In addition, there exist various synthesis problems where one of the responses is desired to be optimized in some sense while keeping some other responses, depending on the same design parameters, within the given tolerances. A typical example is the minimization of the phase distortion of a recursive filter subject to the given amplitude specifications. There are also problems where some of the design parameters are fixed or there are constraints among them.

In order to solve the above-mentioned problems effectively, in very few cases analytic or simple iterative design schemes can be used. In most cases, there is a need to use optimization. In some cases like in designing linear-phase FIR filters subject to some constraints, linear programming can be used. In many other cases, nonlinear optimization has to be applied to give the optimum solution.

This paper focuses on using two techniques for solving various unconstrained and constrained optimization problems for DSP systems. The first one uses linear programming for optimizing linear-phase FIR filters subject to some linear constraints, whereas the second one utilizes an efficient two-step strategy for solving other types of problems. First, a suboptimum solution is generated using a simple design scheme. Second, this starting-point solution is further improved using an efficient general-purpose nonlinear optimization algorithm, giving the desired optimum solution.

Three alternatives are considered for constructing the general-purpose nonlinear optimization algorithm. The first one is generated by modifying the second algorithm of Dutta and Vidyasagar, the second one uses a transformation of the minimax problem into a nonlinearly constrained problem, whereas the third one is based on the use of sequential quadratic programming (SQP) methods. It should be pointed out that in order to guarantee the convergence to the optimum solution, the first step in the overall procedure is of great importance.

The efficiency and the flexibility of using optimization for finding optimized DSP algorithms is illustrated by means of six applications. The first five applications utilize the above-mentioned two-step strategy, whereas the last one is based on the use of linear programming.

In the first application, cosine-modulated multichannel analysis-synthesis filter banks are optimized such that the filter bank performance is optimized subject to the given allowable reconstruction and aliasing errors. In this case,

a starting-point solution is a perfect-reconstruction filter bank generated using a systematic multi-step design scheme. Then, one of the above-mentioned general-purpose optimization algorithms is applied. It is shown that by allowing very small reconstruction and aliasing errors, the filter bank performance can be significantly improved compared to the perfect-reconstruction case. Alternatively, approximately the same filter bank performance can be achieved with a significantly reduced overall filter bank delay.

In the second application, the phase distortion of a recursive digital filter is minimized subject to the given amplitude criteria. The filter structures under consideration are conventional cascade-form realizations and lattice wave digital filters. For both cases, there exist very efficient design schemes for generating the starting-point solutions, making the further optimization with the aid of the general-purpose optimization algorithm very straightforward.

The third application concentrates on optimizing the modified Farrow structure proposed by Vesma and Saramäki to generate a system with an adjustable fractional delay. For this system, the overall delay is of the form $D_{\rm int} + \mu$, where $D_{\rm int}$ is an integer delay depending of the order of the building-block non-recursive digital filters and $\mu \in [0,\ 1)$ is the desired fractional delay. This fractional delay is a direct control parameter of the system. The goal is to optimize the overall system in such a way that for each value of μ the amplitude response stays within the given limits in the passband region, and the worst-case phase delay deviation from $D_{\rm int} + \mu$ in the given passband is minimized. Also in this case, it is easy to generate the starting-point solution for further optimization.

The fourth application addresses the optimization of the magnitude response for the pipelined recursive filters. In this case, there exist several algorithms for generating a start-up filter for further optimization. It is shown that by applying one of the above-mentioned optimization algorithms, the magnitude response of the pipelined filters compared to that of the initial filter can be considerably improved.

The last two applications show how the coefficients of the digital filters can be conveniently quantized utilizing optimization techniques. The first class of filters under consideration consists of conventional lattice wave digital (LWD) filters, cascades of low-order LWD filters providing a very low sensitivity and roundoff noise, and LWD filters with an approximately linear phase in the passband. The second class of filters are conventional linear-phase non-recursive filters. For both filter types a similar systematic technique is applied for finding the optimized finite-precision solution.

For filters belonging in the first class, it was observed that by first finding the largest and smallest values for both the radius and the angle of all the complex-conjugate poles, as well as the largest and smallest values for the radius of a possible real pole, in such a way that the given criteria are still met, we are able to find a parameter space which includes the feasible space where the filter specifications are satisfied. After finding this larger space, all what is needed is to check whether in this space there exist the desired discrete values for the coefficient representations. To solve these problems one of the above-mentioned optimization algorithm are utilized. For filters belonging in the second class, the largest and smallest values for all the coefficients are determined in a similar manner in order to find the feasible space. In this case, the desired smallest and largest values can be conveniently found by using linear programming.

The full-length paper can be found in `http://www.cs.tut.fi/~ts`.