Centre for Process Systems Engineering

THE QUEEN'S ANNIVERSARY PRIZES 2002

Nonlinear MPC via Novel Multiparametric Programming Techniques

Diogo Narciso
d.narciso@imperial.ac.uk

Stratos Pistikopoulos
e.pistikopoulos@imperial.ac.uk
Centre for Process Systems Engineering (CPSE), Imperial College London

MARIE CURIE ACTIONS

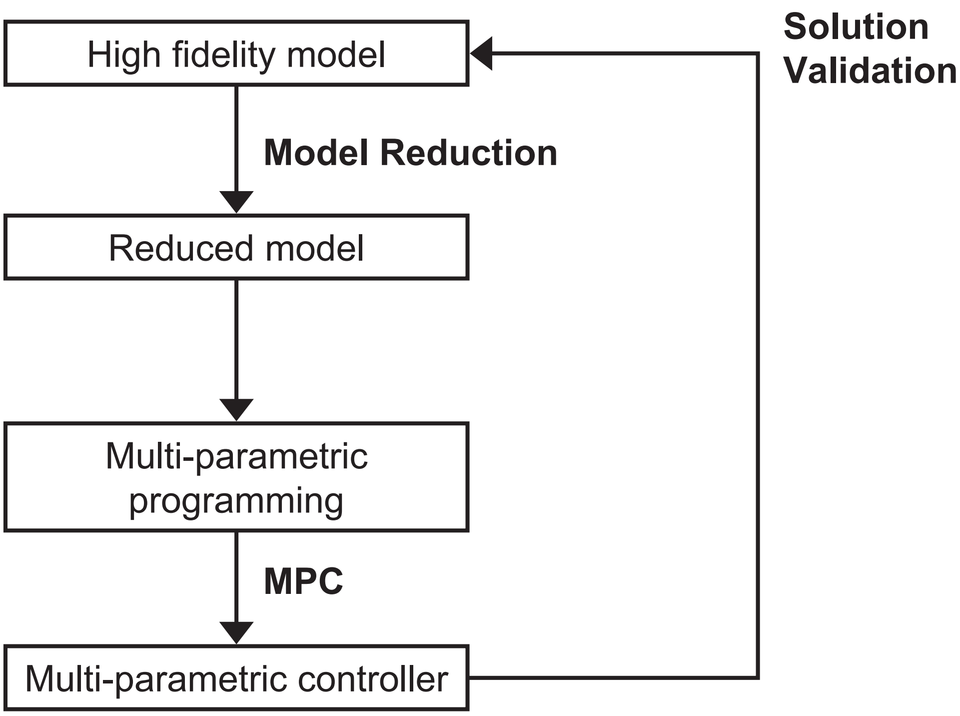
PROMATCH

REDUCIT Workshop, Frankfurt, Germany *04/11/2008*

Outline

- Framework and objectives
- Nonlinear Multiparametric discrete-time MPC
  - Formulation & overall strategy
  - Off-line strategy
  - Parametric Algorithm
  - On-line strategy
  - Research and Development Achievements
- Concluding remarks and future work

Framework and objectives

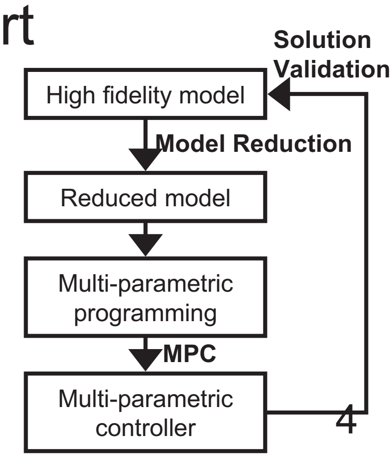


Framework and objectives

Model Reduction → Reduces model complexity/size

Multiparametric programming → Reduces online control effort

Reduce computational effort or simulation/optimisation time



Nonlinear Multiparametric discrete-time MPC

- Formulation & overall strategy
- Off-line strategy
- Parametric Algorithm
- On-line strategy

Formulation & overall strategy

Traditional MPC:

$$\begin{aligned} \min_{U} \ & \text{Objective Function}\,(U, x(t), z(t)) \\ s.t. \ & \text{Dynamic Model}\,(U, x(t), z(t)) \\ & \text{Process Constraints}\,(U, x(t), z(t)) \\ & \text{Initial Conditions} \end{aligned}$$

- Requires specification of initial conditions
- Optimal inputs calculated for the vector of initial conditions

Multiparametric MPC:

$$\begin{aligned} \min_{U} \ & \text{Objective Function}\,(U, \theta) \\ s.t. \ & \text{Dynamic Model}\,(U, \theta) \\ & \text{Process Constraints}\,(U, \theta) \\ & \text{Parameter Space Definition} \end{aligned}$$

- States/disturbances recast as parameters (θ)
- The space of expected parameters defined as an independent space
- Problem is solved for all combinations of parameters (semi-infinite problem)

Formulation & overall strategy

OFF-LINE

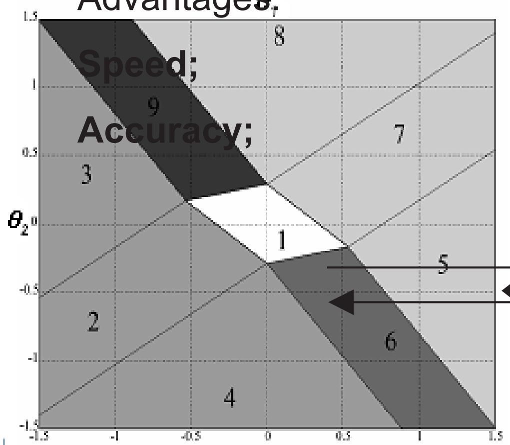
MPC Formulation

Multiparametric MPC Formulation

Advantages:

Speed;

Accuracy;



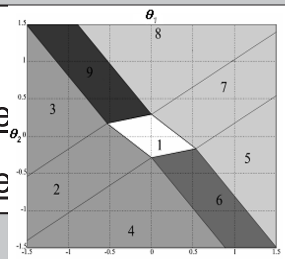
Recast:

$U(\theta)_i = \omega_i + W_i\theta$

User defines parameters

$CR_i: \Phi_i\theta \le \phi_i$

User defines parameters



$U(\theta)_i = \omega_i + W_i\theta$

CR_i: $\Phi_i\theta \le \phi_i$

States (Parameters)

Critical Regions

Parametric Solutions

Optimal inputs

ON-LINE

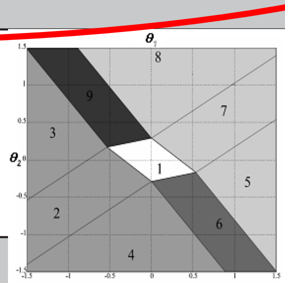
Multiparametric MPC Solution

Off-line strategy

OFF-LINE

MPC Formulation

Multiparametric MPC Formulation



$U(\theta)_i = \omega_i + W_i\theta$

CR_i: $\Phi_i\theta \le \phi_i$

States (Parameters)

ON-LINE

Process

Multiparametric MPC Solution

Optimal inputs

Off-line strategy

1) Select parameters
   1) States
   2) Load disturbances
2) Define parameter space
   1) e. g. Reactor temperature between 25 and 75C

$$\min_U \text{ Objective Function } (U, x(t), z(t))$$
$$s.t. \text{ Dynamic Model } (U, x(t), z(t))$$
$$\text{Process Constraints } (U, x(t), z(t))$$
$$\text{Initial Conditions}$$

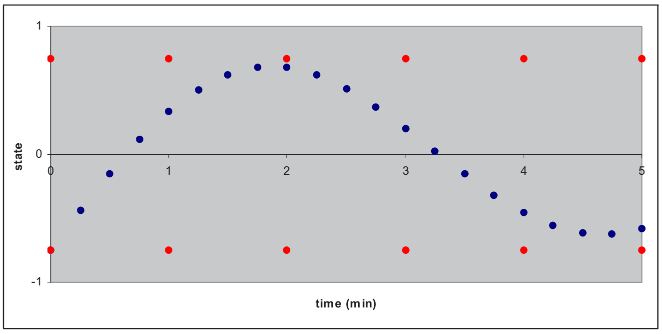
$$\min_U \text{ Objective Function } (U, \theta)$$
$$s.t. \text{ Dynamic Model } (U, \theta)$$
$$\text{Process Constraints } (U, \theta)$$

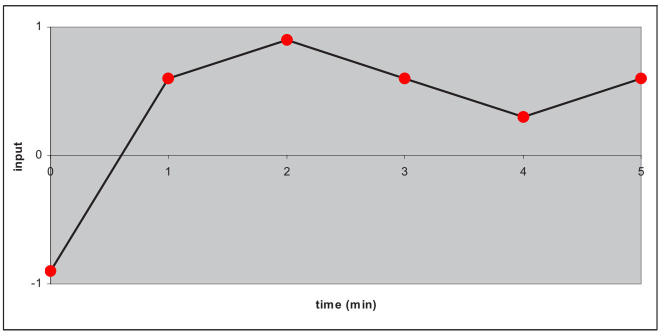
$$\min_U \text{ Objective Function } (U, \theta)$$
$$s.t. \text{ Dynamic Model } (U, \theta)$$
$$\text{Process Constraints } (U, \theta)$$
$$\text{Parameter Space Definition}$$

Centre for Process Systems Engineering
PROMATCH

Off-line strategy

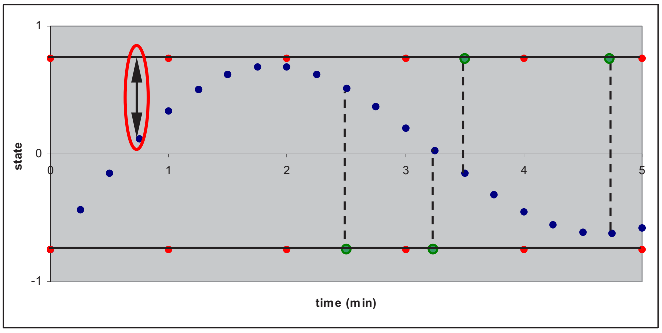
3) Discretise time domain
   1) Need to express state constraints with finite number of equations
4) Discretise controls and load disturbances
   1) Control vector parameterization





Off-line strategy

5) Assess constraints
   1) Formulate candidate point constraints
   2) Solve relaxed optimisation problem
   3) Add new constraints to mp-NLP problem



$$\begin{aligned} &\min_{U,\theta}\ \text{Objective Function}\,(U,\theta) \\ s.t.\ &\text{Dynamic Model}\,(U,\theta) \\ &\text{Process Constraints}\,(U,\theta) \\ &\max_i\,(\text{Process Constraint Violation})_i, \\ &(\text{Process Constraint Violation})_i^{max} > \epsilon_r \\ &\theta \in \Theta \end{aligned}$$

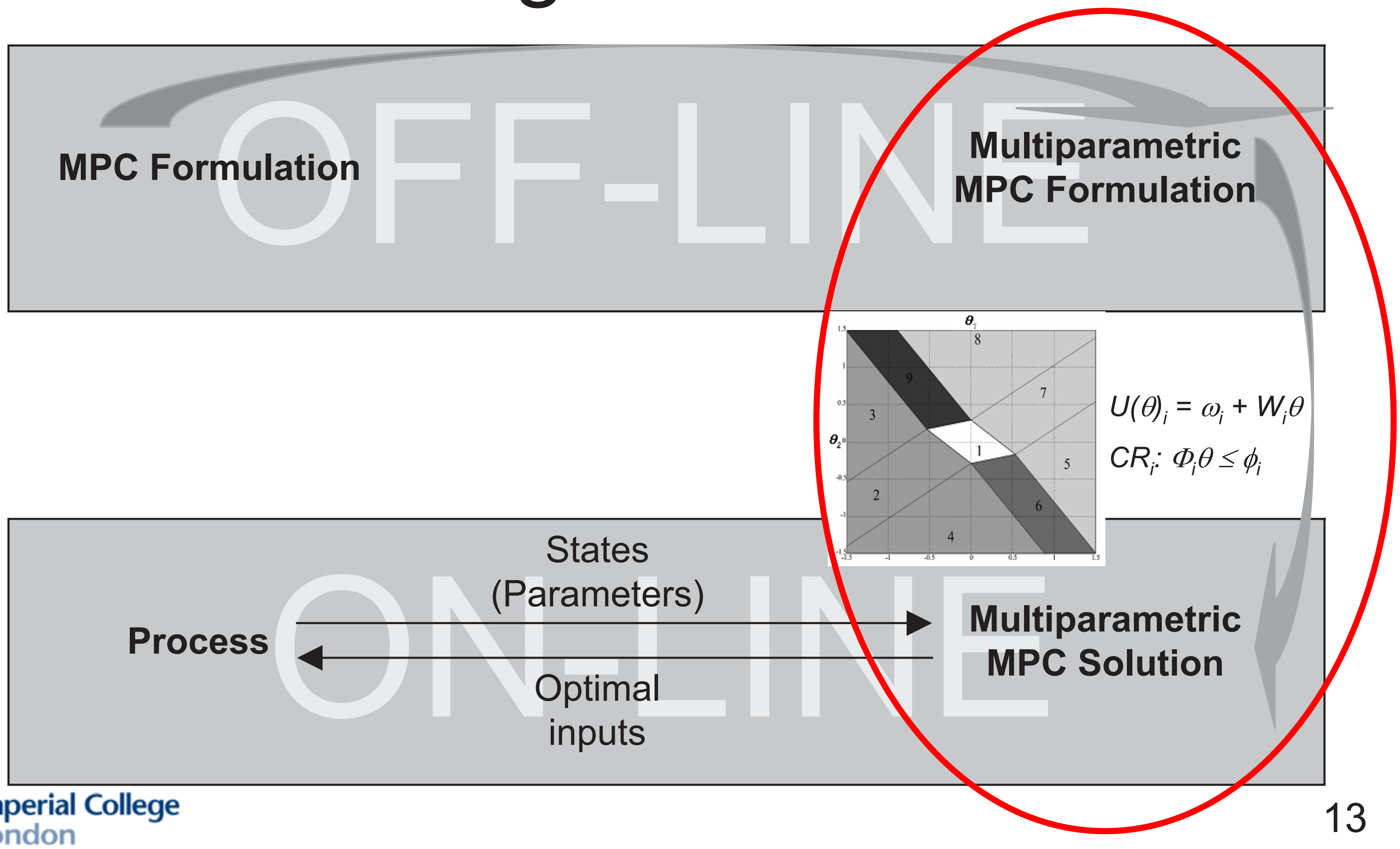
Centre for Process Systems Engineering
PROMATCH

Off-line strategy

$$\begin{aligned} &\min_{U}\ \text{Objective Function}\,(U,\theta) \\ s.t.\ &\text{Dynamic Model}\,(U,\theta) \\ &\text{Process Constraints}\,(U,\theta) \\ &\text{Parameter Space Definition} \end{aligned}$$

- Vector of inputs (finite)
- Inputs parameterized
- States discretised in time
- Vector of constraints (finite)

Parametric Algorithm



Parametric Algorithm

- Driven by accurate active-set characterization of parameter space
- Vertex search method
- Valid for convex formulations

Parametric Algorithm

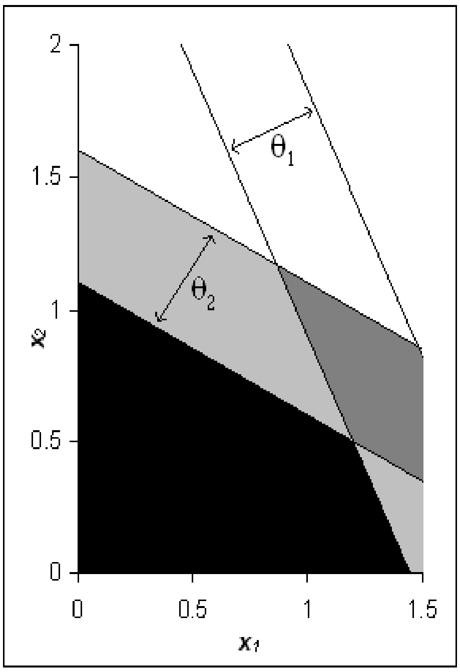
- Components of the new algorithm
  - Search of vertices
  - Construction of critical regions
  - Approximation of optimal solutions

Parametric Algorithm

Problem formulation:

$$\min_{x_1,x_2} x_1^3 + 2x_1^2 - 5x_1 + 6x_2^2 - 3x_2 - 6$$
$$s.t.$$
$$2x_1 + x_2 \le 2.5 + \theta_1$$
$$0.5x_1 + x_2 \le 1.5 + \theta_2$$
$$x_1 \ge 0$$
$$x_2 \ge 0$$
$$0 \le \theta_1 \le 1$$
$$0 \le \theta_2 \le 1$$

Feasible (***x***) space:

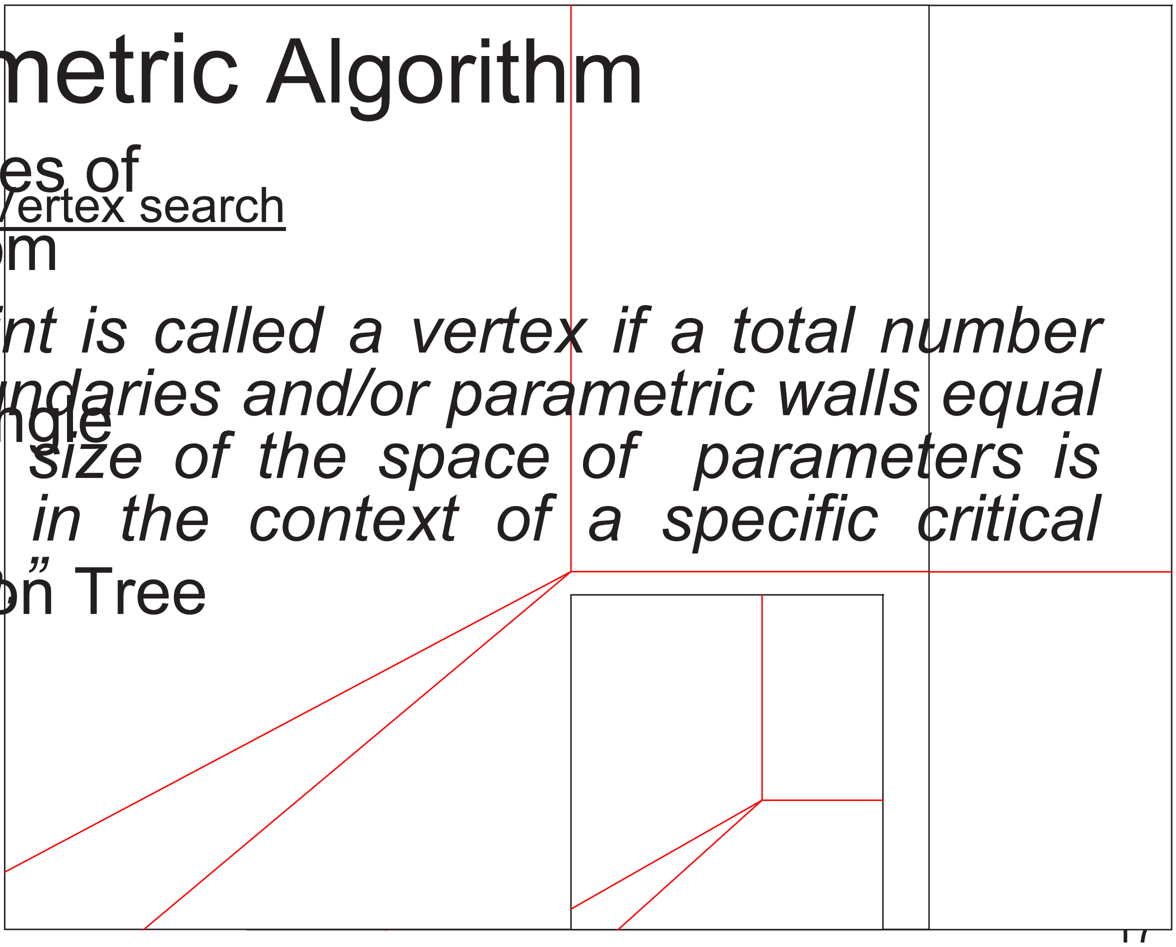


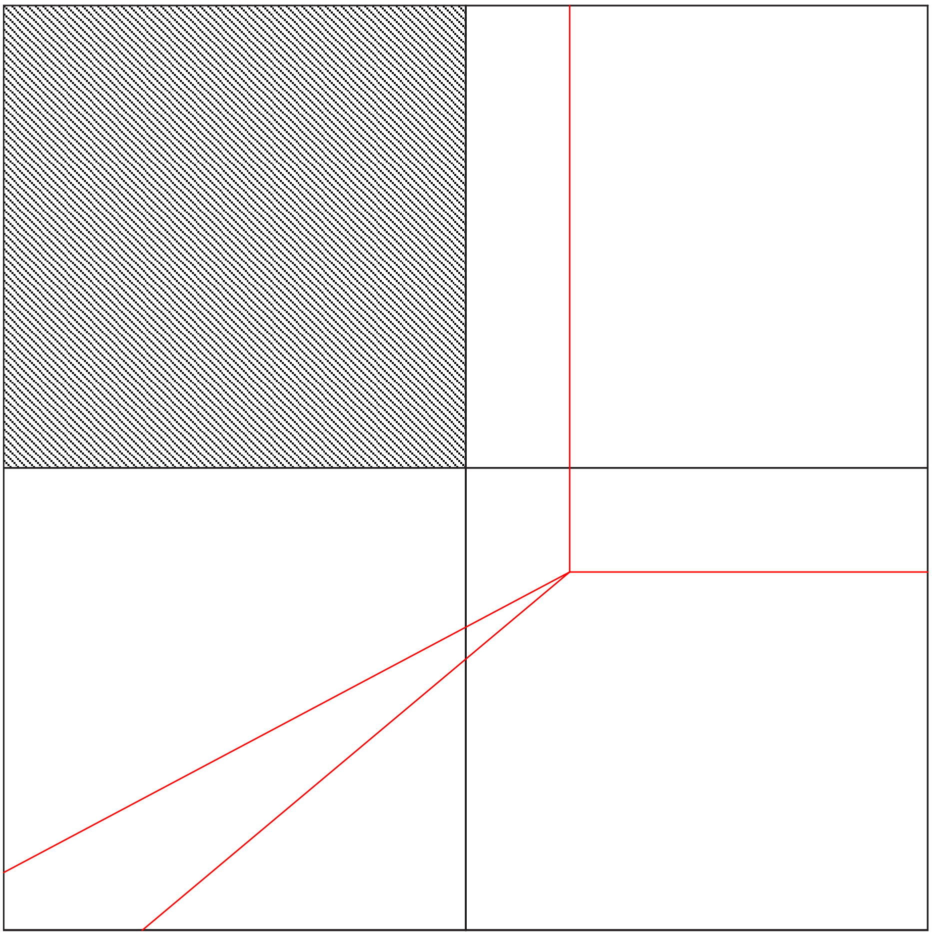
Parametric Algorithm

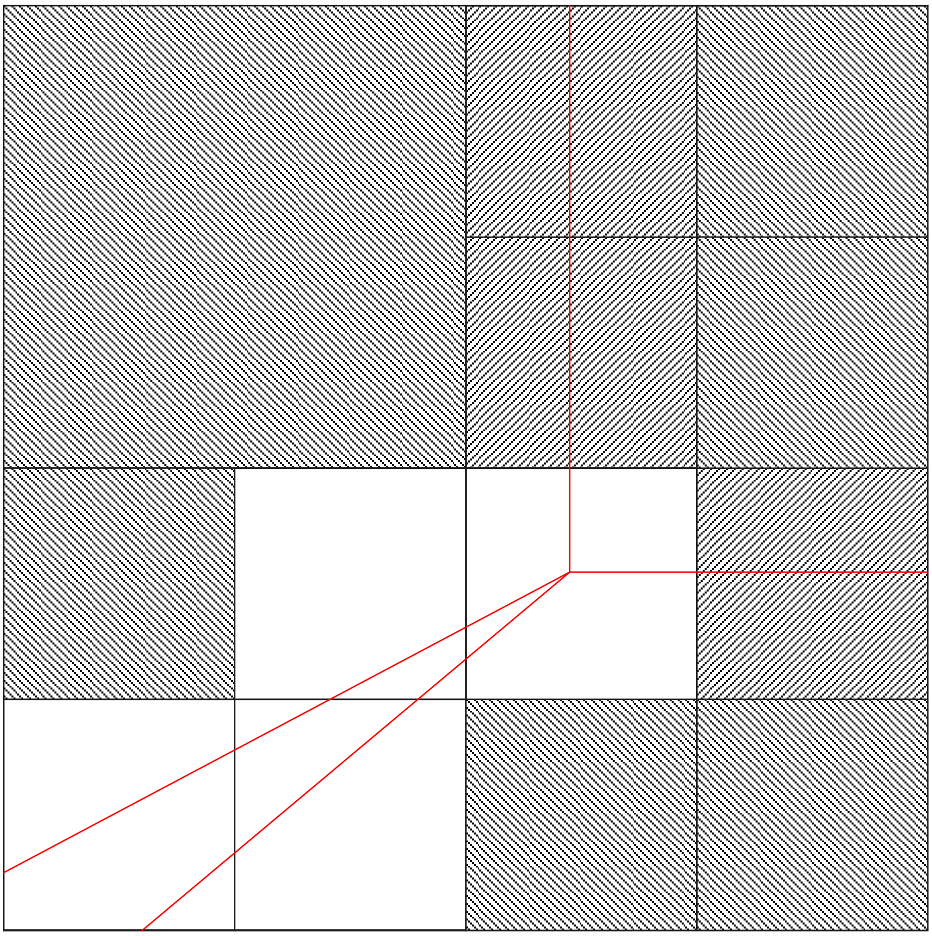
- Degrees of freedom
- Hyper Rectangle
- Local Solution Tree

<u>Stage 1: Vertex search</u>

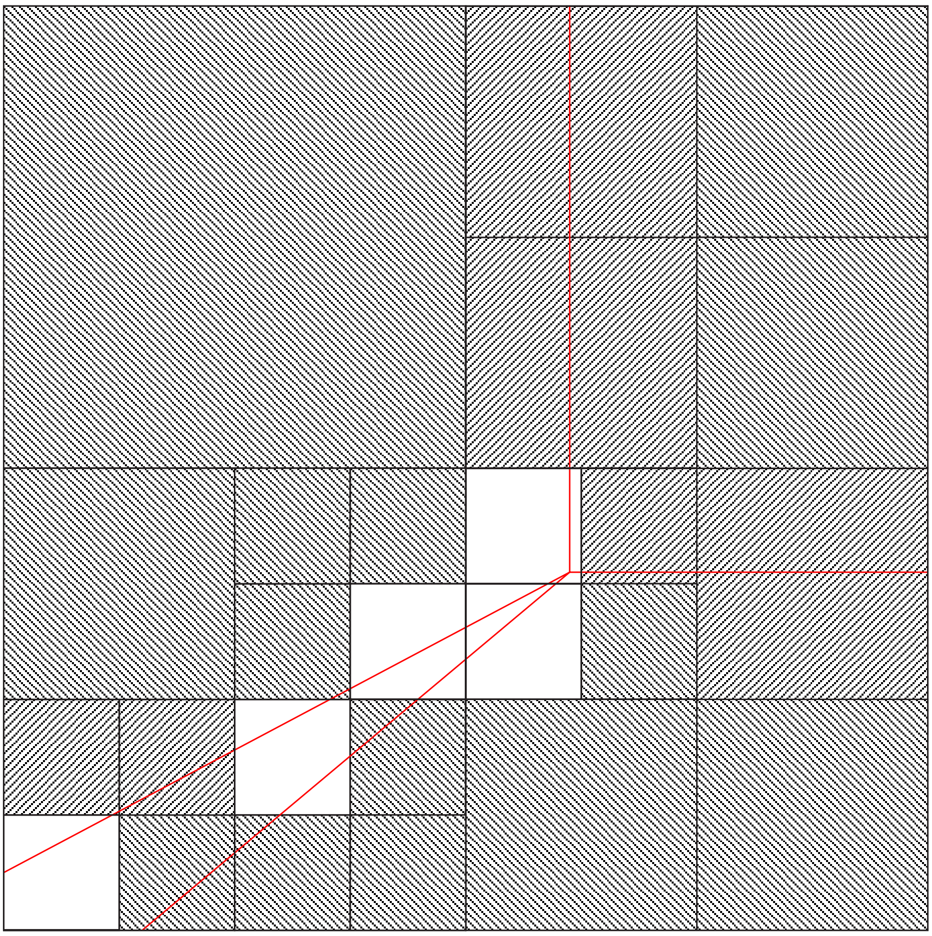
"A point is called a vertex if a total number of boundaries and/or parametric walls equal to the size of the space of parameters is active in the context of a specific critical region"



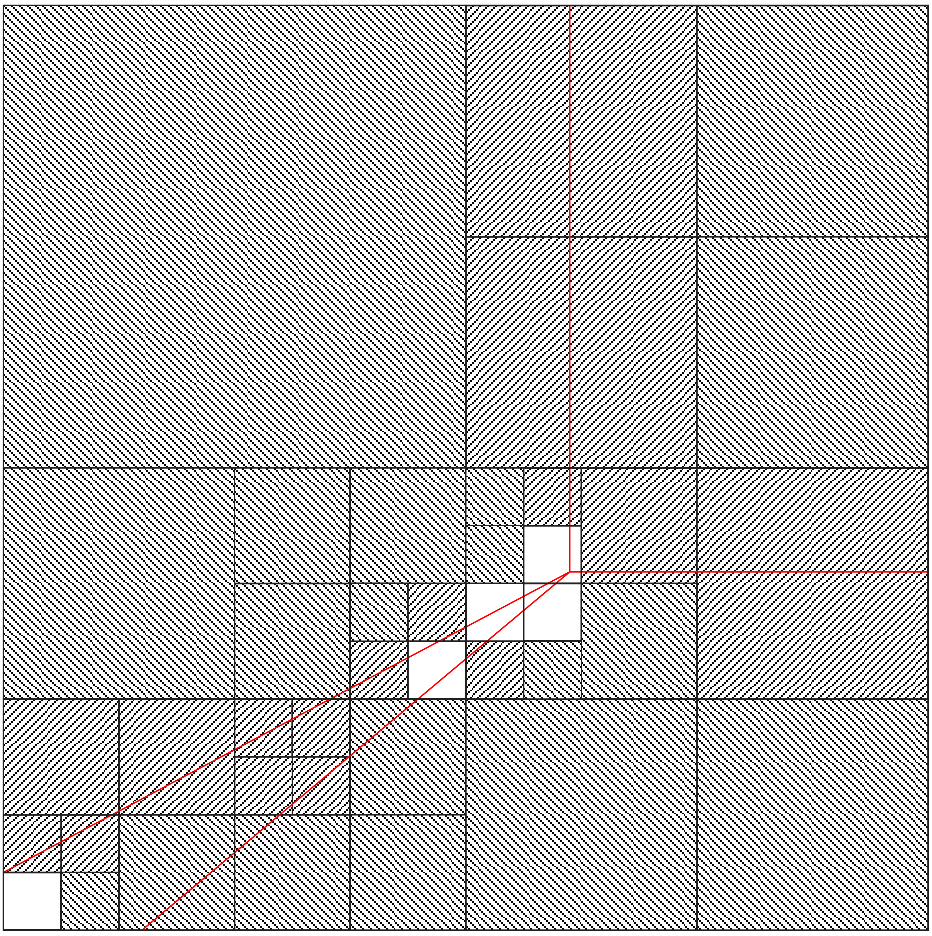


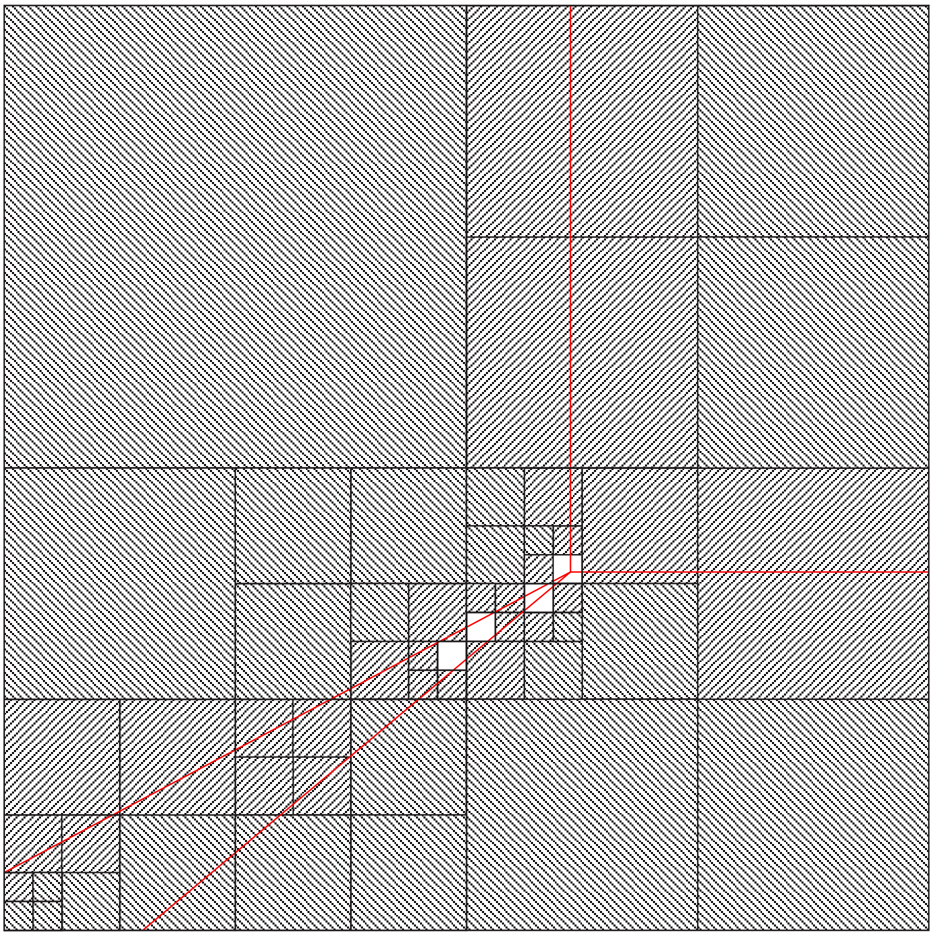


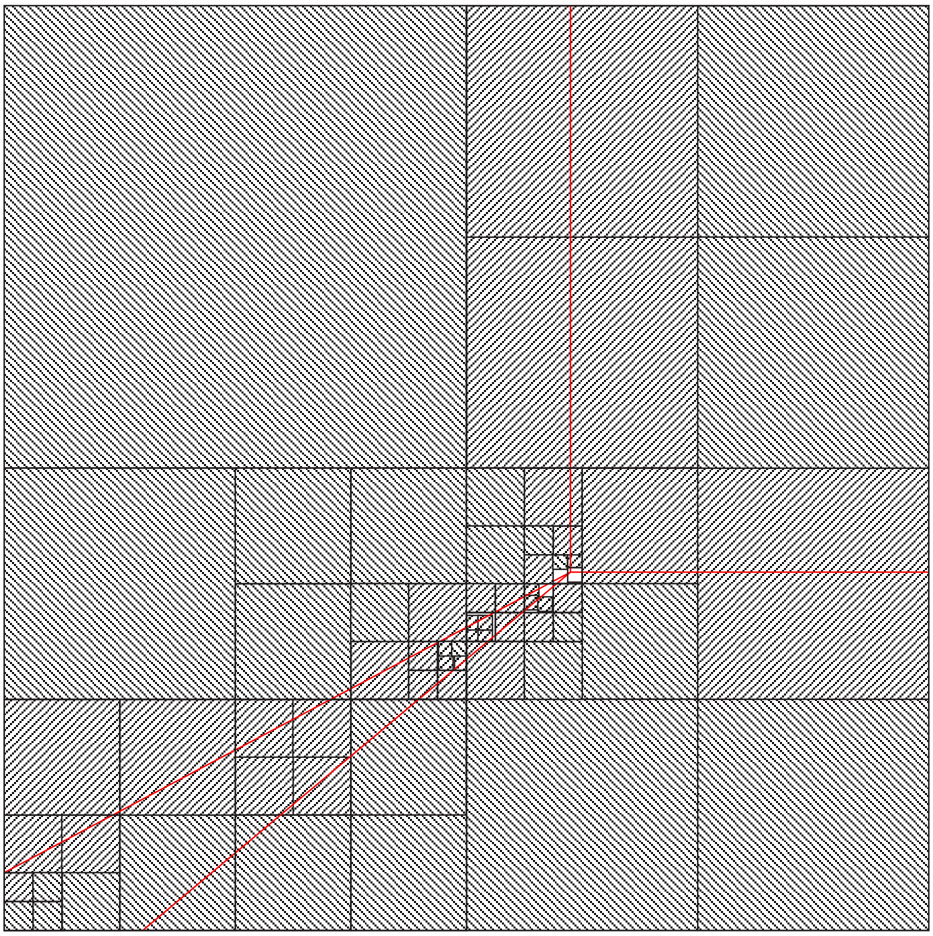
Centre for Process Systems Engineering
PROMATCH



Imperial College London







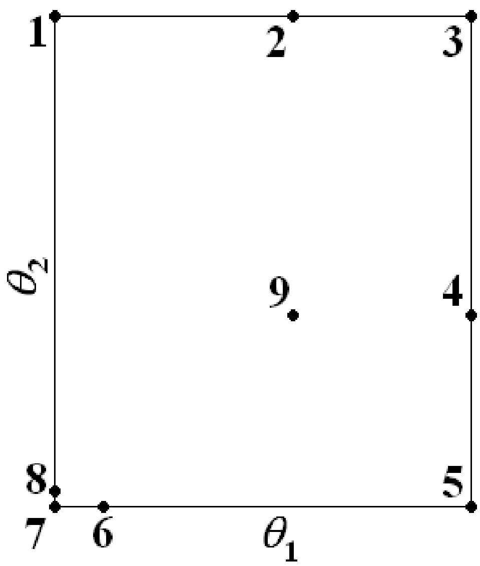
Centre for Process Systems Engineering
PROMATCH

Parametric Algorithm

Stage 1: Vertex search

- The vertices are just points of the parameter space!
- How to use this information to approximate the critical regions?

Parameter space (Θ):
Found Vertices



Imperial College London

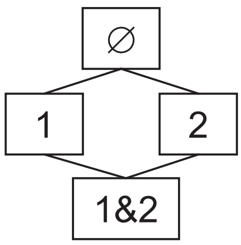
Parametric Algorithm

Stage 2: Boundary construction

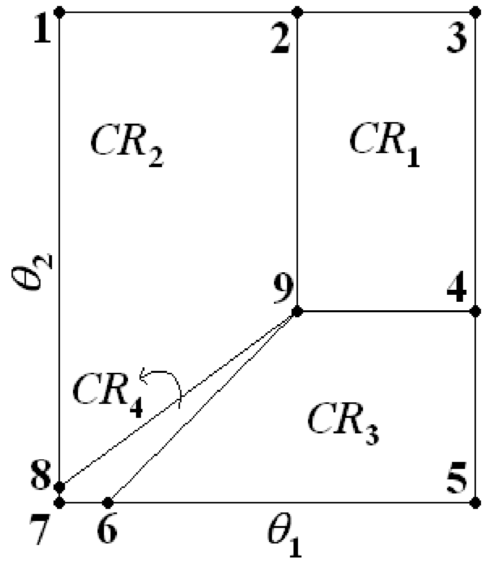
- We assess all the active sets in the vicinity of each of the found vertices

Vertex	1	2	3	4	5	6	7	8	9
Active Constraint	1	∅ 1	∅	∅ 2	2	2 1&2	1&2	1 1&2	∅ 1 2 1&2

- This information enables one to derive a list of all existing boundaries between active sets, based on the **solution tree:**



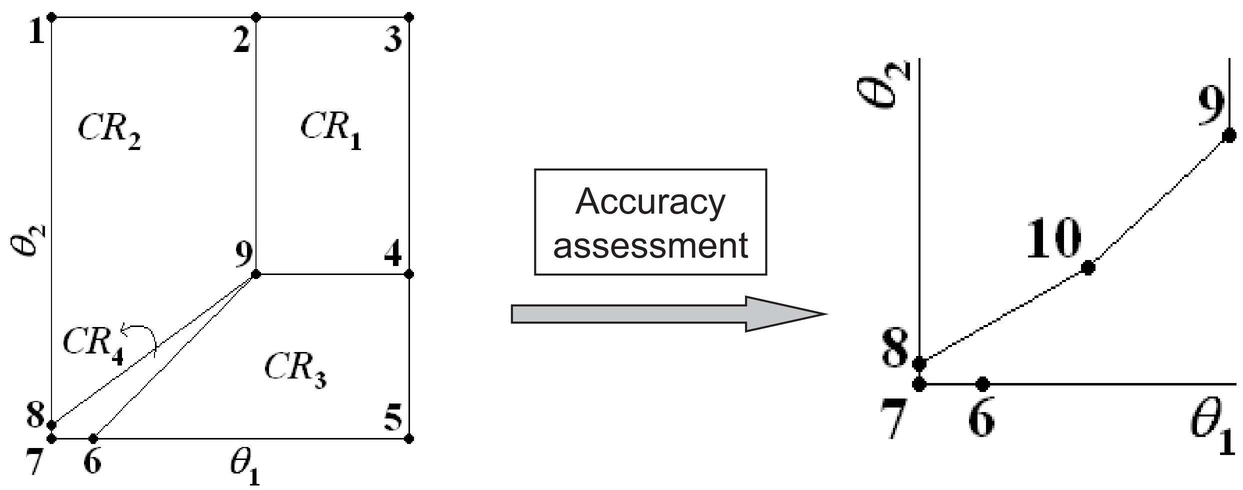
Parameter space (Θ): Found Vertices



Parametric Algorithm

Stage 2: Boundary construction

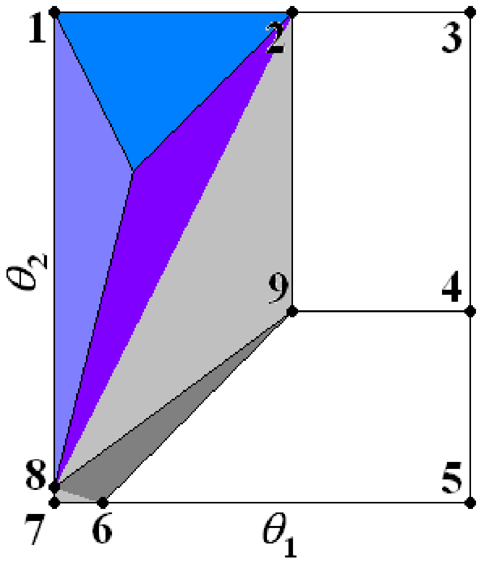
- We take the vertices found for each boundary and create linear approximations, based on the defined solution tree
- The accuracy is assessed



Parametric Algorithm

Stage 3: Parameterization of optimal solutions

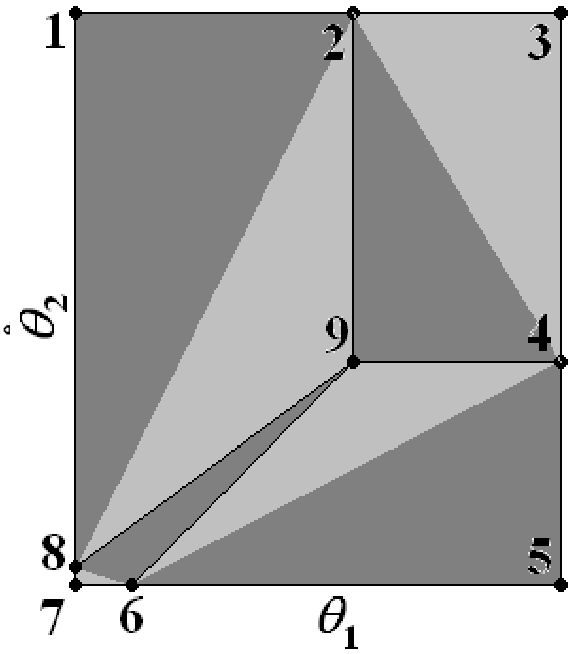
- Convex hulls
- Linear Interpolation
- Assess the error
- Create partitions

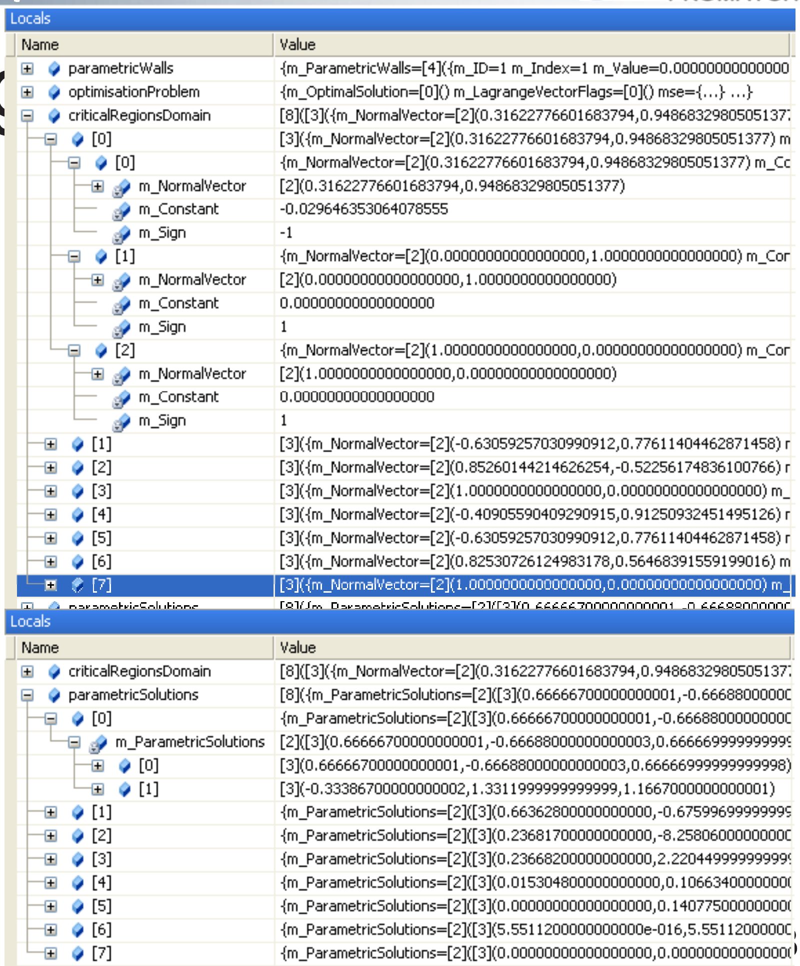


Parametric Alg

Final Solution:

Map of critical regions:





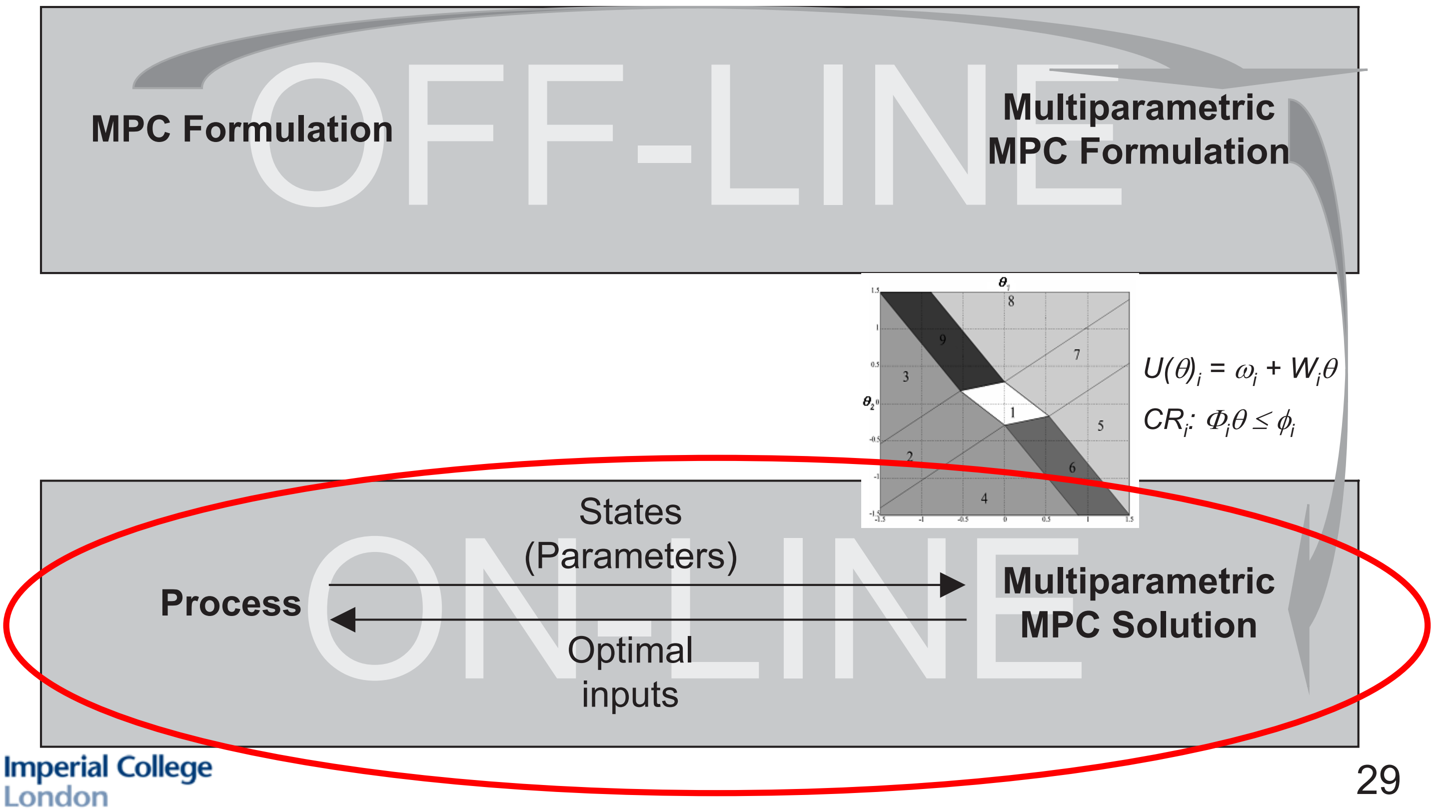
Locals

Name	Value
parametricWalls	{m_ParametricWalls=[4]({m_ID=1 m_Index=1 m_Value=0.00000000000000
optimisationProblem	{m_OptimalSolution=[0]() m_LagrangeVectorFlags=[0]() mse={...} ...}
criticalRegionsDomain	[8]([3]({m_NormalVector=[2](0.31622776601683794,0.9486832980505137
[0]	[3]({m_NormalVector=[2](0.31622776601683794,0.94868329805051377) m
[0]	{m_NormalVector=[2](0.31622776601683794,0.94868329805051377) m_Co
m_NormalVector	[2](0.31622776601683794,0.94868329805051377)
m_Constant	-0.029646353064078555
m_Sign	-1
[1]	{m_NormalVector=[2](0.00000000000000000,1.0000000000000000) m_Cor
m_NormalVector	[2](0.00000000000000000,1.0000000000000000)
m_Constant	0.00000000000000000
m_Sign	1
[2]	{m_NormalVector=[2](1.0000000000000000,0.00000000000000000) m_Cor
m_NormalVector	[2](1.0000000000000000,0.00000000000000000)
m_Constant	0.00000000000000000
m_Sign	1
[1]	[3]({m_NormalVector=[2](-0.63059257030990912,0.77611404462871458) r
[2]	[3]({m_NormalVector=[2](0.85260144214626254,-0.52256174836100766) r
[3]	[3]({m_NormalVector=[2](1.0000000000000000,0.00000000000000000) m_
[4]	[3]({m_NormalVector=[2](-0.40905590409290915,0.91250932451495126) r
[5]	[3]({m_NormalVector=[2](-0.63059257030990912,0.77611404462871458) r
[6]	[3]({m_NormalVector=[2](0.82530726124983178,0.56468391559199016) m
[7]	[3]({m_NormalVector=[2](1.0000000000000000,0.00000000000000000) m_

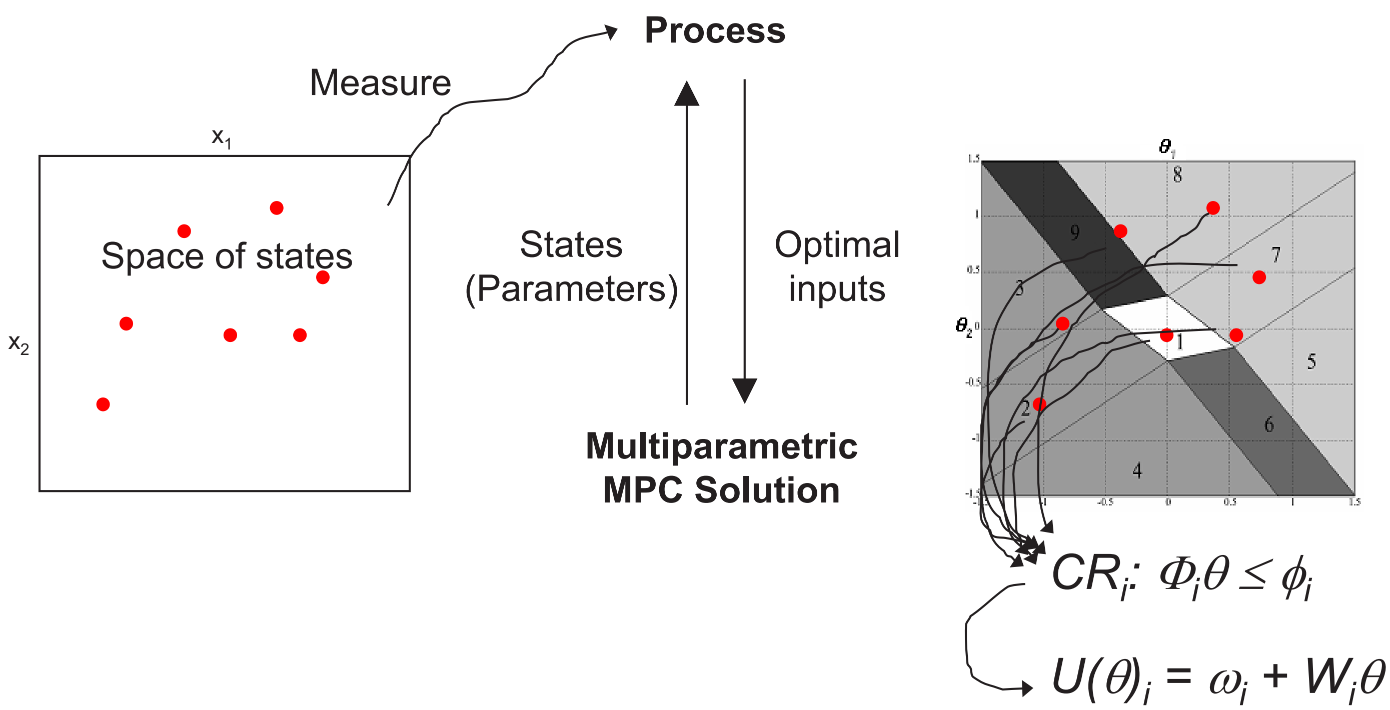
Locals

Name	Value
criticalRegionsDomain	[8]([3]({m_NormalVector=[2](0.31622776601683794,0.9486832980505137
parametricSolutions	[8]({m_ParametricSolutions=[2]([3](0.66666700000000001,-0.66688000000
[0]	{m_ParametricSolutions=[2]([3](0.66666700000000001,-0.66688000000000
m_ParametricSolutions	[2]([3](0.66666700000000001,-0.66688000000000003,0.66666999999999
[0]	[3](0.66666700000000001,-0.66688000000000003,0.66666999999999998)
[1]	[3](-0.33386700000000002,1.3311999999999999,1.1667000000000001)
[1]	{m_ParametricSolutions=[2]([3](0.66362800000000000,-0.67599699999999
[2]	{m_ParametricSolutions=[2]([3](0.23681700000000000,-8.25806000000000
[3]	{m_ParametricSolutions=[2]([3](0.23668200000000000,2.22044999999999
[4]	{m_ParametricSolutions=[2]([3](0.015304800000000000,0.10663400000000
[5]	{m_ParametricSolutions=[2]([3](0.00000000000000000,0.14077500000000
[6]	{m_ParametricSolutions=[2]([3](5.5511200000000000e-016,5.55112000000
[7]	{m_ParametricSolutions=[2]([3](0.00000000000000000,0.00000000000000

On-line strategy

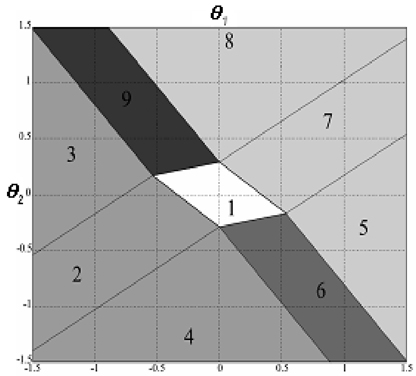


On-line strategy



On-line strategy

- Performs only linear calculations
- Consists of an input-output methodology based on the multiparametric solution



CR_i: $\Phi_i\theta \le \phi_i$

$U(\theta)_i = \omega_i + W_i\theta$

Centre for Process Systems Engineering
PROMATCH

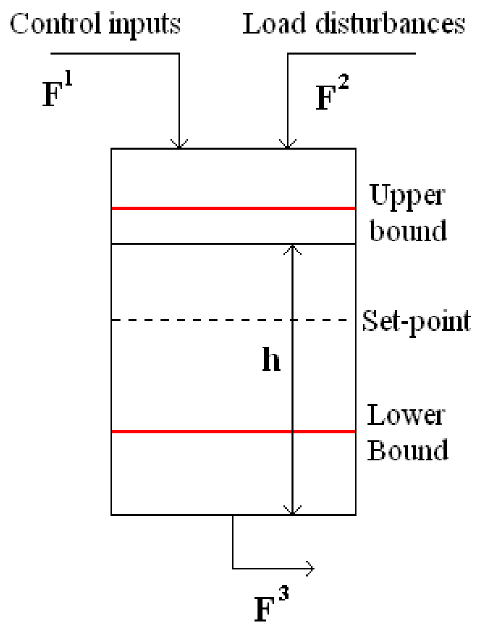
Research and Development Achievements

- Theoretical framework completed
  - On-line strategy
  - Parametric Algorithm
  - Off-line strategy
- Software has been developed (C++/gPROMS)
  - Parametric Solver (Implementation of Algorithm)
  - Off-line strategy

Concluding remarks

- A novel framework for the use of Multiparametric MPC has been developed;
- Theoretical and practical developments have been made for each of its components;
- Novel developments on the Parametric Solver will enable to solve more complex problems;
- Small chemical engineering problem solved;
- Further testing will enable to improve the methodology.

Example



- F^1 – control stream
- F^2 – load disturbance
- F^3 – outlet stream (gravity)
- Problem defined for a control horizon of 3 time units

Example

Problem formulation:

$$\min_{F_1^1, F_2^1, F_3^1} P\left(h_3 - h^{sp}\right)^2 + \sum_{i=0}^{2} Q\left(h_i - h^{sp}\right)^2 + R\left(u_{i+1} - u_i\right)^2$$

$$s.t. \quad h_{l+1} = h_l + D_l.\Delta t, \quad l \geq 0$$

$$D_l = \frac{F_l^1 + F_l^2 - C_D\sqrt{h_l}}{A}, \quad l \geq 0$$

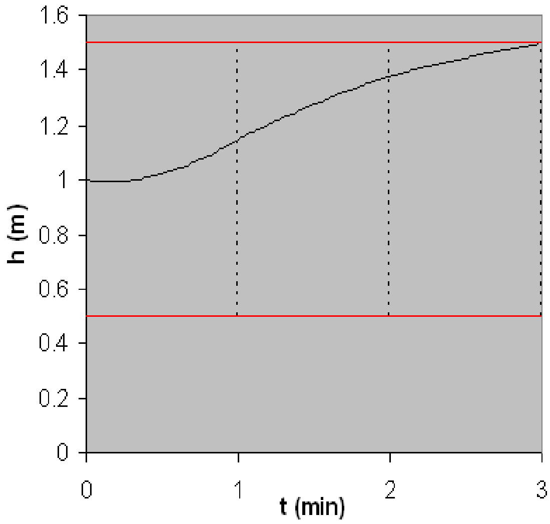
$$h_0 = h_0; \quad u_0 = u_0; \quad \left(F_2^1 \quad F_2^2 \quad F_2^3\right)^T = \left(\theta_1 \quad \theta_2 \quad \theta_3\right)^T;$$

$$h^{\min} \leq h_r \leq h^{\max}, \quad r \geq 1;$$

$$\left(0 \quad 0 \quad 0\right)^T \leq \left(F_1^1 \quad F_1^2 \quad F_1^3\right)^T \leq \left(1.5 \quad 1.5 \quad 1.5\right)^T$$

$$\left(0 \quad 0 \quad 0\right)^T \leq \left(\theta_1 \quad \theta_2 \quad \theta_3\right)^T \leq \left(1 \quad 1 \quad 1\right)^T$$

Example of optimal solution ($(\theta_1\ \theta_2\ \theta_3)^T = (1\ 1\ 1)^T$):

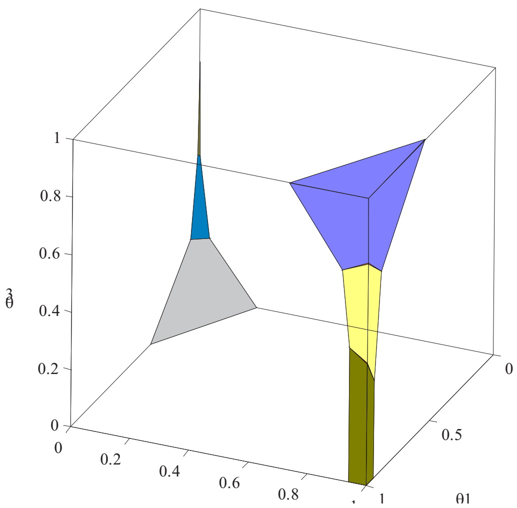


Centre for Process Systems Engineering
PROMATCH

Example

Final Solution:

Map of critical regions:



Parametric solutions for Critical Region 1:

$$\begin{cases} F_1^1 = 1.029 - 0.284\theta_1 - 0.278\theta_2 - 0.010\theta_3 \\ F_2^1 = 1.385 - 0.473\theta_1 - 0.466\theta_2 - 0.019\theta_3 \\ F_3^1 = 1.461 - 0.513\theta_1 - 0.509\theta_2 - 0.025\theta_3 \end{cases}$$

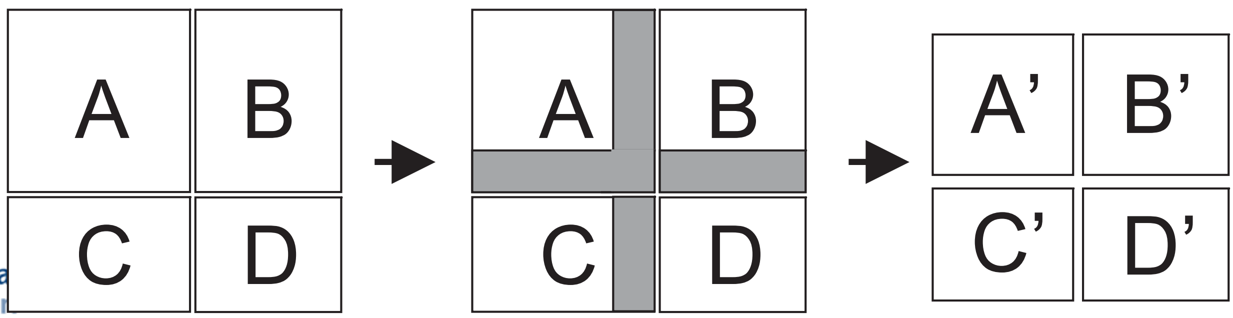
Reduction of the control/optimisation model

- Given a linear state space, we seek to decrease the number of states

Centre for Process Systems Engineering
PROMATCH

Balanced Truncation - fundamentals

- From a dynamics point of view, we want to neglect the states which are harder to reach and harder to observe
- Using a balanced realization, to reduce a system from order *n* to *n-p*, we neglect the last *p* states.



Centre for Process Systems Engineering

PROMATCH

Reduction of the control/optimisation model

- Order of reduction, *p*, is defined (size = *n*)
  - x_1 – first *n-p* components of *x*
  - x_2 – last *p* components of *x*
- Relevant partitions of the states vector and matrices are made

$$\min_U J(U,\bar{x}(t)) = \begin{bmatrix} \bar{x}_1 \\ \hline \bar{x}_2 \end{bmatrix}^T_{t+N_y|t} \left(\begin{array}{c|c} P^b_{11} & P^b_{12} \\ \hline P^b_{21} & P^b_{22} \end{array} \right) \begin{bmatrix} \bar{x}_1 \\ \hline \bar{x}_2 \end{bmatrix}_{t+N_y|t} +$$

$$+\sum_{k=0}^{N_y-1} \begin{bmatrix} \bar{x}_1 \\ \hline \bar{x}_2 \end{bmatrix}^T_{t+k|t} \left(\begin{array}{c|c} Q^b_{11} & Q^b_{12} \\ \hline Q^b_{21} & Q^b_{22} \end{array} \right) \begin{bmatrix} \bar{x}_1 \\ \hline \bar{x}_2 \end{bmatrix}_{t+k|t} + u^T_{t+k} R u_{t+k}$$

$$s.t. \quad y_{min} \le y_{t+k|t} \le y_{max}, \; k = 1, ..., N_c$$

$$u_{min} \le u_{t+k} \le u_{max}, \; k = 0, 1, ..., N_c$$

$$\begin{bmatrix} \bar{x}_1 \\ \hline \bar{x}_2 \end{bmatrix}_{t|t} = \begin{bmatrix} \bar{x}_1(t) \\ \hline \bar{x}_2(t) \end{bmatrix}$$

$$\begin{bmatrix} \bar{x}_1 \\ \hline \bar{x}_2 \end{bmatrix}_{t+k+1|t} = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right) \begin{bmatrix} \bar{x}_1 \\ \hline \bar{x}_2 \end{bmatrix}_{t+k|t} + \left(\begin{array}{c} B_1 \\ \hline B_2 \end{array} \right) u_{t+k}, \; k \ge 0$$

$$y_{t+k|t} = \left(\begin{array}{c|c} C_1 & C_2 \end{array} \right) \begin{bmatrix} \bar{x}_1 \\ \hline \bar{x}_2 \end{bmatrix}, \; k \ge 0$$

$$u_{t+k} = \left(\begin{array}{c|c} K_1 & K_2 \end{array} \right) \begin{bmatrix} \bar{x}_1 \\ \hline \bar{x}_2 \end{bmatrix}, \; N_u \le k \le N_y$$

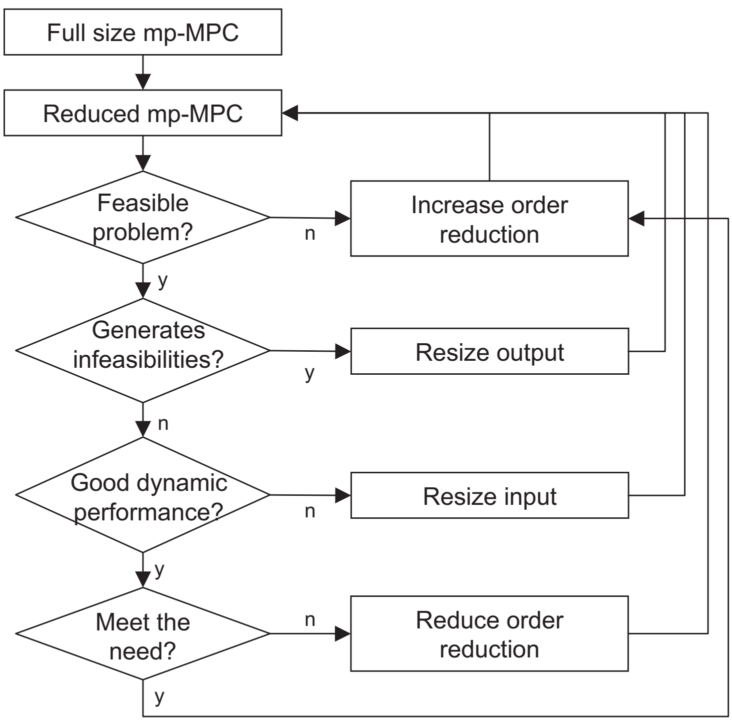
Imperial College London

Centre for Process Systems Engineering

PROMATCH

Combined Balanced Truncation/ mp control – framework

- Iterate with order reduction
- Objective: find the minimum order reduction for which feasibility and optimality are guaranteed

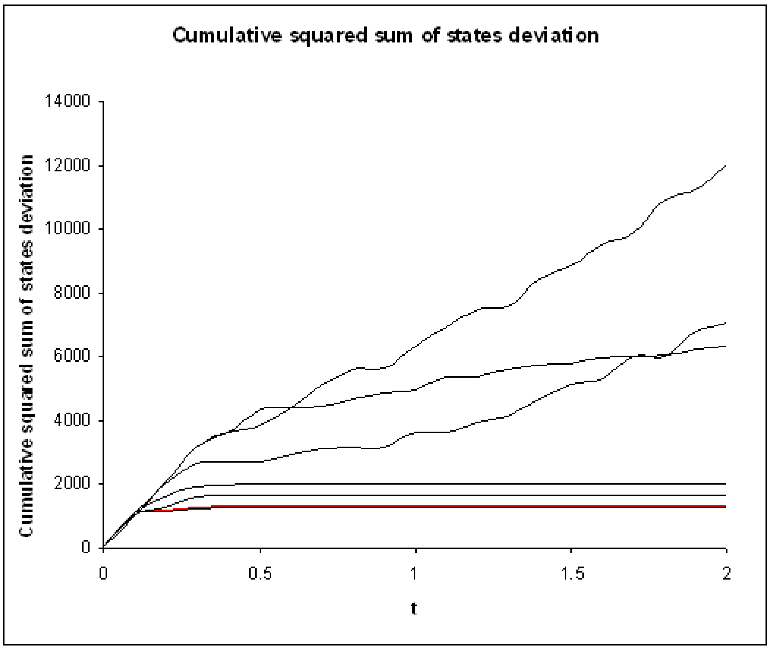


Imperial College London

PROMATCH

Illustrative example

- Full size closed loop response is close to open loop response
- Reasonable performance for small order reductions
- Appropriate control design is key!



Optimal Model Reduction order: from 30 to 20 states