

# Evolutionary Multiobjective Optimization for Digital Predistortion Architectures

Lin Li<sup>1</sup>(✉), Amanullah Ghazi<sup>2</sup>, Jani Boutellier<sup>2</sup>, Lauri Anttila<sup>3</sup>,  
Mikko Valkama<sup>3</sup>, and Shuvra S. Bhattacharyya<sup>1,3</sup>

<sup>1</sup> ECE Department, University of Maryland, College Park, MD 20742, USA  
{l1i12311,ssb}@umd.edu

<sup>2</sup> Department of Computer Science and Engineering, University of Oulu,  
Oulu, Finland

{amanullah.ghazi,jani.boutellier}@ee.oulu.fi

<sup>3</sup> Department of Electronics and Communications Engineering,  
Tampere University of Technology, Tampere, Finland  
{lauri.anttila,mikko.e.valkama}@tut.fi

**Abstract.** In wireless communication systems, high-power transmitters suffer from nonlinearities due to power amplifier (PA) characteristics, I/Q imbalance, and local oscillator (LO) leakage. *Digital Predistortion (DPD)* is an effective technique to counteract these impairments. To help maximize agility in cognitive radio systems, it is important to investigate dynamically reconfigurable DPD systems that are adaptive to changes in the employed modulation schemes and operational constraints. To help maximize effectiveness, such reconfiguration should be performed based on multidimensional operational criteria. With this motivation, we develop in this paper a novel evolutionary algorithm framework for multi-objective optimization of DPD systems. We demonstrate our framework by applying it to develop an adaptive DPD architecture, called the *adaptive, dataflow-based DPD architecture (ADDA)*, where Pareto-optimized DPD parameters are derived subject to multidimensional constraints to support efficient predistortion across time-varying operational requirements and modulation schemes. Through extensive simulation results, we demonstrate the effectiveness of our proposed multiobjective optimization framework in deriving efficient DPD configurations for run-time adaptation.

**Keywords:** Digital predistortion · Multiobjective optimization · Evolutionary algorithms

## 1 Introduction

In wireless communication systems, I/Q mismatch, power amplifier (PA) nonlinearities, and signal leakage in the local oscillator (LO) are implementation-related problems that must be addressed before the direct-conversion principal can be deployed. In the frequency domain of the transmitted signal, the effects of

these impairments are translated as power leakage into adjacent channels. Digital predistortion (DPD) is a widely investigated technique (e.g., see [2–4, 7, 9]) to counteract such impairments by applying carefully-calculated distortion to the signal prior to transmission.

A major challenge in deploying DPD architectures for cognitive radio systems is the dynamic optimization of key DPD parameters subject to time-varying and multidimensional constraints on system performance. A general approach to such optimization is to perform efficient search at design time (i.e., off-line) across alternative DPD configurations, and to then select from the search results a set of configurations that are Pareto-optimal, and that effectively cover the targeted range of operational scenarios and their trade-offs. These selected, “Pareto-optimized” configurations can then be stored in memory, and switched across during system operation based on time-varying changes in communication system requirements. Here, “Pareto-optimized” configurations refer to configurations that are Pareto-optimal with respect to the applied search process, while “Pareto-optimal” configurations refer to configurations that are globally optimal in a Pareto sense.

In this paper, we develop a novel framework for systematic derivation of Pareto-optimized DPD system configurations that can be applied to adaptive DPD implementations. Our framework builds on the methodology of multiobjective evolutionary algorithms (e.g., see [12]), and incorporates adaptations of this methodology to efficiently handle distinguishing characteristics of DPD system optimization. We refer to our framework for DPD system optimization as the framework for *Evolutionary Adaptive DPD Implementation (EADI)* or (“EADI Framework”).

We demonstrate the EADI Framework in this paper by applying it to develop an adaptive DPD architecture, called the *adaptive, dataflow-based DPD architecture (ADDA)*, where Pareto-optimized DPD parameters are derived subject to multidimensional constraints to support efficient predistortion across time-varying operational requirements and modulation schemes. While the ADDA architecture is used to concretely demonstrate the capabilities of the EADI Framework, the EADI Framework is not specific to any particular DPD architecture, and can readily be adapted to work across a variety of parameterized DPD architectures. Exploring such adaptations is a useful direction for future work that emerges from the developments of this paper.

The design evaluation metrics (optimization objectives) targeted in our development of the EADI Framework and ADDA architecture in this paper are system energy consumption, adjacent channel power ratio (ACPR), and system accuracy. We abbreviate this set of metrics as *EAA*.

The ADDA is a parameterized architecture that can be configured dynamically to achieve a range of EAA trade-offs. The DPD design space that we consider consists of three design parameters: the polynomial order, bit-width, and filter order. This design space is modeled in the EADI Framework, and optimization results from the framework are used to extract a subset of generated Pareto-optimized configurations (settings of the DPD parameter values).

This subset of configurations provides the set of DPD system modes that will be implemented in the ADDA architecture. The set of DPD modes provided in the ADDA configuration set is made available during operation such that predistortion trade-offs can be reconfigured among the different options in the configuration set based on dynamically changing operational requirements.

To demonstrate and experiment with the ADDA, we apply the *lightweight dataflow environment (LIDE)*, which is a design tool for dataflow-based design and implementation of signal processing systems [8]. Dataflow graphs provide a useful form of model-based design in many areas of signal processing, and wireless communications (e.g., see [11]). We map the signal flow structure of the ADDA into actors (dataflow-based signal processing components) in LIDE, and implement the internal functionality of these actors using the Verilog hardware description language (HDL).

We demonstrate the effectiveness of the EADI Framework through extensive simulations, and validate the capabilities of the ADDA through hardware synthesis.

## 2 Related Work

Unlike earlier DPD architectures (e.g., see [3, 5]), the DPD algorithm proposed in [1] is one of the first DPD techniques that jointly compensates for PA nonlinearities and I/Q modulator impairments. This DPD architecture employs an extended parallel Hammerstein structure, which decomposes DPD operation into direct and conjugate predistortion subsystems. Such a decomposed structure provides additional degrees of freedom in system design. In this paper, we exploit the decomposed, parallel structure of the DPD method introduced in [1] and we present new methods to search the design space, and derive Pareto-optimized realizations for this form of DPD architecture.

In architectures for cognitive radios, adaptive DPD systems that operate under Pareto-optimized configurations are highly desirable due to the multidimensional space of relevant implementation metrics. However, prior work on system-level DPD optimization has emphasized single-objective optimization of ACPR [2, 9]. These works employ a form of search technique called genetic algorithms, which are closely related to evolutionary algorithms, to optimize DPD ACPR performance. However, the resulting solutions may not be efficient in terms of energy consumption or accuracy. Furthermore, the underlying design methodology does not produce multiple alternative configurations that may be employed for dynamic reconfiguration based on time-varying changes in operational requirements. The methods that we develop in this paper address these limitations, respectively, through development of the (1) EADI Framework for multidimensional, Pareto-optimized DPD configuration, and (2) ADDA for reconfigurable DPD architecture implementation based on configurations that are derived by the EADI Framework.

### 3 Adaptive Dataflow-Based DPD Architecture

The ADDA architecture developed in this paper is based on the algorithm presented in [1]. This DPD algorithm operates in two stages. In the *coefficient estimation* stage, the DPD filtering coefficients are estimated. The estimated coefficients are then employed in the *DPD filtering* stage for actual predistortion of the input signal. Since the first stage is intended for off-line computation, the ADDA architecture and EADI optimization process are focused only on the second (filtering) stage.

Figure 1 illustrates the dataflow model of the DPD filtering subsystem that is employed in the ADDA. Here, the mode selection actor dynamically selects the DPD operational mode based on the current application scenario (i.e., based on the current modulation and requirements on EAA) and finds the corresponding parameter settings for that mode in its local memory, and distributes these DPD parameter values to the polynomial computation actor and all of the filter actors. Following [1], we decompose the signal processing for the applied DPD algorithm into separate direct and conjugate parts.

With the parameters obtained from the mode selection actor, the polynomial computation actor computes the polynomial basis function defined in [1] for both the direct and conjugate branches. The computed polynomials are then sent to their corresponding branches and filtered by the filter actors in those branches. These filter actors are implemented with integrated use of LIDE and Verilog, as described in Sect. 1. As shown in Fig. 1, the filtered samples (one output sample from each filter) are summed to produce a single sample as the final predistorted output. Based on the analysis in [4], where a similar dataflow model is constructed for the DPD algorithm in [1], most of the computation and energy consumption is concentrated in the filter actors. Thus, in this paper, we map only the filter actors to hardware, and focus our design optimization processes on the filter actors.

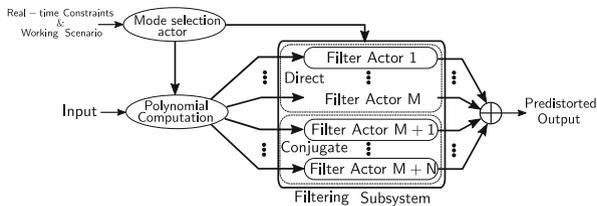


Fig. 1. Dataflow graph model of the predistortion filter.

## 4 Optimization Metrics and Design Space

### 4.1 Optimization Metrics

In this subsection, we elaborate on the three objectives in our targeted design optimization problem. As defined in Sect. 1, we refer to these metrics collectively as *EAA*.

**Energy Measurement.** As explained in Sect. 3, we focus our energy measurement on the energy consumed by the filtering subsystem, and the figure of merit that we employ is the filtering energy expended to producing a single output sample, which is denoted by the *energy per sample (eps)*. To calculate eps, we use the total power consumption of all FIR filters used in the predistortion subsystem, which we denote as  $P_{FIR}$ . The eps metric is then defined as  $\text{eps} = P_{FIR} \times C/F$ , where  $C$  represents the average number of clock cycles required by the filter actors to process a single new input sample, and  $F$  represents the clock frequency. In our design, both  $F$  and  $C$  are fixed for each configuration. Thus, eps is proportional to  $P_{FIR}$ , and we can therefore use  $P_{FIR}$  as optimization objective for our evolutionary algorithm process. Also, we report results for  $P_{FIR}$  in Sect. 6 (instead of eps) as our assessment of the energy efficiency of each configuration.

We implement the DPD filtering subsystem using the Altera EP2C35F672C6 FPGA from the Cyclone II family. To facilitate efficient design space exploration within the EADI optimization process, we model the power consumption as a function of the design vector  $[P \ Q \ \mathbf{B}\mathbf{W}^T \ \mathbf{F}\mathbf{O}^T]^T$ . The definitions of the quantities  $P$ ,  $Q$ ,  $\mathbf{B}\mathbf{W}$  and  $\mathbf{F}\mathbf{O}$  are given in Sect. 5.

Our approach to system-level DPD power estimation starts by first measuring the total power consumption of a single branch under all valid filter order and bit-width values using Altera PowerPlay Analyzer. The power consumption for a specific DPD configuration is then estimated as

$$Power_{est} = \sum_{p \in I_P} Power_p(bw_p, fo_p) + \sum_{q \in I_Q} Power_q(bw_q, fo_q), \quad (1)$$

where  $I_P$  and  $I_Q$  are the set of direct branches and conjugate branches, respectively;  $bw_x$  and  $fo_x$  are the bit-width and filter order for branch  $x$ , respectively, and  $Power_x(bw_x, fo_x)$ , the power consumed by branch  $x$  with bit-width  $bw_x$  and filter order  $fo_x$ , is obtained from the aforementioned power measurement process.

**ACPR Measurement.** ACPR is a metric that is commonly used to assess the extent of out-of-band energy leakage [7]. ACPR is defined as the ratio of the mean power centered on the adjacent channel to the mean power centered on the desired channel, as shown in (2).

$$\text{ACPR} = 10 \log_{10} \frac{\int_{\omega_A} S(\omega) d\omega}{\int_{\omega_D} S(\omega) d\omega}. \quad (2)$$

Here,  $S(\omega)$  denotes the power spectral density of the postdistorter input signal  $s_n$ , and  $\omega_A$  and  $\omega_D$  denote the frequency bands of the adjacent channel and desired channel, respectively.

**Accuracy Measurement.** We measure the accuracy of candidate DPD designs by the *error vector magnitude (EVM)* and *symbol error rate (SER)*. The former is considered as an optimization objective and the latter as a constraint on the

derived configurations. The EVM measures the distortion of original symbols under the influence of non-linearities introduced by the PA and DPD. This distortion is calculated as

$$\text{EVM}(\text{Pf}) = 10 \log_{10} \left( \frac{1}{K} \sum_{k=1}^K |X_0(k) - \hat{X}^{\text{Pf}}(k)|^2 \right), \quad (3)$$

where Pf represents a certain profile (finite sequence)  $X_0(1), X_0(2), \dots, X_0(K)$  of symbols to be transmitted, and  $\hat{X}^{\text{Pf}}(k)$  is the  $k$ th actual transmitted symbol under Pf.

SER is measured as the average rate of erroneous symbol transmissions. This rate is determined as

$$\text{SER}(\text{Pf}) = \frac{1}{K} \sum_{k=1}^K I(X_0(k) - \hat{X}^{\text{Pf}}(k)), \quad (4)$$

where  $I(x)$  (the *indicator function*), has value 1 if  $x \neq 0$  and 0 otherwise. We require that all of the configurations extracted for mapping into the ADDA must have zero SER.

## 4.2 Design Space

In this section, we elaborate on the selected DPD parameters that define the predistorter design space associated with the ADDA.

**Polynomial Orders.** As mentioned in Sect. 3, the DPD algorithm proposed in [1] splits its signal processing into a direct part and a conjugate part, which enables use of different polynomial orders for direct and conjugate signal terms. For example, a DPD system can be realized with fifth-order for the direct signal and only third-order for the conjugate signal. We denote the polynomial order for the direct signal and conjugate signal by  $P$  and  $Q$ , respectively. Following [1], only odd values for  $P$  and  $Q$  are considered. Thus, the number of branches (or filter actors) that is employed in a specific DPD configuration is given by  $N_{\text{branch}} = (P + 1)/2 + (Q + 1)/2$ . In our experiments, we set the domain  $D$  of valid values for both  $P$  and  $Q$  as  $D = \{1, 3, 5, 7, 9\}$ . Thus, there are in total 25  $P - Q$  combinations in our targeted design space.

**Bit-Widths.** Intuitively, smaller bit-widths for data storage and computation lead to less energy consumption. However, signal processing accuracy may be traded off as a consequence. To incorporate this trade-off between energy efficiency and accuracy, we incorporate bit-width as a parameter of ADDA, and as a design space component of EADI. Considering requirements on system accuracy and constraints on hardware resources, we set the range of allowable bit-widths in our experiments as  $\{5, 6, \dots, 15\}$ . Additionally, we allow different branches to be configured with different bit-widths in the same design. This leads to great

flexibility in design optimization, and a correspondingly large design space — if there are  $m$  branches used in a specific design, then the total number of valid bit-width combinations is  $11^m$ .

**Filter Orders.** Similar to the bit-width design, the filter used in each branch may also have different number of coefficients. We denote this parameter as *filter order*. The filter order parameters would also significantly affect the trade-offs among EAA. The range of filter order in this work is set to be  $\{1, 2, 3, 4, 5\}$ .

According to the above description, our design space is too huge for exhaustive search. As a numerical example, given the aforementioned ranges for the system parameters, the design space would contain more than  $55^{10}$  configurations.

## 5 Multiobjective Optimization Using Evolutionary Algorithm

As motivated in Sect. 4, the DPD design space addressed in this work is a complex multidimensional space that is too large to be evaluated using exhaustive search techniques. Therefore, we apply a heuristic search strategy called *evolutionary algorithms (EAs)*, including a particular form of EA, called *strength Pareto EA (SPEA)*, that is suited for multiobjective optimization [12]. We select the SPEA approach due to its efficiency and scalability in addressing complex optimization problems, and its customizability to different kinds of design spaces and optimization criteria. This latter feature makes the EADI Framework readily adaptable across different kinds of DPD architectures and communication system constraints.

### 5.1 Problem Encoding

The parameters involved in the DPD design optimization problem are polynomial orders, bit-widths, and filter orders. Each configuration can be represented throughout the EA process by a vector, specified as  $[P \ Q \ \mathbf{BW}^T \ \mathbf{FO}^T]^T$ . Here,  $P$  and  $Q$  are the direct and conjugate polynomial order, respectively. As described in Sect. 4, the maximum number of branches considered in the design space is 10 (at most 5 branches for both the direct signals and the conjugate signals). Thus,  $\mathbf{BW}$  is a vector with 10 dimensions representing bit-width settings for up to 10 branches, where each dimension represents the bit-width associated with the corresponding branch. For the branches that are not used, the corresponding vector elements are set to zero. Similar conventions are applied to generate the 10-dimensional vector  $\mathbf{FO}$  of filter order settings.

As discussed in Sect. 1, the objective space of the EADI Framework encompasses average power consumption, ACPR and EVM. Thus, the objective vector can be formulated as  $[P_{FIR} \ \text{ACPR} \ \text{EVM}]$  with units (mW, dBc, dB). Here,  $P_{FIR}$  is the power consumption, as estimated by the method discussed in Sect. 4, and ACPR and EVM are calculated according to (2) and (3), respectively.

## 5.2 Optimization Process

The EADI optimization process is executed separately for each modulation type that is to be supported in the targeted ADDA platform. The resulting Pareto-optimized configurations for the different modulation types are then collected and stored in the ADDA memory. This enables the ADDA to dynamically to select among different modulation types, and among different operational trade-offs for each modulation type.

As mentioned previously, the work flow of the EADI optimization process is based on the SPEA methodology for multidimensional search. For details on SPEA, we refer the reader to [12].

According to SPEA, the population set (set of candidate solutions or *individuals*)  $\rho$  contains the individuals generated during each SPEA iteration, and the external set  $\bar{\rho}$  maintains selected non-dominated individuals among all individuals generated so far up through the current iteration. Here, we say that an individual  $x$  dominates another individual  $y$  if  $x$  is superior to  $y$  in terms of at least one design evaluation metric, and  $x$  is not inferior to  $y$  in terms of any metric. A *non-dominated* individual is one that is not dominated by any individual.

We initialize  $\rho$  with a well-distributed population across the design space. For each possible  $P - Q$  combination, we generate two design vectors by selecting the corresponding bit-width and filter order values randomly from their valid ranges. Thus, the size of  $\rho$ , denoted by  $\mathbf{N}$ , is 50 individuals.

During each iteration, each individual in  $\rho$  is evaluated to generate the objective vector  $[P_{FIR} \text{ ACPR EVM}]$ . The individuals that do not satisfy certain modulation-specific constraints (defined in Sect. 6) are ignored. Only the remaining non-dominated individuals are copied to  $\bar{\rho}$ . If the size of  $\bar{\rho}$  exceeds a predefined maximum population size  $\bar{N}_{max}$ , a k-means clustering algorithm is used to classify the members in  $\bar{\rho}$  into  $\bar{N}_{max}$  groups. This allows us to limit the size of  $\bar{\rho}$  while maintaining a diverse population in  $\bar{\rho}$  by retaining a “representative” individual of each group in  $\bar{\rho}$  [12].

After updating of  $\bar{\rho}$  during an optimization iteration (*generation*), individuals from both  $\rho$  and  $\bar{\rho}$  are selected to generate a “mating pool”  $\rho'$ . This selection process is performed randomly in a manner such that the probability of an individual’s selection for the mating pool is larger for individuals with smaller fitness values. Here, “fitness” is a measure of the quality of an individual; smaller fitness values imply higher quality solutions. The *recombination operator* selects pairs of individuals (“parents”) in  $\rho'$ , and for each selected pair, two new individuals (“children”) are generated with probability  $p_r$ .

Each generated child (from recombination) undergoes a process of random modification by a *mutation operator* with probability  $p_m$ . After all recombination and mutation operations are completed on the mating pool  $\rho'$ , the resulting new population is assigned as the current population  $\rho$  for the next generation. The individuals that comprise the set  $\bar{\rho}$  after  $\mathbf{T}$  generations are the Pareto-Optimized solutions obtained by the EADI Framework. Here,  $\mathbf{T}$  is a pre-defined number of optimization iterations that is to be executed by the SPEA.

The values  $p_r$ ,  $p_m$ , and  $\mathbf{T}$  are design parameters of the optimization process that can be set through experimentation or by selecting commonly-used values from the literature.

These general concepts of fitness measures, recombination operators, and mutation operators are standard components of EAs. They are applied to form an optimization process that has analogies to processes by which living species evolve. However, these three operators need to be designed specifically for each optimization context. In the remainder of this section, we discuss how these operators have been designed in the EADI Framework.

### 5.3 Fitness Measure

Based on the SPEA approach, each individual  $\mathbf{i} \in \bar{\rho}$  is assigned a real value  $S(\mathbf{i}) \in [0, 1)$ , which is referred to as the *strength* of  $\mathbf{i}$ . If  $N$  represents the number of individuals in the set  $\rho$ , then  $S(\mathbf{i})$  is calculated as the ratio of (a) the number of individuals in  $\rho$  that are dominated by  $\mathbf{i}$  to (b)  $(N + 1)$ . The fitness of  $\mathbf{i}$  is equal to  $S(\mathbf{i})$ . The fitness of an individual  $\mathbf{i} \in \rho$  is calculated by summing the strengths of all individuals  $\mathbf{j} \in \bar{\rho}$  that dominate  $\mathbf{i}$ , and then adding one to this sum. We add one to the sum here in order to guarantee that members in  $\bar{\rho}$  have better fitness than members in  $\rho$  (since fitness is to be minimized).

### 5.4 Recombination Operator

Recombination is a process of selecting parent solutions and producing child solutions from them that integrate properties of the corresponding parent solutions. The inputs of the recombination operation are the configuration vectors of the two selected parents  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$ , and the outputs are either (a) the same two parents  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  (with probability  $(1 - p_r)$ ) or (b) the configuration vectors of two generated children (with probability  $p_r$ ), denoted by  $\mathbf{C}_1$  and  $\mathbf{C}_2$ .

In the latter case (when children are generated), the process of generating each child individual  $\mathbf{C}_k$ ,  $k = 1, 2$  from the two parents is summarized as follows: (i) assign  $P$ ,  $Q$  values (polynomial orders) from  $\mathbf{Y}_1$  or  $\mathbf{Y}_2$  to  $\mathbf{C}_k$  with equal probability subject to the requirement that the generated pair of  $P$  and  $Q$  values for  $C_1$  and  $C_2$  cannot be identical to each other; (ii) set the bit-width and filter order values of each child  $\mathbf{C}_k$  to the corresponding values of an average vector  $Y_{avg}$ :  $Y_{avg} = \gamma(Y_1, Y_2)$ , where  $\gamma(Y_1, Y_2)$  first computes the average  $(\mathbf{Y}_1 + \mathbf{Y}_2)/2$ , and for each component in this average vector that is not integer-valued, the operator replaces the component by its floor or ceiling with equal probability; and (iii) set the bit-widths and filter orders of the unused branches in the children to be zero.

### 5.5 Mutation Operator

In EAs, mutation operators are employed to help promote diversity from one generation of a population to the next by randomly modifying selected solution

components (“genes”) within individuals. In the EADI Framework for ADDA implementation, the genes for potential mutation are taken to be the vector-valued settings of **BW** and **FO**. The specific gene (**BW** or **FO**) to which modification is to be applied is selected randomly with equal probability, and then a single component of the selected vector that is to be modified is selected randomly (with equal probability among all vector components). The mutation operator replaces the value of the selected vector component with a uniform random value drawn between the given upper and lower bounds for that component.

## 6 Experimental Setup and Simulation Results

To validate the EADI Framework and ADDA platform, and to demonstrate their capabilities, we experiment with three LTE modulation schemes — QPSK, 16-QAM, and 64-QAM. The multiobjective optimization process is performed separately for each of the three modulation schemes, and then the resulting Pareto-optimized solution sets are integrated into the ADDA as discussed in Sect. 5. For all three modulation schemes, we employ the following SPEA parameter settings: (i)  $\mathbf{T} = 100$  (number of generations); (ii)  $\mathbf{N} = 50$  (population size); (iii)  $\bar{\mathbf{N}}_{\max} = 20$  (maximum size of external set); (iv)  $p_r = 0.8$  (recombination rate); (v)  $p_m = 0.2$  (mutation rate). These values for generic SPEA settings are values that are commonly used in the literature (e.g., see [10, 12]).

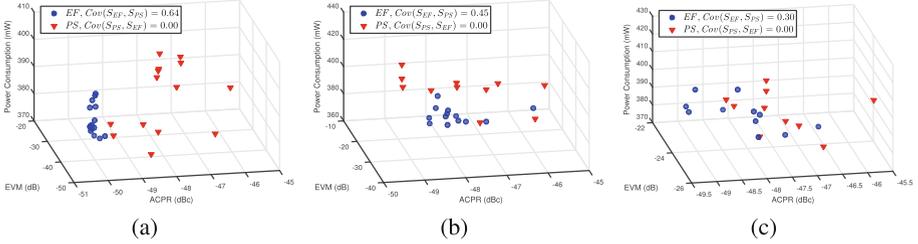
The constraint on ACPR used in the EADI Framework for all three modulations is  $-45.0$  dBc. The constraints on EVM are  $-15.1$  dB,  $-18.1$  dB, and  $-22.0$  dB for QPSK, 16-QAM, and 64-QAM, respectively. The constraint on SER is that it should be zero.

To help validate the effectiveness of the EADI Framework in deriving high quality DPD configurations, we apply a *partial search* (PS) method to solve the same multiobjective optimization problem. PS involves performing a complete search on a reduced design space. PS is also a widely-applied method for obtaining Pareto fronts in multiobjective optimization problems (e.g., see [6]).

In our PS approach, we reduce the search space by equalizing the bit-widths and filter orders of all the filters used in all branches and apply the same valid parameter value ranges as used in the SPEA process. Thus, the reduced design space contains  $5 \times 5 \times 11 \times 5 = 1375$  configurations. We evaluate these 1375 configurations exhaustively with the  $P_{FIR}$ , ACPR, SER and EVM computations, as described in Sect. 4. We then remove the undesirable solutions based on the same SER, ACPR and EVM constraints as applied in the SPEA. Finally, we collect all of the non-dominated configurations from the resulting design space as the Pareto front obtained by the PS.

In the PS process, we estimate  $P_{FIR}$  using relevant FPGA design tools (Altera PowerPlay Analyzer), while in the EADI process, we estimate  $P_{FIR}$  using the power estimator introduced in Sect. 4. The estimator of Sect. 4 enables faster power estimation (at some expense in accuracy), which is important because very large numbers of candidate solutions are evaluated during the EADI process.

For the Pareto-optimized configurations achieved by EADI, we also estimate  $P_{FIR}$  using FPGA tools to obtain more accurate power estimation results for the derived Pareto front. In the results that we report in the remainder of this section, the comparison between the quality of the two solution sets (PS and EADI) is based on the same (more accurate) power estimation method — i.e., using FPGA tools.



**Fig. 2.** Pareto-optimized solutions obtained from the EADI Framework and PS for (a) QPSK, (b) 16-QAM, (c) 64-QAM.

The Pareto fronts derived by the EADI Framework and PS for the three selected modulations are shown in Fig. 2(a) to (c). We use *coverage of two sets* ( $Cov$ ) measurements [12] to evaluate the quality of the solution sets produced by the EADI Framework and PS, which we denote by  $S_{EF}$  and  $S_{PS}$ , respectively. Given a multiobjective design space, and two sets  $\alpha$  and  $\beta$  of candidate solutions in this space,  $Cov(\alpha, \beta) = dom(\alpha, \beta)/size(\beta)$ , where  $dom(\alpha, \beta)$  is the number of solutions in  $\beta$  that are dominated by at least one solution in  $\alpha$ . Coverage results for each of the three modulation schemes are given in Fig. 2(a) to (c) along with plots of  $S_{EF}$  and  $S_{PS}$ . Here, we see that  $Cov(S_{PS}, S_{EF})$  is uniformly zero over all three modulations, while the values for  $Cov(S_{EF}, S_{PS})$  indicate that significant proportions of the PS solutions are dominated by results from the EADI Framework.

We also measured that the PS method requires approximately 91 hours to evaluate the three optimization metrics for the 1375 given configurations, and extract the Pareto front, while the evaluation and Pareto front extraction by the EADI Framework takes only about 1 hour. We conclude from these results involving  $Cov$  and optimization time that the EADI Framework significantly outperforms the PS method in terms of both the quality of the obtained Pareto fronts and run-time efficiency.

To concretely demonstrate DPD performance trade-offs realized in the proposed ADDA architecture, we first classify the individuals in the Pareto front obtained by EADI into three groups according to their power consumption levels. Then we select one representative individual in each group and store it in ADDA as a DPD working mode. The selected design vectors and their corresponding  $P_{FIR}$ -ACPR-EVM measurements under three modulations in LTE are listed in Table 1. From this table, we see that for the Pareto-optimized parameter settings

**Table 1.** Selected pareto-optimized parameter settings for LTE under different modulations. The design evaluation metrics are shown in the format  $(P_{FIR}, ACPR, EVM)$  with units (mW, dBc, dB).

	Power Level	$P, Q$	BW		FO		Performance
			Direct	Conj	Direct	Conj	
QPSK	Low	3, 1	12, 9	12	5, 3	4	380.47, -49.99, -43.41
	Medium	3, 1	13, 11	12	3, 4	4	387.57, -50.15, -43.75
	High	3, 3	15, 10	13, 5	4, 5	4, 1	397.52, -50.14, -44.07
16-QAM	Low	3, 1	11, 12	12	4, 4	5	380.96, -48.38, -24.29
	Medium	3, 3	12, 12	12, 5	3, 4	5, 3	385.34, -48.15, -30.90
	High	5, 3	12, 12, 10	11, 5	5, 4, 2	4, 4	395.11, -46.09, -31.69
64-QAM	Low	3, 1	11, 11	10	4, 3	3	382.97, -48.28, -22.89
	Medium	3, 1	12, 12	15	4, 3	4	395.76, -48.08, -25.21
	High	5, 3	12, 12, 11	14, 11	3, 3, 5	4, 4	409.62, -48.34, -24.71

obtained by EADI,  $P$  is always greater than or equal to  $Q$ , which validates the argument in [1] that the higher orders of the conjugate predistorters are weak, and a smaller  $Q$  value is therefore preferred. Also, in general, the branches corresponding to the lower polynomial orders are configured with higher bit-widths and filter orders compared to the branches corresponding to higher polynomial orders. This results from the higher order signals being relatively weak for both direct and conjugate parts.

## 7 Conclusions

In this paper, we have presented a novel framework, called the Evolutionary Adaptive DPD Implementation (EADI) Framework, for multiobjective optimization of digital predistortion (DPD) systems. The targeted optimization objectives include system energy consumption, adjacent channel power ratio (ACPR), and system accuracy. We apply the EADI Framework to develop an architecture, called the adaptive, dataflow-based DPD architecture (ADDA), where Pareto-optimized DPD parameter settings are derived to support efficient, adaptive predistorter operation. Simulation results demonstrate the effectiveness of the EADI Framework in deriving efficient DPD configurations across time-varying modulation schemes subject to multidimensional constraints. The extracted Pareto-optimized configurations also help to validate assumptions in the DPD literature about preferred DPD parameter settings. Finally, the EADI Framework is shown to significantly outperform a partial search method in terms of both optimization time efficiency and the quality of the derived Pareto fronts.

**Acknowledgements.** This research was supported in part by Tekes, the Finnish Funding Agency for Innovation; and the U.S. National Science Foundation.

## References

1. Anttila, L., Händel, P., Valkama, M.: Joint mitigation of power amplifier and I/Q modulator impairments in broadband direct-conversion transmitters. *IEEE Transactions on Microwave Theory and Techniques* **58**(4), 730–739 (2010)
2. Çiflikli, C., Yapıcı, A.: Genetic algorithm optimization of a hybrid analog/digital predistorter for RF power amplifiers. *Analog Integrated Circuits and Signal Processing* **52**(1), 25–30 (2007)
3. Ding, L., et al.: Compensation of frequency-dependent gain/phase imbalance in predistortion linearization systems. *IEEE Transactions on Circuits and Systems I: Regular Papers* **55**(1), 390–397 (2008)
4. Ghazi, A., et al.: Low power implementation of digital predistortion filter on a heterogeneous application specific multiprocessor. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 8391–8395. Florence, Italy (2014)
5. Hilborn, D.S., Stapleton, S.P., Cavers, J.K.: An adaptive direct conversion transmitter. *IEEE Transactions on Vehicular Technology* **43**(2), 223–233 (1994)
6. Llamocca, D., Pattichis, M.: Dynamic energy, performance, and accuracy optimization and management using automatically generated constraints for separable 2D FIR filtering for digital video processing. *Transactions on Reconfigurable Technology and Systems* **7**(4). Article No. 31(2015)
7. Nizamuddin, M.: Predistortion for nonlinear power amplifiers with memory. Ph.D. thesis, Virginia Polytechnic Institute and State University (2002)
8. Shen, C., Plishker, W., Wu, H., Bhattacharyya, S.S.: A lightweight dataflow approach for design and implementation of SDR systems. In: *Proceedings of the Wireless Innovation Conference and Product Exposition* (2010)
9. Sills, J.A., Sperlich, R.: Adaptive power amplifier linearization by digital predistortion using genetic algorithms. In: *Proceedings of the Radio and Wireless Conference*, pp. 229–232 (2002)
10. Sindhya, K., Miettinen, K., Deb, K.: A hybrid framework for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation* **17**(4), 495–511 (2013)
11. Wang, L.H., et al.: Dataflow modeling and design for cognitive radio networks. In: *Proceedings of the International Conference on Cognitive Radio Oriented Wireless Networks*, pp. 196–201 (2013)
12. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications. Ph.D. thesis, Swiss Federal Institute of Technology (ETH) Zurich (1999)