Re-thinking Polynomial Optimization: Efficient Programming of Reconfigurable Radio Frequency (RF) Systems by Convexification

Fa Wang, Shihui Yin, Minhee Jun, Xin Li, Tamal Mukherjee, Rohit Negi and Larry Pileggi
ECE Department, Carnegie Mellon University, Pittsburgh, PA 15213
{fwang1, syin, mjun, xinli, tamal, negi, pileggi}@ece.cmu.edu

ABSTRACT

Reconfigurable radio frequency (RF) system has emerged as a promising avenue to achieve high communication performance while adapting to versatile commercial wireless environment. In this paper, we propose a novel technique to optimally program a reconfigurable RF system in order to achieve maximum performance and/or minimum power. Our key idea is to adopt an equation-based optimization method that relies on general-purpose, non-convex polynomial performance models to determine the optimal configurations of all tunable circuit blocks. Most importantly, our proposed approach guarantees to find the globally optimal solution of the non-convex polynomial programming problem by solving a sequence of convex semi-definite programming (SDP) problems based on convexification. A reconfigurable RF front-end example designed for WLAN 802.11g demonstrates that the proposed method successfully finds the globally optimal configuration, while other traditional techniques often converge to local optima.

1. INTRODUCTION

Rapidly introduced new wireless standards and applications have presented grand challenges for wireless chip design. Reconfigurable RF system [1]-[2] has emerged as a promising avenue to achieve high communication performance while adapting to versatile wireless environment. In a reconfigurable RF system, circuit blocks (e.g., low noise amplifier, mixer, filter etc.) can be adaptively reconfigured to adjust their block-level performance metrics (e.g., gain, bandwidth, noise figure, etc.) to meet a set of system-level performance specifications (e.g., signal-to-noise ratio, bit error rate, power, etc.).

Programming a reconfigurable RF system is a critical-yet-challenging task. The objective is to find the optimal configurations of all tunable circuit blocks for the given wireless standard, spectrum condition, and system specifications. In practice, a reconfigurable RF system often carries numerous configuration options. Hence, exhaustively searching all these options is not practically feasible. Instead, we must find the optimal configuration by using a "smart" optimization algorithm.

It is important to note that most conventional analog/RF optimization techniques are ill-equipped to address the aforementioned programming problem for reconfigurable RF systems. The conventional optimization methods fall into two broad categories: (i) simulation-based [3]-[6], and (ii) equation-based [7]-[9]. The simulation-based methods rely on numerical simulations (e.g., by SPICE) to evaluate the performance metrics, and adopt stochastic optimization algorithms to avoid local optima. While these methods have been successfully applied to a number of applications, their computational cost is often prohibitively high, thereby limiting their practical utility. On the other hand, the equation-based methods use analytical performance models (i.e., design equations) for analog/RF circuit optimization. These performance models are often expressed in a special form (e.g., as convex posynomial functions [7]-[8]) so that the resulting optimization problem is convex and can be efficiently solved. However, since various simplifications are made to derive these analytical models, the equation-based methods may lead to an optimal solution that does not accurately match the actual circuit behavior.

In this paper, we propose a novel equation-based optimization method to efficiently program reconfigurable RF systems. Instead of relying on a convex model template, we adopt the general-purpose polynomials to model analog/RF performance functions. Such a choice has a two-fold implication. First, polynomial models are not necessarily convex and, hence, can accurately capture a broad range of analog/RF performance functions. In other words, our proposed polynomial models are expected to be more accurate than the conventional convex models (e.g., the posynomial models [7]-[8]) in the literature.

Second, but more importantly, since the polynomial performance models may not be convex, the resulting optimization problem is not convex in general and, hence, is non-trivial to solve [10]. To address this issue, we further borrow the convexification technique that is derived from the theory of moments and positive polynomials [11]-[16]. The key idea here is to search the globally optimal solution of a non-convex polynomial programming problem by solving a sequence of convex SDP problems. As will be demonstrated by our example of a reconfigurable RF system designed for WLAN 802.11g in Section 4, the aforementioned convexification method can find the optimal configuration both robustly (i.e., with guaranteed global optimum) and efficiently (i.e., with low computational cost).

The remainder of this paper is organized as follows. In Section 2 we briefly review the background on reconfigurable RF system, and then describe the proposed approach for polynomial modeling and optimization in Section 3. The efficacy of our proposed method is demonstrated by a reconfigurable RF system example in Section 4. Finally, we conclude in Section 5.

2. BACKGROUND

To cope with multiple wireless standards, a traditional communication system often uses multiple fixed narrow-band RF front-ends, with each front-end designed for a particular standard. However, with the overwhelming introduction of wireless standards and applications, the design cost of fixed RF front-ends has increased dramatically. In order to alleviate the design cost and improve the user experience, software-defined radio (SDR) has been proposed. Early SDRs use a wide-band RF front-end to enable the operation over a wide range of frequencies. However, this wide-band solution is vulnerable to nonlinear effects such as inter-modulation, and may lead to bad performances when large interferers are present.

978-1-4673-9569-4/16/$31.00 ©2016 IEEE

In this context, reconfigurable RF system has been proposed. An example of reconfigurable RF receiver, as shown in Figure 1, is composed of tunable RF/IF filters, low noise amplifier (LNA), mixer and local oscillator (LO). The RF/IF filters can be configured to select the desired frequency bands. The LO can be tuned to adjust the operating frequency. The LNA and mixer can be configured to achieve the optimal system-level performances (e.g. signal-to-noise ratio, power, etc.) by leveraging the tradeoffs between different block-level performances (e.g. linearity vs. gain). The optimal configuration for each circuit block may vary dramatically under different environmental conditions and applications. For example, with or without the presence of strong interference, the required linearity can be substantially different for the LNA.

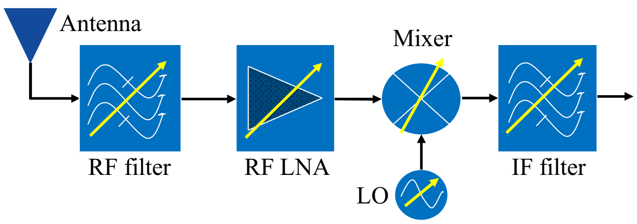

Figure 1. An example of reconfigurable RF front-end is shown, including two filters, an LNA, a mixer and an LO.

Programming a reconfigurable RF system is one of the most important tasks in order to maximally exploit the benefit of its reconfigurability. Here, the objective of programming is to find the optimal configurations of all tunable circuit blocks for the given wireless standard, spectrum condition, and system specifications. Mathematically, we formulate such a programming problem as the following optimization problem:

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & g_m(\mathbf{x}) \le G_m \quad (m = 1, 2, \cdots, M), \\ & \mathbf{x} \in S \end{array} \tag{1}$$

where

$$\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_N]^T \tag{2}$$

contains the tuning knobs for the reconfigurable system, $f(\mathbf{x})$ denotes the system-level performance (e.g. power consumption) that should be minimized, $\{g_m(\mathbf{x}); m = 1, 2, \cdots, M\}$ stands for the system-level performances (e.g. signal-to-noise ratio) that are constrained by their specifications $\{G_m; m = 1, 2, \cdots, M\}$, and the set S defines the lower and upper bounds of $\mathbf{x}$ (i.e., the tuning ranges of these knobs):

$$S = \left\{ \mathbf{x} \middle| l_n \le x_n \le u_n \quad (n = 1, 2, \cdots, N) \right\}. \tag{3}$$

In general, the optimization problem in (1) is non-convex, because the cost function $f(\mathbf{x})$ and/or the constraint functions $\{g_m(\mathbf{x}); m = 1, 2, \cdots, M\}$ may not be convex. Finding the globally optimal solution of a non-convex optimization problem has been considered as a grant challenge in the literature. There is no elegant solution that has been proposed yet. In what follows, we will adopt the recent breakthrough on polynomial programming from the mathematic community to develop a novel CAD methodology that addresses this fundamental challenge.

3. PROPOSED APPORACH

To efficiently program a reconfigurable RF system, we propose the following two-step flow: (i) polynomial performance modeling, and (ii) polynomial programming. To solve the optimization in (1), we first approximate $f(\mathbf{x})$ and $\{g_m(\mathbf{x}); m = 1, 2, \cdots, M\}$ by polynomial functions:

$$f(\mathbf{x}) \approx \sum_{r=1}^{R} \alpha_r \cdot x_1^{\theta_1(r)} x_2^{\theta_2(r)} \cdots x_N^{\theta_N(r)} \tag{4}$$

$$g_m(\mathbf{x}) \approx \sum_{r=1}^{R} \beta_{m,r} \cdot x_1^{\theta_1(r)} x_2^{\theta_2(r)} \cdots x_N^{\theta_N(r)} \quad (m = 1, 2, \cdots, M), \tag{5}$$

where

$$\sum_{n=1}^{N} \theta_n(r) \le t. \tag{6}$$

In (4)-(6), $\{\alpha_r; r = 1, 2, \cdots, R\}$ contains the model coefficients of $f(\mathbf{x})$, $\{\beta_{m,r}; r = 1, 2, \cdots, R\}$ contains the model coefficients of $g_m(\mathbf{x})$, $\{\theta_n(r); n = 1, 2, \cdots, N\}$ defines the non-negative integer exponents of the r-th polynomial term, R denotes the total number of polynomial terms, and t stands for the degree of these polynomial functions. The model coefficients $\{\alpha_r; r = 1, 2, \cdots, R\}$ and $\{\beta_{m,r}; m = 1, 2, \cdots, M; r = 1, 2, \cdots, R\}$ can be fitted by various performance modeling techniques [17].

Once the polynomial performance models $f(\mathbf{x})$ and $\{g_m(\mathbf{x}); m = 1, 2, \cdots, M\}$ are available, solving the polynomial programming problem in (1), however, is still not trivial. The grand challenge stems from the fact that the polynomial functions $f(\mathbf{x})$ and $\{g_m(\mathbf{x}); m = 1, 2, \cdots, M\}$ are not necessarily convex and, hence, the optimization problem in (1) is not convex. It has been shown in the literature that a general polynomial programming problem is NP-hard [15]. In this paper, we will adopt a novel convexification technique to find the globally optimal solution of (1) by solving a sequence of convex SDP problems. In other words, the convexification technique can solve a non-convex polynomial programming problem both efficiently (i.e., with low computational cost) and robustly (i.e., with guaranteed global optimum) in practice. In what follows, we will discuss the convexification procedures for the cost function and the constraints in (1), respectively.

3.1 Convexifying Polynomial Cost Function

To start with, we convert the cost function in (1) to a new representation based on a probability density function (PDF):

$$\min_{\mu(\mathbf{x})} \quad \int f(\mathbf{x}) \cdot \mu(\mathbf{x}) \cdot d\mathbf{x}. \tag{7}$$

where $\mu(\mathbf{x})$ is a multivariate PDF. To understand (7), we re-write the integration in (7) as:

$$\int f(\mathbf{x}) \cdot \mu(\mathbf{x}) \cdot d\mathbf{x} \approx \sum_i f(\mathbf{x}_i) \cdot \mu(\mathbf{x}_i) \cdot \Delta\mathbf{x}, \tag{8}$$

where $\Delta\mathbf{x}$ is sufficiently small and

$$\sum_i \mu(\mathbf{x}_i) \cdot \Delta\mathbf{x} \approx 1. \tag{9}$$

In other words, the integration in (7) is a weighted sum of $f(\mathbf{x})$ over all $\Delta\mathbf{x}$'s, where the weights are determined by $\mu(\mathbf{x})$. Let f^* be the optimal value of $f(\mathbf{x})$, and $\mathbf{x}^*$ be the corresponding $\mathbf{x}$ value. (Here, we assume that only a single global optimum $\mathbf{x}^*$ exists). To reach the minimum of the integration, one can construct an optimal PDF $\mu^*(\mathbf{x})$ that is non-zero at $\mathbf{x}^*$ only. Mathematically, we can represent such an optimal $\mu^*(\mathbf{x})$ as:

$$\mu^*(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}^*), \tag{10}$$

where

$$\delta(\mathbf{x} - \mathbf{x}^*) = \begin{cases} +\infty, & \mathbf{x} = \mathbf{x}^* \\ 0, & \mathbf{x} \ne \mathbf{x}^* \end{cases} \tag{11}$$

$$\int \delta\left(\mathbf{x}-\mathbf{x}^*\right)\cdot d\mathbf{x} = 1 . \tag{12}$$

A simple one-dimensional example is shown in Figure 2 for illustration purposes.

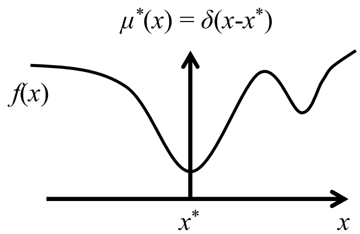

Figure 2. A one-dimensional example of the optimal PDF $\mu^*(x)$ is shown for illustration purposes.

Substituting (4) into the integration in (7) yields:

$$\begin{aligned}\int f(\mathbf{x})\mu(\mathbf{x})d\mathbf{x} &= \int\left[\sum_{r=1}^{R}\alpha_r x_1^{\theta_1(r)}x_2^{\theta_2(r)}\cdots x_N^{\theta_N(r)}\right]\mu(\mathbf{x})d\mathbf{x} \\ &= \sum_{r=1}^{R}\alpha_r\left[\int x_1^{\theta_1(r)}x_2^{\theta_2(r)}\cdots x_N^{\theta_N(r)}\mu(\mathbf{x})d\mathbf{x}\right]\end{aligned} . \tag{13}$$

Define the moments:

$$y_r = \int\left[x_1^{\theta_1(r)}x_2^{\theta_2(r)}\cdots x_N^{\theta_N(r)}\mu(\mathbf{x})\right]d\mathbf{x} \tag{14}$$

and the moment vector:

$$\mathbf{y}_R = \begin{bmatrix} y_1 & y_2 & \cdots & y_R\end{bmatrix}^T . \tag{15}$$

Without loss of generality, we extend the moment sequence to infinity:

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots\end{bmatrix}^T . \tag{16}$$

Given (7) and (13)-(16), we can write out a moment representation for the cost function:

$$\min_{\mathbf{y}} \quad e(\mathbf{y}) = \sum_{r=1}^{R}\alpha_r \cdot y_r . \tag{17}$$

Note that the moments are considered as the problem unknowns and the new cost function is a linear combination of these moments. Thus the cost function in (17) is convex with respect to the unknown moments.

It should be noted that the moment sequence $\mathbf{y}$ in (16) cannot take arbitrary values, because a valid PDF function $\mu(\mathbf{x})$ must exist to generate the sequence. Hence, additional constraints must be posed for the moment values. For example, in a one-dimensional case, a negative second-order moment is not valid. In what follows, we will further discuss how to add extra constraints for the moment sequence.

Consider a polynomial $q(\mathbf{x})$ with degree of d:

$$q(\mathbf{x}) = \sum_{s=1}^{S} q_s \cdot b_s(\mathbf{x}) , \tag{18}$$

where

$$\begin{gathered} b_s(\mathbf{x}) = x_1^{\theta_1(s)}x_2^{\theta_2(s)}\cdots x_N^{\theta_N(s)} \\ (s = 1, 2, \cdots, S)\end{gathered} \tag{19}$$

$$\sum_{n=1}^{N}\theta_n(s) \le d . \tag{20}$$

In (18)-(20), S is the total number of polynomial terms, and $\{q_s;\ s = 1, 2, \cdots, S\}$ stands for the polynomial coefficients. We define the coefficient vector of $q(\mathbf{x})$ as:

$$\mathbf{q} = \begin{bmatrix} q_1 & q_2 & \cdots & q_S\end{bmatrix}^T \tag{21}$$

and the basis function vector as:

$$\mathbf{b}(\mathbf{x}) = [b_1(\mathbf{x}) \quad b_2(\mathbf{x}) \quad \cdots \quad b_S(\mathbf{x})]^T . \tag{22}$$

Thus we have:

$$q(\mathbf{x}) = \mathbf{q}^T \cdot \mathbf{b}(\mathbf{x}) . \tag{23}$$

Given that both $q(\mathbf{x})^2$ and $\mu(\mathbf{x})$ must be non-negative, we can easily derive the following constraint:

$$\int q(\mathbf{x})^2 \cdot \mu(\mathbf{x}) \cdot d\mathbf{x} \ge 0 . \tag{24}$$

Substituting (23) into (24) yields:

$$\begin{aligned}&\int q(\mathbf{x})^2 \cdot \mu(\mathbf{x}) \cdot d\mathbf{x} = \int \mathbf{q}^T \cdot \mathbf{b}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x})^T \cdot \mathbf{q} \cdot \mu(\mathbf{x}) \cdot d\mathbf{x} \\ &= \mathbf{q}^T \cdot \left[\int \mathbf{b}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x})^T \cdot \mu(\mathbf{x}) \cdot d\mathbf{x}\right] \cdot \mathbf{q} \ge 0\end{aligned} . \tag{25}$$

The integration in (25) is a moment matrix:

$$C_d(\mathbf{y}) = \int \mathbf{b}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x})^T \cdot \mu(\mathbf{x}) \cdot d\mathbf{x} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1S} \\ c_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \\ c_{S1} & \cdots & & c_{SS}\end{bmatrix} , \tag{26}$$

where

$$c_{ij} = \int b_i(\mathbf{x}) \cdot b_j(\mathbf{x}) \cdot \mu(\mathbf{x}) \cdot d\mathbf{x} \quad (i, j = 1, 2, \cdots, S) . \tag{27}$$

In other words, each entry in the matrix is a moment, and can be found in the moment sequence $\mathbf{y}$. Combining (25) and (26), we have the following constraint:

$$\mathbf{q}^T \cdot C_d(\mathbf{y}) \cdot \mathbf{q} \ge 0 . \tag{28}$$

For any polynomial $q(\mathbf{x})$ with arbitrary values of $\mathbf{q}$ and d, the inequality constraint in (25) must be satisfied. Hence, the moment matrix $C_d(\mathbf{y})$ must be positive semi-definite:

$$C_d(\mathbf{y}) \succ= 0 \quad (d = 1, 2, \cdots) . \tag{29}$$

To help understand the aforementioned mathematical concepts, we now consider a simple one-dimensional example where the cost function is given as:

$$\min_x \quad f(x) = \alpha_1 + \alpha_2 \cdot x + \alpha_3 \cdot x^2 . \tag{30}$$

According to (7) and (13)-(17), the cost function in (30) can be converted to a moment representation:

$$\min_{\mathbf{y}} \quad e(\mathbf{y}) = \alpha_1 \cdot y_1 + \alpha_2 \cdot y_2 + \alpha_3 \cdot y_3 , \tag{31}$$

where

$$y_r = \int x^{r-1} \cdot \mu(x) \cdot dx \quad (r = 1, 2, 3) . \tag{32}$$

We then set up the constraints based on (18)-(29). For $q(x)$ with degree of $d = 1$:

$$q(x) = q_1 + q_2 \cdot x , \tag{33}$$

we can enforce the linear matrix inequality (LMI) constraint:

$$C_1(\mathbf{y}) \succ= 0 , \tag{34}$$

where

$$C_1(\mathbf{y}) = \begin{bmatrix} y_1 & y_2 \\ y_2 & y_3\end{bmatrix} . \tag{35}$$

As the order d increases, we can get a sequence of the LMI constraints shown in (29).

3.2 Convexifying Polynomial Constraints

Note that the constraints in (1) are polynomial functions. Hence, we can re-write these constraints in a standard form:

$$\hat{g}_m(\mathbf{x}) \ge 0 \quad (m = 1, 2, \cdots, M + N) , \tag{36}$$

where

$$\hat{g}_m(\mathbf{x})=\begin{cases} G_m-g_m(\mathbf{x}) & (m=1,2,\cdots,M) \\ (x_{m-M}-l_{m-M})\cdot(u_{m-M}-x_{m-M}) & (m=M+1,\cdots,M+N) \end{cases}. \quad (37)$$

The polynomial functions $\{\hat{g}_m(\mathbf{x});\ m = 1, 2, \cdots, M + N\}$ can be expressed as:

$$\hat{g}_m(\mathbf{x})=\sum_{r=1}^{R}\omega_{m,r}\cdot b_r(\mathbf{x}) \quad (m=1,2,\cdots,M+N), \quad (38)$$

where $\{\omega_{m,r};\ m = 1, 2, \cdots, M + N;\ r = 1, 2, \cdots, R\}$ contains the polynomial coefficients and

$$b_r(\mathbf{x})=x_1^{\theta_1(r)}x_2^{\theta_2(r)}\cdots x_N^{\theta_N(r)} \quad (r=1,2,\cdots,R) \quad (39)$$

are the basis functions.

In (29), a set of constraints have been added to the moment sequence $\mathbf{y}$ without considering the optimization constraints in (36). In what follows, we will discuss how to further bound the moment sequence $\mathbf{y}$ based on these optimization constraints. We define a polynomial $q(\mathbf{x})$ with degree of d, as shown in (18)-(23). Given that $q(\mathbf{x})^2$, $\mu(\mathbf{x})$, and $\{\hat{g}_m(\mathbf{x});\ m = 1, 2, \cdots, M + N\}$ must be non-negative, we derive the following constraints:

$$\int\hat{g}_m(\mathbf{x})\cdot q(\mathbf{x})^2\cdot\mu(\mathbf{x})\cdot d\mathbf{x}\geq 0 \quad (m=1,2,\cdots,M+N). \quad (40)$$

The constraints in (40) enforce that $\mu(\mathbf{x})$ can only take non-zero values where $\hat{g}_m(\mathbf{x})$ is non-negative. If there exists $\mathbf{x}_v$ such that $\mu(\mathbf{x}_v) > 0$ and $\hat{g}_m(\mathbf{x}_v) < 0$, we have $\hat{g}_m(\mathbf{x}_v)\cdot\mu(\mathbf{x}_v) < 0$. In this case, one can easily construct a polynomial $q_v(\mathbf{x})$ such that $q_v(\mathbf{x})$ is peaked around $\mathbf{x}_v$ and the integration in (40) is negative.

Substituting (23) into (40) yields:

$$\begin{aligned} &\int\hat{g}_m(\mathbf{x})\cdot q(\mathbf{x})^2\cdot\mu(\mathbf{x})\cdot d\mathbf{x} \\ &=\int\hat{g}_m(\mathbf{x})\cdot\mathbf{q}^T\cdot\mathbf{b}(\mathbf{x})\cdot\mathbf{b}^T(\mathbf{x})\cdot\mathbf{q}\cdot\mu(\mathbf{x})\cdot d\mathbf{x} \qquad (m=1,2,\cdots,M+N) \\ &=\mathbf{q}^T\cdot\left[\int\hat{g}_m(\mathbf{x})\cdot\mathbf{b}(\mathbf{x})\cdot\mathbf{b}^T(\mathbf{x})\cdot\mu(\mathbf{x})\cdot d\mathbf{x}\right]\cdot\mathbf{q}\geq 0 \end{aligned} . \quad (41)$$

The integration in (41) is a moment matrix:

$$\hat{C}_{m,d}(\mathbf{y})=\int\hat{g}_m(\mathbf{x})\cdot\mathbf{b}(\mathbf{x})\cdot\mathbf{b}(\mathbf{x})^T\cdot\mu(\mathbf{x})\cdot d\mathbf{x}=\begin{bmatrix} \hat{c}_{11} & \hat{c}_{12} & \cdots & \hat{c}_{1S} \\ \hat{c}_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \\ \hat{c}_{S1} & \cdots & & \hat{c}_{SS} \end{bmatrix}, \quad (42)$$

where

$$\hat{c}_{ij}=\int\hat{g}_m(\mathbf{x})\cdot b_i(\mathbf{x})\cdot b_j(\mathbf{x})\cdot\mu(\mathbf{x})\cdot d\mathbf{x} \quad (i,j=1,2,\cdots,S). \quad (43)$$

Substituting (38) into (43), we have:

$$\begin{aligned} \hat{c}_{ij}&=\int\left[\sum_{r=1}^{R}\omega_{m,r}\cdot b_r(\mathbf{x})\right]\cdot b_i(\mathbf{x})\cdot b_j(\mathbf{x})\cdot\mu(\mathbf{x})\cdot d\mathbf{x} \\ &=\sum_{r=1}^{R}\omega_{m,r}\cdot\left[\int b_r(\mathbf{x})\cdot b_i(\mathbf{x})\cdot b_j(\mathbf{x})\cdot\mu(\mathbf{x})\cdot d\mathbf{x}\right]. \\ &(m=1,2,\cdots,M+N) \quad (i,j=1,2,\cdots,S) \end{aligned} \quad (44)$$

In other words, each element in the moment matrix (42) is a linear combination of several moments. Combining (41) and (42), we can derive the following constraints:

$$\mathbf{q}^T\cdot\hat{C}_{m,d}(\mathbf{y})\cdot\mathbf{q}\geq 0 \quad (m=1,2,\cdots,M+N). \quad (45)$$

For any polynomial $q(\mathbf{x})$ with arbitrary values of $\mathbf{q}$ and d, the inequality constraints in (45) must be satisfied. Therefore, the moment matrix $\hat{C}_{m,d}(\mathbf{y})$ must be positive semi-definite:

$$\hat{C}_{m,d}(\mathbf{y})\succ=0 \quad (m=1,2,\cdots,M+N) \quad (d=1,2,\cdots). \quad (46)$$

To help understand the LMI constraints in (46), we now consider a simple one-dimensional example with the following constraint:

$$\hat{g}_1(x)=x\geq 0. \quad (47)$$

For a polynomial $q(\mathrm{x})$ with degree $d = 1$:

$$q(x)=q_1+q_2\cdot x, \quad (48)$$

we enforce the LMI constraint:

$$\hat{C}_{1,1}(\mathbf{y})\succ=0, \quad (49)$$

where

$$\hat{C}_{1,1}(\mathbf{y})=\begin{bmatrix} y_2 & y_3 \\ y_3 & y_4 \end{bmatrix} \quad (50)$$

$$y_r=\int x^{r-1}\cdot\mu(x)\cdot dx \quad (r=2,3,4). \quad (51)$$

As the order d increases, we can further get a sequence of the LMI constraints:

$$\hat{C}_{1,d}(\mathbf{y})\succ=0 \quad (d=1,2,\cdots). \quad (52)$$

3.3 Sequential Semidefinite Programming

Combining (17), (29) and (46), we can formulate a sequence of SDP problems corresponding to different polynomial degrees:

$$\mathbf{H}^d:\begin{cases} \min_{\mathbf{y}} & e(\mathbf{y})=\sum_{r=1}^{R}\alpha_r\cdot y_r \\ \text{s.t.} & C_d(\mathbf{y})\succ=0 \\ & \hat{C}_{m,d}(\mathbf{y})\succ=0 \quad (m=1,2,\cdots,M+N) \end{cases}. \quad (53)$$
$$(d=1,2,\cdots)$$

Let $\{h^d;\ d = 1, 2, \cdots\}$ be the optimal values of the cost functions of $\{\mathbf{H}^d;\ d = 1, 2, \cdots\}$. Note that the sequence $\{h^d;\ d = 1, 2, \cdots\}$ satisfies the following inequalities:

$$h^d\leq h^{d+1} \quad (d=1,2,\cdots), \quad (54)$$

because $C_d(\mathbf{y})$ and $\hat{C}_{m,d}(\mathbf{y})$ are the principal sub-matrices of $C_{d+1}(\mathbf{y})$ and $\hat{C}_{m,d+1}(\mathbf{y})$ respectively and, hence, the constraint set of $\mathbf{H}^{d+1}$ is a subset of that of $\mathbf{H}^d$.

Let $\mathbf{x}^*$ be the global optimum of the optimization in (1) and $f^* = f(\mathbf{x}^*)$ be the corresponding cost function value. It has been proven in [11] that: (i) the SDP problems $\{\mathbf{H}^d;\ d = 1, 2, \cdots\}$ are feasible, and (ii) the sequence $\{h^d;\ d = 1, 2, \cdots\}$ asymptotically converges to f^*. Considering both the inequalities in (54) and the aforementioned convergence property, we have:

$$h^1\leq h^2\leq\cdots\leq f^*. \quad (55)$$

It is further shown in [11]-[12] that there exists a finite value D such that:

$$h^1\leq h^2\leq\cdots\leq h^D=h^{D+1}=\cdots=f^*, \quad (56)$$

where D is determined by the inherent complexity of (1). Once the optimal moment sequence $\mathbf{y}^*$ is solved from $\mathbf{H}^D$, the global optimum $\{x_n^*;\ n = 1, 2, \cdots, N\}$ of (1) is determined by the first-order moments $\{y_n^*;\ n = 1, 2, \cdots, N\}$:

$$x_n^*=\int x_n\cdot\mu^*(\mathbf{x})\cdot d\mathbf{x}=y_n^* \quad (n=1,2,\cdots,N). \quad (57)$$

Since D is unknown in practice, a simple criterion must be derived to determine its value. Let $\{\mathbf{y}^d;\ d = 1, 2, \cdots\}$ be the optimal moment sequences solved from $\{\mathbf{H}^d;\ d = 1, 2, \cdots\}$, and $\{\mathbf{x}^d;\ d = 1, 2, \cdots\}$ be the first-order moments calculated by (57). A given value d is equal to D, if the following conditions hold:

$$\hat{g}_m(\mathbf{x}^d)\geq 0 \quad (m=1,2,\cdots,M+N) \quad (58)$$

and

$$f(\mathbf{x}^d)=h^d. \quad (59)$$

To understand (58)-(59), we consider (55) and the definition of f^*:

$$h^d \le f^* \le f\left(\mathbf{x}^d\right) \ \ where\ \ \hat{g}_m\left(\mathbf{x}^d\right) \ge 0 \ \ \left(m = 1, 2, \cdots, M+N\right). \quad (60)$$

Combining (58)-(60) yields:

$$h^d = f^* = f\left(\mathbf{x}^d\right). \quad (61)$$

Therefore, the conditions in (58)-(59) guarantee that $\mathbf{x}^d$ is the global optimum of (1). More details about the SDP formulation can be found in [11].

3.4 Summary

Algorithm 1 summarizes the major steps of our proposed method for programming RF systems based on convexification. It consists of two major steps: (i) performance modeling and (ii) sequential SDP. The efficacy of the proposed method will be further demonstrated by our numerical examples in the next section.

Algorithm 1: Programming of RF System by Convexification

1. For the cost function $f(\mathbf{x})$ and the constraint functions $\{g_m(\mathbf{x});\ m = 1, 2, \cdots, M\}$ in (1), fit the polynomial performance models in (4)-(5).
2. Initialize $d = 1$.
3. Solve the SDP problem $\mathbf{H}^d$ in (53) to obtain h^d and $\mathbf{y}^d$.
4. Calculate $\mathbf{x}^d$ using (57).
5. Check the conditions in (58)-(59). If these conditions are satisfied, take $\mathbf{x}^d$ as the global optimum of (1). Otherwise, set $d = d + 1$ and go to step 3.

4. NUMERICAL EXPERIMENTS

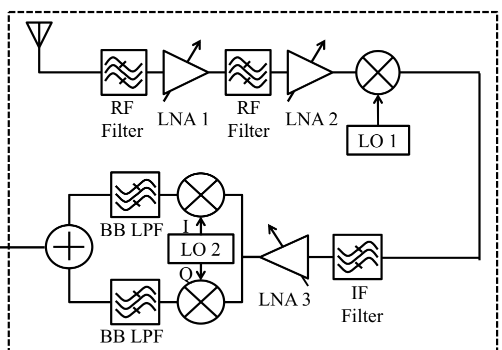

Figure 3. The simplified block diagram is shown for a reconfigurable RF front-end designed for WLAN 802.11g.

Table 1. Polynomial modeling results for SNR

Model order	2	3	4
# of Coefficients	28	148	411
Maximum Error (dB)	3.03	1.91	1.81
Average Error (dB)	1.81	1.01	0.69

In this section, a reconfigurable RF front-end is used to demonstrate the efficacy of the proposed RF system programming algorithm. The RF front-end is designed for the wireless communication standard WLAN 802.11g, where the center frequency is 2.4GHz. It is composed of three tunable LNAs, as shown in Figure 3. Each LNA consists of three stages where the bias current of each stage can be independently tuned. We consider the total power consumption (Power) of the three LNAs and the signal-to-noise ratio (SNR) of the entire front-end as the performances of interest. In this example, the objective of programming is to find the optimal bias currents for all LNAs so that the power consumption is minimized subject to a given SNR specification. Our numerical experiments are performed on a 3.4GHz Linux server.

Towards this goal, we first build the polynomial performance models for SNR and Power. The power consumption can be approximated as an analytical function of the bias currents based on hand analysis. To fit the SNR model, the RF front-end is simulated by MATLAB SIMULINK and 800 sampling points are collected. Next, a polynomial model with nine variables (i.e., three variables for each LNA) is fitted for SNR by using the sparse regression method [17]. Table 1 summarizes the modeling error for SNR, as the polynomial order varies from two to four. Based on the results in Table 1, we conclude that a 4th-order polynomial model is sufficiently accurate for SNR in this example.

Table 2. Optimization results of different methods (SNR ≥ 13dB)

	Fitted SNR (dB)	Simulated SNR (dB)	Power (mW)
IP	14.71	14.35	11.97
SA	14.72	14.35	11.99
Proposed	14.71	14.35	11.97

Table 3. Optimization results of different methods (SNR ≥ 15dB)

	Fitted SNR (dB)	Simulated SNR (dB)	Power (mW)
IP	15.00	15.58	15.48
SA	15.00	14.99	13.49
Proposed	15.00	15.04	12.58

Table 4. Optimization results of different methods (SNR ≥ 17dB)

	Fitted SNR (dB)	Simulated SNR (dB)	Power (mW)
IP	17.00	16.05	31.48
SA	17.00	16.61	21.95
Proposed	17.00	17.30	19.24

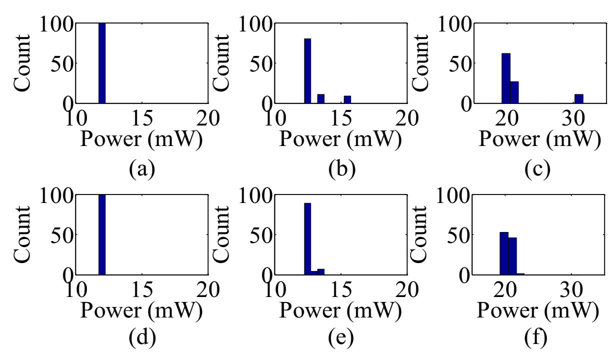

Figure 4. The optimization results are shown for 100 independent runs with randomly selected initial guess. (a), (b) and (c) show the results by IP with the SNR specification set to 13dB, 15dB and 17dB, respectively. (d), (e) and (f) show the results by SA with the SNR specification set to 13dB, 15dB and 17dB, respectively.

Once the polynomial performance models are available, we apply three different methods to solve the optimization problem in (1) for testing and comparison purposes: (i) the interior-point method (IP) [10], (ii) the simulated annealing method (SA) [3], and (iii) the proposed method based on convexification (Proposed). The optimization parameters are carefully tuned for SA to achieve good results. When implementing Algorithm 1, the

SDP problems are solved by a commercial optimization tool MOSEK.

Table 2-Table 4 compare the optimization results for three different methods where the SNR specification is set to different values. Note that the power consumption achieved by our proposed method is less than those by IP and SA in most cases. It demonstrates that the proposed method is able to find an optimal solution that is superior over those solved by IP and SA. In particular, as the SNR specification becomes tight, it is increasingly difficult for IP and SA to converge to a good solution with low power consumption.

To further study the robustness for IP and SA, Figure 4 summarizes their optimization results for 100 independent runs with randomly selected initial guess. Since neither IP nor SA guarantees to find the global optimum, their optimization results depend on the initial guess. In most cases, only a local optimum is found by IP or SA.

Table 5. Runtime comparison for different methods

SNR Spec (dB)	Model Fitting (Sec.)	Optimization (Sec.)		
		IP	SA	Proposed
13.00	3.852×10^5	0.10	76	2.4
15.00		0.11	76	67
17.00		0.21	76	76

Table 5 compares the runtime for three different methods. Here, the computational cost consists of two major components: (i) model fitting cost (i.e., the cost of generating sampling points by SIMULINK and solving polynomial model coefficients), and (ii) optimization cost (i.e., the cost of solving the optimization problem in (1) based on polynomial performance models). As shown in Table 5, the computational cost is dominated by model fitting in this example. Since the three optimization methods rely on the same polynomial performance models, the overall computational cost is similar for these three methods.

It should be noted that the complexity of the SDP problem in (53) grows quickly with the number of basis functions used for performance modeling. Therefore, performance models must be carefully simplified in order to apply our proposed convexification approach to practical applications.

5. CONCLUSIONS

In this paper, a novel equation-based optimization method based on polynomial programming is proposed for efficient programming of reconfigurable RF systems. The proposed method first builds polynomial performance models to capture analog/RF performance functions. Next, a non-convex polynomial programming problem is formulated and its globally optimal solution is efficiently found by solving a sequence of convex SDP problems based on convexification. As is demonstrated by a reconfigurable RF front-end example designed for WLAN 802.11g, the proposed method guarantees to find the globally optimal configuration, while other traditional techniques often converge to local optima. In our future work, we will further apply the proposed method to program large-scale reconfigurable RF systems in silicon.

6. ACKNOWLEDGEMENTS

This work is sponsored in part by the DARPA RF-FPGA (Radio Frequency-Field Programmable Gate Arrays) program under Grant HR0011-12-1-0005 and the National Science Foundation. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

REFERENCES

[1] H. Darabi, A. Mirzaei and M. Mikhemar, "Highly integrated and tunable RF front ends for reconfigurable multiband transceivers: A tutorial," *IEEE Trans. on CAS-I*, vol. 58, no. 9, pp. 2038-2050, Aug. 2011.

[2] J. Craninckx, M. Liu, D. Hauspie, V. Giannini, T. Kim, J. Lee, M. Liois, B. Debaillie, C. Soens, M. Ingels, A. Baschirotto, J. Van Driessche, L. Van der Perre and P. Vanbekbergen, "A fully reconfigurable software-defined radio transceiver in 0.13um CMOS," *IEEE ISSCC*, pp. 346-607, 2007.

[3] G. Gielen, H. Walscharts and W. Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing," *IEEE JSSC*, vol. 25, no. 3, pp. 707-713, Jun. 1990.

[4] M. Krasnicki, R. Phelps, R. Rutenbar and R. Carley, "MAELSTROM: Efficient simulation-based synthesis for custom analog cells," *IEEE DAC*, pp. 945-950, 1999.

[5] J. Tao, Y. Wang, M. Jun, X. Li, R. Negi, T. Mukherjee and L. Pileggi, "Toward efficient programming of reconfigurable radio frequency (RF) receivers," *IEEE ASP-DAC*, pp. 256-261, 2014.

[6] H. Habal and H. Graeb, "Constraint-based layout-driven sizing of analog circuit," *IEEE Trans. on CAD*, vol. 30, no. 8, pp. 1089-1102, Aug. 2011.

[7] M. Hershenson, S. Boyd and T. Lee, "GPCAD: a tool for CMOS op-amp synthesis," *IEEE ICCAD*, pp. 296-303, 1998.

[8] W. Daems, G. Gielen and W. Sansen, "Simulated-based automatic generation of signomial and posynomial performance models for analog integrated circuit sizing," *IEEE ICCAD*, pp. 70-74, 2001.

[9] G. Yu and P. Li, "Yield-aware hierarchical optimization of large analog integrated circuits," *IEEE ICCAD*, pp. 79-84, 2008.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[11] J. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, no. 3, pp. 796-817, Jan. 2001.

[12] D. Bertsimas and C. Caramanis, "Geometry, moments, and semidefinite optimization," *MTNS*, pp. 1-15, 2002.

[13] J. Lasserre, "Convergent SDP-relaxations in polynomial optimization with sparsity," *SIAM J. Optim.*, vol. 17, no. 3, pp. 822-843, Apr. 2006.

[14] Z. Luo and S. Zhang, "A semidefinite relaxation scheme for multivariate quartic polynomial optimization with quadratic constraints," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1716-1736, Jan. 2010.

[15] G. Blekherman, P. Parrilo and R. Thomas, *Semidefinite Optimization and Convex Algebraic Geometry*, SIAM Publications, 2012.

[16] Y. Wang, M. Orshansky and C. Caramanis, "Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization," *IEEE DAC*, pp. 1-6, 2014.

[17] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance modeling of analog/RF circuits," *IEEE Trans. on CAD*, vol. 29, no. 11, pp. 1661-1668, Nov. 2010.