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

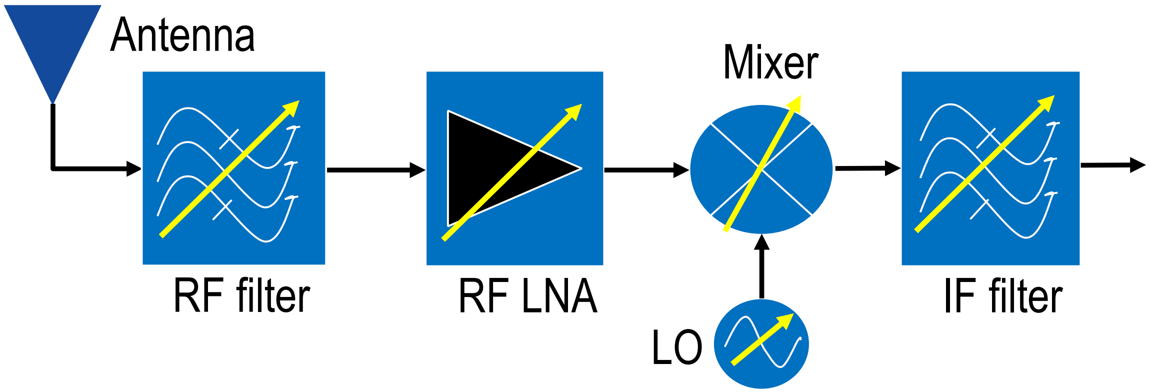
Fa Wang, Shihui Yin, Minhee Jun, Xin Li, Tamal Mukherjee, Rohit Negi and Larry Pileggi

Carnegie Mellon University, Pittsburgh, PA 15213 USA

Electrical & Computer ENGINEERING

Motivation

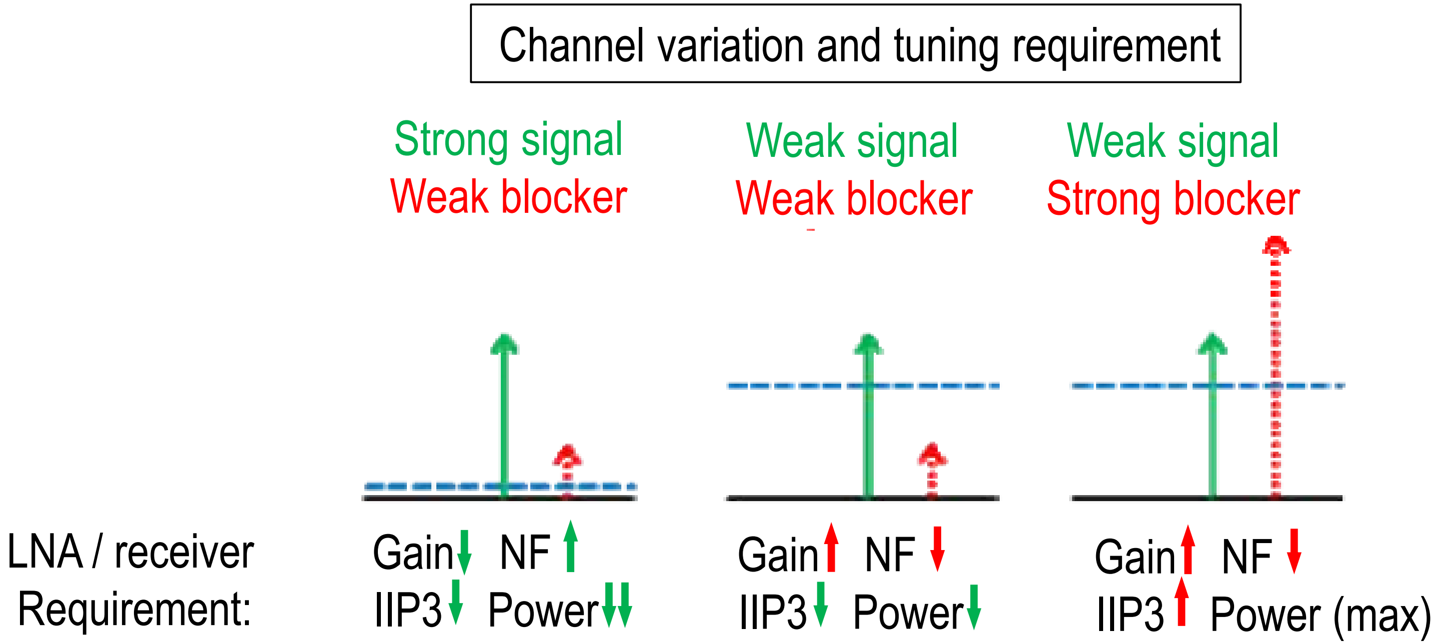
- **Rapidly introduced wireless standards and applications pose grand challenges for wireless chip design**
  - High cost in designing multiple fixed narrow-band RF front-ends
  - Nonlinearity problems for wide band RF front-end
- **In this context, reconfigurable RF system has been proposed**
  - Introduce tunable control knobs in circuit blocks so that the performances can be adaptively changed



[Reconfig-RF1] D. Banerjee, et. al., "Self-learning MIMO-RF receiver systems: process resilient real-time adaptation to channel conditions for low power operation", *IEEE ICCAD*, 2014.

Motivation

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

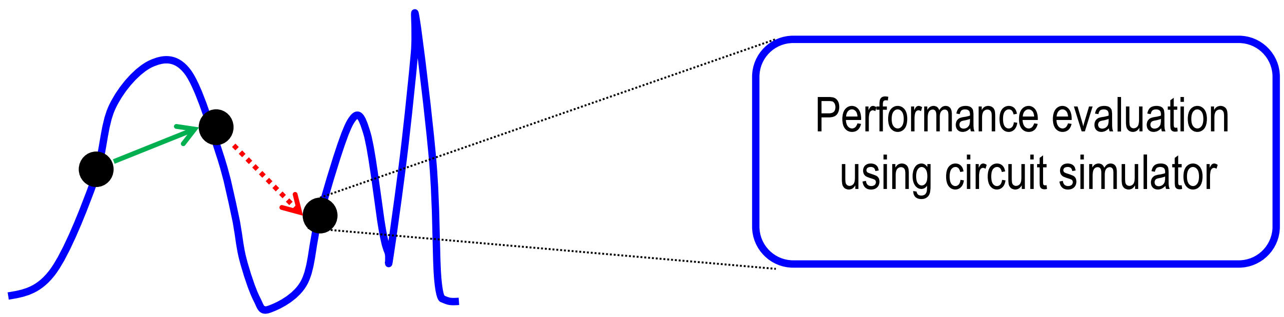


- **Programming a reconfigurable RF system is an important task in order to maximally exploit the benefit of its reconfigurability**

[Reconfig-RF2] S. Sen, "Channel-adaptive zero-margin & process-adaptive self-healing communication circuits/systems," *IEEE ICCAD*, 2014.

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**

[SA1] G. Gielen, et. al., "Analog circuit design optimization based on symbolic simulation and simulated annealing," *IEEE JSSC*, 1990.
[SA2] M. Krasnicki, et. al., "MAELSTROM: Efficient simulation-based synthesis for custom analog cells," *IEEE DAC*, 1999.

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

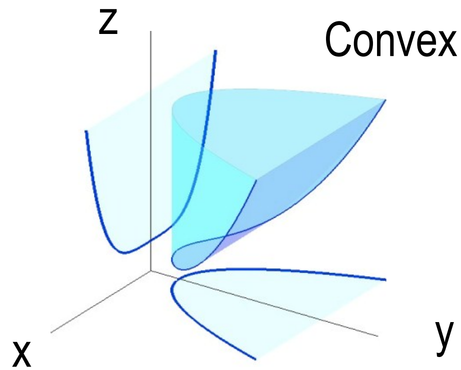
Special form

$$\text{Performance}_1 = f_1(\text{Parameters})$$

$$\vdots \qquad \vdots$$

$$\text{Performance}_K = f_K(\text{Parameters})$$

Convex optimization

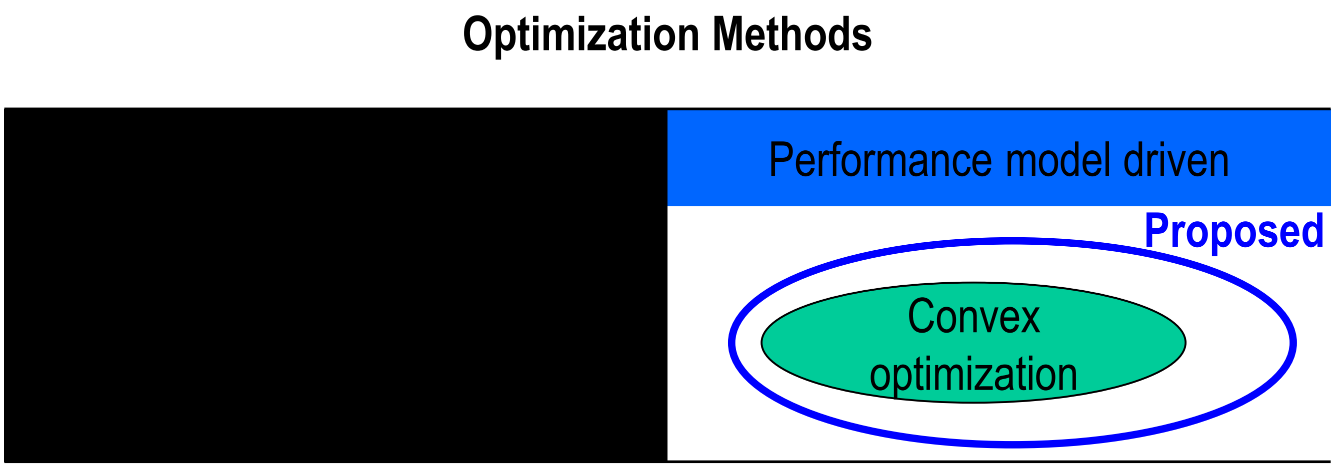


- **Pros: can capture global behavior using performance model; convex optimization is easy to solve**
- **Cons: relies on the accuracy of model template; the optimal solution may not accurately match the actual circuit behavior**

[GP1] M. Hershenson, et. al., "GPCAD: a tool for CMOS op-amp synthesis," *IEEE ICCAD*, 1998.
[GP2] W. Daems, et. al., "Simulated-based automatic generation of signomial and posynomial performance models for analog integrated circuit sizing," *IEEE ICCAD*, 2001.

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



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

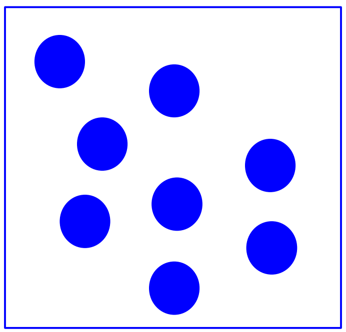
[PoP] J. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, 2001.

Outline

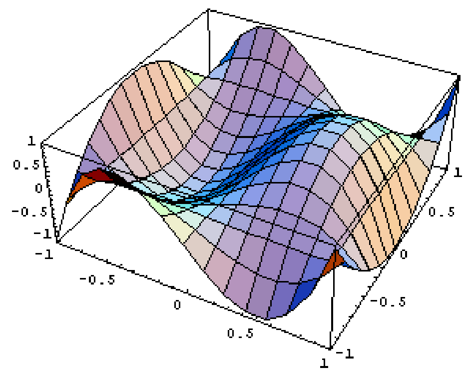
- Motivation
- The Proposed Approach
- Numerical Results
- Conclusions

The Proposed Flow

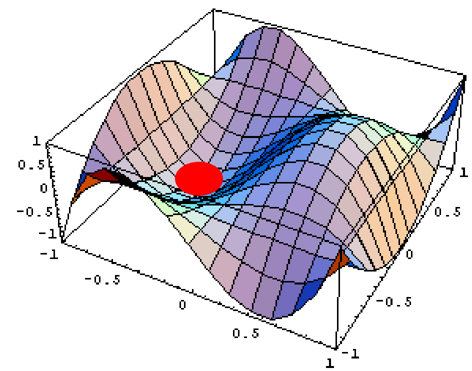
- **Two Steps**
  - Polynomial performance modeling
  - Polynomial optimization based on convexification



Simulation samples are collected



Polynomial performance model



Polynomial optimization

$$\min_{\mathbf{x}} \quad f(\mathbf{x})$$

$$\text{s.t.} \quad g_m(\mathbf{x}) \le G_m \quad (m = 1, 2, \cdots$$

$$\mathbf{x} \in S$$

$$\mathbf{x} = [x_1 \quad x_2 \quad \cdots$$

Polynomial performance cost function

Control knob values

Additional linear constraints on **x**

Polynomial performance constraints

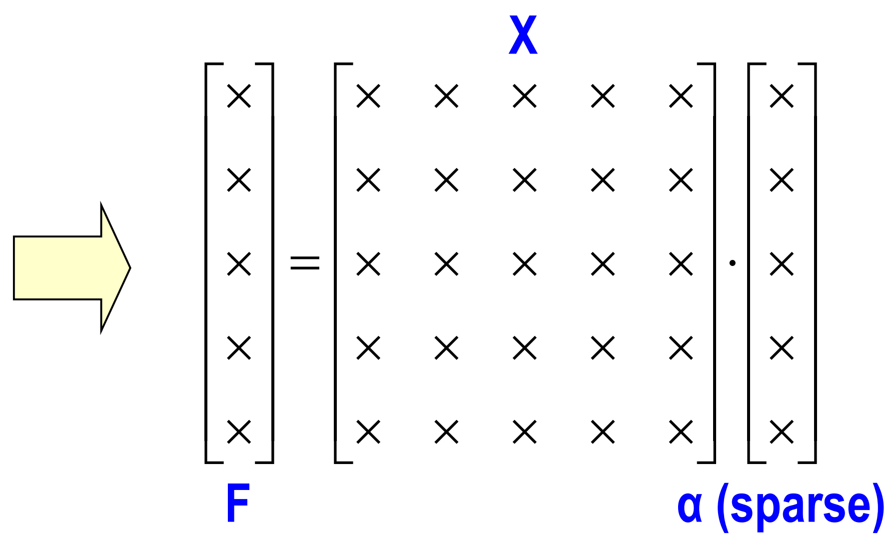
Polynomial Performance Modeling

- 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

$$f(x_1, x_2, \cdots \quad \cdots$$

$$f^{(1)} = \alpha_0 + \alpha_1 \cdot x_1^{(1)} + \alpha_2 \cdot x_2^{(1)} + \cdots$$
$$f^{(2)} = \alpha_0 + \alpha_1 \cdot x_1^{(2)} + \alpha_2 \cdot x_2^{(2)} + \cdots$$
$$f^{(3)} = \alpha_0 + \alpha_1 \cdot x_1^{(3)} + \alpha_2 \cdot x_2^{(3)} + \cdots$$
$$\vdots$$



- L_1-norm regularization is used to find the sparse solution α

$$\begin{aligned} &\textit{minimize} && \|\mathbf{X} \cdot \boldsymbol{\alpha} - \mathbf{F}\|_2^2 \\ &\textit{subject to} && \|\boldsymbol{\alpha}\|_1 \le \lambda \end{aligned}$$

- $\|\mathbf{X} \cdot \boldsymbol{\alpha} - \mathbf{F}\|_2^2$ → Mean squared error
- $\|\boldsymbol{\alpha}\|_1 \le \lambda$ → L_1-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

[PoP] J. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, 2001.

Convexifying Polynomial Cost Function

Convexifying polynomial cost function

- **Convert the polynomial cost function to a PDF view**

Original cost function

$$\min_{\mathbf{x}} \quad f(\mathbf{x})$$

PDF view

Optimize over all **PDF function** $\mu(\mathbf{x})$

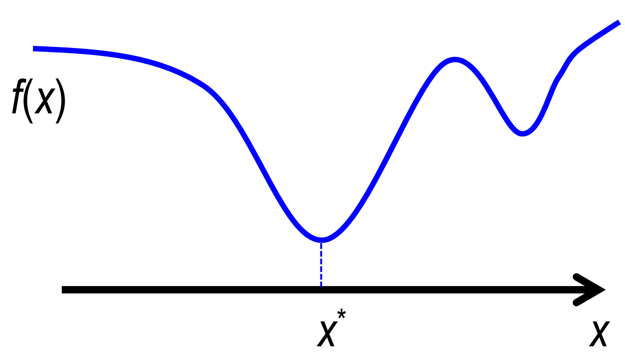
$$\min_{\mu(\mathbf{x})} \quad \int f(\mathbf{x}) \cdot \mu(\mathbf{x}) \cdot d\mathbf{x}$$

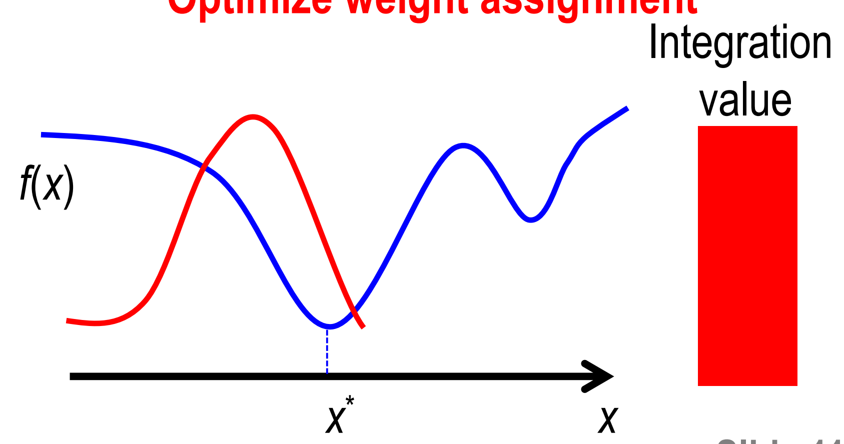
$$\sum_i f(\mathbf{x}_i) \cdot \mu(\mathbf{x}_i) \cdot \Delta\mathbf{x}$$

Equivalent to **weighted sum** of $f(\mathbf{x}_i)$

Optimize weight assignment

For example:





Convexifying polynomial cost function

- **Convert the polynomial cost function to a PDF view**

Original cost function

$$\min_{\mathbf{x}} \quad f(\mathbf{x})$$

PDF view

Optimize over all **PDF function** $\mu(\mathbf{x})$

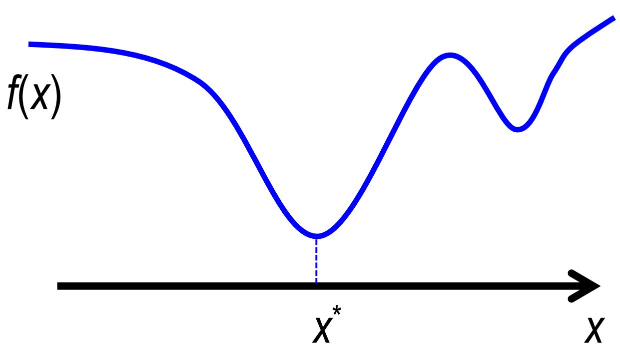
$$\min_{\mu(\mathbf{x})} \quad \int f(\mathbf{x}) \cdot \mu(\mathbf{x}) \cdot d\mathbf{x}$$

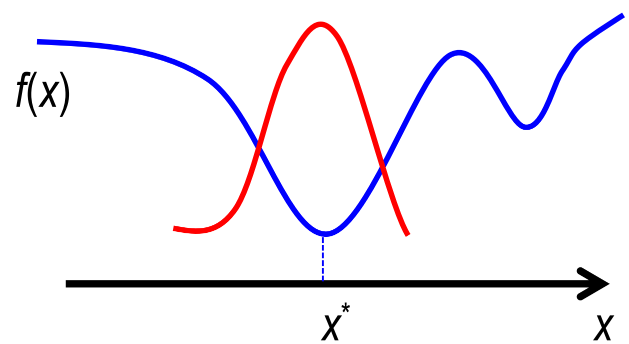
$$\sum_i f(\mathbf{x}_i) \cdot \mu(\mathbf{x}_i) \cdot \Delta\mathbf{x}$$

Equivalent to **weighted sum** of $f(\mathbf{x}_i)$

Optimize weight assignment

For example:





Integration value

Convexifying Polynomial Cost Function

Convexifying polynomial cost function

- **Convert the polynomial cost function to a PDF view**

Original cost function

$$\min_{\mathbf{x}} \quad f(\mathbf{x})$$

PDF view

Optimize over all **PDF function** $\mu(\mathbf{x})$

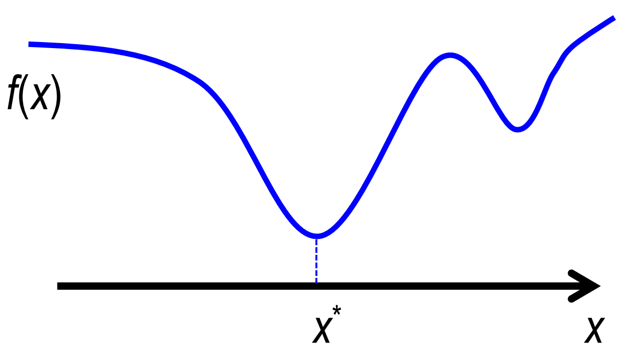
$$\min_{\mu(\mathbf{x})} \quad \int f(\mathbf{x}) \cdot \mu(\mathbf{x}) \cdot d\mathbf{x}$$

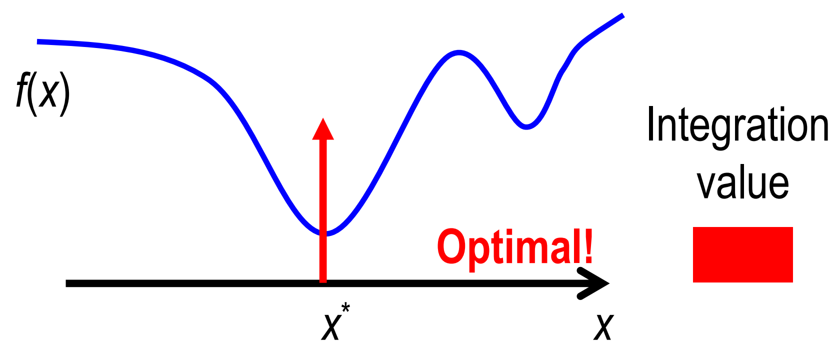
$$\sum_i f(\mathbf{x}_i) \cdot \mu(\mathbf{x}_i) \cdot \Delta\mathbf{x}$$

Equivalent to **weighted sum** of $f(\mathbf{x}_i)$

Optimize weight assignment

For example:





Convexifying Polynomial Cost Function

Convexifying polynomial cost function

- **Convert the polynomial cost function to a PDF view**

Original cost function

$$\min_{\mathbf{x}} \quad f(\mathbf{x})$$

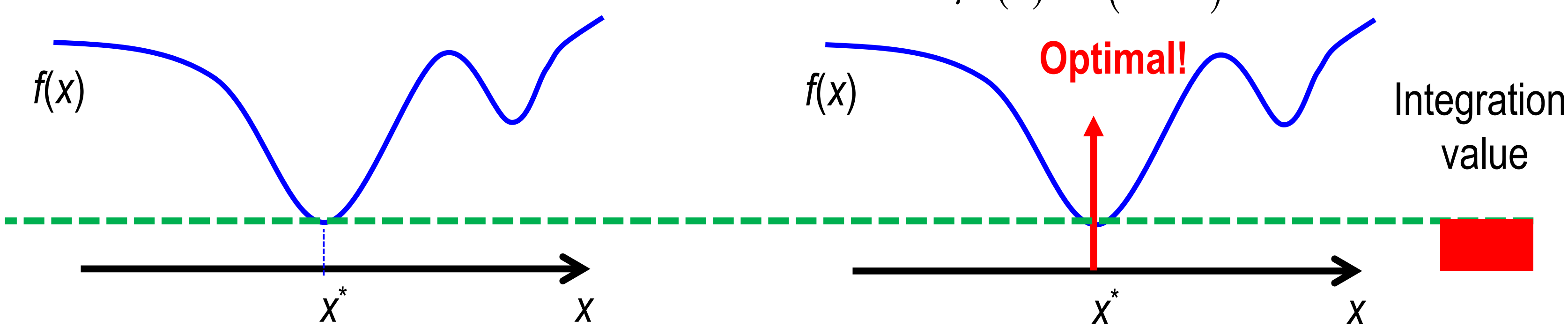
PDF view

Optimize over all **PDF function** $\mu(\mathbf{x})$

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

For example:

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



- **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**

Original polynomial cost function → PDF view → Moment view

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

PDF view: → Moment view:

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

$$\int f(x)\cdot\mu(x)\cdot dx = \begin{cases} \int x^3\cdot\mu(x)\cdot dx \\ + \\ \int x^2\cdot\mu(x)\cdot dx \\ + \\ 2\cdot\int x\cdot\mu(x)\cdot dx \end{cases}$$

Moments

Convexifying Polynomial Cost Function

Convexifying polynomial cost function

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

PDF view: → Moment view:

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

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

Linear cost function in moments

$$\int x^3 \cdot \mu(x) \cdot dx \qquad \int x^2 \cdot \mu(x) \cdot dx \qquad 2 \cdot \int x \cdot \mu(x) \cdot dx$$

Moments

To summarize

$$\min_x f(x)$$ 1 variable

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: $\min_{y_1, y_2, y_3} \; y_3 + y_2 + 2 \cdot y_1$

$\int x^3 \cdot \mu(x) \cdot dx$ $\quad$ $\int x^2 \cdot \mu(x) \cdot dx$ $\quad$ $2 \int x \cdot \mu(x) \cdot dx$ $\quad$ **Moments**

Question: can the moments take arbitrary values?

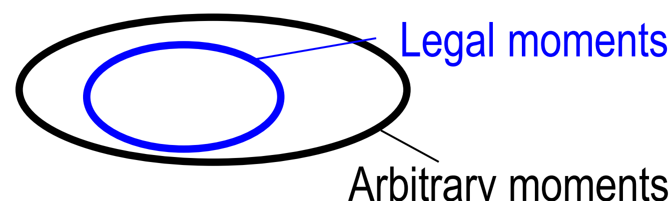
Answer: No!

For example: $y_2 = \int x^2 \cdot \mu(x) \cdot dx \geq 0$ $\quad$ 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 *μ*(x) must exist to generate the moments: (i.e. legalization)

Mathematical formulation



A sequence of semidefinite positive constraints on moment matrix

Such as: $\begin{bmatrix} y_0 & y_1 \\ y_1 & y_2 \end{bmatrix} \succ \quad \begin{bmatrix} y_0 & y_1 & y_2 \\ y_1 & y_2 & y_3 \\ y_2 & y_3 & y_4 \end{bmatrix} \succ \quad \bullet\bullet\bullet$

Sequential semidefinite programming

- A sequence of SDP problems {H^d; *d* = 1, 2, …} are obtained

Cost function is linear in terms of **y**

$$\mathbf{H}^d : \begin{cases} \min\limits_{\mathbf{y}} & e(\mathbf{y}) = \sum_{r=1}^{R} \alpha_r \cdot y_r \\ \text{s.t.} & C_d(\mathbf{y}) \succ \end{cases} (d = 1, 2, \cdots$$

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

Semidefinite positive constraint

where

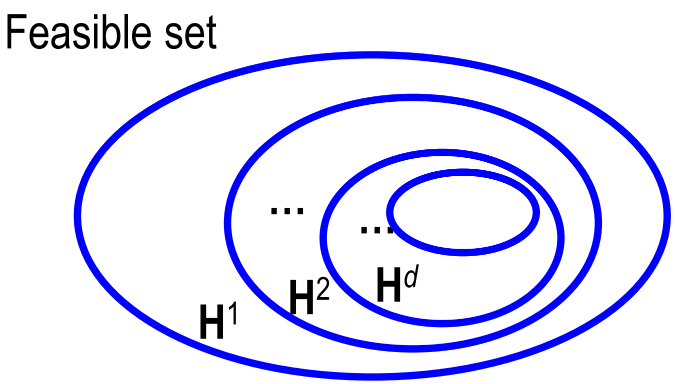
$$C_1(\mathbf{y}) = \begin{bmatrix} y_0 & y_1 \\ y_1 & y_2 \end{bmatrix} \succ$$

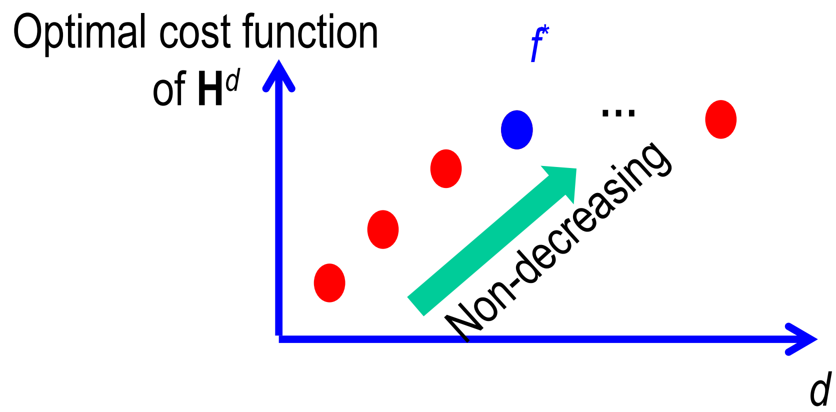
$$C_2(\mathbf{y}) = \begin{bmatrix} y_0 & y_1 & y_2 \\ y_1 & y_2 & y_3 \\ y_2 & y_3 & y_4 \end{bmatrix} \succ$$

⋮

Sequential SDP

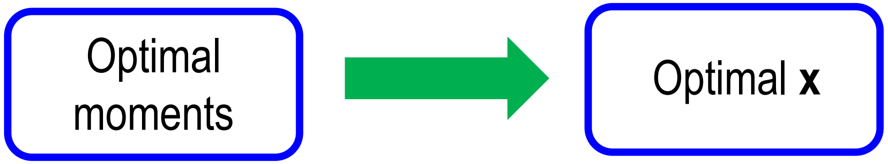
- **Features of sequential SDP**





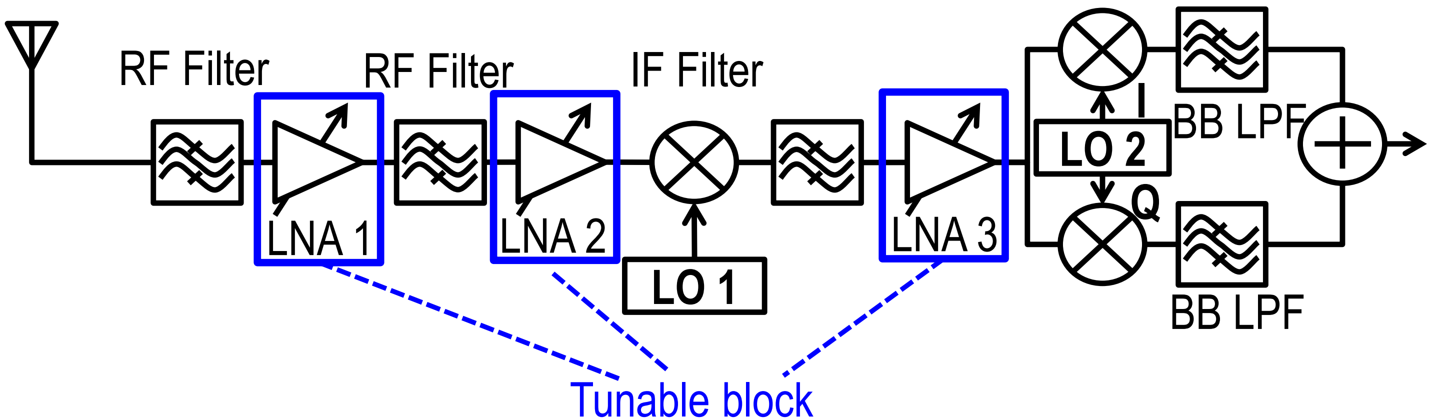
- **For the optimization problem with constraints, additional SDP constraints are needed**
  - The constraint can be formulated similarly as the unconstrained case

- **Optimal solution of x**



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



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

Model order	2	3	4
[redacted]			
Maximum Error (dB)	3.03	1.91	1.81
[redacted]			

Numerical Experiments

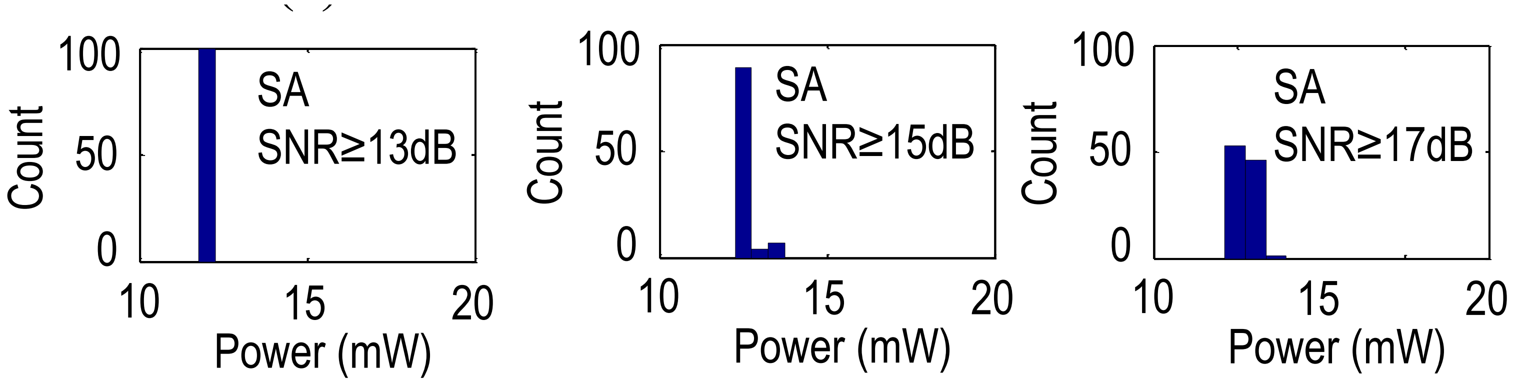
- Based on performance model, two approaches are compared
  - Simulated annealing (SA)
  - Moment method

SNR specification	Methods	Optimization results		
		Fitted SNR (dB)	Simulated SNR (dB)	Power (mW)
13	SA	14.72	14.35	11.99
	Proposed	14.71	14.35	11.97
15	SA	15.00	14.99	13.49
	Proposed	15.00	15.04	12.58
17	SA	17.00	16.61	21.95
	Proposed	17.00	17.30	19.24

- 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

SNR Spec (dB)	Model Fitting (Sec.)	Optimization (Sec.)		
		Exhaustive search (estimated)	SA	Proposed
13.00	3.852×10^5	1.718×10^{10}	76	2.4
15.00		1.718×10^{10}	76	67
17.00		1.718×10^{10}	76	76

- **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