A DIFFERENTIAL GEOMETRIC APPROACH TO DISCRETE-COEFFICIENT FILTER DESIGN

Subramanian Ramamoorthy

School of Informatics
The University of Edinburgh
Edinburgh EH8 9AB, UK
s.ramamoorthy@ed.ac.uk

Lothar Wenzel, James Nagle, Bin Wang, Michael Cerna

Mathematics and Signal Processing Group
National Instruments Corp.
Austin, TX 78759, USA
{lothar.wenzel, james.nagle, bin.wang,
michael.cerna}@ni.com

ABSTRACT

This paper is concerned with the problem of computing a discrete-coefficient approximation to a digital filter. In contrast to earlier works that have approached this problem using standard combinatorial optimization tools, we take a geometric approach. We define a Riemannian manifold, arising from the difference in frequency response between the two systems of interest, on which we design efficient algorithms for sampling and approximation. This additional structure enables us to tame the computational complexity of the native combinatorial optimization problem. We illustrate the benefits of this approach with design examples involving IIR and FIR filters.

Index Terms— System analysis and design, Digital filters, Differential geometry

1. INTRODUCTION

The notions of distance between linear systems, computations of near neighbors and paths between them that minimize the distance in a suitable sense are fundamental to a variety of problems in system theory [1, 2]. In practice, design techniques do not always pay attention to this underlying geometric structure. For instance, in the problem of discrete-coefficient filter design, one tries to minimize a cost function that includes a penalty for deviation from a desired frequency response, and perhaps a few other terms. A naïve approach that generates a discrete-coefficient filter by rounding coefficients of a corresponding infinite or arbitrary-precision design is rarely sufficient in terms of quality. However, more systematic design methods based on optimization face the problem of dealing with a multimodal cost function over a discretized domain - representing significant computational complexity. In this paper, we propose an algorithm for dealing with such problems that more directly utilizes the geometry of the space of systems. In particular, we use the distance between filters, in the sense of frequency response, to describe a nonlinear manifold on which we can focus the search for a nearest discrete-coefficient neighbor of the given arbitrary-precision filter.

Owing to its practical significance, the problem area has attracted the attention of several research groups. Here, we cite only a few representative works to illustrate methods that have been adopted. A common early thread in this area is the use of integer programming to directly minimize an error with respect to desired frequency response [3]. With the availability of increased computing power, this has recently been extended to parallel implementation of large-scale mixed-integer linear programs [4]. The optimization problem has also been approached using numerous heuristic search methods including simulated annealing [5], evolution strategies [6], [7], ant colony optimization [8] and other local search methods [9]. In most of these approaches, the cost function is represented in a weighted sum form - with terms for frequency response error, concerns such as psycho-physical requirements [6] and a measure of sparsity or complexity (e.g., number of non-zero bits or multipliers).

Our approach is to view the error function as yielding the ‘local’ distance on a Riemannian manifold of digital filters. In general, even with a proper definition of such a nonlinear manifold, it can be hard to derive closed form results for geodesics and neighborhoods. Instead, we design approximation algorithms that leverage this structure while keeping computation tractable. We achieve this by drawing on earlier work, regarding sampling in abstract spaces, to define a finite point-set on the manifold that is well-distributed with respect to the chosen metric. This has the effect of significantly reducing complexity as compared to uninformed search.

2. GEOMETRY OF THE SPACE OF DIGITAL FILTERS

A discrete IIR filter can be defined in time-domain as,

\[ \sum_{j=0}^{M} a_j y[n-j] = \sum_{k=0}^{N} b_k x[n-k] \quad (1) \]

where \( N, M \) are the orders of the numerator and denominator, \( a_j, b_k \) are the reverse and forward coefficients, respectively. By convention, \( a_0 = 1 \). After a Z-transform, \( z = e^{j2\pi f} \) or \( e^{j\phi} \),

\[ H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{j=1}^{M} a_j z^{-j}} \quad (2) \]

There are standard ways of designing the coefficients of this system subject to a variety of performance specifications. When dealing with two such systems, say, \( H(e^{j\phi}) \) and \( K(e^{j\phi}) \) where the former is the designed system and the latter is a ‘desired’ system, then one can define an error,

\[ \Phi(a_1, ..., a_M, b_0, ..., b_N) = \int_{0}^{2\pi} \left( |H(e^{j\phi})| - |K(e^{j\phi})| \right)^2 d\phi \quad (3) \]

A typical design goal is to minimize this functional over a suitable field of coefficients. In particular, we can address the problem with \( K(e^{j\phi}) \) representing an arbitrary-precision designed goal and \( H(e^{j\phi}) \) representing a target system involving coefficients \( a_j, b_k \) that are restricted to integers, powers-of-two numbers, etc.
Consider two filters defined by the coefficients, \( c^{(1,2)} = (a^{(1,2)}_1, a^{(1,2)}_2, \ldots, a^{(1,2)}_M, b^{(1,2)}_0, \ldots) \), with a distance between them,

\[
D(c^{(1)}, c^{(2)}) = \sqrt{\frac{1}{2 \pi} \int_0^{2 \pi} \left| H_{c^{(1)}}(e^{j \phi}) - H_{c^{(2)}}(e^{j \phi}) \right|^2 d\phi}
\]

with \( c^{(2)} = c^{(1)} + dc^{(1)} \), we can write the distance as,

\[
ds^2 = D^2(c^{(1)}, c^{(1)}) + dc^{(1)} = \sum_{i,j=1}^{N+M+1} g_{ij}(c^{(1)}) dc_i dc_j + O((dc^{(1)})^3)
\]

where \( \alpha_1 = a^{(1)}_1, \ldots, \alpha_M = a^{(1)}_M, \alpha_{M+1} = b^{(1)}_0, \ldots, \alpha_{M+N+1} = b^{(1)}_N \) and \( g \) becomes a metric tensor in the sense of Riemannian geometry [2].

The above equation, although derived from differences in frequency response, captures more information about the nature of the problem. For instance, one could compute the Gauss curvature [2], \( K \), of the manifold represented by this metric and infer facts such as that zero curvature corresponds to a simple re-parametrization for which standard parameter optimization suffices whereas a manifold with non-zero curvature calls for more sophisticated treatment.

We remark that this is one among many ways to define a metric appropriate to the desired system response. The practical difficulty is that near neighbors are situated, we define the sampling region (section 5). Having established the geometric structure, there is still the issue of how to compute on this manifold. Analytically determining paths and regions on general manifolds is a hard problem, infeasible except in special cases. We approach this computation indirectly, by first sampling the space to create a finite point-set that is fair (in a precisely defined sense) with respect to the metric. The exact procedure is described below.

Consider a nonlinear manifold that is defined in terms of coordinates \( u_1, \ldots, u_n \) and the metric tensor \( g \). If one considers this manifold to be generated by some mapping from the Euclidean plane, then we define fair mappings to be those that preserve area or volume in the sense of

\[
\Psi(u_1, \ldots, u_n) = \sqrt{det(g_{ij}(u_1, \ldots, u_n))} = c
\]

Once we have such a mapping (for some \( c \)), if we had an efficient procedure for incrementally sampling the Euclidean plane, then - through this mapping - the same procedure lets us efficiently sample on the manifold. The following theorem, from [12], provides conditions that such a mapping should satisfy.

**Theorem 3.1** Let \( \Psi(u_1, u_2, \ldots, u_n) \) be a nonnegative function on \([0, 1]^n\) where \( \Psi^3 \) is continuously differentiable. Let \( \Psi(u_1, u_2, \ldots, u_n) \) be positive with exception of a set \( L \) of Lebesgue-measure 0. For \( (u_1, u_2, \ldots, u_n) \notin L \), let

\[
f_3(u_1, u_2, \ldots, u_n) = \int_0^{u_1} \Psi(u_1, \ldots, u_n) du_1 \int_0^{u_2} \Psi(u_1, \ldots, u_n) du_1 \ldots \int_0^{u_n} \Psi(u_1, \ldots, u_n) du_1 \ldots du_n
\]

Furthermore, let the functions \( f_1, f_2, \ldots, f_n \) be extendable to continuously differentiable mappings defined on \([0, 1]^n\). Then the extension functions \( \tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_n \) define a diffeomorphism of \([0, 1]^n \setminus L\) where equation 6 is valid for a constant \( c \).

Based on this, we can define the following algorithm for sampling on a manifold:

1. Given an abstract surface \( S \) defined on \([0, 1]^n\), where \( \Psi(u_1, \ldots, u_n) \) satisfies propositions of theorem 3.1 and where \( x(u_1, \ldots, u_n) \) is an embedding of \( S \) in \( \mathbb{R}^n \), construct \( f(u_1, \ldots, u_n) = (f_1(u_1, \ldots, u_n), \ldots, f_n(u_1, \ldots, u_n)) \) defined over \([0, 1]^n\).

2. Compute the inverse, \( f^{-1}(u_1, \ldots, u_n) \).

3. Beginning with a well-distributed set \( D \in \mathbb{R}^n \) (incrementally generated by a quasi-Monte Carlo method), compute the image of \( D \) under the transform \( x(f^{-1}(u_1, \ldots, u_n)) \).

The output of the above procedure is a well-distributed point-set on the nonlinear manifold.

**4. ALGORITHM FOR FINDING THE NEAREST DISCRETE-COEFFICIENT NEIGHBOR OF A FILTER**

We proceed with the design in a two-stage process. The first stage involves local optimization to determine a candidate discrete-coefficient filter (a point in a multi-dimensional lattice). This computation is local in the sense that it focusses search to a unimodal neighborhood, on the error surface, of the arbitrary-precision design. The outcome of this step is an initial guess that is much better than the Lie group structure of the space of linear systems [10] or other approaches have been focussed on analysis rather than synthesis, e.g., to understand properties of spectral density functions [11]. Indeed, the procedure described in the following sections could be adapted to work with these advanced metrics as well.

**3. WELL-DISTRIBUTED SETS ON NONLINEAR MANIFOLDS**

Our design problem is essentially that of finding a nearest neighbor to the desired system response. The practical difficulty is that this neighborhood relation is not very well behaved in the coefficient space and naïve approaches such as rounding yield very poor approximations. Instead, we make use of the fact that the proper definition of neighborhood is provided by a metric such as in equation 5. Having established the geometric structure, there is still the issue of how to compute on this manifold. Analytically determining paths and regions on general manifolds is a hard problem, infeasible except in special cases. We approach this computation indirectly, by first sampling the space to create a finite point-set that is fair (in a precisely defined sense) with respect to the metric. The exact procedure is described below.

Consider a nonlinear manifold that is defined in terms of coordinates \( u_1, \ldots, u_n \) and the metric tensor \( g \). If one considers this manifold to be generated by some mapping from the Euclidean plane, then we define fair mappings to be those that preserve area or volume in the sense of

\[
\Psi(u_1, \ldots, u_n) = \sqrt{det(g_{ij}(u_1, \ldots, u_n))} = c
\]

Once we have such a mapping (for some \( c \)), if we had an efficient procedure for incrementally sampling the Euclidean plane, then - through this mapping - the same procedure lets us efficiently sample on the manifold. The following theorem, from [12], provides conditions that such a mapping should satisfy.
There are many efficient algorithms for solving this problem [13]. In high dimensions, the point $\hat{c}^*$ represents a significantly better starting point for further sampling than, e.g., rounding.

Next, we need to determine the extent of the region within which to sample. Here, we draw on a famous result by Minkowski [14]. We ask - given our knowledge of the curvature of the space, which manifests itself as the aspect ratio and principal direction of an ellipsoid centered on the current point, how long should the axes be so as to guarantee inclusion of at least one non-trivial grid point (representing a discrete coefficient design)?

**Theorem 4.1 (Minkowski)** Let $S$ be a subset in $\mathbb{R}^n$ that is symmetric with respect to the origin. If the volume $V$ of $S$ is greater than $2^n$, then $S$ must contain at least 3 grid points from a uniform axis-aligned lattice - the origin and two non-trivial points $\pm P$.

Consider the case of an $n$-dimensional ellipsoid $S$ with principal axes of length $r_1, \ldots, r_d$. The corresponding volume is,

$$V = \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2} + 1\right)} r_1 \cdots r_d$$  \hspace{1cm} (9)

If this ellipsoid were scaled by a positive factor $s$, guaranteeing the existence of a grid point within it, then

$$s \geq \frac{2}{\sqrt{n}} \sqrt{\Gamma\left(\frac{d}{2} + 1\right)}$$  \hspace{1cm} (10)

Knowing that,

$$\frac{2}{\sqrt{n}} \sqrt{\Gamma\left(\frac{d}{2} + 1\right)} < \sqrt{d}$$  \hspace{1cm} (11)

the scaling factor reduces to,

$$s > \frac{\sqrt{d}}{\sqrt{r_1 \cdots r_d}}$$  \hspace{1cm} (12)

**Algorithm 1 Design of Discrete-Coefficient Digital Filter**

**INPUT**: Floating point filter defined by coefficients $C_0$; discretization level for desired filter; Error Tolerance

**OUTPUT**: Discrete-coefficient filter defined by $\hat{C}$

Compute metric tensor $g$, equation 5.

Construct a semi-definite matrix $Q$ that agrees with $g$ at $C_0$

Solve for initial guess $C_0$ by a minimization, equation 8

Compute scale factor for an ellipse centered at $C_0$, equation 12

while $\text{Error} > \text{Tolerance}$ do

Generate samples within ellipse, section 3

These samples are not necessarily grid points - compute closest grid point using equation 8

end while

5. EXPERIMENTS AND EXAMPLES

In this section, we illustrate the use of the above algorithm using concrete examples. We begin with a first-order IIR filter, for which we work out the metric and illustrate the geometric issues discussed thus far. Then, we present results for more complex systems, demonstrating tangible benefits in terms of design objectives.

Consider the filter,

$$H(z) = \frac{b}{1 + az^{-1}}$$  \hspace{1cm} (13)

The corresponding Riemannian metric is,

$$ds^2 = \frac{1}{2\pi} \int_0^{2\pi} \left| \frac{b + db}{1 + (a + da)e^{-j\phi}} - \frac{b}{1 + ae^{-j\phi}} \right|^2 d\phi$$  \hspace{1cm} (14)

$$ds^2 = \frac{(1 + a^2)b^2}{(1 - a^2)^3} da^2 + \frac{4ab}{(1 - a^2)^2} dadb + \frac{1}{1 - a^2} db^2 + O(da^3, da^2 db, db^2, db^3)$$  \hspace{1cm} (15)

The resulting behavior is summarized in figures 1 and 2. Next, we present further design examples in figures 3 and 4, where we compare the performance of algorithm 1 against uninformed random sampling in the neighborhood of the floating-point design. We make this comparison by computing a factor $q$ which measures the number of samples required by uninformed random sampling over that required by the procedure of algorithm 1. To the extent that many heuristic search and global optimization algorithms utilize such random sampling as an essential ingredient, this measure illustrates the benefits of incorporating geometric information into the process.
We conclude this section with a simple, somewhat counterintuitive, example that underscores the benefits of the geometric approach. Consider an FIR filter of the form,

$$H(z) = (1 - az^{-1})(1 - bz^{-1})$$  \hspace{1cm} (16)

The corresponding Riemannian metric can be derived as,

$$ds^2 = (1 + t^2)da^2 + 2(1 + ab)dadb + (1 + a^2)db^2$$  \hspace{1cm} (17)

A specific floating-point design is $(a = 0.5, b = 200.5)$. A naïve approach might suggest the discrete equivalent $(a = 1, b = 200)$. In fact, in the sense of the metric tensor above, a much better design is $(a = 1, b = 150)$, significantly further away from the original design than one might have intuitively expected. Moreover, even with a heuristic search algorithm such as simulated annealing, this global optimum would only be found with an excessive amount of randomized search when compared to the geometric alternative described above.

6. CONCLUSIONS

We present an algorithm for computing a discrete-coefficient approximation to an arbitrary-precision filter, utilizing the non-euclidean geometry arising from the difference in frequency response. This structure allows us to significantly focus search so that the computational complexity of the native combinatorial optimization problem can be tamed. The algorithm, as presented, is the first step in a larger research program. In future, we seek other systematic procedures to improve estimates such as in equation 12 and other sophisticated versions of local optimization so that, in combination with the geometric sampling procedure, the efficiency of design can be further improved - enabling more complex systems and specifications.

7. REFERENCES