Filter Approximation Using Explicit Time and Frequency Domain Specifications

Varun Aggarwal
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA
varun_ag@mit.edu

Wesley O Jin
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA
wes_jin@mit.edu

Una-May O'Reilly
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA
unamay@csail.mit.edu

ABSTRACT

We demonstrate that enhanced particle swarm optimization (PSO) can be successfully used to evolve high performance filter approximations. These evolved approximations use sets of quantitative specifications which conventional analytically derived approximations can not directly employ. The conventional derivations use only a subset of the quantitative specifications in their algorithm and the remaining specifications are side-effect results of the algorithm. Thus, with enhanced PSO, instead of a filter designer having access to a limited set of " specification knobs" that directly and indirectly achieve performance, a designer has a "knob" for each specification that consequently drives the approximation to desired performance. Our future vision to do performance space modeling for filters is also discussed.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: Integrated Circuits-Design Aids

General Terms

Algorithms, Design

Keywords

filter design, particle swarm optimization

1. INTRODUCTION

A filter is a linear frequency selective circuit, which attenuates certain frequencies and amplifies certain others. An ideal (lowpass) filter with so-called "brickwall" characteristic has frequency domain characteristics as shown in Figure 1. It has a gain of 1 (0 dB) in the passband and a gain of 0 in the stopband. The frequency at which the gain reduces from 1 to 0 is called the cut-off frequency (ω_c).

The primary performance criterion of a filter is its frequency domain magnitude response (henceforth called, magnitude response, defined in Section 3.4). A second set of performance criteria are time domain characteristics. These can be tested by examining the step and impulse responses of the filter. Ideally there should be no oscillations in the step response, a small overshoot in case of oscillations, a low rise time and settling time. A filter meeting the brickwall characteristic has ideal frequency band selection but will oscillate on being activated by a step (in addition to voiding causality). Another criterion is a linear phase response in frequency domain (henceforth called phase response). Finally, because there is a tradeoff between filter complexity (i.e. order) and implementation feasibility, complexity is a performance criterion. [1]

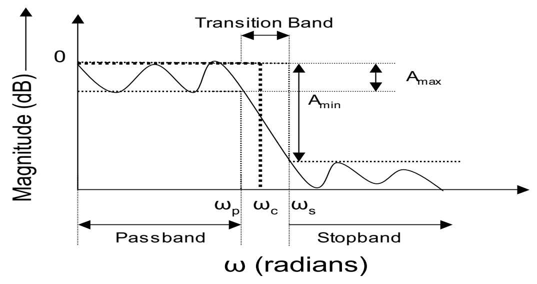

Figure 1: Characteristics of a normalized lowpass filter.

Ideal filter characteristics are practically unrealizable and designers accept this limitation. More vexing to the designer is the fact that performance characteristics are coupled to each other in non-linear and indirect ways. Thus optimizing on solely one criterion may result in undesired performance in another. For instance, a better magnitude response may result in a worse time domain response. Hence, the designer has to contend with trade-off solutions.

Design of analog filters is hierarchical in nature. From an initial specification, the final goal is to synthesize a filter circuit as either lumped components for discrete component implementation or as a layout for silicon implementation. The high level steps are:

[1]A filter must be causal for realizability. A filter meeting brickwall characteristics requires a non-causal system.

1. Filter Approximation: Conversion of specifications to filter transfer function
2. Realization of transfer function as a prototype i.e. set of abstract blocks realizable in circuits.
3. Realization of prototype into an actual circuit.
4. Layout design for the circuit.

In this submission we focus on filter approximation which is the step of constructing the Laplace domain transfer function [13] of the filter given its specifications. For a set of conditions (satisfied by all real filters), the Laplace domain transfer function can completely express any linear-time invariant system [13]. Equation 1 show the general form of transfer function.

$$H(s) = \frac{a_m s^m + a_{m-1}s^{m-1} + \cdots + a_0}{s^n + b_{n-1}s^{n-1} + \cdots + b_0} \tag{1}$$

$$H(s) = \frac{a_m(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)} \tag{2}$$

The variable s is the Laplace domain variable. Symbols a_k and b_k are coefficients of the polynomial in s in the numerator and denominator respectively. They are required to be real for synthesizing a realizable filter. Equation 1 can be factored into Equation 2, where z_k and p_k are called the zeros and poles of the transfer function. Zeroes and poles can be imaginary but need to exist in complex-conjugate pairs whenever imaginary. The Laplace domain has a one-to-one mapping to the frequency domain and one gets the frequency domain characteristics by substituting $j\omega$ for s in Equation 1 or Equation 2. The maximum power to which the s term is raised in the numerator or denominator is the order of the transfer function. A filter transfer function is completely specified by its order and the value of coefficients for all s terms (or equivalently value of all pole-zero pairs). The complexity of a filter is directly correlated to its order, i.e. the higher the order, the higher complexity of the filter.

A designer starts with a set of specifications which encapsulate the filter performance criteria. Some of these specifications are "hard", that is, the designer has specific quantities or values as goals. Typically, magnitude response specifications are quantitatively defined. The designer has a clear idea of the desired amplification and attenuation in magnitude and frequency range. It is straight forward to stipulate the maximum gain variation tolerable in the passband and the minimum required attenuation in the stopband. These specifications, shown in Figure 1, are listed in Table 1.

Specification	Symbol	Units
Passband Frequency	ω_p	rads/s
Stopband Frequency	ω_s	rads/s
Max allowed variation of passband	A_{max}	dB
Min reqd Stopband Attenuation	A_{min}	dB

Table 1: "Hard" Filter Design Specifications

Interestingly, the remaining specifications (i.e. those driving time domain, phase response and complexity) are usually qualitatively defined. For instance, the designer simply expresses a desire for a linear phase in the passband. Other qualitative specifications are: no ripple in passband, or stopband or both, fast settling time, low complexity. A good question to ask is why these specifications (or their error margins, e.g. minimize the error in linearity) are not quantitatively defined? The answer is that, with conventional approximation methods, hard specifications for these performance criteria cannot be incorporated. Conventional methods do not algorithmically reference and use these quantities to determine the transfer function. Instead, these quantities "fall out" or are consequentially derived from the key equations of the algorithm.

Engineers employ a small set (e.g. Butterworth, Elliptical, Chebyshev, Inverse Chebyshev, Bessel and 2 or 3 others) of such conventional deterministic methods to synthesize a transfer function. Because the methods do **not** address soft specifications, engineers use rules-of-tumb and experiential knowledge to select a method. For example, a Butterworth filter [15] (i.e a filter transfer function designed using the methods and equations given by Butterworth) will meet the magnitude response specifications. Additionally, it will have no ripples in passband and a monotonic behavior. However, it may have a long transition band. Similarly, there exists a Chebyshev [15] filter meeting the hard specifications. While it will have ripples, it will be a lower complexity (i.e. order) than a Butterworth. Qualitative claims such as a Butterworth filter having better time domain properties than Chebyshev can also be made.

Before the era of fast, cheap computation, each method would be encapsulated as a table of transfer function coefficients and engineers would realize a filter with the table data. The tables (and algorithms that are directly used today) supply order-based coefficients for a normalized low-pass filter. The designer used to mathematically transform the coefficients to the non-normalized form in a straight forward manner.

A vast space of filter coefficients remains *unexplored* by classical approximations. Our investigation asks (i) whether this space is interesting with respect to optimizing a performance criterion or providing a desirable tradeoff between criteria and (ii) can it be effectively explored with an evolutionary algorithm to find useful solutions? Is it possible to integrate, within an optimization tool, quantitative specifications that are conventionally qualitative in order to drive the *design* of filter? Can solutions which completely dominate the conventional approximations be discovered? Our investigation conducts experiments where conventional soft filter specifications are used either as objectives and constraints for filter design. In the context of a specific design problem, we ask whether our evolutionary technique is superior to the state of the art approximation method that employs Sequential Quadratic Programming (SQP) [2, 10].

The paper proceeds in the following manner: In Section 2 we review Evolutionary Hardware approaches to filter design and position our work as novel within its context. Techniques other than evolutionary algorithms (EAs) applied to filter approximation are also discussed in this section. The state of the art employs Sequential Quadratic Programming (SQP) [10, 2]. Both conventional hard specifications and soft specifications (formerly defined qualitatively) are expressed in the SQP's objective function and constraints. We compare this with our approaches.

Section 3 describes OptimFilt, our algorithm for filter approximation. Section 4 describes OptimFilt's results. We have selected our specifications to match those published for SQP so that a fair comparison can be drawn. Section 5 summarizes the work done till now and Section 6 discusses our future vision for this project and preliminary experiments.

2. RELATED WORK

Since the inception of the field, researchers in evolvable hardware considered automatic design of analog filters. Seminal work by Grimbleby, Koza, and Goh-Li [6, 8, 4] showed how analog filters could be evolved extrinsically using SPICE simulations, while Higuchi [12, 11] designed real-time adaptive analog filters using intrinsic evolution. The extrinsic approach merged the first 3 steps of filter design (transfer function approximation, prototype, circuit realization) into one. A magnitude response specification combined with a vocabulary of circuit elements evolved a sized topology for a filter. This method is useful because it directly integrates the magnitude response specification to the specifications of subsequent steps. For example, in the multi-step approach coefficients from the transfer function may not be amenable to the component ranges and precision available. Goh-Li and Vondras [4, 17] showed that sensitivity of response with respect to a component value can be addressed with this approach.

This merged-steps approach, however, is not without disadvantages. In some sense the EA's search encompasses the space of possible transfer functions, the space of prototypes and the set of circuit realizations for any transfer function. This search space is larger and has more complex relationships than the space of coefficients of a transfer function or that of circuits for a given transfer function, thus making the search harder (resultantly, probably sub-optimal). Secondly, a different EA must be developed and invoked for each technology (e.g. passive filters; OTA-C filters) and the method loses it generality. From the practical aspect, one might question, whether these methods are too "automatic". The EA assumes more control than a human might desire by not allowing the designer to inspect and consider tradeoffs between specifications across steps.

These approaches use the following performance criteria: In [5, 9] the criterion was to mimic the brickwall in the passband and stopband. In [8, 4] A_{max} and A_{min} specifications were also added to the brickwall objective. Here, the evolved circuit just met magnitude response objective, but disregard other criteria that may fall out of acceptable range. In [9, 4], the criterion is to mimic the Butterworth characteristics in magnitude response, which amalgamates prototype synthesis and circuit realization. This is more practical, since the realized circuit inherits the characteristics of the Butterworth filter. In [5] a time domain filter is evolved, but frequency domain specifications are not considered. None of these approaches address multiple objectives concerning magnitude response, phase response and time response criteria together. In [17], a multi-criteria framework for sizing a filter topology incorporates magnitude response, ranges for component values, and maximum voltage across components as objectives and constraints. Another instance of multi-objective evolution of active filters is [18] which solely considers magnitude response specifications. More recently, filter circuits have been generated using non-linear components [16], where specifications of magnitude response and the DC range are used. There has been a lot of work on digital filter synthesis using evolutionary algorithms [1], some of which addresses filter approximation. The focus has been to decrease the effect of discrete coefficient space on the filter performance. Our literature search revealed no incorporation of multiple performance criteria in both the time and the frequency domain.

This submission is novel because it incorporates time and frequency specifications simultaneously and it chooses to address only Step 1, filter approximation. This latter feature should make the problem easier for the EA to solve. It is general because the evolved filter approximation can be synthesized in any technology and used for analog or digital filters. It fits into the hierarchy of conventional filter design and can thus be integrated smoothly in industry design flow. It sets up potential development of an optimization tool that appropriately assists a human designer with evaluating tradeoffs between each design step.

The results of OptimFilt must be compared to state of the art which we consider to be [2, 10]. In [2, 10], multiple objectives in time and frequency domain were considered. This approach approximates closed form expressions for filter specifications that have no closed mathematical forms, for instance, peak overshoot. A non-linear, gradient-based optimization technique called Sequential Quadratic Programming (SQP) optimizes the objectives given the constraints. This approach suffers from the following disadvantages: a) error in the approximation may misguide the optimization, b) it is not extensible because it cannot deal with objectives and constraints for which no reasonably accurate mathematical approximations are available, c) SQP does not find a global optimum d) multiple objectives are aggregated and weighted to express a single objective, that precludes trade-off solutions. As is well recognized, EAs do not require closed form mathematical expressions or gradient information. We use Particle Swarm Optimization (PSO) [7] in OptimFilt, which is a population-based, simple to implement yet powerful algorithm for global optimization.

3. OptimFilt

We have designed OptimFilt to be able to explicitly trade-off the performance of a subset of filter criteria with another subset while being constrained with respect to yet another subset. For example, it could be asked to achieve maximal linearity in phase response in tradeoff with magnitude response with constraints on settling time value. To do this, OptimFilt maps the designer's filter specifications into a PSO problem formulation that turns the multiple specifications into design drivers (i.e. the knobs that a designer can control to search among design candidates). There are three elements in the problem formulation: 1) requirements, 2) objectives and 3) constraints.

OptimFilt must meet two requirements: causality and stability [13]. It partially enforces these requirements through its particle representation in the PSO. For a rational transfer function to be casual and stable, the order of the numerator needs to be less than that of the denominator and the particle representation enforces this. As well, poles should be in the left-hand s-plane. This implies all coefficients of the denominator must be positive. The PSO enforces this. Since these conditions are necessary but not sufficient, the settling time constraint (discussed in Section 3.2) also addresses stability.

An objective refers to the goal of the optimization. Presently, (with easy extension) OptimFilt can be directed to optimize the following objectives:

Phase Linearity Maximize linearity of phase response in the passband. (Linearity is simply one choice. Other arbitrary response can be defined).

Peak Overshoot Minimize peak overshoot [2] in time domain.

Magnitude Response Shape Meet a specified shape in magnitude response. For an ideal filter, this shape has brick-wall characteristics.

In 3.4 we explain how progress on these objectives is measured to determine their contribution to a particle's fitness.

The optimization takes place under one or more of the following constraints:

- ω_p, ω_s, A_{max} and A_{min}
- The settling time of the filter should be less than a given value. The settling time is the time taken by the filter to reach a stable value (10% of the stable value) on being excited by a step function.
- The quality factor of all poles [2] should be less than a given value (10 in our case).
- any unused objective: peak overshoot, phase linearity and error in magnitude response

A constraint is a specification that a solution must meet. In 3.2 we explain how these values are calculated and checked. The PSO uses a constraint as a basis for comparing two candidate solutions before it considers their relative performance on objectives.

3.1 Particle Swarm Optimization

In OptimFilt candidate transfer function coefficients are represented by a particle. A particle has $m + n$ dimensions, see Equation 1, because the highest order coefficient values in the denominator and numerator are unnecessary extra degrees of freedom for expressing the solution.

One particle in the initial generation is the exact coefficients of a conventional approximation, e.g. Butterworth. The remaining particles are statistical variants of the first. We vary each coefficient with a Gaussian distribution centered at the exact coefficient, with variance the maximum of 10 and the coefficient value. Negative coefficients of the denominator are discarded and redrawn. We multiply the variance by a so-called Diversity Factor (usually 1.0) to allow control over diversity in the initial population. The velocity update equation is:

$$v(i) = v(i-1) + c1 * r1 * (p_{pbest} - p(i)) + c2 * r2 * (p_{gbest} - p(i))$$

The position update equation is:

$$p(i+1) = p(i) + v(i)$$

where:

$v(i)$: Current velocity of the particle

$v(i-1)$: Velocity of the particle in last generation

$p(i)$: Current position of a particle

$p(i+1)$: Position of the particle in the next generation

p_{pbest}: The best position the particle has been

p_{gbest}: The best position the swarm has been

$c1$, $c2$: Parameters set to 2.

$r1$,$r2$: Two uniform random variables with values between 0 and 1.

If the position of any coefficient of the denominator becomes less than zero after an update, it is set to zero. The maximum velocity for each coefficient is set proportional to the maximum of 10 and the respective coefficient value in the conventional approximation used to seed the initial population. The proportionality constant used is henceforth called 'maximum velocity factor'.

3.2 Constraint Evaluation

To derive ω_s, ω_p, A_{max} and A_{min} of a particle, the corresponding filter is normalized to have a maximum magnitude response of 1. The minima in the passband and the maxima in the stopband are calculated numerically (with magnitude response sampled at points separated by 0.25 rads/s). The end of the stopband is set at 1000 rads/s and can be reset by the designer. A_{max} and A_{min} for the filter are calculated as per Equations 3 and 4 where g_{max} and g_{min} are the maximum and minimum gains in the stopband and passband respectively. These calculated values are compared to specifications to check if the constraint is violated or satisfied. A quantitative measure of constraint violation is calculated with Equation 5, where the value of $r(...)$ is 0 for input less than 0 and equal to the input when it is more than 0.

$$A_{max} = 20 * log10(1/g_{min}) \tag{3}$$

$$A_{min} = 20 * log10(1/g_{max}) \tag{4}$$

$$\begin{aligned} violation = \quad & r(particle(A_{max}) - spec(A_{max})) + && (5) \\ & r(spec(A_{min}) - particle(A_{min})) && (6) \end{aligned}$$

The time response for the normalized filter (with DC gain 1) is calculated numerically for a duration (henceforth called t_{tot}) slightly more than t_s (0.5s in our case). The maximum deviation of the time response from 1 between t_s and t_{tot} is calculated. If this deviation is more than 10%, the constraint is considered violated. The quantitative measure of this violation is the maximum deviation measured with respect to 1 in the given duration. The settling time constraint also expresses the stability requirement, since unstable candidates will not settle.

The quality factor, Q of a pole indicates whether the pole will lead to oscillations in the output. It is calculated for each pole as per [2] and if the value of Q for any pole is more than 10, the constraint is considered violated. The quantified violation is the sum of excess Q (Q-10) for each pole in violation.

3.3 Constraint Enforcement

OptimFilt enforces constraints by taking advantage of the fact that PSO updates the global best and particle best via a relative comparison not using an absolute metric. We have written an extended comparison algorithm that integrates consideration of constraints. Each constraint has a priority and is compared in terms of being satisfied or violated one by one. If the candidate and optima satisfy all constraints, they are compared on the objective function. The optima is replaced in the following situations.

1. When it violates a constraint while the candidate doesn't.

2. When both particles violate a constraint but the candidate's violation is less severe.

3. When all constraints of the candidate and optima are satisfied, the candidate's objectives are calculated, weighed and summed. The optima is replaced if its fitness on the objectives is less than that of the candidate's.

The priority for constraints in our case is A_{max} and A_{min} criteria, followed by settling time, Q-factor and any constraint on an unused objective. This scheme works in the following way. Each particle is guided to satisfy the constraints (according to the priority) irrespective of the objectives. The feasible areas are then explored for optimizing objectives irrespective of extent of constraint satisfaction.

3.4 Objective Evaluation

All the objectives are combined into a single fitness function. A lower fitness implies a more fit solution. The error with respect to different objectives is calculated in the following way.

Maximum Phase Linearity: The phase response of the candidate is evaluated in the passband and sampled at points separated by 1 rad/s. A best fit line for the phase response is found by linear regression. The squared error between the phase response and the best-fit line in passband forms the phase error, ph_e.

Minimum Peak Overshoot: The filter is normalized to have a DC gain (gain at 0 Hz) of 1. The time response for the filter is found numerically for a duration that is longer than the settling time constraint and is sampled every 5 milliseconds. The maximum value of the time response of the filter is called the peak value, p_v. The peak objective, pk_e is expressed by percentage peak overshoot with steady state value of 1:

$$pk_e = (p_v - 1) * 100$$

Specified Shape in Magnitude Response: The filter is normalized to have a maximum magnitude response of 1. The sum of squared error between the magnitude response of the solution and the specified magnitude response sampled at points (separated by 1 rad/s) gives the magnitude error, mr_e.

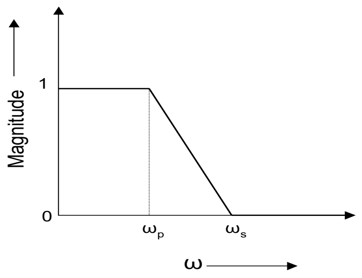

Figure 2: Ideal Magnitude Response Characteristics

In case of an ideal lowpass filter, the error is found with respect to characteristics as depicted in Figure 2. The response has a flat response with magnitude 1 in the passband, a flat response of 0 in stopband and a linear response in the transition band. The user can set the value of the highest frequency of interest, by default it is set to 100 rads/sec for squared magnitude error.

The fitness function is a weighted sum of these errors:

$$f(particle) = w_1 * pk_e + w_2 * mr_e + w_3 * ph_e$$

4. EVOLVING FILTER APPROXIMATIONS

We now present experiments that investigate OptimFilt's performance. We exercised and assessed OptimFilt with the filter specifications in Table 2. These specifications were chosen to allow comparison with [2].

Specification	Value
ω_p	20 rads/s
ω_s	30 rads/s
A_{max}	2.0475 rads/s
A_{min}	10.1728 rads/s

Table 2: Filter Specification for Experiments

We conducted 2 experiments: Experiment 1 has parts (a), (b), (c) and is shown in Table 4. Experiment 2 is a direct comparison in terms of constraints and objectives (in addition to specifications) to [2]. It is shown in Table 6. Each experiment started from an existing filter approximation with the minimum number of poles and zeroes. We varied the set of objectives and constraints in each to evaluate different aspects of OptimFilt. Each experiment consisted of 20 runs. We ran the PSO with the parameters given in Table 3.

Parameter	Value
c1	2
c2	2
Swarm size	50
Diversity Factor	1.0
Max velocity factor	0.5
Number of generations	600

Table 3: PSO parameters

4.1 Experiment 1

Experiment 1(a): The design requires minimization of peak overshoot given Table 2's specifications and settling time as constraints. The number of particles which violate Table 2's constraint in each generation is shown in Figure 3. This number gently decreases to about half the swarm size. The number of circuits violating settling time constraint quickly reduced to 0. Figure 4 shows the best individual of runs (BOR) with a decrease in peak overshoot of 90.1% (from 10.83% to 1.08%). Phase linearity was not constrained and it improved in comparison to Butterworth (ph_e changed from 193.48 to 60.19). A_{max} and A_{min} satisfied the constraints but changed from 1.357 and 10.172 to 2.047 and 10.358 respectively. The algorithm traded off magnitude response in passband for improvement in peak overshoot. Table 5 reports exact fitness results and the generation after which no substantial fitness improvement was observed.

It reports that the improvement in peak overshoot is very large (see BORF) and PSO is able to consistently achieve an improvement of more than 80% (with respect to initial coefficients) across multiple runs.

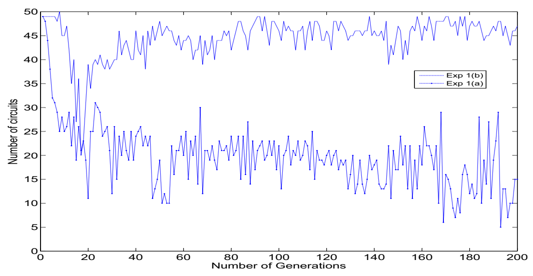

Figure 3: Exp. 1(a) and (b): Constraint violation

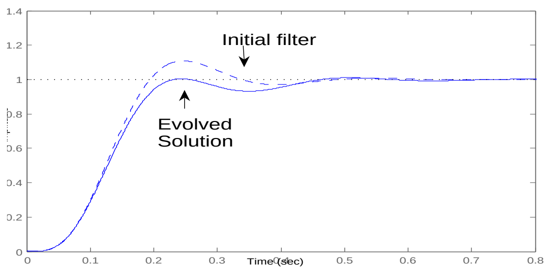

Figure 4: Exp. 1(a): Step Response of BOR and Butterworth filter. X-axis: Time(s), Y-axis: Magnitude

Experiment 1(b): The design requires maximal phase linearity given Table 2's specifications, with constraints on Q value and settling time. We choose an elliptic approximation as a starting point in this experiment. It is 2 orders less than Butterworth. As shown in Figure 3, the number of circuits violating Table 2's constraints in this experiment are much higher as compared to Exp. 1(a). This is expected since as one lowers the order, the feasible space meeting the constraint becomes smaller and finally non-existent below a certain order. However, it is commendable that with only a few circuits in each generation satisfying the constraint, the PSO is still able to optimize well.

Figure 5 compares the phase response of the best of run solution with the elliptic filter (an improvement of 92% in ph_e). What is interesting is the fact that the solution dominates elliptic filter on all objectives considered, decreasing pk_e by 63%, decreasing m_e by 29% and pushing A_{max} further inside the constraint range. However, the maximum frequency of interest considered by our experiment is 1000 rads/s and the filter may fall out of specifications out of this range. Assuming this is indeed the range, such a result maybe extremely useful in a practical setting.

Experiment 1(c): The design requires maximal phase linearity given Table 2's specifications, peak overshoot and settling time as constraints. The initial approximation used is Chebyshev which has a peak overshoot of 3.22%. We constrained the peak overshoot to be less than 2.00%. The diversity factor was set to 2, so that PSO could explore areas far away from Chebyshev. The best of run solution improves phase linearity by 95% (initially 1611.30) and meets the peak overshoot constraint. It trades-off magnitude response in stopband to achieve this. PSO consistently improves phase linearity by an amount more than 80% across runs. This experiments suggests how OptimFilt can be used to optimize on a criteria given a hard-specification by the user. This is in contrast with the approach to optimize on a criteria and later check if the final solution meets the hard-specification. Results are shown in the appropriate column of Table 5.

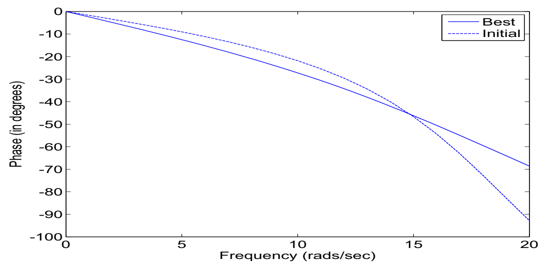

Figure 5: Exp. 1(b): Phase Response of BOR and Elliptic filter

Exp	**1(a)**	**1(b)**	**1(c)**
Approx	Butterworth	Elliptic	Chebyshev
P,Z	4, 0	2, 2	3, 0
Constr 1	Table 2	Table 2	Table 2
Constr 2	Settling Time	Settling Time	Settling Time
Constr 3			Peak Overshoot
Obj 1	Peak Overshoot	Phase Linearity	Phase Linearity

Table 4: Definition of Exp. 1(a) to 1(c). (P,Z: Number of Poles, Zeros)

Experiment	**1(a)**	**1(b)**	**1(c)**
Avg BF	1.28 (0.13)	119.84 (4.81)	175.0 (282.2)
BORF	1.08	117.02	70.29
GC	102.5 (149.6)	420.05 (86.38)	210.6 (140.2)
t_{EVAL}	0.032	0.026	0.036
Init Fit	10.83	1499.50	1611.30

Table 5: Results of Exp. 1(a) to 1(c), Avg BF: Average Best Individual Fitness, BORF: Best of Runs Fitness, GC: Generations to converge, t_{EVAL}: average time (sec) to evaluate objectives and constraints of one individual (On a Pentium M, 2.13GHz Windows machine), Init Fit: fitness of conventional approximation.

4.2 Experiment 2

Experiment 2, see Table 6, permits direct comparison with Experiment 1 in [2] (henceforth called SQP solution). The design goals are to minimize peak overshoot and phase linearity with Table 2's specifications, Q value and settling time as constraints. The fitness function is an weighted sum of the two objectives: percentage peak overshoot is weighted by 0.75 and phase linearity by 1.0.

Out of 20 runs, 4 OptimFilt solutions dominated the SQP solution and had better performance in terms of both peak overshoot and phase linearity. Fourteen solutions dominated the SQP solution in percentage peak overshoot, but were worse on phase linearity. There were 2 solutions which were completely dominated by the SQP solution. Figure 6 shows the time domain response of the Butterworth, OptimFilt's best of run and SQP solution. OptimFilt's best of run solution improved ph_e by 13.66% and pk_e by 52.1%.

This experiment shows that PSO can find better solutions than SQP. This highlights a compelling feature of evolutionary algorithms in contrast with gradient based techniques which get stuck at the local optima. It is encouraging to see that PSO can beat the SQP solution 20% of the time and we are optimistic that this rate can be improved. An alternative is to have a tool which runs PSO multiple times to find the optimal solution.

The transfer function for BOR for Exp. 1(a),(b),(c) and 2 are stated below in order:

$$\frac{1}{s^4 + 73.996s^3 + 1923.174s^2 + 40787.676s + 300958.583}$$

$$\frac{0.309s^2 + 2.311s + 526.765}{1.000s^2 + 24.839s + 531.585}$$

$$\frac{1}{s^3 + 25.553s^2 + 683.703s + 5721.031}$$

$$\frac{1}{s^4 + 54.346s^3 + 1512.278s^2 + 28820.945s + 214001.244}$$

Experiment	**2**
Approximation	Butterworth
Poles, Zeros	4, 0
Objective 1	Phase Linearity
Objective 2	Peak Overshoot
Constraint 1	Table 2
Constraint 2	Settling Time
Constraint 3	Q Value

Table 6: Definition of Experiment 2.

5. SUMMARY

In this paper we have described a system named OptimFilt that evolves filter approximations in the form of coefficients of a transfer function.

The key feature of OptimFilt in terms of being a useful design tool is that it can take all the specifications in the initial step of filter design into account in producing an approximation. In conventional filter design, only a few of a filter's specifications are quantitatively specified and the designer has no control over the rest. OptimFilt provides quantitative control knobs to a designer for multiple performance specifications. It offers a filter designer the choice of translating a specification into a constraint or an objective.

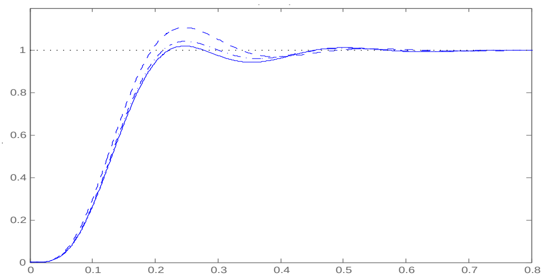

Figure 6: Exp. 2: Step Response for Butterworth (- -), SQP (-.) and OptimFilt's BOR (line). X-axis: time(s), Y-axis: Magnitude

It further allows constraints to be placed on objectives. Using the tool, one can design, for example, a filter for a given order, A_{max}, and A_{min}, which is monotonic, but sacrifices maximum flatness to get higher phase linearity given a constraint on the maximum peak overshoot. Such design flexibility is not achievable by contemporary filter approximation practices.

We showed that PSO is able to uncover interesting portions of the design space which remained hidden in conventional filter design. In contrast to SQP, OptimFilt offers independence from inaccurate mathematical models that can mislead a search for the optima. We have experimented with OptimFilt and the experiments documents its effectiveness. It is capable of evolving a solution that is better than the SQP state of the art method.

6. FUTURE VISION AND PRELIMINARY EXPERIMENTS

We want to recast the problem of filter approximation as Performance Space Modeling [3]. Consider that a designer wants a filter approximation with a given A_{max}, A_{min}, w_s, w_p specification and objectives of maximizing phase linearity and minimizing peak-overshoot. The designer is unable to decide the order of the filter since he doesn't have a concrete idea of how the order influences the performance of the filter. We can help the designer in making this decision by telling him what performance objectives are achievable for a given order apriori. As his design flow proceeds top-down, performance information flows bottom-up to inform his choice of order. We shall provide him two pieces of information.

- For a given order, what are the trade-off solutions given the multiple objectives.
- How these trade-off solutions change with respect to order.

Thus, the designer can make an informed decision whether to use a higher order which implies higher power and area, but results in improvement on objectives. We use constraint-based multi-objective PSO to enumerate a front of pareto-optimal solutions for each order in range. The designer can then pick the order and from among its tradeoff points, a design meeting the objective.

For preliminary experiments, we coded constraint based multi-objective PSO as given in [14] with slight modifica-

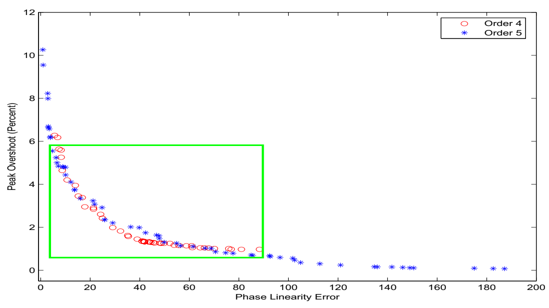

Figure 7: Performance Space for an all-pole filter

tion. We used the same filter design problem as in Experiment 2. All parameters of PSO are as before, except the following: Swarm Size: 100, Size of Repository: 50 (see [14]) and Number of Generations: 300.

The pareto-optimal solutions for filters of the orders 4 and 5 generated by combining the pareto-optimal set of several runs is shown in Figure 7. If the designer wants a design in the trade-off space bounded by the rectangle, there is no advantage to using a higher order. However, a higher order would be advantageous if the design is in the space outside the rectangle. Now the order decision can be made comfortably while knowing a quantitative measure of performance improvement.

Design of reliable algorithms for multi-objective optimization is underway. We will address real-world case studies to asses the usefulness and performance of the algorithms.

7. REFERENCES

[1] T. Arslan and D. Horrocks. A genetic algorithm for the design of finite word length arbitrary response cascaded IIR digital filters. In *GALESIA. First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 276–281. IEE Conf. Publ No. 414, 1995.

[2] N. Damera-Venkata and B. L. Evans. An automated framework for multicriteria optimization of analog filter designs. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(8):981–990, Aug. 1999.

[3] G. G. E. Gielen, T. McConaghy, and T. Eeckelaert. Performance space modeling for hierarchical synthesis of analog integrated circuits. In W. H. J. Jr., G. Martin, and A. B. Kahng, editors, *DAC*, pages 881–886. ACM, 2005.

[4] C. Goh and Y. Li. GA automated design and synthesis of analog circuits with practical constraints. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 170–177. IEEE Press, 2001.

[5] J. Grimbleby. Automatic analogue circuit synthesis using genetic algorithms. *IEE Proceedings Circuits, Devices and Systems*, 147(6):319–323, Dec. 2000.

[6] J. B. Grimbleby. An automatic analogue network synthesis using genetic algorithms. In A. M. S. Zalzala, editor, *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA*, volume 414, pages 53–58, Sheffield, UK, 12-14 Sept. 1995. IEE.

[7] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the Fourth IEEE International Conference on Neural Networks*. IEEE Press, 1995.

[8] J. R. Koza, F. H. Bennett III, D. Andre, M. A. Keane, and F. Dunlap. Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE Transactions on Evolutionary Computation*, 1(2):109–128, July 1997.

[9] J. D. Lohn and S. P. Colombano. A circuit representation techniqueor automated circuit design. *IEEE-EC*, 3(3):205, September 1999.

[10] M. Lutovac, D. Tosic, and B. Evans. *Filter Design for Signal Processing using MATLAB and Mathematica*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2001.

[11] M. Murakawa, T. Adachi, Y. Niino, Y. Kasai, E. Takahashi, K. Takasuka, and T. Higuchi. An AI-calibrated IF filter: a yield enhancement method with area and power dissipation reductions. *IEEE Journal of Solid-State Circuits*, 38(3):495–502, 2003.

[12] M. Murakawa, S. Yoshizawa, T. Adachi, S. Suzuki, K.Takasuka, M. Iwata, and T. Higuchi. Analog EHW chip for intermediate frequency filters. In *Proc. of the 2nd Int. Conf. on Evolvable Systems*, pages 295–298. IEEE Press, 1998.

[13] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab. *Signals & systems (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[14] C. R. Raquel and J. Prospero C. Naval. An effective use of crowding distance in multiobjective particle swarm optimization. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 257–264, New York, NY, USA, 2005. ACM Press.

[15] A. S. Sedra and K. C. Smith. *Microelectronic circuits, 2nd ed.* Holt, Rinehart & Winston, Austin, TX, USA, 1987.

[16] P. Vieira, L. Sa, J. Botelho, , and A. Mesquita. Evolutionary synthesis of analog circuits using only mos transistors. In *2004 NASA/DoD Conference on Evolvable Hardware*, pages 38–45, Chicago, Illinois, 24-26 June 2004. NASA Ames Research Center, IEEE Computer Society.

[17] J. Vondras and P. Martinek. Multi-criterion filter design via differential evolution method for function minimization. In *Proceedings. ICCSC '02. 1st IEEE International Conference on Circuits and Systems for Communications*, pages 106–109, 2002.

[18] S. Zhao, L. Jiao, J. Zhao, and Y. Wang. Evolutionary design of analog circuits with a uniform-design based multi-objective adaptive genetic algorithm. In *2005 NASA/DoD Conference on Evolvable Hardware*, pages 26–29, Chicago, Illinois, 2005. NASA Ames Research Center, IEEE Computer Society.