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
Carnegie Mellon University, Pittsburgh, PA 15213 USA
Rapidly introduced wireless standards and applications pose grand challenges for wireless chip design

- High cost in designing multiple fixed narrow-band RF front-ends
- Nonlinearity problems for wide band RF front-end

In this context, reconfigurable RF system has been proposed

- Introduce tunable control knobs in circuit blocks so that the performances can be adaptively changed

The optimal configuration of RF systems may change dramatically under different environmental conditions.

Channel variation and tuning requirement

- Strong signal
  - Weak blocker
- Weak signal
  - Weak blocker
- Weak signal
  - Strong blocker

LNA / receiver

Requirement: Gain ↓ NF ↑ IIP3 ↓ Power ↓
Gain ↑NF ↓ IIP3 ↓ Power ↓
Gain ↑NF ↓ IIP3 ↑ Power (max)

Programming a reconfigurable RF system is an important task in order to maximally exploit the benefit of its reconfigurability.

Traditional Optimization Approach

- **Simulated annealing**
  - **Idea**: Relies on numerical simulations (e.g. by SPICE) to evaluate the performance metrics, and adopt stochastic optimization algorithms to avoid local optima

- **Pros**: Can avoid trapping in undesirable local optimum
- **Cons**: Can be very time consuming

Traditional Optimization Approach

- **Geometric programming**
  - Builds performance model in a special form which results in a convex optimization problem
  - Applies convex optimization to efficiently solve the optimization

\[
\begin{align*}
\text{Performance}_1 &= f_1(\text{Parameters}) \\
&\vdots \\
\text{Performance}_K &= f_K(\text{Parameters})
\end{align*}
\]

- **Pros**: can capture global behavior using performance model; convex optimization is easy to solve
- **Cons**: relies on the accuracy of model template; the optimal solution may not accurately match the actual circuit behavior

Challenges

- The traditional design optimization techniques are still ill-equipped to program a reconfigurable RF system, due to:
  - The high cost of performance evaluations
  - Convex limitation of model template

Optimization Methods

- Performance model driven
  - Proposed
  - Convex optimization

We propose a performance model driven optimization method based on general purpose polynomial model template

The Proposed Flow

**Two Steps**

- Polynomial performance modeling
- Polynomial optimization based on convexification

Simulation samples are collected

Polynomial performance model

Polynomial optimization

\[
\begin{align*}
\min_{\mathbf{x}} & \quad f(\mathbf{x}) \\
\text{s.t.} & \quad g_m(\mathbf{x}) \leq G_m \quad (m = 1, 2, \ldots) \\
& \quad \mathbf{x} \in S
\end{align*}
\]

Control knob values

Additional linear constraints on \( \mathbf{x} \)

**Polynomial performance cost function**

**Polynomial performance constraints**
Polynomial Performance Modeling

- The polynomial performance model is built based on a set of Monte Carlo samples
- Sparse regression is adopted for the polynomial performance modeling task

Polynomial performance model

\[ f(x_1, x_2, \ldots) = \ldots \]

\[ f^{(1)} = \alpha_0 + \alpha_1 \cdot x_1^{(1)} + \alpha_2 \cdot x_2^{(1)} + \ldots \]

\[ f^{(2)} = \alpha_0 + \alpha_1 \cdot x_1^{(2)} + \alpha_2 \cdot x_2^{(2)} + \ldots \]

\[ f^{(3)} = \alpha_0 + \alpha_1 \cdot x_1^{(3)} + \alpha_2 \cdot x_2^{(3)} + \ldots \]

\[ \vdots \]

- L₁-norm regularization is used to find the sparse solution \( \alpha \)

\[
\begin{align*}
\text{minimize} & \quad \|X \cdot \alpha - F\|_2^2 \\
\text{subject to} & \quad \|\alpha\|_1 \leq \lambda
\end{align*}
\]

Mean squared error

L₁-norm constraint to promote sparsity
Polynomial Optimization

- This optimization problem can be non-convex, due to the general model template assumption.

- To solve the global optimal solution of this non-convex problem, we adopt a convexification approach.

The proposed flow:

- **Part 1**: Convexifying polynomial cost function
- **Part 2**: Convexifying polynomial constraints
- **Part 3**: Sequential semidefinite programming

In what follows, we consider a simplified problem w/o constraints.

Convexifying Polynomial Cost Function

Convexifying polynomial cost function

- Convert the polynomial cost function to a PDF view

Original cost function

\[ \min_x f(x) \]

PDF view

\[ \min_{\mu(x)} \int f(x) \cdot \mu(x) \cdot dx \]

\[ \sum_i f(x_i) \cdot \mu(x_i) \cdot \Delta x \]

Equivalent to weighted sum of \( f(x_i) \)

Optimize weight assignment

Optimize over all PDF function \( \mu(x) \)

Integration value

For example:

\( f(x) \)

\( x^* \)

\( x \)

Slide 11
Convexifying Polynomial Cost Function

Convexifying polynomial cost function

- Convert the polynomial cost function to a PDF view

Original cost function

\[ \min_x f(x) \]

PDF view

Optimize over all PDF function \( \mu(x) \)

\[ \min_{\mu(x)} \int f(x) \cdot \mu(x) \cdot dx \]

\[ \sum_i f(x_i) \cdot \mu(x_i) \cdot \Delta x \]

Equivalent to weighted sum of \( f(x_i) \)

Optimize weight assignment

Integration value

For example:

\[ f(x) \]
Convexifying Polynomial Cost Function

Convexifying polynomial cost function

- Convert the polynomial cost function to a PDF view

Original cost function

\[
\min_x f(x)
\]

PDF view

Optimize over all PDF function \( \mu(x) \)

\[
\min_{\mu(x)} \int f(x) \cdot \mu(x) \cdot dx
\]

\[
\sum_i f(x_i) \cdot \mu(x_i) \cdot \Delta x
\]

Equivalent to weighted sum of \( f(x_i) \)

Optimize weight assignment

For example:

- Integration value
- Optimal!
Convexifying polynomial cost function

- Convert the polynomial cost function to a PDF view

Original cost function

$$
\min_x f(x)
$$

PDF view

Optimize over all PDF function $\mu(x)$

$$
\min_{\mu(x)} \int f(x) \cdot \mu(x) \cdot dx
$$

For example:

$$
\mu^*(x) = \delta(x - x^*)
$$

Integration value

Optimal!

**Equivalency:** solving the problem in a PDF view gives the optimal cost function of the original problem
Convexifying Polynomial Cost Function

Convexifying polynomial cost function

- Actually, **optimizing PDF** is not simpler than the original problem
- The PDF view leads to a **moment view**

For example: \( f(x) = x^3 + x^2 + 2x \)

**PDF view:**

\[ \int f(x) \cdot \mu(x) \cdot dx = f(x) = x^3 + x^2 + 2x \]

**Moment view:**

\[ \int x^3 \cdot \mu(x) \cdot dx + \int x^2 \cdot \mu(x) \cdot dx + 2 \cdot \int x \cdot \mu(x) \cdot dx \]

**Moments**
Convexifying Polynomial Cost Function

Convexifying polynomial cost function

For example: $f(x) = x^3 + x^2 + 2x$

PDF view:

$$\min_{\mu(x)} \int f(x) \cdot \mu(x) \cdot dx$$

Moment view:

$$f(x) = x^3 + x^2 + 2x$$

$$\min_{\mu(x)} \int f(x) \cdot \mu(x) \cdot dx = \min_{y_1, y_2, y_3} y_3 + y_2 + 2 \cdot y_1$$

To summarize

$$\min_{x} f(x) \quad 1 \text{ variable}$$

Linear cost function in moments

Moments

$$\int x^3 \cdot \mu(x) \cdot dx$$

$$2 \int x \cdot \mu(x) \cdot dx$$

$$\int x^2 \cdot \mu(x) \cdot dx$$

3 variables: one for linear, one for quadratic, one for cubic

Introduce more variables to remove nonlinearity in cost function
Convexifying Polynomial Cost Function

Convexifying polynomial cost function

Moment view:

\[
\begin{align*}
\min_{y_1, y_2, y_3} & \quad y_3 + y_2 + 2 \cdot y_1 \\
& \quad \int x^3 \cdot \mu(x) \cdot dx \\
& \quad \int x^2 \cdot \mu(x) \cdot dx \\
& \quad 2 \int x \cdot \mu(x) \cdot dx
\end{align*}
\]

Question: can the moments take arbitrary values?

Answer: No!

For example:

\[y_2 = \int x^2 \cdot \mu(x) \cdot dx \geq 0\]

Negative \(y_2\) is not possible

A set of constraints on moments need to be set up similar to the above one

Criteria: a PDF \(\mu(x)\) must exist to generate the moments: (i.e. legalization)

Mathematical formulation

A sequence of semidefinite positive constraints on moment matrix

Such as:

\[
\begin{bmatrix}
y_0 & y_1 \\
y_1 & y_2
\end{bmatrix} \succ
\begin{bmatrix}
y_0 & y_1 & y_2 \\
y_1 & y_2 & y_3 \\
y_2 & y_3 & y_4
\end{bmatrix} \succ 0
\]
Sequential SDP

Sequential semidefinite programming

- A sequence of SDP problems $\{H^d; d = 1, 2, \ldots\}$ are obtained

$$\begin{align*}
H^d : & \min_{y} e(y) = \sum_{r=1}^{R} \alpha_r \cdot y_r \\
& \text{s.t. } C_d(y) \succeq 0
\end{align*}$$

where $y = [y_1 \quad y_2 \quad \cdots]^{T}$

Cost function is linear in terms of $y$

Semidefinite positive constraint

$C_1(y) = \begin{bmatrix} y_0 & y_1 \\ y_1 & y_2 \end{bmatrix} \succeq 0$

$C_2(y) = \begin{bmatrix} y_0 & y_1 & y_2 \\ y_1 & y_2 & y_3 \\ y_2 & y_3 & y_4 \end{bmatrix} \succeq 0$
Sequential SDP

Features of sequential SDP

For the optimization problem with constraints, additional SDP constraints are needed

- The constraint can be formulated similarly as the unconstrained case

Optimal solution of $x$

Optimal moments $\rightarrow$ Optimal $x$
Numerical Experiments

- A reconfigurable RF front-end designed for WLAN 802.11g is used
  - In this problem, we consider to minimize power subject to SNR constraint

![RF front-end diagram]

- To fit the SNR model, the RF front-end is simulated by MATLAB SIMULINK and 800 samples are collected

<table>
<thead>
<tr>
<th>Model order</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Error (dB)</td>
<td>3.03</td>
<td>1.91</td>
<td>1.81</td>
</tr>
</tbody>
</table>
Numerical Experiments

Based on performance model, two approaches are compared
- Simulated annealing (SA)
- Moment method

<table>
<thead>
<tr>
<th>SNR specification</th>
<th>Methods</th>
<th>Optimization results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fitted SNR (dB)</td>
<td>Simulated SNR (dB)</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>14.71</td>
</tr>
<tr>
<td>15</td>
<td>SA</td>
<td>15.00</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>15.00</td>
</tr>
<tr>
<td>17</td>
<td>SA</td>
<td>17.00</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>17.00</td>
</tr>
</tbody>
</table>

- Moment method achieves superior performance than SA
- Simulated SNR is close to SNR value in performance model
Numerical Experiments

- **Statistical behavior is studied for SA**
  - 100 independent runs with random initial guess are performed

- **SA is not guaranteed to converge to global optimum**
  - It is not possible to know if SA reaches global optimum or not

- **Moment method is guaranteed to find global optimum**
Numerical Experiments

- Runtime comparison

<table>
<thead>
<tr>
<th>SNR Spec (dB)</th>
<th>Model Fitting (Sec.)</th>
<th>Optimization (Sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Exhaustive search (estimated)</td>
</tr>
<tr>
<td>13.00</td>
<td>3.852x10^5</td>
<td>1.718x10^10</td>
</tr>
<tr>
<td>15.00</td>
<td></td>
<td>1.718x10^10</td>
</tr>
<tr>
<td>17.00</td>
<td></td>
<td>1.718x10^10</td>
</tr>
</tbody>
</table>

- Exhaustive search is not practical due to high cost

- For SA and the proposed approach
  - Modeling cost is dominated by model fitting cost
  - Proposed approach is guaranteed to find the global optimum while SA can be trapped in sub-optimal solutions
Conclusions

- For reconfigurable RF system programming problem, a performance model driven optimization approach based on polynomial programming is proposed.

- The non-convex polynomial programming problem is converted to a sequence of convex SDP problems based on convexification.

- The proposed approach is validated in a reconfigurable RF front-end example designed for WLAN 802.11g. As demonstrated in the example, efficient and robust programming can be achieved by applying the proposed approach.