Approximate nonlinear explicit MPC based on reachability analysis

D.M. Raimondo[1] , M. Schulze Darup[2], M. Mönnigmann[2]

Università degli studi di Pavia

Ruhr-Universität Bochum

Introduction: Nonlinear MPC

The system

$$x_{k+1} = f(x_k, u_k) \quad k \geq 0$$

- $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is a $\mathscr{C}^0$ nonlinear function and $f(\bar{x}, \bar{u}) = \bar{x}$
- $x_k \in \mathcal{X}$, $\mathcal{X}$ is a compact set
- $u_k \in \mathcal{U}$, $\mathcal{U}$ is a compact set
- A target set $\mathcal{T} \subseteq \mathcal{X}$

Introduction: Nonlinear MPC

The optimization problem

$$
\begin{aligned}
J^*(x) \quad = \min_{\{u_0,\ldots,u_{N-1}\}} \quad & V_N(x_N) + \sum_{i=0}^{N-1} L(x_i, u_i) \\
\text{subject to} \quad & x_{i+1} = f(x_i, u_i),\ \forall i = 0, \ldots, N-1 \\
& (x_i, u_i) \in \mathcal{X} \times \mathcal{U},\ \forall i = 0, \ldots, N-1 \\
& x_N \in \mathcal{T},\ \ x_0 = x,
\end{aligned}
$$

where $L(x_i, u_i) = (x_i - \bar{x})' Q (x_i - \bar{x}) + (u_i - \bar{u})' R (u_i - \bar{u})$

Introduction: Nonlinear MPC

The optimization problem

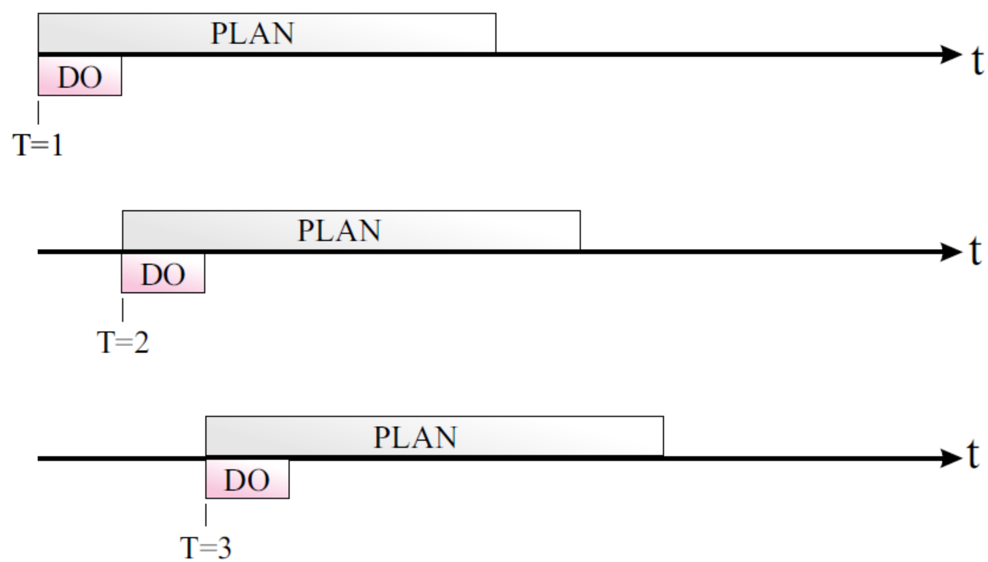
$$
\begin{aligned}
J^*(x) \quad &= \min_{\{u_0,\ldots,u_{N-1}\}} \quad V_N(x_N) + \sum_{i=0}^{N-1} L(x_i, u_i) \\
&\text{subject to} \qquad x_{i+1} = f(x_i, u_i),\ \forall i = 0, \ldots, N-1 \\
&\qquad\qquad\qquad (x_i, u_i) \in \mathcal{X} \times \mathcal{U},\ \forall i = 0, \ldots, N-1 \\
&\qquad\qquad\qquad x_N \in \mathcal{T},\ \ x_0 = x,
\end{aligned}
$$

where $L(x_i, u_i) = (x_i - \bar{x})' Q (x_i - \bar{x}) + (u_i - \bar{u})' R (u_i - \bar{u})$

Given a stabilizing control law $\kappa_f(x)$ defined in $\mathcal{T}$, V_N is a Lyapunov function and $\mathcal{T}$ is a positively invariant set.

Introduction: Nonlinear MPC

The Receding Horizon approach



Closed-loop control law $k(x) = u_o^*(x)$

Objectives

Controller requirements

- Fast online computation
- Suitable for inexpensive hardware
- Feasibility and stability guarantees

Objectives

Controller requirements

- Fast online computation
- Suitable for inexpensive hardware
- Feasibility and stability guarantees

Explicit MPC

Explicit MPC

Linear Explicit MPC

- multi-parametric program can be solved exactly
- feasible set is closed and convex
- continuous polyhedral piecewise affine control law

Explicit MPC

Linear Explicit MPC

- multi-parametric program can be solved exactly
- feasible set is closed and convex
- continuous polyhedral piecewise affine control law

Drawbacks?

Explicit MPC

Complexity

- Strongest dependence on the number of constraints
- Strong dependence on the number of free moves
- Weak dependence on the number of states

Approximate Explicit MPC

One approach to alleviate complexity (will be used in the nonlinear case..)

Solve iteratively *1-step optimization problems* with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$
\begin{array}{ll}
\min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\
\text{subject to} & x_1 = f(x_0, u_0) \\
 & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U}
\end{array}
$$

where $\mathcal{T}_i = \begin{cases} \mathcal{T} & i = 0 \\ \mathcal{Q}(\mathcal{T}_{i-1}) & i = 1, \dots, N-1 \end{cases}$

Approximate Explicit MPC

One approach to alleviate complexity (will be used in the nonlinear case..)

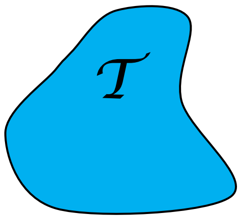
Solve iteratively *1-step optimization problems* with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$
\begin{array}{ll}
\min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\
\text{subject to} & x_1 = f(x_0, u_0) \\
 & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U}
\end{array}
$$

One-step set

Set of states which can be steered to a target set within one time step (Bertsekas, 1971)

$$\mathcal{Q}(\mathcal{T}) = \{x \in \mathcal{X} \,|\, \exists u \in \mathcal{U} : f(x,u) \in \mathcal{T}\}$$



Approximate Explicit MPC

One approach to alleviate complexity (will be used in the nonlinear case..)

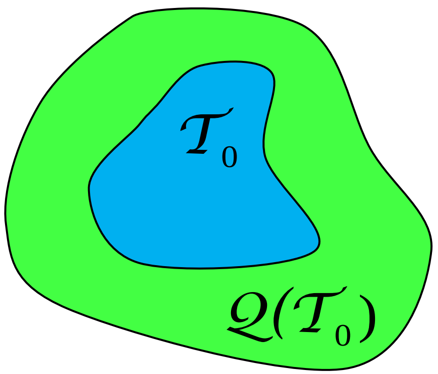
Solve iteratively *1-step optimization problems* with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$\begin{array}{ll} \min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\ \text{subject to} & x_1 = f(x_0, u_0) \\ & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U} \end{array}$$

One-step set

Set of states which can be steered to a target set within one time step (Bertsekas, 1971)

$$\mathcal{Q}(\mathcal{T}) = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U} : f(x,u) \in \mathcal{T}\}$$



Approximate Explicit MPC

One approach to alleviate complexity (will be used in the nonlinear case..)

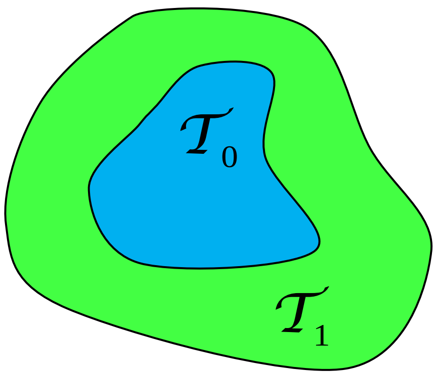
Solve iteratively *1-step optimization problems* with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$\begin{array}{ll} \min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\ \text{subject to} & x_1 = f(x_0, u_0) \\ & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U} \end{array}$$

One-step set

Set of states which can be steered to a target set within one time step (Bertsekas, 1971)

$$\mathcal{Q}(\mathcal{T}) = \{x \in \mathcal{X} \,|\, \exists u \in \mathcal{U} : f(x,u) \in \mathcal{T}\}$$



Approximate Explicit MPC

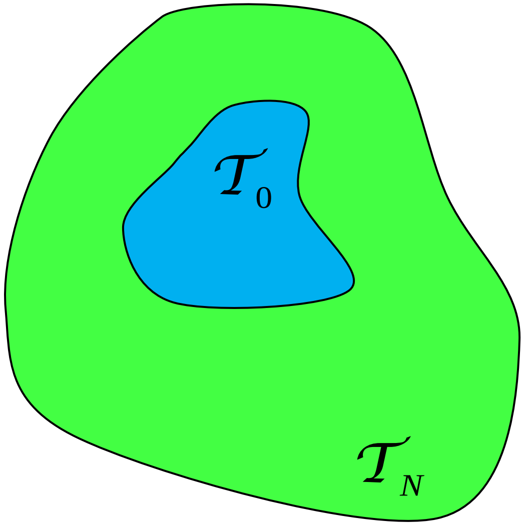
One approach to alleviate complexity (will be used in the nonlinear case..)

Solve iteratively *1-step optimization problems* with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$
\begin{array}{ll}
\min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\
\text{subject to} & x_1 = f(x_0, u_0) \\
 & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U}
\end{array}
$$

Feasible set of the original OP

$$\mathcal{F}_N(\mathcal{T}) = \underbrace{\mathcal{Q}(\mathcal{Q}(\ldots \mathcal{Q}(\mathcal{Q}(\mathcal{T}))))}_{N \text{ times}}$$



Approximate Explicit MPC

One approach to alleviate complexity (will be used in the nonlinear case..)

Solve iteratively *1-step optimization problems* with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$\begin{array}{ll} \min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\ \text{subject to} & x_1 = f(x_0, u_0) \\ & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U} \end{array}$$

Online: since the regions of the 1-step multiparametric programs may overlap apply the *feedback control* computed at the smallest iteration number *i.*

i=0

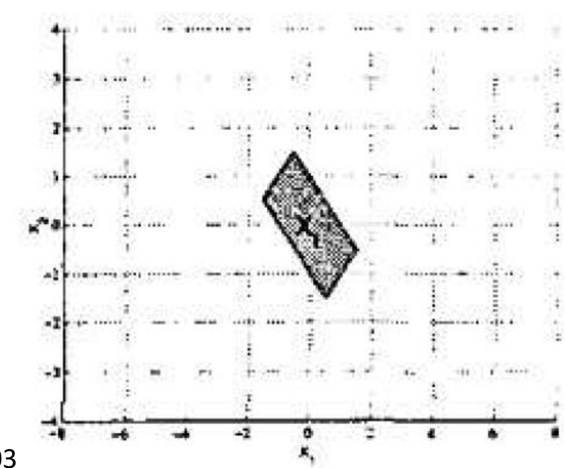


Image from P. Grieder and M. Morari CDC 2003

Approximate Explicit MPC

One approach to alleviate complexity (will be used in the nonlinear case..)

Solve iteratively 1-step optimization problems with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$\begin{array}{ll} \min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\ \text{subject to} & x_1 = f(x_0, u_0) \\ & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U} \end{array}$$

Online: since the regions of the 1-step multiparametric programs may overlap apply the *feedback control* computed at the smallest iteration number *i.*

i=1

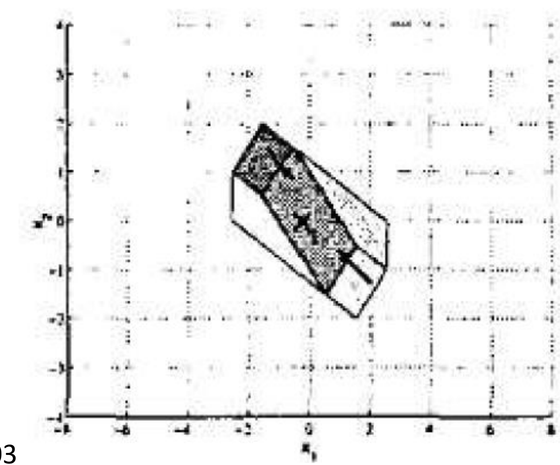


Image from P. Grieder and M. Morari CDC 2003

Approximate Explicit MPC

One approach to alleviate complexity (will be used in the nonlinear case..)

Solve iteratively 1-step optimization problems with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$\begin{array}{ll} \min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\ \text{subject to} & x_1 = f(x_0, u_0) \\ & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U} \end{array}$$

Online: since the regions of the 1-step multiparametric programs may overlap apply the *feedback control* computed at the smallest iteration number *i.*

i=2

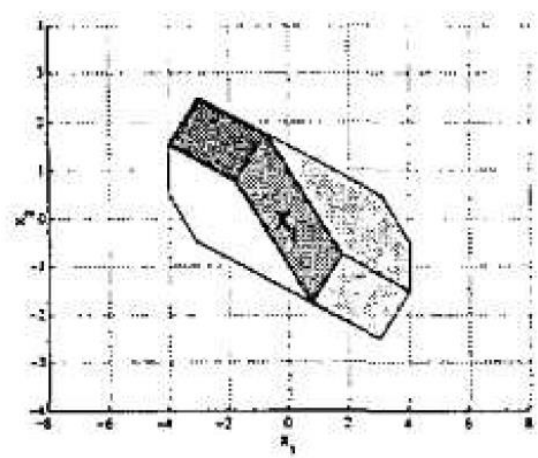


Image from P. Grieder and M. Morari CDC 2003

Approximate Explicit MPC

One approach to alleviate complexity (will be used in the nonlinear case..)

Solve iteratively 1-step optimization problems with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$\begin{array}{ll} \min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\ \text{subject to} & x_1 = f(x_0, u_0) \\ & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U} \end{array}$$

Online: since the regions of the 1-step multiparametric programs may overlap apply the *feedback control* computed at the smallest iteration number *i.*

i=3

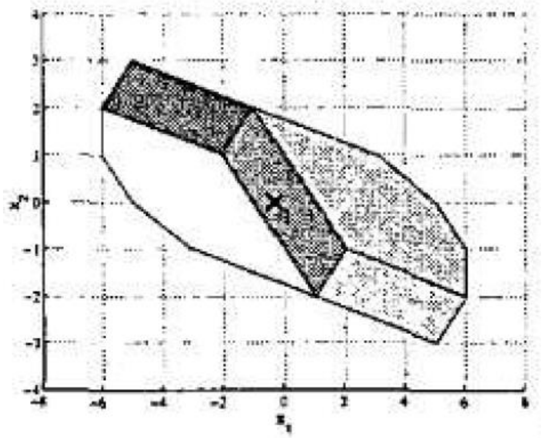


Image from P. Grieder and M. Morari CDC 2003

Approximate Explicit MPC

One approach to alleviate complexity (will be used in the nonlinear case..)

Solve iteratively 1-step optimization problems with varying terminal set constraint (P. Grieder and M. Morari 2003)

$$\begin{array}{ll} \min_{u_0} & x_1' Q x_1 + u_0' R u_0 \\ \text{subject to} & x_1 = f(x_0, u_0) \\ & x_0 \in \mathcal{X}, x_1 \in \mathcal{T}_i, u_0 \in \mathcal{U} \end{array}$$

Online: since the regions of the 1-step multiparametric programs may overlap apply the *feedback control* computed at the smallest iteration number *i.*

Guaranteed feasibility and stability of the approach

Approximate Explicit NMPC

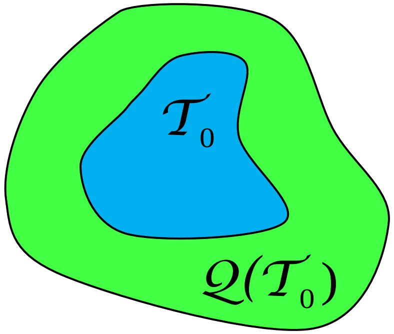
Making use of the previous approach in the *nonlinear* case

- Solving mp-NLP is difficult
- Exact solutions cannot be found in the general nonlinear case
- Feasible set non convex in general

Approximate Explicit NMPC

One-step set in the nonlinear case

$$\mathcal{Q}(\mathcal{T}) = \{x \in \mathcal{X} \,|\, \exists u \in \mathcal{U} : f(x,u) \in \mathcal{T}\}$$

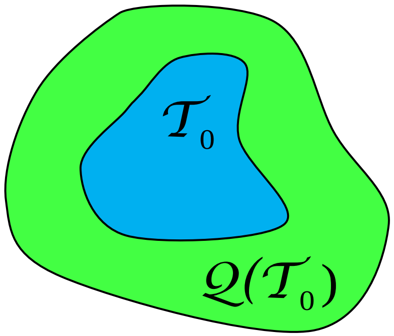


- The exact computation of this set is not possible in general
- The objective is to obtain a tight inner approximation

Approximate Explicit NMPC

One-step set in the nonlinear case

$$\mathcal{Q}(\mathcal{T}) = \{x \in \mathcal{X} \,|\, \exists u \in \mathcal{U} : f(x,u) \in \mathcal{T}\}$$



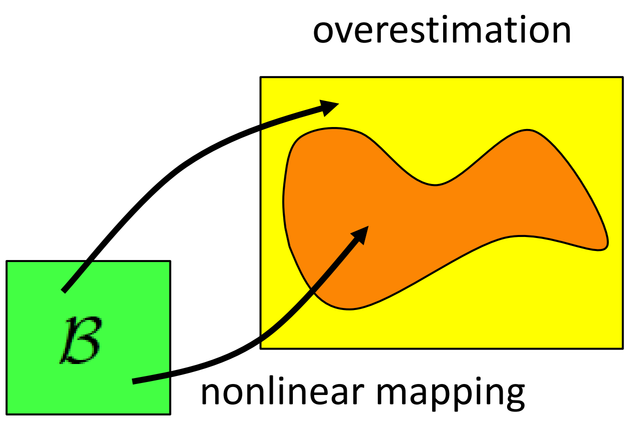
- The exact computation of this set is not possible in general
- The objective is to obtain a tight inner approximation

Reachability analysis

- Overestimate one-step ahead reachable set for

$$\{f(x,u(x)) | x \in \mathcal{B},\, u(x) \in \mathcal{U}\}$$

- *Methods:* interval-arithmetic, DC-programming

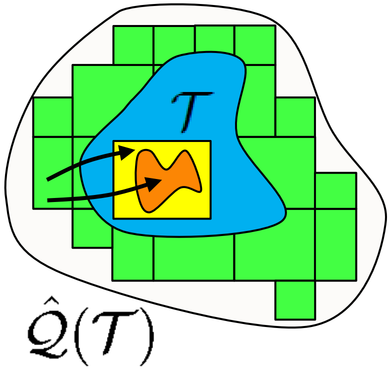


Approximate Explicit NMPC

One-step set in the nonlinear case

$$\mathcal{Q}(\mathcal{T}) = \{x \in \mathcal{X} \,|\, \exists u \in \mathcal{U} : f(x,u) \in \mathcal{T}\}$$

- The exact computation of this set is not possible in general
- The objective is to obtain a tight inner approximation

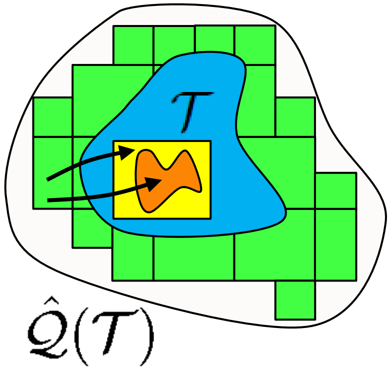


Approximate Explicit NMPC

One-step set in the nonlinear case

$$\mathcal{Q}(\mathcal{T}) = \{x \in \mathcal{X} \,|\, \exists u \in \mathcal{U} : f(x,u) \in \mathcal{T}\}$$

- The exact computation of this set is not possible in general
- The objective is to obtain a tight inner approximation

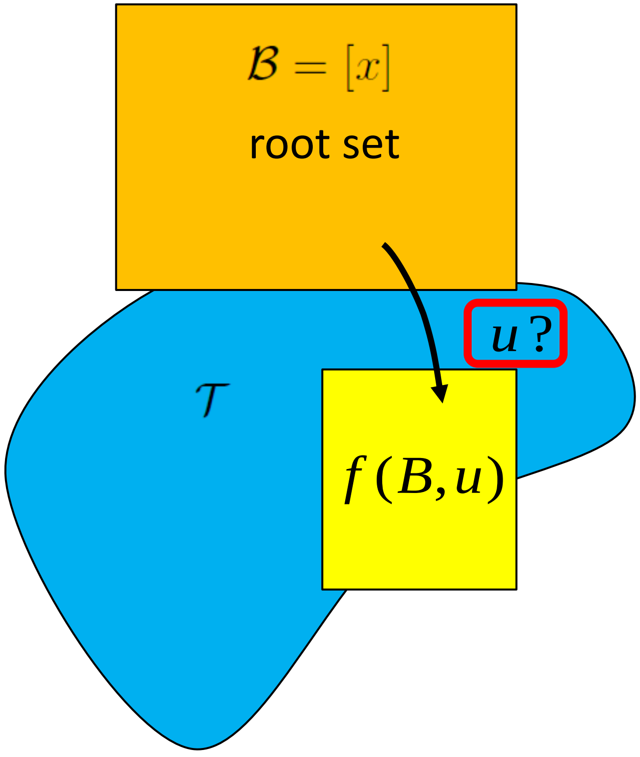


Partitions

- Choose a partitioning that results in fast online computation
- Hyperrectangles + binary tree structure

Bisection, binary tree

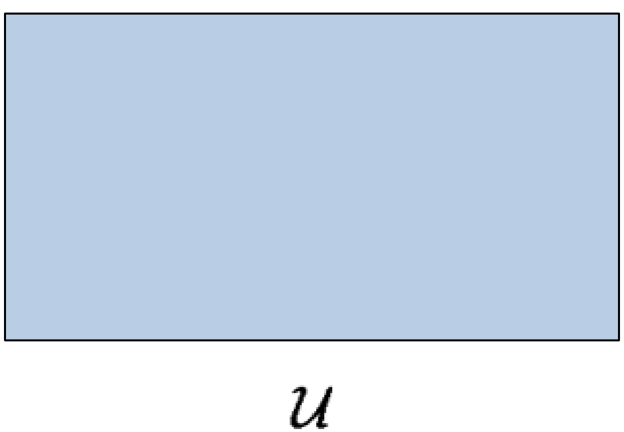
How to choose u(x) for each hyperrectangle?

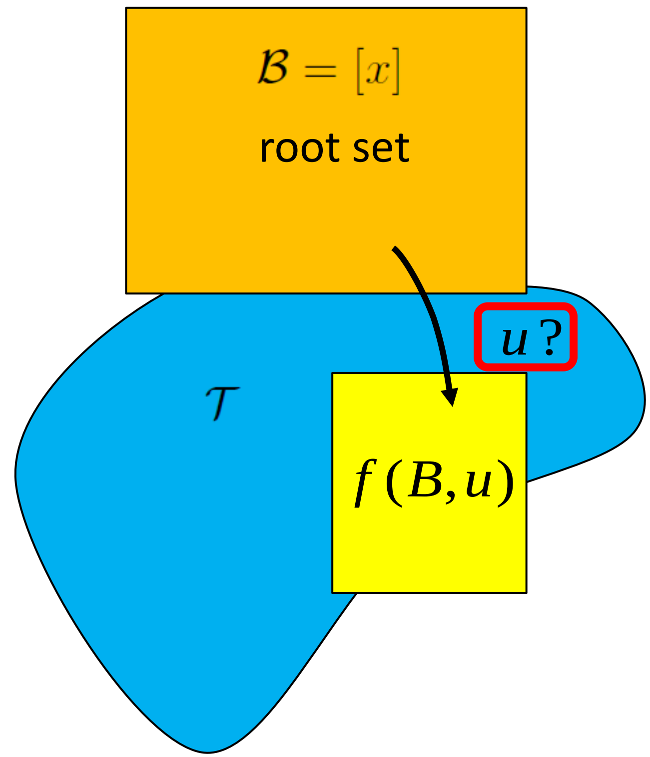


Bisection, binary tree

How to choose u(x) for each hyperrectangle?

- Entire *U* set

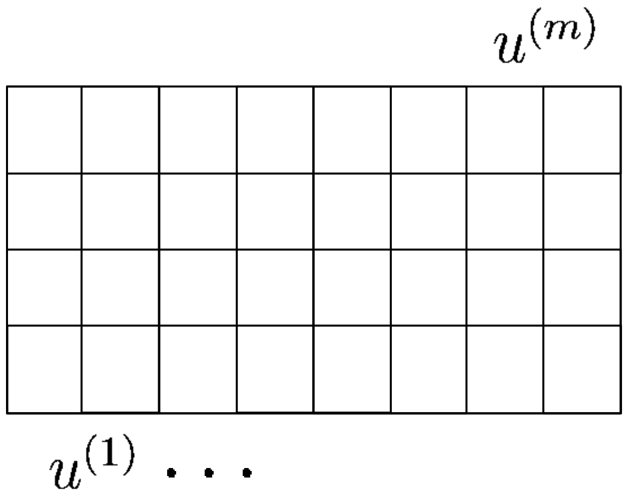


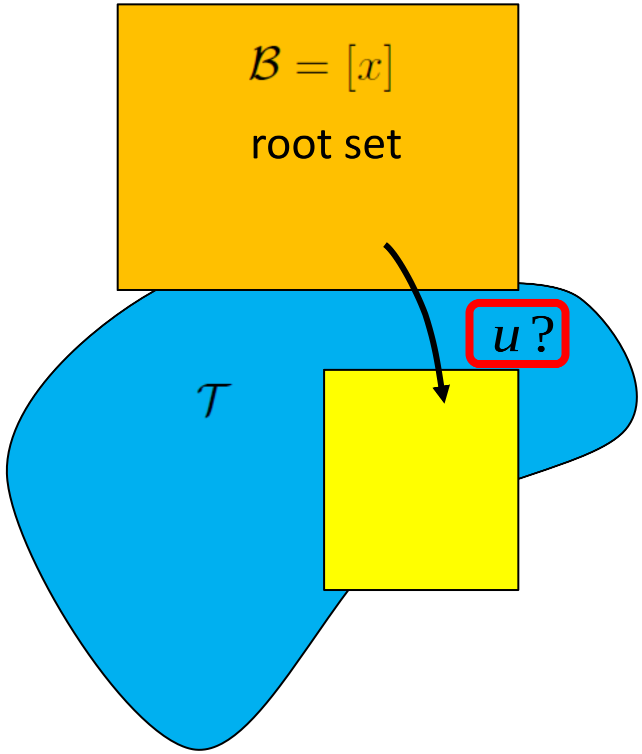


Bisection, binary tree

How to choose u(x) for each hyperrectangle?

- Set of hyperrectangles

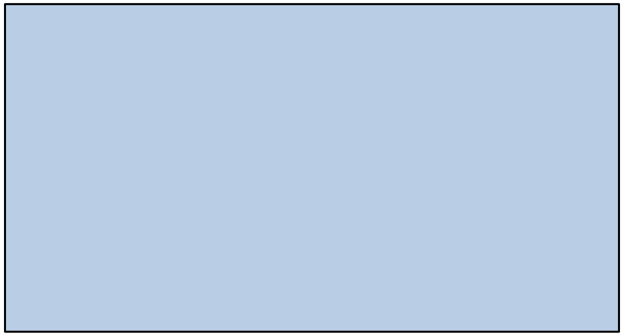


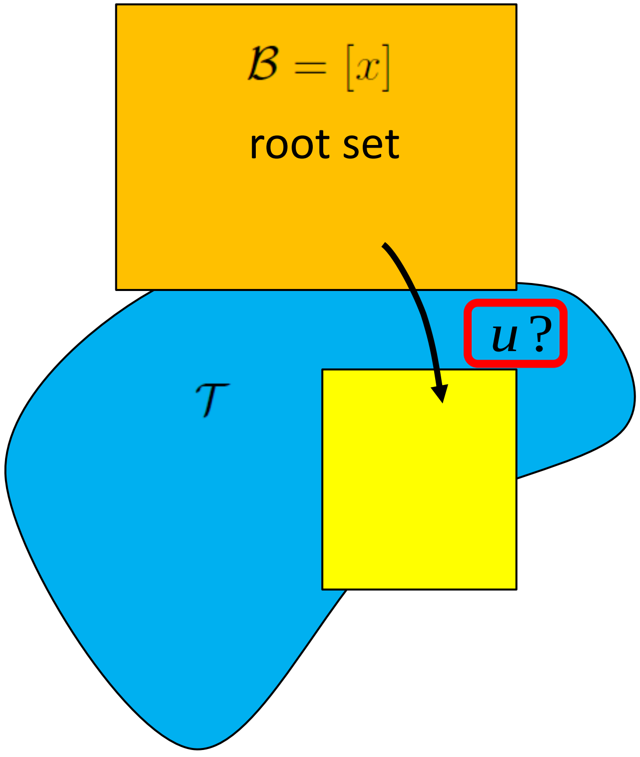


Bisection, binary tree

How to choose u(x) for each hyperrectangle?

- Bisect the input space

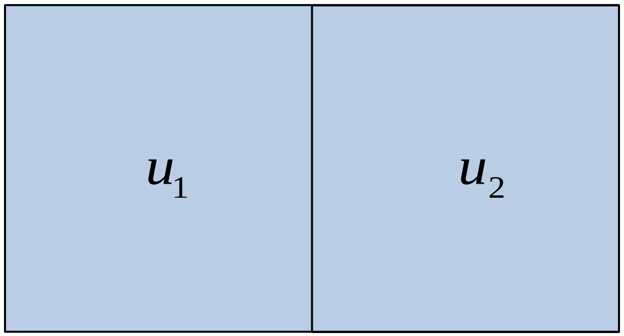


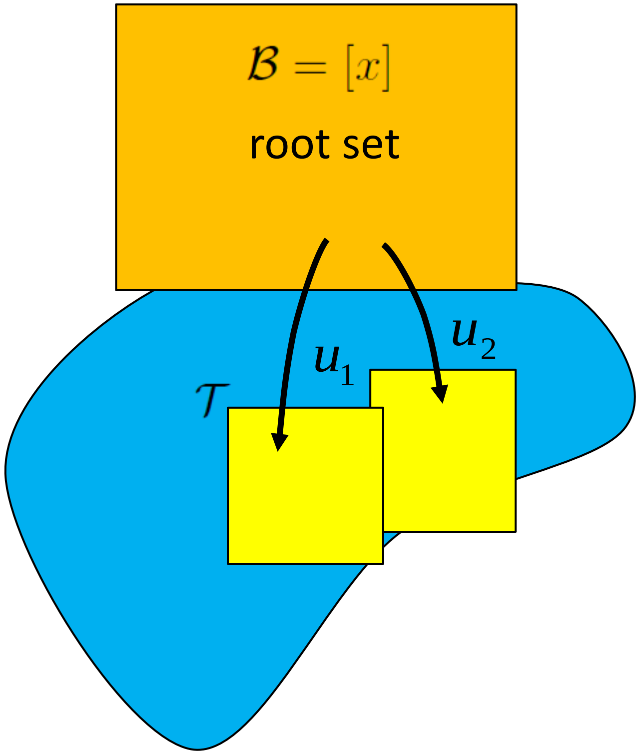


Bisection, binary tree

How to choose u(x) for each hyperrectangle?

- Bisect the input space

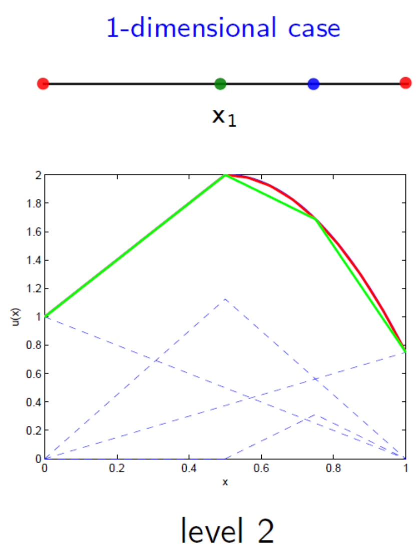




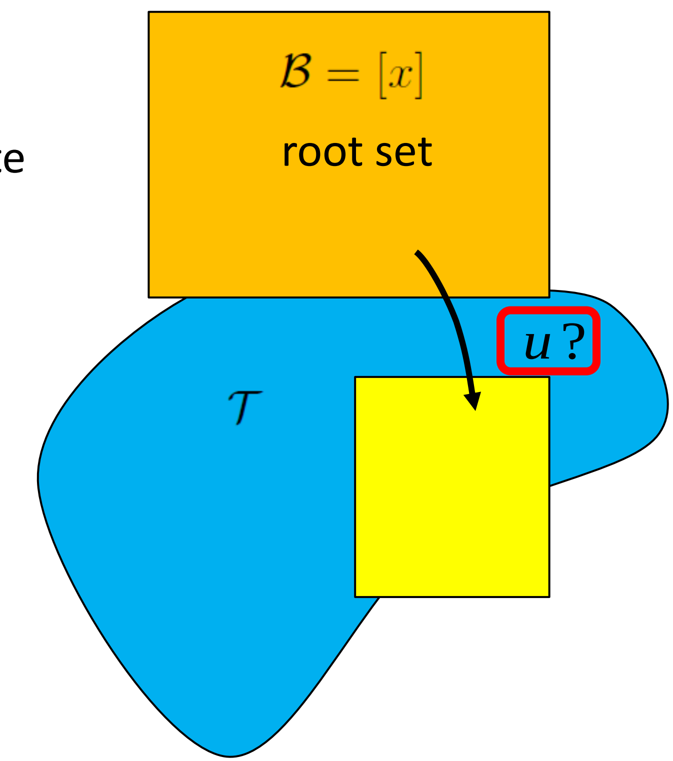
Bisection, binary tree

How to choose u(x) for each hyperrectangle?

- Solve the NLP at the vertices and interpolate



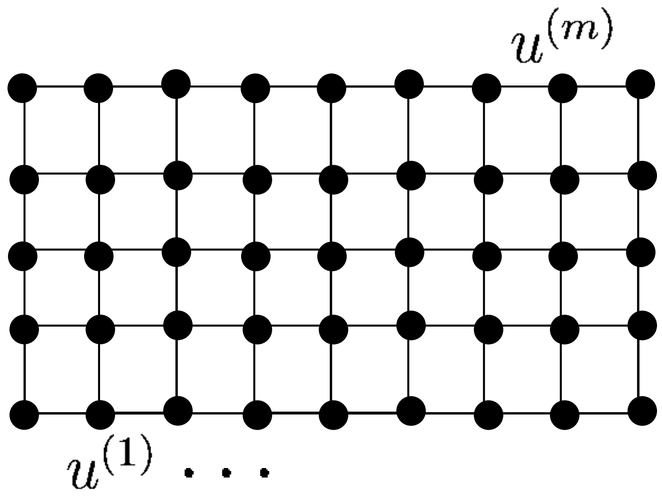
e.g. Hierarchical function approximation with interpolets

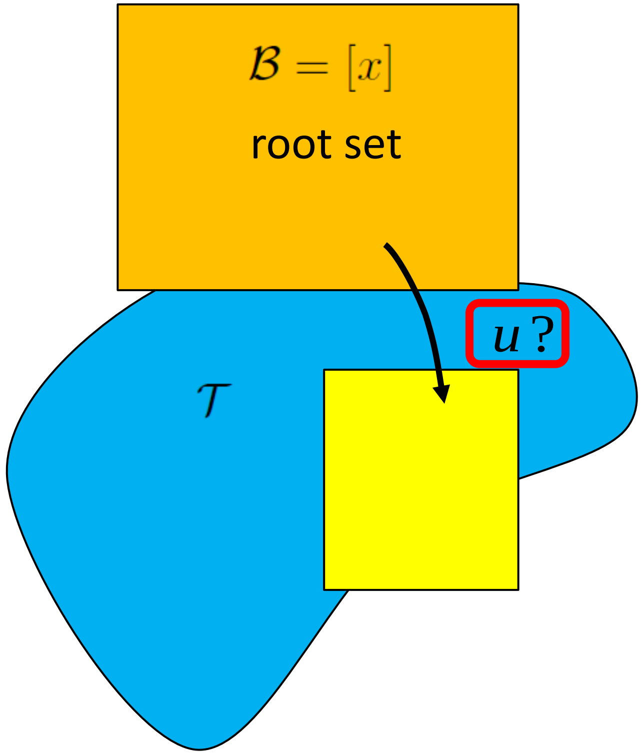


Bisection, binary tree

How to choose u(x) for each hyperrectangle?

- Input space gridding

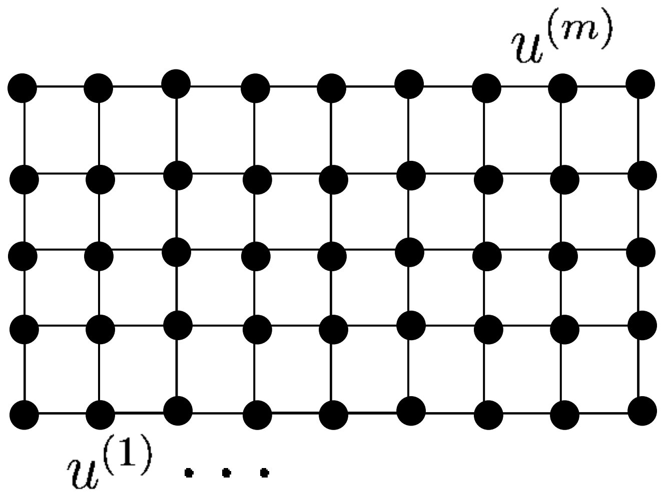


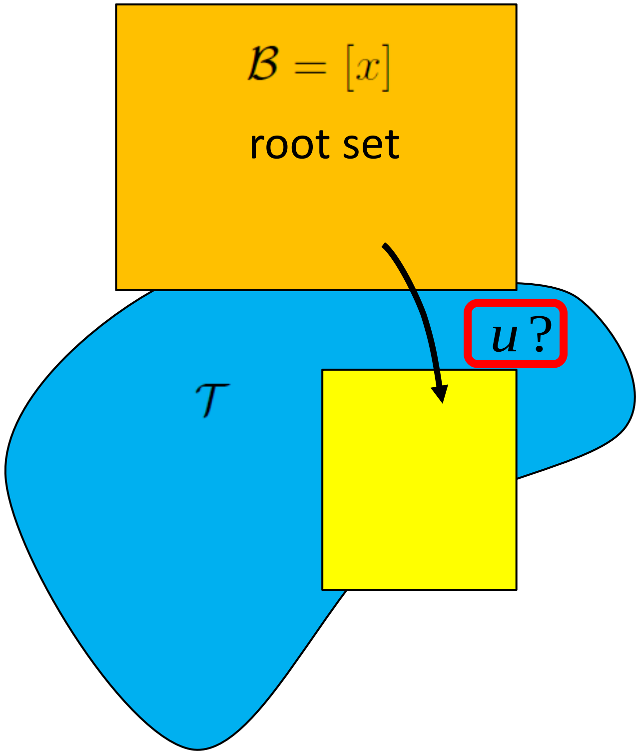


Bisection, binary tree

How to choose u(x) for each hyperrectangle?

- Input space gridding

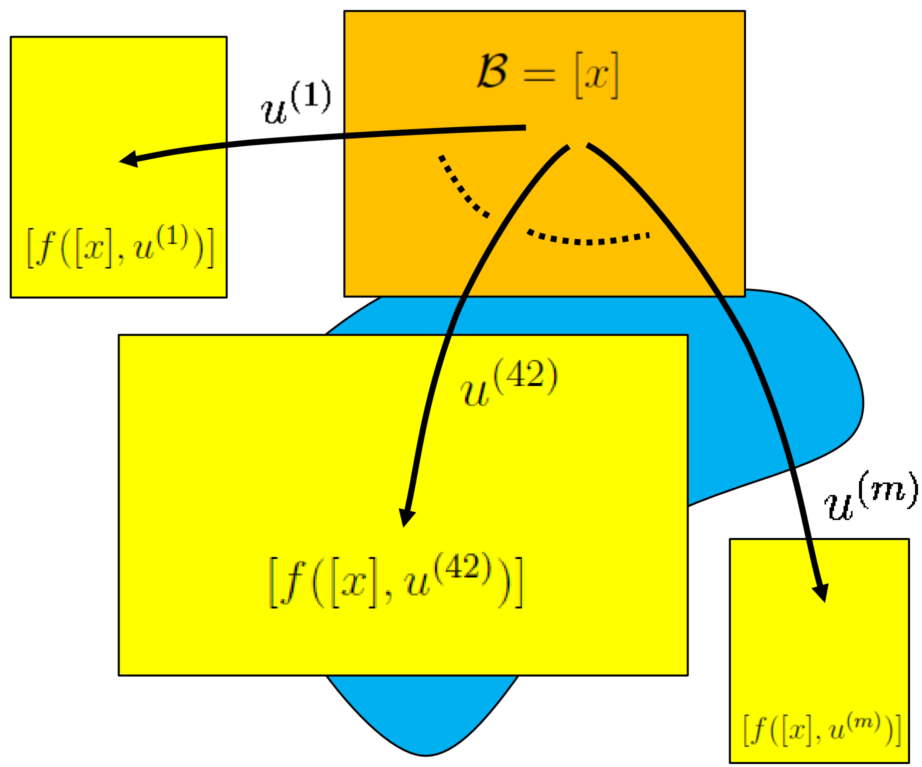


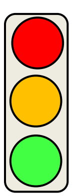


Bisection, binary tree

How to organize hyperrectangular state space representation?

- Use bisection and binary tree





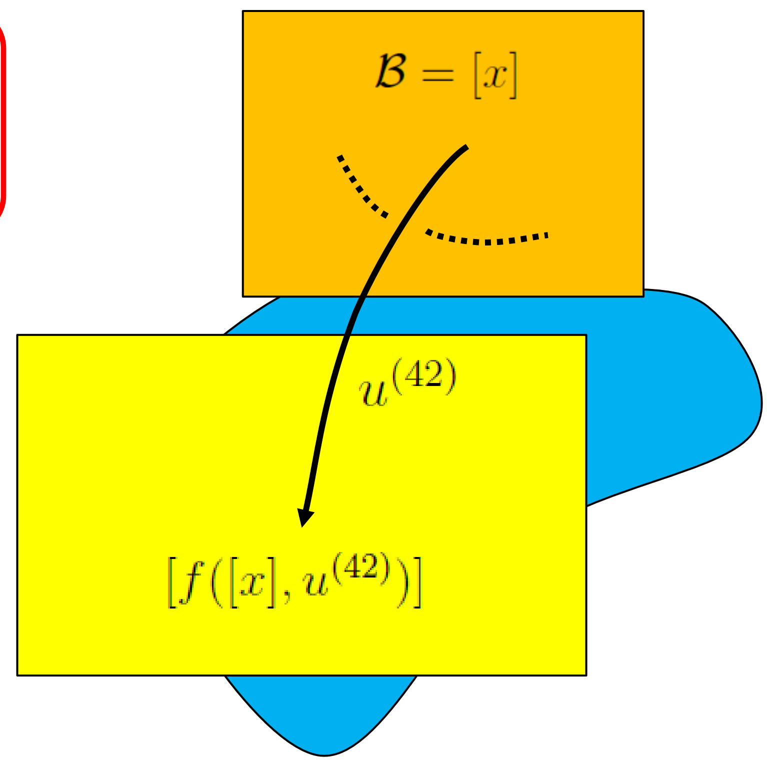
can not be steered towards $\mathcal{T}$
uncertain
can be steered towards $\mathcal{T}$

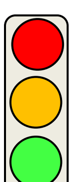
Bisection, binary tree

How to organize hyperrectangular state space representation?

- Use bisection and binary tree

If $f(\mathcal{B}, u)$ intersects the target set
bisect $\mathcal{B}$ to find the subsets entering $\mathcal{T}$



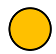


can not be steered towards $\mathcal{T}$
uncertain
can be steered towards $\mathcal{T}$

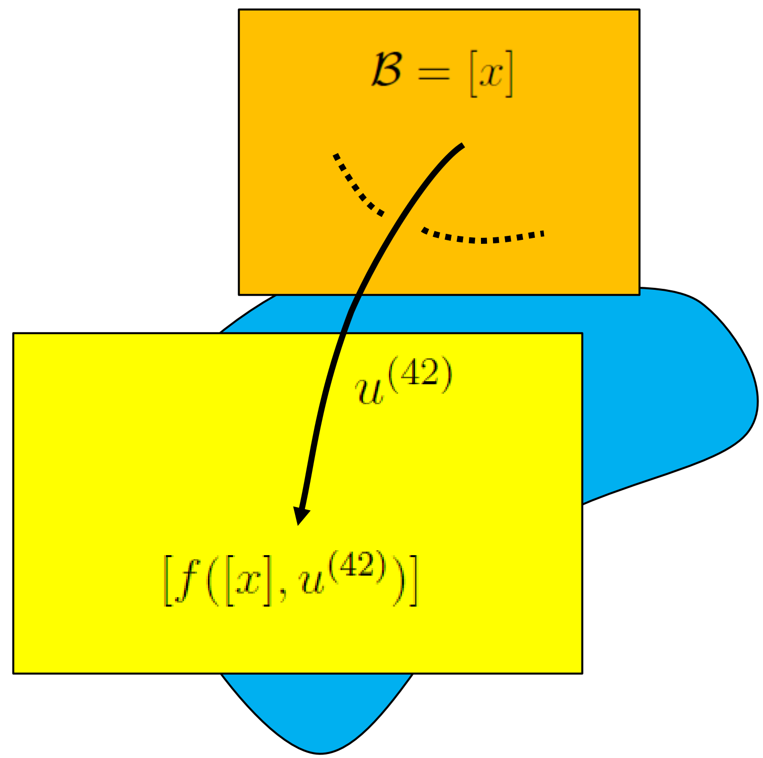
Bisection, binary tree

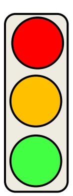
How to organize hyperrectangular state space representation?

- Use bisection and binary tree



root node



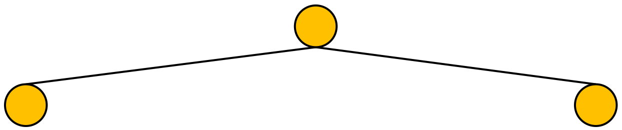


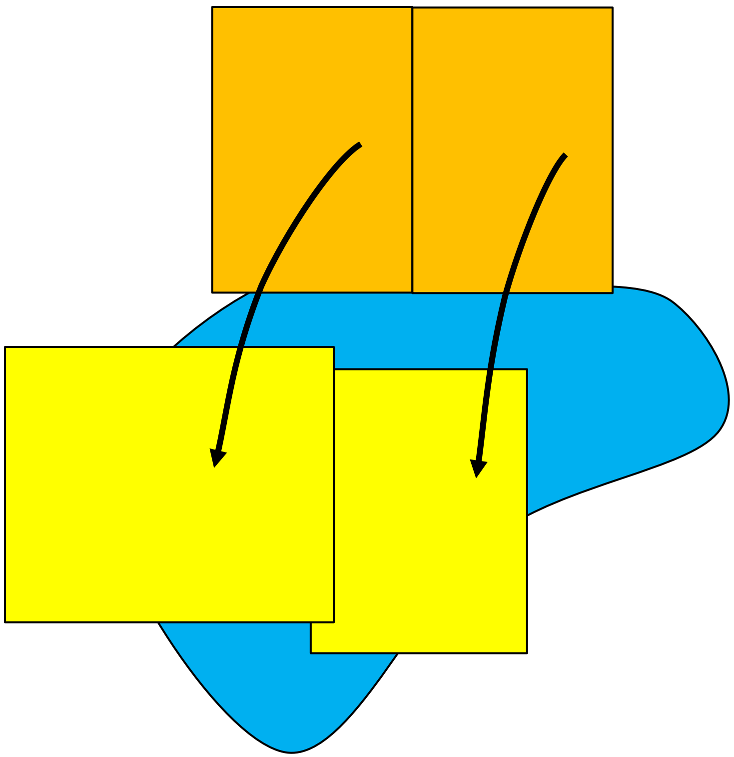
can not be steered towards $\mathcal{T}$
uncertain
can be steered towards $\mathcal{T}$

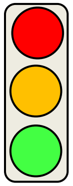
Bisection, binary tree

How to organize hyperrectangular state space representation?

- Use bisection and binary tree





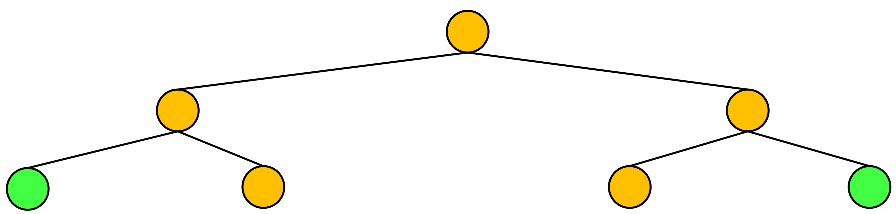


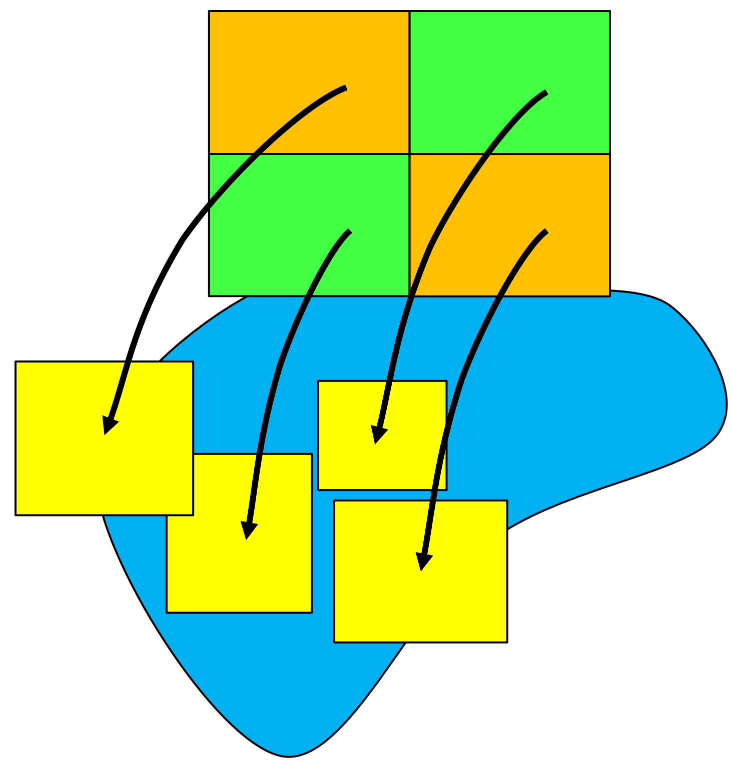
can not be steered towards $\mathcal{T}$
uncertain
can be steered towards $\mathcal{T}$

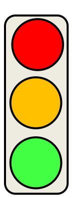
Bisection, binary tree

How to organize hyperrectangular state space representation?

- Use bisection and binary tree







can not be steered towards $\mathcal{T}$

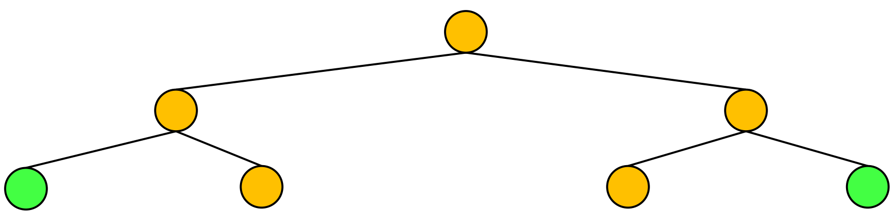
uncertain

can be steered towards $\mathcal{T}$

Bisection, binary tree

How to organize hyperrectangular state space representation?

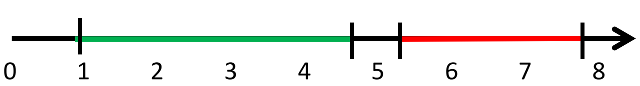
- Use bisection and binary tree

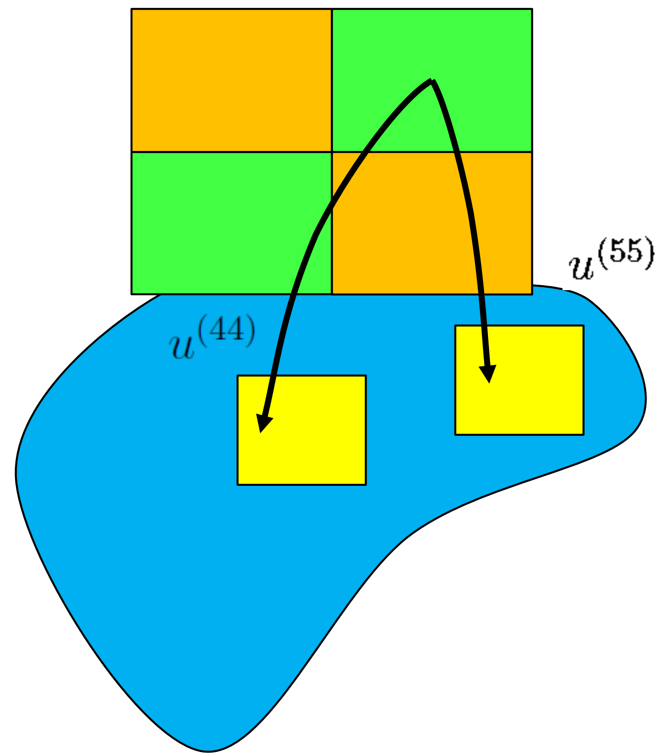


Which u do we choose?

The one that minimizes the upper bound of the interval

$$x_1' Q x_1 + u_0' R u_0$$

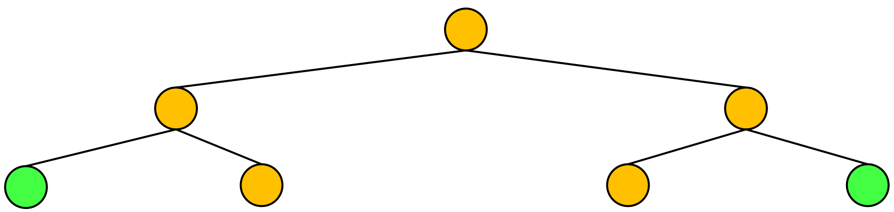


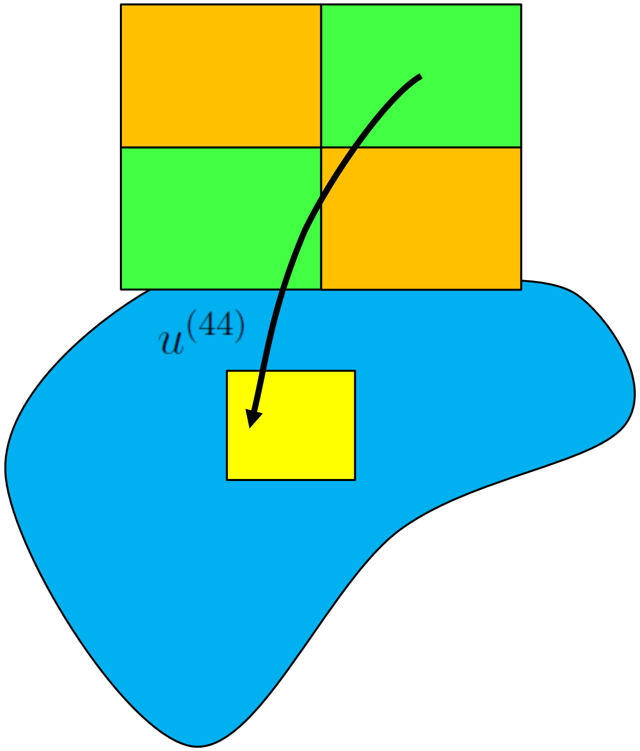


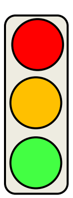
Bisection, binary tree

How to organize hyperrectangular state space representation?

- Use bisection and binary tree







can not be steered towards $\mathcal{T}$

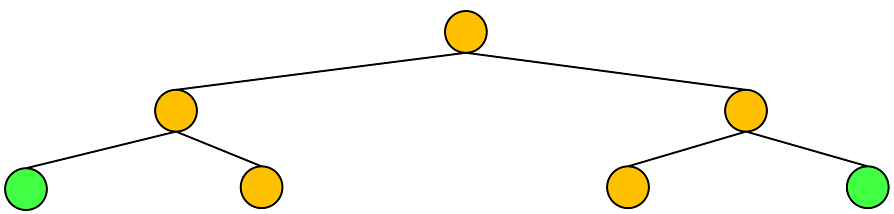
uncertain

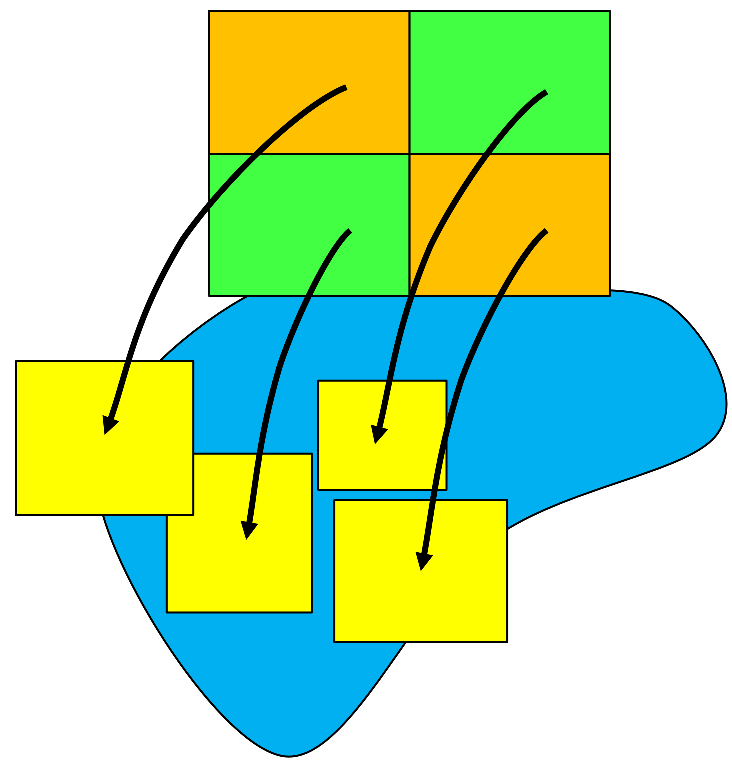
can be steered towards $\mathcal{T}$

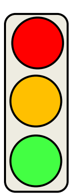
Bisection, binary tree

How to organize hyperrectangular state space representation?

- Use bisection and binary tree







can not be steered towards $\mathcal{T}$

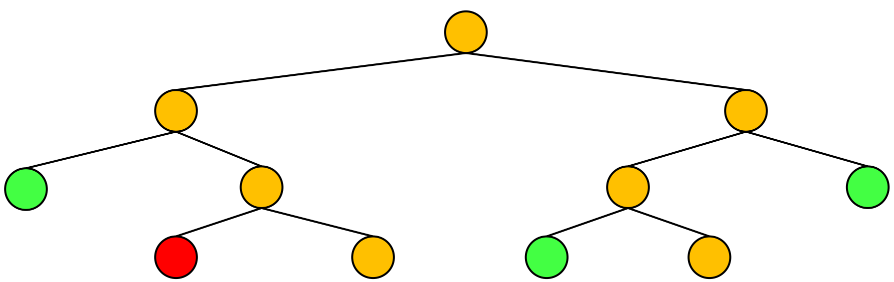
uncertain

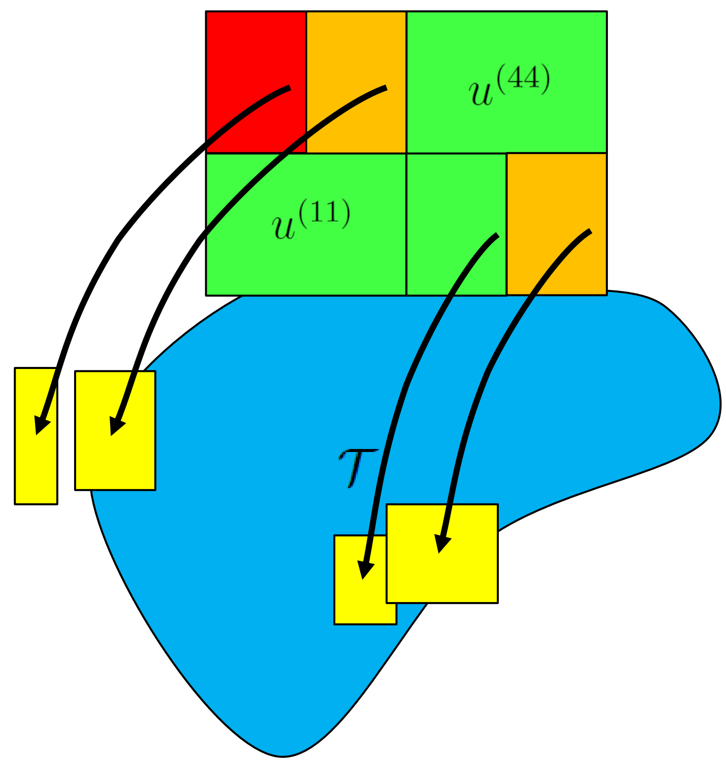
can be steered towards $\mathcal{T}$

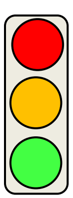
Bisection, binary tree

How to organize hyperrectangular state space representation?

- Use bisection and binary tree





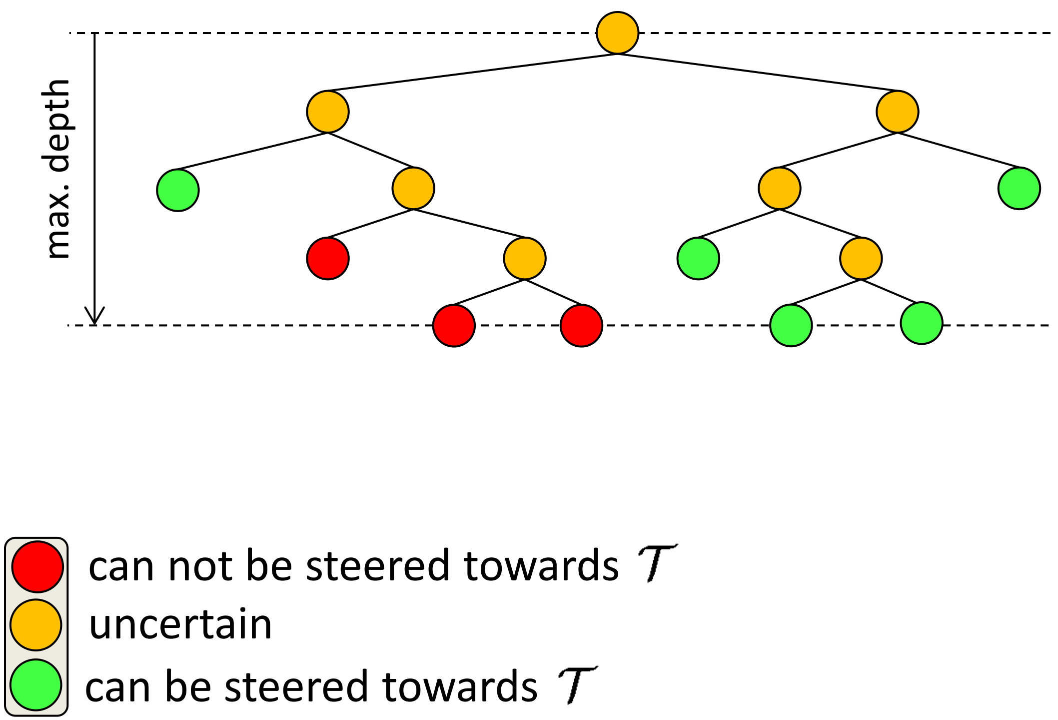


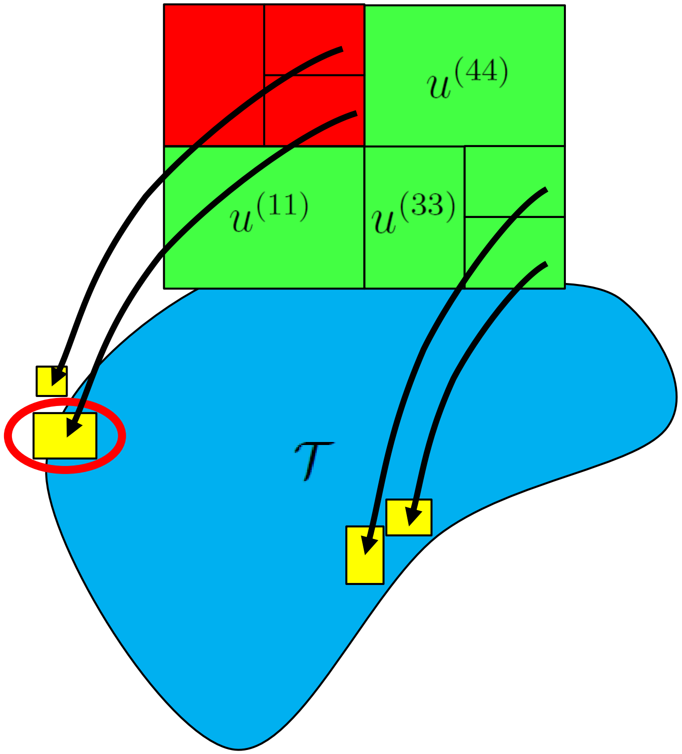
can not be steered towards $\mathcal{T}$
uncertain
can be steered towards $\mathcal{T}$

Bisection, binary tree

How to organize hyperrectangular state space representation?

- Use bisection and binary tree

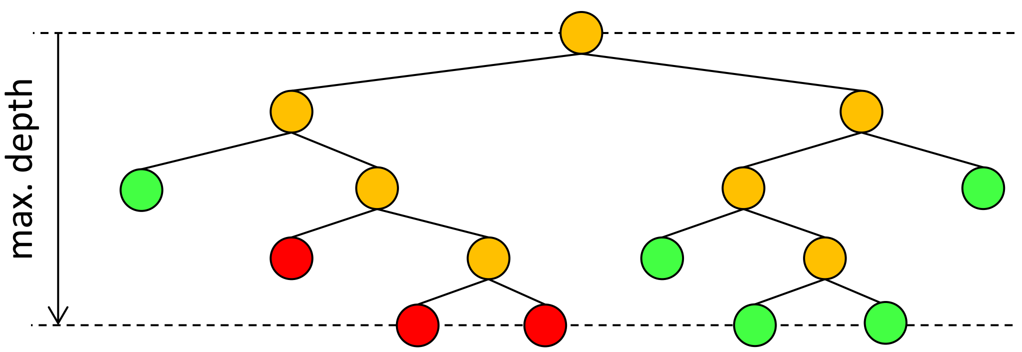


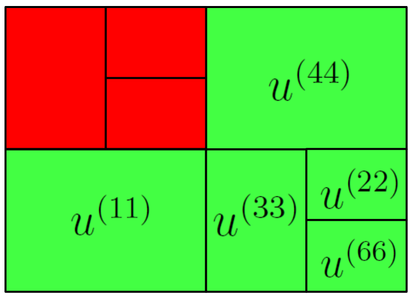


Bisection, binary tree

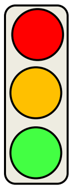
How to organize hyperrectangular state space representation?

- Use bisection and binary tree





control law



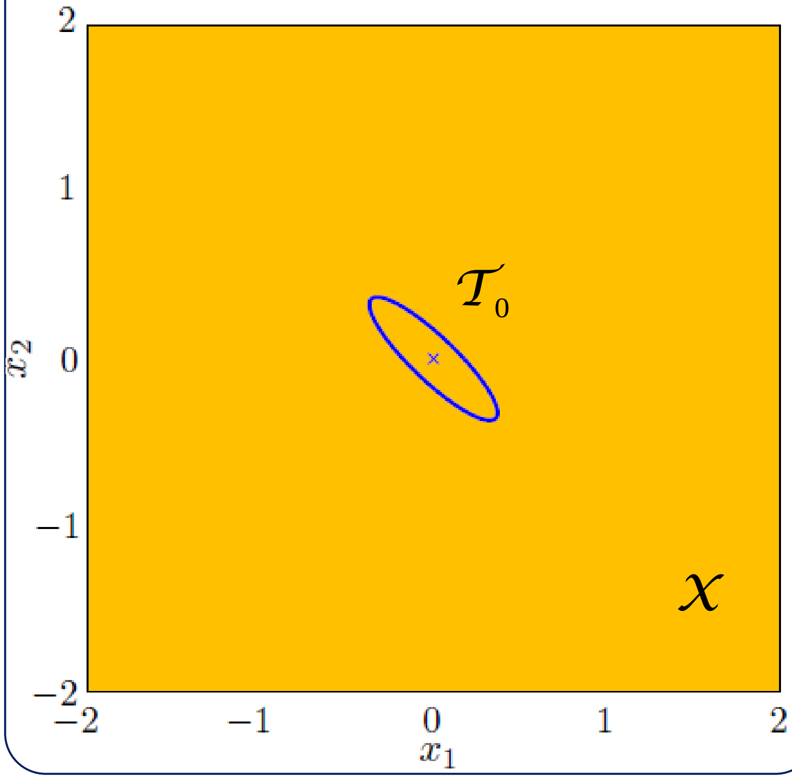
can not be steered towards $\mathcal{T}$

uncertain

can be steered towards $\mathcal{T}$

Numerical example

Terminal set / state constraints



Example

- bilinear system (Chen and Allgöwer, 1998)

$$\begin{cases} f_1(x,u) = x_1 + 0.1x_2 + 0.1\cdot(0.5+0.5x_1)\cdot u \\ f_2(x,u) = x_2 + 0.1x_1 + 0.1\cdot(0.5+0.5x_2)\cdot u \end{cases}$$

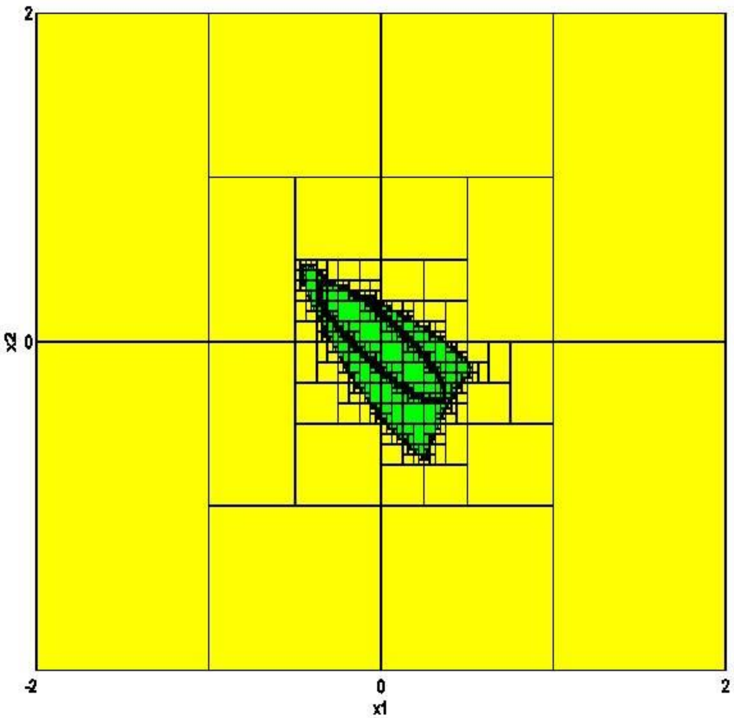
- state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 \,\middle|\, \|x\|_\infty \leq 2 \right\}$$

$$\mathcal{U} = \left\{ u \in \mathbb{R} \,\middle|\, |u| \leq 2 \right\}$$

Numerical example

Feasible set



N=1

Example

- bilinear system (Chen and Allgöwer, 1998)

$$\begin{cases} f_1(x,u) = x_1 + 0.1x_2 + 0.1\cdot(0.5+0.5x_1)\cdot u \\ f_2(x,u) = x_2 + 0.1x_1 + 0.1\cdot(0.5+0.5x_2)\cdot u \end{cases}$$

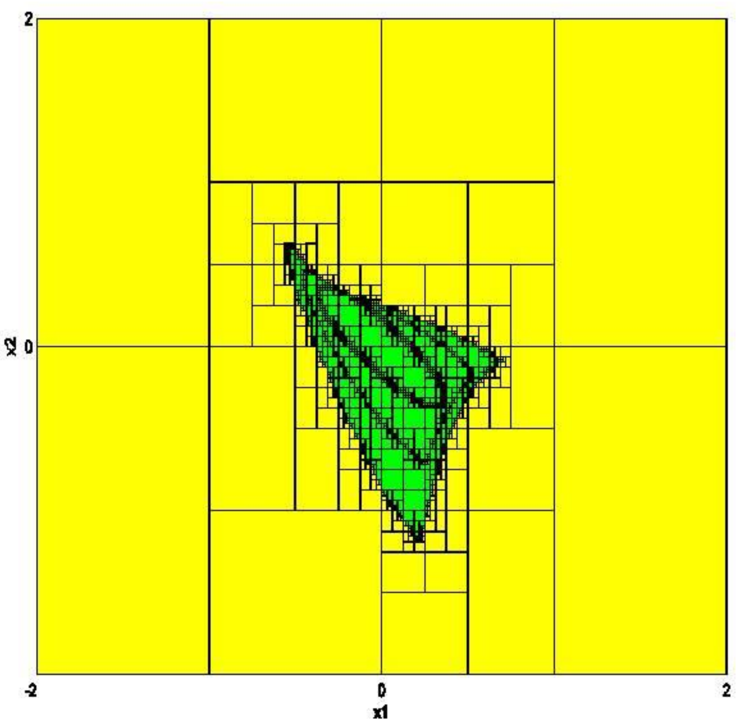
- state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 \mid \|x\|_\infty \le 2 \right\}$$

$$\mathcal{U} = \left\{ u \in \mathbb{R} \mid |u| \le 2 \right\}$$

Numerical example

Feasible set



N=2

Example

- bilinear system (Chen and Allgöwer, 1998)

$$\begin{cases} f_1(x,u) = x_1 + 0.1x_2 + 0.1\cdot(0.5+0.5x_1)\cdot u \\ f_2(x,u) = x_2 + 0.1x_1 + 0.1\cdot(0.5+0.5x_2)\cdot u \end{cases}$$

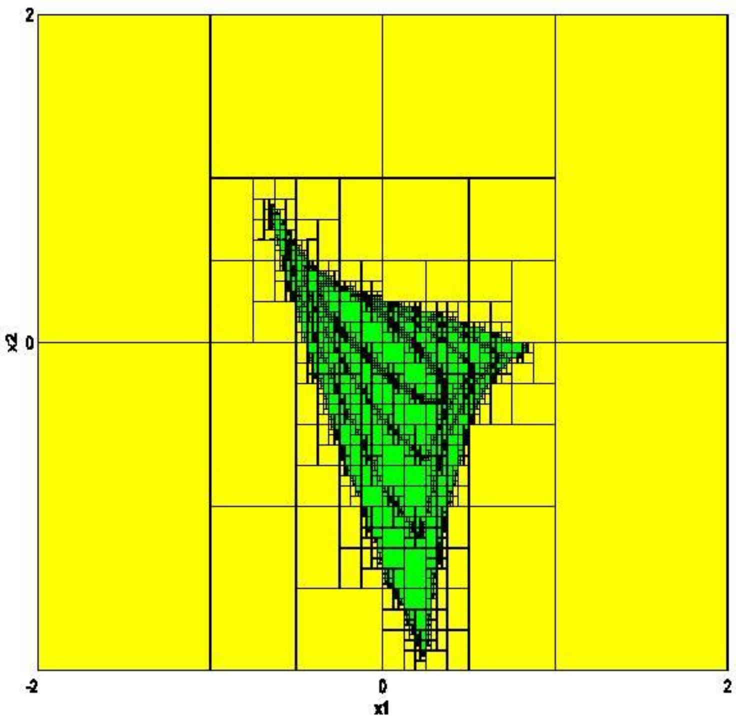
- state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 \,\middle|\, \|x\|_\infty \le 2 \right\}$$

$$\mathcal{U} = \left\{ u \in \mathbb{R} \,\middle|\, |u| \le 2 \right\}$$

Numerical example

Feasible set



N=3

Example

- bilinear system (Chen and Allgöwer, 1998)

$$\begin{cases} f_1(x,u) = x_1 + 0.1x_2 + 0.1\cdot(0.5+0.5x_1)\cdot u \\ f_2(x,u) = x_2 + 0.1x_1 + 0.1\cdot(0.5+0.5x_2)\cdot u \end{cases}$$

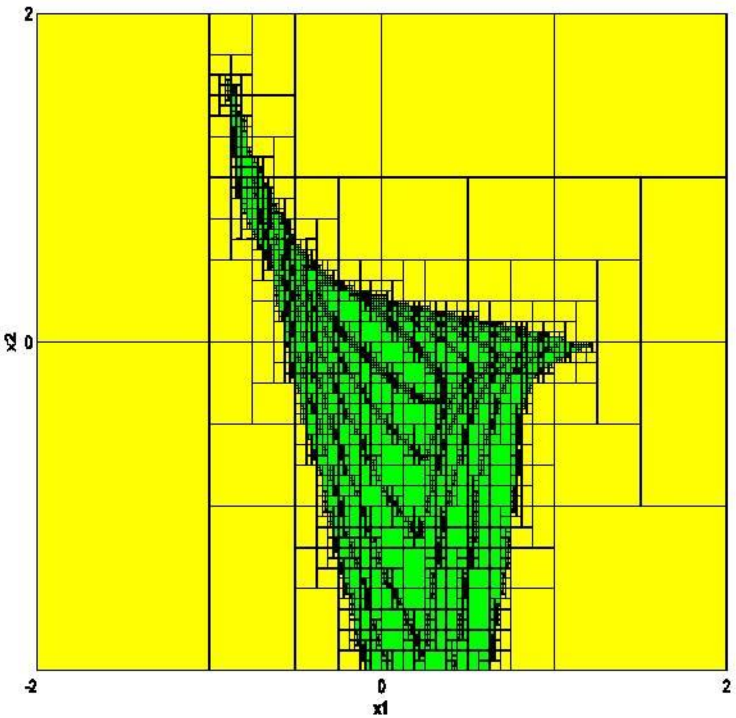
- state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 \mid \|x\|_\infty \leq 2 \right\}$$

$$\mathcal{U} = \left\{ u \in \mathbb{R} \mid |u| \leq 2 \right\}$$

Numerical example

Feasible set



N=5

Example

- bilinear system (Chen and Allgöwer, 1998)

$$\begin{cases} f_1(x,u) = x_1 + 0.1x_2 + 0.1\cdot(0.5+0.5x_1)\cdot u \\ f_2(x,u) = x_2 + 0.1x_1 + 0.1\cdot(0.5+0.5x_2)\cdot u \end{cases}$$

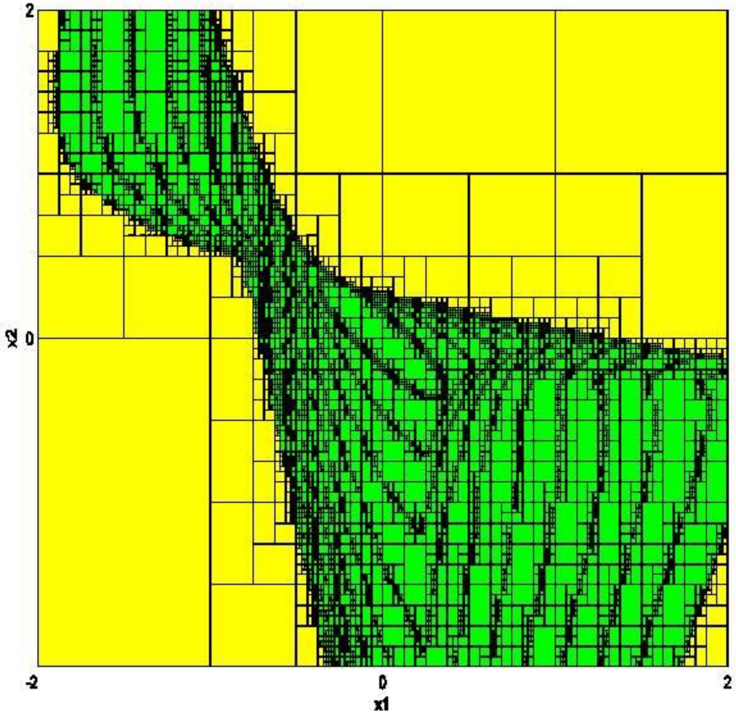
- state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 \,\middle|\, \|x\|_\infty \le 2 \right\}$$

$$\mathcal{U} = \left\{ u \in \mathbb{R} \,\middle|\, |u| \le 2 \right\}$$

Numerical example

Feasible set



N=10

Example

- bilinear system (Chen and Allgöwer, 1998)

$$\begin{cases} f_1(x,u) = x_1 + 0.1x_2 + 0.1\cdot(0.5+0.5x_1)\cdot u \\ f_2(x,u) = x_2 + 0.1x_1 + 0.1\cdot(0.5+0.5x_2)\cdot u \end{cases}$$

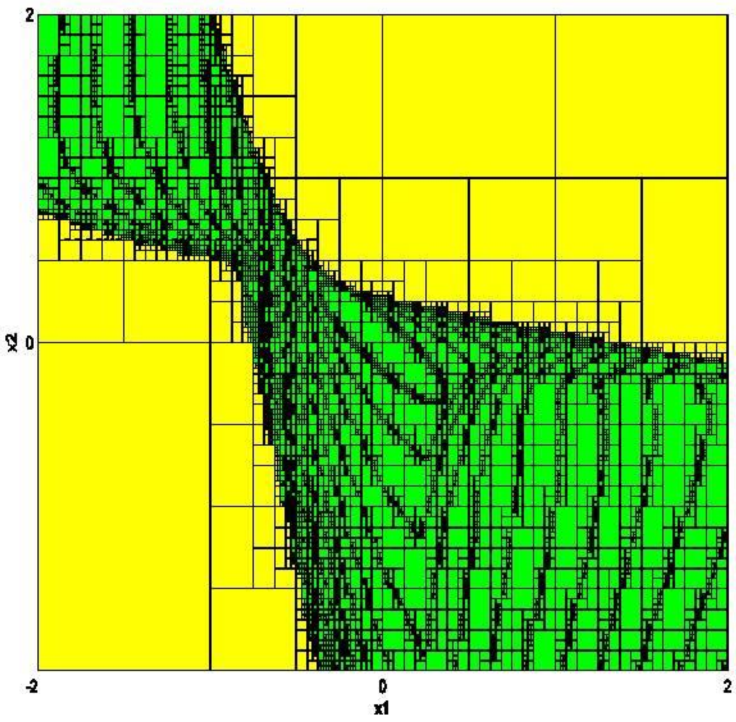
- state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 \mid \|x\|_\infty \le 2 \right\}$$

$$\mathcal{U} = \left\{ u \in \mathbb{R} \mid |u| \le 2 \right\}$$

Numerical example

Feasible set



N=12

Example

- bilinear system (Chen and Allgöwer, 1998)

$$\begin{cases} f_1(x,u) = x_1 + 0.1x_2 + 0.1\cdot(0.5+0.5x_1)\cdot u \\ f_2(x,u) = x_2 + 0.1x_1 + 0.1\cdot(0.5+0.5x_2)\cdot u \end{cases}$$

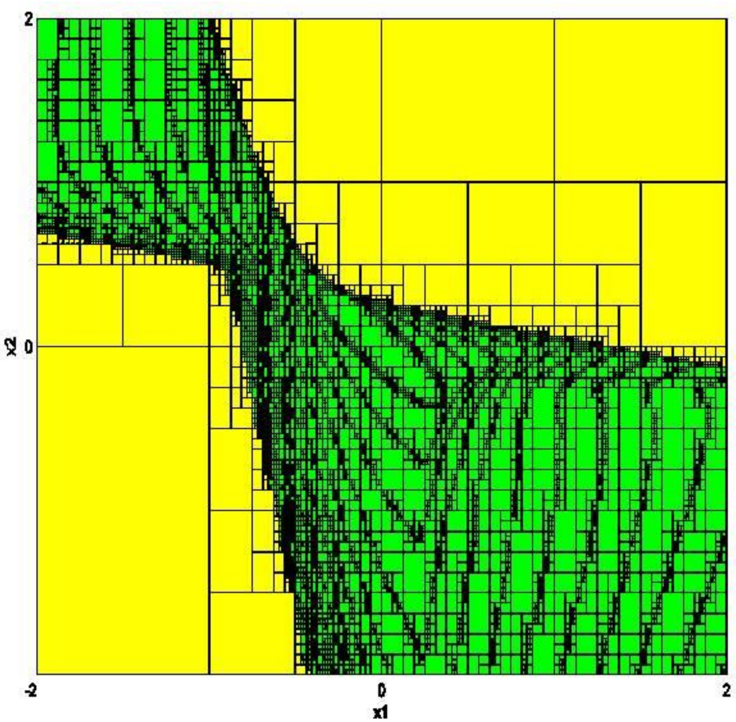
- state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 \middle| \ \|x\|_\infty \le 2 \right\}$$

$$\mathcal{U} = \left\{ u \in \mathbb{R} \ \middle| \ |u| \le 2 \right\}$$

Numerical example

Feasible set



N=15

Example

- bilinear system (Chen and Allgöwer, 1998)

$$\begin{cases} f_1(x,u) = x_1 + 0.1x_2 + 0.1\cdot(0.5+0.5x_1)\cdot u \\ f_2(x,u) = x_2 + 0.1x_1 + 0.1\cdot(0.5+0.5x_2)\cdot u \end{cases}$$

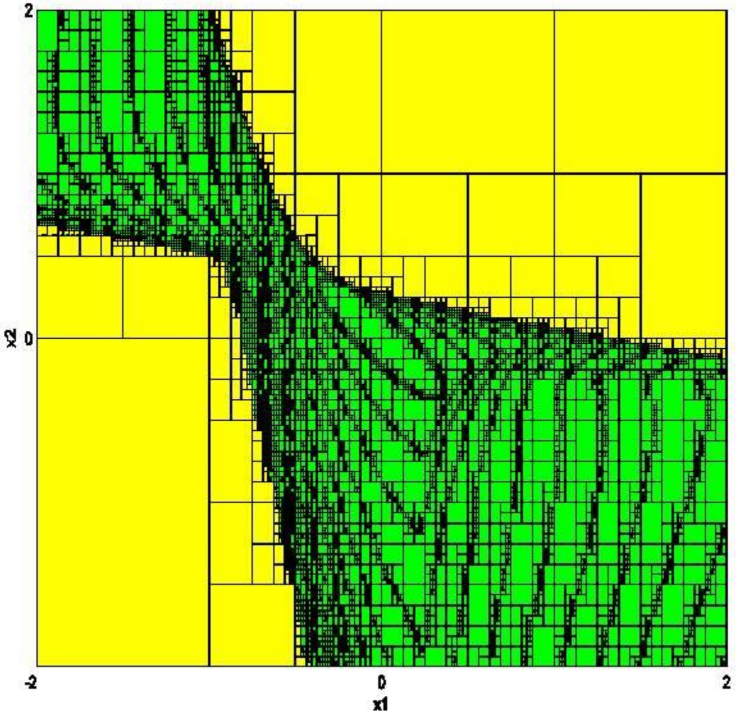
- state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 \mid \|x\|_\infty \leq 2 \right\}$$

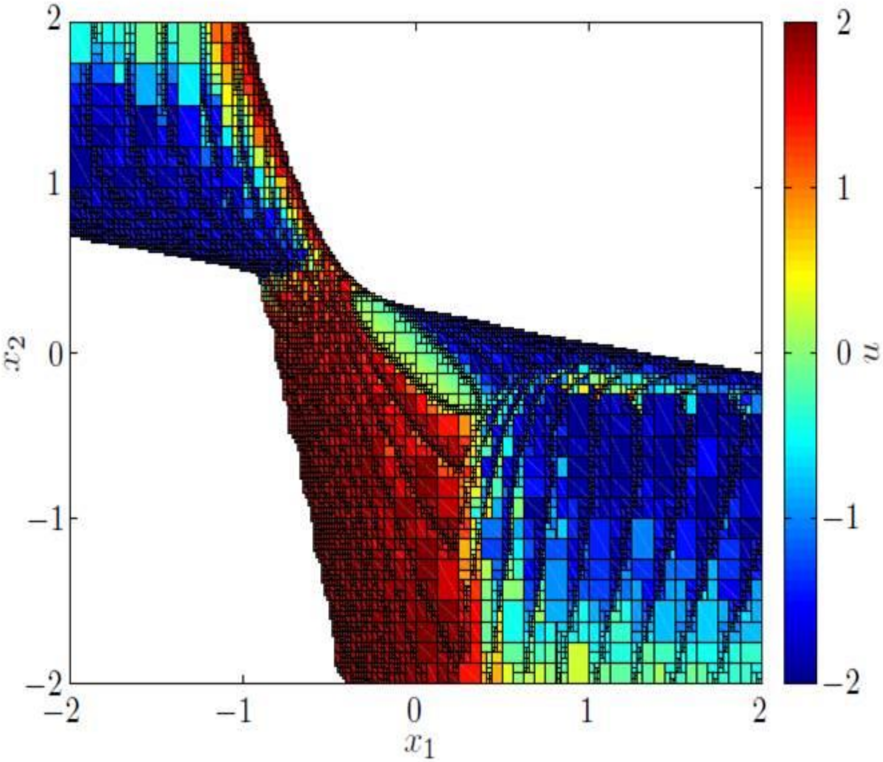
$$\mathcal{U} = \left\{ u \in \mathbb{R} \mid |u| \leq 2 \right\}$$

Numerical example

Feasible set

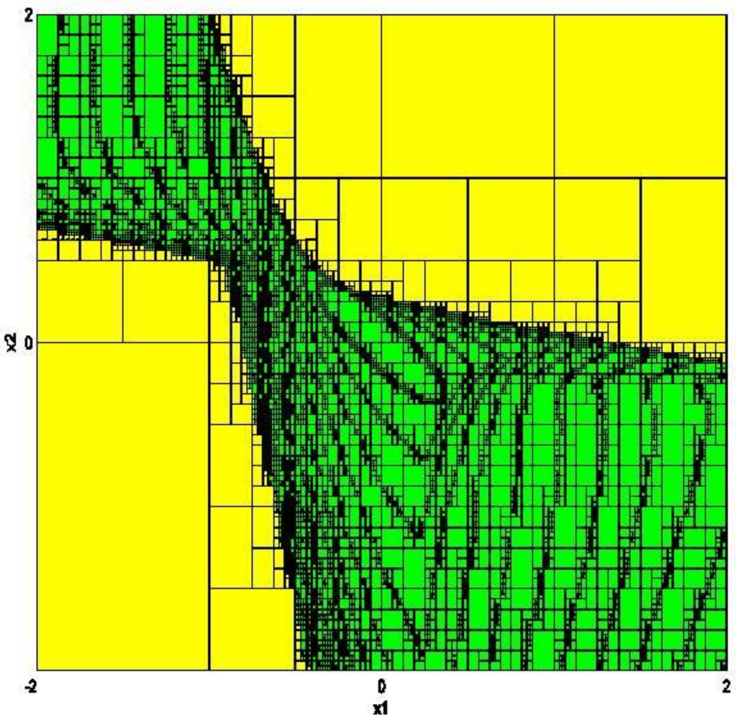


PWC-controller



Numerical example

Feasible set



Considerations

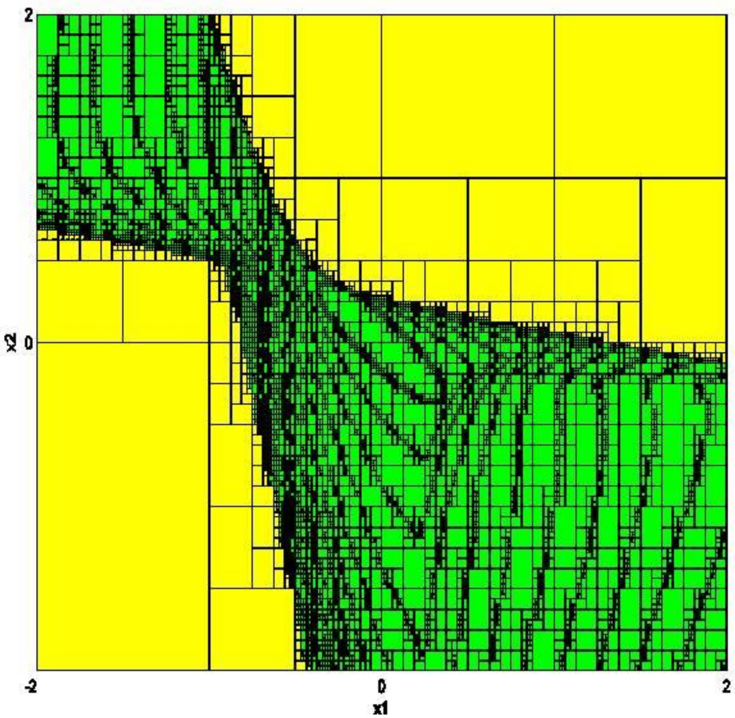
Stability is guaranteed

Once the terminal set is attained we switch to the auxiliary controller $k_f(x)$

Differently from the original method, the use of hyperrectangles provides non-overlapping regions

Numerical example

Feasible set



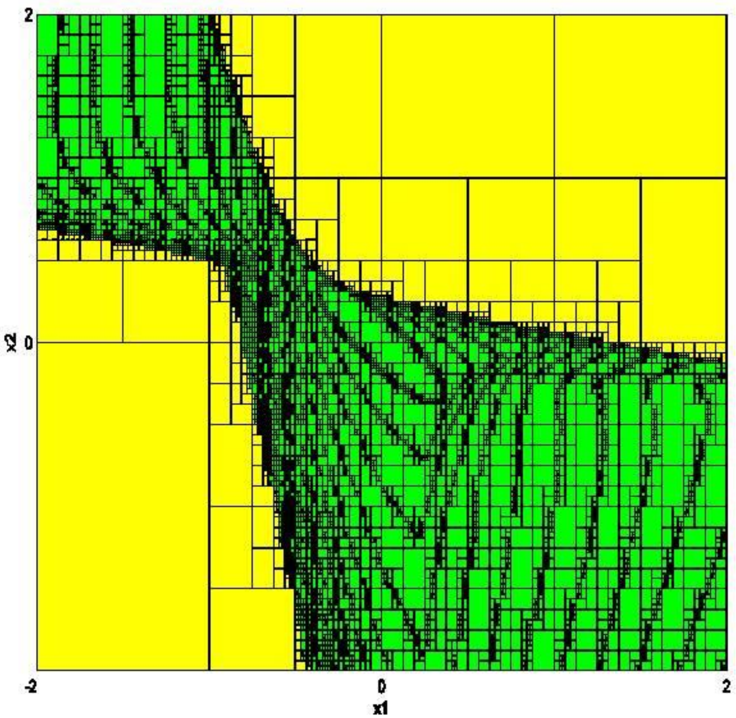
Considerations

In order to reduce suboptimality

Candidate regions can be further split according to the cost function $x_1' Q x_1 + u_0' R u_0$ and stop when the gap between keeping the same controller or differentiating it for each subset is smaller than a threshold ε.

Numerical example

Feasible set

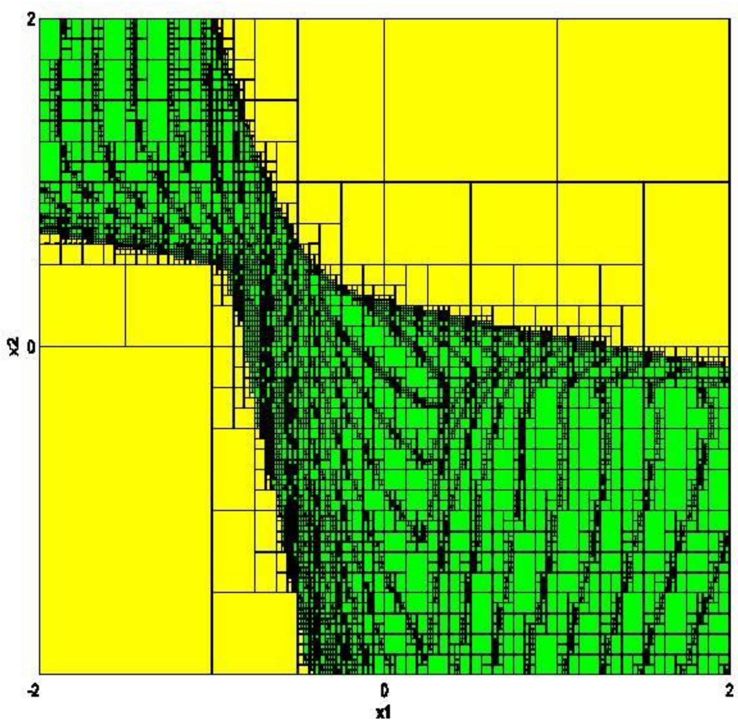


Considerations

How conservative is the inner approximation of the feasible set?

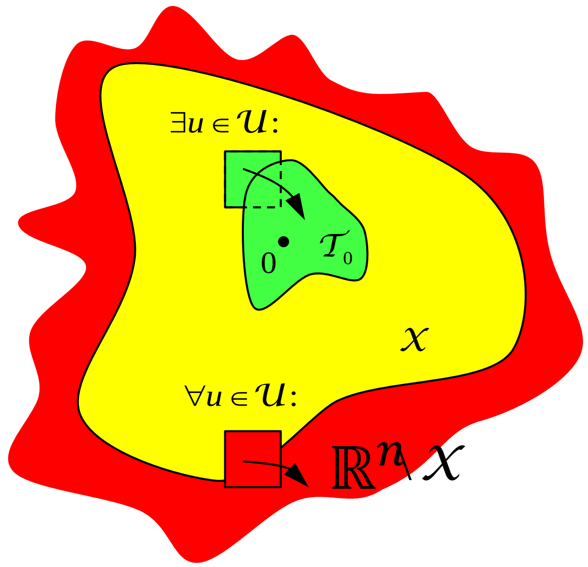
Numerical example

Feasible set



Considerations

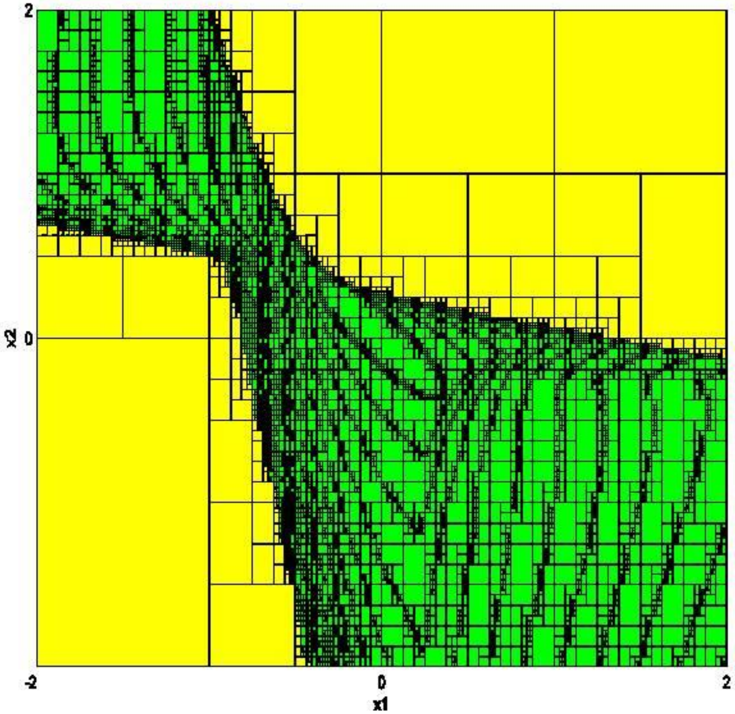
Inner approximation of infeasible set



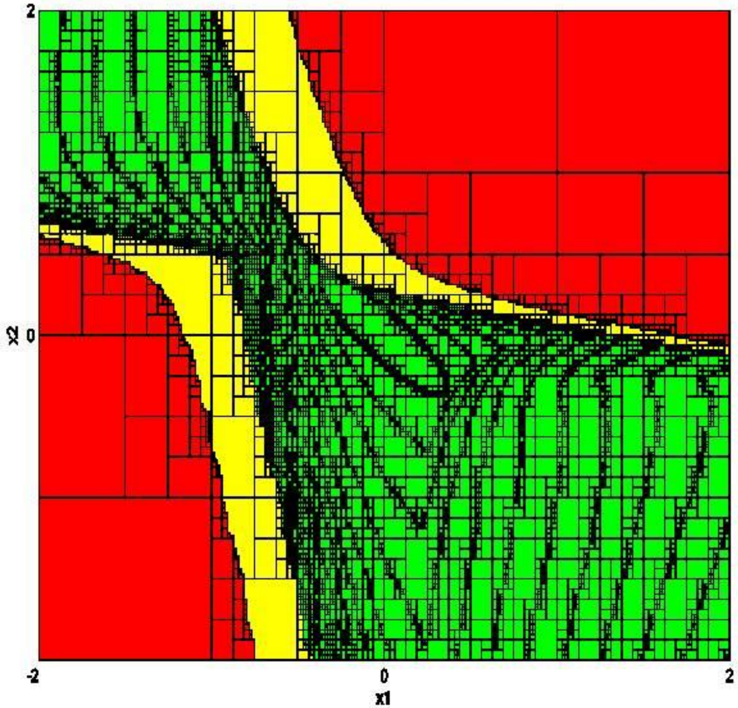
Compute the regions for which all U lead to $R^n \setminus X$ *with* $N \rightarrow \infty$

Numerical example

Feasible set



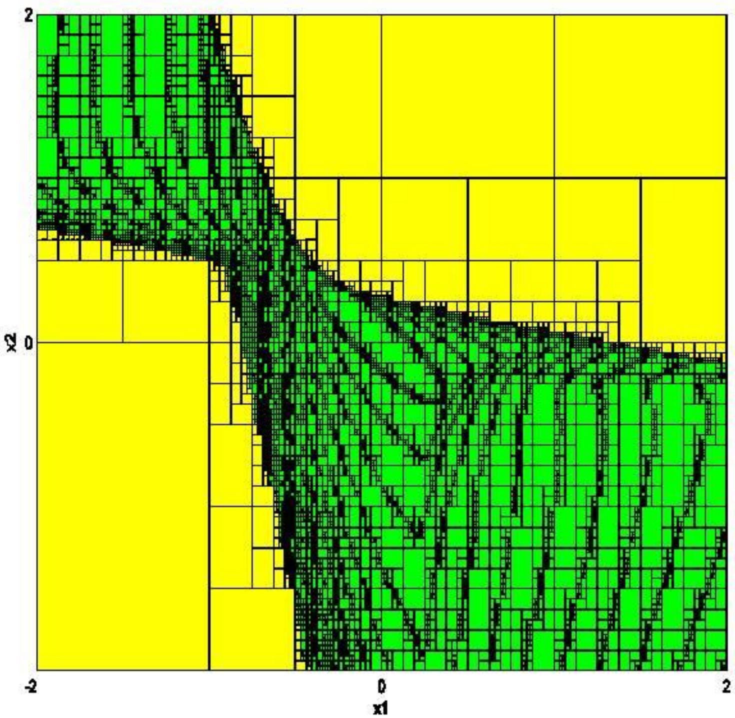
Considerations



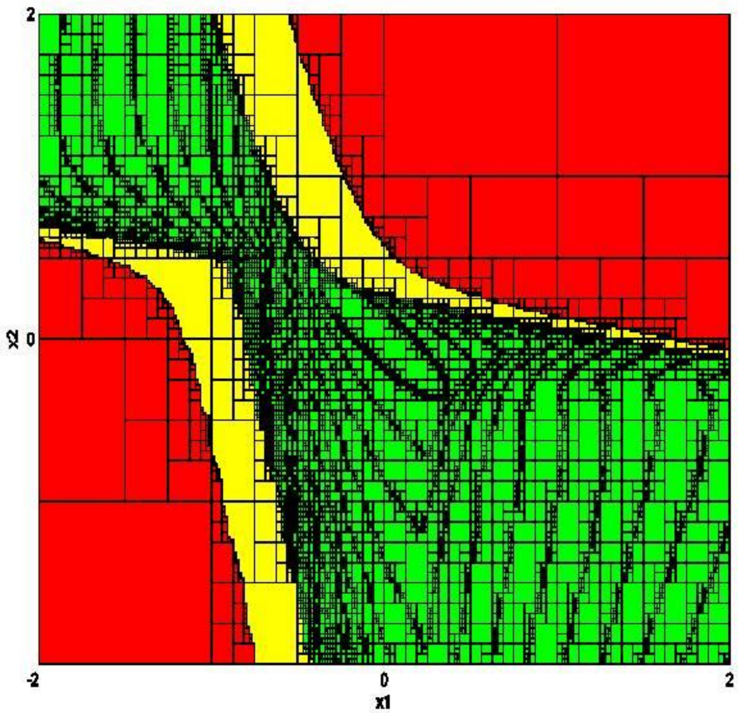
Why the gap?

Numerical example

Feasible set



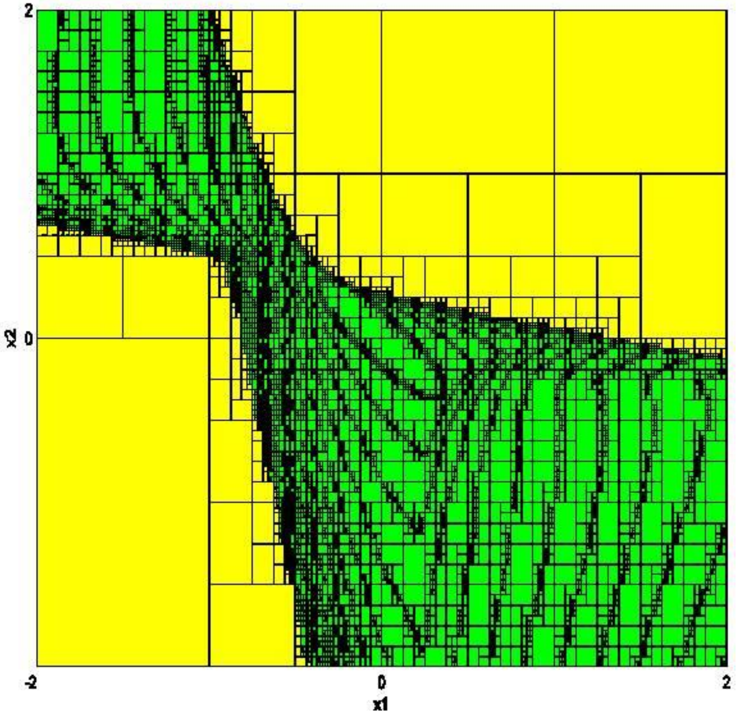
Considerations



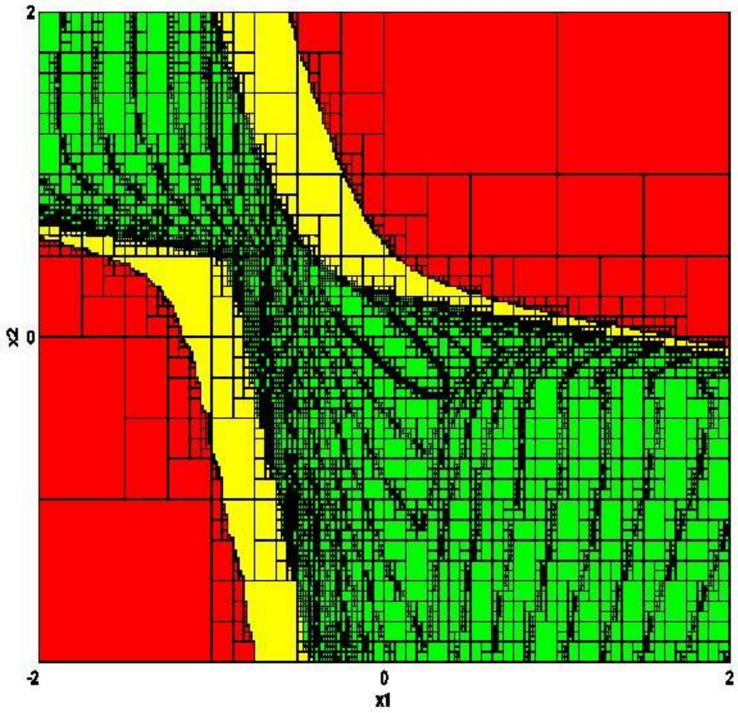
Finite horizon for the green region

Numerical example

Feasible set



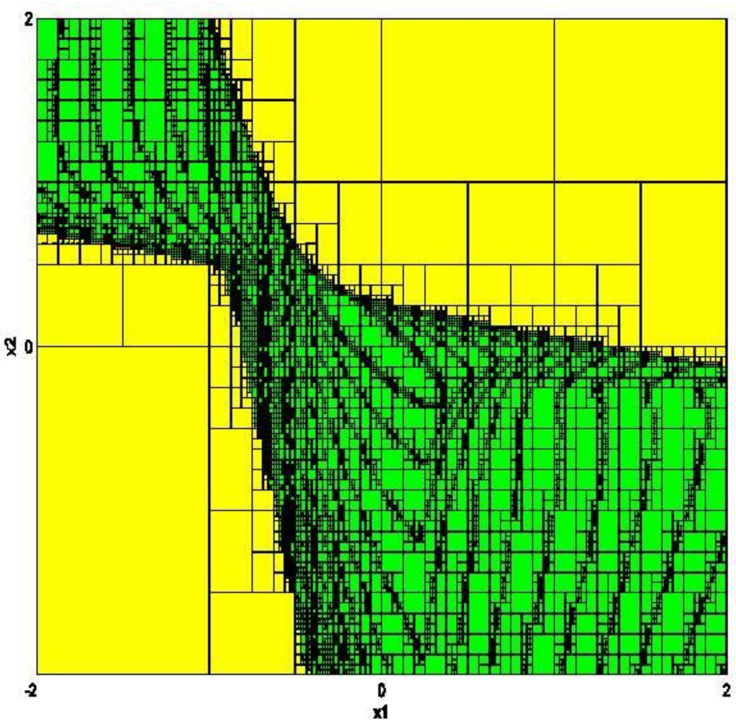
Considerations



Not feasible also when $\mathcal{T}$ is not attained in N steps

Numerical example

Feasible set



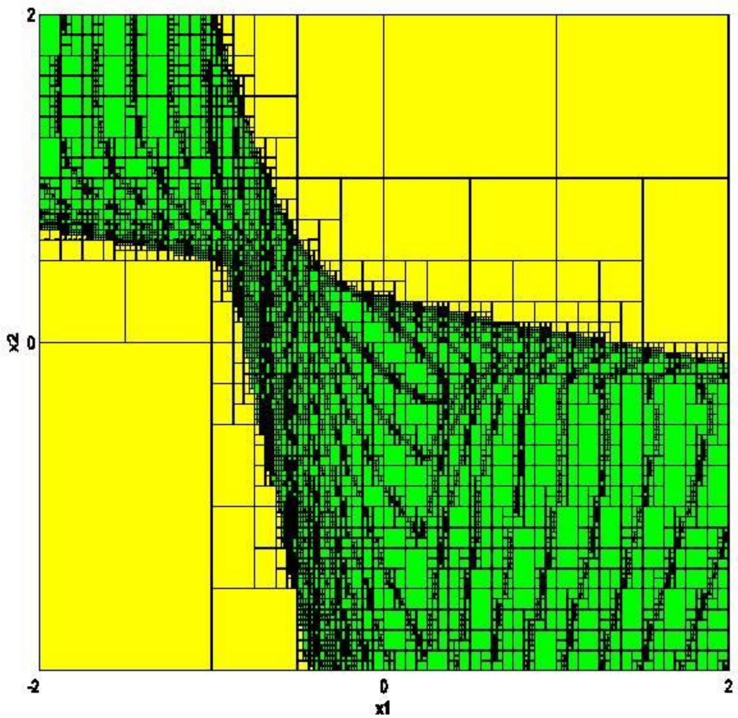
Considerations

What about solving directly the N-step problem?

Stability: difficult to get a suboptimality gap from the optimal solution.

Numerical example

Feasible set



3240 regions

Considerations

Fast online evaluation

Minimal storage requirements

Hash Map Representation:

Memory saving

Search Tree Representation

Fast online evaluation time

The control law can be evaluated in *31ns* (Hash) or *0.5μs* (Search Tree)