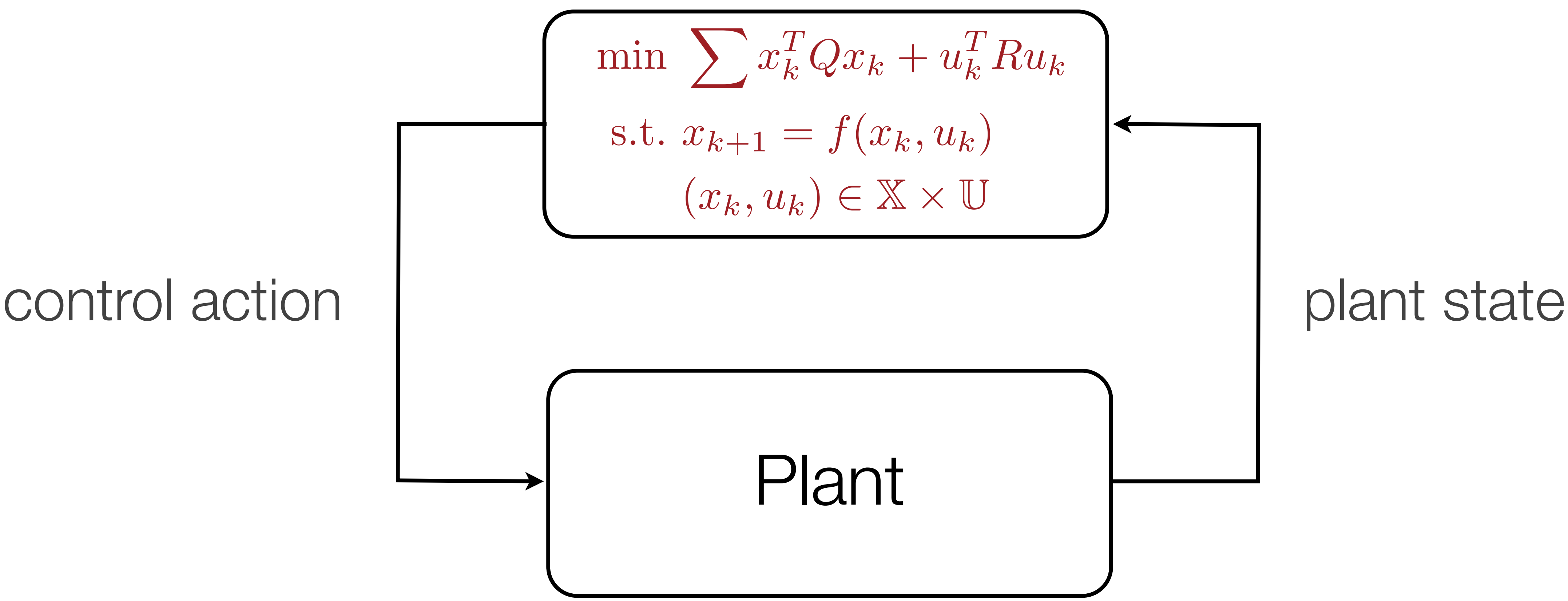
Implicit vs Explicit MPC: Similarities, Differences, and a Path towards a Unified Method

Martin Klaučo and Michal Kvasnica

Introduction

Off-line

On-line

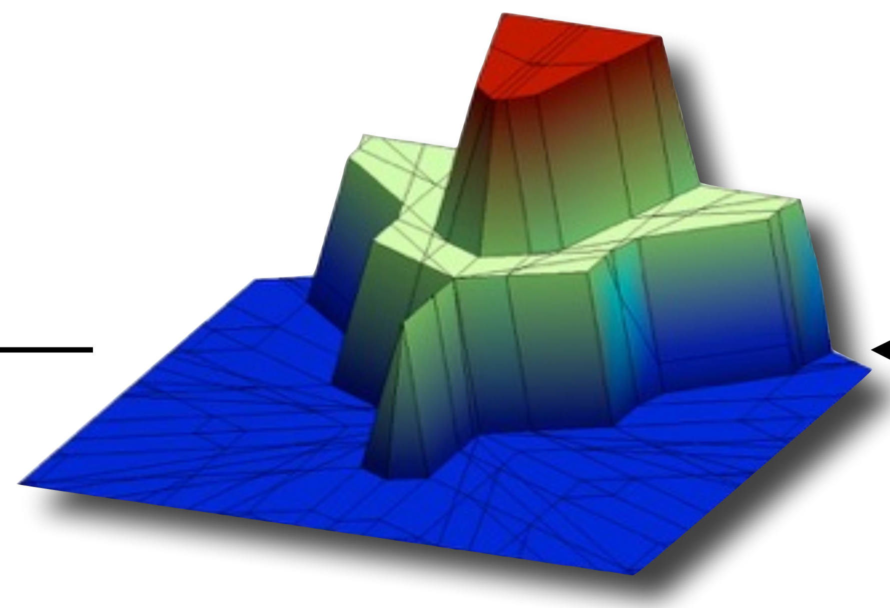


Introduction

$$\min \sum x_k^T Q x_k + u_k^T R u_k$$
$$\text{s.t. } x_{k+1} = f(x_k, u_k)$$
$$(x_k, u_k) \in \mathbb{X} \times \mathbb{U}$$

Off-line

On-line



control action

plant state

Plant

Introduction

MOTD1: explicit MPC is an active set method with a complete factorization of all KKT systems

F. Borrelli et al.: On the computation of linear MPC laws; Automatica, 2010

Introduction

MOTD1: explicit MPC is an active set method with a complete factorization of all KKT systems

F. Borrelli et al.: On the computation of linear MPC laws; Automatica, 2010

MOTD2: explicit MPC can be used even for systems 100+ states

A. Gupta et al.: A novel approach to multiparametric quadratic programming; Automatica, 2011

M. Kvasnica et al.: On region-free explicit model predictive control; CDC 2015

Introduction

MOTD1: explicit MPC is an active set method with a complete factorization of all KKT systems

F. Borrelli et al.: On the computation of linear MPC laws; Automatica, 2010

MOTD2: explicit MPC can be used even for systems 100+ states

A. Gupta et al.: A novel approach to multiparametric quadratic programming; Automatica, 2011

M. Kvasnica et al.: On region-free explicit model predictive control; CDC 2015

MOTD3: explicit MPC can benefit from strategies known in active set methods

Implicit MPC

$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Active set method (feasible-start Newton):

1. choose an initial active set *A* and a feasible solution *U*
2. solve for *Δ* from *min 1/2*$(U+\Delta)^T H(U+\Delta)+\theta^T F(U+\Delta)$ *s.t.* $G_A(U+\Delta)=w_A+E_A\theta$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} \Delta \\ \lambda \end{bmatrix} = \begin{bmatrix} -HU - F^T\theta \\ 0 \end{bmatrix}$$

Implicit MPC

$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Active set method (feasible-start Newton):

1. choose an initial active set *A* and a feasible solution *U*
2. solve for Δ from $min\ 1/2(U+\Delta)^T H(U+\Delta)+\theta^T F(U+\Delta)$ s.t. $G_A(U+\Delta)=w_A+E_A\theta$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} \Delta \\ \lambda \end{bmatrix} = \begin{bmatrix} -HU - F^T\theta \\ 0 \end{bmatrix}$$

3. heuristic update of the active set:
   - if some $\lambda_i<0$, drop a constraint from *A*
   - if some inactive constraint is violated by $U+\Delta$, add a constraint to *A*

Implicit MPC

$$\min_U \ {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \le w + E\theta$$

Active set method (feasible-start Newton):

1. choose an initial active set *A* and a feasible solution *U*
2. solve for *Δ* from *min 1/2*$(U+\Delta)^T H(U+\Delta)+\theta^T F(U+\Delta)$ *s.t.* $G_A(U+\Delta)=w_A+E_A\theta$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} \Delta \\ \lambda \end{bmatrix} = \begin{bmatrix} -HU - F^T\theta \\ 0 \end{bmatrix}$$

3. heuristic update of the active set:
   - if some $\lambda_i<0$, drop a constraint from *A*
   - if some inactive constraint is violated by $U+\Delta$, add a constraint to *A*
4. determine the step length *α*, update $U := U+\alpha\Delta$
5. repeat until convergence

Implicit MPC

$$\min_U \; {}^1\!/\!{}_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Active set method (feasible-start Newton):

1. choose an initial active set A and a feasible solution U
2. solve for Δ from $min\ 1/2(U+\Delta)^T H(U+\Delta)+\theta^T F(U+\Delta)$ $s.t.\ G_A(U+\Delta)=w_A+E_A\theta$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} \Delta \\ \lambda \end{bmatrix} = \begin{bmatrix} -HU - F^T\theta \\ 0 \end{bmatrix}$$

expensive

3. heuristic update of the active set:
   - if some $\lambda_i<0$, drop a constraint from A
   - if some inactive constraint is violated by $U+\Delta$, add a constraint to A
4. determine the step length α, update $U := U+\alpha\Delta$
5. repeat until convergence

Implicit MPC

$$\min_U \ {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \le w + E\theta$$

Active set method (feasible-start Newton):

1. choose an initial active set *A* and a feasible solution *U*
2. solve for *Δ* from *min* $1/2(U+\Delta)^T H(U+\Delta)+\theta^T F(U+\Delta)$ *s.t.* $G_A(U+\Delta)=w_A+E_A\theta$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} \Delta \\ \lambda \end{bmatrix} = \begin{bmatrix} -HU - F^T\theta \\ 0 \end{bmatrix}$$

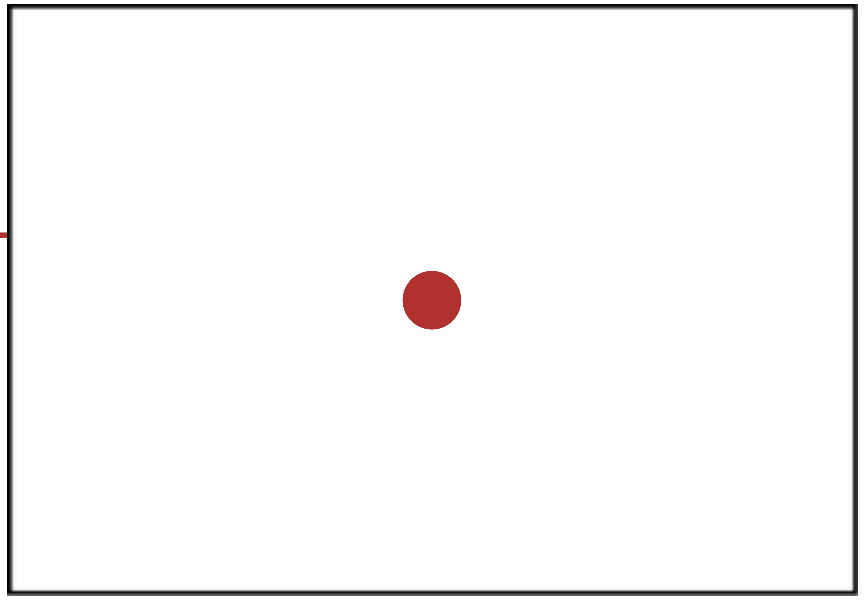
expensive

3. heuristic update of the active set:

works remarkably well

   - if some $\lambda_i<0$, drop a constraint from *A*
   - if some inactive constraint is violated by *U+Δ*, add a constraint to *A*
4. determine the step length *α*, update *U := U+αΔ*
5. repeat until convergence

Explicit MPC



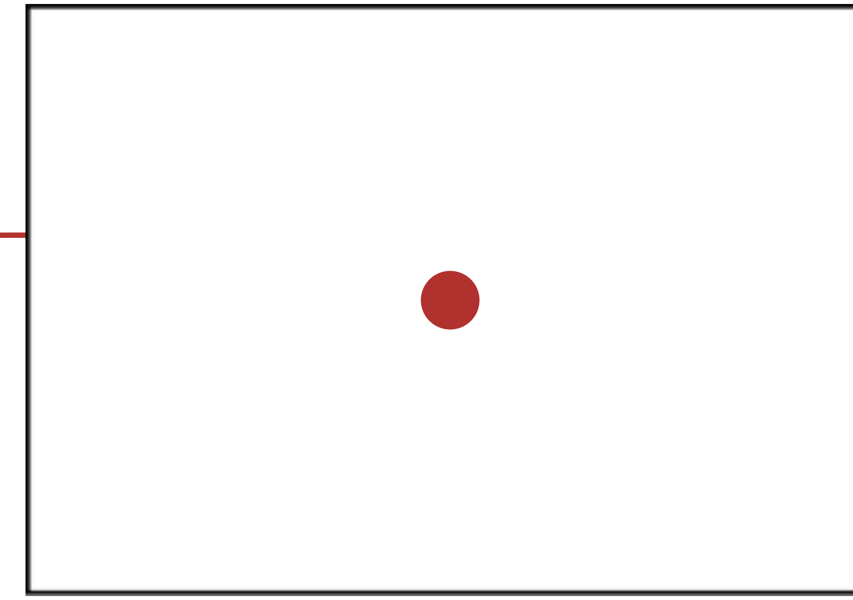
$$\min_{U} \ {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Geometric parametric programming:

1. choose a feasible parameter θ
2. determine the active set A from *min 1/2* $U^T HU+\theta^T FU$ *s.t.* $GU \leq w+E\theta$

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

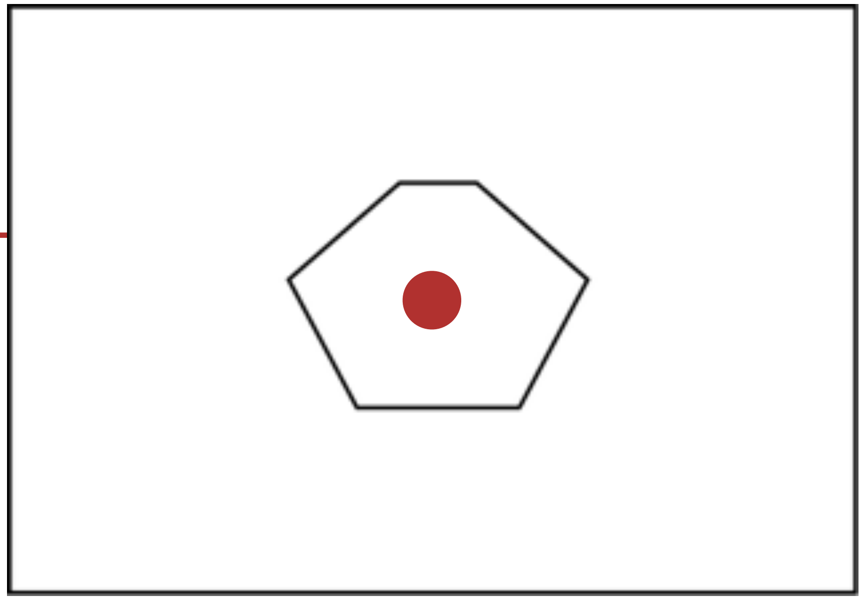
Geometric parametric programming:

1. choose a feasible parameter θ
2. determine the active set A from $\min\ 1/2\ U^T H U + \theta^T F U$ s.t. $GU \leq w + E\theta$
3. factorize the KKT system, obtain $U_A(\theta) = S_A\theta + s_A$, $\lambda_A(\theta) = Q_A\theta + q_A$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -F^T\theta \\ w_A + E_A\theta \end{bmatrix} \Rightarrow \begin{array}{l} U_A(\theta) = S_A\theta + s_A \\ \lambda_A(\theta) = Q_A\theta + q_A \end{array}$$

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_{U} \ {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Geometric parametric programming:

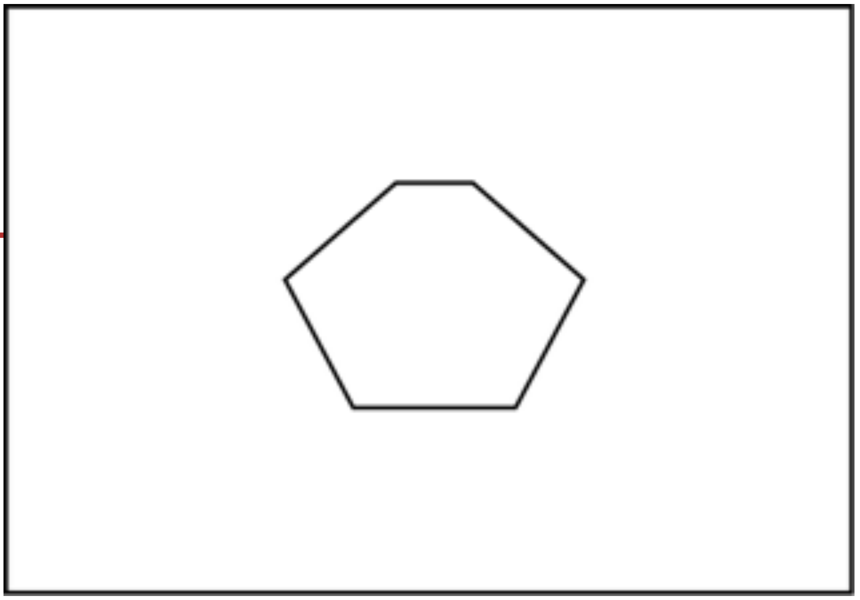
1. choose a feasible parameter θ
2. determine the active set A from *min 1/2* $U^T HU+\theta^T FU$ *s.t.* $GU \leq w+E\theta$
3. factorize the KKT system, obtain $U_A(\theta)=S_A\theta+s_A$, $\lambda_A(\theta)=Q_A\theta+q_A$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -F^T\theta \\ w_A + E_A\theta \end{bmatrix} \Rightarrow \begin{matrix} U_A(\theta) = S_A\theta + s_A \\ \lambda_A(\theta) = Q_A\theta + q_A \end{matrix}$$

4. determine the critical region $R = \{ \theta \mid GU_A(\theta) \leq w+E\theta, \lambda_A(\theta) \geq 0 \} = \{ \theta \mid H_A\theta \leq h_A \}$

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_{U} \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Geometric parametric programming:

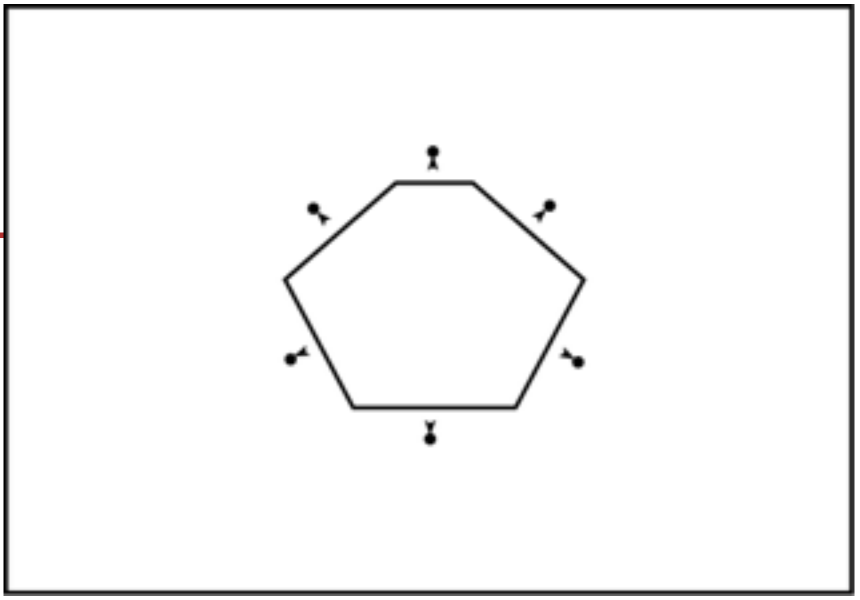
1. choose a feasible parameter θ
2. determine the active set A from *min 1/2* $U^T H U + \theta^T F U$ *s.t.* $GU \leq w + E\theta$
3. factorize the KKT system, obtain $U_A(\theta) = S_A\theta + s_A$, $\lambda_A(\theta) = Q_A\theta + q_A$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -F^T\theta \\ w_A + E_A\theta \end{bmatrix} \Rightarrow \begin{matrix} U_A(\theta) = S_A\theta + s_A \\ \lambda_A(\theta) = Q_A\theta + q_A \end{matrix}$$

4. determine the critical region $R = \{ \theta \mid GU_A(\theta) \leq w + E\theta, \lambda_A(\theta) \geq 0 \} = \{ \theta \mid H_A\theta \leq h_A \}$
5. explore the neighbors of R until convergence

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_U \; 1/2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Geometric parametric programming:

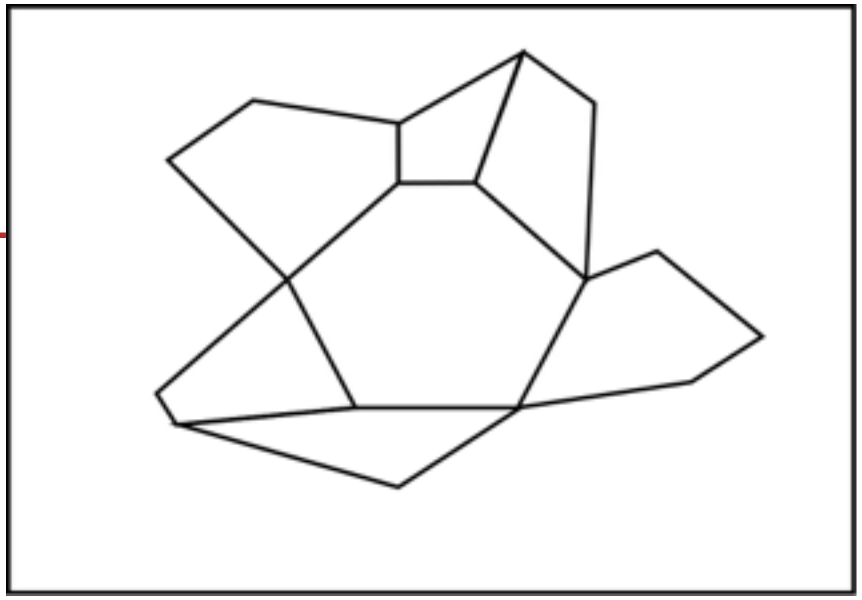
1. choose a feasible parameter θ
2. determine the active set A from *min 1/2* $U^T HU+\theta^T FU$ *s.t.* $GU \leq w+E\theta$
3. factorize the KKT system, obtain $U_A(\theta)=S_A\theta+s_A$, $\lambda_A(\theta)=Q_A\theta+q_A$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -F^T\theta \\ w_A + E_A\theta \end{bmatrix} \Rightarrow \begin{array}{c} U_A(\theta) = S_A\theta + s_A \\ \lambda_A(\theta) = Q_A\theta + q_A \end{array}$$

4. determine the critical region $R = \{ \theta \mid GU_A(\theta) \leq w+E\theta, \lambda_A(\theta) \geq 0 \} = \{ \theta \mid H_A\theta \leq h_A \}$
5. explore the neighbors of R until convergence

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_U \; {}^1\!/\!{}_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \le w + E\theta$$

Geometric parametric programming:

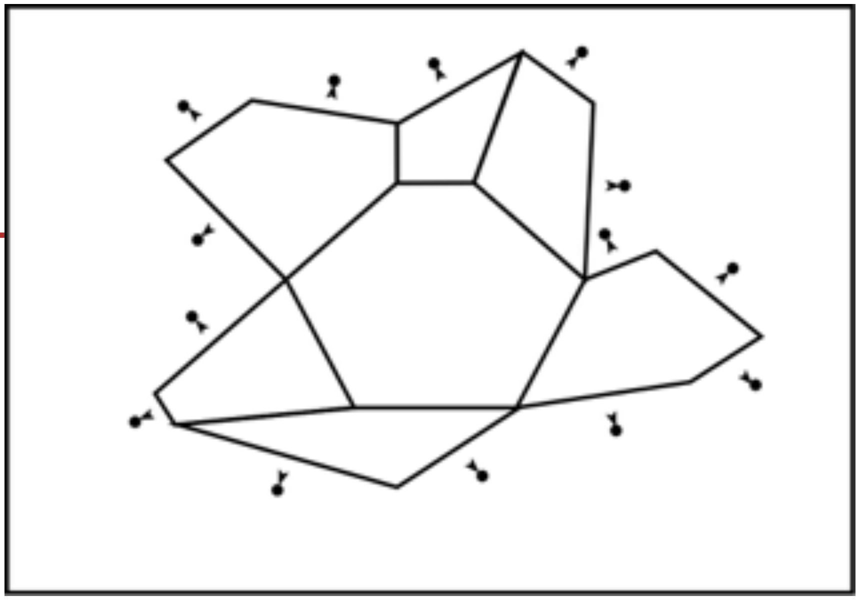
1. choose a feasible parameter θ
2. determine the active set A from *min 1/2* $U^T HU+\theta^T FU$ *s.t.* $GU \le w+E\theta$
3. factorize the KKT system, obtain $U_A(\theta)=S_A\theta+s_A$, $\lambda_A(\theta)=Q_A\theta+q_A$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -F^T\theta \\ w_A + E_A\theta \end{bmatrix} \Rightarrow \begin{matrix} U_A(\theta) = S_A\theta + s_A \\ \lambda_A(\theta) = Q_A\theta + q_A \end{matrix}$$

4. determine the critical region $R = \{ \theta \mid GU_A(\theta) \le w+E\theta, \lambda_A(\theta) \ge 0 \} = \{ \theta \mid H_A\theta \le h_A \}$
5. explore the neighbors of R until convergence

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \le w + E\theta$$

Geometric parametric programming:

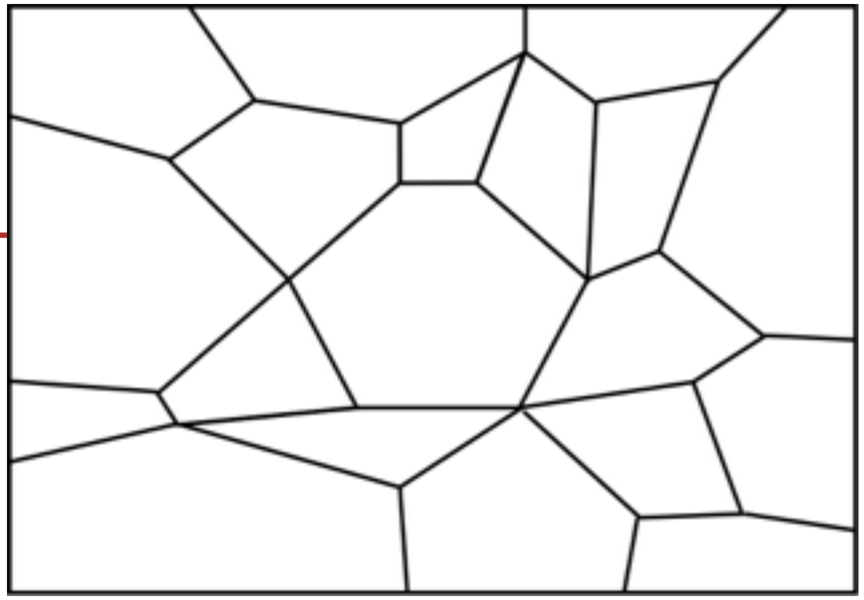
1. choose a feasible parameter θ
2. determine the active set A from *min 1/2* $U^T HU+\theta^T FU$ *s.t.* $GU \le w+E\theta$
3. factorize the KKT system, obtain $U_A(\theta)=S_A\theta+s_A$, $\lambda_A(\theta)=Q_A\theta+q_A$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -F^T\theta \\ w_A + E_A\theta \end{bmatrix} \Rightarrow \begin{array}{c} U_A(\theta) = S_A\theta + s_A \\ \lambda_A(\theta) = Q_A\theta + q_A \end{array}$$

4. determine the critical region $R = \{ \theta \mid GU_A(\theta) \le w+E\theta,\ \lambda_A(\theta) \ge 0 \} = \{ \theta \mid H_A\theta \le h_A \}$
5. explore the neighbors of R until convergence

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Geometric parametric programming:

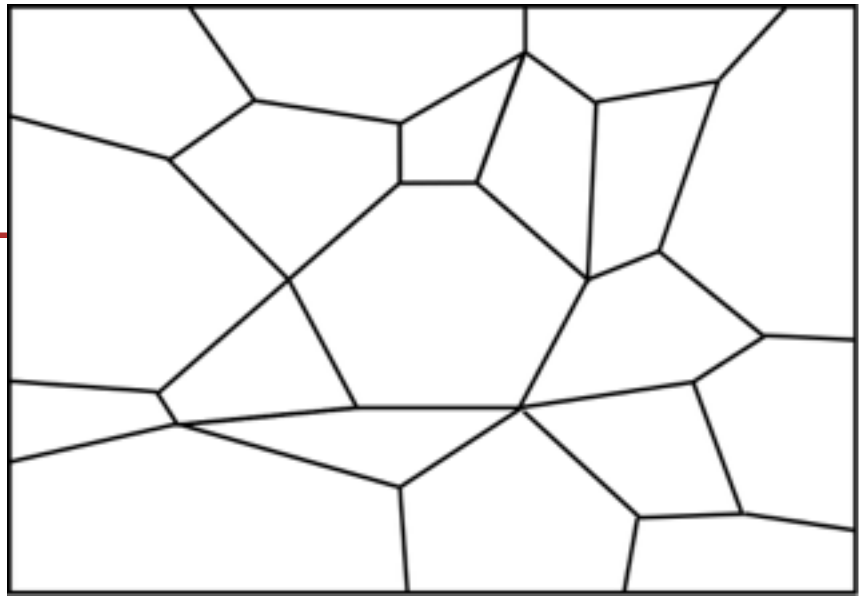
1. choose a feasible parameter θ
2. determine the active set A from *min 1/2* $U^T HU+\theta^T FU$ *s.t.* $GU \leq w+E\theta$
3. factorize the KKT system, obtain $U_A(\theta)=S_A\theta+s_A$, $\lambda_A(\theta)=Q_A\theta+q_A$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -F^T\theta \\ w_A + E_A\theta \end{bmatrix} \Rightarrow \begin{array}{c} U_A(\theta) = S_A\theta + s_A \\ \lambda_A(\theta) = Q_A\theta + q_A \end{array}$$

4. determine the critical region $R = \{ \theta \mid GU_A(\theta) \leq w+E\theta, \lambda_A(\theta) \geq 0 \} = \{ \theta \mid H_A\theta \leq h_A \}$
5. explore the neighbors of R until convergence

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_{U} \ {}^1\!/{}_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \le w + E\theta$$

Geometric parametric programming:

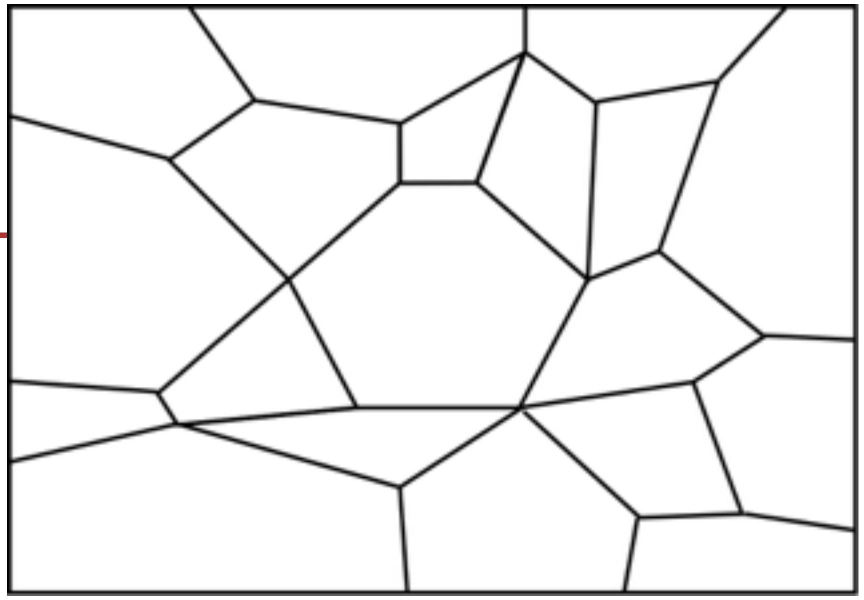
1. choose a feasible parameter θ
2. determine the active set A from *min 1/2* $U^T HU+\theta^T FU$ *s.t.* $GU \le w+E\theta$
3. factorize the KKT system, obtain $U_A(\theta)=S_A\theta+s_A$, $\lambda_A(\theta)=Q_A\theta+q_A$

$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -F^T\theta \\ w_A + E_A\theta \end{bmatrix} \Rightarrow \begin{matrix} U_A(\theta) = S_A\theta + s_A \\ \lambda_A(\theta) = Q_A\theta + q_A \end{matrix}$$

4. determine the critical region $R = \{ \theta \mid GU_A(\theta) \le w+E\theta,\ \lambda_A(\theta) \ge 0 \} = \{ \theta \mid H_A\theta \le h_A \}$
5. explore the neighbors of R until convergence
6. store S_A, s_A, H_A, h_A for on-line implementation

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Geometric parametric programming:

1. choose a feasible parameter θ
2. determine the active set A from $min\ 1/2\ U^T HU+\theta^T FU\ s.t.\ GU \leq w+E\theta$
3. factorize the KKT system, obtain $U_A(\theta)=S_A\theta+s_A$, $\lambda_A(\theta)=Q_A\theta+q_A$

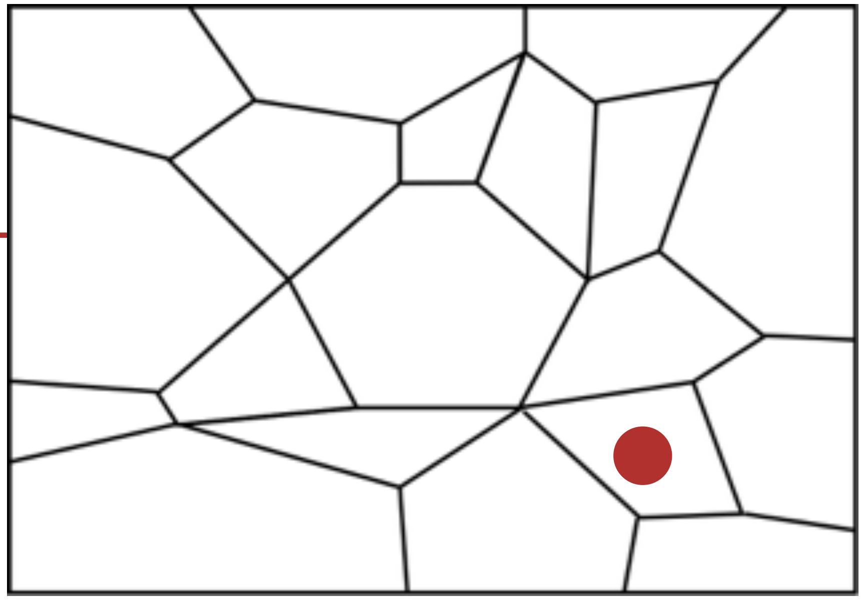
$$\begin{bmatrix} H & G_A^T \\ G_A & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -F^T\theta \\ w_A + E_A\theta \end{bmatrix} \Rightarrow \begin{matrix} U_A(\theta) = S_A\theta + s_A \\ \lambda_A(\theta) = Q_A\theta + q_A \end{matrix}$$

4. determine the critical region $R = \{\ \theta \mid GU_A(\theta) \leq w+E\theta,\ \lambda_A(\theta) \geq 0\ \} = \{\ \theta \mid H_A\theta \leq h_A\ \}$
5. explore the neighbors of R until convergence
6. store S_A, s_A, H_A, h_A for on-line implementation

lots of data!

A. Bemporad et al.: The explicit linear quadratic regulator for constrained systems; Automatica, 2002

Explicit MPC



$$\min_{U} \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Inputs: H_A, h_A, S_A, s_A for each region

On-line implementation: sequential search for the active region

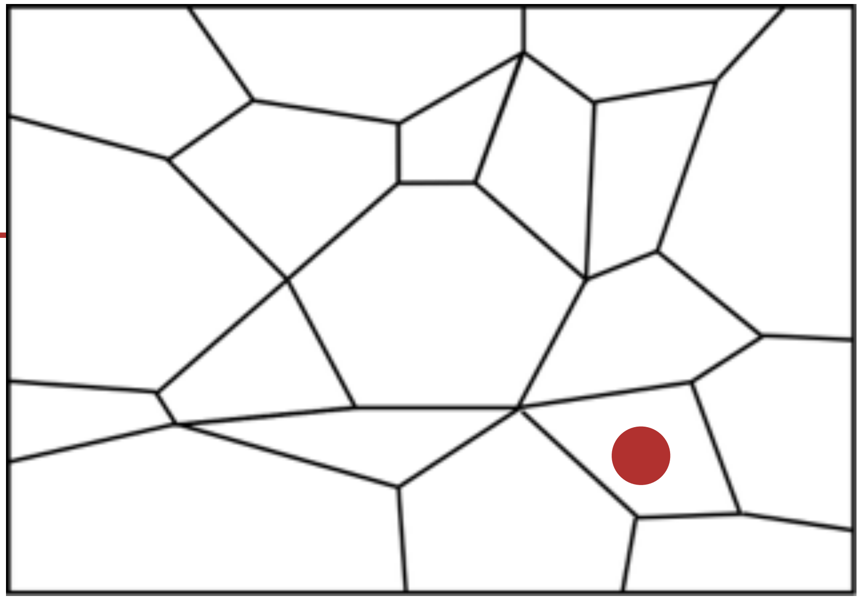
for each region **do**:

if $H_A\theta \leq h_A$ **then return** $U^* = S_A\theta + s_A$

end for

Extremely simple (10 lines of C code, no libraries, no divisions)

Explicit MPC



$$\min_{U} \ {}^{1}\!/_{2} U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Inputs: H_A, h_A, S_A, s_A for each region

On-line implementation: sequential search for the active region

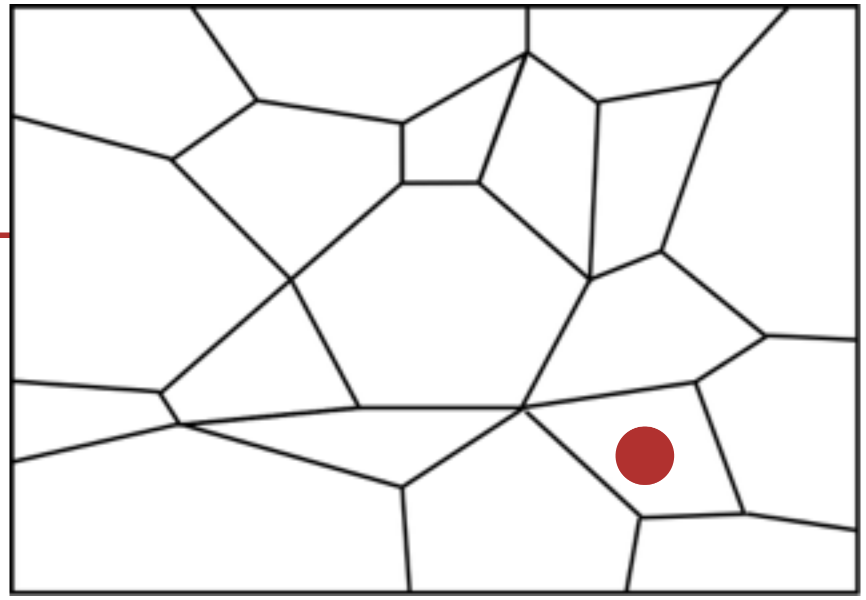
for each region **do**:

if $H_A\theta \leq h_A$ **then return** $U^* = S_A\theta + s_A$ — cheap

end for

Extremely simple (10 lines of C code, no libraries, no divisions)

Explicit MPC



$$\min_{U} \ {}^1\!/\!{}_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \le w + E\theta$$

Inputs: H_A, h_A, S_A, s_A for each region

On-line implementation: sequential search for the active region

for each region **do**: ← not so great

  if $H_A\theta \le h_A$ **then return** $U^* = S_A\theta + s_A$ ← cheap

end for

Extremely simple (10 lines of C code, no libraries, no divisions)

Side-by-side Comparison

	Implicit MPC	Explicit MPC
Problem size	Large	Small

Side-by-side Comparison

	Implicit MPC	Explicit MPC
Problem size	Large	Small
KKT solver	Online matrix inverse	Off-line factorization

Side-by-side Comparison

	Implicit MPC	Explicit MPC
Problem size	Large	Small
KKT solver	Online matrix inverse	Off-line factorization
Active set update	Heuristics	Sequential

Side-by-side Comparison

	Implicit MPC	Explicit MPC
Problem size	Large	Small
KKT solver	Online matrix inverse	Off-line factorization
Active set update	Heuristics	Sequential
Memory storage	Small	Large

Side-by-side Comparison

	Implicit MPC	Explicit MPC	Today
Problem size	Large	Small	Moderate
KKT solver	Online matrix inverse	Off-line factorization	Off-line factorization
Active set update	Heuristics	Sequential	Heuristics
Memory storage	Small	Large	Moderate

Side-by-side Comparison

	Implicit MPC	Explicit MPC	Today
Problem size	Large	Small	Moderate
KKT solver	Online matrix inverse	Off-line factorization	Off-line factorization
Active set update	Heuristics	Sequential	Heuristics
Memory storage	Small	Large	Moderate

Issue #1: Problem Size

$$\min_U \; {}^1\!/\!{}_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Three drivers of complexity:

- number of optimization variables $n_U = n_u N_c$
- number of constraints n_w
- number of parameters n_θ

Issue #1: Problem Size

$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Three drivers of complexity:

- number of optimization variables $n_U = n_u N_c$
- number of constraints n_w
- number of parameters n_θ

Geometric approach to explicit MPC:

- n_θ is the main limitation because geometric operations are required (finding a point on a facet, redundancy removal)
- usually applicable to problems with $n_\theta \leq 5$
- regions are essential to find all locally optimal active sets

Issue #1: Problem Size

$$\min_U \; {}^1\!/\!{}_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Three drivers of complexity:

- number of optimization variables $n_U = n_u N_c$
- number of constraints n_w
- number of parameters n_θ

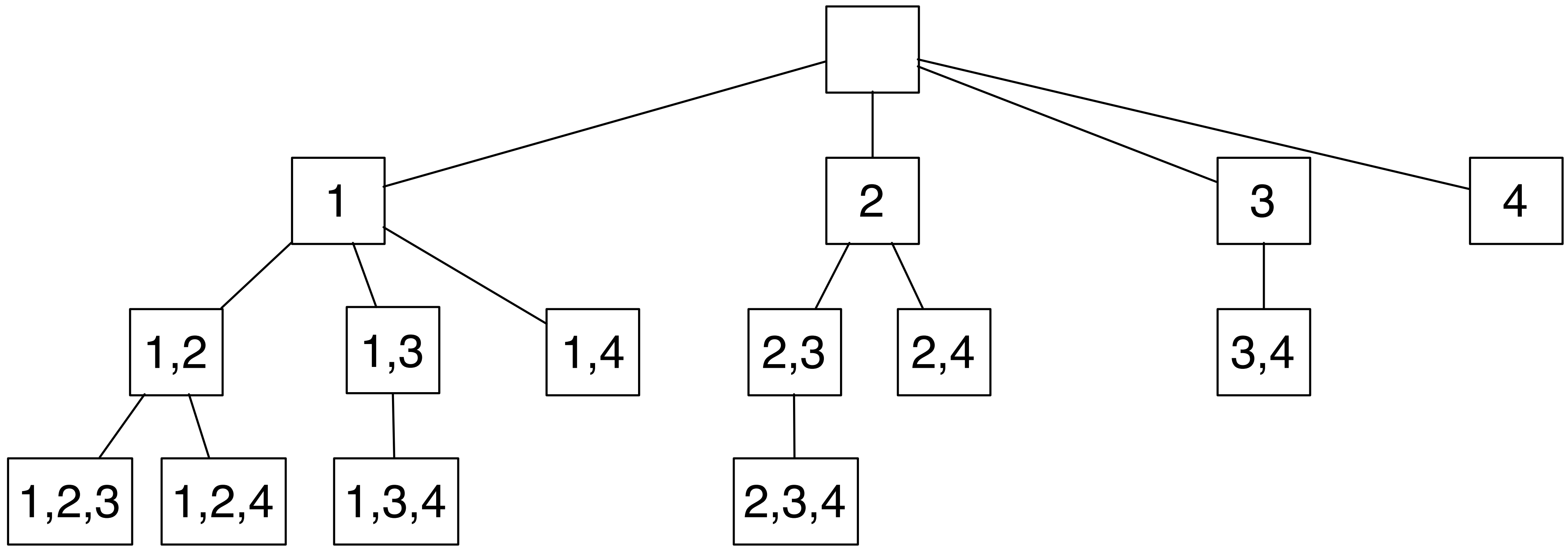
Geometric approach to explicit MPC:

- n_θ is the main limitation because geometric operations are required (finding a point on a facet, redundancy removal)
- usually applicable to problems with $n_\theta \leq 5$
- regions are essential to find all locally optimal active sets

Alternative: extensive enumeration of all locally optimal active sets

Extensive Enumeration Approach

Example: $n_U = 3$, $n_W = 4$, n_θ *free (but influences* n_W*)*



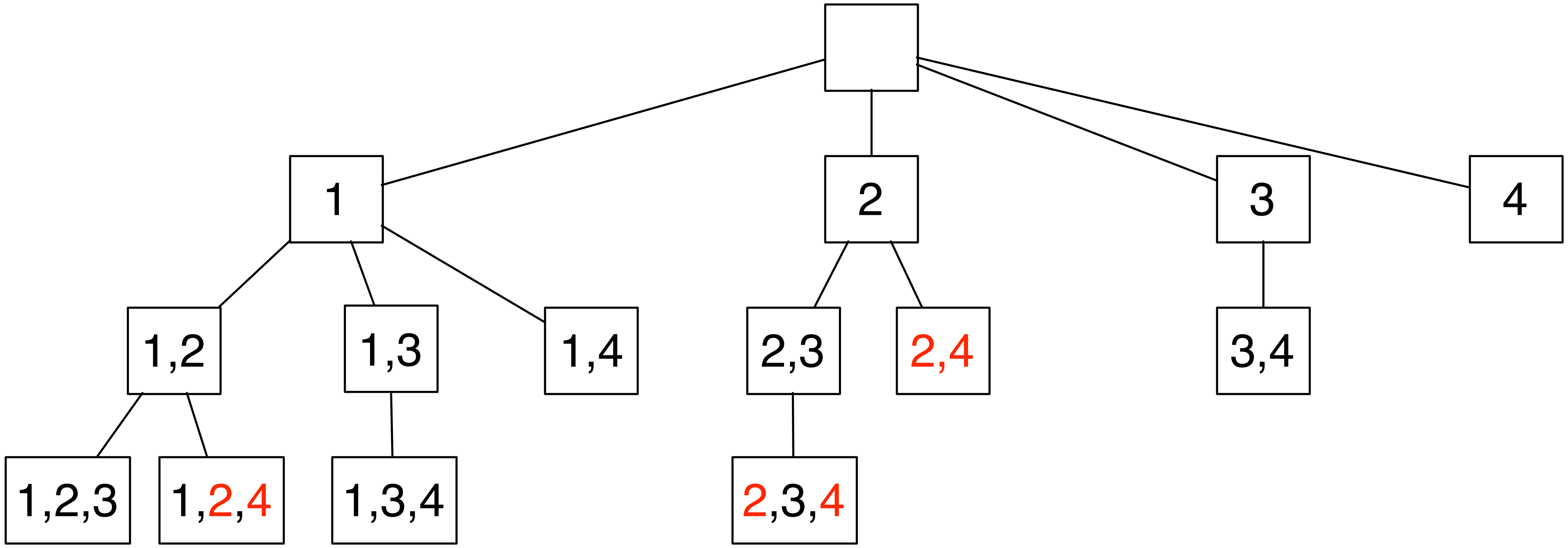
Off-line phase:

1. enumerate all possible combinations of active constraints

A. Gupta et al.: A novel approach to multiparametric quadratic programming; Automatica, 2011

Extensive Enumeration Approach

Example: $n_U = 3,\ n_W = 4,\ n_\theta$ *free (but influences* n_W*)*



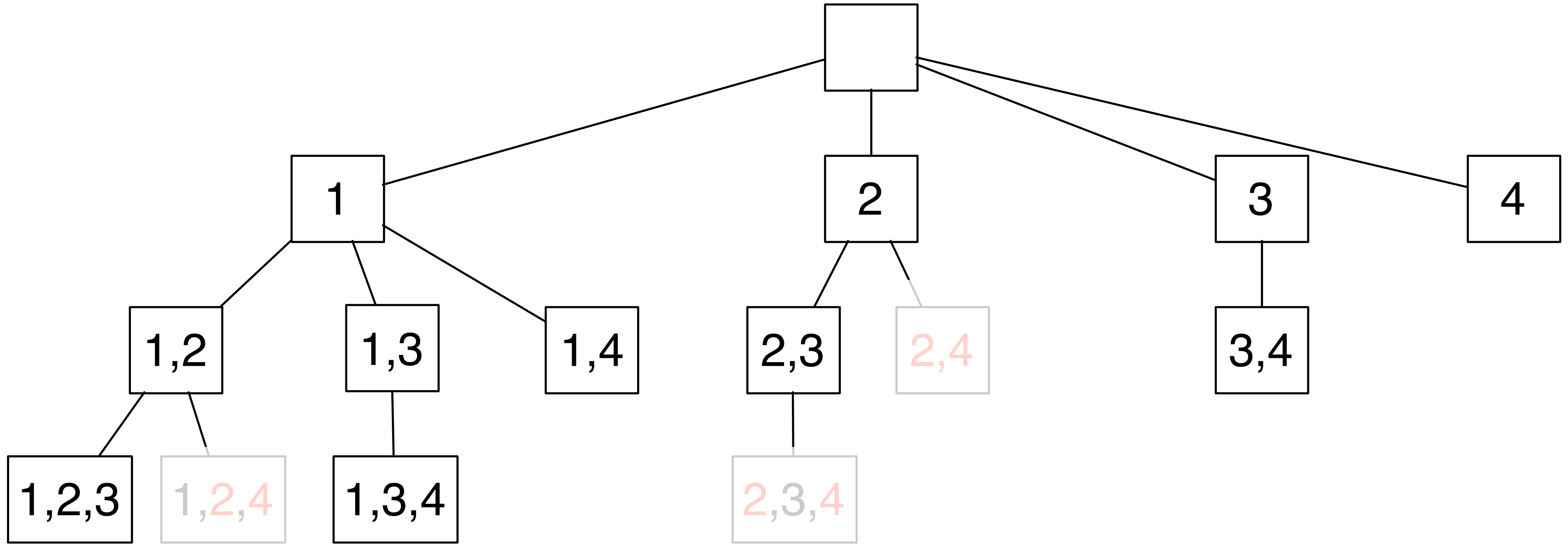
Off-line phase:

1. enumerate all possible combinations of active constraints
2. prune infeasible combinations

A. Gupta et al.: A novel approach to multiparametric quadratic programming; Automatica, 2011

Extensive Enumeration Approach

Example: $n_U = 3,\ n_w = 4,\ n_\theta$ *free (but influences* n_w*)*



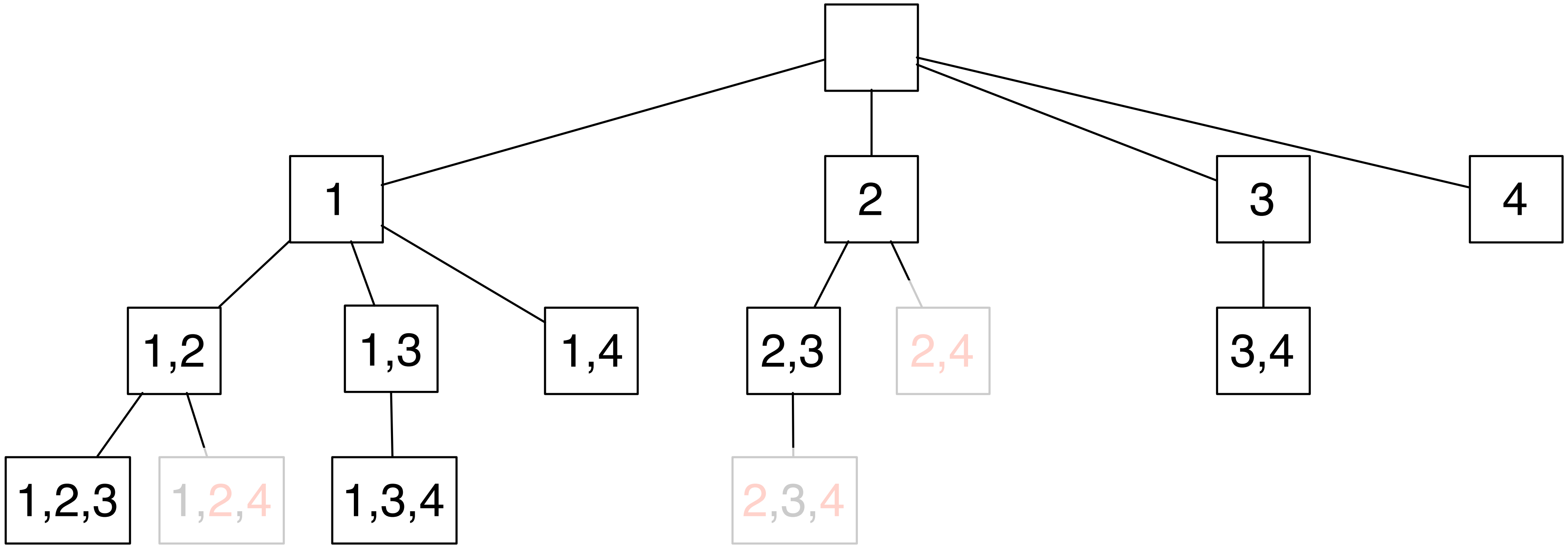
Off-line phase:

1. enumerate all possible combinations of active constraints
2. prune infeasible combinations
3. compute $U_A(\theta)=S_A\theta+s_A$, $\lambda_A(\theta)=Q_A\theta+q_A$ for each optimal active set
4. determine critical regions $R(A) = \{\ \theta \mid GU_A(\theta) \le w+E\theta,\ \lambda_A(\theta) \ge 0\ \} = \{\ \theta \mid H_A\theta \le h_A\ \}$

A. Gupta et al.: A novel approach to multiparametric quadratic programming; Automatica, 2011

Extensive Enumeration Approach

Example: $n_U = 3,\ n_w = 4,\ n_\theta$ *free (but influences* n_w*)*



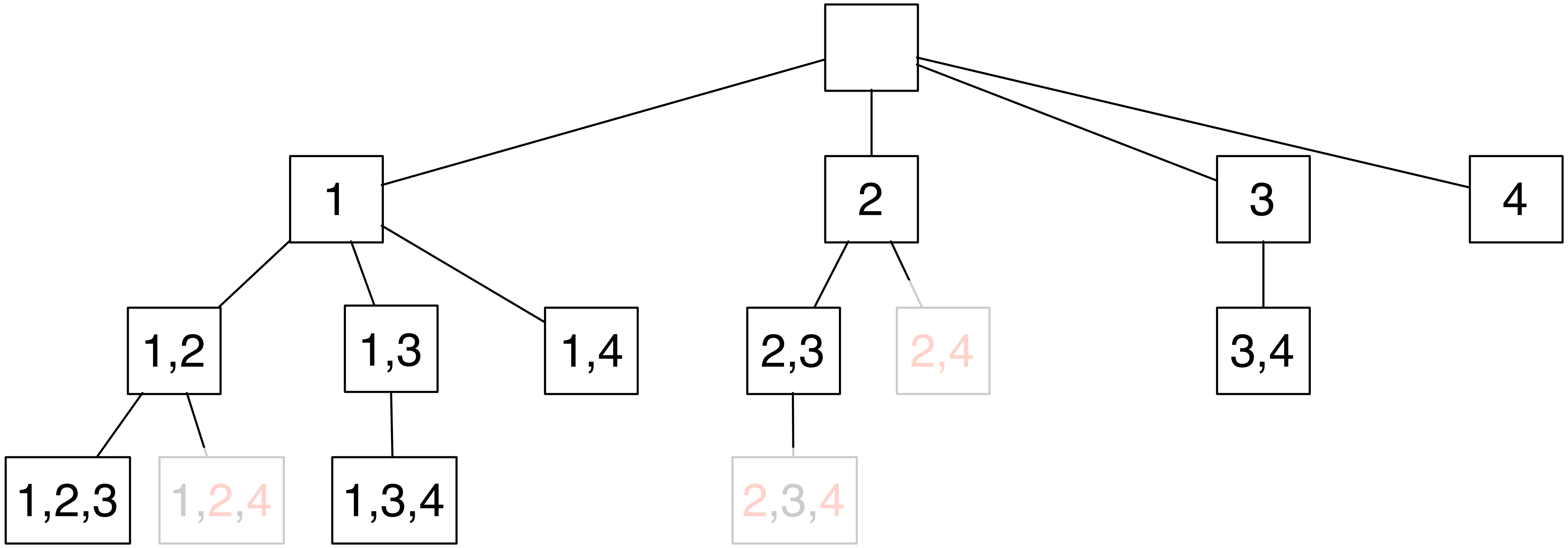
Off-line phase:

1. enumerate all possible combinations of active constraints
2. prune infeasible combinations
3. compute $U_A(\theta)=S_A\theta+s_A$, $\lambda_A(\theta)=Q_A\theta+q_A$ for each optimal active set
4. determine critical regions $R(A) = \{\ \theta \mid GU_A(\theta)\leq w+E\theta,\ \lambda_A(\theta)\geq 0\ \} = \{\ \theta \mid H_A\theta\leq h_A\ \}$
5. store S_A, s_A, H_A, h_A, for each $A \in \{A_1, \ldots, A_M\}$

A. Gupta et al.: A novel approach to multiparametric quadratic programming; Automatica, 2011

Extensive Enumeration Approach

Example: $n_U = 3,\ n_w = 4,\ n_\theta$ *free (but influences* n_w*)*



The good: size of the exploration tree is mainly a function of n_U

- works well for $n_U \leq 10$ (use move blocking!)
- number of parameters n_θ can be (very) large, $n_\theta \geq 100$ is not an issue

The bad: regions are still needed for on-line implementation

A. Gupta et al.: A novel approach to multiparametric quadratic programming; Automatica, 2011

Side-by-side Comparison

	Implicit MPC	Explicit MPC	Today
Problem size	Large	Small	Moderate
KKT solver	Online matrix inverse	Off-line factorization	Off-line factorization
Active set update	Heuristics	Sequential	Heuristics
Memory storage	Small	Large	Moderate

Side-by-side Comparison

	Implicit MPC	Explicit MPC	Today
Problem size	Large	Small	Moderate
KKT solver	Online matrix inverse	Off-line factorization	Off-line factorization
Active set update	Heuristics	Sequential	Heuristics
Memory storage	Small	Large	Moderate

Regionless Explicit MPC

Parametric solution consists of:

- polyhedral regions $R(A) = \{ \theta \mid GU_A(\theta) \leq w + E\theta,\ \lambda_A(\theta) \geq 0 \} = \{ \theta \mid H_A\theta \leq h_A \}$
- local affine optimizers $U_A(\theta) = S_A\theta + s_A$
- total memory: number of regions × size of one region

Regionless Explicit MPC

Parametric solution consists of:

- polyhedral regions $R(A) = \{ \theta \mid GU_A(\theta) \leq w + E\theta,\ \lambda_A(\theta) \geq 0 \} = \{ \theta \mid H_A\theta \leq h_A \}$
- local affine optimizers $U_A(\theta) = S_A\theta + s_A$
- total memory: number of regions × size of one region

Current practice (leads nowhere):

- simplify regions by removing redundant constraints from $R(A) = \{ \theta \mid H_A\theta \leq h_A \}$

Regionless Explicit MPC

Parametric solution consists of:

- polyhedral regions $R(A) = \{ \theta \mid GU_A(\theta) \leq w + E\theta,\ \lambda_A(\theta) \geq 0 \} = \{ \theta \mid H_A\theta \leq h_A \}$
- local affine optimizers $U_A(\theta) = S_A\theta + s_A$
- total memory: number of regions × size of one region

Current practice (leads nowhere):

- simplify regions by removing redundant constraints from $R(A) = \{ \theta \mid H_A\theta \leq h_A \}$

Better idea:

- get rid of the regions altogether!
- store local dual optimizers $\lambda_A(\theta) = Q_A\theta + q_A$ and matrices G, w, E (just once!)
- replace $\theta \in R(A)$ by direct check of primal and dual feasibility:
  $\theta \in R(A) \Longleftrightarrow \theta \in \{ \theta \mid GU_A(\theta) \leq w + E\theta,\ \lambda_A(\theta) \geq 0 \}$

F. Borrelli et al.: On the computation of linear MPC laws; Automatica, 2010

Regionless Explicit MPC

	Regions	Primal optimizer $U_A(\theta) = S_A\theta + s_A$	Dual optimizer $\lambda_A(\theta) = Q_A\theta + q_A$
Region-based	stored	stored	not stored

Regionless Explicit MPC

	Regions	Primal optimizer $U_A(\theta) = S_A\theta + s_A$	Dual optimizer $\lambda_A(\theta) = Q_A\theta + q_A$
Region-based	stored	stored	not stored
Regionless	not stored	stored	stored

Regionless Explicit MPC

	Regions	Primal optimizer $U_A(\theta) = S_A\theta + s_A$	Dual optimizer $\lambda_A(\theta) = Q_A\theta + q_A$
Region-based	stored	stored	not stored
Regionless	not stored	stored	stored
Improved regionless*	not stored	not stored	stored

**M. Kvasnica et al.: On region-free explicit model predictive control; CDC 2015*

Regionless Explicit MPC

Illustrative example: $n_U = 3,\ n_w = 298,\ n_\theta = 40$

	Region-based approach	Regionless approach of Borrelli et al.	Improved regionless approach of Kvasnica et al.
# of regions / active sets	27 544		
Memory storage	949 MB	3.6 MB	1.8 MB

Bottom line: we can <u>construct</u> and <u>store</u> explicit MPC even for systems with 100+ states (if $n_U \leq 10$)

Side-by-side Comparison

	Implicit MPC	Explicit MPC	Today
Problem size	Large	Small	Moderate
KKT solver	Online matrix inverse	Off-line factorization	Off-line factorization
Active set update	Heuristics	Sequential	Heuristics
Memory storage	Small	Large	Moderate

Side-by-side Comparison

	Implicit MPC	Explicit MPC	Today
Problem size	Large	Small	Moderate
KKT solver	Online matrix inverse	Off-line factorization	Off-line factorization
Active set update	Heuristics	Sequential	Heuristics
Memory storage	Small	Large	Moderate

Implicit MPC

$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Active set method (feasible-start Newton):

1. choose an initial active set A and a feasible solution U
2. solve for Δ from $\min 1/2(U+\Delta)^T H(U+\Delta)+\theta^T F(U+\Delta)$ s.t. $G_A(U+\Delta)=w_A+E_A\theta$
3. heuristic update the active set:
   - if some $\lambda_i<0$, drop a constraint from A
   - if some inactive constraint is violated by $U+\Delta$, add a constraint to A
4. determine the step length α, update $U := U+\alpha\Delta$
5. repeat until convergence

What makes this work nicely: optimization over increments Δ

Incremental Regionless Explicit MPC

Instead of this:

$$\min_{U} \ ^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Solve this:

$$\min_{\Delta} \ ^1\!/_2 (U+\Delta)^T H (U+\Delta) + \theta^T F (U+\Delta)$$
$$\text{s.t. } G(U+\Delta) \leq w + E\theta$$

More parameters: θ and U, but can be easily done via the region-less approach

Explicit optimizers: $\Delta = S_A (U^T, \theta^T)^T + s_A$ and $\lambda = Q_A (U^T, \theta^T)^T + q_A$

Incremental Regionless Explicit MPC

$$\min_U \ {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Active set method (feasible-start Newton):

1. choose an initial active set *A* and a feasible solution *U*
2. **solve for *Δ* from *min 1/2(U+Δ)^{T}H(U+Δ)+θ^TF(U+Δ) s.t. G_A(U+Δ)=w_A+$E_A\theta$***
3. heuristic update the active set:
   - if some λ_i<0, drop a constraint from *A*
   - if some inactive constraint is violated by U+*Δ*, add a constraint to *A*
4. determine the step length *α*, update U := U+*αΔ*
5. repeat until convergence

What makes this work nicely: optimization over increments *Δ*

Incremental Regionless Explicit MPC

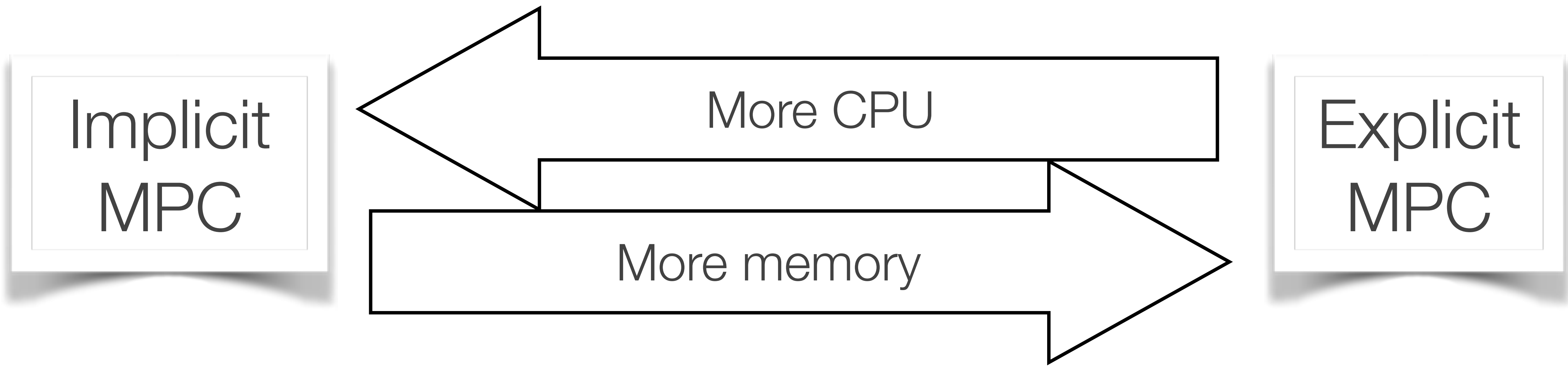
$$\min_{U} \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \le w + E\theta$$

Active set method (feasible-start Newton):

1. choose an initial active set *A* and a feasible solution *U*
2. **evaluate $\Delta = S_A(U^T,\theta^T)^T + s_A$ and $\lambda = Q_A(U^T,\theta^T)^T + q_A$**
3. heuristic update the active set:
   - if some $\lambda_i < 0$, drop a constraint from *A*
   - if some inactive constraint is violated by $U+\Delta$, add a constraint to *A*
4. determine the step length α, update $U := U + \alpha\Delta$
5. repeat until convergence

Bottom line: we can construct, store, and implement explicit MPC even for systems with 100+ of states

Conclusions



Bottom line: implicit MPC and explicit MPC are two sides of the same coin!

Backups

Conclusions

$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

	Implicit MPC (active-set method)	Explicit MPC (incremental regionless)
Memory storage	*H, F, G, w, E*	*G, w, E* primal optimizer, dual optimizer

Conclusions

$$\min_{U} \ {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

	Implicit MPC (active-set method)	Explicit MPC (incremental regionless)
Memory storage	*H, F, G, w, E*	*G, w, E* primal optimizer, dual optimizer
KKT solver	Matrix inversions	Matrix/vector multiplication

Conclusions

$$\min_U \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \le w + E\theta$$

	Implicit MPC (active-set method)	Explicit MPC (incremental regionless)
Memory storage	*H, F, G, w, E*	*G, w, E* primal optimizer, dual optimizer
KKT solver	Matrix inversions	Matrix/vector multiplication
Active set update	Identical algorithm	

Conclusions

$$\min_{U} \; {}^{1}\!/_{2} U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

	Implicit MPC (active-set method)	Explicit MPC (incremental regionless)
Memory storage	*H, F, G, w, E*	*G, w, E* primal optimizer, dual optimizer
KKT solver	Matrix inversions	Matrix/vector multiplication
Active set update	Identical algorithm	

Bottom line: implicit MPC and explicit MPC are two sides of the same coin!

Regionless Explicit MPC - Implementation

$$\min_{U} \; {}^1\!/_2 U^T H U + \theta^T F U$$
$$\text{s.t. } GU \leq w + E\theta$$

Inputs:

- constraints G, w, E
- primal optimizers $U_A(\theta)=S_A\theta+s_A$
- dual optimizers $\lambda_A(\theta) = Q_A\theta+q_A$

Online implementation via sequential search:

for each feasible active set $A \in \{A_1, \ldots, A_M\}$ **do**:

evaluate $U_A=S_A\theta+s_A$ and $\lambda_A = Q_A\theta+q_A$

if $GU_A \leq w_A + E_A\theta$ and $\lambda_A \geq 0$ **then return** $U^* = U_A$

end do

Can we do better than the sequential search?