# Solving Parametric Polynomial Optimization via Triangular Decomposition


Changbo Chen[1], Marc Moreno Maza[1,2]

[1]Chongqing Institute of Green and Intelligent Technology,
Chinese Academy of Sciences, China

[1,2]Western University, Canada

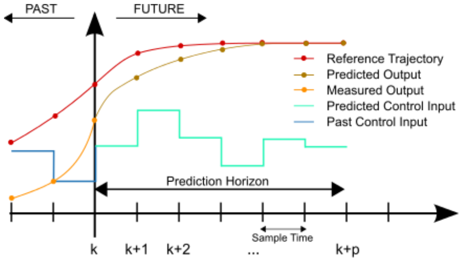

ICIAM 2015
August 10, 2015

# Plan

# Plan

# Model Predictive Control

- Model Predictive Control (MPC) is widely used in process control.
- At each *control step*, MPC predicts a sequence of future control actions by solving an *optimization problem* which depends on the current values of the state variables.
- Only the first control action is applied to the process.
- The control then moves on to the next time interval and repeats the previous control step based on the new values of the state variables.

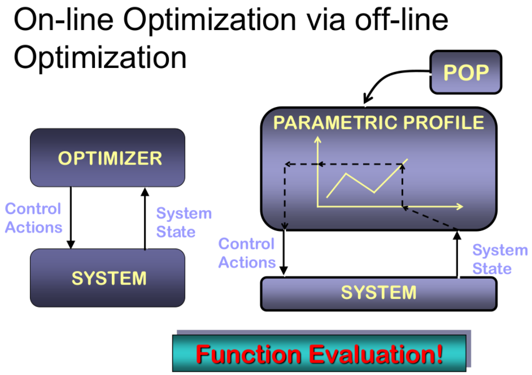



## Key observation

All these on-line problems have the same structure.

# The offline-online strategy

- It is natural to divide the whole computational procedure into two phases: the *off-line* and *on-line*.
- The *off-line* phase computes the optimal solution as a function of the state variables (regarded now as parameters) while the *on-line* phase reduces optimization problems to function evaluation calculations.



# Plan

# Formulation of the problem

## Notations

- let $\mathbf{x} := x_1 \prec x_2 \prec \cdots \prec x_m$
- let $\mathbf{u} := u_1 \prec u_2 \prec \cdots \prec u_d$
- let $f \in \mathbb{Q}[\mathbf{u}, \mathbf{x}]$
- let $F = \{f_1, \ldots, f_r\} \subset \mathbb{Q}[\mathbf{u}, \mathbf{x}]$
- let $G = \{g_1, \ldots, g_q\} \subset \mathbb{Q}[\mathbf{u}, \mathbf{x}]$

## The problem to solve

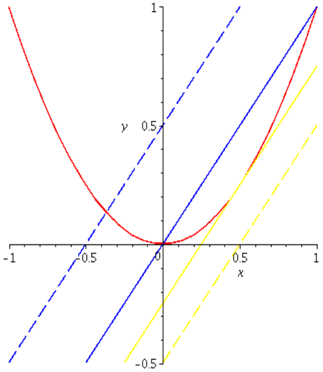
- Input:

$$\begin{array}{rcl} z(\mathbf{u}) & = & \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{u}) \text{ s.t.} \\ & & g_1(\mathbf{u}, \mathbf{x}) \leq 0, \ldots, g_q(\mathbf{u}, \mathbf{x}) \leq 0 \\ & & f_1(\mathbf{u}, \mathbf{x}) = 0, \ldots, f_r(\mathbf{u}, \mathbf{x}) = 0 \end{array} \tag{1}$$

- Output: function $Z(\mathbf{u})$, $X(\mathbf{u})$ and their domain $\mathcal{U} \subset \mathbb{R}^d$.

# The first example

$$z(u) = \min_{x,y} y \text{ s.t.}$$
$$-x + y + u \leq 0$$
$$y - x^2 = 0$$



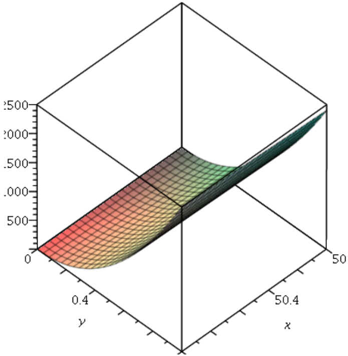

- $u \leq 0,\ x(u) = 0, y(u) = 0,\ z(u) = 0$
- $0 < u \leq 1/4,\ x(u) = \frac{1}{2} - \frac{1}{2}\sqrt{-4u+1},\ y(u) = -u + \frac{1}{2} - \frac{1}{2}\sqrt{-4u+1},$ $z(u) = -u + \frac{1}{2} - \frac{1}{2}\sqrt{-4u+1}$

## The second example

This example is to illustrate that infinimum may be not attained. It is adapted from Example 5.1 in the ISSAC 2010 paper by F. Guo, et. al.

$$z(u) \quad = \quad \min_{x,y} \, (u - xy)^2 + y^2$$

- $u = 0$, $z(u) = 0$
- $u \neq 0$, $z(u) = 0$, but $z(u)$ is not attained.

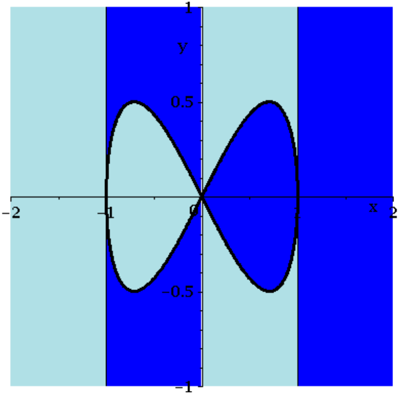


# Plan

# Cylindrical Algebraic Decomposition (CAD) of $\mathbb{R}^n$

A CAD of $\mathbb{R}^n$ is a partition of $\mathbb{R}^n$ such that each cell in the partition is a connected semi-algebraic subset of $\mathbb{R}^n$ and all the cells are cylindrically arranged.

Two subsets $A$ and $B$ of $\mathbb{R}^n$ are called cylindrically arranged if for any $1 \leq k < n$, the projections of $A$ and $B$ on $\mathbb{R}^k$ are either equal or disjoint.



# CAD based on projection-lifting scheme (PL-CAD)

## Projection

- Let $Proj$ be a projection operator.
- Repeatedly apply $Proj$:

$$F_n(x_1, \ldots, x_n) \xrightarrow{Proj} F_{n-1}(x_1, \ldots, x_{n-1}) \xrightarrow{Proj} \cdots \xrightarrow{Proj} F_1(x_1).$$

## Lifting

- The real roots of the polynomials in $F_1$ plus the open intervals between them form an $F_1$-invariant CAD of $\mathbb{R}^1$.
- For each cell $C$ of the $F_{k-1}$ invariant CAD of $\mathbb{R}^{k-1}$, isolating the real roots of the polynomials of $F_k$ at a sample point of $C$, produces all the cells of the $F_k$-invariant CAD of $\mathbb{R}^k$ above $C$.

# CAD based on regular chains (RC-CAD)

## Motivation: potential drawback of Collins' projection-lifting scheme

- The projection operator is a function defined independently of the input system.
- As a result, a strong projection operator (Collins-Hong operator) usually produces much more polynomials than needed.
- A weak projection operator (McCallum-Brown operator) may fail for non-generic cases.

## Solution: Make case distinction during projection

- Case distinction (zero-test, regularity test) is common for algorithms computing triangular decompositions.
- At ISSAC'09, we introduced the idea and technique of case distinction (by computing regular GCDs) into CAD computation.
- The new method consists of two phases. The first phase computes a complex cylindrical tree (CCT). The second phase decomposes each cell of CCT into its real connected components.

# Parametric polynomial optimization by RC-CAD (I)

- Algorithm: MinCAD
- Input: The minimization problem (**??**)
- Output: A set of CAD cells encoding solutions to (**??**)

1. Introduce a new variable $z$ to denote the optimal value
2. Add the equational constraint $z - f(\mathbf{u}, \mathbf{x}) = 0$ to $F$ (or add the inequation $f(\mathbf{u}, \mathbf{x}) - z \leq 0$ to $G$
3. Define the input system $S := \{F = 0, G \neq 0\}$.
4. Define the elimination order $\mathbf{x} > z > \mathbf{u}$
5. Call RC-CAD to compute a truth invariant CAD of $\mathbb{R}^{m+d+1}$ w.r.t. $S$
6. If there are no true cells, return $\emptyset$
7. Categorizing the true cells such that the cells having the same projection onto $\mathbf{u}$-space are in the same group
8. Define $L := \emptyset$
9. For each group, add a cell whose $z$-index is the smallest into $L$
10. Output $L$

# Parametric polynomial optimization by RC-CAD (II)

- Input:

$$\begin{aligned} z(\mathbf{u}) &= \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{u}) \text{ s.t.} \\ & g_1(\mathbf{u}, \mathbf{x}) \leq 0, \ldots, g_q(\mathbf{u}, \mathbf{x}) \leq 0 \\ & f_1(\mathbf{u}, \mathbf{x}) = 0, \ldots, f_r(\mathbf{u}, \mathbf{x}) = 0 \end{aligned}$$

- Output: function $Z(\mathbf{u})$, $X(\mathbf{u})$ and their domain $\mathcal{U} \subset \mathbb{R}^d$.

### Theorem

*Let $L$ be a set of CAD cells computed by* MinCAD. *Then we have*

- *If $L = \emptyset$, then no feasible solutions exist for problem (**??**).*
- *The set of parametric values, such that problem (**??**) has feasible solutions is: $\mathcal{U} = \cup_{c \in L} \pi_{\mathbf{u}}(c)$*
- *Let $C$ be a cell in $L$.*
  - *If $C^z$ is of type $z < \phi(\mathbf{u})$ or $z = z$, then $z(\mathbf{u}) = -\infty$.*
  - *If $C^z$ is of type $z = \phi(\mathbf{u})$ , then $z(\mathbf{u}) = \phi(\mathbf{u})$, the minimum is attained and $C^{>z}$ defines at least one optimizer.*
  - *If $C^z$ is of type $z > \phi(\mathbf{u})$ or $\phi(\mathbf{u}) < z < \psi(\mathbf{u})$, then $z(\mathbf{u}) = \phi(\mathbf{u})$. But the minimum is not attained.*

# An example

$$z(\mathbf{u}) = \min_{x_1,x_2} \quad x_1 + u_1 x_2$$
$$\text{s.t.} \quad u_2^2 x_1^2 - x_2 \leq 0$$
$$x_1 \leq 0$$

The computation steps

- Let $F := \emptyset$ and $G := [u_2^2 x_1^2 - x_2, x_1]$.
- The input system is $S := \{z - (x_1 + u_1 x_2), u_2^2 x_1^2 - x_2 \leq 0, x_1 \leq 0\}$.
- The elimination order is $x_2 > x_1 > z > u_2 > u_1$.
- Call CylindricalAlgebraicDecompose in `RegularChains` library to compute a truth invariant CAD w.r.t. $S$. The output has $42$ cells.
- From the output, $7$ cells are selected to encode solutions of the minimization problem.

The solution

- If $u_1 \leq 0$, then $z(\mathbf{u}) = -\infty$. If $u_1 > 0$ and $u_2 = 0$, then $z(\mathbf{u}) = -\infty$.
- If $u_1 > 0$ and $u_2 \neq 0$, then $z(\mathbf{u}) = -\frac{1}{4u_1 u_2^2}$, $x_1(\mathbf{u}) = -\frac{1}{2u_1 u_2^2}$ and $x_2(\mathbf{u}) = \frac{1}{4u_1 u_2^2}$.

# A screen shot showing part of the solutions

```
> restart; read "optimization.mpl";
> f := x_1+u_1*x_2: G:=[u_2^2*x_1^2-x_2, x_1]: F:=[]: X:=[x_2,x_1]: U:=[u_2,u_1]:
  sols, R := MinimizeByCAD(f, F, G, X, U);
```

$$sols, R := \left[ [cad\_cell, \_z < 0], [cad\_cell, \_z < 0], [cad\_cell, \_z < 0], [cad\_cell, \_z < 0], \left[cad\_cell, \_z = -\frac{1}{4\ u\_1\ u\_2^2}\right], [cad\_cell, \_z < 0], \left[cad\_cell, \_z = -\frac{1}{4\ u\_1\ u\_2^2}\right]\right], polynomial\_ring$$

```
> Display(sols[6..7], R);
```

$$\left[ \left[ \left\{ \begin{array}{c} x\_2 = \dfrac{-x\_1 + \_z}{u\_1} \\ x\_1 < \_z \\ \_z < 0 \\ u\_2 = 0 \\ 0 < u\_1 \end{array} \right., \_z < 0 \right], \left[ \left\{ \begin{array}{c} x\_2 = \dfrac{-x\_1 + \_z}{u\_1} \\ x\_1 = -\dfrac{1}{2\ u\_1\ u\_2^2} \\ \_z = -\dfrac{1}{4\ u\_1\ u\_2^2} \\ 0 < u\_2 \\ 0 < u\_1 \end{array} \right., \_z = -\dfrac{1}{4\ u\_1\ u\_2^2} \right] \right]$$

# Plan

# The KKT Conditions

- Input:

$$z(\mathbf{u}) = \min_{\mathbf{x}} f(\mathbf{x},\mathbf{u}) \text{ s.t.}$$
$$g_1(\mathbf{u},\mathbf{x}) \le 0, \ldots, g_q(\mathbf{u},\mathbf{x}) \le 0$$
$$f_1(\mathbf{u},\mathbf{x}) = 0, \ldots, f_r(\mathbf{u},\mathbf{x}) = 0$$

- Output: function $Z(\mathbf{u})$, $X(\mathbf{u})$ and their domain $\mathcal{U} \subset \mathbb{R}^d$.

Under certain constraint qualifications, any local and global minima of (**??**) occur at the so-called "critical points", namely the solution set defined by the following KKT conditions:

$$\left\{ \begin{array}{rcl} \bigtriangledown_{\mathbf{x}} f(\mathbf{u},\mathbf{x}) + \sum_{i=1}^{q} v_i \bigtriangledown_{\mathbf{x}} g_i(\mathbf{u},\mathbf{x}) + \sum_{i=1}^{r} w_i \bigtriangledown_{\mathbf{x}} f_i(\mathbf{u},\mathbf{x}) & = & 0 \\ f_i(\mathbf{u},\mathbf{x}) & = & 0 \\ v_i g_i(\mathbf{u},\mathbf{x}) & = & 0 \\ g_i(\mathbf{u},\mathbf{x}) & \le & 0 \\ v_i & \ge & 0 \end{array} \right. . \qquad (2)$$

# Parametric polynomial optimization by RC-CAD using KKT condition

- Algorithm: MinCAD
- Input: The minimization problem (**??**)
- Output: A set of CAD cells encoding solutions to (**??**)

1. Introduce a new variable $z$ to denote the optimal value
2. Let $S$ be the semi-algebraic system (**??**)
3. Add the equational constraint $z - f(\mathbf{u}, \mathbf{x}) = 0$ to $S$
4. Let $\mathbf{v} = \{v_1, \ldots, v_1\}$ and $\mathbf{w} = \{w_1, \ldots, w_r\}$ Define the eliminate order $\mathbf{v} > \mathbf{w} > \mathbf{x} > z > \mathbf{u}$.
5. Call RC-CAD to compute a truth invariant CAD of $\mathbb{R}^{m+d+1}$ w.r.t. $S$
6. If there are no true cells, return $\emptyset$
7. Categorizing the true cells such that the cells having the same projection onto $\mathbf{u}$-space are in the same group
8. Define $L := \emptyset$
9. For each group, add a cell whose $z$-index is the smallest into $L$
10. Let $L := \{\pi_{\mathbf{x},z,\mathbf{u}}(c) \mid c \in L\}$ and return $L$

# Parametric polynomial optimization by RC-CAD (II)

- Input:

$$\begin{aligned} z(\mathbf{u}) \quad &= \quad \min_{\mathbf{x}} f(\mathbf{x},\mathbf{u}) \text{ s.t.} \\ & \quad\quad g_1(\mathbf{u},\mathbf{x}) \le 0, \ldots, g_q(\mathbf{u},\mathbf{x}) \le 0 \\ & \quad\quad f_1(\mathbf{u},\mathbf{x}) = 0, \ldots, f_r(\mathbf{u},\mathbf{x}) = 0 \end{aligned}$$

- Output: function $Z(\mathbf{u})$, $X(\mathbf{u})$ and their domain $\mathcal{U} \subset \mathbb{R}^d$.

### Theorem

*Let $L$ be a set of CAD cells computed by* MinCAD. *Then we have*

- *If $L = \emptyset$, then no feasible solutions exist for problem (**??**).*
- *The set of parametric values, such that problem (**??**) has feasible solutions is: $\mathcal{U} = \cup_{c \in L} \pi_{\mathbf{u}}(c)$*
- *Let $C$ be a cell in $L$.*
  - *If $C^z$ is of type $z < \phi(\mathbf{u})$ or $z = z$, then $z(\mathbf{u}) = -\infty$.*
  - *If $C^z$ is of type $z = \phi(\mathbf{u})$ , then $z(\mathbf{u}) = \phi(\mathbf{u})$, the minimum is attained and $C^{>z}$ defines at least one optimizer.*
  - *If $C^z$ is of type $z > \phi(\mathbf{u})$ or $\phi(\mathbf{u}) < z < \psi(\mathbf{u})$, then $z(\mathbf{u}) = \phi(\mathbf{u})$. But the minimum is not attained.*

# An example

$$z(\mathbf{u}) = \min_{x_1,x_2} \quad x_1 + u_1 x_2$$
$$\text{s.t.} \quad u_2^2 x_1^2 - x_2 \le 0$$
$$x_1 \le 0$$

- Let $F := \emptyset$ and $G := [u_2^2 x_1^2 - x_2, x_1]$.
- The KKT system is $S := \{2v_1u_2^2x_1 + v_2 + 1 = 0, u_1 - v_1 = 0, v_1(x_1^2u_2^2 - x_2) = 0, v_2x_1 = 0, x_1^2u_2^2 - x_2 \le 0, x_1 \le 0, 0 \le v_1, 0 \le v_2\}$.
- The input system is $S := \{z - (x_1 + u_1x_2)\} \cup S$.
- The elimination order is $v_1 > v_2 > x_2 > x_1 > z > u_2 > u_1$.
- Call CylindricalAlgebraicDecompose in `RegularChains` library to compute a truth invariant CAD w.r.t. $S$. The output has $2$ cells.
- From the output, both two cells are selected to encode solutions of the minimization problem.

The solution: If $u_1 > 0$ and $u_2 \neq 0$, then $z(\mathbf{u}) = -\frac{1}{4u_1u_2^2}$, $x_1(\mathbf{u}) = -\frac{1}{2u_1u_2^2}$ and $x_2(\mathbf{u}) = \frac{1}{4u_1u_2^2}$.

# A screen shot showing the solutions obtained by using KKT condition

```
> restart; read "optimization.mpl";
> f := x_1+u_1*x_2: G:=[u_2^2*x_1^2-x_2, x_1]: F:=[]: X:=[x_2,x_1]: U:=[u_2,u_1]:
  sols, R := MinimizeByCAD(f, F, G, X, U, method='kkt');
```

"The method is sound assuming KKT condition is valid and the feasible set is bounded from below."

$$sols, R := \left[\left[cad\_cell, \_z = -\frac{1}{4\, u\_1\, u\_2^2}\right], \left[cad\_cell, \_z = -\frac{1}{4\, u\_1\, u\_2^2}\right]\right], polynomial\_ring$$

```
> Display(sols, R);
```

$$\left[\left[\left\{\begin{array}{c} \_v1 = u\_1 \\ \_v2 = 0 \\ x\_2 = \frac{-x\_1 + \_z}{u\_1} \\ x\_1 = -\frac{1}{2\, u\_1\, u\_2^2} \\ \_z = -\frac{1}{4\, u\_1\, u\_2^2} \\ u\_2 < 0 \\ 0 < u\_1 \end{array}\right., \_z = -\frac{1}{4\, u\_1\, u\_2^2}\right], \left[\left\{\begin{array}{c} \_v1 = u\_1 \\ \_v2 = 0 \\ x\_2 = \frac{-x\_1 + \_z}{u\_1} \\ x\_1 = -\frac{1}{2\, u\_1\, u\_2^2} \\ \_z = -\frac{1}{4\, u\_1\, u\_2^2} \\ 0 < u\_2 \\ 0 < u\_1 \end{array}\right., \_z = -\frac{1}{4\, u\_1\, u\_2^2}\right]\right]$$

# Plan

## An example solved by numeric method

$$
\begin{aligned}
z(\mathbf{u}) \quad = \quad & \min_{x_1,x_2} \quad x_1x_2 \\
& \text{s.t.} \quad -2x_1 - x_2 + u \leq 0 \\
& \quad -x_1 - 3x_2 + 1/2u \leq 0 \\
& \quad -x_i - 1 \leq 0, i = 1, 2 \\
& \quad x_i - 1 \leq 0, i = 1, 2 \\
& \quad -u \leq 0 \\
& \quad u - 1 \leq 0
\end{aligned}
$$

### Numeric solution

- If $0 \leq u \leq 0.5$, then $z(u) = 0.5u - 0.4922$.
- If $0.5 \leq u \leq 1$, then $z(u) = 0.1666u - 0.3255$.

### Symbolic solution

- If $0 \leq u \leq 1/2$, then $z(u) = 1/2u - 1/2$, $x_1(u) = 1/2u - 1/2$, $x_2(u) = 1$.
- If $1/2 < u \leq 1$, then $z(u) = 1/6u - 1/3$, $x_1(u) = 1$,

# Plan

# Related work

There are many related work, this slide needs to be expanded to do.

- Numerical Methods for solving MPC directly without using parametric polynomial optimization
- Numerical method for solving parametric polynomial optimization
- Symbolic approach for solving parametric polynomial optimization (Open CAD, KKT+Gröbner basis, real quantifier elimination)

# Conclusion and future work

- We introduced a complete method for solving parametric polynomial optimization by RC-CAD.
- The method can determine if an infinimum can be attained.
- We proposed also a method combining RC-CAD and the KKT condition.
- The general method can be used to verify if the use of KKT condition is valid.
- The method can solve simple yet non-trivial examples.
- Future work is needed to exploit the structure of MPC problem and the KKT condition and develop a customized RC-CAD.
- Combining with numerical method?