Explicit Model Predictive Control

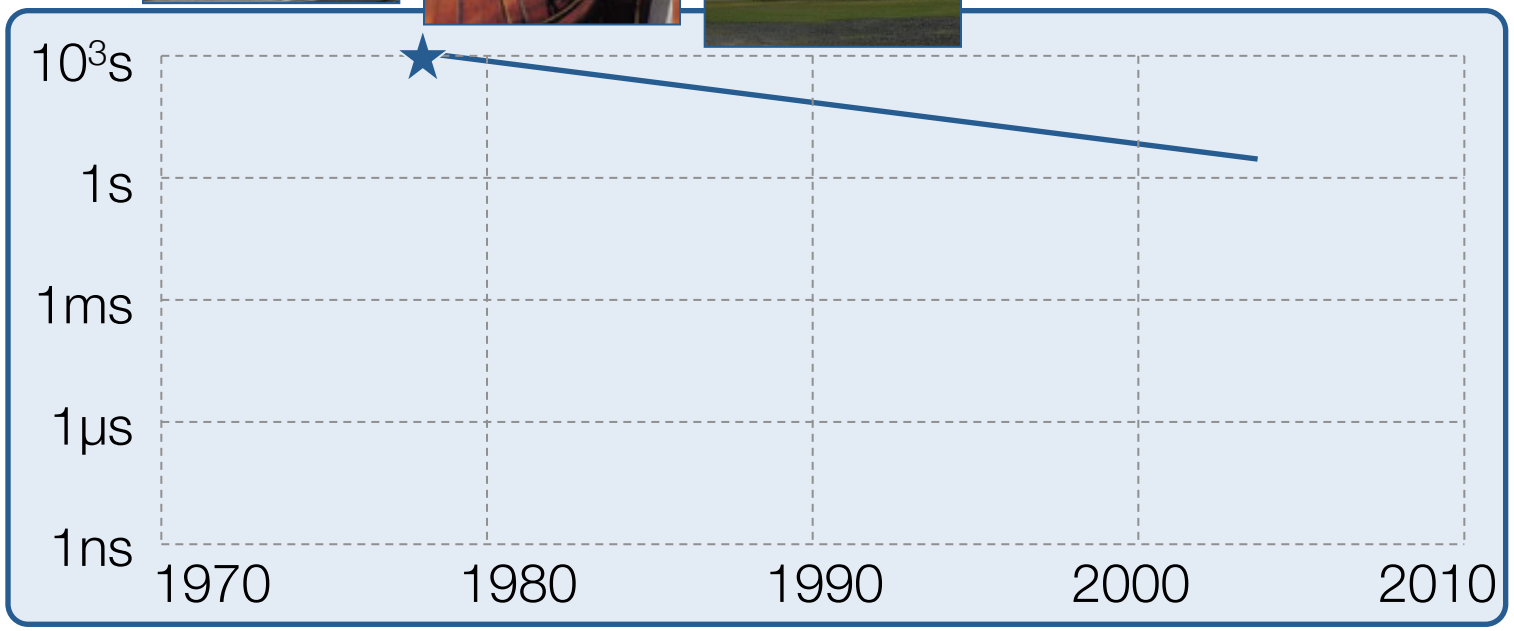
Colin Jones and Melanie Zeilinger

Automatic Control Laboratory, EPFL

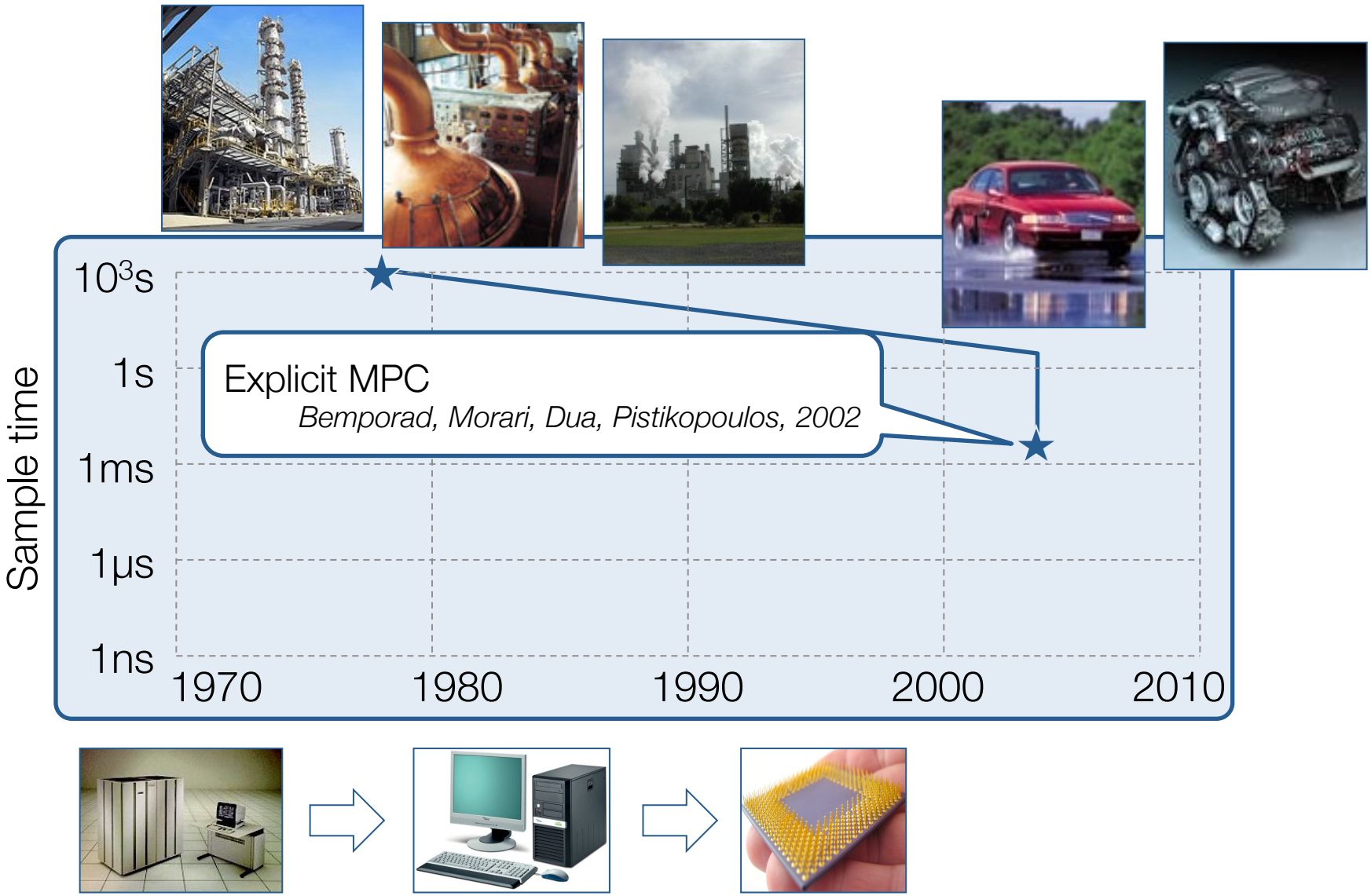
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Evolution of MPC – beyond Moore's Law

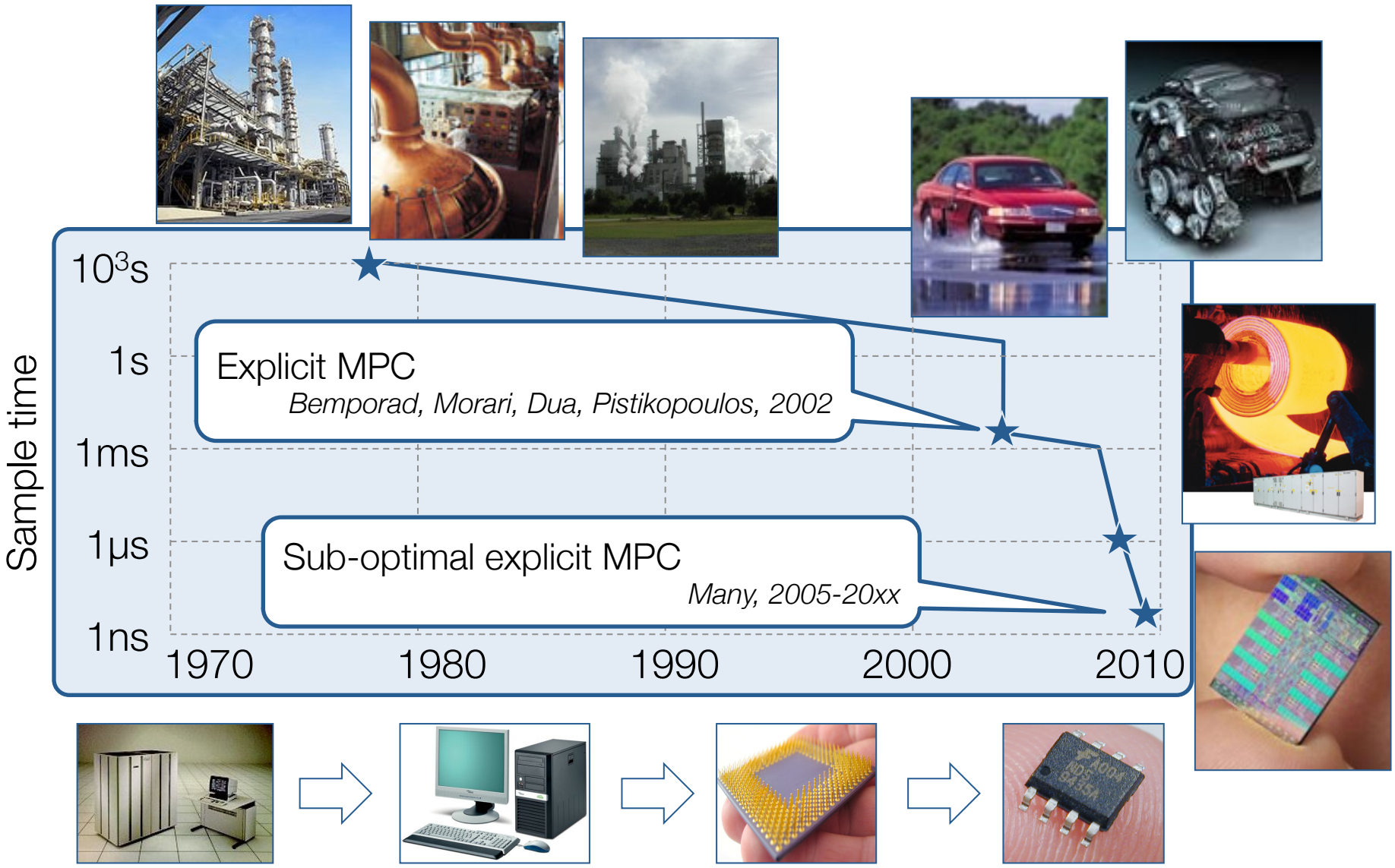




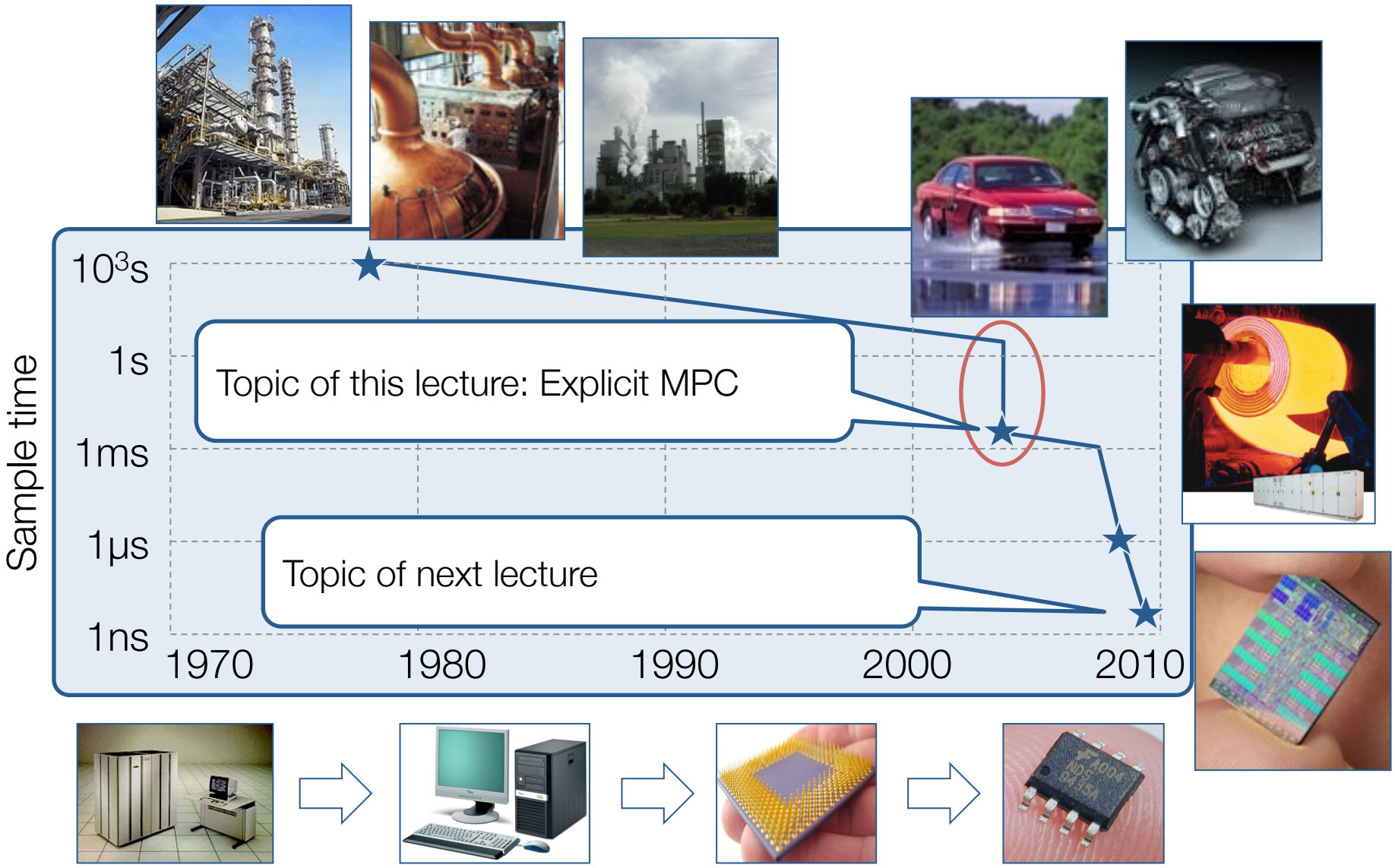

Evolution of MPC – beyond Moore's Law



Evolution of MPC – beyond Moore's Law



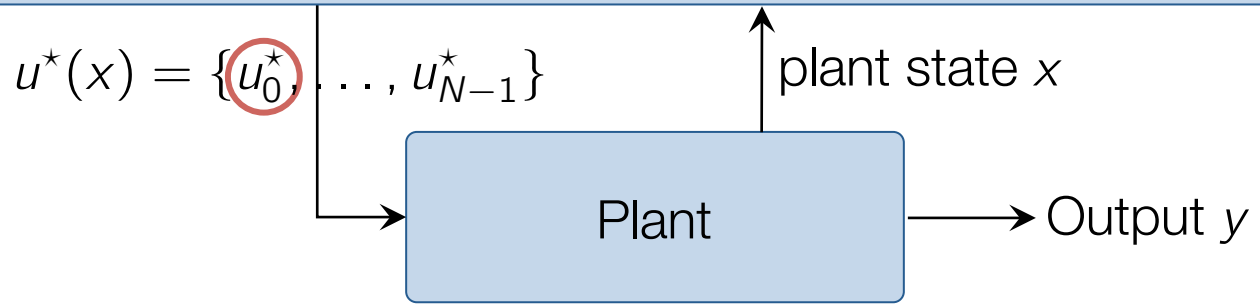
Evolution of MPC – beyond Moore's Law



Receding Horizon Control Synthesis

$$u^\star(x) := \text{argmin} \quad x_N^T Q_f x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i$$

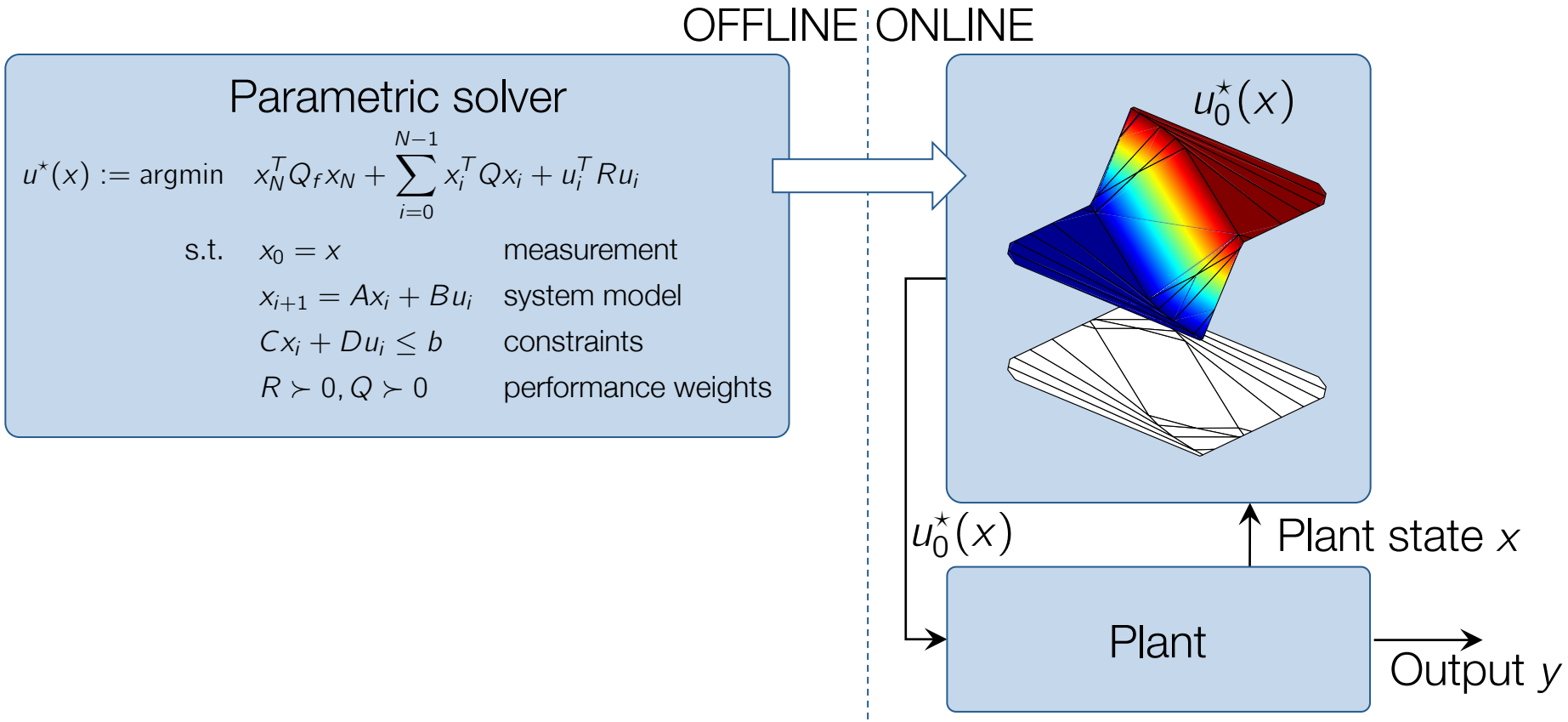
s.t.	$x_0 = x$	measurement
	$x_{i+1} = Ax_i + Bu_i$	system model
	$Cx_i + Du_i \leq b$	constraints
	$R \succ 0, Q \succ 0$	performance weights



Solve optimization problem each time sample

- Computationally complex and relatively slow
- *Not real-time*

Receding Horizon Control Synthesis

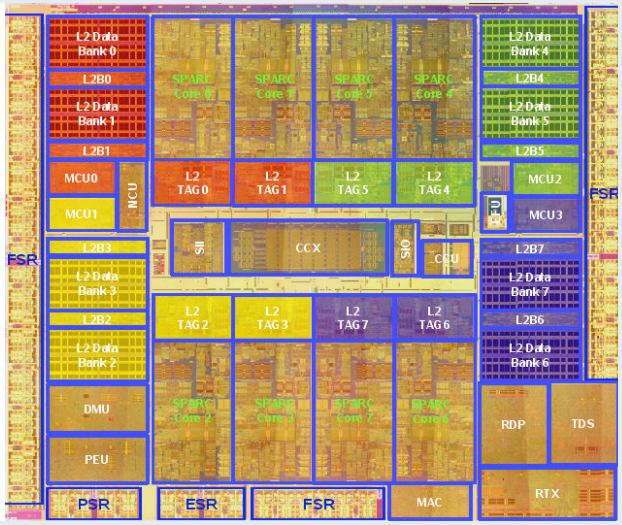


- Optimization problem is function parameterized by state
- Control law piecewise affine for linear systems/constraints
- Pre-compute control law as function of state x

Result : Online computation dramatically reduced and *real-time*

Example : How fast is fast?
Temperature Regulation of Multi-Core Processor

- Goals
  - Track workload requests
  - Minimize power usage
  - Respect temperature limits
- Quadratic nonlinear dynamics
  - Convex PWA approximation
- Stringent computational and storage requirements



$$J^\star(x_0, w) = \min_{f_i} \sum_{t=0}^{N} \sum_{i=0}^{t} (w_i - f_i)$$

$$\text{s.t. } x_{i+1} = Ax_i + Bf_i^2$$

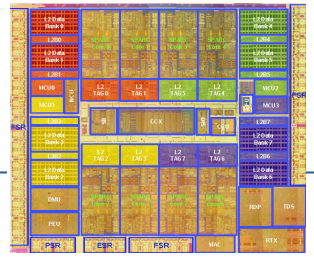
$$\sum_{i=0}^{t} w_i \leq \sum_{i=0}^{t} f_i$$

$$x_i \leq T_{\max}$$

$$f_{\min} \leq f_i \leq f_{\max}$$

[F. Zanini, C.N. Jones, D. Atienza, and G. De Micheli, 2010]

Example : How fast is fast?
Temperature Regulation of Multi-Core Processor



Work to do at time *i*

Frequency of processors at time *i* (work that can be done)

$$J^\star(x_0, w) = \min_{f_i} \sum_{t=0}^{N} \sum_{i=0}^{t} (w_i - f_i)$$

$$\text{s.t. } x_{i+1} = Ax_i + Bf_i^2$$

Temperature *x* is a quadratic function of frequency (Have to approximate)

$$\sum_{i=0}^{t} w_i \leq \sum_{i=0}^{t} f_i$$

Can't do work before it's requested

$$x_i \leq T_{\max}$$

Don't overheat

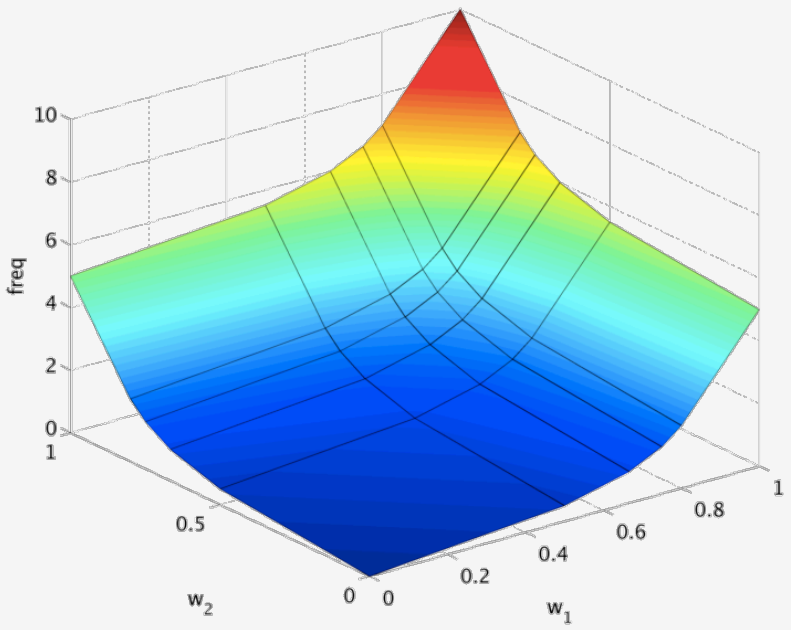
$$f_{\min} \leq f_i \leq f_{\max}$$

Clock frequency is bounded

Multi-core thermal regulation : Control law

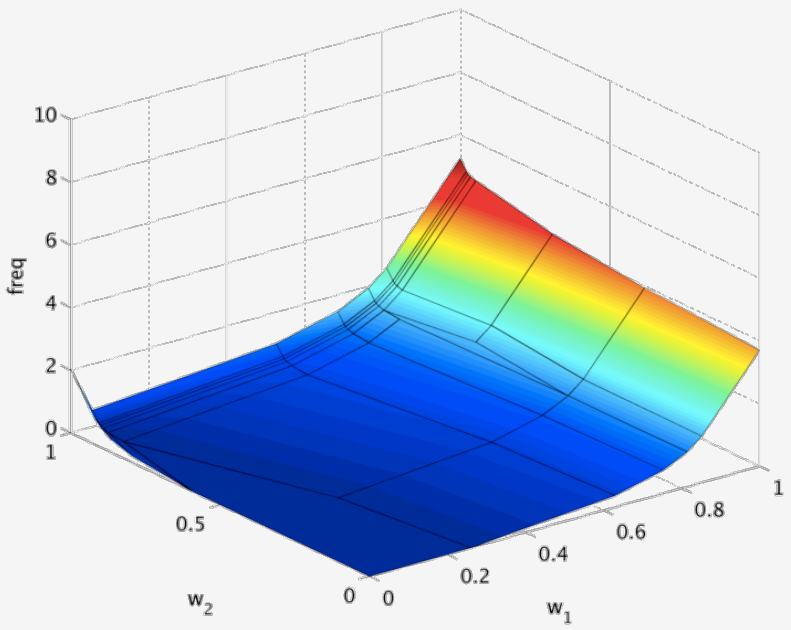
Cold chip

Do all requested work



Warm chip

Move work to cooler cores



Example : How fast is fast?

Offline processing

- Time to compute control law : 196 sec

Online processing

- Required storage : 17’969 numbers (71 kB)
- Required online computation : 10’737 FLOPS

Result

- Possible to compute control action in *145 ns*
  - (Assuming 70 GFLOPS)
- Compare to commercial optimizer CPLEX : 4’120’000 FLOPS (~59 us)

We’ll see in the third lecture that it’s possible to go much faster!

- Or use much slower/cheaper computational platform

Outline

- Motivating Example
- MPC = Parametric Quadratic Programming
- Parametric Linear Complementarity Problems
  - The Geometry
  - The Algebra
  - Efficient Solution Methods
- Online Computation : Point Location Problem
- Example

MPC $\rightarrow$ Parametric programming

$$J^\star(x) = \min\ x_N^T Q_f x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i$$

$$\text{s.t. } x_0 = x$$

$$x_{i+1} = Ax_i + Bu_i$$

$$Cx_i + Du_i \leq b$$

Linear, quadratic or convex piecewise affine cost functions. Tracking and regulation.

Linear (or affine) dynamics

Linear constraints on states and inputs

Equivalent representation

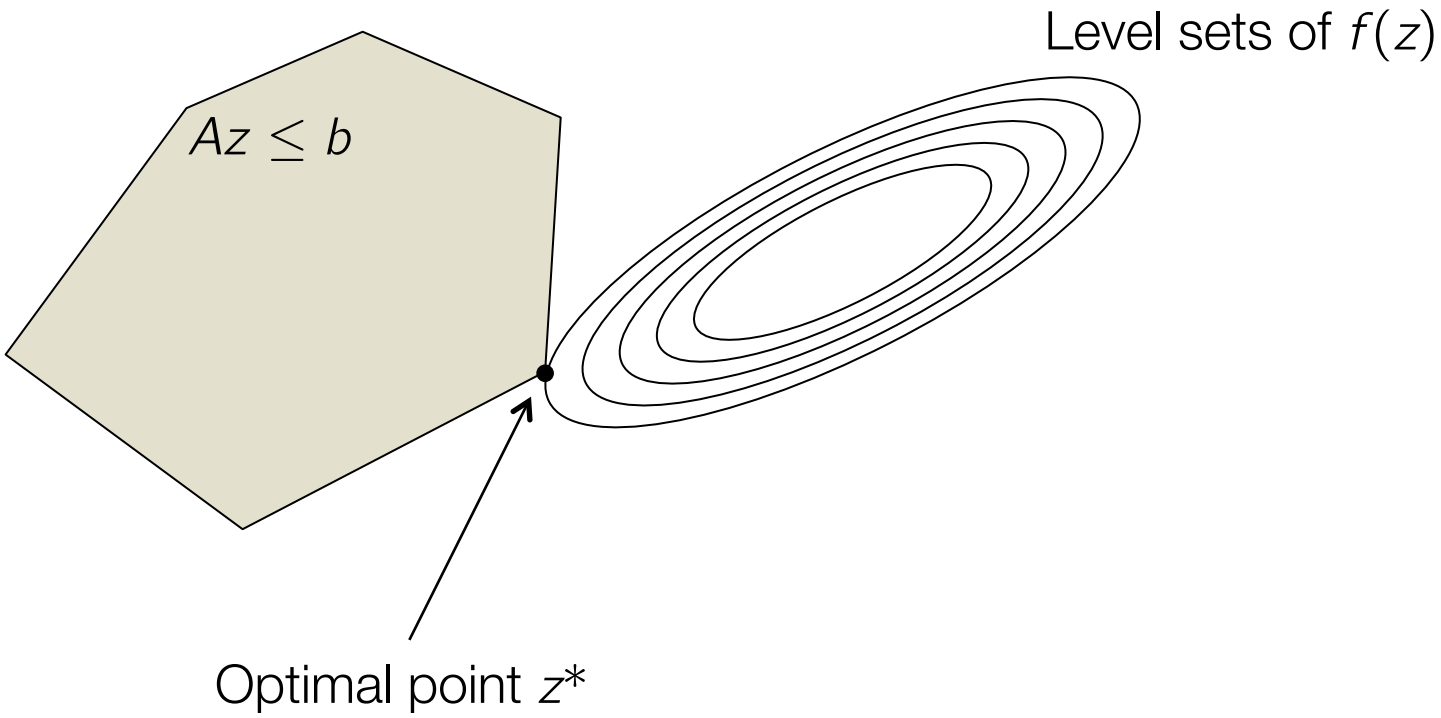
$$J^\star(x) = \min_u\ \frac{1}{2} u^T Q u + (Fx + f)^T u$$

$$\text{s.t. } Gu \leq Ex + e$$

It is also possible to represent piecewise affine systems or mixed-logic dynamic systems as *parametric mixed-integer programs*, but this is not covered in this tutorial.

QP Optimality Conditions

$$\min f(z) := \frac{1}{2} z^T Q z$$
$$\text{s.t. } Az \leq b$$



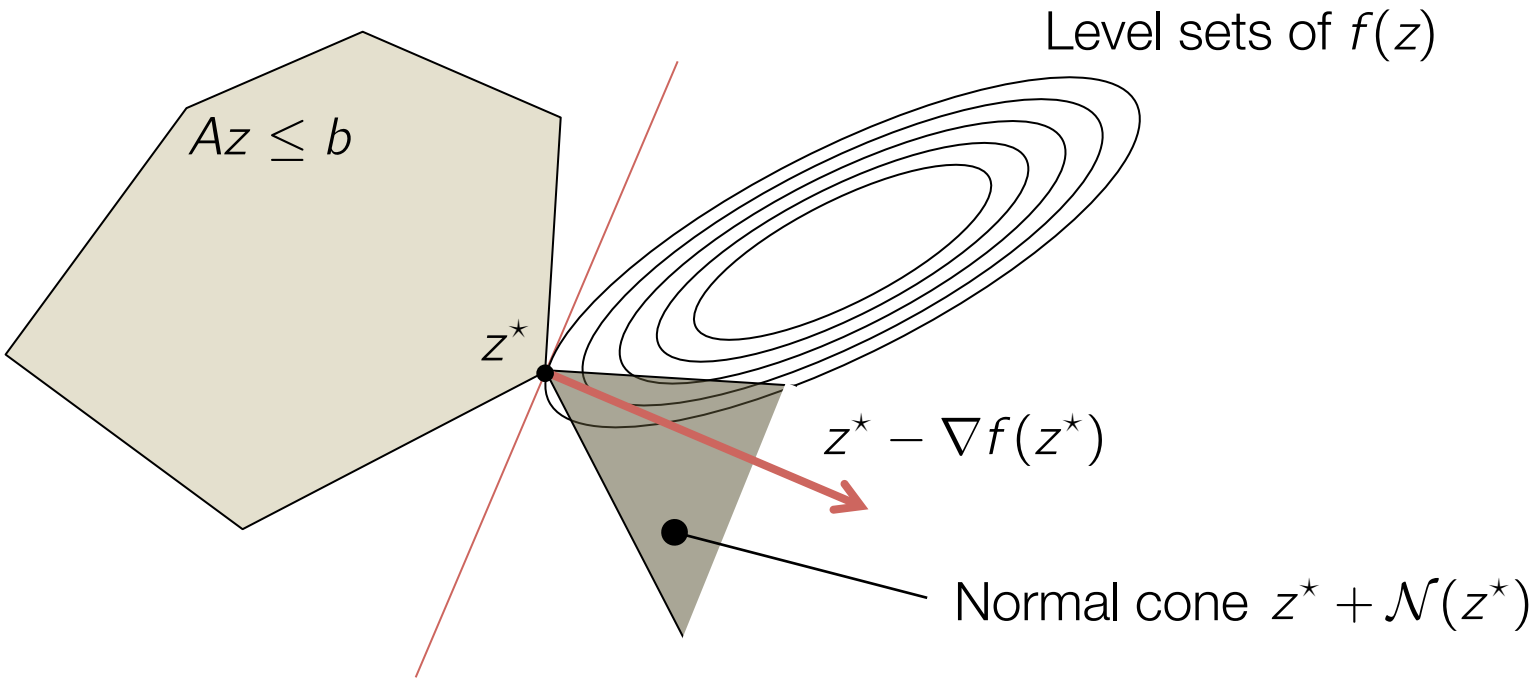
QP Optimality Conditions

$$\min f(z) := \frac{1}{2} z^T Q z$$
$$\text{s.t. } Az \leq b$$

Necessary optimality condition:

$$-\nabla f(z^\star) \in \mathcal{T}(z^\star)^\star = \mathcal{N}(z^\star)$$

(Negative gradient is in the normal cone)



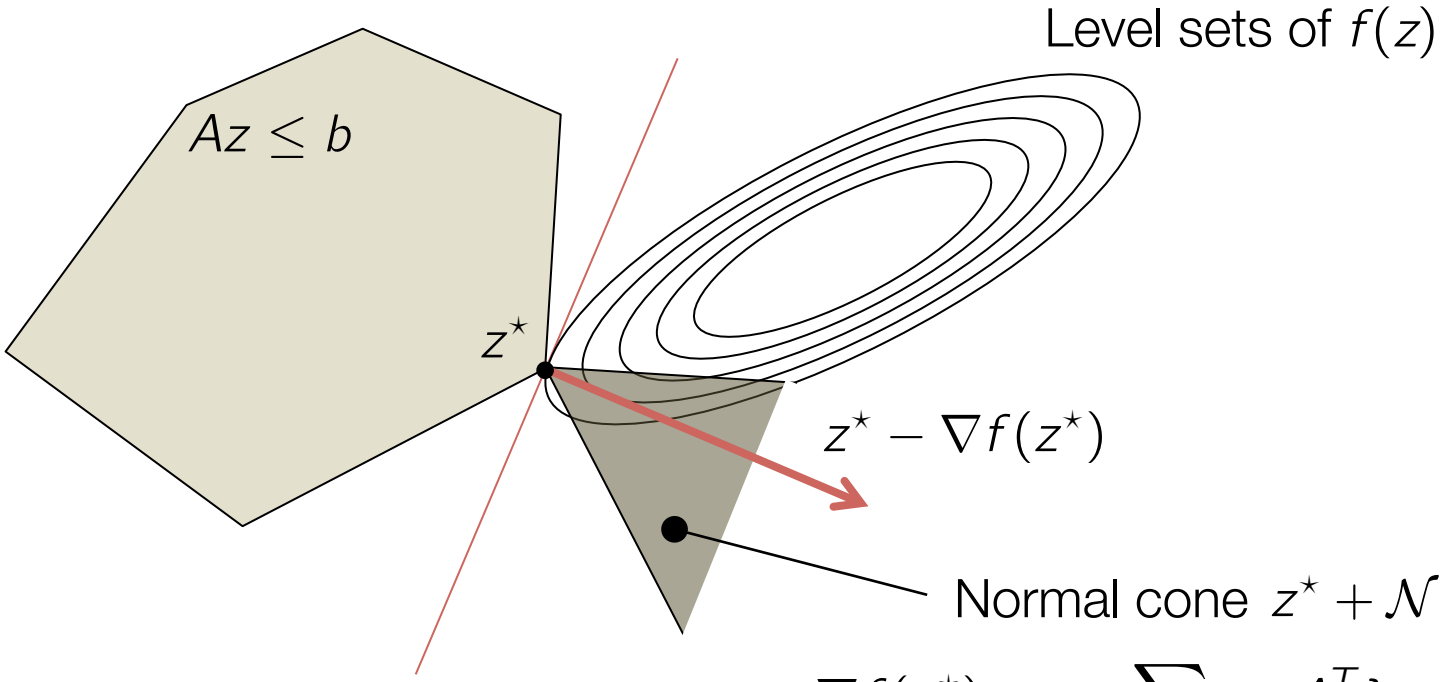
QP Optimality Conditions

$$\min f(z) := \frac{1}{2} z^T Q z$$
$$\text{s.t. } Az \leq b$$

Necessary optimality condition:

$$-\nabla f(z^\star) \in \mathcal{T}(z^\star)^\star = \mathcal{N}(z^\star)$$

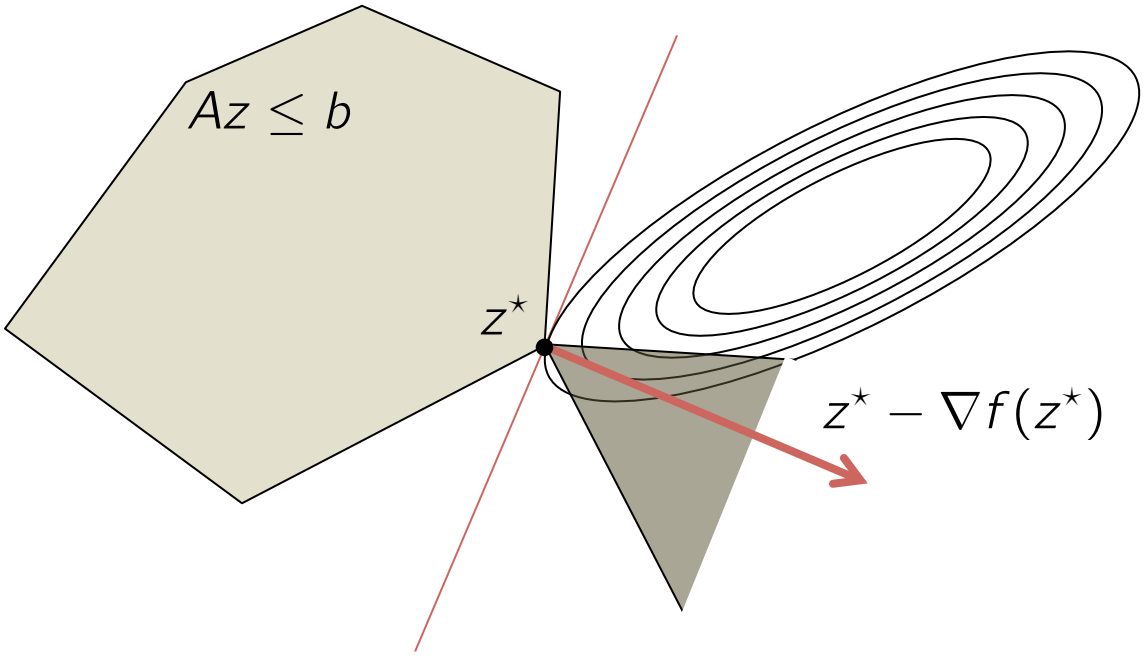
(Negative gradient is in the normal cone)



$$\nabla f(z^\star) = \sum_{\{i \mid A_i z^\star = b_i\}} A_i^T \lambda_i, \ \lambda_i \geq 0$$

QP Optimality Conditions

$$\min f(z) := \frac{1}{2} z^T Q z$$
$$\text{s.t. } Az \leq b$$



KKT Necessary and Sufficient Optimality Conditions for ConvexQPs

$Qz = A^T\lambda,\ \lambda \geq 0$	Gradient is in the normal cone
$Az \leq b$	Optimal point must be feasible
$\lambda^T(Az - b) = 0$	Normal cone contains only active constraints

Simple Parametric Programming Example

One-dimensional example

$$f^\star(x) = \min_z \ \frac{1}{2}z^2 + 2xz$$
$$\text{s.t. } z \geq x - 1$$
$$z \geq 0$$

Find:

- Optimizer $z(x)$
- All x for which problem has a solution
- Value function $f^*(x)$

- KKT conditions

$$\nabla_z \mathcal{L} = z + 2x - \lambda - \nu = 0 \quad \text{Stationarity}$$
$$x - 1 - z \leq 0, \ z \geq 0 \quad \text{Primal feasibility}$$
$$\lambda, \nu \geq 0 \quad \text{Dual feasibility}$$
$$\lambda(z - x - 1) = \nu z = 0 \quad \text{Complementarity}$$

Simple Parametric Programming Example

$$\nabla_z \mathcal{L} = z + 2x - \lambda - \nu = 0 \quad \text{Stationarity}$$
$$x - 1 - z \leq 0,\ z \geq 0 \quad \text{Primal feasibility}$$
$$\lambda, \nu \geq 0 \quad \text{Dual feasibility}$$
$$\lambda(z - x - 1) = \nu z = 0 \quad \text{Complementarity}$$

Four complementarity cases:

$$\begin{matrix}\lambda = 0 & z \geq x - 1 \\ \nu = 0 & z \geq 0\end{matrix} \rightarrow \begin{cases} z^\star(x) & = -2x \\ f^\star(x) & = -2x^2 \\ x & \leq 0 \end{cases}$$

$$\begin{matrix}\lambda = 0 & z \geq x - 1 \\ \nu \geq 0 & z = 0\end{matrix} \rightarrow \begin{cases} z^\star(x) = 0 \\ f^\star(x) = 0 \\ 0 \leq x \leq 1 \end{cases}$$

$$\begin{matrix}\lambda \geq 0 & z = x - 1 \\ \nu = 0 & z \geq 0\end{matrix} \rightarrow \begin{cases} z^\star(x) & = x - 1 \\ f^\star(x) & = \frac{5}{2}x^2 - 3x + \frac{1}{2} \\ x & \geq 1 \end{cases}$$

$$\begin{matrix}\lambda \geq 0 & z = x - 1 \\ \nu \geq 0 & z = 0\end{matrix} \rightarrow \begin{cases} z^\star(x) & = 0 \\ f^\star(x) & = 0 \\ x & = 1 \end{cases}$$

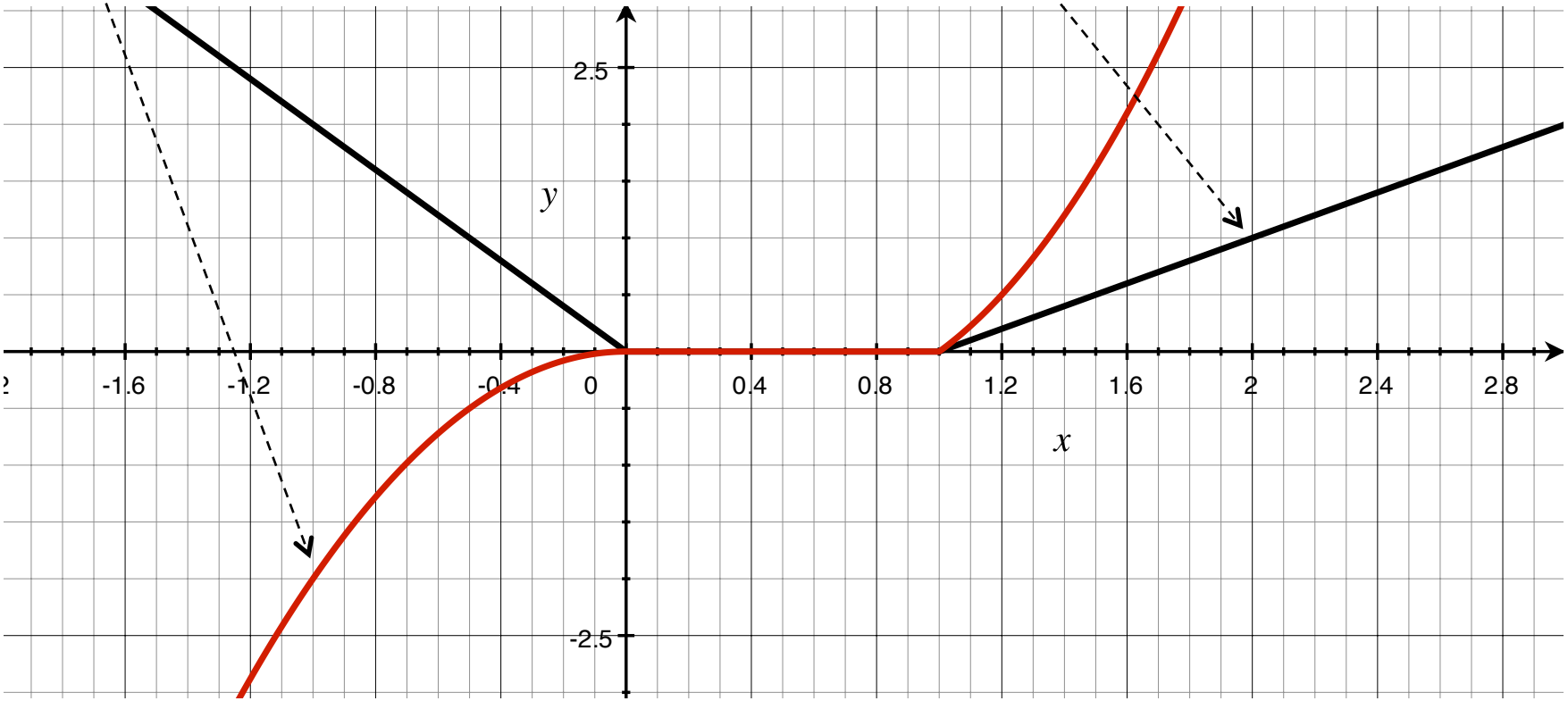
Simple Parametric Programming Example

Optimal value: Piecewise quadratic

$$f^\star(x) = \begin{cases} -2x^2 & x \leq 0 \\ 0 & 0 \leq x \leq 1 \\ \frac{5}{2}x^2 - 3x + \frac{1}{2} & x \geq 1 \end{cases}$$

Optimizer : Piecewise affine

$$z^\star(x) = \begin{cases} -2x & x \leq 0 \\ 0 & 0 \leq x \leq 1 \\ x - 1 & x \geq 1 \end{cases}$$



Outline

- Motivating Example
- MPC = Parametric Quadratic Programming
- Parametric Linear Complementarity Problems
  - The Geometry
  - The Algebra
  - Efficient Solution Methods
- Online Computation : Point Location Problem
- Example

General Formulation: Parametric Linear Complementarity

Parametric Linear Complementarity Problem

Given matrices M, q and Q, find functions $w(x), z(x)$ such that

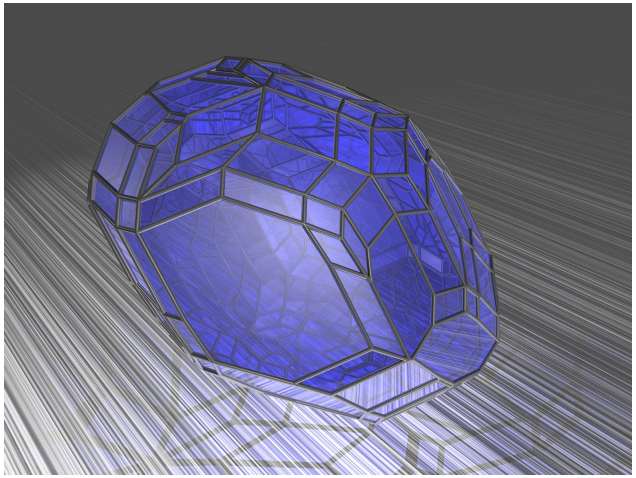
$$w - Mz = q + Qx$$
$$w^T z = 0$$
$$w, z \geq 0$$

- General form of the KKT conditions for quadratic programming
- Encompasses RIM linear and quadratic programming
  - Linear parameters in the cost and RHS
- Includes 'standard' MPC problems

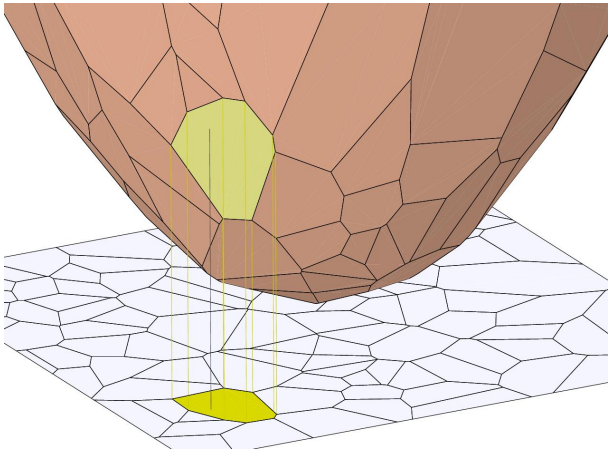
… but much more general

Geometric Problems Solvable by Parametric Linear Complementary Programming

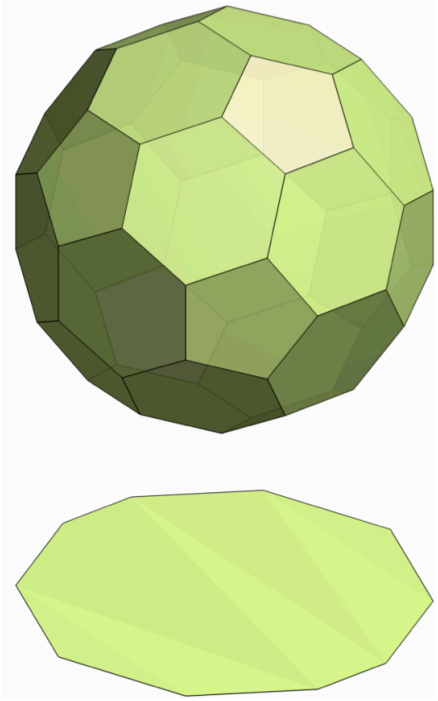
Invariant Sets



Voronoi diagrams



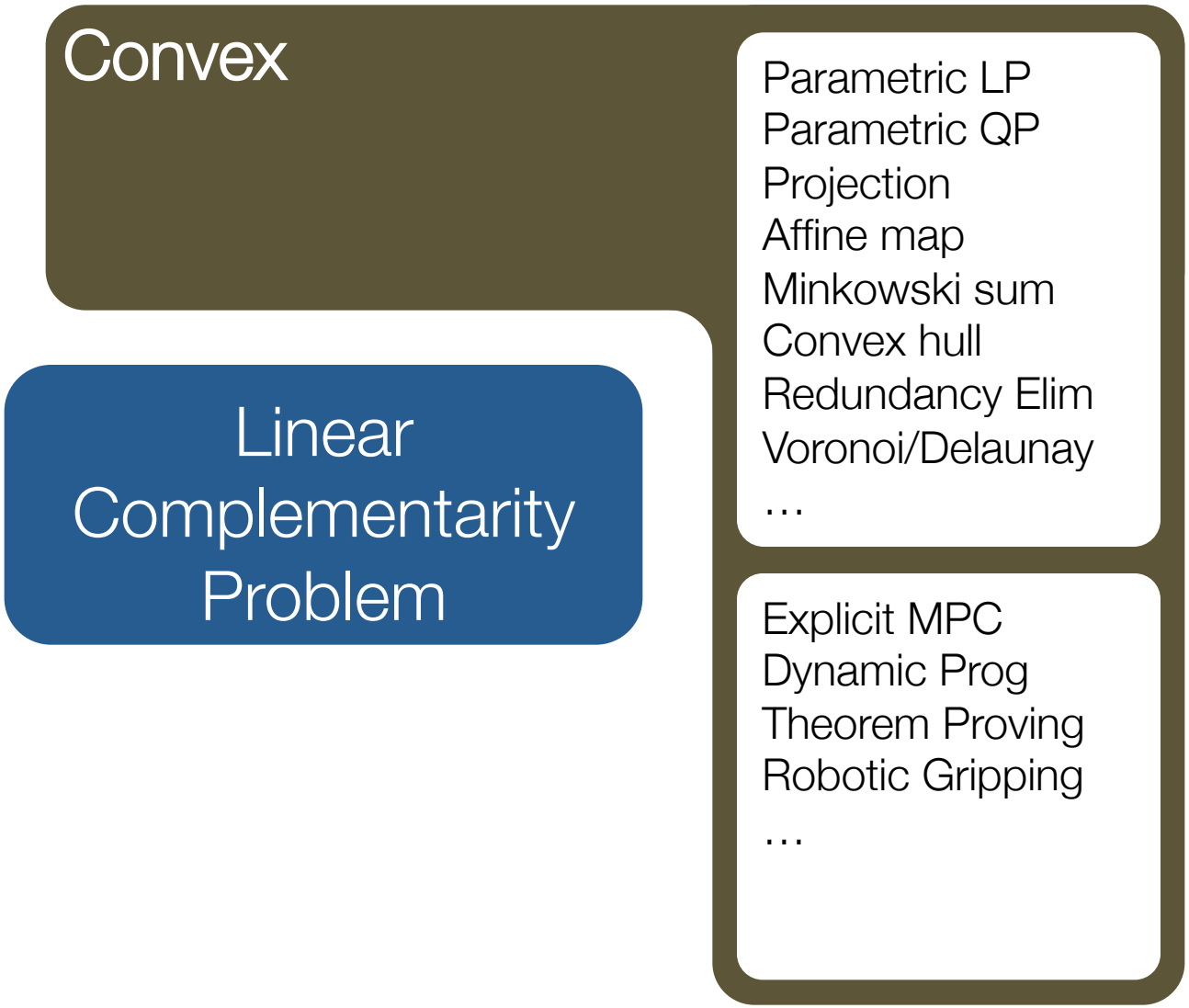
Polytopic projection



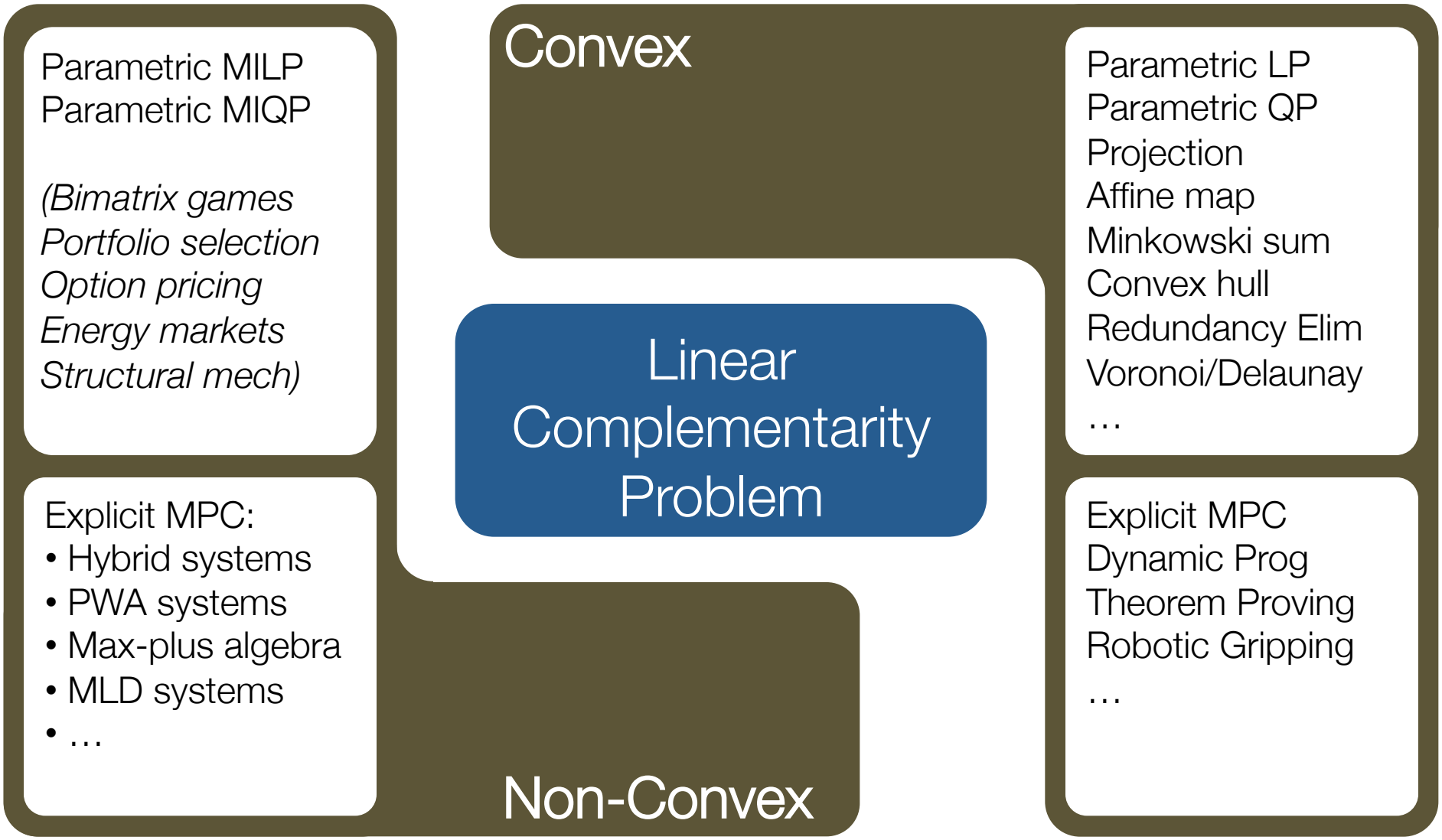
Parametric LP
Parametric QP
Projection
Affine map
Minkowski sum
Convex hull
Redundancy Elim
Voronoi/Delaunay
…

Explicit MPC
Dynamic Prog
Theorem Proving
Robotic Gripping
…

Geometric Problems Solvable by Parametric Linear Complementary Programming



Geometric Problems Solvable by Parametric Linear Complementary Programming



Geometric Problems Solvable by Parametric Linear Complementary Programming

This tutorial covers the 'convex' class (LCPs with sufficient matrices)

Convex

Parametric LP
Parametric QP
Projection
Affine map
Minkowski sum
Convex hull
Redundancy Elim
Voronoi/Delaunay
…

Linear Complementarity Problem

Explicit MPC
Dynamic Prog
Theorem Proving
Robotic Gripping
…

Conversion of pQP to pLCP

Parametric quadratic optimization problem:

$$J^\star(x) := \min_u \ \frac{1}{2} u^T Q u + (Fx + f)^T u$$
$$\text{s.t. } Gu \geq Ex + e$$
$$u \geq 0$$

KKT Conditions:

$Qu + Fx + f - G^T\lambda - \nu = 0$	Stationarity
$-s + Gu = Ex + e\ ,\ u \geq 0$	Primary feasibility
$\lambda,\ \nu \geq 0$	Dual feasibility
$\nu^T u = 0\ ,\ \lambda^T s = 0$	Complementarity

Note: Quadratic program is in slightly different form to make the derivation of the LCP simpler. This is always possible through a simple change of variables.

Conversion of pQP to pLCP

KKT Conditions:

$$Qu + Fx + f - G^T\lambda - \nu = 0 \qquad \text{Stationarity}$$

$$-s + Gu = Ex + e\ ,\ u \geq 0 \qquad \text{Primary feasibility}$$

$$\lambda,\ \nu \geq 0 \qquad \text{Dual feasibility}$$

$$\nu^T u = 0\ ,\ \lambda^T s = 0 \qquad \text{Complementarity}$$

Stationarity
Primal feasibility

$$\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{pmatrix} \nu \\ s \end{pmatrix} - \begin{bmatrix} Q & -G^T \\ G & 0 \end{bmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{bmatrix} F \\ -E \end{bmatrix} x + \begin{bmatrix} f \\ -e \end{bmatrix}$$

Primal and dual feasibility $\nu, s, u, \lambda \geq 0$

Complementarity $\nu^T u = s^T \lambda = 0$

Conversion of pQP to pLCP

KKT Conditions:

$$Qu + Fx + f - G^T\lambda - \nu = 0 \qquad \text{Stationarity}$$

$$-s + Gu = Ex + e\ ,\ u \geq 0 \qquad \text{Primary feasibility}$$

$$\lambda,\ \nu \geq 0 \qquad \text{Dual feasibility}$$

$$\nu^T u = 0\ ,\ \lambda^T s = 0 \qquad \text{Complementarity}$$

Stationarity
Primal feasibility

$$\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{pmatrix} \nu \\ s \end{pmatrix} - \begin{bmatrix} Q & -G^T \\ G & 0 \end{bmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{bmatrix} F \\ -E \end{bmatrix} x + \begin{bmatrix} f \\ -e \end{bmatrix}$$

Primal and dual feasibility $\nu, s, u, \lambda \geq 0$

Complementarity $\nu^T u = s^T \lambda = 0$

Standard pLCP:

$$Iw - Mz = Qx + q$$

$$w, z \geq 0\ ,\ w^T z = 0$$

Note: Optimizer is linear transform of pLCP solution $u^\star(x) = \begin{bmatrix} I & 0 \end{bmatrix} z(x)$

Parametric Linear Complementarity

Parametric Linear Complementarity Problem

Given matrices M, q and Q, find functions $w(x), z(x)$ such that

$$w - Mz = q + Qx$$
$$w^T z = 0$$
$$w, z \geq 0$$

- Recall:
  - MPC control law is a linear transformation of z
  - Can represent 'standard' MPC problems as pLCPs
- Key questions:
  - What is the domain of w and z?
  - What class of functions are w and z? (and hence the control law)
  - How can we efficiently compute w and z?

Outline

- Motivating Example
- MPC = Parametric Quadratic Programming
- Parametric Linear Complementarity Problems
  - The Geometry
  - The Algebra
  - Efficient Solution Methods
- Online Computation : Point Location Problem
- Example

Geometry of the LCP

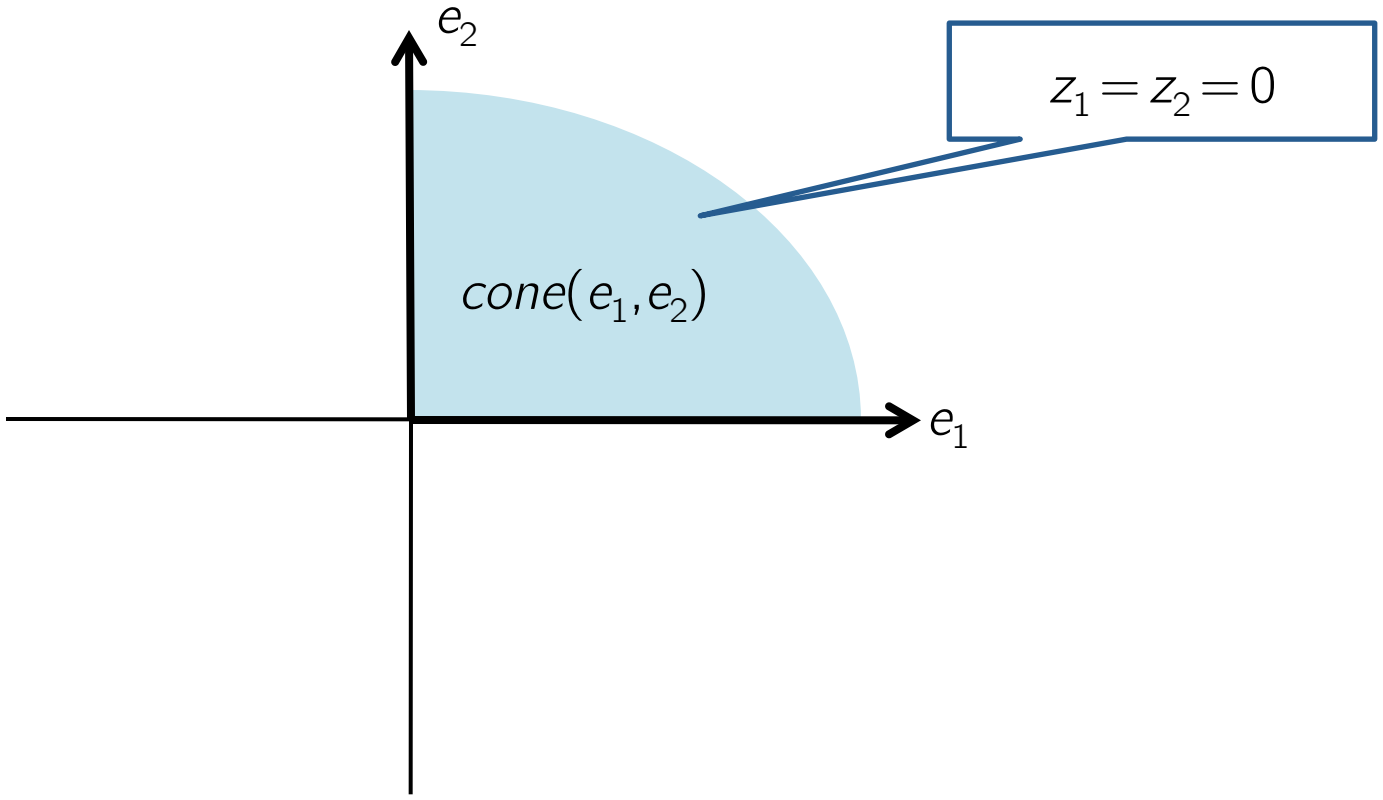
LCP feasibility conditions

1. $w^T z = 0,\ w, z \geq 0 \quad \rightarrow$ either w_i or z_i is zero for all i
2. $Iw - Mz = q \quad \rightarrow$ q is in the cone of non-zero variables

Geometry of the LCP

LCP feasibility conditions

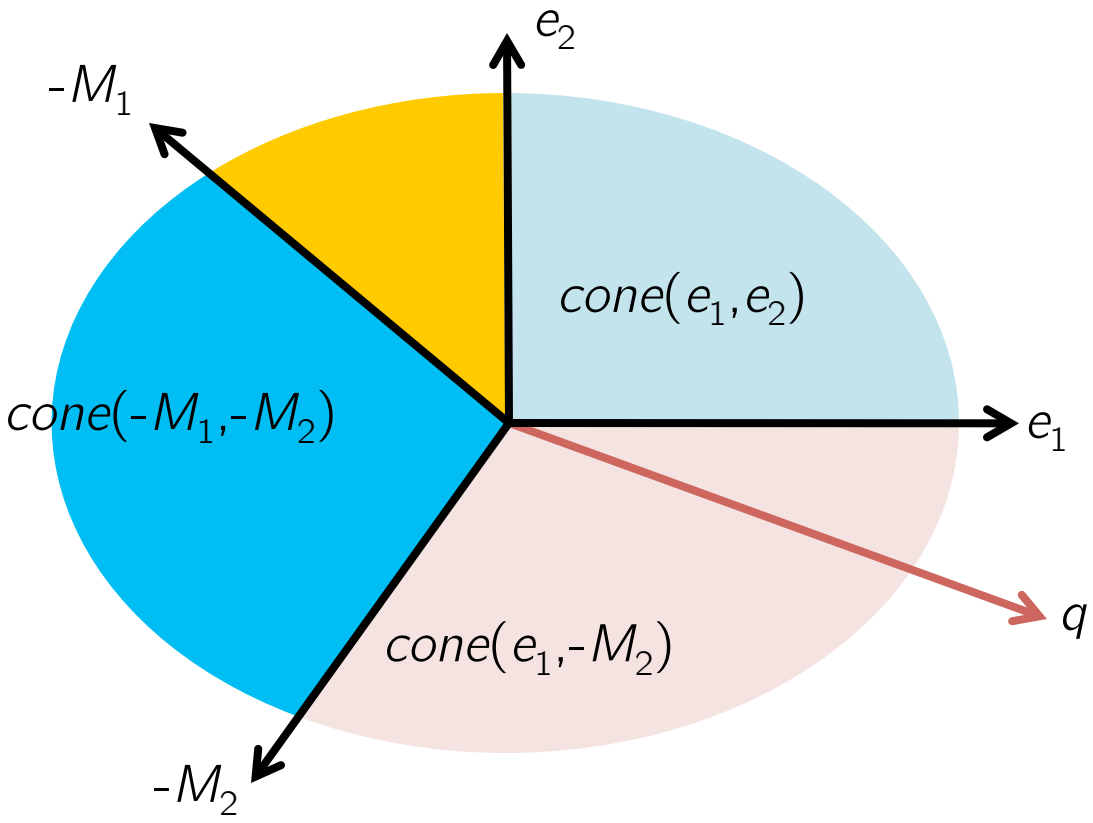
1. $w^T z = 0,\ w, z \geq 0$ $\rightarrow$ either w_i or z_i is zero for all i
2. $Iw - Mz = q$ $\rightarrow$ q is in the cone of non-zero variables



Geometry of the LCP

LCP feasibility conditions

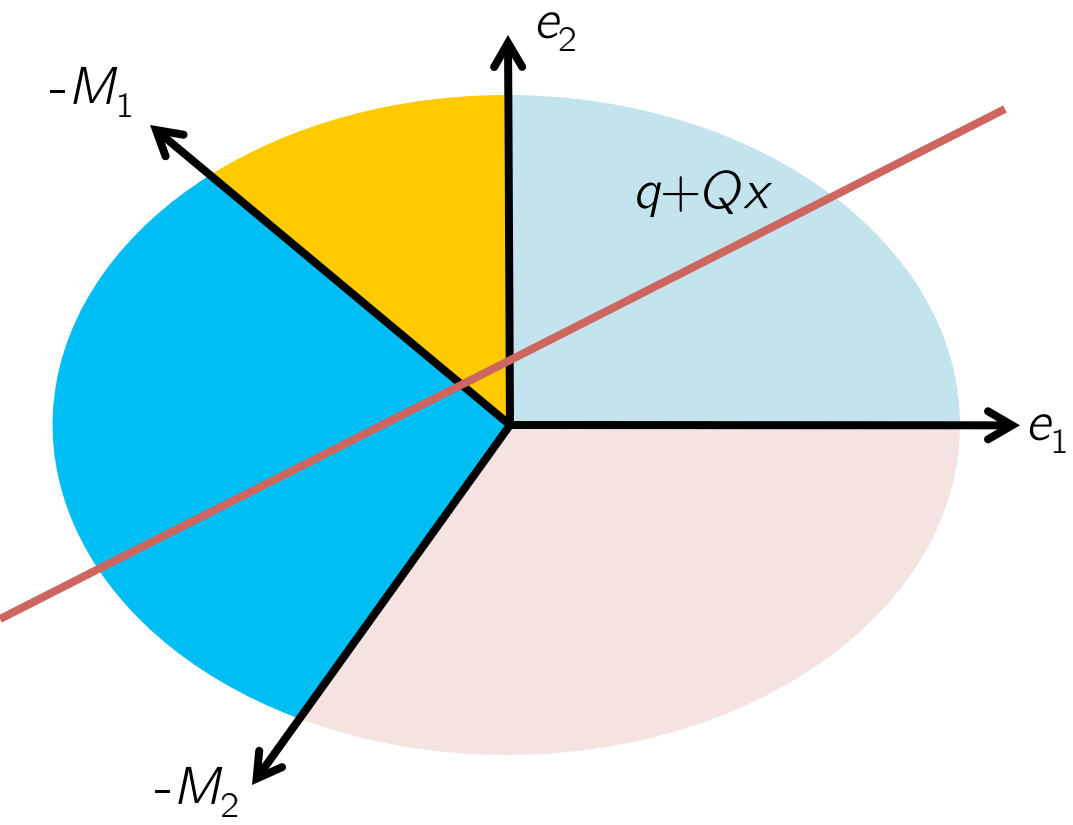
1. $w^T z = 0,\ w, z \geq 0$ $\rightarrow$ either w_i or z_i is zero for all i
2. $Iw - Mz = q$ $\rightarrow$ q is in the cone of non-zero variables



Goal: Find cone containing q

Geometry of the Parametric LCP

Goal: Find all cones intersecting $q + Qx$



$$w(x) - Mz(x) = q + Qx$$
$$w(x)^T z(x) = 0$$
$$w(x), z(x) \geq 0$$

Re-visit Simple Example

Parametric QP

$$f^\star(x) = \min_z \ \frac{1}{2}z^2 + 2xz$$
$$\text{s.t. } z \geq x - 1$$
$$z \geq 0$$

KKT Conditions

$$\nabla_z \mathcal{L} = z + 2x - \lambda - \nu = 0 \quad \text{Stationarity}$$
$$x - 1 - z + s = 0, \ s, z \geq 0 \quad \text{Primal feasibility}$$
$$\lambda, \nu \geq 0 \quad \text{Dual feasibility}$$
$$\lambda(z - x - 1) = \nu z = 0 \quad \text{Complementarity}$$

Re-visit Simple Example

Parametric QP

$$f^\star(x) = \min_z \ \frac{1}{2}z^2 + 2xz$$
$$\text{s.t. } z \geq x - 1$$
$$z \geq 0$$

KKT Conditions

$$\nabla_z \mathcal{L} = z + 2x - \lambda - \nu = 0 \quad \text{Stationarity}$$
$$x - 1 - z + s = 0, \quad s, z \geq 0 \quad \text{Primal feasibility}$$
$$\lambda, \nu \geq 0 \quad \text{Dual feasibility}$$
$$\lambda(z - x - 1) = \nu z = 0 \quad \text{Complementarity}$$

Equivalent pLCP

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \nu \\ s \end{pmatrix} - \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} z \\ \lambda \end{pmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{pmatrix} \nu \\ s \end{pmatrix}^T \begin{pmatrix} z \\ \lambda \end{pmatrix} = 0 \qquad \nu, s, z, \lambda \geq 0$$

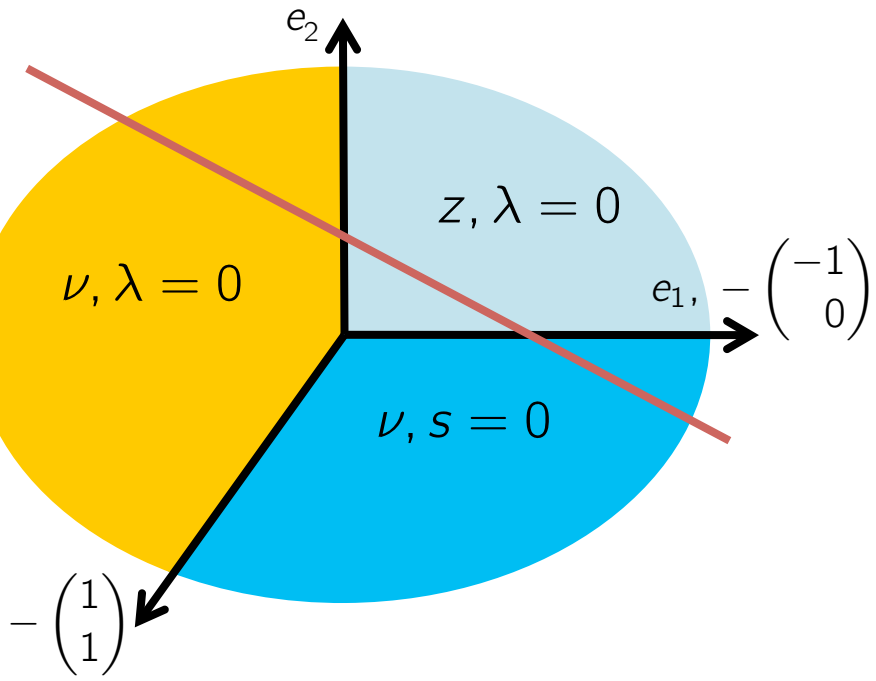
Re-visit Simple Example

Parametric QP

$$f^\star(x) = \min_z \frac{1}{2}z^2 + 2xz$$
$$\text{s.t. } z \geq x - 1$$
$$z \geq 0$$

Parametric Linear Complementarity Problem

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \nu \\ s \end{pmatrix} - \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} z \\ \lambda \end{pmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\begin{pmatrix} \nu \\ s \end{pmatrix}^T \begin{pmatrix} z \\ \lambda \end{pmatrix} = 0 \qquad \nu, s, z, \lambda \geq 0$$



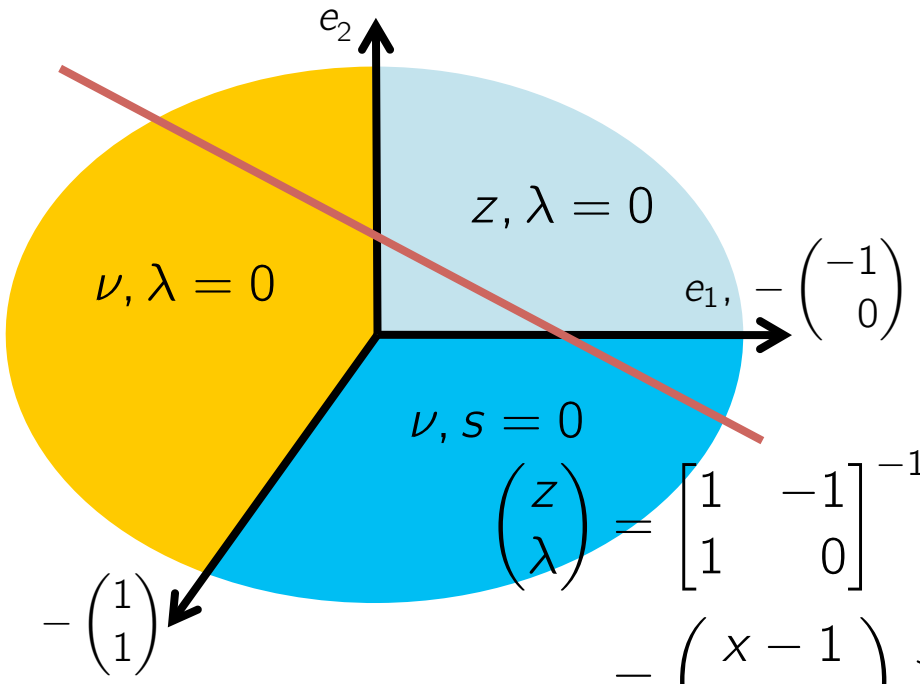
Re-visit Simple Example

Parametric QP

$$f^\star(x) = \min_z \frac{1}{2}z^2 + 2xz$$
$$\text{s.t. } z \geq x - 1$$
$$z \geq 0$$

Parametric Linear Complementarity Problem

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \nu \\ s \end{pmatrix} - \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} z \\ \lambda \end{pmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\begin{pmatrix} \nu \\ s \end{pmatrix}^T \begin{pmatrix} z \\ \lambda \end{pmatrix} = 0 \qquad \nu, s, z, \lambda \geq 0$$



$$\begin{pmatrix} z \\ \lambda \end{pmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} 2 \\ -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$
$$= \begin{pmatrix} x - 1 \\ 3x - 1 \end{pmatrix} \geq 0$$

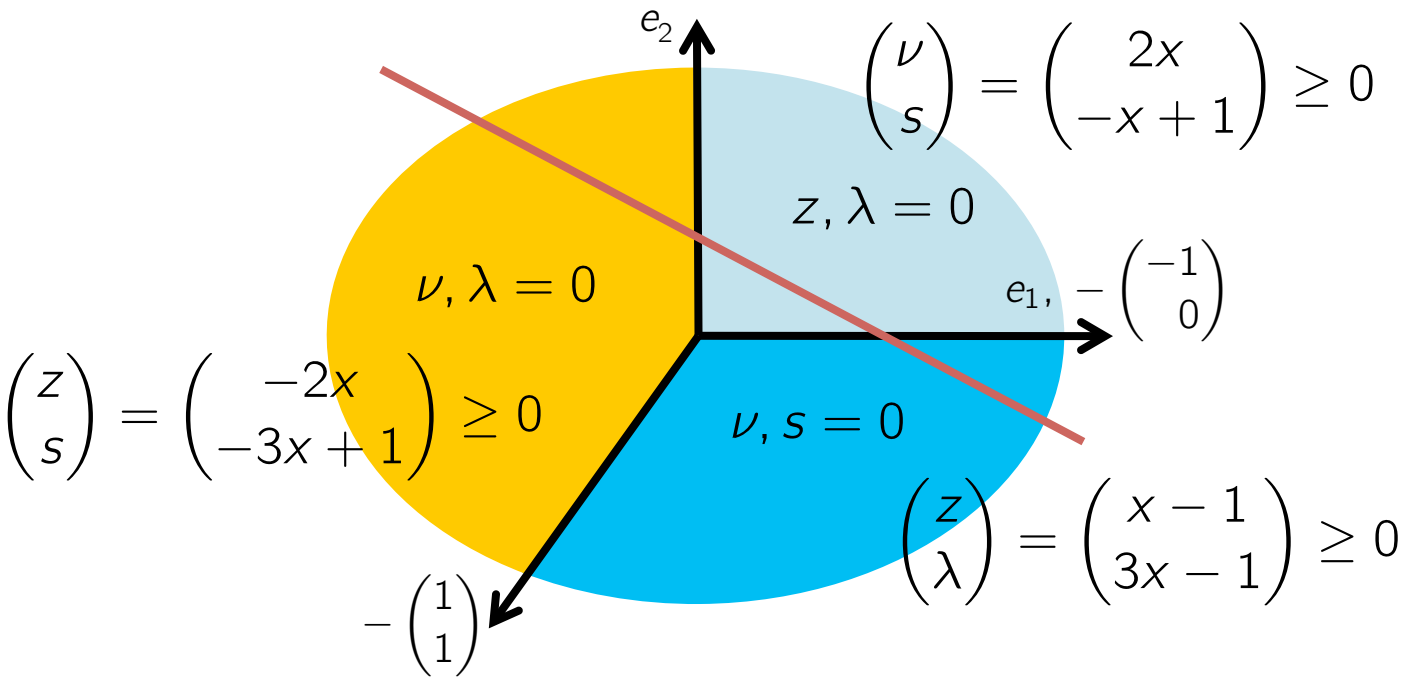
Re-visit Simple Example

Parametric QP

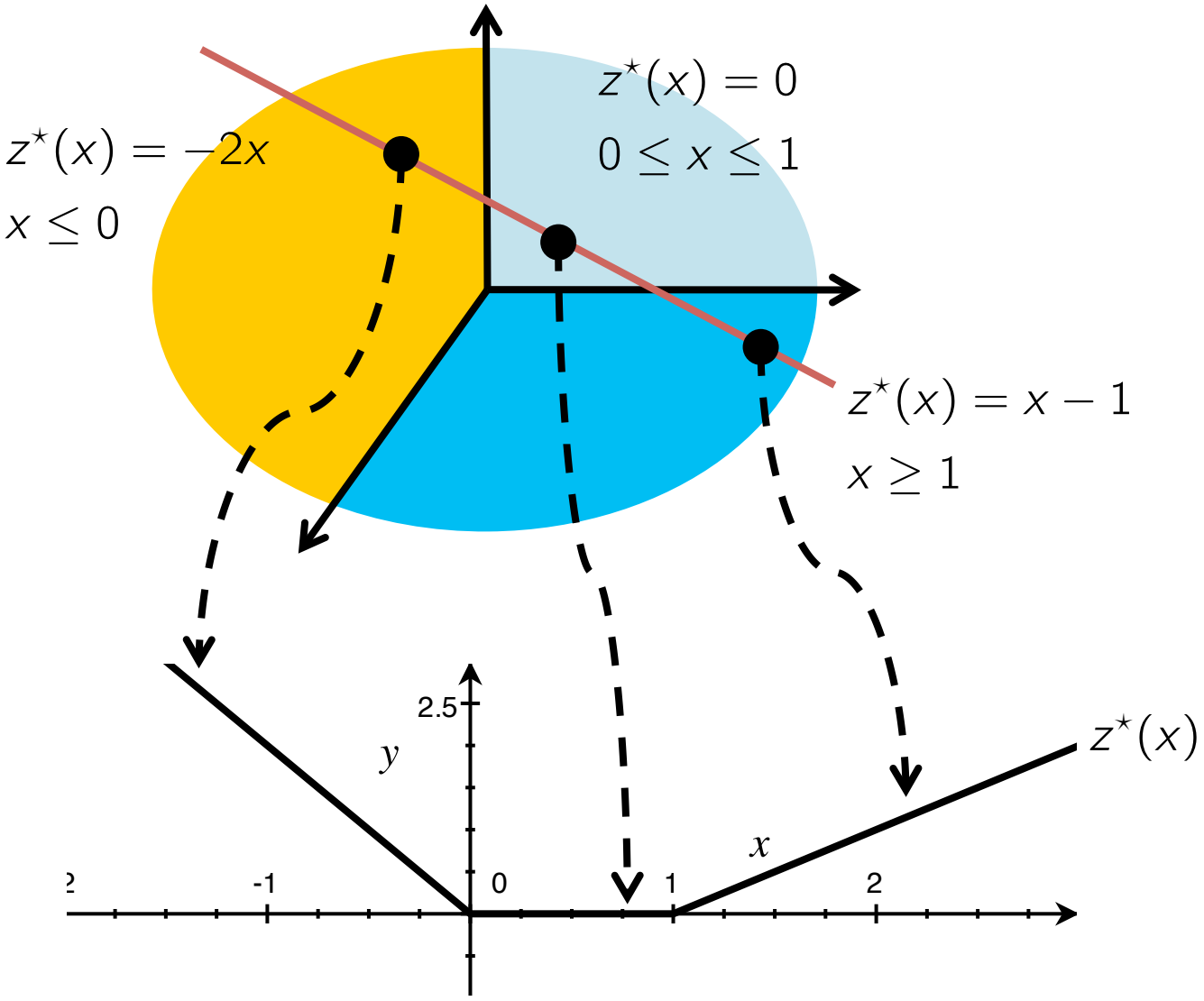
$$f^\star(x) = \min_z \frac{1}{2}z^2 + 2xz$$
$$\text{s.t. } z \geq x - 1$$
$$z \geq 0$$

Parametric Linear Complementarity Problem

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \nu \\ s \end{pmatrix} - \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} z \\ \lambda \end{pmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\begin{pmatrix} \nu \\ s \end{pmatrix}^T \begin{pmatrix} z \\ \lambda \end{pmatrix} = 0 \qquad \nu, s, z, \lambda \geq 0$$



Re-visit Simple Example



Outline

- Motivating Example
- MPC = Parametric Quadratic Programming
- Parametric Linear Complementarity Problems
  - The Geometry
  - The Algebra
  - Efficient Solution Methods
- Online Computation : Point Location Problem
- Examples

Algebra of the LCP

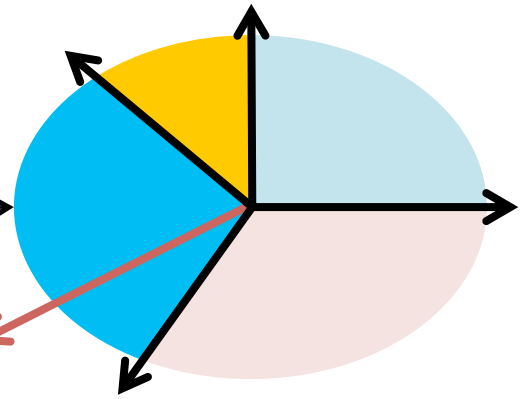
- Define the matrix $A := [I \;\; -M] \in \mathbb{R}^{n \times 2n}$
- The index set $B \subset \{1, \ldots, 2n\}$ is a *basis* if
  - B contains n elements $\quad |B| = n$
  - Columns of A indexed by B are full-rank $\quad \operatorname{rank} A_B = n$
- B is a complementary basis if

$$i \in B \Leftrightarrow i + n \notin B \text{ for all } i \in \{1, \ldots, n\}$$

i.e., the columns satisfy the complementarity conditions $w^T z = 0\,, \;\; w, z \geq 0$

- Complementary bases define complementary cones

$$\mathcal{C}(B) := \{q \in \mathbb{R}^n \mid A_B^{-1} q \geq 0\}$$



Basis B 'solves' the LCP (M, q) if and only if $q \in \mathcal{C}(B)$

Solution Properties

What are $w(x)$ and $z(x)$ in $\mathcal{C}(B)$?

- $w^T z = 0 \quad \rightarrow \quad B$ is a complementary basis
- $A \begin{bmatrix} w \\ z \end{bmatrix} = q + Qx \quad \rightarrow \quad \begin{bmatrix} w \\ z \end{bmatrix}_B = A_B^{-1}(q + Qx)$
- $w, z \geq 0 \quad \rightarrow \quad A_B^{-1}(q + Qx) \geq 0$

Defines a polyhedral ***critical region*** in which B is the solution

$$CR(B) := \{x \mid A_B^{-1}(q + Qx) \geq 0\}$$

Solution is ***affine*** in each complementary cone

$$\begin{bmatrix} w \\ z \end{bmatrix}_B = A_B^{-1}(q + Qx)$$

Control law is piecewise affine defined over a polyhedral partition!

Simple Solution Algorithm

Simple Parametric LCP Solver

- For each complementary basis B
  - If $CR(B)$ is non-empty

$$\begin{bmatrix} w(x) \\ z(x) \end{bmatrix}_B = A_B^{-1}(q + Qx) \quad \text{for all } x \in CR(B)$$

- Works for all pLCPs (including 'non-convex' examples)
  - Testing non-emptiness of a polyhedron is easy (Linear Program)

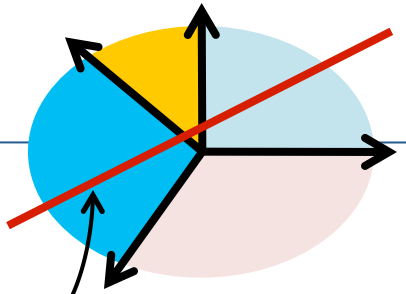
- Consider MPC problem with: 2 states, 2 inputs, horizon of 5 and upper/lower bounds on states and inputs
  - 2^{20} = 1'048'576 possible bases!

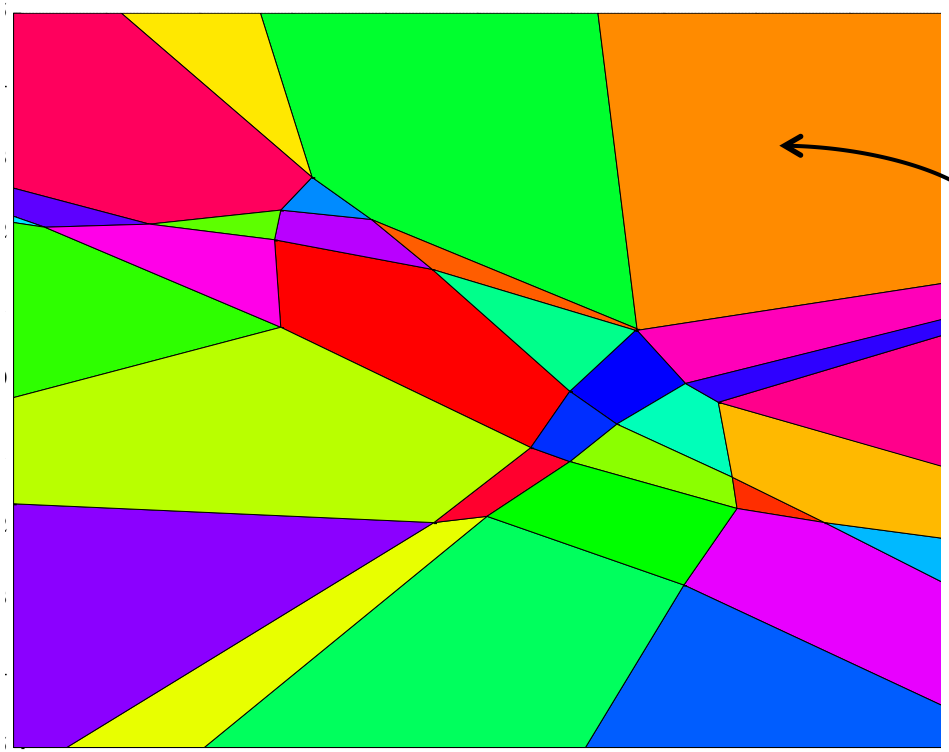
- We need a more efficient algorithm!

Outline

- Motivating Example
- MPC = Parametric Quadratic Programming
- Parametric Linear Complementarity Problems
  - The Geometry
  - The Algebra
  - Efficient Solution Methods
- Online Computation : Point Location Problem
- Examples

Efficient Enumeration Algorithm

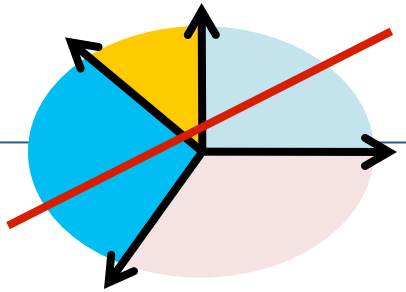


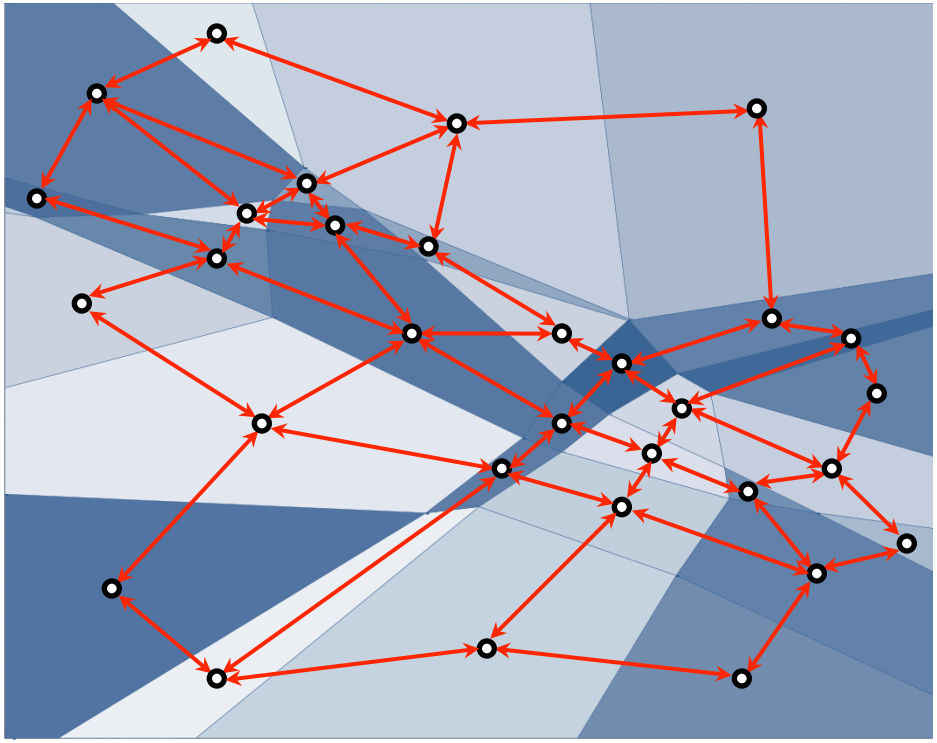


Critical
Regions

P-matrix example.
Figure generated using MPT toolbox

Efficient Enumeration Algorithm





Define graph $\mathcal{G}$

Vertices: Non-empty CRs

Edges: Adjacent CRs

Find all critical regions by standard graph enumeration

When does this work?

- Connected graph
- Non-overlapping critical regions

Well behaving matrix classes

Definition : Sufficient matrix

A matrix $M \in^{n \times n}$ is called *column sufficient* if it satisfies the implication

$$[z_i(Mz)_i \leq 0 \text{ for all } i] \implies [z_i(Mz)_i = 0 \text{ for all } i] \ .$$

The matrix M is called *row sufficient* if its transpose is column sufficient. If M is both column and row sufficient, then it is called *sufficient*.

- Weaker form of semi-definite matrices

Proposition

Positive semi-definite matrices are sufficient.

*Note that this applies to non-symmetric PSD matrices too

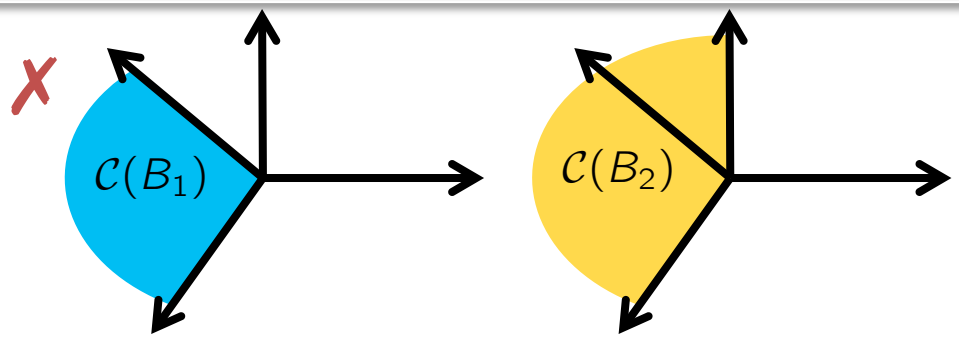
Convex quadratic programs give rise to LCPs with sufficient matrices

LCPs with Sufficient Matrices

Proposition

If M is a sufficient matrix, then the relative interiors of any two distinct complementary cones are disjoint.

Cones cannot overlap:

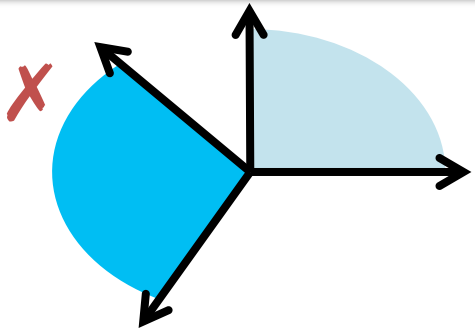


⇒ **Solution is unique!**

Proposition

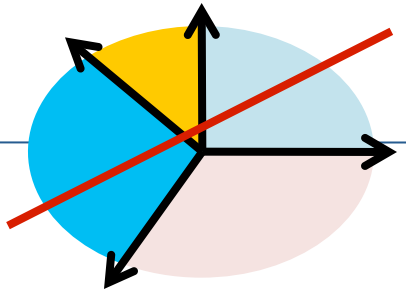
If M is a sufficient matrix, then the union of all complementary cones $K(M)$ is a convex polyhedral cone $K(M) = \text{cone}([I \quad -M])$.

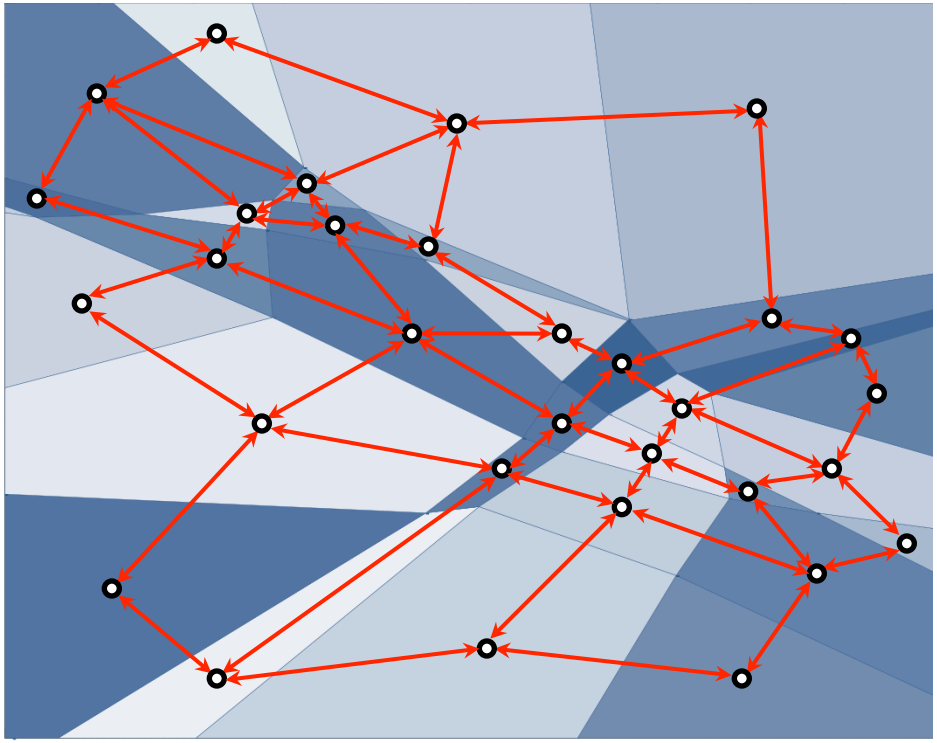
Domain is connected:



⇒ **Neighbour graph is connected!**

Efficient Enumeration Algorithm





Define graph $\mathcal{G}$

Vertices: Non-empty CRs

Edges: Adjacent CRs

Find all critical regions by standard graph enumeration

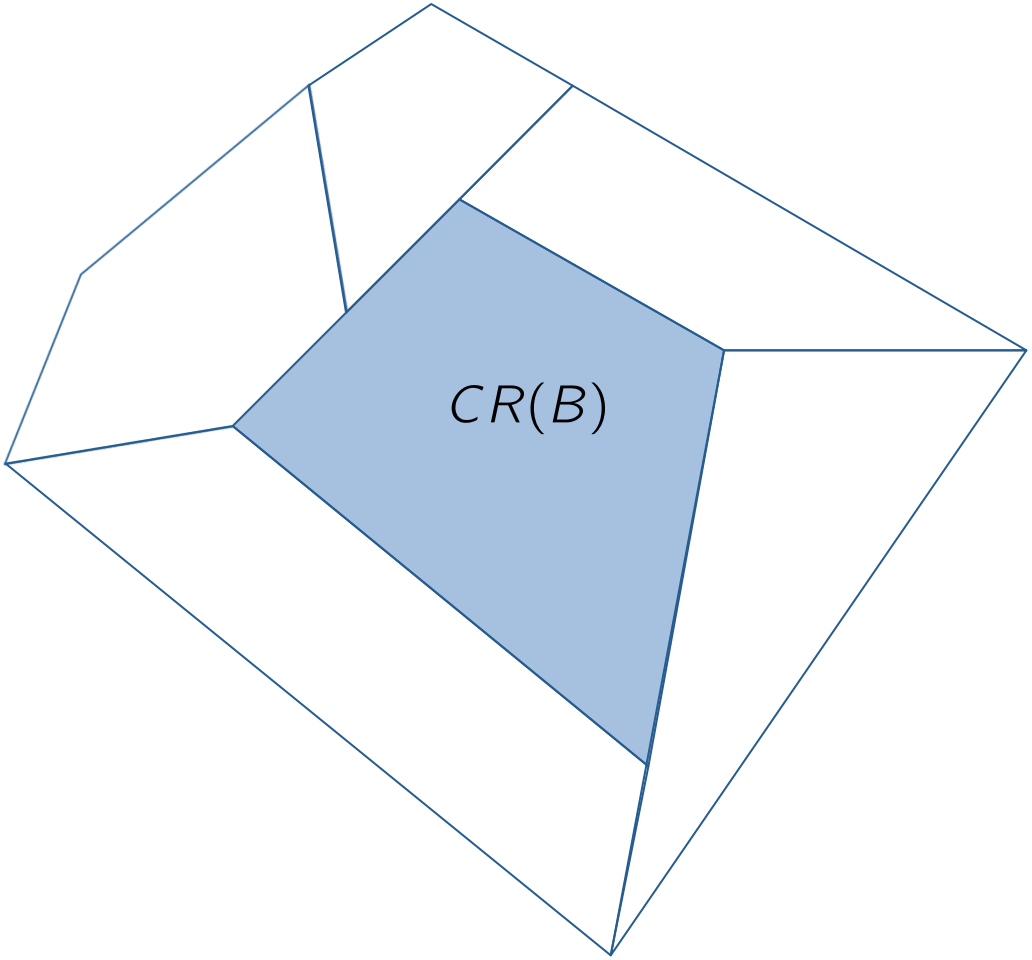
When does this work?

- Connected graph ✔
- Non-overlapping critical regions ✔

If we can compute the neighbours in poly-time, then the enumeration algorithm is polynomial

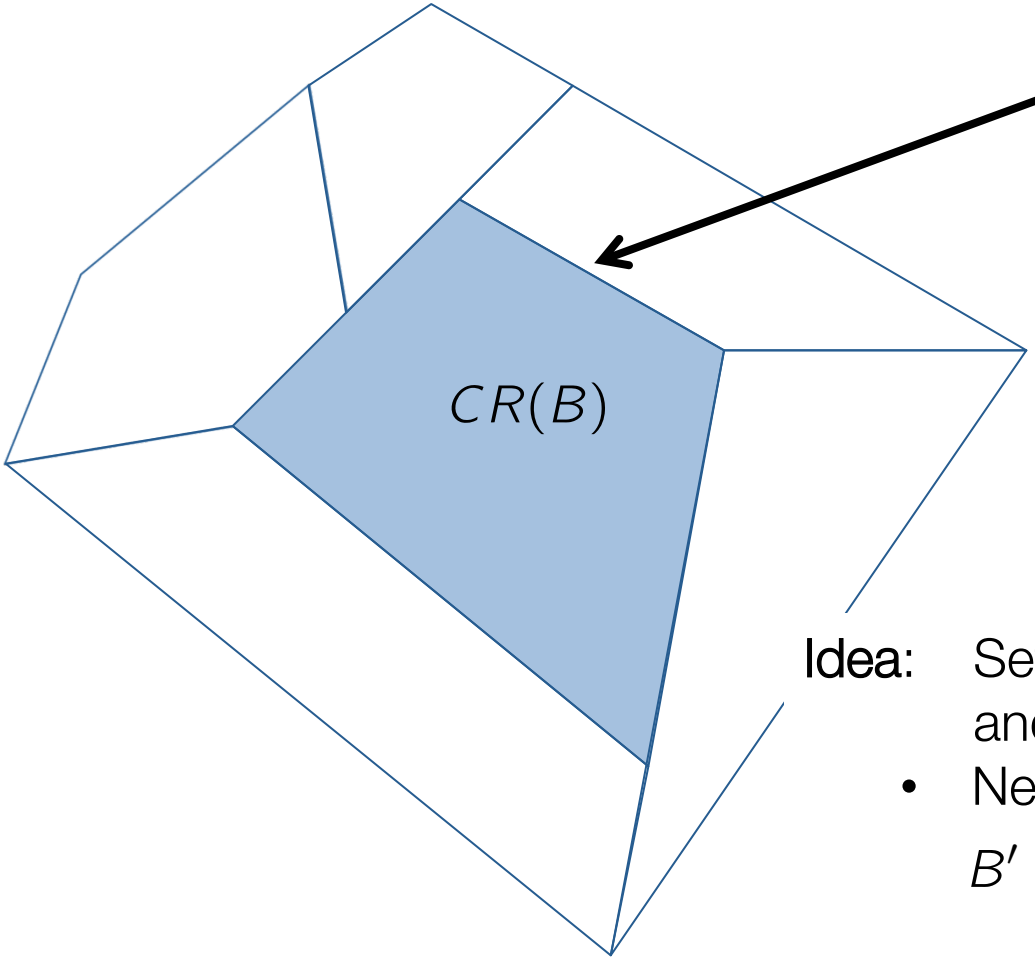
Computing adjacent regions : The idea

Key operation : Find neighbours of $CR(B) := \{x \mid A_B^{-1}(q + Qx) \geq 0\}$



Computing adjacent regions : The idea

Key operation : Find neighbours of $CR(B) := \{x \mid A_B^{-1}(q + Qx) \geq 0\}$



What happens here?

$$\begin{bmatrix} w \\ z \end{bmatrix}_{B_i} = (A_B^{-1}(q + Qx))_i \to 0$$

- One of the positive variables is going to zero
- Any further and the problem would become infeasible

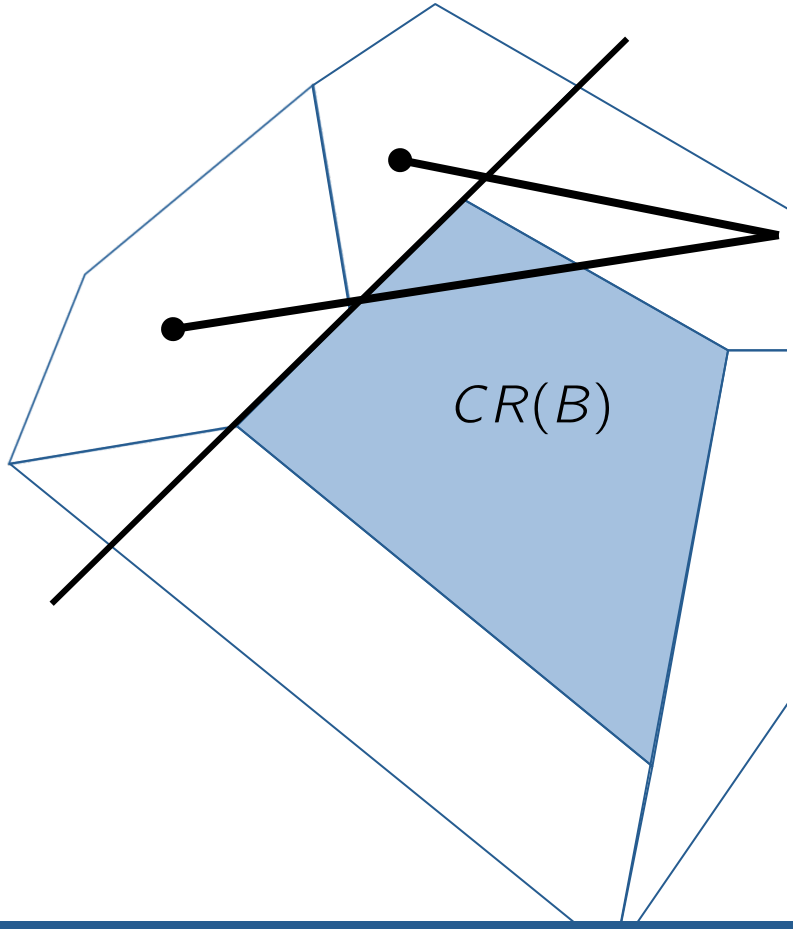
Idea: Set offending variable to zero and allow its complement to increase

- New basis is complementary

$$B' = B \backslash \{i\} \cup \{\bar{i}\}$$

Cost : One linear program to determine if *i* forms a facet

Computing adjacent regions : The complexity



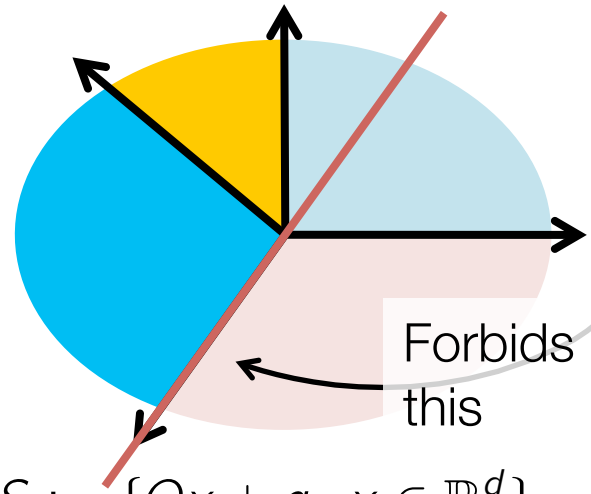
- Multiple regions can neighbour along a single facet
- Requires an *exchange pivot* *(those that differ by two elements)*

$$B \setminus \{i, j\} \cup \{\bar{i}, \bar{j}\}$$

- Cost : n linear programs to determine if i,j forms an adjacent region

Total complexity ≤ (Number of regions) • (number of variables)^2

The fine print : Degeneracy

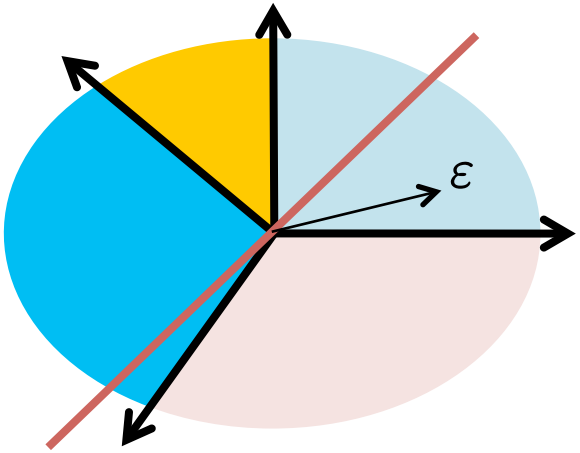


$S := \{Qx + q,\ x \in \mathbb{R}^d\}$

All previous statements rely on S being in *general position*

S intersects $CR(B) \Rightarrow S$ intersects $int(C(B))$

- Multiple solutions for same parameter
- Neighbourhood statements do not hold
  - Enumeration algorithm does not work



$S_\epsilon := \{Qx + q + \bar{\epsilon},\ x \in \mathbb{R}^d\}$

We can ***simulate*** general position artificially

- Adds complexity to the algorithm
- Restores all positive properties

Properties of Model Predictive Control Laws

Model Predictive Control

$$J^\star(x) = \min\ x_N^T Q_f x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i$$

$$\text{s.t. } x_0 = x$$

$$x_{i+1} = A x_i + B u_i$$

$$C x_i + D u_i \leq b$$

Theorem

- Feasible set $X^\star$ is polyhedral (closed and convex)
- The optimizer $u^\star(x) : X^\star$ is a piecewise affine function defined over a polyhedral partition
- The pLCP algorithm selects a continuous optimizer

Also applies for convex polyhedral cost functions

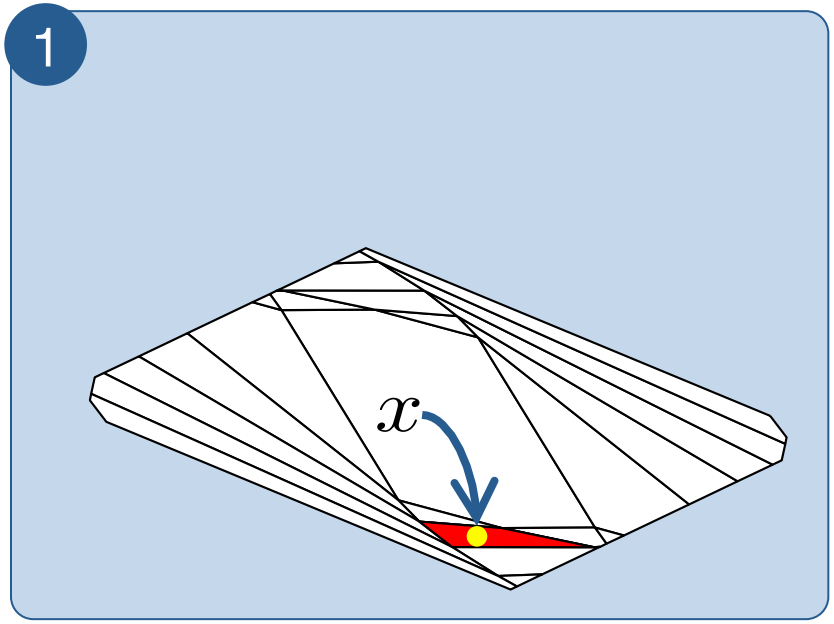
Outline

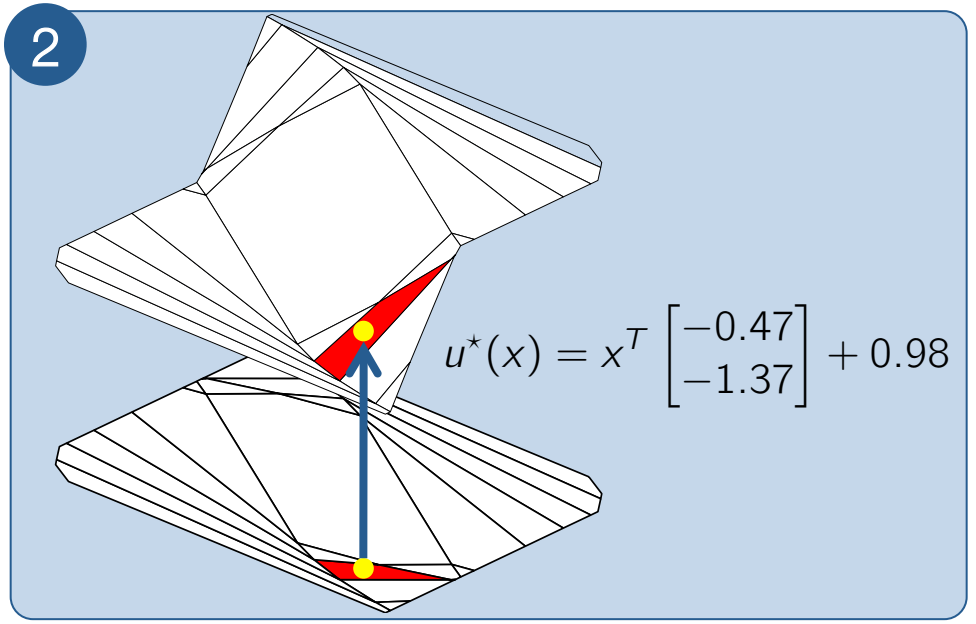
- Motivating Example
- MPC = Parametric Quadratic Programming
- Parametric Linear Complementarity Problems
  - The Geometry
  - The Algebra
  - Efficient Solution Methods
- **Online Computation : Point Location Problem**
- Example

Online evaluation : Point location

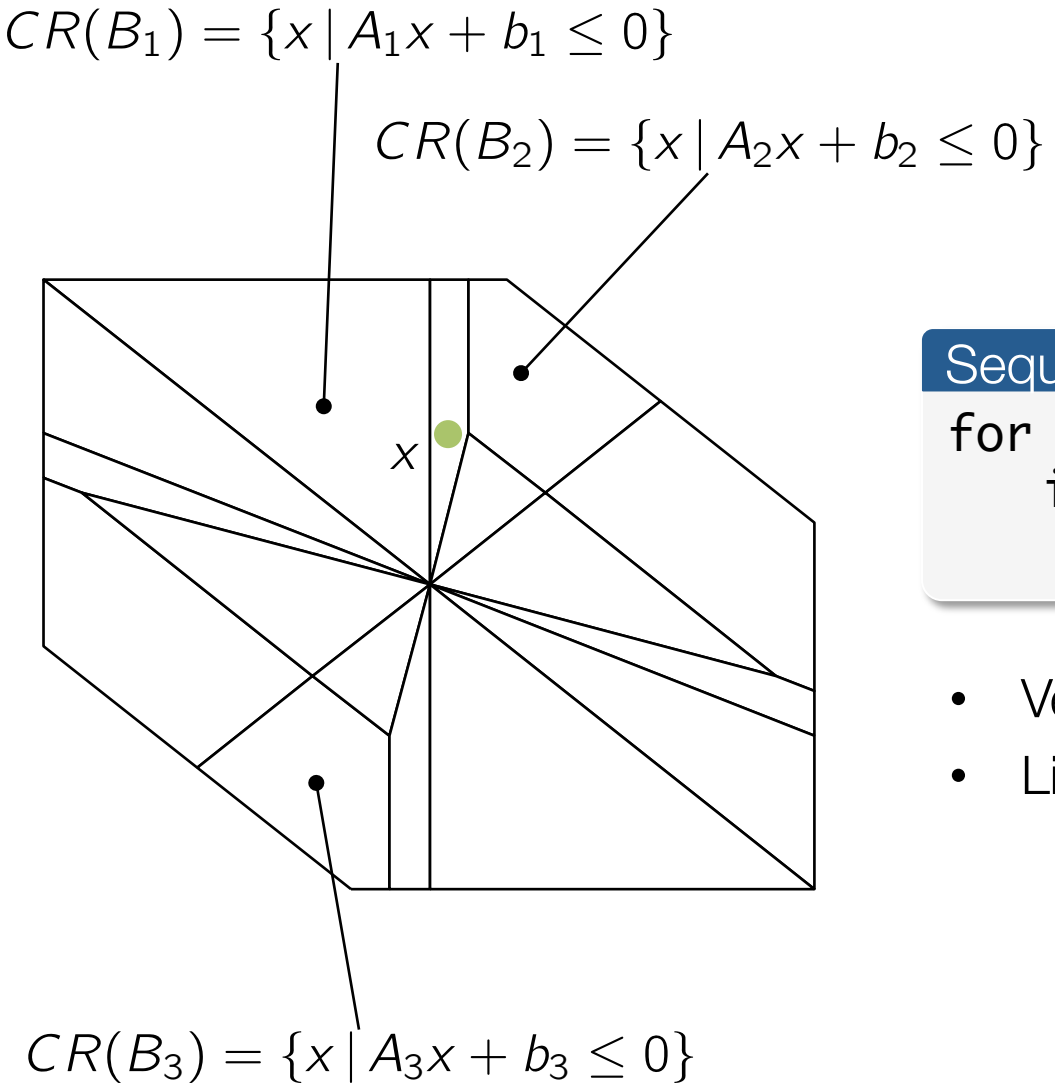
Calculation of piecewise affine function:

1. Point location
2. Evaluation of affine function





Point Location – Sequential search

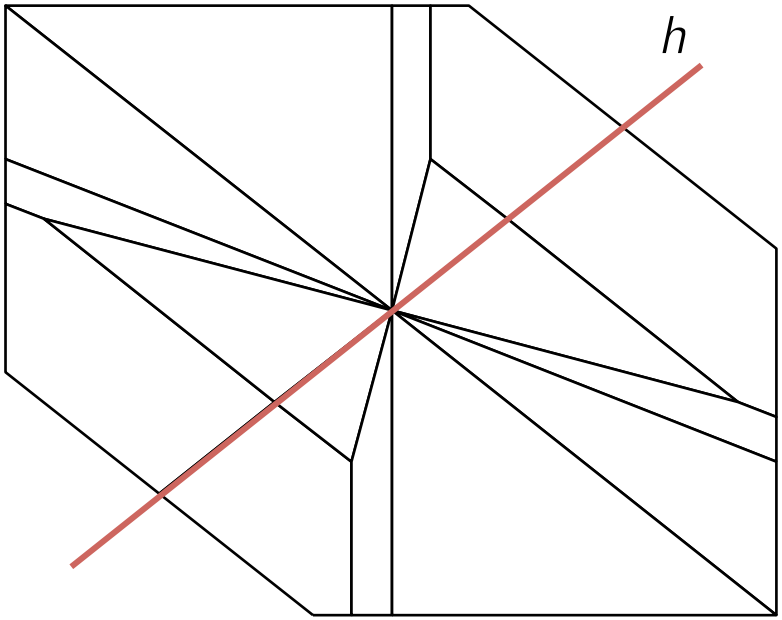


Sequential search

```
for each i
    if A_i x + b_i ≤ 0 then
        x is in region i
```

- Very simple
- Linear in number of regions

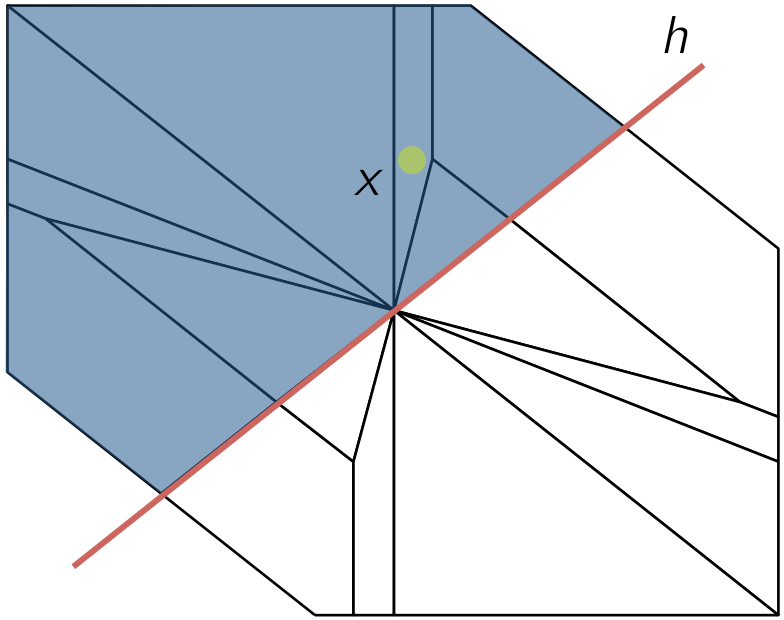
Point Location – Logarithmic search

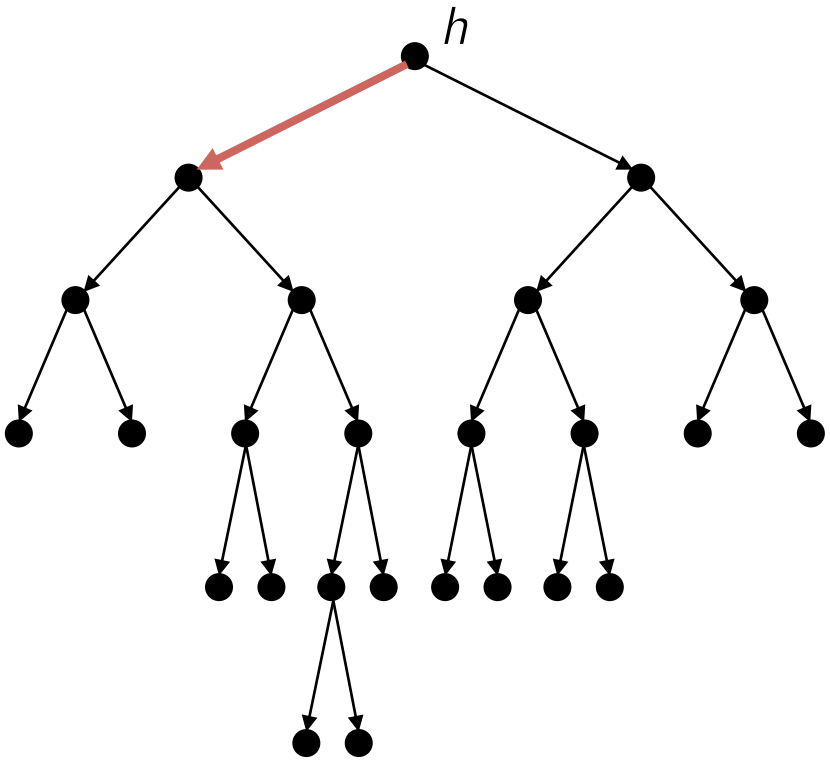


Offline construction of search tree

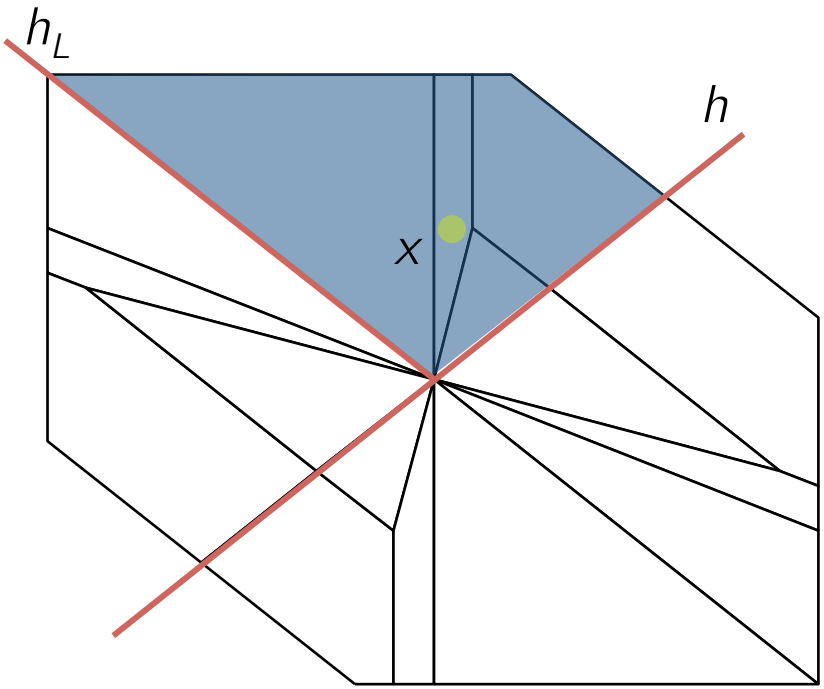
- Find hyperplane that separates regions into two equal sized sets
- Repeat for left and right sets

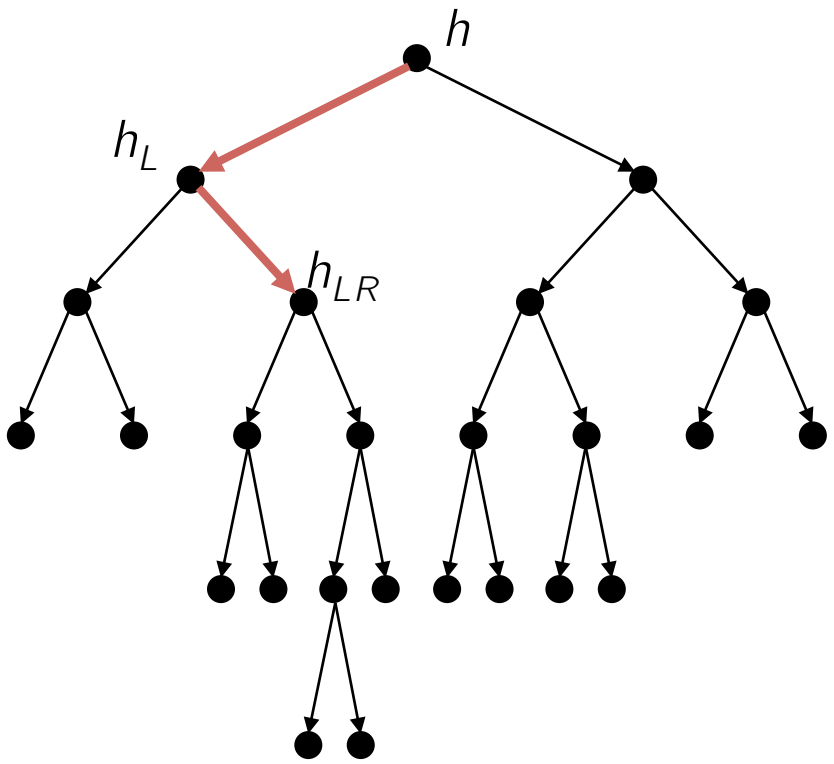
Point Location – Logarithmic search



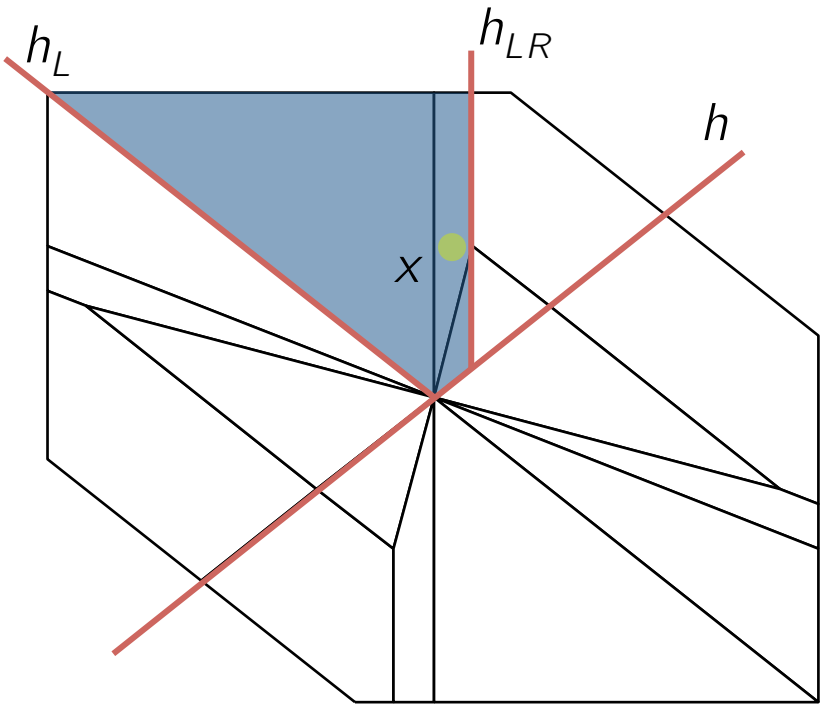


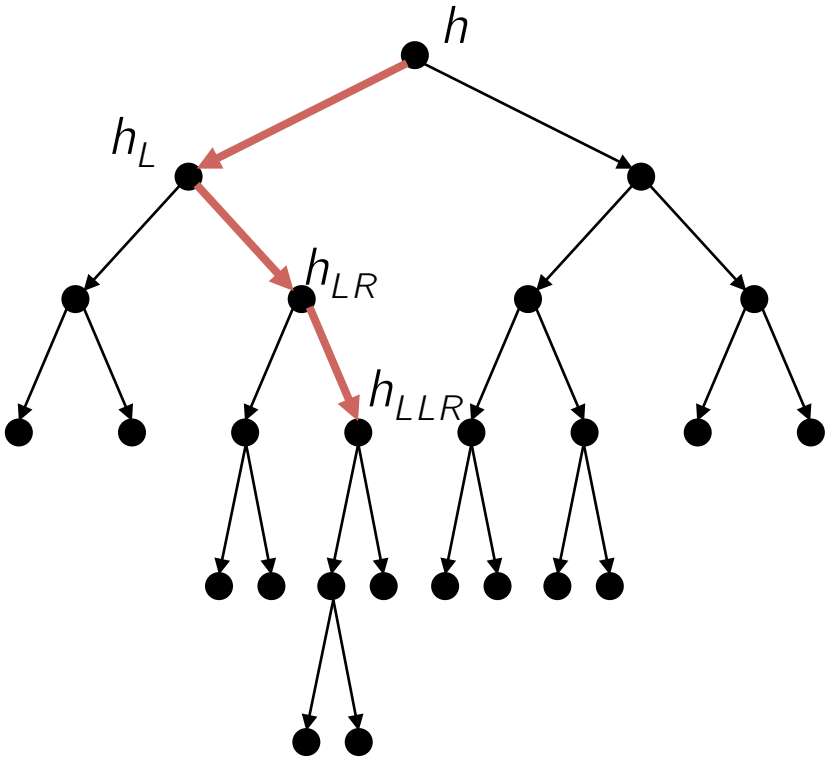
Point Location – Logarithmic search



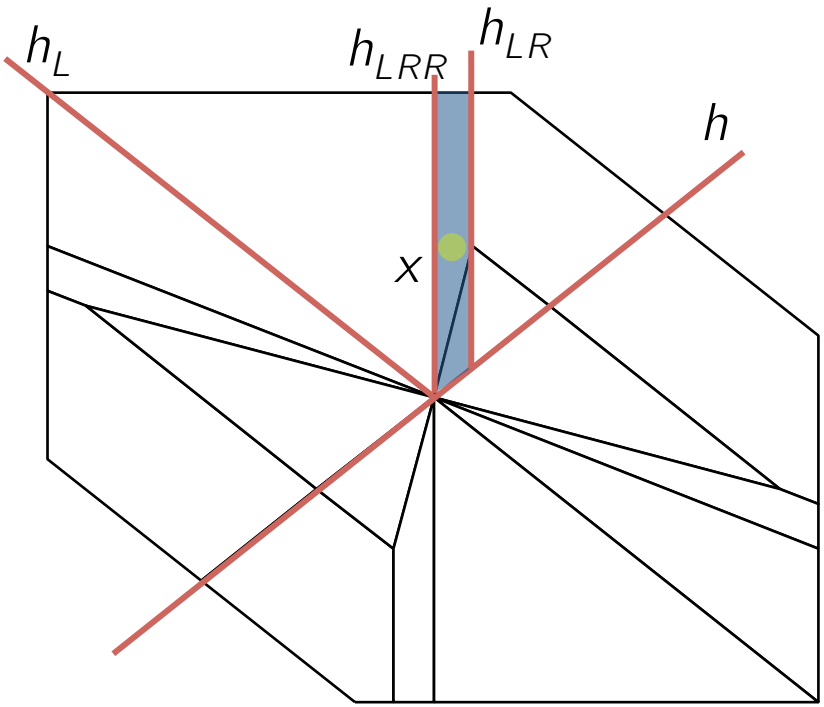


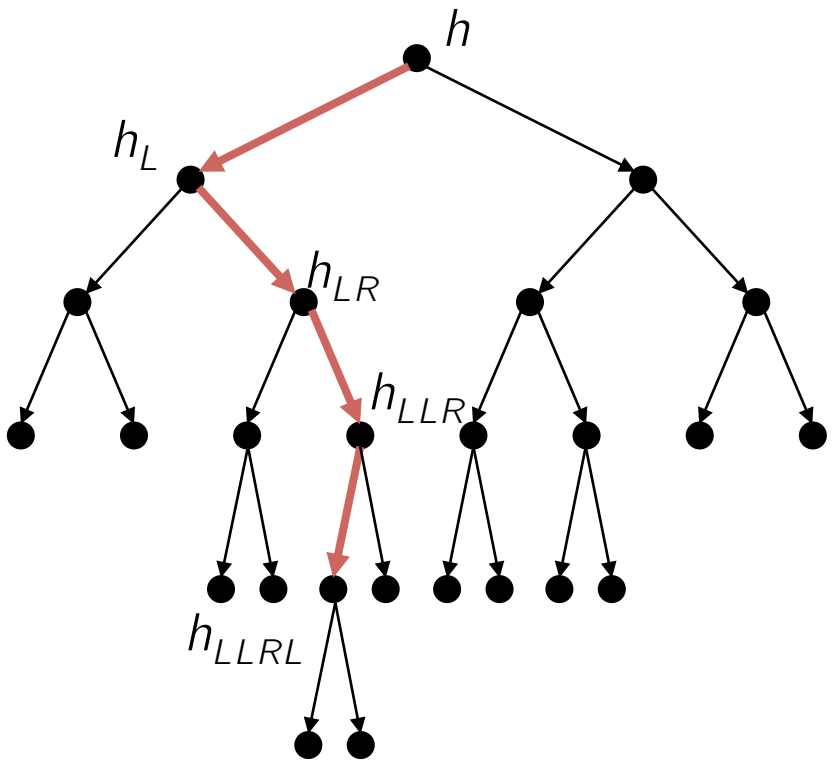
Point Location – Logarithmic search



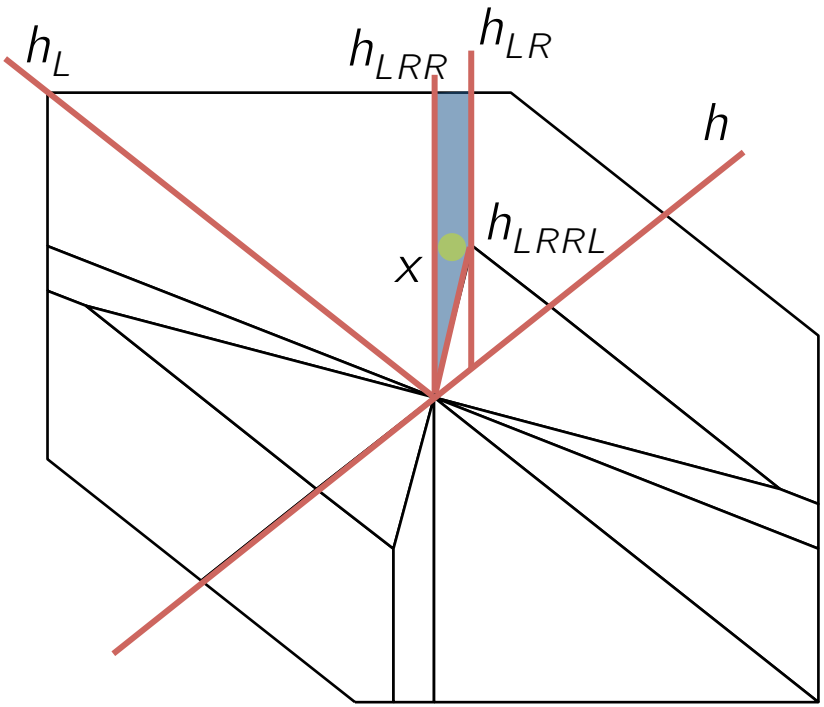


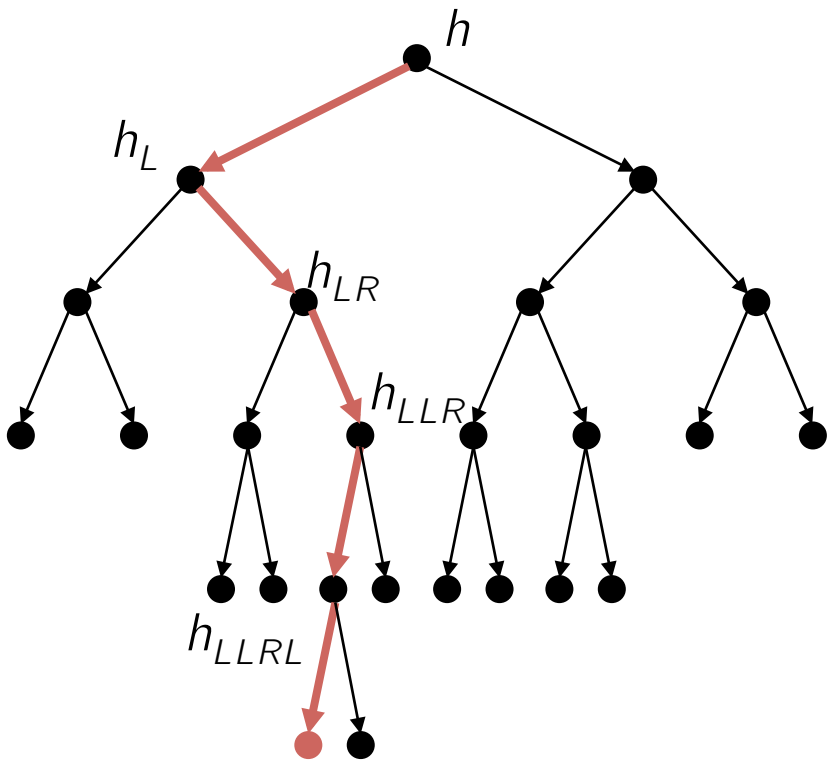
Point Location – Logarithmic search





Point Location – Logarithmic search





Point Location

- Sequential search
  - Very simple
  - Works for all problems

- Search tree
  - Potentially logarithmic
  - Significant offline processing (reasonable for < 1’000 regions)

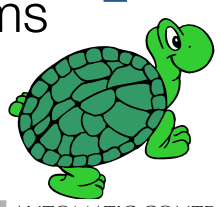
- Many other options for special cases

Outline

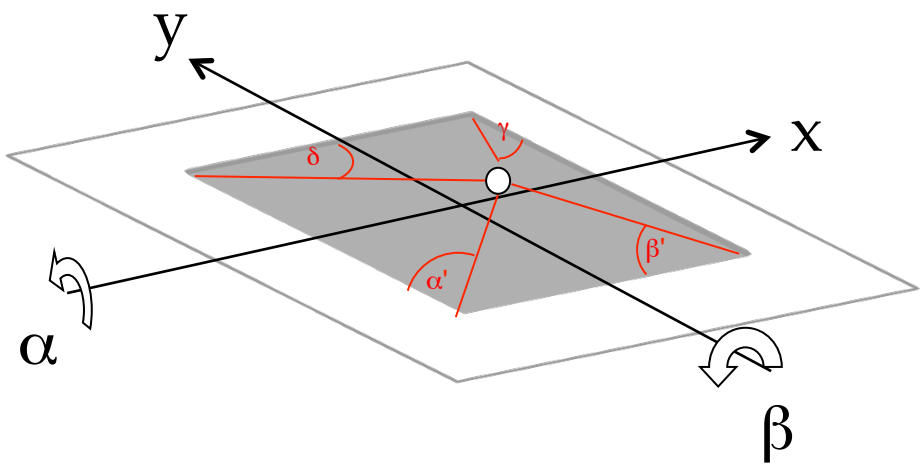
- Motivating Example
- MPC = Parametric Quadratic Programming
- Parametric Linear Complementarity Problems
  - The Geometry
  - The Algebra
  - Efficient Solution Methods
- Online Computation : Point Location Problem
- Example

Applications by the Automatic Control Lab

18 ns	Multi-core thermal management (EPFL)	*[Zanini et al 2010]*
10 µs	Voltage source inverters	*[Mariethoz et al 2008]*
20 µs	DC/DC converters (STM)	*[Mariethoz et al 2008]*
25 µs	Direct torque control (ABB)	*[Papafotiou 2007]*
50 µs	AC / DC converters	*[Richter et al 2010]*
5 ms	Electronic throttle control (Ford)	*[Vasak et al 2006]*
20 ms	Traction control (Ford)	*[Borrelli et al 2001]*
30 ms	**Ball and plate**	
50 ms	Autonomous vehicle steering (Ford)	*[Besselmann et al 2008]*
500 ms	Energy efficient building control (Siemens)	*[Oldewurtel et al 2010]*



Ball and Plate



- Linearized model : four states for each axis
- Constraints on inputs and states
  - Plate angle
  - Ball position
  - Acceleration
- MPC objective : Path tracking

Ball and Plate System

Matlab
+ RTW

Computer:	Pentium 166
Sampling Time:	30 ms

TCP/IP

Code

Data

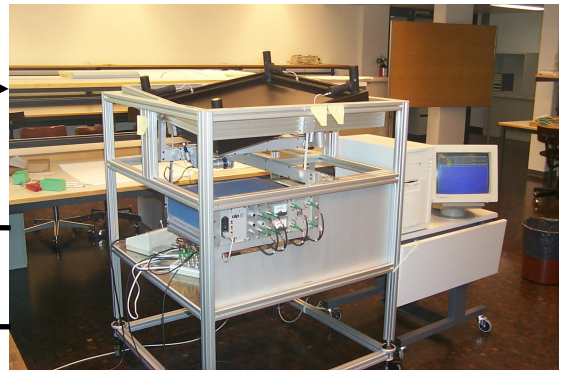
Ball & Plate

xPC
Target

u_1, u_2

α, β

x, y

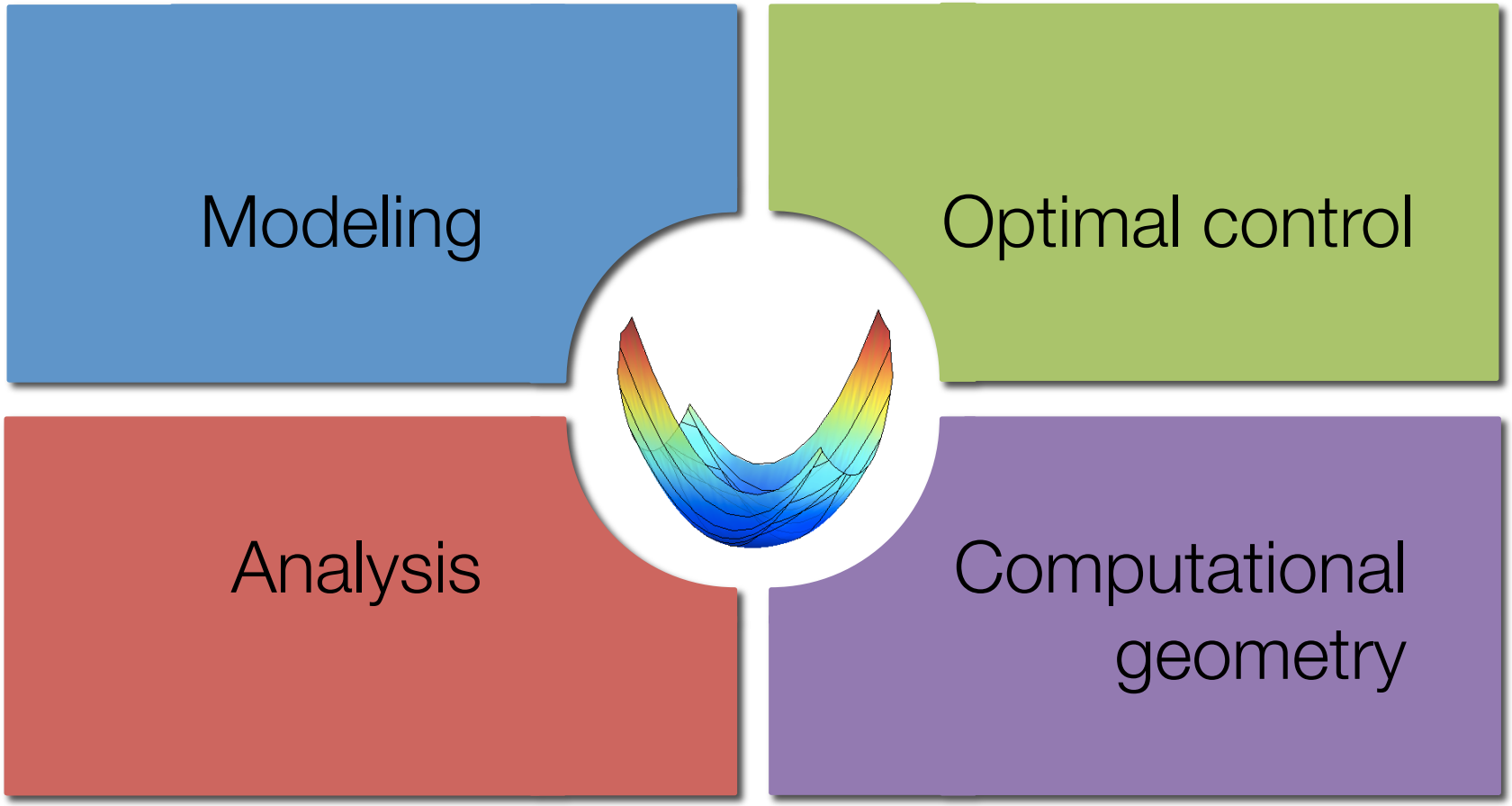


MPC Problem

- 4 states + 1 tracking variable = 5 parameters
- Move-blocking reduces complexity
  - Horizon of 10
  - Inputs 2-10 must be equal

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t.} \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

Multi-Parametric Toolbox



MPT Toolbox: M. Kvasnica, P. Grieder and M. Baotic

Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools
Michal Kvasnica
ISBN: 3639206444

Multi-Parametric Toolbox Formulation

MPT

```
% linear discrete-time prediction model
model.A = A;  model.B = B;
model.C = C;  model.D = D;

% Input constraints
model.umax = 10;  model.umin = -10;
% Output constraints
model.ymax = 30;  model.ymin = -30;
% State constraints
model.xmax = [30; 15; 15*pi/180; 1];
model.xmin = -model.xmax;
```

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t.} \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

Multi-Parametric Toolbox Formulation

MPT

```
% linear discrete-time prediction model
model.A = A;  model.B = B;
model.C = C;  model.D = D;

% Input constraints
model.umax = 10;  model.umin = -10;
% Output constraints
model.ymax = 30;  model.ymin = -30;
% State constraints
model.xmax = [30; 15; 15*pi/180; 1];
model.xmin = -model.xmax;
```

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t.} \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

Multi-Parametric Toolbox Formulation

MPT

```
% linear discrete-time prediction model
model.A = A;  model.B = B;
model.C = C;  model.D = D;

% Input constraints
model.umax = 10;  model.umin = -10;
% Output constraints
model.ymax = 30;  model.ymin = -30;
% State constraints
model.xmax = [30; 15; 15*pi/180; 1];
model.xmin = -model.xmax;
```

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t.} \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

Multi-Parametric Toolbox Formulation

MPT

```
% linear discrete-time prediction model
model.A = A;  model.B = B;
model.C = C;  model.D = D;

% Input constraints
model.umax = 10;  model.umin = -10;
% Output constraints
model.ymax = 30;  model.ymin = -30;
% State constraints
model.xmax = [30; 15; 15*pi/180; 1];
model.xmin = -model.xmax;
```

MPT

```
% Prediction horizon
problem.N = 10;
% Control horizon
problem.Nc = 1;
% Quadratic performance objective
problem.norm = 2;
% Penalty on outputs
problem.Qy = 100;
% Penalty on control inputs
problem.R = 0.1;
% Tracking
problem.tracking = 2;
```

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t.} \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

Multi-Parametric Toolbox Formulation

MPT

```
% linear discrete-time prediction model
model.A = A;  model.B = B;
model.C = C;  model.D = D;

% Input constraints
model.umax = 10;  model.umin = -10;
% Output constraints
model.ymax = 30;  model.ymin = -30;
% State constraints
model.xmax = [30; 15; 15*pi/180; 1];
model.xmin = -model.xmax;
```

MPT

```
% Prediction horizon
problem.N = 10;
% Control horizon
problem.Nc = 1;
% Quadratic performance objective
problem.norm = 2;
% Penalty on outputs
problem.Qy = 100;
% Penalty on control inputs
problem.R = 0.1;
% Tracking
problem.tracking = 2;
```

$$\min \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2$$

$$\text{s.t. } x_0 = x$$

$$x_{i+1} = Ax_i + Bu_i$$

$$y_i = Cx_i$$

$$u_{\min} \leq u_i \leq u_{\max}$$

$$y_{\min} \leq y_i \leq y_{\max}$$

$$x_{\min} \leq x_i \leq x_{\max}$$

$$u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}$$

Multi-Parametric Toolbox Formulation

MPT

```
% linear discrete-time prediction model
model.A = A;  model.B = B;
model.C = C;  model.D = D;

% Input constraints
model.umax = 10;  model.umin = -10;
% Output constraints
model.ymax = 30;  model.ymin = -30;
% State constraints
model.xmax = [30; 15; 15*pi/180; 1];
model.xmin = -model.xmax;
```

MPT

```
% Prediction horizon
problem.N = 10;
% Control horizon
problem.Nc = 1;
% Quadratic performance objective
problem.norm = 2;
% Penalty on outputs
problem.Qy = 100;
% Penalty on control inputs
problem.R = 0.1;
% Tracking
problem.tracking = 2;
```

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t.} \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

Multi-Parametric Toolbox Formulation

MPT

```
% linear discrete-time prediction model
model.A = A;  model.B = B;
model.C = C;  model.D = D;

% Input constraints
model.umax = 10;  model.umin = -10;
% Output constraints
model.ymax = 30;  model.ymin = -30;
% State constraints
model.xmax = [30; 15; 15*pi/180; 1];
model.xmin = -model.xmax;
```

MPT

```
% Prediction horizon
problem.N = 10;
% Control horizon
problem.Nc = 1;
% Quadratic performance objective
problem.norm = 2;
% Penalty on outputs
problem.Qy = 100;
% Penalty on control inputs
problem.R = 0.1;
% Tracking
problem.tracking = 2;
```

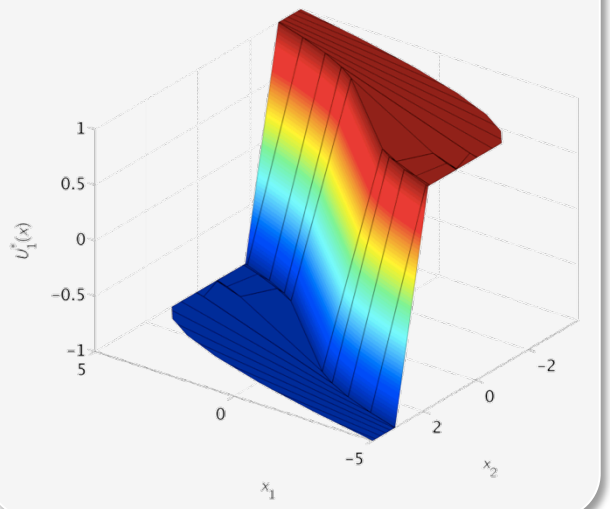
$$
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t.} \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

Explicit Solution

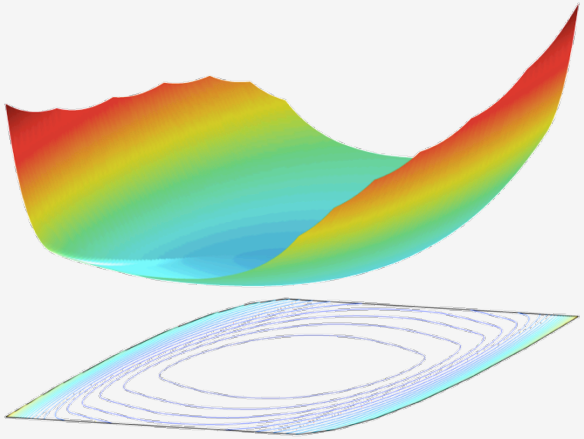
MPT

```
mpt_controller = mpt_control(model, problem, 'explicit');

% Plot controller partition
plot(mpt_controller)

% Plot control law
plotU(mpt_controller)

% Plot optimal value function
plotJ(mpt_controller)
```

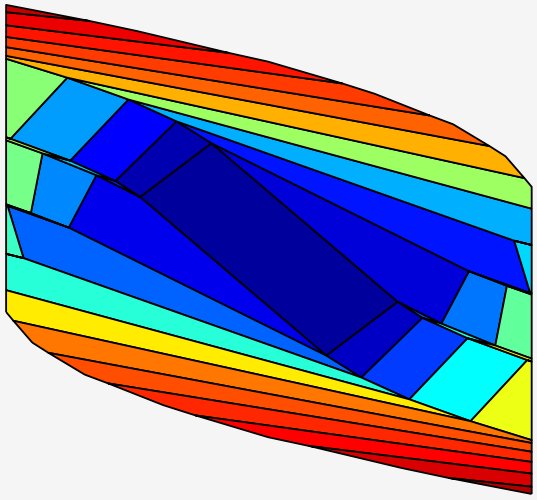
Control input



Value function



Controller partition



Exporting Explicit Solutions

Search tree and export to flat c-file

MPT

```
mpt_searchTree(mpt_controller)
mpt_exportST(mpt_controller)
```

```
#define MPT_NU 1
#define MPT_NX 2
static float MPT_ST[] = {
-5.225216e-01,  8.526260e-01,   2.007907e+00,   2.000000e+00,
-6.669647e-01,  7.450893e-01,   3.576296e+00,   4.000000e+00,
6.347758e-01,   -7.726964e-01,  2.699058e+00,   6.000000e+00,
-2.990218e-01,  -9.542463e-01,  -2.286671e-01,  8.000000e+00,
-4.546596e-01,  -8.906653e-01,  3.576036e-01,   1.000000e+01,
3.630512e-01,   9.317692e-01,   4.490581e-02,   1.200000e+01,
4.546596e-01,   8.906653e-01,   3.576036e-01,   1.400000e+01,
-2.528581e-01,  -9.675034e-01,  -4.781395e-01,  -5.000000e+00,
-6.811331e-01,  7.321596e-01,   4.632449e+00,   1.600000e+01,
-3.630512e-01,  -9.317692e-01,  4.490581e-02,   -5.000000e+00,
-6.347758e-01,  7.726964e-01,   2.699058e+00,   1.800000e+01,
6.811331e-01,   -7.321596e-01,  4.632449e+00,   2.000000e+01,
6.669647e-01,   -7.450893e-01,  3.576296e+00,   2.200000e+01,
5.845669e-01,   8.113455e-01,   7.438927e-01,   -4.000000e+00,
5.225216e-01,   -8.526260e-01,  2.007907e+00,   2.400000e+01,
2.528581e-01,   9.675034e-01,   1.703078e+00,   -4.000000e+00,
2.990218e-01,   9.542463e-01,   1.489549e+00,   -4.000000e+00,
3.630512e-01,   9.317692e-01,   1.263985e+00,   -4.000000e+00,
4.546596e-01,   8.906653e-01,   1.018374e+00,   -4.000000e+00,
2.528581e-01,   9.675034e-01,   -4.781395e-01,  -4.000000e+00,
2.990218e-01,   9.542463e-01,   -2.286671e-01,  -4.000000e+00,
```

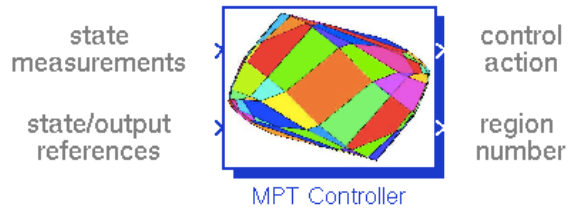
Export to flat c-file, sequential search

MPT

```
mpt_exportc(mpt_controller)
```

```
#ifdef tmwtypes_h
  /* RTW is used, switch to real_T data types to avoid problems with TLC compi
lation */
  static long mpt_searchTree(const real_T *X, real_T *U)
#else
  static long mpt_searchTree(const float *X, float *U)
#endif
{
    int ix, iu;
    long node = 1, row;
    float hx, k;

    /* initialize U to zero*/
    for (iu=0; iu<MPT_NU; iu++) {
        U[iu] = 0;
    }

    /* find region which contains the state x0 */
    while (node > 0) {
        hx = 0;
        row = (node-1)*(MPT_NX+3);
        for (ix=0; ix<MPT_NX; ix++) {
            hx = hx + MPT_ST[row+ix]*X[ix];
        }
        k = MPT_ST[row+MPT_NX];
```

RTW



- Explicit MPC block in Simulink
- Compatible with RTW

Ball and Plate Explicit Controller

- 4 states + 1 tracking variable = 5 parameters
- Move-blocking reduces complexity
  - Horizon of 10
  - Inputs 2-10 must be equal

$$
\begin{aligned}
J^\star(x, y_t) = \min \; & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t. } & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

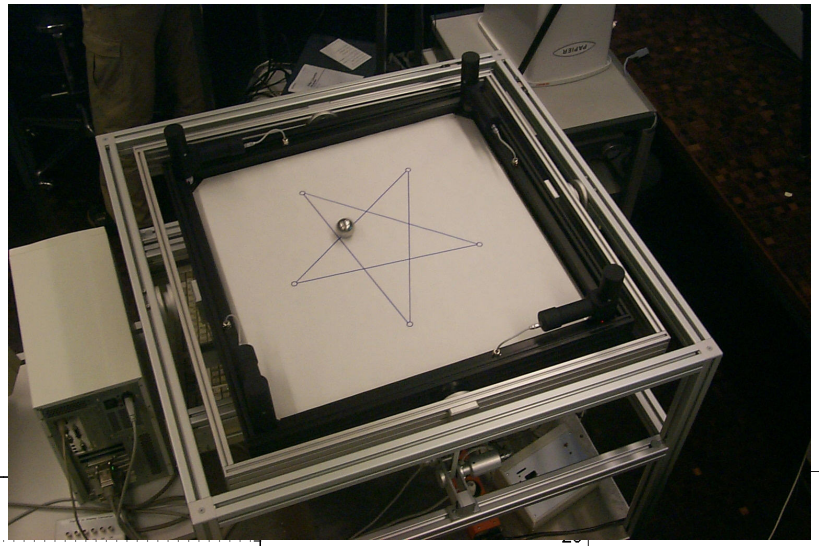
Explicit solution (per dimension)

Regions : 529

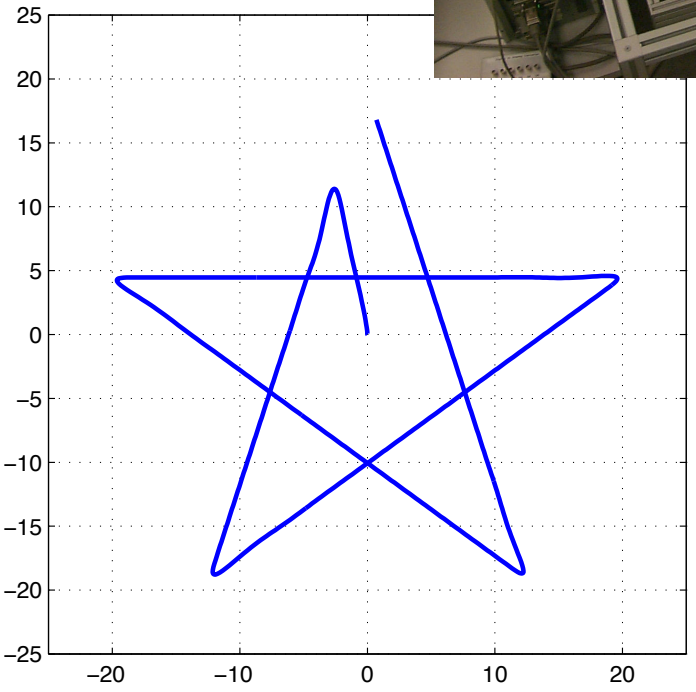
Storage : 48'000 numbers (192 kB)

Computation : 89'000 FLOPS (~1ms)

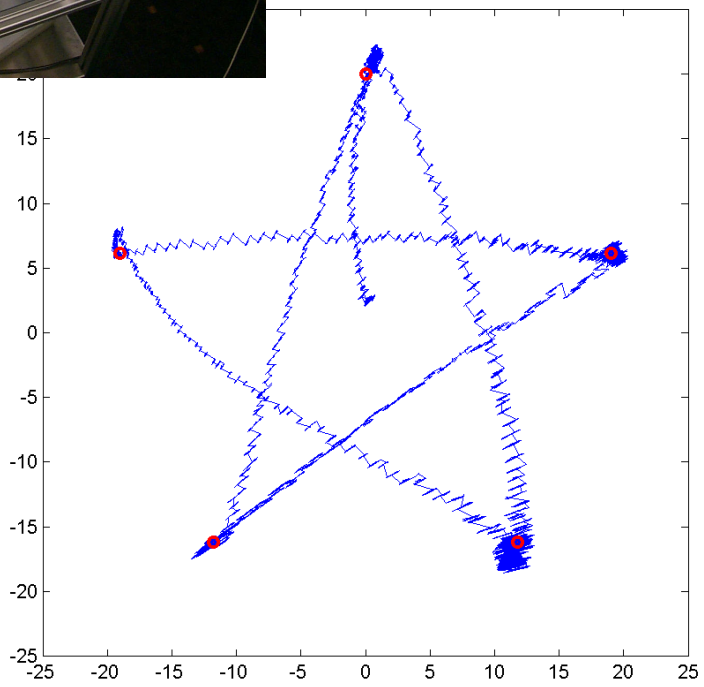
Ball and Plate : Pentagram



Simulation



Experiment



Real-time MPC Software Toolbox

Formal specification
- YALMIP
- HYSDEL
- Linear + Hybrid models

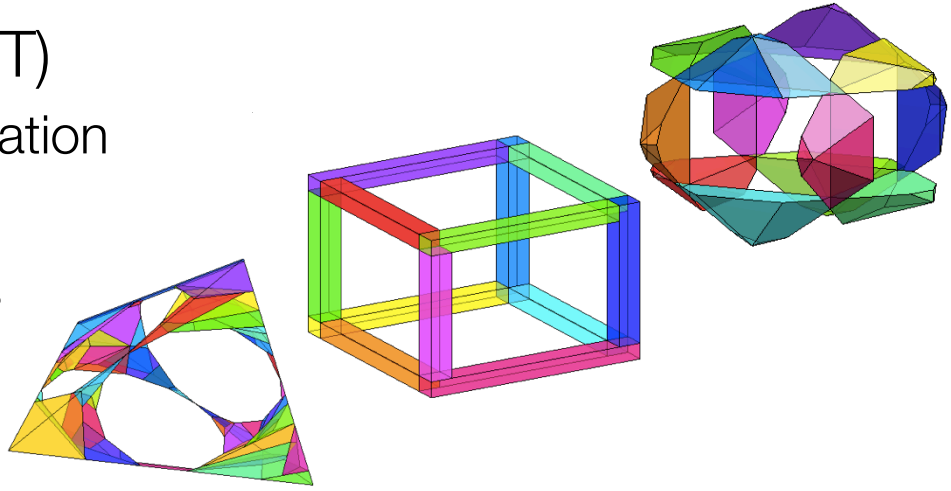
Control law
- Explicit MPC
- Fixed-complexity solutions

Software synthesis
- Real-time workshop
- Bounded-time solvers
- Verifiable code generation

Verified controller

Multi-Parametric Toolbox (MPT)

- (Non)-Convex Polytopic Manipulation
- Multi-Parametric Programming
- Control of PWA and LTI systems
- > 22,000 downloads to date



MPT 3.0 coming in 2011

Workshop Outline

14:00 – 14:30	Introduction	
14:30 – 15:15	Lecture 1:	MPC Theory
15:15 – 15:20	Break	
15:20 – 15:40	Lecture 2 :	Explicit MPC
15:40 – 16:00	Break	
16:00 – 16:40	Lecture 2 :	Explicit MPC
16:40 – 17:10	Lecture 3 :	Real-time explicit MPC
17:10 – 17:20	Break	
17:20 – 18:00	Lecture 4 :	Real-time online optimization