Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2013, Article ID 741251, 14 pages
http://dx.doi.org/10.1155/2013/741251

Research Article

Holistic Biquadratic IIR Filter Design for Communication Systems Using Differential Evolution

Alexander Melzer,[1,2] Andreas Pedross,[2] and Manfred Mücke[3]

[1] *Maxim Integrated GmbH, 8403 Lebring, Austria*
[2] *Graz University of Technology, Signal Processing and Speech Communication Laboratory, 8010 Graz, Austria*
[3] *Sustainable Computing Research, 1040 Vienna, Austria*

Correspondence should be addressed to Manfred Mücke; manfred.muecke@scr.or.at

Received 11 December 2012; Revised 4 March 2013; Accepted 19 March 2013

Academic Editor: Oscar Gustafsson

Digital IIR filter implementations are important building blocks of most communication systems. The chosen number format (fixed-point, floating-point; precision) has a major impact on achievable performance and implementation cost. Typically, filter design for communication systems is based on filter specifications in the frequency domain. We consider IIR filter design as an integral part of communication system optimisation with implicit filter specification in the *time domain* (via symbol/bit error rate). We present a holistic design flow with the system's bit error rate as the main objective. We consider a discrete search space spanned by the quantised filter coefficients. *Differential Evolution* is used for efficient sampling of this huge finite design space. We present communication system performance (based on bit-true simulations) and both measured and estimated receiver IIR chip areas. The results show that very small number formats are acceptable for complex filters and that the choice between fixed-point and floating-point number formats is nontrivial if precision is a free parameter.

1. Introduction

In signal processing, filters are building blocks removing unwanted signal components (often, but not exclusively, specified in the frequency domain). Design (i.e., identification of suitable structures and coefficients) of digital filters given some specification in the frequency domain (e.g., pass-band ripple, cutoff frequency, stop-band frequency, or stop-band attenuation) is a well-established field [1]. In many advanced systems, however, the ultimate goal is rarely specifiable in the frequency domain but is reflected in a more complex measure. Examples include bit error rate, peak power consumption, or power trace entropy. Designing efficient systems therefore requires embedding of the filter design process in a larger system design context. Implementation of digital filters requires identification of arithmetic units to be implemented, the choice of a specific number format for each arithmetic unit, and quantisation of filter coefficients. These actions alter the filter characteristics and, if not foreseen in the design process, can have a severe impact on the system's performance.

There is a vast literature on digital filter design and optimisation of resource usage under some constraints. All design approaches we are aware of have in common that the number format's underlying error model has to be chosen in advance; that is, the designer has to take the decision if fixed-point or floating-point arithmetic is used prior to optimisation. Optimisation then identifies acceptable filter coefficients. Some systems allow for estimating minimally required precision.

We hypothesise that both the choice of the best error model (i.e., if fixed-point or floating-point arithmetic allows for the most efficient implementation) and identification of quantised filter coefficients are nontrivial and do not lend themselves to direct specification. In contrast to current practice, identification of number format and filter coefficients should be included in the communication system design process which should be seen as an integrated optimisation process. To verify our hypothesis, we developed a bit-true time domain simulation of a prototypical communication system including an IIR receiver filter. This simulation is used

for cost function evaluation in a global optimisation algorithm directly searching for quantised filter coefficients. The central requirement of a communication system is considered achieving some target bit error rate (BER) while minimising implementation costs (like chip area and power consumption). We employ Differential Evolution (DE) [2] as a direct search method to identify the most efficient implementation under varying constraints.

The scientific contributions of our work are

(i) considering IIR filter design in the context of communication systems (i.e., BER-guided filter coefficient identification);

(ii) providing a bit-accurate simulation framework allowing for evaluation of the objective function;

(iii) employing DE to the previous process;

(iv) considering the previous process without prior choice of number format (fixed- or floating-point; precision);

(v) identification of minimum required precision as a function of system constraints;

(vi) extending the objective function by implementation specific measures allowing for ASIC- or FPGA-specific ranking of BER-equivalent filter realisations.

The paper is structured as follows. Section 2 reviews related work. Section 3 introduces a prototypical communication system and related measures. Section 4 details the general IIR filter design process. It recapitulates the idea of formulating digital filter design as an optimisation problem. Filter realisation is assumed to be a cascade of biquadratic filter stages. Section 5 presents the DE algorithm as a means to quickly sample a huge multimodal search space. Section 6 considers the problem of filter design under the objective of optimising a communication system's performance and details the use of DE to directly search the resulting design space. Section 7 describes in detail all implementation and choices made for performing experiments. Section 8 reports results of different experiments employing DE to search directly for the most suitable IIR filter implementation minimising BER of the communication system considered. Giving an acceptable BER allows to search for the minimal acceptable precision of floating-point and fixed-point implementations. Further experiments extend the objective function by measures relevant to the filter's implementation cost. Section 9 discusses the experiment's results. Section 10 summarises the work and discusses challenges ahead. Section 11 details possible future research directions.

2. Related Work

Many authors have described strategies to derive filter coefficients for some given filter structure such that a given frequency response is matched. Usually, quantisation of coefficients is considered as a distinct and subsequent task.

Shyu and Lin [3] employ a variational approach to identify quantised coefficients by trying to match an ideal impulse response. After initial identification of continuous coefficients, they are iteratively substituted by quantised ones while the remaining coefficients are modified to compensate for the resulting modification.

Alternatively, direct derivation of quantised filter coefficients can be done by considering filter design a search problem over a huge (yet finite) discrete space. Storn [4, 5] used differential evolution to search for realisable filter coefficients using specifications in the frequency domain. Ramos and López [6] combine a direct search method with a heuristic parametric optimization to identify optimal realisable filter coefficients of an equaliser structure for loudspeakers implemented using biquadratic filter stages.

There are several optimisation algorithms available that tackle this issue. Especially the use of population-based computation algorithms such as genetic algorithms [7–9], differential evolution [2, 9–12], particle swarm optimisation [13], and the seeker optimization algorithm [14] is common.

Irrespective of the method employed, many different objectives for guiding the filter design process are possible. Typically, filters are specified in the frequency domain. Consequently, most filter design methods employ an objective function derived from filter specifications in frequency domain (like the integral over the frequency response mismatch).

The matched filter theory readily provides optimal filter specification in frequency domain relevant under communication system constraints. The matched filter theory relies on linear system characteristics, though. This assumption is violated many times for real-world systems. Stücke et al. [15] investigate nonlinear effects in communication systems and their impact on ZigBee transceiver systems in terms of BER. To the best of our knowledge, our work is the first one employing a BER objective function in a direct search approach, basically solving the inverted problem compared to the work of Stücke et al. A related approach for optimising transmitter and receiver FIR filter specifications under BER constraints in a MIMO setting is presented by Hjorungnes et al. [16].

The DE algorithm was presented by Storn in 1995 in a technical report followed by a paper authored by Storn and Price in 1997 [2]. Design of digital filters was one of the original motivations for the development of the algorithm. In 2005, DE was presented in the IEEE Signal Processing Magazine [4]. The up-to-date references for DE are the books by Price et al. [17] and Chakraborty [18] as well as the recent survey by Das and Suganthan [19]. Compared to the original schemes suggested by Storn and Price [2], many additional variants have been conceived and methods for selecting parameters have been described. Notably, Zhu et al. [20] propose a method to adapt the (usually static) population size, while Pan [21] suggests a scheme where the DE parameters F and λ are adapted during optimisation. Zaharie [22] investigates the effect of different crossover schemes.

Pan [21] applies DE for pole-placement design, while Zhu et al. [20] apply DE for IIR coefficient identification in the context of system identification.

Most current design methods do not perform a rigour verification of the actual filter being implemented, that is, taking all quantisation effects into account.

Widrow and Kollár [23] have suggested a generic method for modeling the probability density functions of intermediate results yielded by limited-precision arithmetic and have employed it to (both arithmetic and coefficient) error analysis of IIRs. They point out, however, that the method becomes more unreliable with decreasing precision. Cox et al. [24] present bit-precise verification techniques and show cases where reasoning on the bit level is necessary. Kar [25] presents a stability criterion for digital filters considering both quantisation and overflow effects.

While many methods derive filter coefficients in double-precision floating point, this is a very costly number format to be implemented in hardware. Hardware implementation of digital filters typically resorts to low-precision fixed-point arithmetic as the respective operations are considered less expensive in terms of time-area product. Very few authors have investigated low-precision floating-point number formats for the implementation of digital filters.

To the best of our knowledge, our work is the first one considering floating-point precision a free parameter on a per-bit resolution in the IIR design process.

Our work is the first in-depth comparison between respective implementation costs of fixed-point and floating-point custom-precision IIR filter implementations.

3. System Model

This section presents the communication system which will be used throughout the rest of this paper. The communication system is a base-band transceiver concept as illustrated in Figure 1. Without loss of generality, the system is modeled in discrete-time. The signaling under consideration is a one-dimensional pulse-amplitude modulation (PAM). We assume absence of intersymbol interference (ISI). Therefore, only a single symbol period of duration N_{sym} needs to be considered. The sent signal $s[n]$ is therefore given as

$$s[n] = a * h_{\text{T}}[n], \tag{1}$$

where $a \in \{a_l\}$ is the transmitted symbol, $\{a_l\}$ is the set of L available complex valued symbols, $h_{\text{T}}[n]$ is the pulse shape, and $*$ denotes to the convolution operator.

For simplicity, the transmission channel is modeled using additive white Gaussian noise (AWGN). The received signal $r[n]$ is therefore given as

$$r[n] = s[n] + \nu[n], \tag{2}$$

where $\nu[n]$ is a white Gaussian noise process with a power spectrum density of $N_0/2$.

To remove out-of-band noise at the receiving side, a receiver filter with impulse response $h_{\text{R}}[n]$ is required. The receiver filter's output is given as

$$y[n] = h_{\text{R}}[n] * s[n] + h_{\text{R}}[n] * \nu[n]. \tag{3}$$

It is known that the optimal (i.e., minimising detection errors) receiver filter for AWGN channels and linear signaling

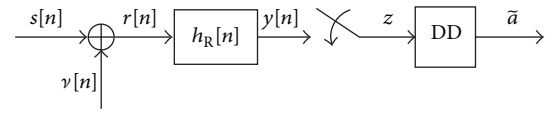


Figure 1: Discrete-time domain communication system model.

schemes is the matched filter [26]. Therefore, $h_{\text{R}}[n] = h_{\text{T}}^*[-n]$, where $(\cdot)^*$ denotes the complex conjugate. It should be noted that the matched filter gives a lower bound with respect to the bit error rate [26]. Any approximation of this matched filter can only perform worse. As shown later, direct search (using DE) can find a filter solution close to the theoretical matched filter BER performance.

After filtering, the signal $y[n]$ is sampled, which results into the decision variable z. The ideal sample time is at the end of the symbol period; hence, the decision variable z is given as

$$\begin{aligned} z &= y\left[N_{\text{sym}}\right] \\ &= s + \nu, \end{aligned} \tag{4}$$

where s is the projection of the transmit pulse $h_{\text{T}}[n]$ onto the receiver filter $h_{\text{R}}[n]$ scaled by the transmitted symbol s, and thus

$$s = a \sum_{k=-\infty}^{+\infty} h_{\text{T}}[k]\, h_{\text{R}}[k], \tag{5}$$

and ν is a zero-mean Gaussian distributed random variable with variance $\sigma_\nu^2 = (N_0/2) \sum_{k=-\infty}^{+\infty} |h_{\text{R}}[k]|^2$. Note that for the case that the receiver's impulse response $h_{\text{R}}[n]$ does have unit energy, hence, $\sum_{k=-\infty}^{+\infty} |h_{\text{R}}[k]|^2 = 1$, and the variance σ_ν^2 of the random variable ν is equal to the noise power spectral density $N_0/2$ of the noise sequence $\nu[n]$ so that $\sigma_\nu^2 = N_0/2$. Note that s is given as $s \in \{a_l\}$ if $h_{\text{T}}[n]$ with unit energy and $h_{\text{R}}[n]$ represents the corresponding unit energy matched filter.

For detection, the decision device (DD) compares the decision variable z with a set of thresholds $\{\gamma_l\}$. The interval between two threshold values γ_l and γ_{l+1} defines the range of the symbol a_l. If z is in the interval of the two threshold values γ_l and γ_{l+1}, the decision device decides for $\tilde{a} = a_l$.

For a binary antipodal one-dimensional PAM, for example, binary phase-shift-keying (BPSK), the symbols a are given as $a \in \{-\sqrt{\mathscr{E}_b}, +\sqrt{\mathscr{E}_b}\}$, where $\mathscr{E}_b$ denotes the bit energy for a unit energy pulse shape. The ratio $\mathscr{E}_b/N_0$ is a well-known metric for comparing the performance of communication systems. The theoretical bit error probability P_e for the optimum matched filter receiver of a binary antipodal PAM is given as

$$P_e = \text{Q}\left\{\sqrt{\frac{2\mathscr{E}_b}{N_0}}\right\}, \tag{6}$$

where $\text{Q}\{\cdot\}$ is the Q-function. As discussed before, this theoretical bit error probability represents a lower bound to technically achievable (i.e., approximating $h_{\text{R}}[n]$) BERs.

4. IIR Filter Design

Infinite impulse response filters provide comparable performance at less computational/hardware requirements compared to finite impulse response (FIR) filters. The disadvantage of IIR filters is the fact that they can become unstable and that the error surface for finite-precision implementations is nonlinear and multimodal [10]. The latter renders finding an optimal set of quantised filter coefficients for implementation in finite precision arithmetic a hard problem.

Due to the relevance of digital filters in general and the resource savings possible through use of IIRs, there exists a high interest in converting continuous into quantised filter coefficients. Consequently, many different respective methods have evolved. Matching a certain frequency response specification is the major target of most of them. However, there are also interesting approaches that design the filter in the time domain. The latter is used in this work as it has several advantages for the design of digital filters used in communication systems.

Especially for IIR filters designed to meet certain filter specifications such as pass-band ripple, cutoff frequency, stop-band frequency, or stop-band attenuation, it is common to adjust poles and zeros in the z-plane as this automatically ensures stability [27]. However, for a digital implementation, coefficients still need to be calculated out of the poles/zeros and quantised. Furthermore, depending on the filter structure, scaling factors that are used to limit the numerical range for finite-wordlength architectures do not come out from such designs in the z-plane. That is why, in contrast to [27], we adjust the coefficients of the filter directly in order to avoid this issue. To verify stability of these coefficients, poles and zeros need to be computed. Verifying location of poles and zeros is a simple operation with low computational effort, though.

Designing the filter in time domain can be incorporated by trying to match, for instance, a reference impulse response. Besides the fact that the evaluation is simple, quantisation errors can be evaluated convenient in time domain. Furthermore, computations of the impulse response can be done bit-true such that the simulation corresponds exactly to the final hardware implementation.

Instead of taking the impulse response, we generate a random data sequence as reference out of the communication system model (Figure 1) and aim at to minimising the bit error rate.

4.1. Filter Structure. There exist many possible filter structures to implement a given transfer function. The cascading of second-order (or biquadratic) filter sections is beneficial as the number range is limited by the scale values in front of each section. Therewith, even quantisation errors are reduced [28] which is the main purpose of utilizing SOS direct-form II filter structures in this work (see Figure 2). Alternative filter structures include lattice wave filters [29, 30], but there is the issue that in general this filter structure needs the double amount of multipliers compared to direct-form II implementations [1].

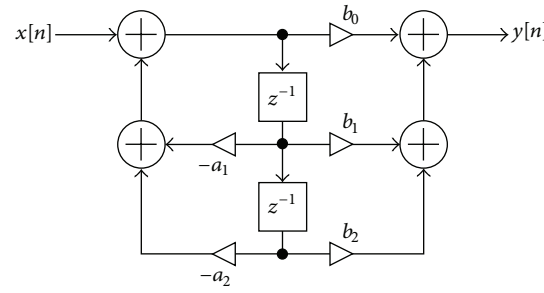


Figure 2: Direct-form 2 implementation of a biquadratic (or second-order) filter stage.

The transfer function of a cascaded SOS is given as

$$H(z) = \prod_{k=1}^{N_s} s_k \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 - a_{1k}z^{-1} + a_{2k}z^{-2}}, \tag{7}$$

where N_s is the number of sections, s_k is the scale value for each SOS, and the filter coefficients are represented by b_{0k}, b_{1k}, b_{2k}, a_{1k}, and a_{2k}. Note that there is full freedom in ordering the sections itself.

Starting at the entry of an SOS, the scale value is a useful factor for limiting the numerical range of finite-precision arithmetic; thus, overflows can be suppressed in an elegant way. Furthermore, this scale value can be used to weight the SOS itself. As each SOS is a filter of order two, quantisation errors are mainly influencing one section. The impact on the overall filter transfer function is reduced as the signal is damped and the noise bandlimited by subsequent sections. Finally, the number of delay elements is decreased compared to direct-form I sections [1, 12, 28].

4.2. Filter Evaluation. The typical quality measure employed to quantify how well a filter matches the system specification is the mean squared error (MSE) between original specification and actual filter characteristic. For demonstration, we compare in the following the magnitude response of the matched filter with the one of the considered standard filter design approaches given the matched filter magnitude response as design objective.

For the transceiver system model presented in Section 3 with root-raised-cosine shaped pulses, the matched filter is a lowpass. We design three standard lowpass filter types to demonstrate the achievable match of magnitude responses. The filter types are Butterworth, Elliptic, and Chebyshev type I [1]. All filters are designed with order 6. A simulation is done exemplary for a binary antipodal PAM. The optimum BER is obtained by varying the cut-off frequency f_c between $0.05\,\pi$ and $0.1\,\pi$ in steps of $0.005\,\pi$. These parameters are selected by evaluating the magnitude response of the pulse shaping filter $h_T[k]$. The reference filters are designed to match it as close as possible (Figure 11). Table 1 lists the identified best cut-off frequency f_c for each filter type together with the respective MSE compared to the matched filter's magnitude response. The Butterworth filter performs best; that is, its magnitude response is closest to the matched filter's

Table 1: Best cut-off frequency f_c and corresponding MSE to the matched filter's magnitude's response.

Filter type	f_c normalised frequency	MSE (dB)
Butterworth	0.07π	−10.63
Elliptic	0.075π	−8.83
Chebyshev	0.075π	−8.61

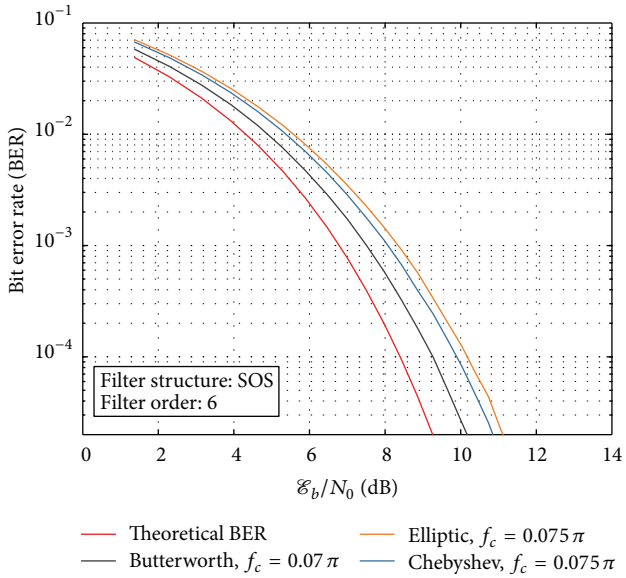


Figure 3: Transceiver BER for a Butterworth, elliptic, and Chebyshev filters based on the exemplary communication system model (Figure 1). All filters are designed with order 6. The simulation is done for $N_b = 10^6$ bits.

one. As illustrated in Figure 3, the theoretical BER given in (6) is not reached.

4.3. Discrete Design Space. Given some filter structure implemented using finite precision arithmetic, the set of possible filter coefficients (and therefore filter characteristics) can be enumerated. This set represents a huge, yet finite design space. Given a sixth-order IIR filter implemented as a cascade of biquadratic filter stages, every possible implementation can be described by an instance of the SOS matrix **C**, representing the filter coefficients where

$$\mathbf{C} = \begin{bmatrix} b_{01} & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & 1 & a_{12} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{0N_s} & b_{1N_s} & b_{2N_s} & 1 & a_{1N_s} & a_{2N_s} \end{bmatrix} \tag{8}$$

and the scale vector **s**

$$\mathbf{s} = \left[s_1, s_2, \ldots, s_{N_s}\right]^{\mathrm{T}}, \tag{9}$$

where N_s is the number of biquadratic filter stages.

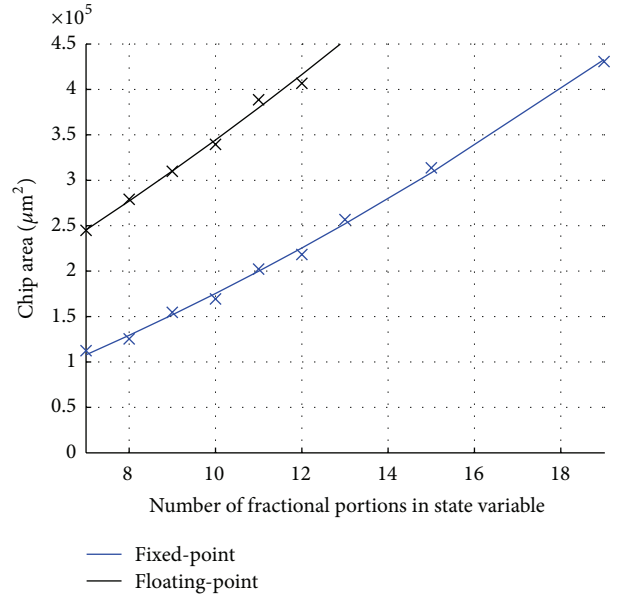


Figure 4: Chip area for programmable coefficients of the IIR filter with three biquadratic filter stages for fixed- and floating-point.

4.4. Arithmetic. Besides the filter architecture, the arithmetic in which the computations are performed in digital hardware is a major design decision. Especially the number of fractional bits for fixed-point and the precision for floating-point implementations is a major design decision as a tradeoff between performance and hardware costs needs to be found. This will be evaluated in detail in Section 7.4.

4.5. Implementation Cost-Chip Area. Digital filters are complex structures requiring a significant amount of chip area for implementation. We have developed a generic code describing the cascaded filter structure in both fixed- and floating-point with the precision as a parameter. There exist two versions. One integrates the coefficients directly into the design while the other one implements a register. The latter option allows for arbitrary changes of parameters through reprogramming of the registers. All multiplier paths, however, need to be fully instantiated, resulting in a much higher area requirement compared to predefined coefficients.

Synthesis results for different floating-point and fixed-point number formats are given in Table 2 and depicted in Figure 4.

4.6. Limit Cycle. An IIR filter can exhibit an unstable behavior under finite precision arithmetic for specific constant input signals. This type of instability usually results in an oscillatory periodic output called a *limit cycle* [31].

There are two types of limit cycles: (1) granular limit cycle (LSB oscillations; typically of low amplitude) and (2) overflow limit cycle (MSB oscillations; typically of large amplitude).

Criteria considering the combined effect of these two phenomena have only recently started to appear [25].

Many filter design methods aim at reducing or avoiding limit cycle effects. Our chosen filter realisation (cascaded

Table 2: Chip area for an IIR filter (order 6, cascade of three biquadratic filter stages) with programmable coefficients.

Fractional bits (bit)	Area fixed-point (μm^2)	Area floating-point (μm^2)
7	112370	244905
8	125379	279039
9	154582	309891
10	169388	339441
11	202143	388644
12	218251	406734
13	256645	458154
15	313606	533592
19	430632	705147

Table 3: Evolutionary computation nomenclature (from [32]).

Individual	A candidate solution
Child, parent	A child is the tweaked copy of its parent
Population	Set of candidate solutions
Fitness	Quality, directly related to the cost function
Selection	Picking individuals based on their fitness
Mutation	Plain tweaking of parameters based on population

biquadratic DF II filter) is known to limit the possible effects of limit cycles but does not exclude them. The current design flow does not check for possible limit cycles.

5. Differential Evolution

The Differential Evolution (DE) optimisation algorithm is proposed in [2]. Since it shows excellent performance compared to other much more complex heuristic optimisation algorithms [19], we utilize this algorithm. As described in Section 2, many variants and improvements of the original schemes have been conceived. As the task at hand is performed offline and the basic scheme does deliver good results, we have not explored further opportunities for tweaking the DE scheme. We discuss future improvements in Section 11. In the following, the DE algorithm's steps are described in detail. As evolutionary computation methods have different naming conventions, a short nomenclature is provided in Table 3.

(1) Initialization. The following constants need to be chosen. N_s is the number of biquadratic filter stages, P is the population size, and G the maximum number of generations. Furthermore, the number format (fixed- or floating-point) and the respective precision needs to be selected.

The initial population of individuals is randomly generated. One individual represents the scale vector $\mathbf{s}$ and the SOS matrix $\mathbf{C}$. The scales are generated as

$$s_i = \mathcal{U}(0, 0.1) \quad \forall i = 1, \ldots, N_s, \tag{10}$$

where $\mathcal{U}(a, b)$ denotes the continuous uniform distribution in between a and b. The coefficient matrix is initialized as

$$C_{ij} = \mathcal{U}(-1, 1) \quad \forall i = 0, \ldots, 5, \; j = 1, \ldots, N_s. \tag{11}$$

In general, the number range of coefficients and scales of the initial population is of little importance. We found that the algorithm finds a satisfying solution with any random initialization.

(2) Scales and Coefficient Quantisation (Optional). In case of optimisation for bit-true filter coefficients (i.e., fixed- or floating-point), the scales and coefficients are quantised to their predefined number format.

(3) Compute and Evaluate the Cost Function. A system simulation is performed for all individual in the current generation (i.e., $s_{i,g}$ and $\mathbf{C}_{i,g}$ $\forall i = 1, \ldots, P$) and the BER computed at a given operating point according to (16). Depending on the objective function used, additional criteria are evaluated to arrive at the final fitness of each individual.

(4) Mutation and Selection. There are two basic schemes for mutation suggested in [2] (DE1 and DE2). DE1 performs mutation relying on a random difference vector. The difference vector is built using two randomly selected individuals (indices $i, \xi, \eta \in \{1, \ldots, P\}$ and mutually different). The scales and coefficients are tweaked according to

$$\begin{aligned} s_{i,g+1} &= s_{i,g} + F \cdot \left(s_{\xi,g} - s_{\eta,g}\right) \quad \forall i = 1, \ldots, P, \\ \mathbf{C}_{i,g+1} &= \mathbf{C}_{i,g} + F \cdot \left(\mathbf{C}_{\xi,g} - \mathbf{C}_{\eta,g}\right) \quad \forall i = 1, \ldots, P. \end{aligned} \tag{12}$$

DE2 performs mutation as DE1 but takes the best solution in the current generation into account, too. The scales and coefficients are tweaked according to

$$\begin{aligned} s_{i,g+1} &= s_{i,g} + \lambda \cdot \left(s_{\text{best},g} - s_{i,g}\right) + F \cdot \left(s_{\xi,g} - s_{\eta,g}\right), \\ \mathbf{C}_{i,g+1} &= \mathbf{C}_{i,g} + \lambda \cdot \left(\mathbf{C}_{\text{best},g} - \mathbf{C}_{i,g}\right) + F \cdot \left(\mathbf{C}_{\xi,g} - s_{\eta,g}\right). \end{aligned} \tag{13}$$

Choice of DE variant and parameters F and λ will be discussed in Section 7.5.

Depending on the cost function, the parents or the mutated children stay within the population or not [2, 4].

(5) Termination. The termination condition is met if the number of maximum generations (i.e., $g = G$) is reached. Otherwise, the algorithm continues with step 2.

6. Filter Design under Communication System Constraints

In this section, the optimisation of a quantised IIR filter considering the overall system model is derived. After definition of the optimisation problem, the theoretical optimum receiver and the DE optimisation algorithm are proposed. The transceiver performance is then analysed with respect to implementation costs in digital hardware.

6.1. Optimisation Problem Definition. Based on the blocks of the transceiver model described so far (Figure 1),

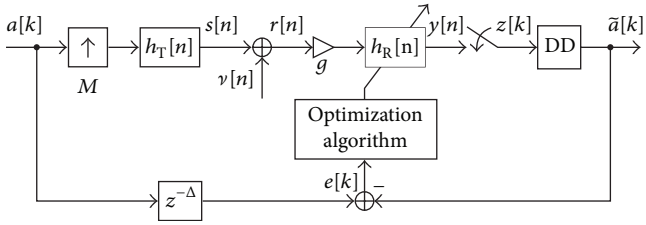


Figure 5: Communication system optimisation. Transmitted symbols are compared against the received ones to design the receiver filter.

the optimisation problem to minimise the bit errors of the receiver is defined and illustrated in Figure 5. The received symbols $\tilde{a}[k]$ are compared against the transmitted symbols $a[k]$ defining the error as

$$e[k] = a[k] - \tilde{a}[k - \Delta], \tag{14}$$

where Δ is a time delay introduced by the communication system.

For stable coefficient sets, we define the bit error rate as the squared error

$$\text{BER} = \frac{1}{N_b} \sum_{k=0}^{N_b - 1} e^2[k], \tag{15}$$

where N_b is the number of transmitted symbols for the training sequence. The cost function is dependent on the SOS matrix $\mathbf{C}$ representing the filter coefficients and the scale vector $\mathbf{s}$.

The specific cost functions used will be detailed in Section 7.

7. Experiment Setup

In the following, we detail the experiment setup. Results of experiments performed are reported in Section 8.

7.1. System Model Specification. Without loss of generality, the signaling scheme is chosen to be a binary antipodal PAM with $a = \{-\sqrt{\mathscr{E}_b}, +\sqrt{\mathscr{E}_b}\}$. The symbol rate R_{sym} is 125 kbit/s and the upsampling factor $M = 16$. The pulse shape of the transmitted pulses is a root-raised-cosine pulse with a roll-off factor of $\alpha = 0.5$. The symbol sequence $\{a[k]\}$ is randomly generated with equiprobal symbols, and the simulation is done for $N_b = 10^6$ bits.

The communication system's target BER is set to 10^{-3} (or better). The corresponding $\mathscr{E}_b/N_0$ ratio is 7.288 dB (see Figure 6 for an illustration).

The gain g which is introduced in Figure 5 is used to prescale the input number range of the receiver filter to ensure fair comparison between fixed- and floating-point.

7.2. Emulation of Finite Precision Hardware. In order to allow for a correct evaluation of quantisation effects, a bit-true simulation of the implementations considered is required. We provide bit-true simulation of BER calculation for both floating-point and fixed-point arithmetics. Both simulations allow the choice of arbitrary values for the number format's precision.

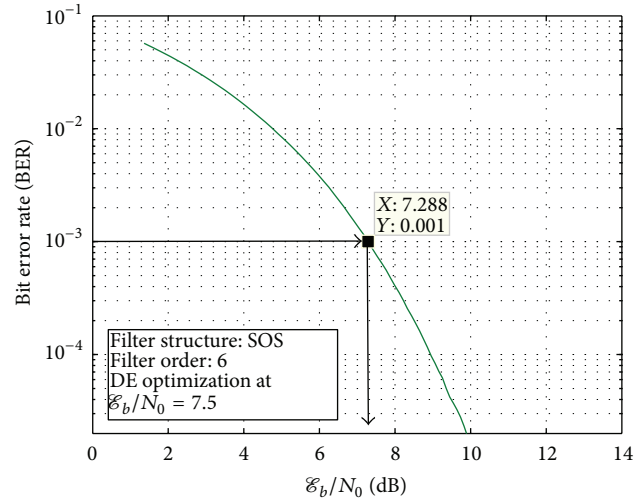


Figure 6: Transceiver BER for the optimum solution found by the DE algorithm. The operating point for fixed- and floating-point analysis is chosen to be at a BER of 10^{-3} resulting in an $\mathscr{E}_b/N_0$ ratio of 7.288 dB.

7.2.1. Fixed-Point. For fixed-point, the BER is evaluated for different $Qm \cdot n$ number formats for the state variable of the filter. This notation defines the number of integer (m) and fractional (n) bits within the representation. m is kept constant at 8 bits. Experimentally, this has been found to be sufficient to avoid overflows for all of the different solutions obtained by the DE. An overflow is indicated by a dedicated flag in the bittrue reference design. However, it might occur depending on the randomly generated noise added to the data sequence, especially when approaching a low number of fractional bits as the quantisation noise increases.

The signal range at the input of the filter is scaled to 90% of the input value range by setting $g = 0.9$ in Figure 5. This choice is significant, especially when comparing the BER to the floating-point scenario. Furthermore, it implies that for any $\mathscr{E}_b/N_0$ ratio the signal needs to be scaled accordingly by a dedicated gain control unit.

The filtering for the analysis is done based on a hardware equivalent implementation in C code and called from MATLAB using MEX [33] interface. Due to this, the filtering operation lasts a multiple of the double-precision floating-point implementation. A comparison is given in Table 4.

7.2.2. Floating-Point. For the floating-point analysis, the BER computation in Figure 12(b) is performed for different precisions while the exponent is constrained to 4 bits.

For hardware equivalent filtering, the GNU MPFR [34] library is used. It allows for floating-point operations on custom number formats (i.e., arbitrary mantissa (precision) and exponent bit widths). As the execution time is dominated by the Matlab-to-C interface, observed times are equivalent to the fixed-point implementation (see Table 4).

Table 4: Execution time analysis of a single generation of the DE1 algorithm (population size: 100).

	Matlab (double) [s]	Bit-true (fixed/float) [s]
Initialization	0.935	0.908
Filtering operation	0.603	27.25
Evaluate fitness/cost function		
Calculate BER	2.488	2.488
Check stability	0.327	0.327
Multiobjective fitness functions	—	0.123
Mutation and selection	0.284	0.284
Total time for single generation	4.637	31.380

7.3. Single-Objective Optimisation for Target Bit Error Rate. The objective function uses the BER definition given in (16).

For stable coefficient sets, we define the cost function to equal the bit error rate

$$J_{\mathrm{BER}}\left(\mathbf{s}, \mathbf{C}, N_b, \frac{\mathscr{E}_b}{N_0}\right) = \mathrm{BER}. \tag{16}$$

(a) Stability. Digital filters can be specified using filter coefficients or pole/zeros. In $\mathbb{R}$, these specifications are interchangeable. Using pole/zero description has the advantage of restricting the search space to stable systems (i.e., all poles reside inside the unit circle) by design. When considering quantised coefficients, however, there is—due to quantisation—not necessarily a one-to-one mapping between pole/zero and parameters anymore. As DE is a direct search method, the native approach appears to be using quantised filter coefficients as parameters. DE might, however, consider parameter combinations resulting in unstable filters. The objective function therefore needs to exclude unstable solutions. From quantised coefficients C, poles can be computed and the stability criterion be checked. We define a condition function $J_{\mathrm{Stable}}(\mathbf{s}, \mathbf{C})$ for the objective function $J(\mathbf{s}, \mathbf{C}, N_b, \mathscr{E}_b/N_0)$. $J_{\mathrm{Stable}}(\mathbf{s}, \mathbf{C})$ is assigned the value 1 if the filter is stable, 0 otherwise.

For an unstable parameter set, we define

$$J_{\mathrm{BER}}\left(\mathbf{s}, \mathbf{C}, N_b, \frac{\mathscr{E}_b}{N_0}\right) = \max\left(J\left(\mathbf{s}, \mathbf{C}, N_b, \frac{\mathscr{E}_b}{N_0}\right)\right) = 1. \tag{17}$$

7.4. Multiobjective Optimisation. The DE optimisation algorithm proposed so far is able to find optimised coefficients with respect to the receiver performance as given by the bit error rate (BER) measure. It already generates fully quantised coefficients that can be plugged into a hardware description. However, for FPGAs/ASICs, the hardware costs, that is, area and power, are of very high relevance. In the vicinity of the optimum found by the DE, there might be several other solutions with the same BER but with less amount of hardware requirements due to different sets of coefficients. We therefore extend the existing framework to a multiobjective optimisation approach that tries to minimise these costs at system-level design. The cost function and optimisation procedure is described later.

(b) Signal Range. In order to maximise the sensitivity of the receiver, the filters' internal signal range is evaluated. The quantisation noise is minimised if the available fractional bits of the implemented registers are used in its whole width. We define

$$J_{\mathrm{SR}} = \begin{cases} 0, & \text{if } 2^{n-1} \le \max\left(\mathrm{abs}\left(\boldsymbol{\delta}\right)\right) < 2^{n+1}, \\ 1 - \dfrac{\max\left(\mathrm{abs}\left(\boldsymbol{\delta}\right)\right)}{2^n}, & \text{if } \max\left(\mathrm{abs}\left(\boldsymbol{\delta}\right)\right) < 2^{n-1}, \\ 1, & \text{else}, \end{cases} \tag{18}$$

where $\boldsymbol{\delta}$ is a vector of all the register values after multiplication in the biquadratic filter stages and n is the number of fractional bits. Below the acceptable value range, the cost function is linearly increasing or directly forced to one if it exceeds it.

(c) Hardware Cost. A multiplier in digital hardware is, in general, built from adders. Still, an adder is only instantiated if the corresponding bit in the multiplicand (i.e., the coefficient) is set (1). As the optimisation algorithm tweaks the coefficients, this can be reformulated as an optimisation criterion. The amount of nonzero bits can be minimised by introducing a dedicated cost function J_{HW}.

For modeling the hardware costs, a full synthesis run for each individual within the population could be investigated to obtain the final area and even power consumption for a certain stimuli. However, this is impractical since the synthesis and simulation would need enormous amount of time. Therefore, the hardware costs need to be approximated. For fixed-point, several synthesis runs with different coefficient sets were performed for the digital filter. They show that the area is maximised if half of the bits are set alternating within the coefficient number representation. It drops if more or less of half of all the bits are set. Thus, we define a function $\Gamma(x, B)$ as

$$\Gamma\left(x, B\right) = \begin{cases} B_{\mathrm{set}}, & \text{if } B_{\mathrm{set}} \le \dfrac{B}{2}, \\ B - B_{\mathrm{set}}, & \text{if } B_{\mathrm{set}} > \dfrac{B}{2}, \end{cases} \tag{19}$$

where B is the bitwidth of x and B_{set} is the count of nonzero bits within the number representation of x. Then, the hardware costs are defined by summing all the set bits for the scales and coefficients and normalizing them, which is

$$J_{\mathrm{HW}}\left(\mathbf{s}, \mathbf{C}\right) = \frac{\sum_{i=1}^{N_s} \Gamma\left(s_i, B_s\right)}{N_s B_s} + \frac{\sum_{i=0}^{5} \sum_{j=1}^{N_s} \Gamma\left(C_{ij}, B_c\right)}{6 N_s B_c}, \tag{20}$$

where B_s and B_c are the bitwidths of the scales and coefficients, respectively.

(d) Summation of Cost Functions. For the multiobjective optimisation, we sum the partial costs by weighting. Given that

the filter is stable, we wish to minimise the communication system's BER. For a given set of parameters resulting in a stable filter, we therefore evaluate the cost function J as

$$J\left(\mathbf{s}, \mathbf{C}, N_b, \frac{\mathscr{E}_b}{N_0}\right) = \alpha_{\mathrm{BER}} J_{\mathrm{BER}}\left(\mathbf{s}, \mathbf{C}, N_b, \frac{\mathscr{E}_b}{N_0}\right) + \alpha_{\mathrm{HW}} J_{\mathrm{HW}}(\mathbf{s}, \mathbf{C}) + \alpha_{\mathrm{SR}} J_{\mathrm{SR}}\left(\mathbf{s}, \mathbf{C}, N_b, \frac{\mathscr{E}_b}{N_0}\right), \tag{21}$$

where J_{BER}, J_{HW}, J_{CR} are the partial cost functions and α_{BER}, α_{HW}, α_{CR} are the weighting parameters, respectively. The cost function is evaluated at step 3 of the algorithm proposed in Section 5.

For an unstable parameter set, we define

$$J\left(\mathbf{s}, \mathbf{C}, N_b, \frac{\mathscr{E}_b}{N_0}\right) = \max\left(J\left(\mathbf{s}, \mathbf{C}, N_b, \frac{\mathscr{E}_b}{N_0}\right)\right). \tag{22}$$

In order to obtain the weighting parameters, we analyse the cost functions itself. J_{BER} is depending on the operation point of the optimisation, that is, defined by $\mathscr{E}_b/\mathrm{N}_0$. J_{HW} is normalized by the total number of possible set bits and is found to be approximately 0.33 for scale and coefficient sets generated in the form of (10) and (11). In order to not influence the BER significantly, J_{HW} is chosen to be in the same number range as J_{BER}. Thus, for an expected BER of 10^{-3}, $\alpha_{\mathrm{HW}} = 0.003$ while $\alpha_{\mathrm{BER}} = 1$. In order to minimise the quantisation noise by fully using the fractional bits of the implemented registers, we expect J_{CR} to vanish. Hence, we set $\alpha_{\mathrm{CR}} = 1$ as well.

Note that even with reprogrammable coefficients via registers, the DE optimised solution is able to reduce the power consumption by minimizing the number of nonzero bits of the coefficients for fixed-point. In this case, the hardware cost function is simply reformulated to

$$\Gamma(x, B) = B_{\mathrm{set}} \tag{23}$$

as the power consumption will rise proportional with the increasing number of nonzero bits.

7.5. Choice of DE Variant. Two DE schemes were presented in Section 5. Here, we discuss identification of the variant most suitable for the design space at hand. Figure 7 shows the objective's function value over 2000 generations for 20 optimisation runs per DE variant (DE1 and DE2). DE1 provides a much more uniform behaviour and often provides better final solutions than DE2. We therefore choose DE1 as the DE variant of choice in all experiments reported in Section 8.

The control variable F in (12) and (13) was found by performing 20 simulation runs for each value between 0.1 and 0.8 with step size 0.1. Comparing the average and best fitness over generations, we found $F = 0.5$ to be the best choice delivering fast initial convergence with satisfying final solutions over multiple runs. This choice coincides with the value suggested in [2]. For λ, a value of 0.2 was found to be a reasonable choice.

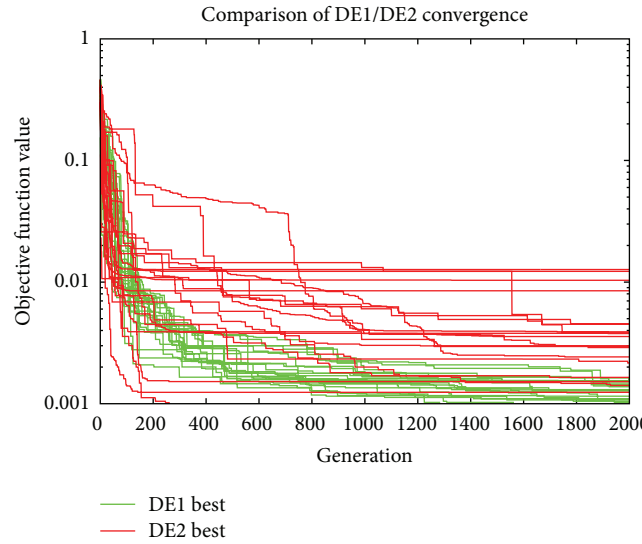


Figure 7: Comparison of convergence rate between DE1 and DE2 with multiobjective cost function. Twenty runs per DE variant. Population size 100.

8. Experiments

In the following, we report on experiments with a single objective (BER) and multiple objectives.

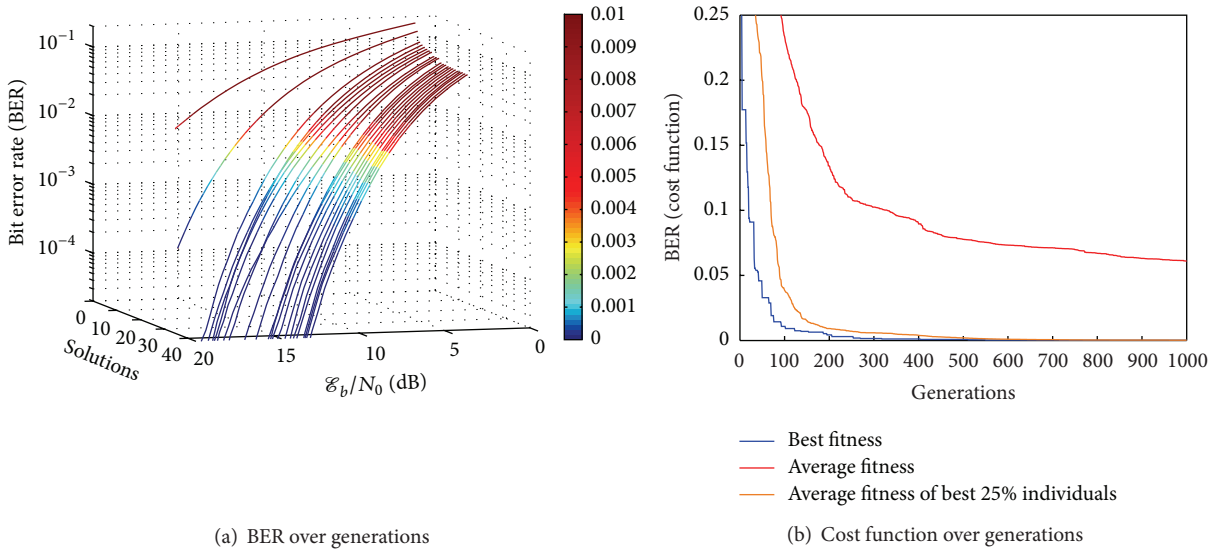
8.1. Single-Objective Optimisation for Target Bit Error Rate. In the first experiment, DE is used to search for a solution using double-precision floating point coefficients. Population size is 100 and maximum simulation run-time is set to 1000 generations (i.e., optimisation is stopped, even if results do not converge). The execution of a single generation takes 4.6 seconds on a Linux server (2x Intel Xeon CPU X5677 running at 3.47 GHz. System memory: 47 GB RAM), resulting in a maximum run time of $1000 \cdot 4.6\,\mathrm{s} = 1\,\mathrm{h}\,17\,\mathrm{min}$. Refer to Table 4 for a detailed breakdown of execution times.

Figure 8(a) shows the BER of the best solution found by DE in every iteration. It shows how the final optimum BER curve is obtained iteratively. Figure 8(b) depicts the population's best and average BER as well as the average BER of the best 25% of all individuals within the population for each generation. Note that the latter converges fast, providing several potential candidate solutions close to the optimum BER.

Figure 9 shows the BER of the individual with the lowest BER in Generation 1000 and the BER curves for the standard filters (see Section 4). As can be seen, the filter coefficients identified by DE result in a superior filter (in terms of BER) compared to all considered standard filter design approaches.

To give an impression of the design space's structure, Figure 10 shows the cost function J in the a_1/a_2 plane while all other parameters are fixed at the optimum solution identified by DE.

8.1.1. Filter Magnitude Response. Because filter specification and evaluation in the frequency domain is so ubiquitous, we compare here the DE's solutions magnitude response with the



(a) BER over generations

(b) Cost function over generations

FIGURE 8: (a) BER of the best solutions found by the DE iteratively. (b) Cost function over iterations for a single optimisation run. The average cost is computed out of the whole population.

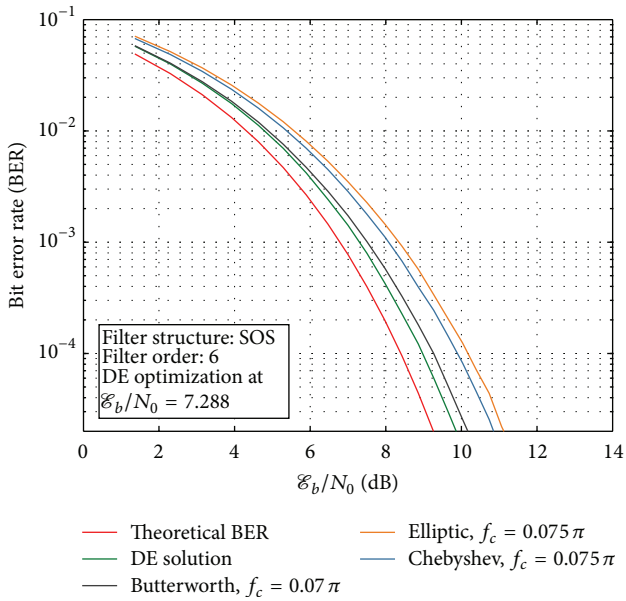


FIGURE 9: Transceiver BER for the optimum solution found by the DE algorithm at an operation point of $\mathcal{E}_b/N_0 = 7.288$. Butterworth, elliptic, and Chebyshev filters are plotted as reference. $N_b = 10^6$ bits.

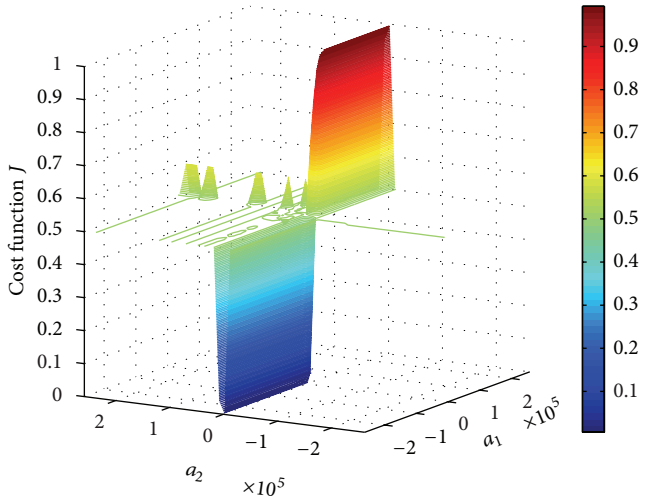


FIGURE 10: Visualisation of the cost function J over a_1 and a_2.

magnitude responses of standard reference filters. Note that the design methods are completely different. The reference filters are designed with a certain cut-off frequency which defines the filters frequency response characteristics. The optimised DE filter considers the overall system in which it is embedded and iteratively identifies suitable filter coefficients relying on a time domain perspective.

Figure 11 shows the magnitude response of Butterworth, elliptic, and Chebyshev filters as detailed in Section 4.2 together with the magnitude response of the solution identified by DE. Observe that the optimal solution's gain is significantly higher than the filter specification derived from the matched filter assumption.

8.1.2. Choice of Number Format. A communication system's performance is significantly depending on the receiving's filter number format and precision chosen. Quantifying this dependency is the goal of the following experiment.

DE was used to search for the optimal solution given a fixed-point receiver filter implementation and given a floating-point receiver implementation.

Figure 12(a) shows the BER over different fixed-point $Qm \cdot n$ formats, where m is constantly set to 8 and n is the number of fractional bits. Figure 12(b) shows the BER over different floating-point precisions, where the exponent is limited to the range of ±15. The red line shows the BER

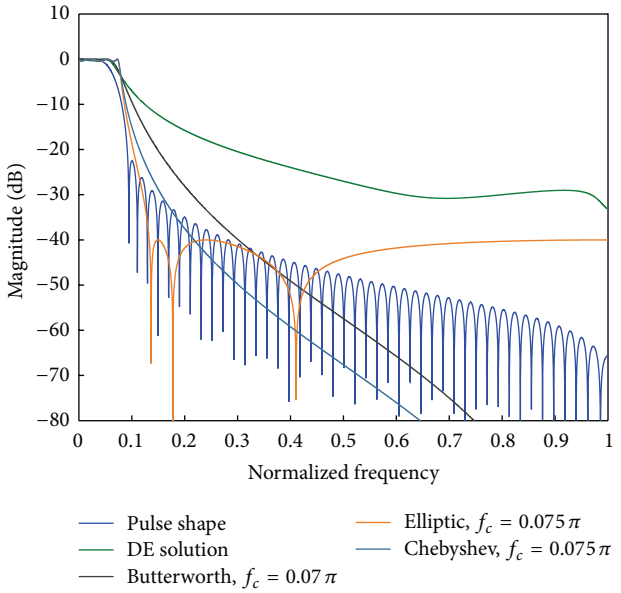
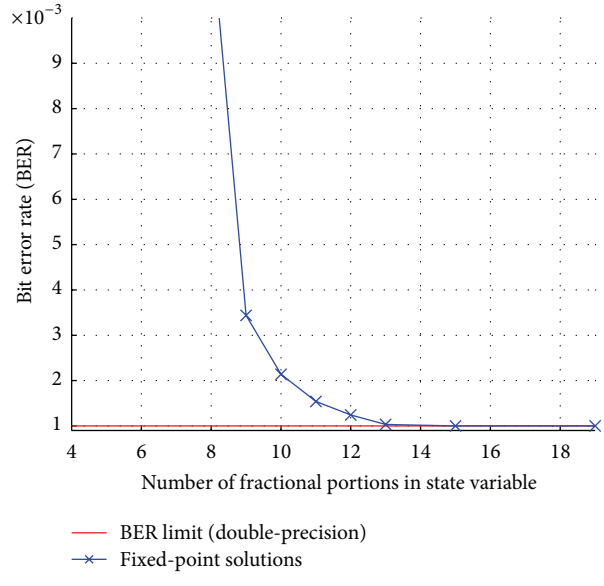


Figure 11: Magnitude responses of transmission, DE optimised and reference filters. DE optimised and reference filters are designed with order 6.

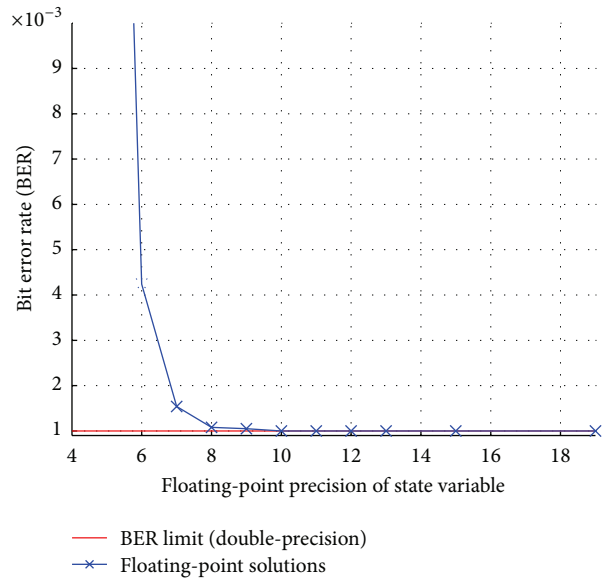
achieved with the software reference implementation using double-precision floating point.

8.2. Multiobjective Optimisation for Hardware Efficiency. To give an impression of the design space's structure when using a multiobjective cost function, Figure 13 shows the cost function J in the a_1/a_2 plane while all other parameters are fixed at the optimum solution identified by DE.

The required chip area found by the DE with and without multiobjective optimisation in fixed-point arithmetic is presented in Figure 14. The synthesis is performed for predefined coefficients, which means that they are constants at synthesis time. On average, around 4.4% can be saved utilizing the multiobjective optimisation with hardware costs (Table 5) while at the same time the BER from Figure 12(a) is retained. It is clear that, for the DE algorithm without dedicated cost function, the hardware costs are purely random. Thus, also the saving in area compared to the multiobjective optimisation is varying between 1% and 12%.



(a) BER fixed-point



(b) BER floating-point

Figure 12: (a) BER over different fixed-point $Qm \cdot n$ formats, where m is constantly set to 8 and n is the number of fractional bits. (b) BER over different floating-point precisions, where the exponent is limited to the range of ±15.

9. Discussion

Using the BER as optimisation objective, the DE algorithm is able to identify coefficient sets which outperform systems relying on conventional filter design methods with specification in the frequency domain (matched filter). Inspecting the resulting filter in the frequency domain reveals that the magnitude response deviates significantly from the matched filter's magnitude response. This result does not interfere with the matched filter theory. It reveals, though, that under the presence of channel noise and nonlinearities (finite-precision arithmetic), a filter optimising the overall BER can be implemented with a magnitude response significantly different than the specification of the matched filter. This is a strong argument against direct specification of filters approximating a matched filter.

Employing a direct search algorithm has the advantage that—given it works reliably on the original problem space—new search criteria (i.e., extended cost functions) can be added (almost) at will. Additional partial cost functions will not break the search algorithms efficiency (i.e., reduce the convergence rate). We have demonstrated this strategy successfully by adding hardware measures. This results in reliably cheaper implementations with identical BERs.

Using the DE algorithm with a single BER objective can result in many solutions with almost identical cost (i.e., BER)

Table 5: Chip area in μm^2 for predefined coefficients (constant at synthesis time) with and without multi-objective optimisation for hardware efficiency (fixed-point). The mean is computed out of the best 25% solutions within the final population. Populations size is 100.

Fractional bits (n)	Area predefined coeffs without hardware opt.		Area predefined coeffs with hardware opt.		Area savings
	Mean best 25%	Best	Mean best 25%	Best	
7	40048	37738	39009	33183	12%
8	45441	42977	44736	42688	1%
9	51879	49506	50513	47065	5%
10	58855	54360	55379	49585	9%
11	67323	62873	66895	61007	3%
12	74079	68240	73835	64848	5%
13	82066	77486	80760	75403	3%
15	99119	91118	98238	90403	1%
19	145077	136627	142296	135049	1%

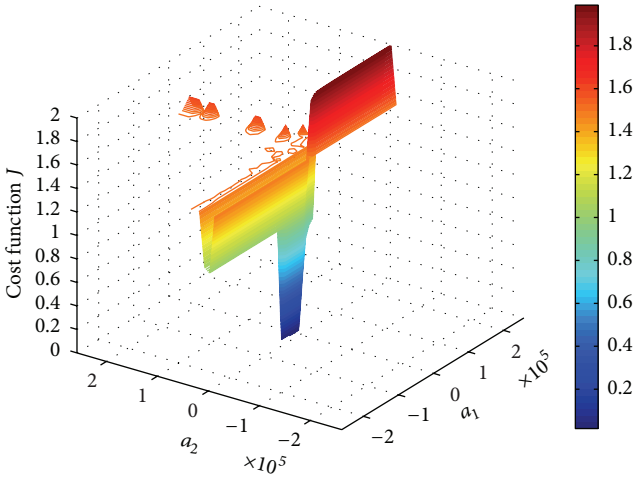


Figure 13: Visualisation of the multiobjective cost function J over a_1 and a_2.

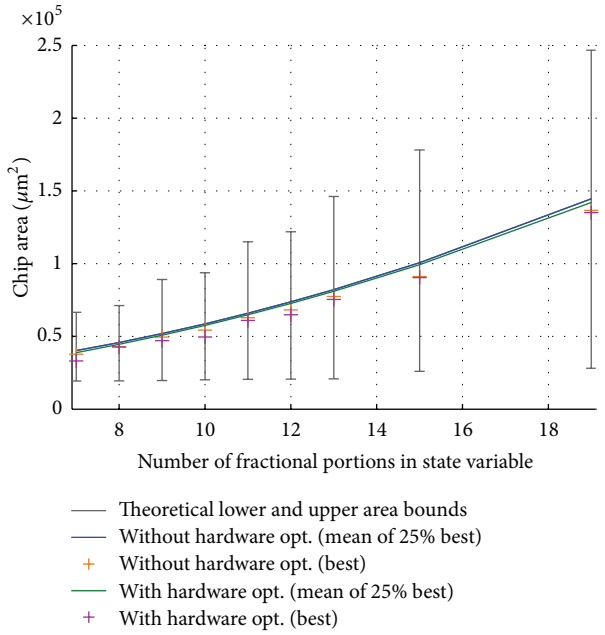


Figure 14: Chip area for predefined coefficients with and without multiobjective optimisation for hardware efficiency (fixed-point) as presented in Table 5. Limit bars indicate the theoretical minimum and maximum area requirements.

but significantly varying hardware costs due to the different amounts of set bits in the coefficients. The uncertainty in hardware cost lies between a factor of 2 and 10 depending on the number format considered. Introducing an additional hardware cost objective allows for significant reduction of this uncertainty. Consequently, using a multiobjective cost function taking the number of set bits into account allows for a much more reliable design process guaranteeing identification of hardware-efficient filter implementations.

We have shown partial visualisations of the cost function. While no general insights can be derived from these minimal projections, it is interesting to observe the change in appearance with modifications to the cost function J. Figure 10 shows the BER as a function of filter coefficients a_1 and a_2 while all other scaling values and coefficients are kept fixed at the values of the DE algorithm's solution. The plot shows that for the depicted partial design space, the constraints for a good solution (low BER) are abs(a_2) being close to zero and a_1 being negative. The exact value of a_1 is of little relevance. Extending the objective function by additional criteria (see (21)) changes the situation significantly. Figure 13 shows the multiobjective cost function J over filter coefficients a_1 and a_2 while all other scaling values and coefficients are kept fixed at the values of the DE algorithm's final solution. The additional criteria have further constrained the acceptable value of a_1 to the interval $0, \ldots, -0.5$.

Filter structures are expensive in terms of chip area, especially when implemented with high precision. It is of major interest to identify the most suitable number format (fixed-point, floating-point; precision) and to minimise the power consumption by suitable choice of coefficients. Comparing Figures 12(a) and 12(b) shows that the fixed-point filter's BER deteriorates much faster with decreased precision than the floating-point filter's BER (13 bit versus 9 bit precision). One can also observe that the deviation from the red line is much

more prominent in the fixed-point case. Typically, floating point arithmetic is considered more costly than fixed-point. Our experiments show that, from a system perspective, the difference in chip area for otherwise comparable system is not so clear. Floating-point arithmetic requires more area per bit precision than fixed-point arithmetic. It requires, however, less precision to achieve a specific BER. Given that our floating-point implementation is far from optimal, the chip area per bit error can be considered roughly equivalent.

10. Conclusion

DE can sample the huge time domain design space of a communication system with a 6th-order receiving filter including bit-true BER simulation using Matlab/Mex/MPFR in 31.38 s · 1000 generations = 8 h 43 min using standard hardware (single core). Given that this design space exploration is performed offline, the time required is acceptable. Each identified filter coefficient set is already quantised. The cost function is therefore accurate and the filter directly implementable.

Using DE as an exploration tool for the huge discrete design space of digital receiver filters allows for interesting insights. We have demonstrated two, namely, that (1) receiving filters with magnitude responses deviating significantly from the matched filter's magnitude response can outperform standard filter design techniques relying on specification in the frequency domain and that (2) from a system perspective, fixed- and floating-point IIR implementations have roughly comparable cost.

Both findings are in contrast to current practice in filter design for communication systems. We therefore mandate further BER-guided exploration of communication system's discrete filter design space.

11. Future Work

We have chosen to investigate communication systems without intersymbol interference (ISI). This typically is the case in low-rate communication systems. It would be interesting to investigate how systems with intersymbol interference could benefit from the implicit filter specification as described in our work.

We have shown that implicit filter specification through system BER specification and system simulation can result in filter designs with significantly different magnitude responses than conventional filter specifications typically used for communication systems. This shows the importance of taking into consideration nonlinear effects, rendering the matched-filter assumption void. We have only considered the most basic nonlinear effects, namely, an AWGN channel, BER calculation in time domain, and the finite-precision effects of digital filter implementations. Many more opportunities for a more accurate system model exist [15]. It would be worthwhile exploring further nonlinear effects and their respective impact on filter design. Furthermore, indoor as well as multipath channel models would make the simulation results more relevant for practical applications.

Differential Evolution has become a widely used tool since its inception almost 20 years ago. We have chosen to implement DE as originally described in [2] as its performance was sufficient for our purpose. It would be worth exploring state-of-the-art DE implementations [19], especially integrating automatic/dynamic schemes for parameter (population size, F, λ) selection. Reducing optimisation time through parallel DE would be another fruitful direction for future work. Not all DE schemes, however, lend themselves equally well to efficient parallelisation [35].

References

[1] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, Prentice Hall Signal Processing Series, Pearson Education, Upper Saddle River, NJ, USA, 2009.

[2] R. Storn and K. Price, "*Differential evolution*-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[3] J.-J. Shyu and Y.-C. Lin, "Design of finite-wordlength IIR digital filters in the time/spatial domain," in *Proceedings of the 1995 IEEE International Symposium on Circuits and Systems (ISCAS '95)*, vol. 3, pp. 2027–2030, May 1995.

[4] R. Storn, "Designing nonstandard filters with *Differential evolution*," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 103–106, 2005.

[5] R. Storn, "System design by constraint adaptation and *Differential evolution*," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 1, pp. 22–34, 1999.

[6] G. Ramos and J. J. López, "Filter design method for loudspeaker equalization based on IIR parametric filters," *Journal of the Audio Engineering Society*, vol. 54, no. 12, pp. 1162–1178, 2006.

[7] M. Haseyama and D. Matsuura, "A filter coefficient quantization method with genetic algorithm, including simulated annealing," *IEEE Signal Processing Letters*, vol. 13, no. 4, pp. 189–192, 2006.

[8] Y. Yang and X. Yu, "Cooperative coevolutionary genetic algorithm for digital IIR filter design," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1311–1318, 2007.

[9] M. Nilsson, M. Dahl, and I. Claesson, "Digital filter design of IIR filters using real valued genetic algorithm," in *Proceedings of the 2nd WSEAS International Conference on Electronics, Control and Signal Processing (ICECS '03)*, pp. 3:1–3:6, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wis, USA, 2003.

[10] N. Karaboga, "Digital IIR filter design using *Differential evolution* algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 7, no. 7, pp. 1269–1276, 2005.

[11] B. Luitel and G. K. Venayagamoorthy, "*Differential evolution* particle swarm optimization for digital filter design," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, IEEE World Congress on Computational Intelligence, pp. 3954–3961, June 2008.

[12] K.-S. Tang, K. F. Man, S. Kwong, and Z. F. Liu, "Design and optimization of IIR filter structure using hierarchical genetic algorithms," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 3, pp. 481–487, 1998.

[13] S. A. Hashemi and B. Nowrouzian, "A novel finite-wordlength particle swarm optimization technique for FRM IIR digital

filters," in *Proceedings of the IEEE International Symposium of Circuits and Systems (ISCAS '11)*, pp. 2745–2748, May 2011.

[14] C. Dai, W. Chen, and Y. Zhu, "Seeker optimization algorithm for digital IIR filter design," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1710–1718, 2010.

[15] T. Stücke, N. Christoffers, R. Kokozinski, and S. Kolnsberg, "BER optimization for micro power receivers using quick and accurate system simulation," in *Proceedings of the 14th IST Mobile & Wireless Communications Summit*, June 2005.

[16] A. Hjorungnes, P. S. R. Diniz, and M. L. R. de Campos, "Jointly minimum BER transmitter and receiver FIR MIMO filters for binary signal vectors," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 1021–1036, 2004.

[17] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution-A Practical Approach to Global Optimization*, Springer, New York, NY, USA, 2005.

[18] U. K. Chakraborty, *Advances in Differential Evolution*, Springer, New York, NY, USA, 2008.

[19] S. Das and P. N. Suganthan, "*Differential evolution*: a survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[20] W. Zhu, J.-A. Fang, Y. Tang, W. Zhang, and W. Du, "Digital IIR filters design using *Differential evolution* algorithm with a controllable probabilistic population size," *PLoS ONE*, vol. 7, no. 7, Article ID e40549, 2012.

[21] S.-T. Pan, "Evolutionary computation on programmable robust IIR filter Pole-Placement design," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 4, pp. 1469–1479, 2011.

[22] D. Zaharie, "Influence of crossover on the behavior of *Differential evolution* algorithms," *Applied Soft Computing*, vol. 9, no. 3, pp. 1126–1138, 2009.

[23] B. Widrow and I. Kollár, *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*, Cambridge University Press, Cambridge, UK, 2008.

[24] A. Cox, S. Sankaranarayanan, and B.-Y. Chang, "A bit too precise? Bounded verification of quantized digital filters," in *Tools and Algorithms for the Construction and Analysis of Systems*, C. Flanagan and B. König, Eds., vol. 7214 of *Lecture Notes in Computer Science*, pp. 33–47, Springer, Berlin, Germany, 2012.

[25] H. Kar, "Asymptotic stability of fixed-point state-space digital filters with combinations of quantization and overflow nonlinearities," *Signal Processing*, vol. 91, no. 11, pp. 2667–2670, 2011.

[26] J. Proakis, *Digital Communications*, McGraw-Hill Series in Electrical and Computer Engineering, McGraw-Hill, New York, NY, USA, 2001.

[27] M. F. Quelhas and A. Petraglia, "On the design of IIR digital filter using linearized equation systems," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '10)*, pp. 2702–2705, June 2010.

[28] A. Eletri, E. Salem, A. Zerek, and S. Elgandus, "Effect of coefficient quantization on the frequency response of an IIR digital filter by using software," in *Proceedings of the 7th International Multi-Conference on Systems, Signals and Devices (SSD '10)*, pp. 1–6, June 2010.

[29] Y. Zhao, J. Zhou, X. Zhou, and G. Sobelman, "General lattice wave digital filter with phase compensation scheme," in *Proceedings of the IEEE 9th International Conference on ASIC (ASICON '11)*, pp. 220–223, October 2011.

[30] C. Huang, Z. Xu, G. Li, H. Xu, and L. Chang, "An improved lattice IIR digital filter structure," in *Proceedings of the 8th International Conference on Information, Communications and Signal Processing (ICICS '11)*, pp. 1–5, December 2011.

[31] B. W. Bomar, *Finite Wordlength Effects*, CRC Press, New York, NY, USA, 2009.

[32] S. Luke, *Essentials of Metaheuristics*, Lulu Press, Raleigh, NC, USA, 2009.

[33] The MathWorks, "Compile MEX-function from C/C++ or Fortran source code," 2012, http://www.mathworks.de/de/help/matlab/ref/mex.html.

[34] MPFR, "The GNU MPFR library," 2012, http://www.mpfr.org/ .

[35] D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Parallel *Differential evolution*," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '04)*, vol. 2, pp. 2023–2029, June 2004.