Model Predictive Control

Lars Grüne

Mathematisches Institut, Universität Bayreuth

IDK Winter School, Thurnau, March 4–6, 2009

Contents

Part 1:

Part 2:

(1) Introduction

What is Model Predictive Control (MPC)?

Setup

We consider nonlinear discrete time control systems

$$x(n+1) = f(x(n), u(n))$$

with $x(n) \in X$, $u(n) \in U$

Setup

We consider nonlinear discrete time control systems

$$x(n+1) = f(x(n), u(n))$$

with $x(n) \in X$, $u(n) \in U$

- we consider discrete time systems for simplicity of exposition

Setup

We consider nonlinear discrete time control systems

$$x(n+1) = f(x(n), u(n))$$

with $x(n) \in X$, $u(n) \in U$

- we consider discrete time systems for simplicity of exposition
- continuous time systems can be treated in an analogous way or as discrete time sampled data systems

UNIVERSITÄT BAYREUTH

Setup

We consider nonlinear discrete time control systems

$$x(n+1) = f(x(n), u(n))$$

with $x(n) \in X$, $u(n) \in U$

- we consider discrete time systems for simplicity of exposition
- continuous time systems can be treated in an analogous way or as discrete time sampled data systems
- X and U depend on the model. These may be Euclidean spaces $\mathbb{R}^n$ and $\mathbb{R}^m$ or more general (e.g., infinite dimensional) spaces

Setup

We consider nonlinear discrete time control systems

$$x(n+1) = f(x(n), u(n))$$

with $x(n) \in X$, $u(n) \in U$

- we consider discrete time systems for simplicity of exposition
- continuous time systems can be treated in an analogous way or as discrete time sampled data systems
- X and U depend on the model. These may be Euclidean spaces $\mathbb{R}^n$ and $\mathbb{R}^m$ or more general (e.g., infinite dimensional) spaces
- state and control constraints can be added explicitly or included implicitly by chosing X and U as suitable subsets of the respective spaces

Prototype Problem

Assume there exists an equilibrium $x^* \in X$ for $u = 0$, i.e.

$$f(x^*, 0) = x^*$$

Prototype Problem

Assume there exists an equilibrium $x^* \in X$ for $u = 0$, i.e.

$$f(x^*, 0) = x^*$$

Task: stabilize the system

$$x(n+1) = f(x(n), u(n))$$

at x^* via static state feedback

Prototype Problem

Assume there exists an equilibrium $x^* \in X$ for $u = 0$, i.e.

$$f(x^*, 0) = x^*$$

Task: stabilize the system

$$x(n+1) = f(x(n), u(n))$$

at x^* via static state feedback

i.e., find $F : X \to U$, such that x^* is asymptotically stable for the feedback controlled system

$$x_F(n+1) = f(x_F(n), F(x_F(n)))$$

Prototype Problem

Recall: Asymptotic stability means

Prototype Problem

Recall: Asymptotic stability means

Attraction: $x_F(n) \to x^*$ as $n \to \infty$ for all $x_F(0) \in X$

plus

Stability: Solutions starting close to 0 remain close to 0

UNIVERSITÄT BAYREUTH

Prototype Problem

Recall: Asymptotic stability means

Attraction: $x_F(n) \to x^*$ as $n \to \infty$ for all $x_F(0) \in X$

plus

Stability: Solutions starting close to 0 remain close to 0 or, formally: for each $\delta > 0$ there exists $\varepsilon > 0$ such that

$$\|x_F(n) - x^*\| \leq \delta \text{ for all } \|x_F(0) - x^*\| \leq \varepsilon,\ n \in \mathbb{N}_0$$

UNIVERSITÄT BAYREUTH

Prototype Problem

Recall: Asymptotic stability means

Attraction: $x_F(n) \to x^*$ as $n \to \infty$ for all $x_F(0) \in X$

plus

Stability: Solutions starting close to 0 remain close to 0 or, formally: for each $\delta > 0$ there exists $\varepsilon > 0$ such that

$$\|x_F(n) - x^*\| \leq \delta \text{ for all } \|x_F(0) - x^*\| \leq \varepsilon,\ n \in \mathbb{N}_0$$

This prototype "equilibrium stabilization problem" is easily generalizable to tracking, set stabilization, ...

Prototype Problem

Recall: Asymptotic stability means

Attraction: $x_F(n) \to x^*$ as $n \to \infty$ for all $x_F(0) \in X$

plus

Stability: Solutions starting close to 0 remain close to 0 or, formally: for each $\delta > 0$ there exists $\varepsilon > 0$ such that

$$\|x_F(n) - x^*\| \leq \delta \text{ for all } \|x_F(0) - x^*\| \leq \varepsilon,\ n \in \mathbb{N}_0$$

This prototype "equilibrium stabilization problem" is easily generalizable to tracking, set stabilization, ...

In the sequel, we always assume that the problem is solvable, i.e., that a stabilizing feedback $F : X \to U$ exists

UNIVERSITÄT BAYREUTH

The basic idea of MPC

(1) At each time $\tau \in \mathbb{N}_0$, for the current state x_τ, use the model to predict solutions

$$x(n+1) = f(x(n), u(n)), \quad n = 0, \ldots, N-1, \;\; x(0) = x_\tau,$$

The basic idea of MPC

(1) At each time $\tau \in \mathbb{N}_0$, for the current state x_τ, use the model to predict solutions

$$x(n+1) = f(x(n), u(n)), \quad n = 0, \dots, N-1, \;\; x(0) = x_\tau,$$

(2) Use these predictions in order to optimize

$$J_N(x_\tau, u) = \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

over the control sequences $u = (u(0), \dots, u(N-1)) \in U^N$, where $\ell(x,u)$ penalizes the distance from the equilibrium and control effort, e.g., $\ell(x,u) = \|x - x^*\|^2 + \lambda\|u\|^2$

The basic idea of MPC

(1) At each time $\tau \in \mathbb{N}_0$, for the current state x_τ, use the model to predict solutions

$$x(n+1) = f(x(n), u(n)), \quad n = 0, \dots, N-1, \;\; x(0) = x_\tau,$$

(2) Use these predictions in order to optimize

$$J_N(x_\tau, u) = \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

over the control sequences $u = (u(0), \dots, u(N-1)) \in U^N$, where $\ell(x,u)$ penalizes the distance from the equilibrium and control effort, e.g., $\ell(x,u) = \|x - x^*\|^2 + \lambda \|u\|^2$

(3) From the optimal control sequence $u^*(0), \dots, u^*(N-1)$, use the first element as feedback value, i.e.,

$$F(x_\tau) := u^*(0)$$

UNIVERSITÄT BAYREUTH

MPC from the control point of view

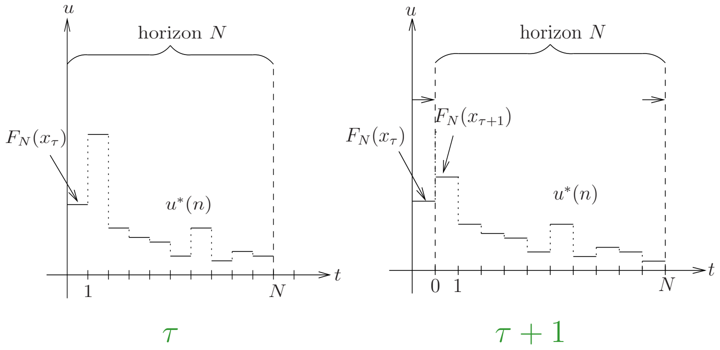
$$\text{minimize } J_N(x_\tau, u) = \sum_{n=0}^{N-1} \ell(x(n), u(t)), \quad x(0) = x_\tau$$

$\rightsquigarrow$ optimal control $u^*(0), \ldots, u^*(N-1)$ $\rightsquigarrow$ set $F_N(x_\tau) := u^*(0)$

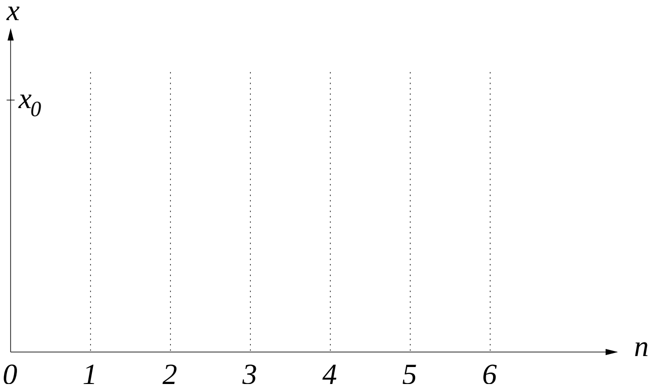
MPC from the control point of view

$$\text{minimize } J_N(x_\tau, u) = \sum_{n=0}^{N-1} \ell(x(n), u(t)), \quad x(0) = x_\tau$$

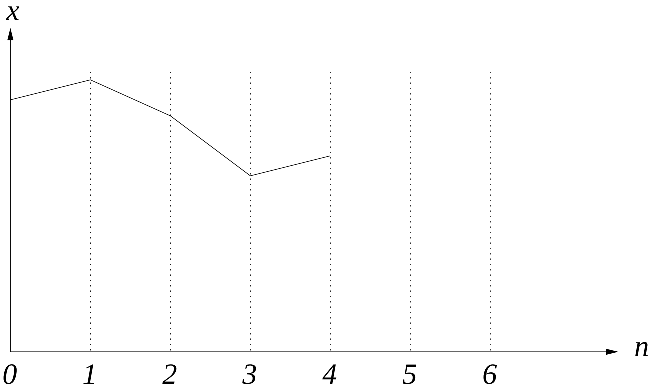
$\rightsquigarrow$ optimal control $u^*(0), \ldots, u^*(N-1)$ $\rightsquigarrow$ set $F_N(x_\tau) := u^*(0)$



MPC from the trajectory point of view

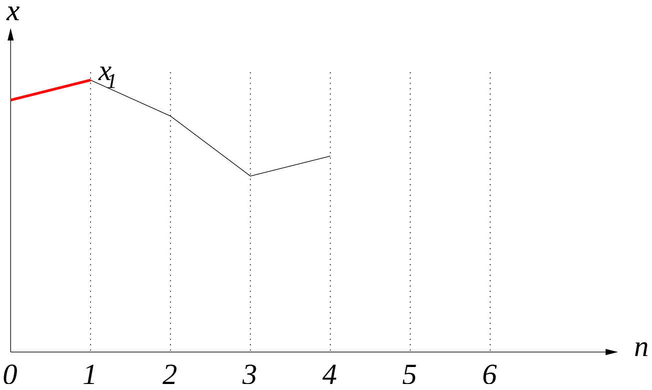


MPC from the trajectory point of view



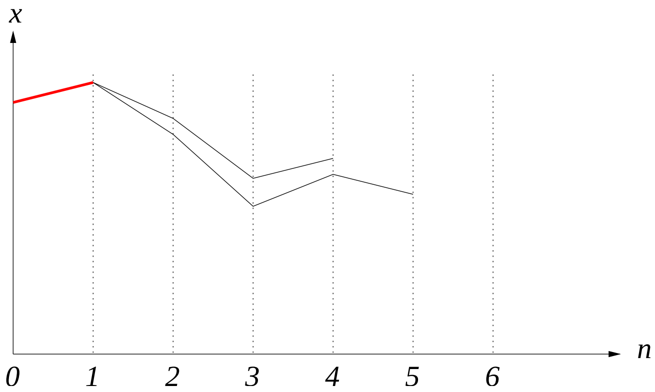
black = predictions (open loop optimization)

MPC from the trajectory point of view



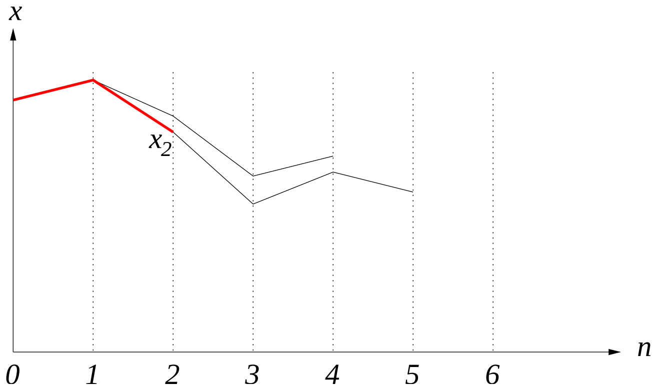
black = predictions (open loop optimization)
red = MPC closed loop

MPC from the trajectory point of view



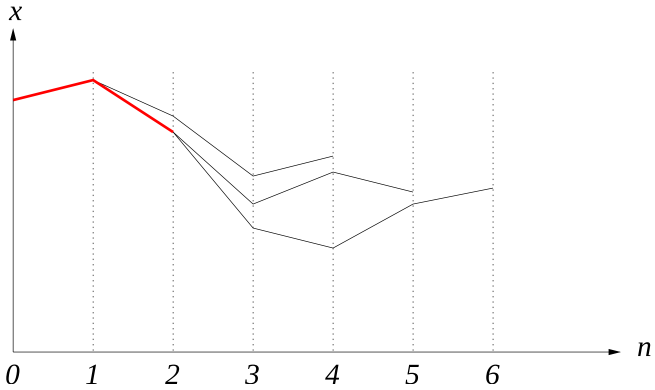
black = predictions (open loop optimization)
red = MPC closed loop

MPC from the trajectory point of view



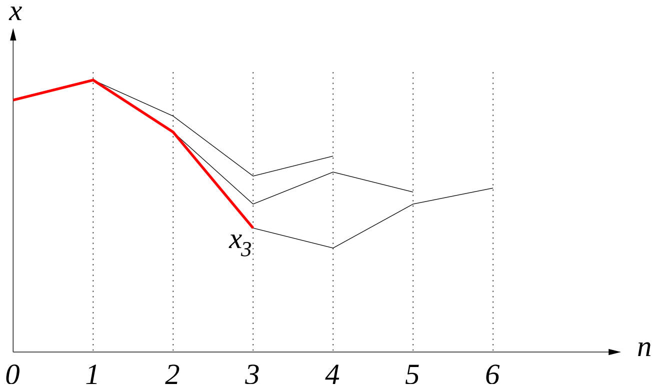
black = predictions (open loop optimization)
red = MPC closed loop

MPC from the trajectory point of view



black = predictions (open loop optimization)
red = MPC closed loop

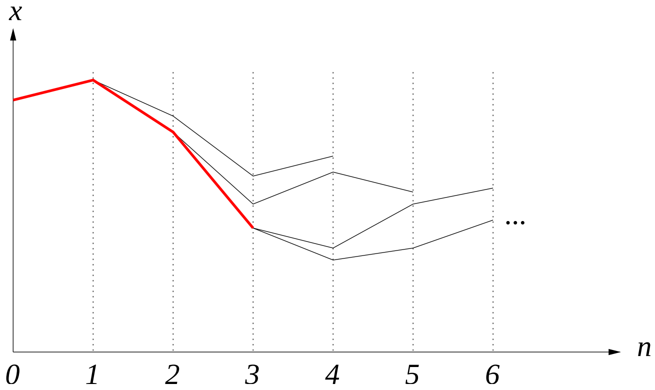
MPC from the trajectory point of view



black = predictions (open loop optimization)
red = MPC closed loop

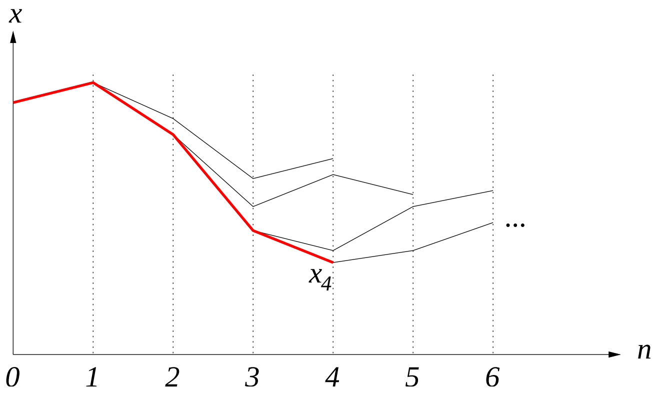
UNIVERSITÄT BAYREUTH

MPC from the trajectory point of view



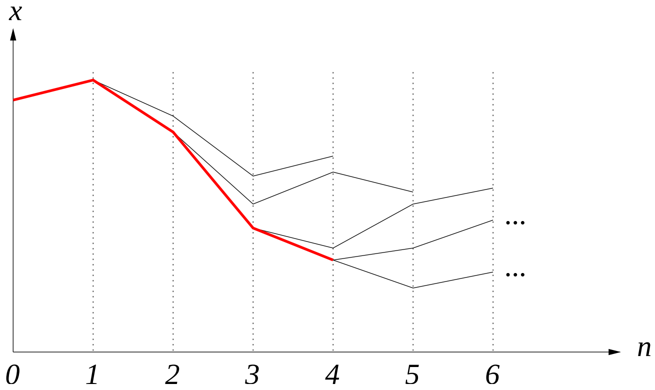
black = predictions (open loop optimization)
red = MPC closed loop

MPC from the trajectory point of view



black = predictions (open loop optimization)
red = MPC closed loop

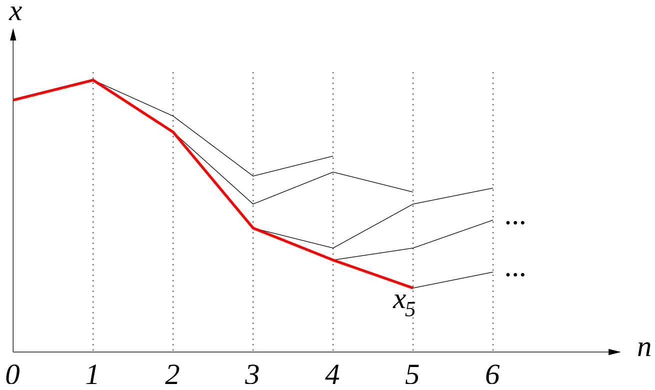
MPC from the trajectory point of view



black = predictions (open loop optimization)

red = MPC closed loop

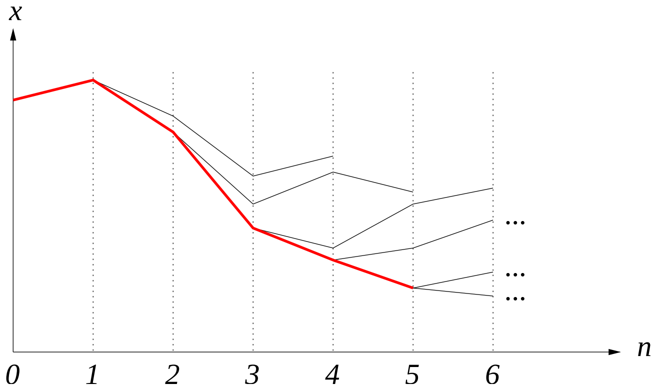
MPC from the trajectory point of view



black = predictions (open loop optimization)
red = MPC closed loop

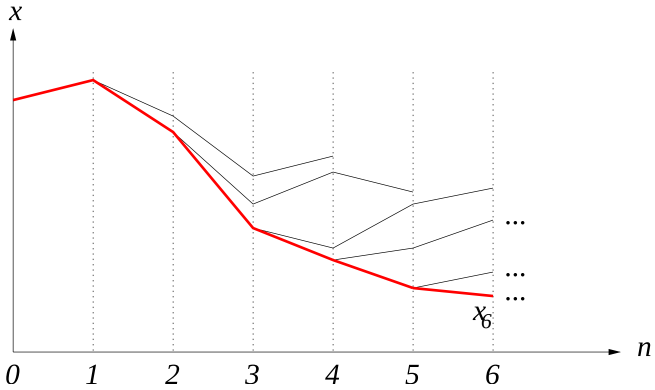
UNIVERSITÄT BAYREUTH

MPC from the trajectory point of view



black = predictions (open loop optimization)
red = MPC closed loop

MPC from the trajectory point of view



black = predictions (open loop optimization)

red = MPC closed loop

Model predictive control (aka Receding horizon control)

Idea first formulated in [A.I. Propoi, *Use of linear programming methods for synthesizing sampled-data automatic systems*, Automation and Remote Control 1963]

Model predictive control (aka Receding horizon control)

Idea first formulated in [A.I. Propoi, *Use of linear programming methods for synthesizing sampled-data automatic systems*, Automation and Remote Control 1963], often rediscovered

Model predictive control (aka Receding horizon control)

Idea first formulated in [A.I. Propoi, *Use of linear programming methods for synthesizing sampled-data automatic systems*, Automation and Remote Control 1963], often rediscovered

used in industrial applications since the mid 1970s, mainly for constrained linear systems [Qin & Badgwell, 1997, 2001]

Model predictive control (aka Receding horizon control)

Idea first formulated in [A.I. Propoi, *Use of linear programming methods for synthesizing sampled-data automatic systems*, Automation and Remote Control 1963], often rediscovered

used in industrial applications since the mid 1970s, mainly for constrained linear systems [Qin & Badgwell, 1997, 2001]

more than 9000 industrial MPC applications in Germany counted in [Dittmar & Pfeifer, 2005]

Model predictive control (aka Receding horizon control)

Idea first formulated in [A.I. Propoi, *Use of linear programming methods for synthesizing sampled-data automatic systems*, Automation and Remote Control 1963], often rediscovered

used in industrial applications since the mid 1970s, mainly for constrained linear systems [Qin & Badgwell, 1997, 2001]

more than 9000 industrial MPC applications in Germany counted in [Dittmar & Pfeifer, 2005]

development of theory since ∼1980 (linear), ∼1990 (nonlinear)

Model predictive control (aka Receding horizon control)

Idea first formulated in [A.I. Propoi, *Use of linear programming methods for synthesizing sampled-data automatic systems*, Automation and Remote Control 1963], often rediscovered

used in industrial applications since the mid 1970s, mainly for constrained linear systems [Qin & Badgwell, 1997, 2001]

more than 9000 industrial MPC applications in Germany counted in [Dittmar & Pfeifer, 2005]

development of theory since ∼1980 (linear), ∼1990 (nonlinear)

Central questions:

- When does MPC stabilize the system?

Model predictive control (aka Receding horizon control)

Idea first formulated in [A.I. Propoi, *Use of linear programming methods for synthesizing sampled-data automatic systems*, Automation and Remote Control 1963], often rediscovered

used in industrial applications since the mid 1970s, mainly for constrained linear systems [Qin & Badgwell, 1997, 2001]

more than 9000 industrial MPC applications in Germany counted in [Dittmar & Pfeifer, 2005]

development of theory since ∼1980 (linear), ∼1990 (nonlinear)

Central questions:

- When does MPC stabilize the system?
- How good is the performance of the MPC feedback law?

UNIVERSITÄT BAYREUTH

Model predictive control (aka Receding horizon control)

Idea first formulated in [A.I. Propoi, *Use of linear programming methods for synthesizing sampled-data automatic systems*, Automation and Remote Control 1963], often rediscovered

used in industrial applications since the mid 1970s, mainly for constrained linear systems [Qin & Badgwell, 1997, 2001]

more than 9000 industrial MPC applications in Germany counted in [Dittmar & Pfeifer, 2005]

development of theory since ∼1980 (linear), ∼1990 (nonlinear)

Central questions:

- When does MPC stabilize the system?
- How good is the performance of the MPC feedback law?
- How long does the optimization horizon N need to be?

Model predictive control (aka Receding horizon control)

Idea first formulated in [A.I. Propoi, *Use of linear programming methods for synthesizing sampled-data automatic systems*, Automation and Remote Control 1963], often rediscovered

used in industrial applications since the mid 1970s, mainly for constrained linear systems [Qin & Badgwell, 1997, 2001]

more than 9000 industrial MPC applications in Germany counted in [Dittmar & Pfeifer, 2005]

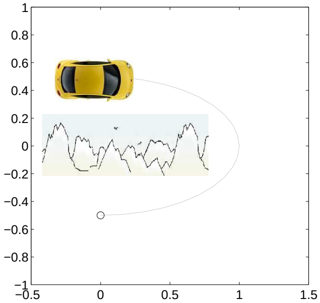
development of theory since ∼1980 (linear), ∼1990 (nonlinear)

Central questions:

- When does MPC stabilize the system?
- How good is the performance of the MPC feedback law?
- How long does the optimization horizon N need to be?

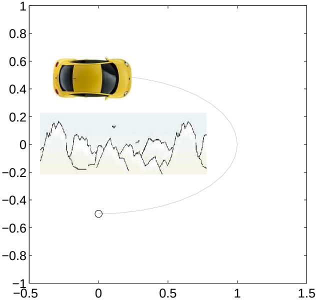
and, of course, the development of good algorithms (not topic of this course)

An example



UNIVERSITÄT BAYREUTH

An example



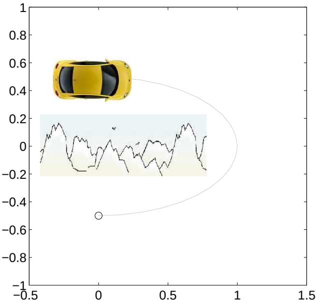
$$\begin{aligned} x_1(n+1) &= \alpha \sin(\varphi + u) \\ x_2(n+1) &= \alpha \cos(\varphi + u)/2 \end{aligned}$$

with $\alpha = \|(x_1, 2x_2)^T\|$, $\varphi = \begin{cases} \arccos(x_2/\alpha), & x_1 \geq 0 \\ 2\pi - \arccos(x_2/\alpha), & x_1 < 0, \end{cases}$

$X = \mathbb{R}^2$, $U = [0, u_{\max}]$, $x^* = (0, -1/2)^T$, $x_0 = (0, 1/2)^T$

UNIVERSITÄT BAYREUTH

An example



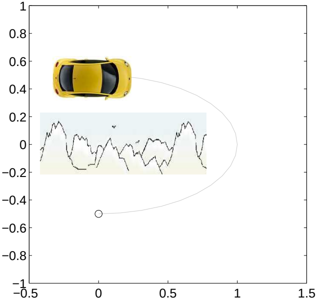
$$x_1(n+1) = \alpha \sin(\varphi + u)$$
$$x_2(n+1) = \alpha \cos(\varphi + u)/2$$

with $\alpha = \|(x_1, 2x_2)^T\|$, $\varphi = \begin{cases} \arccos(x_2/\alpha), & x_1 \geq 0 \\ 2\pi - \arccos(x_2/\alpha), & x_1 < 0, \end{cases}$

$X = \mathbb{R}^2$, $U = [0, u_{\max}]$, $x^* = (0, -1/2)^T$, $x_0 = (0, 1/2)^T$

MPC with $\ell(x,u) = \|x - x^*\|^2 + |u|^2$ and $u_{\max} = 0.2$ yields asymptotic stability for $N = 11$

An example



$$x_1(n+1) = \alpha \sin(\varphi + u)$$
$$x_2(n+1) = \alpha \cos(\varphi + u)/2$$

with $\alpha = \|(x_1, 2x_2)^T\|$, $\varphi = \begin{cases} \arccos(x_2/\alpha), & x_1 \geq 0 \\ 2\pi - \arccos(x_2/\alpha), & x_1 < 0, \end{cases}$

$X = \mathbb{R}^2$, $U = [0, u_{\max}]$, $x^* = (0, -1/2)^T$, $x_0 = (0, 1/2)^T$

MPC with $\ell(x,u) = \|x - x^*\|^2 + |u|^2$ and $u_{\max} = 0.2$ yields asymptotic stability for $N = 11$ but not for $N \leq 10$

UNIVERSITÄT BAYREUTH

(2) Background

Infinite horizon optimal control

Stabilization via optimal control

For continuous running cost $\ell : X \times U \to \mathbb{R}_0^+$ with

$$\min_{u \in U} \ell(x,u) > 0 \text{ for } x \neq x^* \quad \text{and} \quad \ell(x^*,0) = 0$$

define the infinite horizon functional

$$J_\infty(x,u) := \sum_{n=0}^{\infty} \ell(x(n),u(n))$$

and the optimal value function

$$V_\infty(x) := \inf_{u:\mathbb{N}_0 \to U} J_\infty(x,u)$$

Stabilization via optimal control

$$V_\infty(x) = \inf_{u:\mathbb{N}_0 \to U} J_\infty(x,u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{\infty} \ell(x(n), u(n))$$

Stabilization via optimal control

$$V_\infty(x) = \inf_{u:\mathbb{N}_0\to U} J_\infty(x,u) = \inf_{u:\mathbb{N}_0\to U} \sum_{n=0}^{\infty} \ell(x(n),u(n))$$

Facts (for suitable ℓ):

- if the feedback stabilization problem is solvable, then the function V_∞ is finite and continuous

Stabilization via optimal control

$$V_\infty(x) = \inf_{u:\mathbb{N}_0\to U} J_\infty(x,u) = \inf_{u:\mathbb{N}_0\to U} \sum_{n=0}^{\infty} \ell(x(n),u(n))$$

Facts (for suitable ℓ):

- if the feedback stabilization problem is solvable, then the function V_∞ is finite and continuous
- V_∞ satisfies the Dynamic Programming Principle

$$V_\infty(x) = \min_{u\in U} \{\ell(x,u) + V_\infty(f(x,u))\}$$

UNIVERSITÄT BAYREUTH

Stabilization via optimal control

$$V_\infty(x) = \inf_{u:\mathbb{N}_0\to U} J_\infty(x,u) = \inf_{u:\mathbb{N}_0\to U} \sum_{n=0}^{\infty} \ell(x(n),u(n))$$

Facts (for suitable ℓ):

- if the feedback stabilization problem is solvable, then the function V_∞ is finite and continuous
- V_∞ satisfies the Dynamic Programming Principle

$$V_\infty(x) = \min_{u\in U} \{\ell(x,u) + V_\infty(f(x,u))\}$$

- if we choose $F_\infty(x) \in U$ as the minimizer, i.e.,

$$F_\infty(x) = \operatorname*{argmin}_{u\in U}\{\ell(x,u) + V_\infty(f(x,u))\}$$

then F_∞ is the optimal feedback

Asymptotic stability of the optimal feedback law

Furthermore F_∞ is asymptotically stabilizing:

This follows from

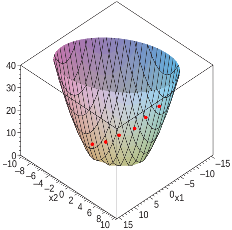
$$V_\infty(f(x, F_\infty(x))) \le V_\infty(x) - \ell(x, F_\infty(x))$$

Asymptotic stability of the optimal feedback law

Furthermore F_∞ is asymptotically stabilizing:

This follows from

$$V_\infty(f(x, F_\infty(x))) \leq \underbrace{V_\infty(x) - \ell(x, F_\infty(x))}_{<V_\infty(x) \text{ for } x \neq x^*}$$



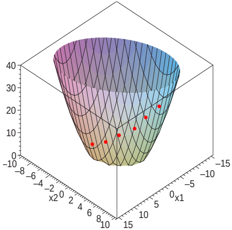
$\Rightarrow$ V_∞ is a Lyapunov function

Asymptotic stability of the optimal feedback law

Furthermore F_∞ is asymptotically stabilizing:

This follows from

$$V_\infty(f(x,F_\infty(x))) \le \underbrace{V_\infty(x) - \ell(x,F_\infty(x))}_{<V_\infty(x) \text{ for } x \neq x^*}$$



$\Rightarrow$ V_∞ is a Lyapunov function

$\rightsquigarrow$ approach for MPC:

Prove similar inequalities for F_N and

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0\to U} J_N(x(0),u) = \inf_{u:\mathbb{N}_0\to U} \sum_{n=0}^{N-1} \ell(x(n),u(n))$$

and use V_N as a Lyapunov function

(3) The Stability Problem

V_N as a Lyapunov Function

Problem: Prove that the MPC feedback law F_N is stabilizing

UNIVERSITÄT BAYREUTH

V_N as a Lyapunov Function

Problem: Prove that the MPC feedback law F_N is stabilizing

Approach: Define the finite time optimal value function

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0 \to U} J_N(x(0), u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

and prove that V_N is Lyapunov function

V_N as a Lyapunov Function

Problem: Prove that the MPC feedback law F_N is stabilizing

Approach: Define the finite time optimal value function

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0 \to U} J_N(x(0), u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

and prove that V_N is Lyapunov function, i.e., that V_N has suitable upper and lower bounds (automatically inherited from ℓ)

UNIVERSITÄT BAYREUTH

V_N as a Lyapunov Function

Problem: Prove that the MPC feedback law F_N is stabilizing

Approach: Define the finite time optimal value function

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0 \to U} J_N(x(0), u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

and prove that V_N is Lyapunov function, i.e., that V_N has suitable upper and lower bounds (automatically inherited from ℓ) and

$$V_N(f(x, F_N(x))) \leq V_N(x) - \tilde{\ell}(x, F_N(x))$$

for some $\tilde{\ell} : X \times U \to \mathbb{R}_0^+$ with $\tilde{\ell}(x, F_N(x)) > 0$ for $x \neq x^*$

V_N as a Lyapunov Function

Problem: Prove that the MPC feedback law F_N is stabilizing

Approach: Define the finite time optimal value function

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0 \to U} J_N(x(0), u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

and prove that V_N is Lyapunov function, i.e., that V_N has suitable upper and lower bounds (automatically inherited from ℓ) and

$$V_N(f(x, F_N(x))) \leq V_N(x) - \tilde{\ell}(x, F_N(x))$$

for some $\tilde{\ell} : X \times U \to \mathbb{R}_0^+$ with $\tilde{\ell}(x, F_N(x)) > 0$ for $x \neq x^*$

$\Rightarrow \quad V_N(x_{F_N}(n)) \to 0$

UNIVERSITÄT BAYREUTH

V_N as a Lyapunov Function

Problem: Prove that the MPC feedback law F_N is stabilizing

Approach: Define the finite time optimal value function

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0 \to U} J_N(x(0), u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

and prove that V_N is Lyapunov function, i.e., that V_N has suitable upper and lower bounds (automatically inherited from ℓ) and

$$V_N(f(x, F_N(x))) \leq V_N(x) - \tilde{\ell}(x, F_N(x))$$

for some $\tilde{\ell} : X \times U \to \mathbb{R}_0^+$ with $\tilde{\ell}(x, F_N(x)) > 0$ for $x \neq x^*$

$\Rightarrow \quad V_N(x_{F_N}(n)) \to 0 \quad \Rightarrow \quad x_{F_N}(n) \to x^*$ + stability

UNIVERSITÄT BAYREUTH

V_N as a Lyapunov Function

Problem: Prove that the MPC feedback law F_N is stabilizing

Approach: Define the finite time optimal value function

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0 \to U} J_N(x(0), u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

and prove that V_N is Lyapunov function, i.e., that V_N has suitable upper and lower bounds (automatically inherited from ℓ) and

$$V_N(f(x, F_N(x))) \leq V_N(x) - \tilde{\ell}(x, F_N(x))$$

for some $\tilde{\ell} : X \times U \to \mathbb{R}_0^+$ with $\tilde{\ell}(x, F_N(x)) > 0$ for $x \neq x^*$

$\Rightarrow \quad V_N(x_{F_N}(n)) \to 0 \quad \Rightarrow \quad x_{F_N}(n) \to x^*$ + stability

(most commonly used approach in the literature)

Why is this difficult?

We want

$$V_N(f(x, F_N(x))) \le V_N(x) - \tilde{\ell}(x, F_N(x)) \qquad (*)$$

Why is this difficult?

We want

$$V_N(f(x, F_N(x))) \le V_N(x) - \underbrace{\tilde{\ell}(x, F_N(x))}_{<0 \text{ for } x \neq x^*} \qquad (*)$$

Why is this difficult?

We want

$$V_N(f(x, F_N(x))) \leq V_N(x) - \underbrace{\tilde{\ell}(x, F_N(x))}_{<0 \text{ for } x \neq x^*} \qquad (*)$$

For $N = \infty$, the dynamic programming principle immediately implies $(*)$ with $\tilde{\ell}(x, F_\infty(x)) = \ell(x, F_\infty(x))$:

$$V_\infty(x) = \ell(x, F_\infty(x)) + V_\infty(f(x, F_\infty(x)))$$

Why is this difficult?

We want

$$V_N(f(x,F_N(x))) \le V_N(x) - \underbrace{\tilde{\ell}(x,F_N(x))}_{<0 \text{ for } x\neq x^*} \qquad (*)$$

For $N=\infty$, the dynamic programming principle immediately implies $(*)$ with $\tilde{\ell}(x,F_\infty(x)) = \ell(x,F_\infty(x))$:

$$V_\infty(x) = \ell(x,F_\infty(x)) + V_\infty(f(x,F_\infty(x)))$$

The dynamic programming principle for V_N reads

$$\begin{aligned} V_N(x) &= \min_{u\in U}\{\ell(x,u) + V_{N-1}(f(x,u))\} \\ &= \ell(x,F_N(x)) + V_{N-1}(f(x,F_N(x))) \end{aligned}$$

Why is this difficult?

We want

$$V_N(f(x, F_N(x))) \leq V_N(x) - \underbrace{\tilde{\ell}(x, F_N(x))}_{<0 \text{ for } x \neq x^*} \qquad (*)$$

For $N = \infty$, the dynamic programming principle immediately implies $(*)$ with $\tilde{\ell}(x, F_\infty(x)) = \ell(x, F_\infty(x))$:

$$V_\infty(x) = \ell(x, F_\infty(x)) + V_\infty(f(x, F_\infty(x)))$$

The dynamic programming principle for V_N reads

$$\begin{aligned} V_N(x) &= \min_{u \in U}\{\ell(x, u) + V_{N-1}(f(x, u))\} \\ &= \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) \end{aligned}$$

Thus, $(*)$ follows with

$$\tilde{\ell}(x, u) = \ell(x, u) + V_{N-1}(f(x, u)) - V_N(f(x, u))$$

Why is this difficult?

We want

$$V_N(f(x, F_N(x))) \le V_N(x) - \underbrace{\tilde{\ell}(x, F_N(x))}_{<0 \text{ for } x \neq x^*} \qquad (*)$$

For $N = \infty$, the dynamic programming principle immediately implies $(*)$ with $\tilde{\ell}(x, F_\infty(x)) = \ell(x, F_\infty(x))$:

$$V_\infty(x) = \ell(x, F_\infty(x)) + V_\infty(f(x, F_\infty(x)))$$

The dynamic programming principle for V_N reads

$$\begin{aligned} V_N(x) &= \min_{u \in U} \{\ell(x, u) + V_{N-1}(f(x, u))\} \\ &= \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) \end{aligned}$$

Thus, $(*)$ follows with

$$\tilde{\ell}(x, u) = \ell(x, u) + V_{N-1}(f(x, u)) - V_N(f(x, u))$$

$\rightsquigarrow$ Problem: ensure $\tilde{\ell}(x, F_N(x)) > 0$ for $x \neq x^*$

Why is this difficult?

Task: Give conditions under which

$$\tilde{\ell}(x, F_N(x)) := \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))) > 0$$

holds for $x \neq x^*$.

UNIVERSITÄT BAYREUTH

Why is this difficult?

Task: Give conditions under which

$$\tilde{\ell}(x, F_N(x)) := \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))) > 0$$

holds for $x \neq x^*$.

For the basic (and most widely used) MPC formulation

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0 \to U} J_N(x(0), u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

this appeared to be out of reach until the mid 1990s

Why is this difficult?

Task: Give conditions under which

$$\tilde{\ell}(x, F_N(x)) := \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))) > 0$$

holds for $x \neq x^*$.

For the basic (and most widely used) MPC formulation

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0 \to U} J_N(x(0), u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

this appeared to be out of reach until the mid 1990s

(note: $V_{N-1} - V_N \leq 0$ by definition; typically with strict "$<$")

Why is this difficult?

Task: Give conditions under which

$$\tilde{\ell}(x, F_N(x)) := \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))) > 0$$

holds for $x \neq x^*$.

For the basic (and most widely used) MPC formulation

$$V_N(x(0)) := \inf_{u:\mathbb{N}_0 \to U} J_N(x(0), u) = \inf_{u:\mathbb{N}_0 \to U} \sum_{n=0}^{N-1} \ell(x(n), u(n))$$

this appeared to be out of reach until the mid 1990s

(note: $V_{N-1} - V_N \leq 0$ by definition; typically with strict "$<$")

$\rightsquigarrow$ additional stabilizing constraints were proposed

UNIVERSITÄT BAYREUTH

(3a) Classical solution of the stability problem: Equilibrium endpoint constraint

Equilibrium endpoint constraint I

Optimal control problem

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n))$$

Equilibrium endpoint constraint I

Optimal control problem

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n))$$

Recall: $f(x^*,0) = x^*$ and $\ell(x^*,0) = 0$

$\rightsquigarrow$ add equilibrium endpoint constraint

$$x(N) = x^*$$

[Keerthi/Gilbert '88, ...]

Equilibrium endpoint constraint II

Then, each feasible trajectory for horizon $N-1$ with control $u(0),\dots,u(N-2)$ can be prolonged with no cost by setting $u(N-1) := 0$

Equilibrium endpoint constraint II

Then, each feasible trajectory for horizon $N-1$ with control $u(0),\dots,u(N-2)$ can be prolonged with no cost by setting $u(N-1) := 0$, i.e.

$$\ell(x(N-1), u(N-1)) = \ell(x^*, 0) = 0$$

Equilibrium endpoint constraint II

Then, each feasible trajectory for horizon $N-1$ with control $u(0),\dots,u(N-2)$ can be prolonged with no cost by setting $u(N-1) := 0$, i.e.

$$\ell(x(N-1),u(N-1)) = \ell(x^*,0) = 0$$

and thus

$$\begin{aligned} J_{N-1}(x(0),u) &= \sum_{n=0}^{N-2} \ell(x(n),u(n)) \\ &= \sum_{n=0}^{N-1} \ell(x(n),u(n)) = J_N(x(0),u). \end{aligned}$$

UNIVERSITÄT BAYREUTH

Equilibrium endpoint constraint II

Then, each feasible trajectory for horizon $N-1$ with control $u(0),\dots,u(N-2)$ can be prolonged with no cost by setting $u(N-1) := 0$, i.e.

$$\ell(x(N-1),u(N-1)) = \ell(x^*,0) = 0$$

and thus

$$\begin{aligned} J_{N-1}(x(0),u) &= \sum_{n=0}^{N-2} \ell(x(n),u(n)) \\ &= \sum_{n=0}^{N-1} \ell(x(n),u(n)) \;=\; J_N(x(0),u). \end{aligned}$$

Since this prolonged trajectory is again feasible, we get

$$V_N(x) \le V_{N-1}(x)$$

UNIVERSITÄT BAYREUTH

Equilibrium endpoint constraint II

Then, each feasible trajectory for horizon $N-1$ with control $u(0), \ldots, u(N-2)$ can be prolonged with no cost by setting $u(N-1) := 0$, i.e.

$$\ell(x(N-1), u(N-1)) = \ell(x^*, 0) = 0$$

and thus

$$\begin{aligned} J_{N-1}(x(0), u) &= \sum_{n=0}^{N-2} \ell(x(n), u(n)) \\ &= \sum_{n=0}^{N-1} \ell(x(n), u(n)) = J_N(x(0), u). \end{aligned}$$

Since this prolonged trajectory is again feasible, we get

$$V_N(x) \leq V_{N-1}(x)$$

Note: $V_{N-1}(x) \leq V_N(x)$ does no longer hold under $x(N) = x^*$

Equilibrium endpoint constraint III

From

$$V_N(x) \le V_{N-1}(x)$$

we get

$$\begin{aligned}\tilde{\ell}(x, F_N(x)) &= \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) \\ &\quad - V_N(f(x, F_N(x))) \\ &\ge \ell(x, F_N(x)) \;>\; 0\end{aligned}$$

for all $x \neq x^*$ by choice of ℓ.

UNIVERSITÄT BAYREUTH

Equilibrium endpoint constraint III

From

$$V_N(x) \le V_{N-1}(x)$$

we get

$$\begin{aligned} \tilde{\ell}(x, F_N(x)) &= \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) \\ &\qquad - V_N(f(x, F_N(x))) \\ &\ge \ell(x, F_N(x)) > 0 \end{aligned}$$

for all $x \neq x^*$ by choice of ℓ.

$\rightsquigarrow$ $$V_N(f(x, F_N(x))) \le V_N(x) - \ell(x, F_N(x)) \qquad (*)$$

i.e., stability with Lyapunov function V_N and $\tilde{\ell} = \ell$

Equilibrium endpoint constraint III

From

$$V_N(x) \le V_{N-1}(x)$$

we get

$$\begin{aligned}\tilde{\ell}(x, F_N(x)) &= \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) \\ &\quad - V_N(f(x, F_N(x))) \\ &\ge \ell(x, F_N(x)) > 0\end{aligned}$$

for all $x \neq x^*$ by choice of ℓ.

$\rightsquigarrow$ $$V_N(f(x, F_N(x))) \le V_N(x) - \ell(x, F_N(x)) \qquad (*)$$

i.e., stability with Lyapunov function V_N and $\tilde{\ell} = \ell$

Note: In general, $x(N) = x^*$ does not imply $x_{F_N}(N) = x^*$

Equilibrium endpoint constraint — Discussion

The additional condition

$$x(N) = x^*$$

ensures asymptotic stability in a rigorously provable way

Equilibrium endpoint constraint — Discussion

The additional condition

$$x(N) = x^*$$

ensures asymptotic stability in a rigorously provable way, but

- online optimization may become harder

Equilibrium endpoint constraint — Discussion

The additional condition

$$x(N) = x^*$$

ensures asymptotic stability in a rigorously provable way, but

- online optimization may become harder
- large feasible set

$$\{x(0) \in \mathbb{R}^n \,|\, x(N) = x^* \text{ for some } u \in \mathcal{U}\}$$

typically needs large optimization horizon N

Equilibrium endpoint constraint — Discussion

The additional condition

$$x(N) = x^*$$

ensures asymptotic stability in a rigorously provable way, but

- online optimization may become harder
- large feasible set

$$\{x(0) \in \mathbb{R}^n \,|\, x(N) = x^* \text{ for some } u \in \mathcal{U}\}$$

  typically needs large optimization horizon N
- system needs to be controllable to x^* in finite time

Equilibrium endpoint constraint — Discussion

The additional condition

$$x(N) = x^*$$

ensures asymptotic stability in a rigorously provable way, but

- online optimization may become harder
- large feasible set

  $$\{x(0) \in \mathbb{R}^n \,|\, x(N) = x^* \text{ for some } u \in \mathcal{U}\}$$

  typically needs large optimization horizon N
- system needs to be controllable to x^* in finite time
- not very often used in industrial practice

(3b) Classical solution of the stability problem: Regional endpoint constraint and terminal cost

Regional constraint and terminal cost I

Optimal control problem

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n))$$

Regional constraint and terminal cost I

Optimal control problem

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n))$$

We want V_N to become a Lyapunov function

$\rightsquigarrow$ add local Lyapunov function $W : B_\delta(x^*) \to \mathbb{R}_0^+$ as terminal cost

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n)) + W(x(N))$$

Regional constraint and terminal cost I

Optimal control problem

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n))$$

We want V_N to become a Lyapunov function

$\rightsquigarrow$ add local Lyapunov function $W : B_\delta(x^*) \to \mathbb{R}_0^+$ as terminal cost

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n)) + W(x(N))$$

and use terminal constraint

$$\|x(N) - x^*\| \leq \delta, \;\; W(x(N)) \leq \varepsilon$$

[Chen & Allgöwer '98, Jadbabaie et al. '98 . . .]

Regional constraint and terminal cost II

$$\text{minimize } J_N(x(0), u) = \sum_{n=0}^{N-1} \ell(x(n), u(n)) + W(x(N))$$

plus terminal constraint

$$\|x(N) - x^*\| \leq \delta, \;\; W(x(N)) \leq \varepsilon$$

Regional constraint and terminal cost II

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n)) + W(x(N))$$

plus terminal constraint

$$\|x(N) - x^*\| \le \delta, \;\; W(x(N)) \le \varepsilon$$

We choose W, ℓ, ε such that

- $\mathrm{cl}\,\{x \in B_\delta(x^*) \,|\, W(x) \le \varepsilon\} \subseteq B_\delta(x^*)$

Regional constraint and terminal cost II

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n)) + W(x(N))$$

plus terminal constraint

$$\|x(N) - x^*\| \le \delta, \;\; W(x(N)) \le \varepsilon$$

We choose W, ℓ, ε such that

- $\text{cl}\,\{x \in B_\delta(x^*)\,|\,W(x) \le \varepsilon\} \subseteq B_\delta(x^*)$
- $W(x) \le \varepsilon$ implies the existence of $F_W(x) \in U$ with

$$W(f(x, F_W(x)) \le W(x) - \ell(x, F_W(x))$$

UNIVERSITÄT BAYREUTH

Regional constraint and terminal cost II

Then, each feasible trajectory for horizon $N-1$ with control $u(0),\ldots,u(N-2)$ can be **prolonged** by setting $u(N-1) := F_W(x(N-1))$.

Regional constraint and terminal cost II

Then, each feasible trajectory for horizon $N-1$ with control $u(0),\dots,u(N-2)$ can be prolonged by setting $u(N-1) := F_W(x(N-1))$. This yields

$$\ell(x(N-1),u(N-1)) \le W(x(N-1)) - W(x(N))$$

and thus

$$\begin{aligned} J_{N-1}(x(0),u) &= \sum_{n=0}^{N-2} \ell(x(n),u(n)) + W(x(N-1)) \\ &\ge \sum_{n=0}^{N-1} \ell(x(n),u(n)) + W(x(N)) \;=\; J_N(x(0),u). \end{aligned}$$

Since this prolonged trajectory is again feasible, we get

$$V_N(x) \le V_{N-1}(x)$$

and we obtain stability just as for the equilibrium constraint

UNIVERSITÄT BAYREUTH

Regional constraint and terminal cost — Discussion

Compared to the equilibrium constraint, the regional constraint

- yields easier online optimization problems

Regional constraint and terminal cost — Discussion

Compared to the equilibrium constraint, the regional constraint

- yields easier online optimization problems
- yields larger feasible sets

Regional constraint and terminal cost — Discussion

Compared to the equilibrium constraint, the regional constraint

- yields easier online optimization problems
- yields larger feasible sets
- does not need exact controllability to x^*

Regional constraint and terminal cost — Discussion

Compared to the equilibrium constraint, the regional constraint

- yields easier online optimization problems
- yields larger feasible sets
- does not need exact controllability to x^*

But:

- large feasible set still needs a large optimization horizon N

Regional constraint and terminal cost — Discussion

Compared to the equilibrium constraint, the regional constraint

- yields easier online optimization problems
- yields larger feasible sets
- does not need exact controllability to x^*

But:

- large feasible set still needs a large optimization horizon N
- additional analytical effort for computing W

Regional constraint and terminal cost — Discussion

Compared to the equilibrium constraint, the regional constraint

- yields easier online optimization problems
- yields larger feasible sets
- does not need exact controllability to x^*

But:

- large feasible set still needs a large optimization horizon N
- additional analytical effort for computing W
- hardly ever used in industrial practice

Regional constraint and terminal cost — Discussion

Compared to the equilibrium constraint, the regional constraint

- yields easier online optimization problems
- yields larger feasible sets
- does not need exact controllability to x^*

But:

- large feasible set still needs a large optimization horizon N
- additional analytical effort for computing W
- hardly ever used in industrial practice

In **Part 2** we will see how stability can be proved without stabilizing terminal constraints

(4) Inverse optimality and suboptimality

Performance of F_N

Once stability can be guaranteed, we can investigate the performance of the MPC feedback law F_N

Performance of F_N

Once stability can be guaranteed, we can investigate the performance of the MPC feedback law F_N

Performance of a feedback $F : X \to U$ is measured via the infinite horizon functional

$$J_\infty(x_F(0), F) := \sum_{n=0}^{\infty} \ell(x_F(n), F(x_F(n)))$$

Performance of F_N

Once stability can be guaranteed, we can investigate the performance of the MPC feedback law F_N

Performance of a feedback $F : X \to U$ is measured via the infinite horizon functional

$$J_\infty(x_F(0), F) := \sum_{n=0}^{\infty} \ell(x_F(n), F(x_F(n)))$$

Recall: $F = F_\infty$ is optimal: $J_\infty(x_{F_\infty}(0), F_\infty) = V_\infty(x_{F_N}(0))$

Performance of F_N

Once stability can be guaranteed, we can investigate the performance of the MPC feedback law F_N

Performance of a feedback $F : X \to U$ is measured via the infinite horizon functional

$$J_\infty(x_F(0), F) := \sum_{n=0}^{\infty} \ell(x_F(n), F(x_F(n)))$$

Recall: $F = F_\infty$ is optimal: $J_\infty(x_{F_\infty}(0), F_\infty) = V_\infty(x_{F_N}(0))$

In the literature, two different concepts can be found:

- Inverse Optimality: show that F_N is optimal for an altered running cost $\tilde{\ell} \neq \ell$

Performance of F_N

Once stability can be guaranteed, we can investigate the performance of the MPC feedback law F_N

Performance of a feedback $F : X \to U$ is measured via the infinite horizon functional

$$J_\infty(x_F(0), F) := \sum_{n=0}^{\infty} \ell(x_F(n), F(x_F(n)))$$

Recall: $F = F_\infty$ is optimal: $J_\infty(x_{F_\infty}(0), F_\infty) = V_\infty(x_{F_N}(0))$

In the literature, two different concepts can be found:

- Inverse Optimality: show that F_N is optimal for an altered running cost $\tilde{\ell} \neq \ell$
- Suboptimality: derive upper bounds for $J_\infty(x_{F_N}(0), F_N)$

Inverse optimality

Theorem: [Poubelle/Bitmead/Gevers '88, Magni/Sepulchre '97]

F_N is optimal for the problem

$$\text{minimize } \widetilde{J}_\infty(x(0), u) = \sum_{n=0}^{\infty} \tilde{\ell}(x(n), u(n))$$

with

$$\tilde{\ell}(x, u) := \ell(x, u) + V_{N-1}(f(x, u)) - V_N(f(x, u))$$

Inverse optimality

Theorem: [Poubelle/Bitmead/Gevers '88, Magni/Sepulchre '97]

F_N is optimal for the problem

$$\text{minimize } \widetilde{J}_\infty(x(0),u) = \sum_{n=0}^{\infty} \tilde{\ell}(x(n),u(n))$$

with

$$\tilde{\ell}(x,u) := \ell(x,u) + V_{N-1}(f(x,u)) - V_N(f(x,u))$$

Idea of proof: By the dynamic programming principle:

$$V_N(x) \quad = \quad \inf_{u \in U} \{\ell(x,u) + V_{N-1}(f(x,u))\}$$

Inverse optimality

Theorem: [Poubelle/Bitmead/Gevers '88, Magni/Sepulchre '97]

F_N is optimal for the problem

$$\text{minimize } \widetilde{J}_\infty(x(0),u) = \sum_{n=0}^{\infty} \tilde{\ell}(x(n),u(n))$$

with

$$\tilde{\ell}(x,u) := \ell(x,u) + V_{N-1}(f(x,u)) - V_N(f(x,u))$$

Idea of proof: By the dynamic programming principle:

$$\begin{aligned} V_N(x) &= \inf_{u\in U}\{\ell(x,u) + V_{N-1}(f(x,u))\} \\ &= \inf_{u\in U}\{\tilde{\ell}(x,u) + V_N(f(x,u))\} \end{aligned}$$

UNIVERSITÄT BAYREUTH

Inverse optimality

Theorem: [Poubelle/Bitmead/Gevers '88, Magni/Sepulchre '97]

F_N is optimal for the problem

$$\text{minimize } \widetilde{J}_\infty(x(0),u) = \sum_{n=0}^{\infty} \tilde{\ell}(x(n),u(n))$$

with

$$\tilde{\ell}(x,u) := \ell(x,u) + V_{N-1}(f(x,u)) - V_N(f(x,u))$$

Idea of proof: By the dynamic programming principle:

$$\begin{aligned} V_N(x) &= \inf_{u\in U}\{\ell(x,u) + V_{N-1}(f(x,u))\} \\ &= \inf_{u\in U}\{\tilde{\ell}(x,u) + V_N(f(x,u))\} \end{aligned}$$

Hence, it satisfies the Bellman equation for $\tilde{\ell}$, implying

$$\widetilde{J}_\infty(x_{F_N}(0),F_N) = V_N(x_{F_N}(0))$$

Inverse optimality

Inverse optimality

- shows that F_N is an infinite horizon optimal feedback law

Inverse optimality

Inverse optimality

- shows that F_N is an infinite horizon optimal feedback law
- thus implies several good properties of F_N, like, e.g., some inherent robustness againts perturbations

Inverse optimality

Inverse optimality

- shows that F_N is an infinite horizon optimal feedback law
- thus implies several good properties of F_N, like, e.g., some inherent robustness againts perturbations

But

- the running cost

$$\tilde{\ell}(x,u) := \ell(x,u) + V_{N-1}(f(x,u)) - V_N(f(x,u))$$

is unknown and difficult to compute

Inverse optimality

Inverse optimality

- shows that F_N is an infinite horizon optimal feedback law
- thus implies several good properties of F_N, like, e.g., some inherent robustness againts perturbations

But

- the running cost

$$\tilde{\ell}(x,u) := \ell(x,u) + V_{N-1}(f(x,u)) - V_N(f(x,u))$$

  is unknown and difficult to compute
- knowing that F_N is optimal for $\widetilde{J}_\infty(x_{F_N}(0), F_N)$ doesn't give us a simple way to estimate $J_\infty(x_{F_N}(0), F_N)$

Suboptimality

Theorem [???]: For both stabilizing terminal constraints the estimate

$$J_\infty(x_{F_N}(0), F_N) \le V_N(x_{F_N}(0))$$

holds.

UNIVERSITÄT BAYREUTH

Suboptimality

Theorem [???]: For both stabilizing terminal constraints the estimate

$$J_\infty(x_{F_N}(0), F_N) \le V_N(x_{F_N}(0))$$

holds.

Sketch of proof: Both constraints imply $V_{N-1} \ge V_N$. Hence

$$l(x_{F_N}(n), F_N(x_{F_N}(n)) \;=\; V_N(x_{F_N}(n)) - V_{N-1}(x_{F_N}(n+1))$$

UNIVERSITÄT BAYREUTH

Suboptimality

Theorem [???]: For both stabilizing terminal constraints the estimate

$$J_\infty(x_{F_N}(0), F_N) \le V_N(x_{F_N}(0))$$

holds.

Sketch of proof: Both constraints imply $V_{N-1} \ge V_N$. Hence

$$\begin{aligned} l(x_{F_N}(n), F_N(x_{F_N}(n)) &= V_N(x_{F_N}(n)) - V_{N-1}(x_{F_N}(n+1)) \\ &\le V_N(x_{F_N}(n)) - V_N(x_{F_N}(n+1)) \end{aligned}$$

Suboptimality

Theorem [???]: For both stabilizing terminal constraints the estimate

$$J_\infty(x_{F_N}(0), F_N) \le V_N(x_{F_N}(0))$$

holds.

Sketch of proof: Both constraints imply $V_{N-1} \ge V_N$. Hence

$$\begin{aligned} l(x_{F_N}(n), F_N(x_{F_N}(n)) &= V_N(x_{F_N}(n)) - V_{N-1}(x_{F_N}(n+1)) \\ &\le V_N(x_{F_N}(n)) - V_N(x_{F_N}(n+1)) \end{aligned}$$

Summing over $n = 0, \ldots, k$ yields

$$\begin{aligned} \sum_{n=0}^{k} l(x_{F_N}(n), F_N(x_{F_N}(n)) &\le V_N(x_{F_N}(0)) - V_N(x_{F_N}(k+1)) \\ &\le V_N(x_{F_N}(0)) \end{aligned}$$

Suboptimality

Theorem [???]: For both stabilizing terminal constraints the estimate

$$J_\infty(x_{F_N}(0), F_N) \le V_N(x_{F_N}(0))$$

holds.

Sketch of proof: Both constraints imply $V_{N-1} \ge V_N$. Hence

$$\begin{aligned} l(x_{F_N}(n), F_N(x_{F_N}(n)) &= V_N(x_{F_N}(n)) - V_{N-1}(x_{F_N}(n+1)) \\ &\le V_N(x_{F_N}(n)) - V_N(x_{F_N}(n+1)) \end{aligned}$$

Summing over $n = 0, \ldots, k$ yields

$$\begin{aligned} \sum_{n=0}^{k} l(x_{F_N}(n), F_N(x_{F_N}(n)) &\le V_N(x_{F_N}(0)) - V_N(x_{F_N}(k+1)) \\ &\le V_N(x_{F_N}(0)) \end{aligned}$$

Now letting $k \to \infty$ yields the assertion.

Suboptimality

Suboptimality gives us an easy to evaluate bound

$$J_\infty(x_{F_N}(0), F_N) \leq V_N(x_{F_N}(0))$$

for the infinite horizon performance of F_N.

Suboptimality

Suboptimality gives us an easy to evaluate bound

$$J_\infty(x_{F_N}(0), F_N) \leq V_N(x_{F_N}(0))$$

for the infinite horizon performance of F_N.

However, due to the terminal constraints, $V_N(x)$ can be much larger than the optimal upper bound $V_\infty(x)$.

Suboptimality

Suboptimality gives us an easy to evaluate bound

$$J_\infty(x_{F_N}(0), F_N) \le V_N(x_{F_N}(0))$$

for the infinite horizon performance of F_N.

However, due to the terminal constraints, $V_N(x)$ can be much larger than the optimal upper bound $V_\infty(x)$.

In **Part 2** we will see that MPC without stabilizing terminal constraints allows for suboptimality estimates in terms of $V_\infty(x)$.

Summary of Part 1

- MPC is an online optimal control based method for computing stabilizing feedback laws

Summary of Part 1

- MPC is an online optimal control based method for computing stabilizing feedback laws
- MPC computes the feedback law by iteratively solving finite horizon optimal control problems using the current state x_τ as initial value

Summary of Part 1

- MPC is an online optimal control based method for computing stabilizing feedback laws
- MPC computes the feedback law by iteratively solving finite horizon optimal control problems using the current state x_τ as initial value
- the feedback value $F_N(x_\tau)$ is the first element of the resulting optimal control sequence

Summary of Part 1

- MPC is an online optimal control based method for computing stabilizing feedback laws
- MPC computes the feedback law by iteratively solving finite horizon optimal control problems using the current state x_τ as initial value
- the feedback value $F_N(x_\tau)$ is the first element of the resulting optimal control sequence
- suitable terminal constraints ensure stability with V_N as Lyapunov function

Summary of Part 1

- MPC is an online optimal control based method for computing stabilizing feedback laws
- MPC computes the feedback law by iteratively solving finite horizon optimal control problems using the current state x_τ as initial value
- the feedback value $F_N(x_\tau)$ is the first element of the resulting optimal control sequence
- suitable terminal constraints ensure stability with V_N as Lyapunov function
- F_N is infinite horizon optimal for a suitably altered running cost

Summary of Part 1

- MPC is an online optimal control based method for computing stabilizing feedback laws
- MPC computes the feedback law by iteratively solving finite horizon optimal control problems using the current state x_τ as initial value
- the feedback value $F_N(x_\tau)$ is the first element of the resulting optimal control sequence
- suitable terminal constraints ensure stability with V_N as Lyapunov function
- F_N is infinite horizon optimal for a suitably altered running cost
- the infinite horizon functional along the F_N-controlled trajectory is bounded by V_N

Part 2

(5) Stability and suboptimality without stabilizing constraints

MPC without stabilizing terminal constraints

We return to the basic MPC formulation

$$\text{minimize } J_N(x(0),u) = \sum_{n=0}^{N-1} \ell(x(n),u(n)), \quad x(0) = x_\tau$$

without any stabilizing terminal constraints

MPC without stabilizing terminal constraints

We return to the basic MPC formulation

$$\text{minimize } J_N(x(0), u) = \sum_{n=0}^{N-1} \ell(x(n), u(n)), \quad x(0) = x_\tau$$

without any stabilizing terminal constraints

How can we prove stability for this setting?

UNIVERSITÄT BAYREUTH

MPC without stabilizing terminal constraints

Recall: we need to prove

$$V_N(f(x, F_N(x))) \leq V_N(x) - \tilde{\ell}(x, F_N(x))$$

for some $\tilde{\ell}(x, F_N(x)) > 0$ for $x \neq x^*$

MPC without stabilizing terminal constraints

Recall: we need to prove

$$V_N(f(x, F_N(x))) \leq V_N(x) - \tilde{\ell}(x, F_N(x))$$

for some $\tilde{\ell}(x, F_N(x)) > 0$ for $x \neq x^*$

Since by dynamic programming we have

$$\tilde{\ell}(x, F_N(x)) = \ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))),$$

this is equivalent to proving

$$\ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))) > 0$$

for $x \neq x^*$

UNIVERSITÄT
BAYREUTH

MPC without stabilizing terminal constraints

Theorem: [Alamir/Bornard '95, Jadbabaie/Hauser '05, Grimm et al. '05]
Under suitable conditions, MPC without terminal constraints stabilizes the system for sufficiently large optimization horizon N.

MPC without stabilizing terminal constraints

Theorem: [Alamir/Bornard '95, Jadbabaie/Hauser '05, Grimm et al. '05]
Under suitable conditions, MPC without terminal constraints stabilizes the system for sufficiently large optimization horizon N.

Idea of proof: Use convergence $\lim_{N\to\infty} V_N = V_\infty$ to prove

$$\ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))) \approx \ell(x, F_N(x)) > 0$$

MPC without stabilizing terminal constraints

Theorem: [Alamir/Bornard '95, Jadbabaie/Hauser '05, Grimm et al. '05]
Under suitable conditions, MPC without terminal constraints stabilizes the system for sufficiently large optimization horizon N.

Idea of proof: Use convergence $\lim_{N\to\infty} V_N = V_\infty$ to prove

$$\ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))) \approx \ell(x, F_N(x)) > 0$$

The crucial condition for sufficiently uniform convergence is

Exponential controllability "through ℓ": for real numbers $C > 0$, $\sigma \in (0, 1)$ and each $x \in X$ there exists $u(\cdot)$ with

$$\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$$

with $\ell^*(x) = \min_u \ell(x, u)$

UNIVERSITÄT BAYREUTH

MPC without stabilizing terminal constraints

Theorem: [Alamir/Bornard '95, Jadbabaie/Hauser '05, Grimm et al. '05]
Under suitable conditions, MPC without terminal constraints stabilizes the system for sufficiently large optimization horizon N.

MPC without stabilizing terminal constraints

Theorem: [Alamir/Bornard '95, Jadbabaie/Hauser '05, Grimm et al. '05]
Under suitable conditions, MPC without terminal constraints stabilizes the system for sufficiently large optimization horizon N.

Question: How large is "sufficiently large" for N?

MPC without stabilizing terminal constraints

Theorem: [Alamir/Bornard '95, Jadbabaie/Hauser '05, Grimm et al. '05]
Under suitable conditions, MPC without terminal constraints stabilizes the system for sufficiently large optimization horizon N.

Question: How large is "sufficiently large" for N?

- the first two references are non-constructive in terms of N

MPC without stabilizing terminal constraints

Theorem: [Alamir/Bornard '95, Jadbabaie/Hauser '05, Grimm et al. '05]
Under suitable conditions, MPC without terminal constraints stabilizes the system for sufficiently large optimization horizon N.

Question: How large is "sufficiently large" for N?

- the first two references are non-constructive in terms of N
- [Grimm et al.] leads to the following estimate: Let

$$\gamma := \sum_{n=0}^{\infty} C\sigma^n = \frac{C}{1-\sigma}$$

for C, σ from $\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$. Then

$$N = \mathcal{O}(\gamma^2)$$

(the constants in "$\mathcal{O}$" can be computed, if desired)

MPC without stabilizing terminal constraints

A better estimate can be obtained, if

$$\ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))) > 0$$

is established via directly estimating $|V_N - V_{N-1}|$ instead of using the detour $|V_N - V_{N-1}| \leq |V_N - V_\infty| + |V_{N-1} - V_\infty|$

MPC without stabilizing terminal constraints

A better estimate can be obtained, if

$$\ell(x, F_N(x)) + V_{N-1}(f(x, F_N(x))) - V_N(f(x, F_N(x))) > 0$$

is established via directly estimating $|V_N - V_{N-1}|$ instead of using the detour $|V_N - V_{N-1}| \leq |V_N - V_\infty| + |V_{N-1} - V_\infty|$

This way, in [Grüne/Rantzer '08] the estimate

$$N = \mathcal{O}(\gamma \log \gamma)$$

is shown, again for

$$\gamma := \sum_{n=0}^{\infty} C\sigma^n = \frac{C}{1-\sigma}$$

with C, σ from $\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$

MPC without stabilizing terminal constraints

All these estimates rely on the parameter

$$\gamma := \sum_{n=0}^{\infty} C\sigma^n = \frac{C}{1-\sigma}$$

with C, σ from $\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$

MPC without stabilizing terminal constraints

All these estimates rely on the parameter

$$\gamma := \sum_{n=0}^{\infty} C\sigma^n = \frac{C}{1-\sigma}$$

with C, σ from $\ell(x(n), u(n)) \le C\sigma^n \ell^*(x(0))$

This is because of the inequality

$$V_N(x) \le V_\infty(x) \le \gamma \ell^*(x),$$

since these estimates rely on bounds on the value functions

MPC without stabilizing terminal constraints

All these estimates rely on the parameter

$$\gamma := \sum_{n=0}^{\infty} C\sigma^n = \frac{C}{1-\sigma}$$

with C, σ from $\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$

This is because of the inequality

$$V_N(x) \leq V_\infty(x) \leq \gamma \ell^*(x),$$

since these estimates rely on bounds on the value functions

Main drawback of these approaches:

we cannot distinguish between the influence of C and σ

MPC without stabilizing terminal constraints

All these estimates rely on the parameter

$$\gamma := \sum_{n=0}^{\infty} C\sigma^n = \frac{C}{1-\sigma}$$

with C, σ from $\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$

This is because of the inequality

$$V_N(x) \leq V_\infty(x) \leq \gamma \ell^*(x),$$

since these estimates rely on bounds on the value functions

Main drawback of these approaches:

we cannot distinguish between the influence of C and σ
(or other parameters in alternative controllability conditions)

Relaxed Lyapunov inequality

We want

$$V_N(f(x, F_\infty(x))) \leq V_N(x) - \tilde{\ell}(x, F_\infty(x))$$

Relaxed Lyapunov inequality

We want

$$V_N(f(x, F_\infty(x))) \leq V_N(x) - \tilde{\ell}(x, F_\infty(x))$$

Ansatz: $\tilde{\ell} = \alpha\ell$ for $\alpha \in (0, 1]$

Relaxed Lyapunov inequality

We want

$$V_N(f(x, F_\infty(x))) \leq V_N(x) - \tilde{\ell}(x, F_\infty(x))$$

Ansatz: $\tilde{\ell} = \alpha\ell$ for $\alpha \in (0, 1]$

Theorem [Grüne/Rantzer '08]: If there exists $\alpha \in (0, 1]$ such that the "relaxed Lyapunov inequality"

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha\ell(x, F_N(x))$$

holds, then asymptotic stability follows (with V_N as Lyapunov function)

Relaxed Lyapunov inequality

We want

$$V_N(f(x, F_\infty(x))) \le V_N(x) - \tilde{\ell}(x, F_\infty(x))$$

Ansatz: $\tilde{\ell} = \alpha\ell$ for $\alpha \in (0, 1]$

Theorem [Grüne/Rantzer '08]: If there exists $\alpha \in (0, 1]$ such that the "relaxed Lyapunov inequality"

$$V_N(f(x, F_N(x))) \le V_N(x) - \alpha\ell(x, F_N(x))$$

holds, then asymptotic stability follows (with V_N as Lyapunov function) and we get the suboptimality estimate

$$J_\infty(x, F_N) \le V_\infty(x)/\alpha$$

Relaxed Lyapunov inequality

We want

$$V_N(f(x, F_\infty(x))) \leq V_N(x) - \tilde{\ell}(x, F_\infty(x))$$

Ansatz: $\tilde{\ell} = \alpha\ell$ for $\alpha \in (0, 1]$

Theorem [Grüne/Rantzer '08]: If there exists $\alpha \in (0, 1]$ such that the "relaxed Lyapunov inequality"

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha\ell(x, F_N(x))$$

holds, then asymptotic stability follows (with V_N as Lyapunov function) and we get the suboptimality estimate

$$J_\infty(x, F_N) \leq V_\infty(x)/\alpha$$

$\rightsquigarrow$ we get stability and suboptimality at once

Computing α

Goal: Compute α in the relaxed Lyapunov inequality

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha \ell(x, F_N(x))$$

Computing α

Goal: Compute α in the relaxed Lyapunov inequality

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha\ell(x, F_N(x))$$

Related approach in the literature:

estimate stability and suboptimality from numerical approximation to V_N [Shamma/Xiong '97, Primbs/Nevestic '01]

Computing α

Goal: Compute α in the relaxed Lyapunov inequality

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha \ell(x, F_N(x))$$

Related approach in the literature:

estimate stability and suboptimality from numerical approximation to V_N [Shamma/Xiong '97, Primbs/Nevestic '01]

Here: compute α analytically from the controllability property

$$\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$$

Computing α

Goal: Compute α in the relaxed Lyapunov inequality

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha\ell(x, F_N(x))$$

Related approach in the literature:

estimate stability and suboptimality from numerical approximation to V_N [Shamma/Xiong '97, Primbs/Nevestic '01]

Here: compute α analytically from the controllability property

$$\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$$

via

$$V_m(x) \leq C \sum_{k=0}^{m-1} \sigma^k \ell^*(x)$$

Computing α

Goal: Compute α in the relaxed Lyapunov inequality

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha\ell(x, F_N(x))$$

Related approach in the literature:

estimate stability and suboptimality from numerical approximation to V_N [Shamma/Xiong '97, Primbs/Nevestic '01]

Here: compute α analytically from the controllability property

$$\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$$

via
$$V_m(x) \leq C \sum_{k=0}^{m-1} \sigma^k \ell^*(x) =: B_m(x)$$

UNIVERSITÄT BAYREUTH

Computing α

Goal: Compute α in the relaxed Lyapunov inequality

$$V_N(f(x, F_N(x))) \le V_N(x) - \alpha\ell(x, F_N(x))$$

Related approach in the literature:

estimate stability and suboptimality from numerical approximation to V_N [Shamma/Xiong '97, Primbs/Nevestic '01]

Here: compute α analytically from the controllability property

$$\ell(x(n), u(n)) \le C\sigma^n \ell^*(x(0))$$

via

$$V_m(x) \le C \sum_{k=0}^{m-1} \sigma^k \ell^*(x) =: B_m(x)$$

using optimality conditions for (pieces of) trajectories

UNIVERSITÄT BAYREUTH

Computing α

The desired α–inequality

$$V_N(f(x, F_N(x))) \le V_N(x) - \alpha\ell(x, F_N(x))$$

is satisfied for all $x \in X$ iff

$$V_N(x^*(1)) \le V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0))$$

holds for all optimal trajectories $x^*(n), u^*(n)$ for V_N.

Computing α

The desired α–inequality

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha\ell(x, F_N(x))$$

is satisfied for all $x \in X$ iff

$$V_N(x^*(1)) \leq V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0))$$

holds for all optimal trajectories $x^*(n), u^*(n)$ for V_N.

From the controllability property we get:

$$V_N(x^*(1)) \quad \leq \quad B_N(x^*(1))$$

Computing α

The desired α–inequality

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha\ell(x, F_N(x))$$

is satisfied for all $x \in X$ iff

$$V_N(x^*(1)) \leq V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0))$$

holds for all optimal trajectories $x^*(n), u^*(n)$ for V_N.

From the controllability property we get:

$$V_N(x^*(1)) \leq B_N(x^*(1))$$

$$V_N(x^*(1)) \leq \ell(x^*(1), u^*(1)) + B_{N-1}(x^*(2))$$

UNIVERSITÄT BAYREUTH

Computing α

The desired α–inequality

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha\ell(x, F_N(x))$$

is satisfied for all $x \in X$ iff

$$V_N(x^*(1)) \leq V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0))$$

holds for all optimal trajectories $x^*(n), u^*(n)$ for V_N.

From the controllability property we get:

$$\begin{aligned} V_N(x^*(1)) &\leq B_N(x^*(1)) \\ V_N(x^*(1)) &\leq \ell(x^*(1), u^*(1)) + B_{N-1}(x^*(2)) \\ V_N(x^*(1)) &\leq \ell(x^*(1), u^*(1)) + \ell(x^*(2), u^*(2)) + B_{N-2}(x^*(3)) \end{aligned}$$

UNIVERSITÄT BAYREUTH

Computing α

The desired α–inequality

$$V_N(f(x, F_N(x))) \leq V_N(x) - \alpha\ell(x, F_N(x))$$

is satisfied for all $x \in X$ iff

$$V_N(x^*(1)) \leq V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0))$$

holds for all optimal trajectories $x^*(n), u^*(n)$ for V_N.

From the controllability property we get:

$$\begin{aligned}
V_N(x^*(1)) &\leq B_N(x^*(1)) \\
V_N(x^*(1)) &\leq \ell(x^*(1), u^*(1)) + B_{N-1}(x^*(2)) \\
V_N(x^*(1)) &\leq \ell(x^*(1), u^*(1)) + \ell(x^*(2), u^*(2)) + B_{N-2}(x^*(3)) \\
\vdots \quad & \quad \vdots \quad \quad \vdots
\end{aligned}$$

UNIVERSITÄT BAYREUTH

Computing α

$\rightsquigarrow V_N(x^*(1))$ is bounded by sums over $\ell(x^*(n), u^*(n))$

For sums of these values, in turn, we get bounds from the optimality principle and the controllability property

Computing α

$\rightsquigarrow V_N(x^*(1))$ is bounded by sums over $\ell(x^*(n), u^*(n))$

For sums of these values, in turn, we get bounds from the optimality principle and the controllability property:

$$\sum_{n=0}^{N-1} \ell(x^*(n), u^*(n)) \;=\; V_N(x^*(0)) \qquad \leq \; B_N(x^*(0))$$

UNIVERSITÄT BAYREUTH

Computing α

$\rightsquigarrow$ $V_N(x^*(1))$ is bounded by sums over $\ell(x^*(n), u^*(n))$

For sums of these values, in turn, we get bounds from the optimality principle and the controllability property:

$$\sum_{n=0}^{N-1} \ell(x^*(n), u^*(n)) = V_N(x^*(0)) \leq B_N(x^*(0))$$

$$\sum_{n=1}^{N-1} \ell(x^*(n), u^*(n)) = V_{N-1}(x^*(1)) \leq B_{N-1}(x^*(1))$$

UNIVERSITÄT BAYREUTH

Computing α

$\rightsquigarrow$ $V_N(x^*(1))$ is bounded by sums over $\ell(x^*(n), u^*(n))$

For sums of these values, in turn, we get bounds from the optimality principle and the controllability property:

$$\begin{aligned}
\sum_{n=0}^{N-1} \ell(x^*(n), u^*(n)) &= V_N(x^*(0)) &\leq B_N(x^*(0)) \\
\sum_{n=1}^{N-1} \ell(x^*(n), u^*(n)) &= V_{N-1}(x^*(1)) &\leq B_{N-1}(x^*(1)) \\
\sum_{n=2}^{N-1} \ell(x^*(n), u^*(n)) &= V_{N-2}(x^*(2)) &\leq B_{N-2}(x^*(2))
\end{aligned}$$

Computing α

$\rightsquigarrow$ $V_N(x^*(1))$ is bounded by sums over $\ell(x^*(n), u^*(n))$

For sums of these values, in turn, we get bounds from the optimality principle and the controllability property:

$$\begin{array}{rcccl}
\sum_{n=0}^{N-1} \ell(x^*(n), u^*(n)) & = & V_N(x^*(0)) & \leq & B_N(x^*(0)) \\
\sum_{n=1}^{N-1} \ell(x^*(n), u^*(n)) & = & V_{N-1}(x^*(1)) & \leq & B_{N-1}(x^*(1)) \\
\sum_{n=2}^{N-1} \ell(x^*(n), u^*(n)) & = & V_{N-2}(x^*(2)) & \leq & B_{N-2}(x^*(2)) \\
 & \vdots & & \vdots &
\end{array}$$

Verifying the relaxed Lyapunov inequality

Find α, such that for all optimal trajectories x^*, u^*:

$$V_N(x^*(1)) \le V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0)) \qquad (*)$$

Verifying the relaxed Lyapunov inequality

Find α, such that for all optimal trajectories x^*, u^*:

$$V_N(x^*(1)) \leq V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0)) \qquad (*)$$

Define $\lambda_n := \ell(x^*(n), u^*(n)), \quad \nu := V_N(x^*(1))$

UNIVERSITÄT BAYREUTH

Verifying the relaxed Lyapunov inequality

Find α, such that for all optimal trajectories x^*, u^*:

$$V_N(x^*(1)) \leq V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0)) \qquad (*)$$

Define $\quad \lambda_n := \ell(x^*(n), u^*(n)), \quad \nu := V_N(x^*(1))$

Then: $\quad (*) \Leftrightarrow \nu \leq \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0$

UNIVERSITÄT BAYREUTH

Verifying the relaxed Lyapunov inequality

Find α, such that for all optimal trajectories x^*, u^*:

$$V_N(x^*(1)) \leq V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0)) \qquad (*)$$

Define $\qquad \lambda_n := \ell(x^*(n), u^*(n)), \quad \nu := V_N(x^*(1))$

Then: $\quad (*) \;\Leftrightarrow\; \nu \leq \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0$

The inequalities from the last slides translate to

$$\sum_{n=k}^{N-1} \lambda_n \leq \sum_{n=0}^{N-k-1} C\sigma^n \lambda_k, \quad k = 0, \ldots, N-2 \qquad (1)$$

$$\nu \leq \sum_{n=1}^{j} \lambda_n + \lambda_{j+1} \sum_{n=0}^{N-j-1} C\sigma^n, \quad j = 0, \ldots, N-2 \qquad (2)$$

Verifying the relaxed Lyapunov inequality

Find α, such that for all optimal trajectories x^*, u^*:

$$V_N(x^*(1)) \le V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0)) \qquad (*)$$

Define $\quad \lambda_n := \ell(x^*(n), u^*(n)), \quad \nu := V_N(x^*(1))$

Then: $\quad (*) \;\Leftrightarrow\; \nu \le \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0$

The inequalities from the last slides translate to

$$\sum_{n=k}^{N-1} \lambda_n \le \sum_{n=0}^{N-k-1} C\sigma^n\lambda_k, \quad k = 0, \ldots, N-2 \qquad (1)$$

$$\nu \le \sum_{n=1}^{j} \lambda_n + \lambda_{j+1} \sum_{n=0}^{N-j-1} C\sigma^n, \quad j = 0, \ldots, N-2 \qquad (2)$$

We call $\lambda_0, \ldots, \lambda_{N-1}, \nu \ge 0$ with (1), (2) admissible

Verifying the relaxed Lyapunov inequality

Let $x^*(n), u^*(n)$ optimal trajectory for V_N

We want $V_N(x^*(1)) \leq V_N(x^*(0)) - \alpha\ell(x^*(0), u^*(0))$ $(*)$

Define $\lambda_n := \ell(x^*(n), u^*(n)), \quad \nu := V_N(x^*(1))$

Then: $(*) \Leftrightarrow \nu \leq \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0$

The inequalities from the last slides translate to

$$\sum_{n=k}^{N-1} \lambda_n \leq \sum_{n=0}^{N-k-1} C\sigma^n \lambda_k, \quad k = 0, \ldots, N-2 \qquad (1)$$

$$\nu \leq \sum_{n=1}^{j} \lambda_n + \lambda_{j+1} \sum_{n=0}^{N-j-1} C\sigma^n, \quad j = 0, \ldots, N-2 \qquad (2)$$

We call $\lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0$ with (1), (2) admissible

Stability and suboptimality condition

Theorem: [Grüne '09] Assume that all admissible $\lambda_0, \ldots, \lambda_{N-1}$, $\nu \geq 0$ satisfy

$$\nu \leq \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0 \quad \text{for some} \quad \alpha > 0,$$

Then the MPC feedback F_N stabilizes all control systems, which satisfy the controllability condition and we get $J_\infty(x, F_N) \leq V_\infty(x)/\alpha$.

Stability and suboptimality condition

Theorem: [Grüne '09] Assume that all admissible $\lambda_0, \ldots, \lambda_{N-1}$, $\nu \ge 0$ satisfy

$$\nu \le \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0 \quad \text{for some} \quad \alpha > 0,$$

Then the MPC feedback F_N stabilizes all control systems, which satisfy the controllability condition and we get $J_\infty(x, F_N) \le V_\infty(x)/\alpha$.

If, conversely, there exist admissible $\lambda_0, \ldots, \lambda_{N-1}, \nu \ge 0$ with

$$\nu \ge \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0 \quad \text{for some} \quad \alpha < 0,$$

then there exists a control system, which satisfies the controllability condition but is not stabilized by F_N.

Verifying the condition by Linear Programming

In order to apply the theorem,we need to check

$$\nu \le \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0$$

for all admissible $\lambda_0, \ldots, \lambda_{N-1}, \nu \ge 0$ and some $\alpha > 0$.

Verifying the condition by Linear Programming

In order to apply the theorem,we need to check

$$\nu \leq \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0$$

for all admissible $\lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0$ and some $\alpha > 0$.

Equivalently:

$$\text{minimize} \quad \alpha = \sum_{n=0}^{N-1} \lambda_n - \nu$$

over all admissible $\lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0$ with $\lambda_0 = 1$

Verifying the condition by Linear Programming

In order to apply the theorem,we need to check

$$\nu \leq \sum_{n=0}^{N-1} \lambda_n - \alpha\lambda_0$$

for all admissible $\lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0$ and some $\alpha > 0$.

Equivalently:

$$\text{minimize} \quad \alpha = \sum_{n=0}^{N-1} \lambda_n - \nu$$

over all admissible $\lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0$ with $\lambda_0 = 1$

This is a (small!) linear program which is explicitly solvable

Computation of stability and optimality bounds

We thus obtain the explicit formula [Grüne/Pannek/Worthmann '09]

$$\alpha = 1 - \frac{(\gamma_N - 1) \prod_{i=2}^{N} (\gamma_i - 1)}{\prod_{i=2}^{N} \gamma_i - \prod_{i=2}^{N} (\gamma_i - 1)} \quad \text{with} \quad \gamma_i = \sum_{k=0}^{i-1} C\sigma^k$$

depending on the optimization horizon N and the parameters C, σ in

$$\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$$

Computation of stability and optimality bounds

We thus obtain the explicit formula [Grüne/Pannek/Worthmann '09]

$$\alpha = 1 - \frac{(\gamma_N - 1)\prod_{i=2}^{N}(\gamma_i - 1)}{\prod_{i=2}^{N}\gamma_i - \prod_{i=2}^{N}(\gamma_i - 1)} \quad \text{with} \quad \gamma_i = \sum_{k=0}^{i-1} C\sigma^k$$

depending on the optimization horizon N and the parameters C, σ in

$$\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$$

In particular, for given α_0 we can compute the minimal horizon N with $\alpha > \alpha_0$

Computation of stability and optimality bounds

We thus obtain the explicit formula [Grüne/Pannek/Worthmann '09]

$$\alpha = 1 - \frac{(\gamma_N - 1)\prod_{i=2}^{N}(\gamma_i - 1)}{\prod_{i=2}^{N}\gamma_i - \prod_{i=2}^{N}(\gamma_i - 1)} \quad \text{with} \quad \gamma_i = \sum_{k=0}^{i-1} C\sigma^k$$

depending on the optimization horizon N and the parameters C, σ in

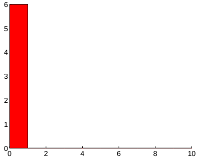
$$\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$$

In particular, for given α_0 we can compute the minimal horizon N with $\alpha > \alpha_0$

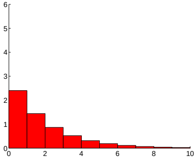
We illustrate this for $\alpha_0 = 0$, i.e., for the minimal stabilizing horizon

Horizon depending on C and σ

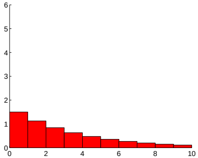
Horizons N for different C, σ with $\sum_{n=0}^{\infty} C\sigma^n = 6$:



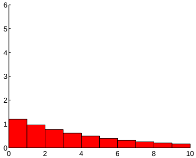
$C = 6$, $\sigma = 0$



$C = 12/5$, $\sigma = 3/5$



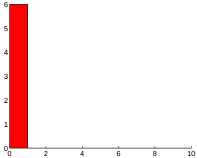
$C = 3/2$, $\sigma = 3/4$



$C = 6/5$, $\sigma = 4/5$

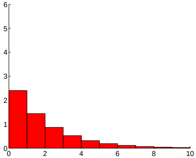
Horizon depending on C and σ

Horizons N for different C, σ with $\sum_{n=0}^{\infty} C\sigma^n = 6$:

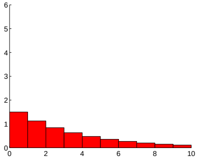


$C = 6$, $\sigma = 0$

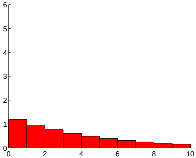
$N = 11$



$C = 12/5$, $\sigma = 3/5$



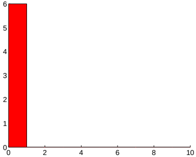
$C = 3/2$, $\sigma = 3/4$



$C = 6/5$, $\sigma = 4/5$

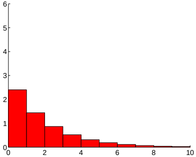
Horizon depending on C and σ

Horizons N for different C, σ with $\sum_{n=0}^{\infty} C\sigma^n = 6$:



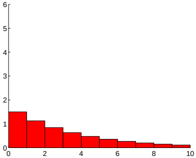
$C = 6$, $\sigma = 0$

$N = 11$

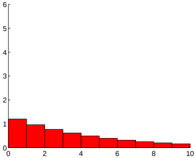


$C = 12/5$, $\sigma = 3/5$

$N = 10$



$C = 3/2$, $\sigma = 3/4$

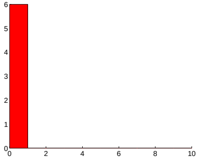


$C = 6/5$, $\sigma = 4/5$

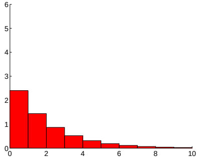
UNIVERSITÄT BAYREUTH

Horizon depending on C and σ

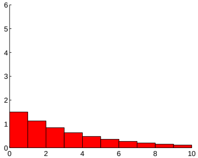
Horizons N for different C, σ with $\sum_{n=0}^{\infty} C\sigma^n = 6$:



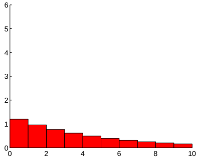
$C = 6$, $\sigma = 0$
$N = 11$



$C = 12/5$, $\sigma = 3/5$
$N = 10$



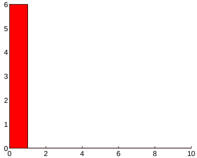
$C = 3/2$, $\sigma = 3/4$
$N = 7$



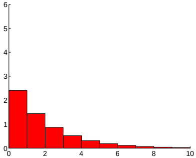
$C = 6/5$, $\sigma = 4/5$

Horizon depending on C and σ

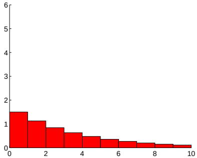
Horizons N for different C, σ with $\sum_{n=0}^{\infty} C\sigma^n = 6$:



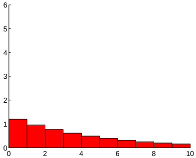
$C = 6$, $\sigma = 0$
$N = 11$



$C = 12/5$, $\sigma = 3/5$
$N = 10$

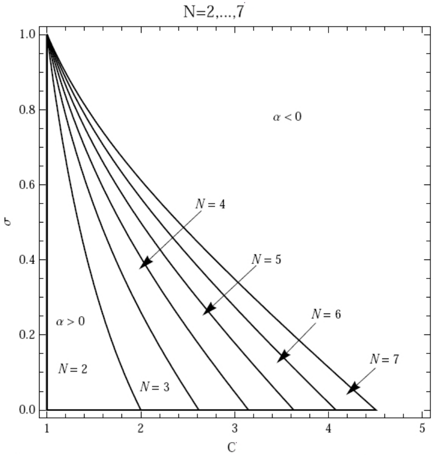


$C = 3/2$, $\sigma = 3/4$
$N = 7$



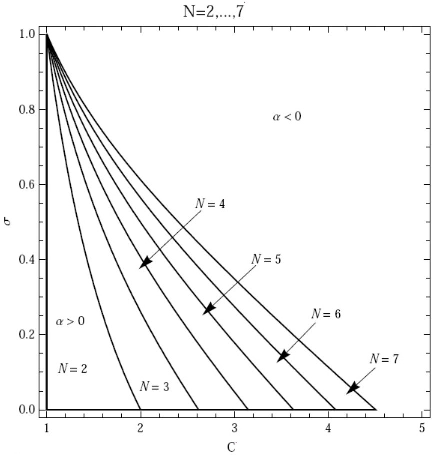
$C = 6/5$, $\sigma = 4/5$
$N = 4$

Stability chart for C and σ



(Figure: Harald Voit)

Stability chart for C and σ



(Figure: Harald Voit)

Conclusion: for short optimization horizon N it is
more important: small C (“small overshoot”)
less important: small σ (“fast decay”)

Other types of controllability condition

The procedure is easily extended to the more general controllability condition:

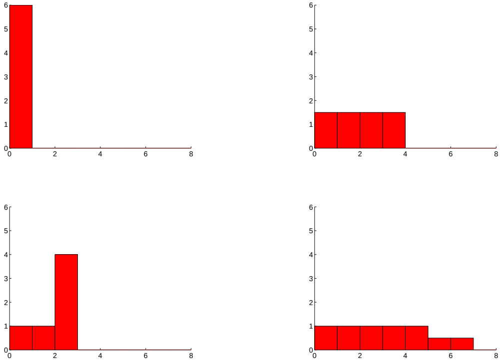
for a sequence $(c_n)_{n\in\mathbb{N}_0}$ with $c_n \to 0$ and every $x \in X$ there exists $u(\cdot)$ with

$$\ell(x(n), u(n)) \leq c_n \ell^*(x(0)), \qquad n = 0, 1, 2, \ldots$$

with $\ell^*(x) = \min_u \ell(x, u)$ (as before)

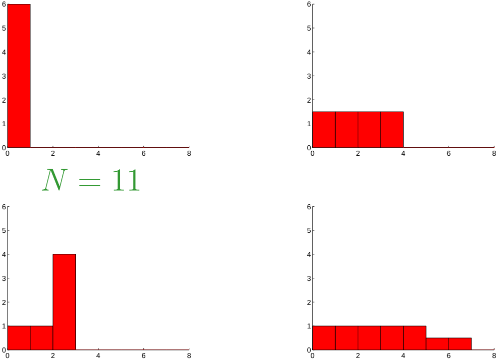
Horizons for finite time controllability

Horizons N for different c_n with $\sum c_n = 6$:



Horizons for finite time controllability

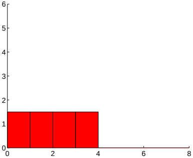
Horizons N for different c_n with $\sum c_n = 6$:



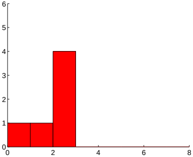
UNIVERSITÄT BAYREUTH

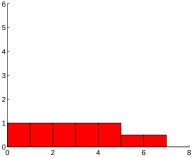
Horizons for finite time controllability

Horizons N for different c_n with $\sum c_n = 6$:



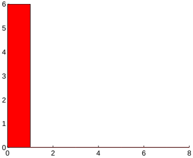
$N = 11$ | $N = 10$



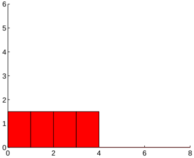


Horizons for finite time controllability

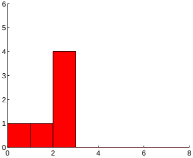
Horizons N for different c_n with $\sum c_n = 6$:



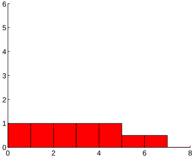
$N = 11$



$N = 10$



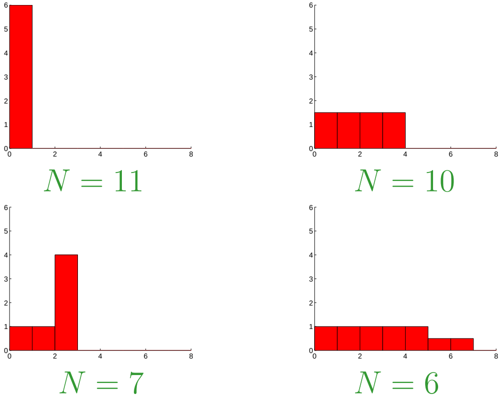
$N = 7$



UNIVERSITÄT BAYREUTH

Horizons for finite time controllability

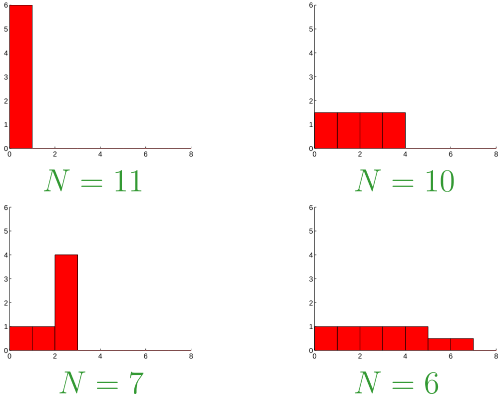
Horizons N for different c_n with $\sum c_n = 6$:



UNIVERSITÄT BAYREUTH

Horizons for finite time controllability

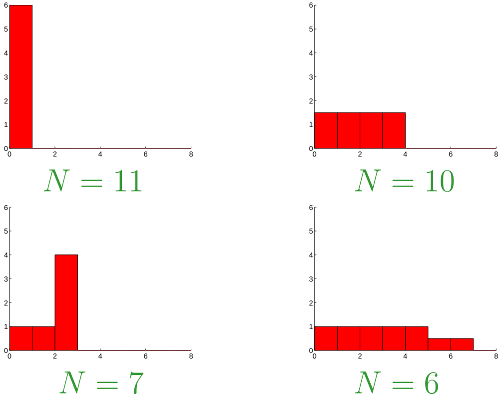
Horizons N for different c_n with $\sum c_n = 6$:



$\rightsquigarrow$ for obtaining short horizons smaller (and later) overshoot is more important than fast controllability

Horizons for finite time controllability

Horizons N for different c_n with $\sum c_n = 6$:



$\rightsquigarrow$ for obtaining short horizons smaller (and later) overshoot is more important than fast controllability

we can use this for the design of ℓ

(6) Examples for the design of MPC schemes

Design of “good” MPC running costs ℓ

We want small overshoot C in the estimate

$$\ell(x(n), u(n)) \le C\sigma^n \ell^*(x(0))$$

or, more generally, small values c_n in

$$\ell(x(n), u(n)) \le c_n \ell^*(x(0))$$

Design of “good” MPC running costs ℓ

We want small overshoot C in the estimate

$$\ell(x(n), u(n)) \le C\sigma^n \ell^*(x(0))$$

or, more generally, small values c_n in

$$\ell(x(n), u(n)) \le c_n \ell^*(x(0))$$

The trajectories $x(n)$ are given

Design of "good" MPC running costs ℓ

We want small overshoot C in the estimate

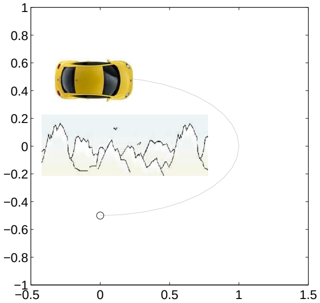
$$\ell(x(n), u(n)) \leq C\sigma^n \ell^*(x(0))$$

or, more generally, small values c_n in

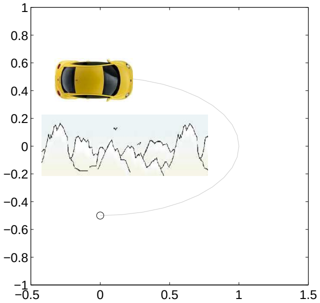
$$\ell(x(n), u(n)) \leq c_n \ell^*(x(0))$$

The trajectories $x(n)$ are given, but we can use the running cost ℓ as design parameter

The car-and-mountains example reloaded



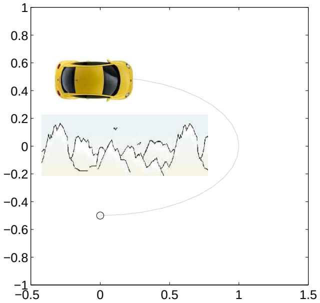
The car-and-mountains example reloaded



MPC with $\ell(x,u) = \|x - x^*\|^2 + |u|^2$ and $u_{\max} = 0.2$

$\rightsquigarrow$ asymptotic stability for $N = 11$ but not for $N \leq 10$

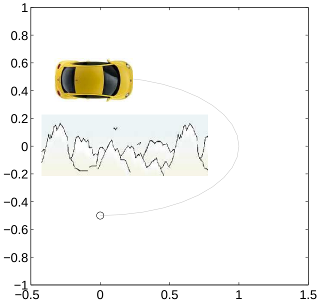
The car-and-mountains example reloaded



MPC with $\ell(x,u) = \|x - x^*\|^2 + |u|^2$ and $u_{\max} = 0.2$
$\rightsquigarrow$ asymptotic stability for $N = 11$ but not for $N \le 10$

Reason: detour around mountains causes large overshoot C

The car-and-mountains example reloaded



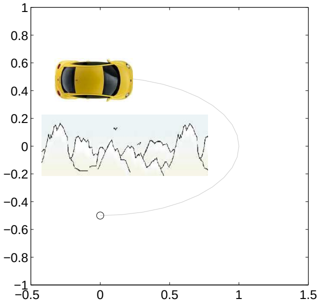
MPC with $\ell(x,u) = \|x - x^*\|^2 + |u|^2$ and $u_{\max} = 0.2$
$\rightsquigarrow$ asymptotic stability for $N = 11$ but not for $N \leq 10$

Reason: detour around mountains causes large overshoot C

Remedy: put larger weight on x_2:

$$\ell(x,u) = (x_1 - x_1^*)^2 + 5(x_2 - x_2^*)^2 + |u|^2$$

The car-and-mountains example reloaded



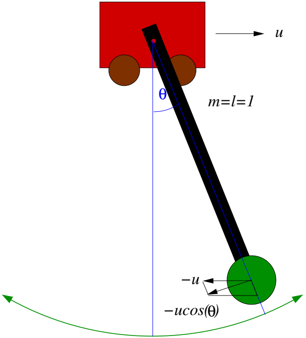
MPC with $\ell(x,u) = \|x - x^*\|^2 + |u|^2$ and $u_{\max} = 0.2$
$\rightsquigarrow$ asymptotic stability for $N = 11$ but not for $N \leq 10$

Reason: detour around mountains causes large overshoot C

Remedy: put larger weight on x_2:

$\ell(x,u) = (x_1 - x_1^*)^2 + 5(x_2 - x_2^*)^2 + |u|^2$ $\rightsquigarrow$ as. stab. for $N = 2$

Example: pendulum on a cart



$x_1 = \theta =$ angle

$x_2 =$ angular velocity

$x_3 =$ cart position

$x_4 =$ cart velocity

$u \;=$ cart acceleration

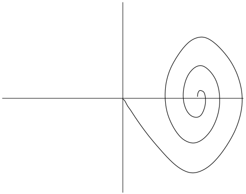
⇝ control system

$$\begin{aligned}\dot x_1 &= x_2(t)\\ \dot x_2 &= -g\sin(x_1) - kx_2\\ &\quad -u\cos(x_1)\\ \dot x_3 &= x_4\\ \dot x_4 &= u\end{aligned}$$

Example: Inverted Pendulum

Reducing overshoot for swingup of the pendulum on a cart:

$$\begin{aligned} \dot{x}_1 &= x_2, & \dot{x}_2 &= -g\sin(x_1) - kx_2 - u\cos(x_1) \\ \dot{x}_3 &= x_4, & \dot{x}_4 &= u \end{aligned}$$



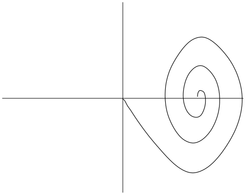
Typical swingup trajectory
x_1 and x_2 component

Example: Inverted Pendulum

Reducing overshoot for swingup of the pendulum on a cart:

$$\dot{x}_1 = x_2, \qquad \dot{x}_2 = -g\sin(x_1) - kx_2 - u\cos(x_1)$$
$$\dot{x}_3 = x_4, \qquad \dot{x}_4 = u$$

Let $\ell(x) = \sqrt{\ell_1(x_1, x_2) + x_3^2 + x_4^2}$



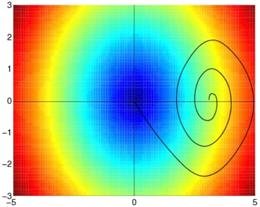
Typical swingup trajectory
x_1 and x_2 component

Example: Inverted Pendulum

Reducing overshoot for swingup of the pendulum on a cart:

$$\dot{x}_1 = x_2, \qquad \dot{x}_2 = -g\sin(x_1) - kx_2 - u\cos(x_1)$$
$$\dot{x}_3 = x_4, \qquad \dot{x}_4 = u$$

Let $\ell(x) = \sqrt{\ell_1(x_1,x_2) + x_3^2 + x_4^2}$ with



$\ell_1(x_1,x_2) = x_1^2 + x_2^2$

$N = 15$

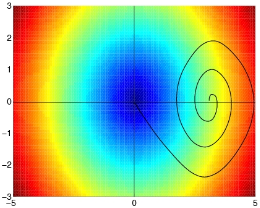
sampling time $T = 0.15$

Example: Inverted Pendulum

Reducing overshoot for swingup of the pendulum on a cart:

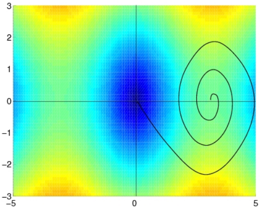
$$\dot{x}_1 = x_2, \qquad \dot{x}_2 = -g\sin(x_1) - kx_2 - u\cos(x_1)$$
$$\dot{x}_3 = x_4, \qquad \dot{x}_4 = u$$

Let $\ell(x) = \sqrt{\ell_1(x_1, x_2) + x_3^2 + x_4^2}$ with



$\ell_1(x_1, x_2) = x_1^2 + x_2^2$

$N = 15$



$4(1 - \cos x_1) + x_2^2$

$N = 10$

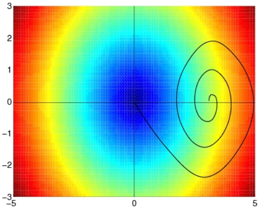
sampling time $T = 0.15$

Example: Inverted Pendulum

Reducing overshoot for swingup of the pendulum on a cart:

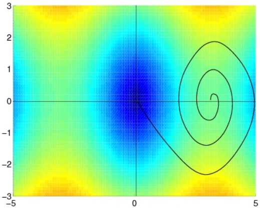
$$\dot{x}_1 = x_2, \qquad \dot{x}_2 = -g\sin(x_1) - kx_2 - u\cos(x_1)$$
$$\dot{x}_3 = x_4, \qquad \dot{x}_4 = u$$

Let $\ell(x) = \sqrt{\ell_1(x_1,x_2) + x_3^2 + x_4^2}$ with



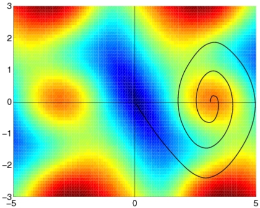
$\ell_1(x_1,x_2) = x_1^2 + x_2^2$

$N = 15$



$4(1-\cos x_1) + x_2^2$

$N = 10$



$(\sin x_1, x_2)P(\sin x_1, x_2)^T$
$+2((1-\cos x_1)(1-\cos x_2)^2)^2$

$N = 4$ (swingup only)

sampling time $T = 0.15$

A PDE example

Our results are also applicable for infinite dimensional system

A PDE example

Our results are also applicable for infinite dimensional system

We illustrate this with the 1d controlled PDE

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y) + u$$

with

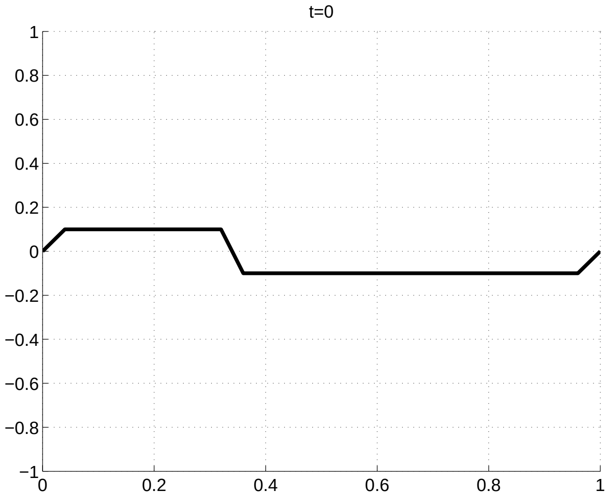
domain $\Omega = [0,1]$

solution $y = y(t,x)$

boundary conditions $y(t,0) = y(t,1) = 0$

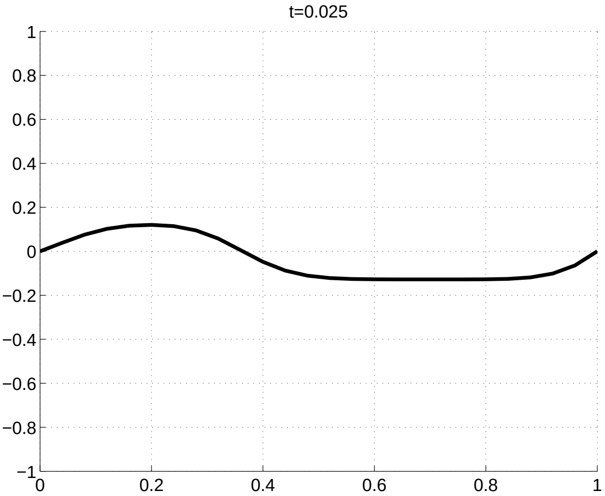
parameters $\nu = 0.1$ and $\mu = 10$

The uncontrolled PDE



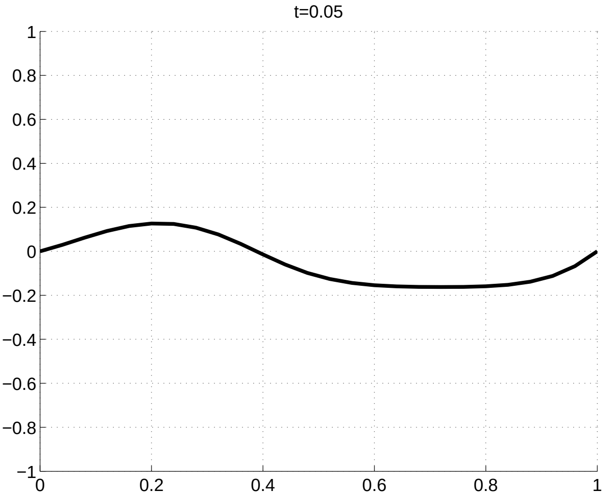
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



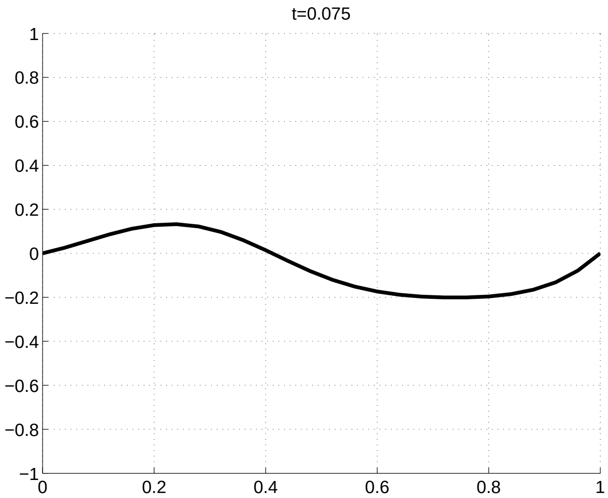
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



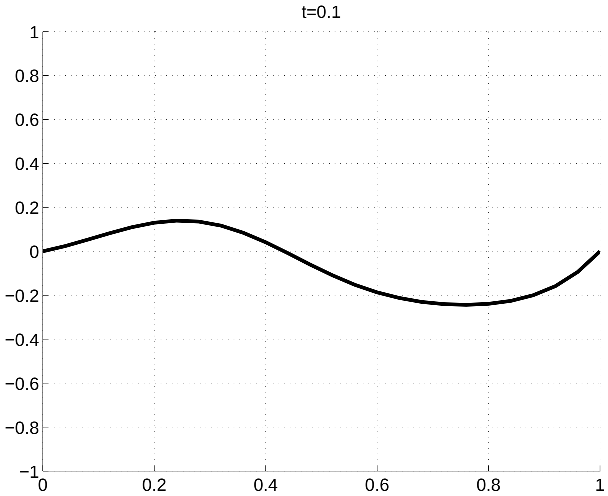
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



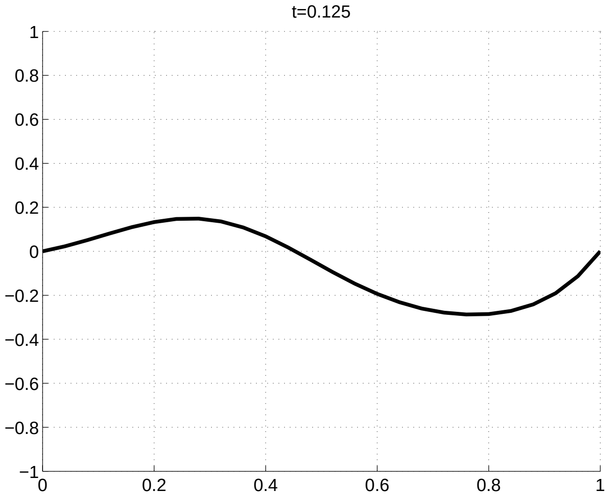
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



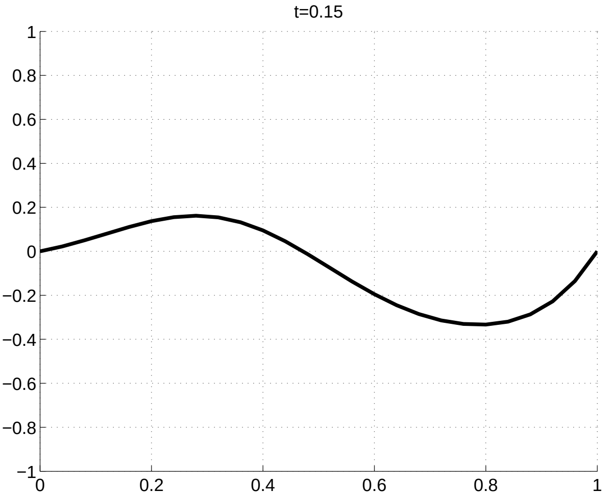
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



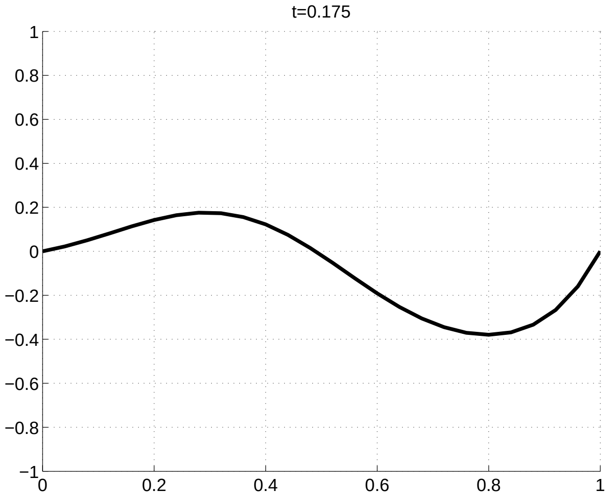
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



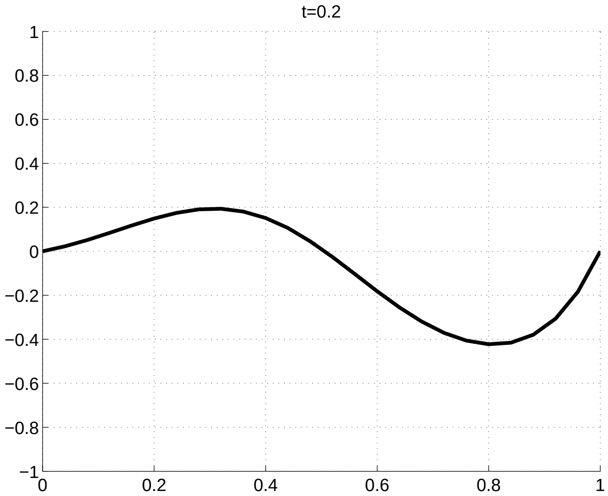
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



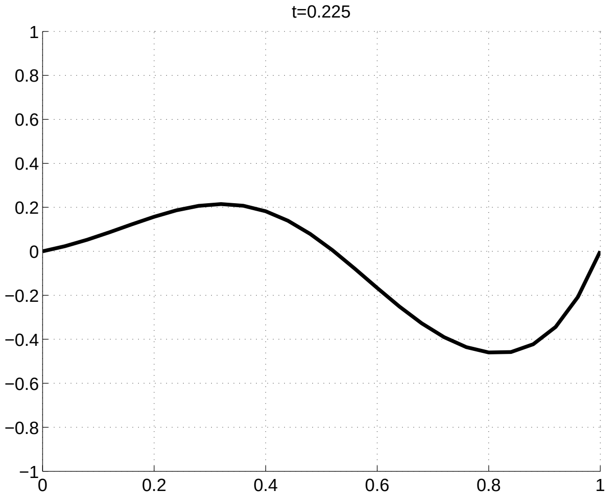
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



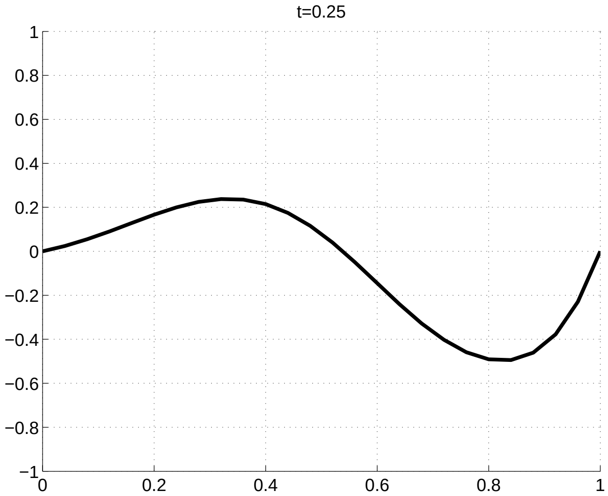
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



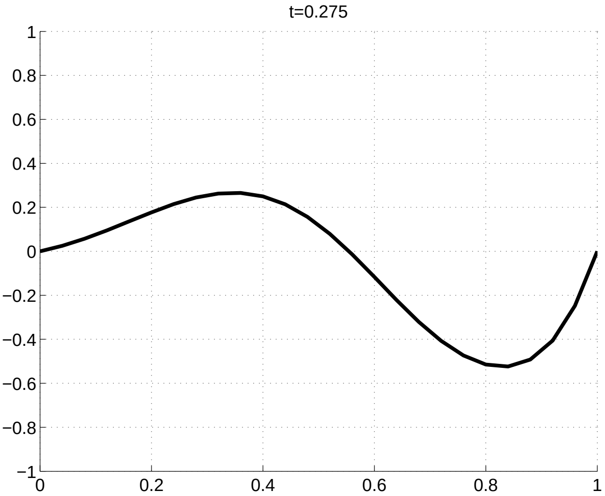
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



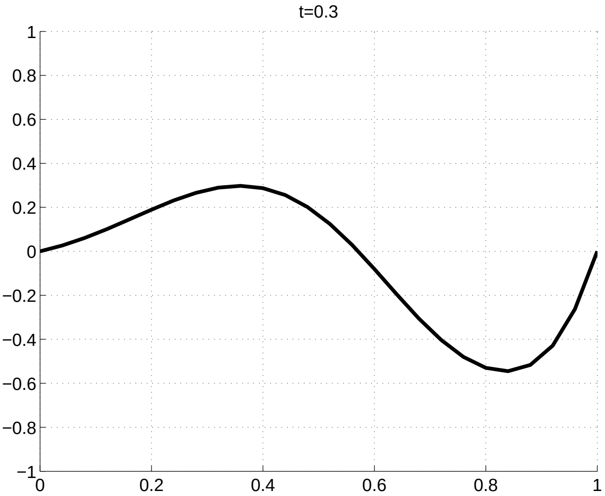
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



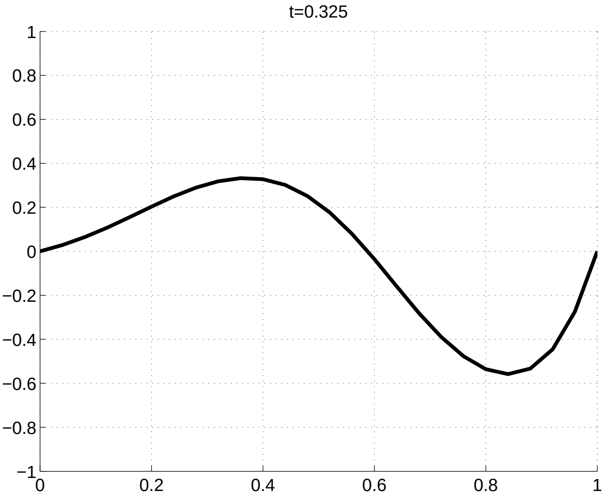
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



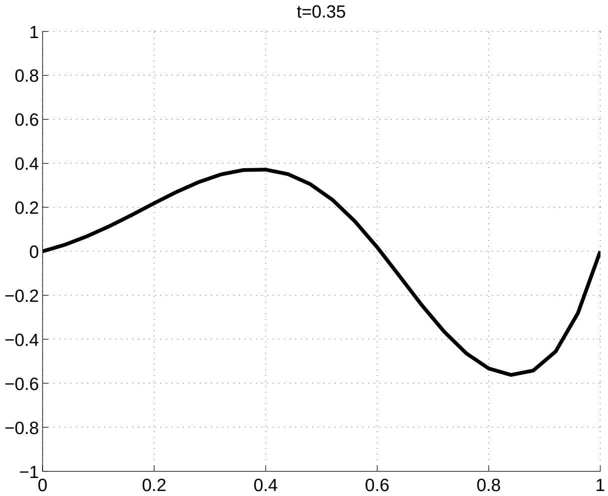
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



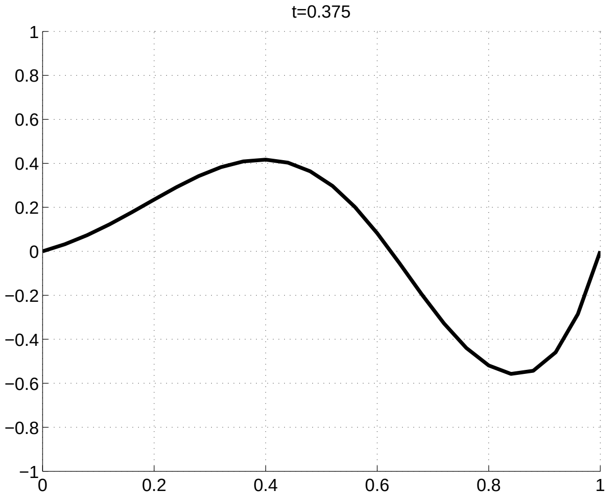
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



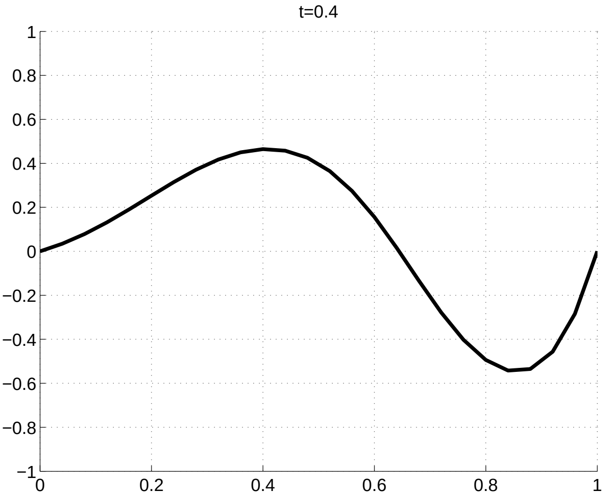
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



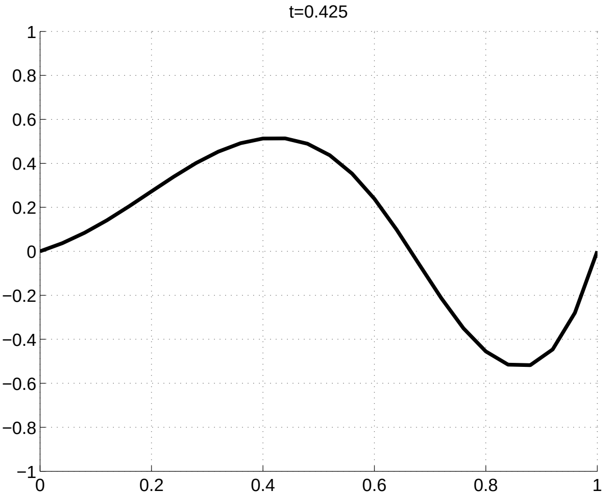
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



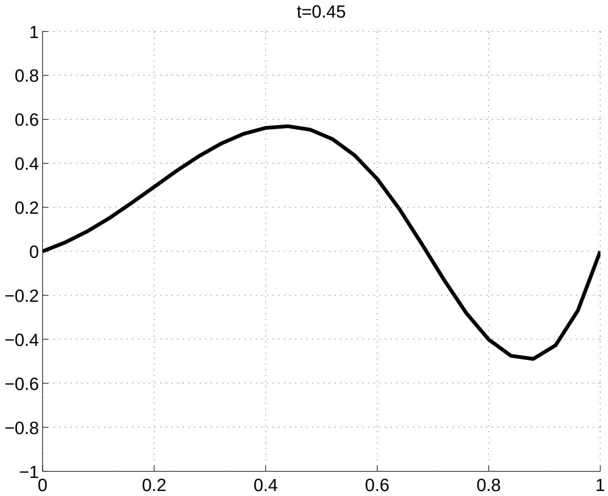
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



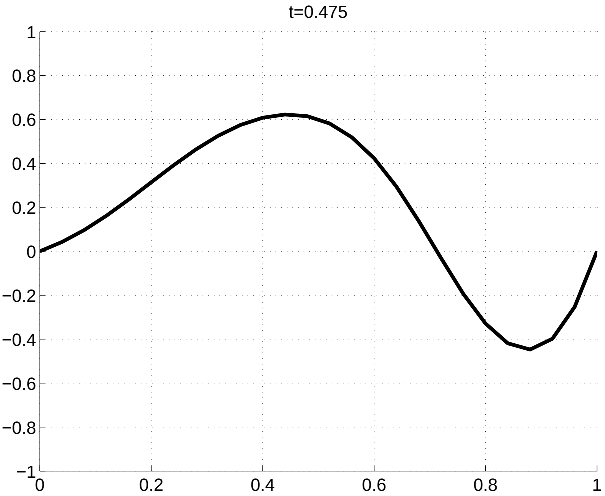
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



uncontrolled ($u \equiv 0$)

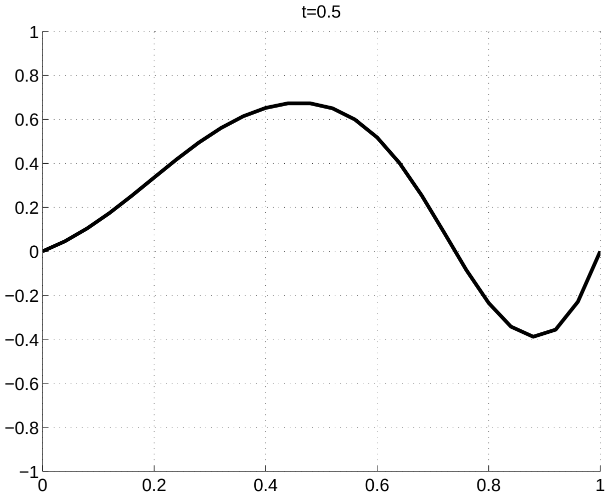
The uncontrolled PDE



uncontrolled ($u \equiv 0$)

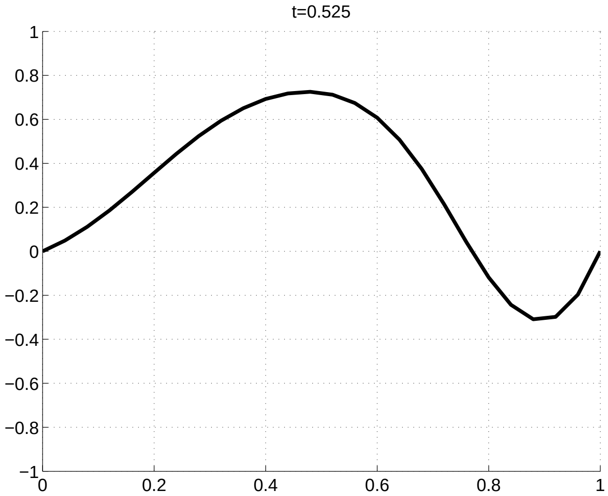
UNIVERSITÄT BAYREUTH

The uncontrolled PDE



uncontrolled ($u \equiv 0$)

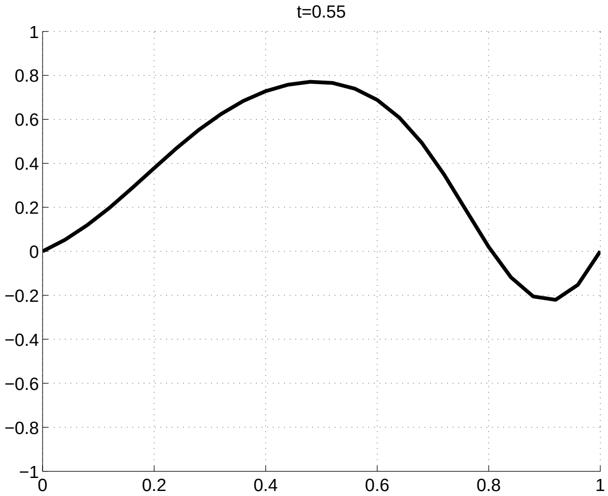
The uncontrolled PDE



uncontrolled ($u \equiv 0$)

UNIVERSITÄT BAYREUTH

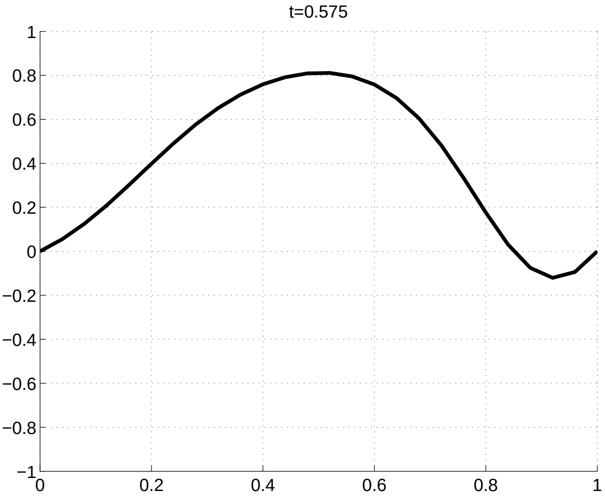
The uncontrolled PDE



uncontrolled ($u \equiv 0$)

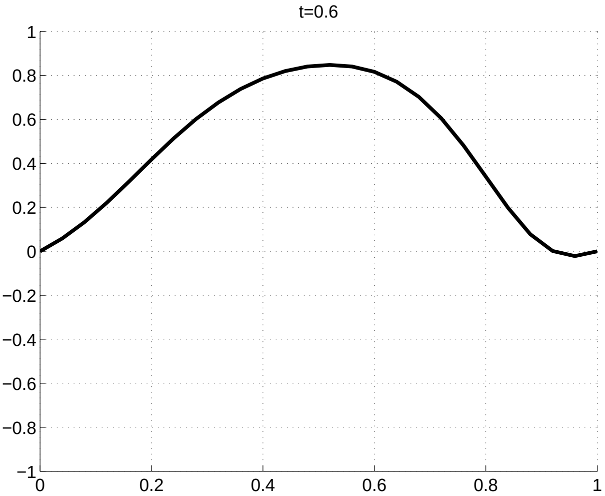
UNIVERSITÄT BAYREUTH

The uncontrolled PDE



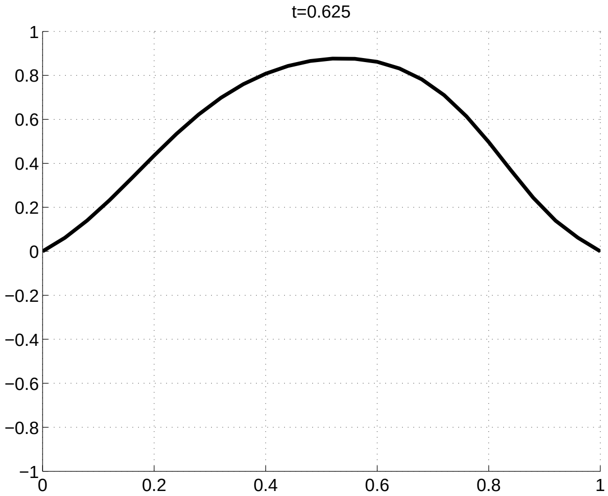
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



uncontrolled ($u \equiv 0$)

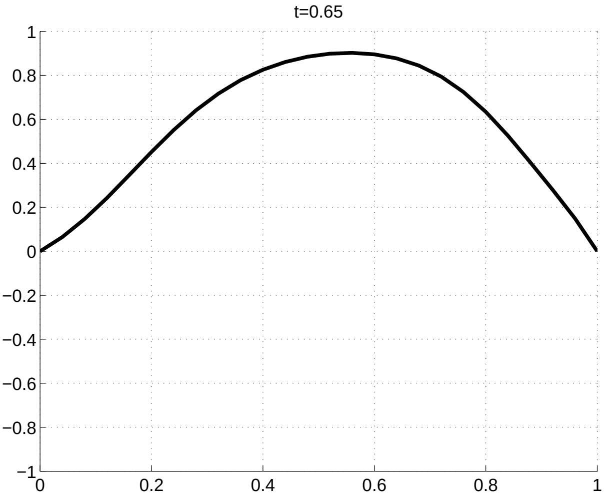
The uncontrolled PDE



uncontrolled ($u \equiv 0$)

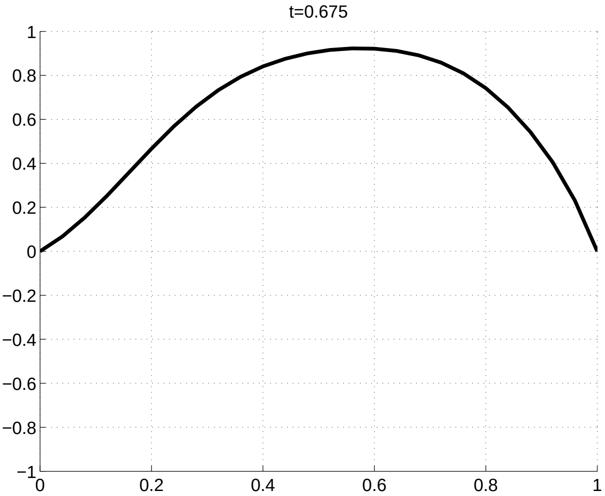
UNIVERSITÄT BAYREUTH

The uncontrolled PDE



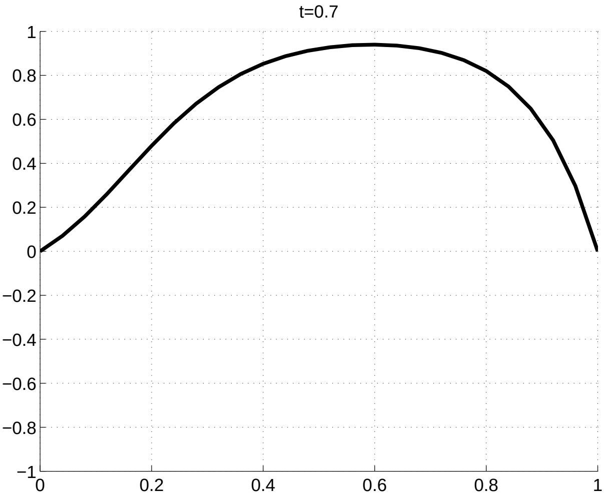
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



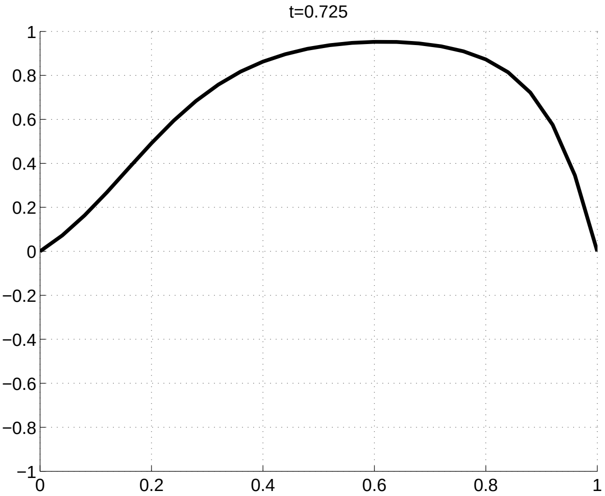
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



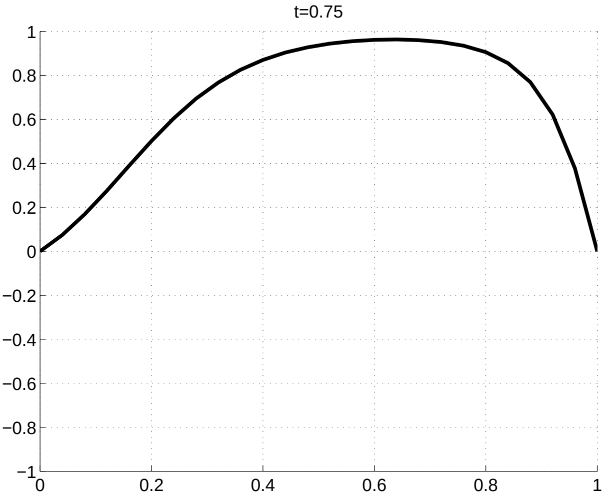
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



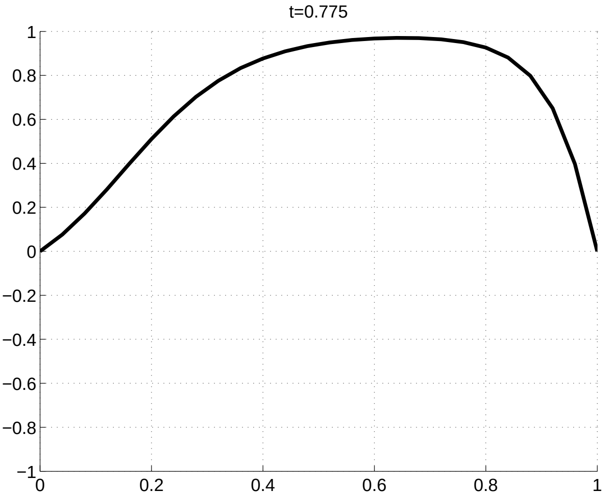
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



uncontrolled ($u \equiv 0$)

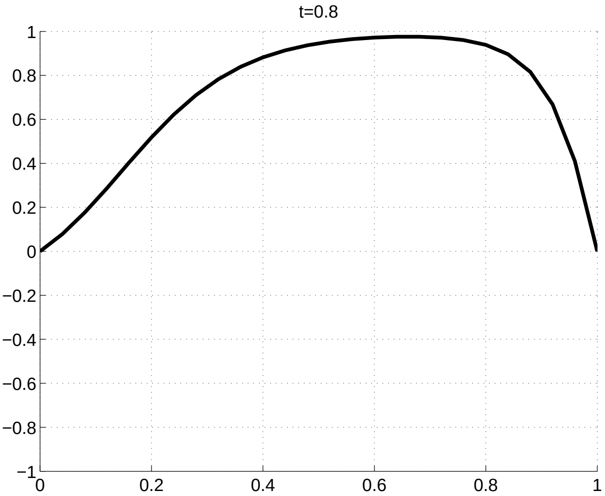
The uncontrolled PDE



uncontrolled ($u \equiv 0$)

UNIVERSITÄT BAYREUTH

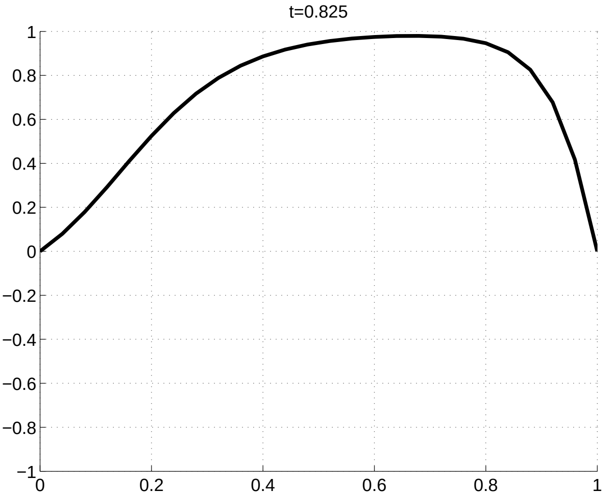
The uncontrolled PDE



uncontrolled ($u \equiv 0$)

UNIVERSITÄT BAYREUTH

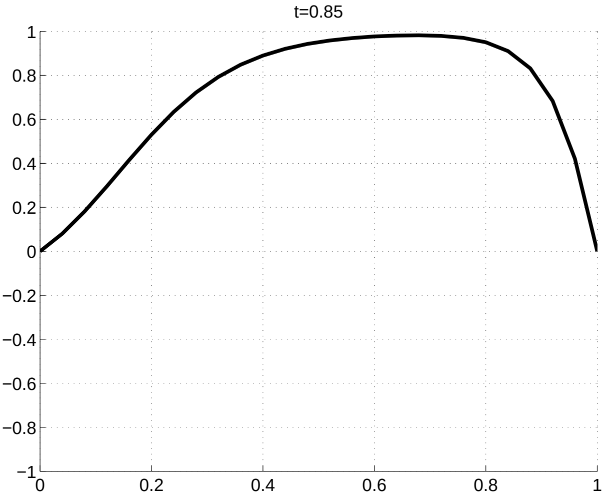
The uncontrolled PDE



uncontrolled ($u \equiv 0$)

UNIVERSITÄT BAYREUTH

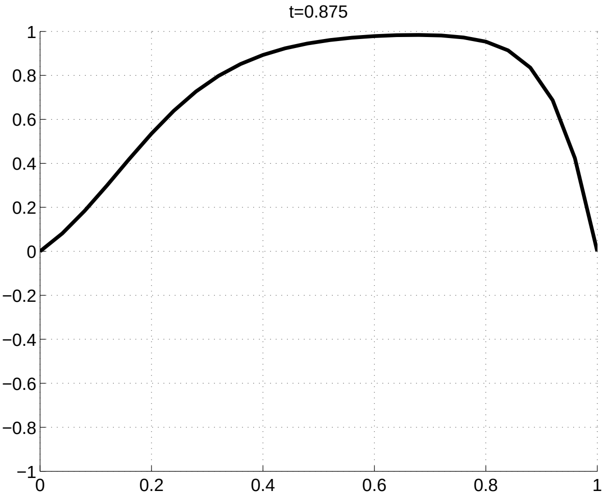
The uncontrolled PDE



uncontrolled ($u \equiv 0$)

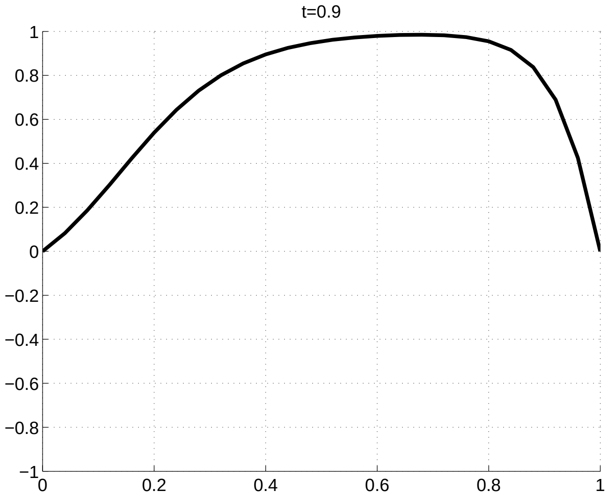
UNIVERSITÄT BAYREUTH

The uncontrolled PDE



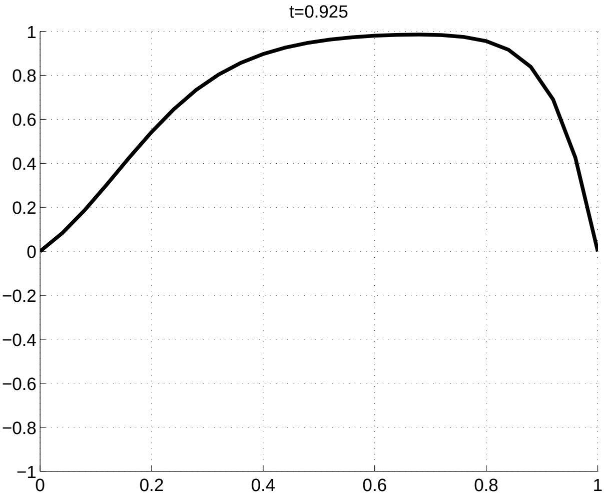
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



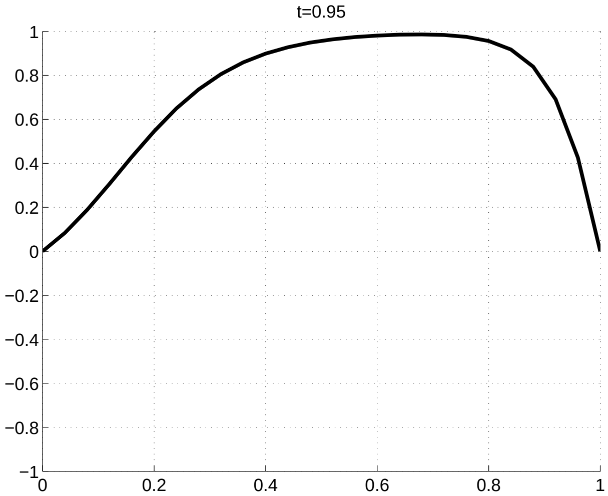
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



uncontrolled ($u \equiv 0$)

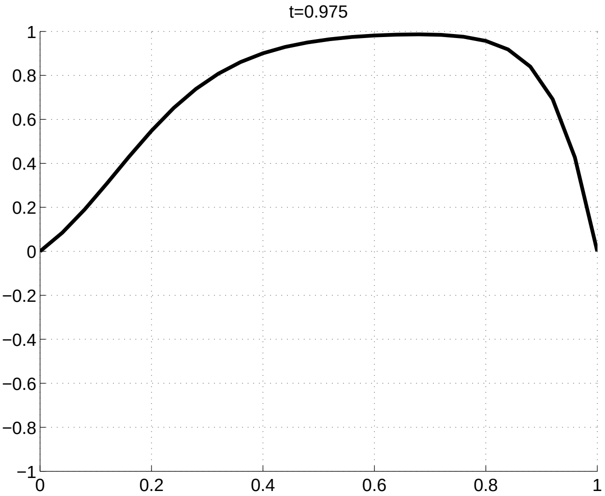
The uncontrolled PDE



uncontrolled ($u \equiv 0$)

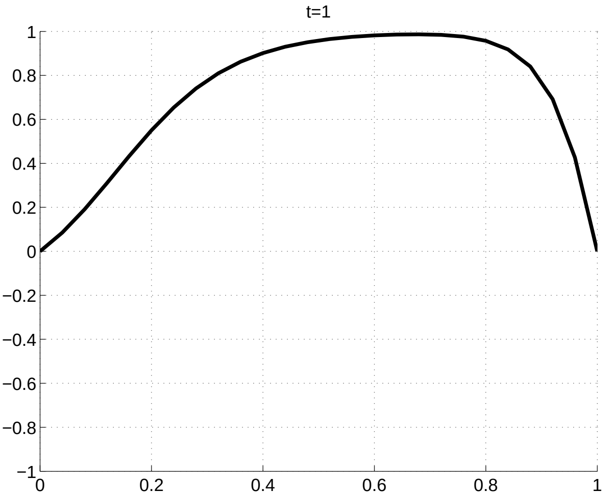
UNIVERSITÄT BAYREUTH

The uncontrolled PDE



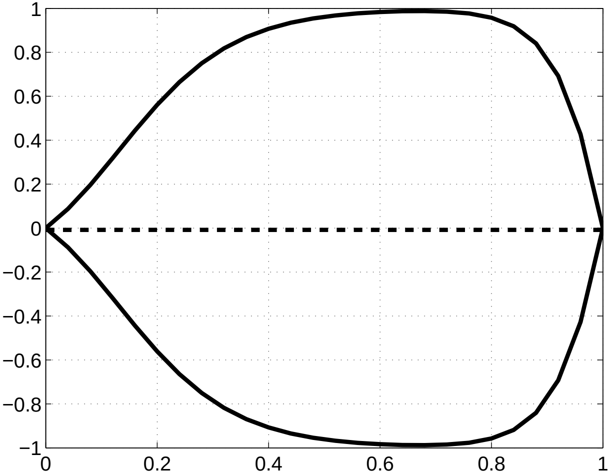
uncontrolled ($u \equiv 0$)

The uncontrolled PDE



uncontrolled ($u \equiv 0$)

The uncontrolled PDE



all equilibrium solutions

MPC for the PDE example

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y) + u$$

MPC for the PDE example

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y) + u$$

Goal: stabilize the sampled data solution $y(n, \cdot)$ at $y \equiv 0$

MPC for the PDE example

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y) + u$$

Goal: stabilize the sampled data solution $y(n, \cdot)$ at $y \equiv 0$

Usual approach: quadratic L^2 cost

$$\ell(y(n,\cdot), u(n,\cdot)) = \|y(n,\cdot)\|_{L^2}^2 + \lambda \|u(n,\cdot)\|_{L^2}^2$$

MPC for the PDE example

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y) + u$$

Goal: stabilize the sampled data solution $y(n,\cdot)$ at $y \equiv 0$

Usual approach: quadratic L^2 cost

$$\ell(y(n,\cdot), u(n,\cdot)) = \|y(n,\cdot)\|_{L^2}^2 + \lambda \|u(n,\cdot)\|_{L^2}^2$$

For $y \approx 0$ the control u must compensate for $y_x \rightsquigarrow u \approx -y_x$

MPC for the PDE example

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y) + u$$

Goal: stabilize the sampled data solution $y(n,\cdot)$ at $y \equiv 0$

Usual approach: quadratic L^2 cost

$$\ell(y(n,\cdot),u(n,\cdot)) = \|y(n,\cdot)\|^2_{L^2} + \lambda\|u(n,\cdot)\|^2_{L^2}$$

For $y \approx 0$ the control u must compensate for $y_x \rightsquigarrow u \approx -y_x$

$\rightsquigarrow$ controllability condition

$$\ell(y(n,\cdot),u(n,\cdot)) \quad \leq \quad C\sigma^n \ell^*(y(0,\cdot))$$

MPC for the PDE example

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y) + u$$

Goal: stabilize the sampled data solution $y(n,\cdot)$ at $y \equiv 0$

Usual approach: quadratic L^2 cost

$$\ell(y(n,\cdot), u(n,\cdot)) = \|y(n,\cdot)\|_{L^2}^2 + \lambda \|u(n,\cdot)\|_{L^2}^2$$

For $y \approx 0$ the control u must compensate for $y_x \rightsquigarrow u \approx -y_x$

$\rightsquigarrow$ controllability condition

$$\begin{aligned} \ell(y(n,\cdot), u(n,\cdot)) &\leq C\sigma^n \ell^*(y(0,\cdot)) \\ \Leftrightarrow \quad \|y(n,\cdot)\|_{L^2}^2 + \lambda \|u(n,\cdot)\|_{L^2}^2 &\leq C\sigma^n \|y(0,\cdot)\|_{L^2}^2 \end{aligned}$$

UNIVERSITÄT BAYREUTH

MPC for the PDE example

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y) + u$$

Goal: stabilize the sampled data solution $y(n,\cdot)$ at $y \equiv 0$

Usual approach: quadratic L^2 cost

$$\ell(y(n,\cdot),u(n,\cdot)) = \|y(n,\cdot)\|_{L^2}^2 + \lambda\|u(n,\cdot)\|_{L^2}^2$$

For $y \approx 0$ the control u must compensate for $y_x \rightsquigarrow u \approx -y_x$

$\rightsquigarrow$ controllability condition

$$\begin{aligned} \ell(y(n,\cdot),u(n,\cdot)) &\le C\sigma^n \ell^*(y(0,\cdot)) \\ \Leftrightarrow \quad \|y(n,\cdot)\|_{L^2}^2 + \lambda\|u(n,\cdot)\|_{L^2}^2 &\le C\sigma^n \|y(0,\cdot)\|_{L^2}^2 \\ \approx \quad \|y(n,\cdot)\|_{L^2}^2 + \lambda\|y_x(n,\cdot)\|_{L^2}^2 &\le C\sigma^n \|y(0,\cdot)\|_{L^2}^2 \end{aligned}$$

MPC for the PDE example

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y) + u$$

Goal: stabilize the sampled data solution $y(n,\cdot)$ at $y \equiv 0$

Usual approach: quadratic L^2 cost

$$\ell(y(n,\cdot),u(n,\cdot)) = \|y(n,\cdot)\|^2_{L^2} + \lambda\|u(n,\cdot)\|^2_{L^2}$$

For $y \approx 0$ the control u must compensate for $y_x \rightsquigarrow u \approx -y_x$

$\rightsquigarrow$ controllability condition

$$\begin{aligned} & \ell(y(n,\cdot),u(n,\cdot)) & \le & \; C\sigma^n \ell^*(y(0,\cdot)) \\ \Leftrightarrow \quad & \|y(n,\cdot)\|^2_{L^2} + \lambda\|u(n,\cdot)\|^2_{L^2} & \le & \; C\sigma^n \|y(0,\cdot)\|^2_{L^2} \\ \approx \quad & \|y(n,\cdot)\|^2_{L^2} + \lambda\|y_x(n,\cdot)\|^2_{L^2} & \le & \; C\sigma^n \|y(0,\cdot)\|^2_{L^2} \end{aligned}$$

for $\|y_x\|_{L^2} \gg \|y\|_{L^2}$ this can only hold if $C \gg 0$

UNIVERSITÄT BAYREUTH

MPC for the PDE example

Conclusion: because of

$$\|y(n,\cdot)\|_{L^2}^2 + \lambda\|y_x(n,\cdot)\|_{L^2}^2 \le C\sigma^n\|y(0,\cdot)\|_{L^2}^2$$

the controllability condition may only hold for very large C

MPC for the PDE example

Conclusion: because of

$$\|y(n,\cdot)\|_{L^2}^2 + \lambda\|y_x(n,\cdot)\|_{L^2}^2 \le C\sigma^n\|y(0,\cdot)\|_{L^2}^2$$

the controllability condition may only hold for very large C

Remedy: use H^1 cost

$$\ell(y(n,\cdot),u(n,\cdot)) = \underbrace{\|y(n,\cdot)\|_{L^2}^2 + \|y_x(n,\cdot)\|_{L^2}^2}_{=\|y(n,\cdot)\|_{H^1}^2} + \lambda\|u(n,\cdot)\|_{L^2}^2.$$

MPC for the PDE example

Conclusion: because of

$$\|y(n,\cdot)\|_{L^2}^2 + \lambda\|y_x(n,\cdot)\|_{L^2}^2 \le C\sigma^n\|y(0,\cdot)\|_{L^2}^2$$

the controllability condition may only hold for very large C

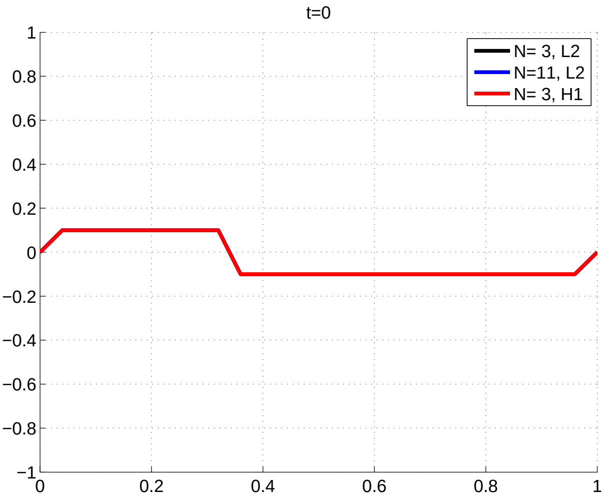
Remedy: use H^1 cost

$$\ell(y(n,\cdot),u(n,\cdot)) = \underbrace{\|y(n,\cdot)\|_{L^2}^2 + \|y_x(n,\cdot)\|_{L^2}^2}_{=\|y(n,\cdot)\|_{H^1}^2} + \lambda\|u(n,\cdot)\|_{L^2}^2.$$

Then an analogous computation yields

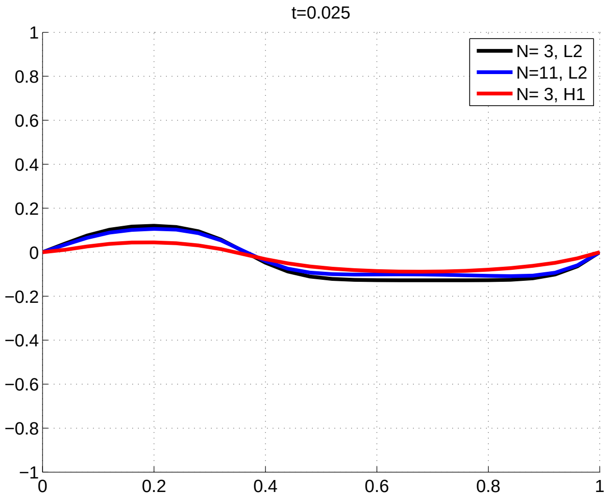
$$\|y(n,\cdot)\|_{L^2}^2 + (1+\lambda)\|y_x(n,\cdot)\|_{L^2}^2 \le C\sigma^n\Big(\|y(0,\cdot)\|_{L^2}^2 + \|y_x(0,\cdot)\|_{L^2}^2\Big)$$

MPC with L_2 vs. H_1 cost



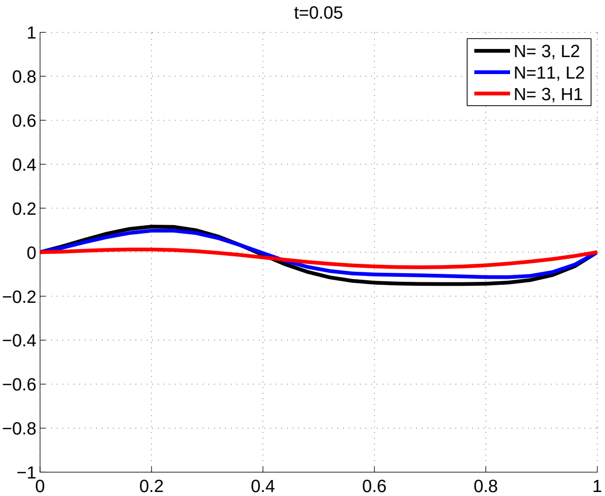
MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

MPC with L_2 vs. H_1 cost



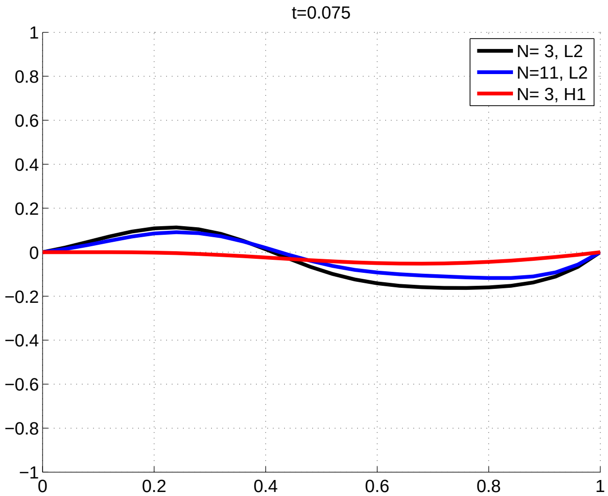
MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

MPC with L_2 vs. H_1 cost



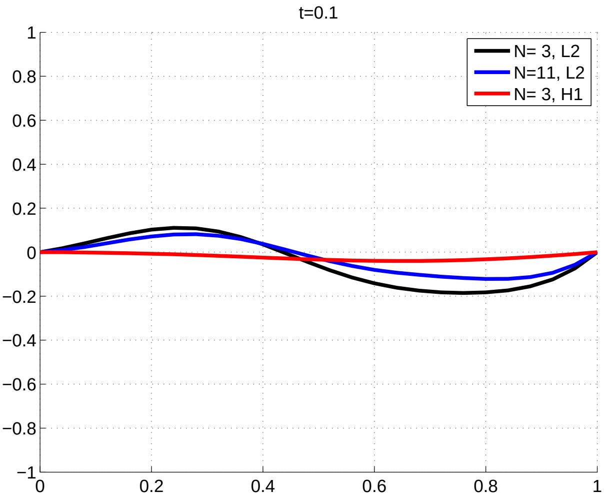
MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

MPC with L_2 vs. H_1 cost



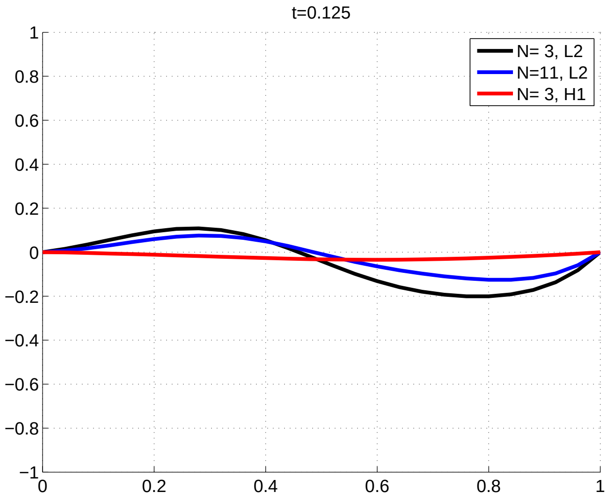
MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

MPC with L_2 vs. H_1 cost



MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

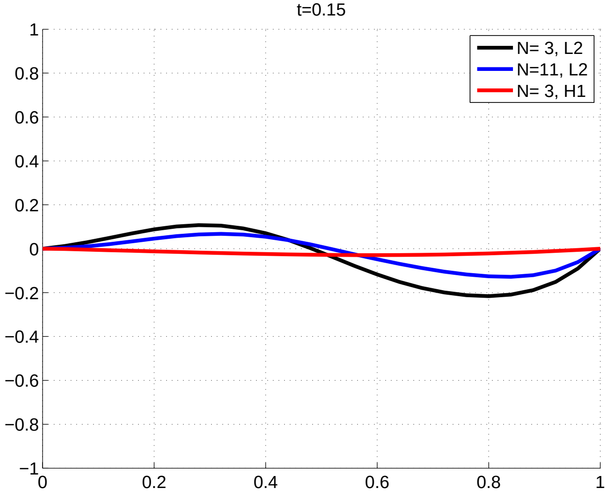
MPC with L_2 vs. H_1 cost



MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

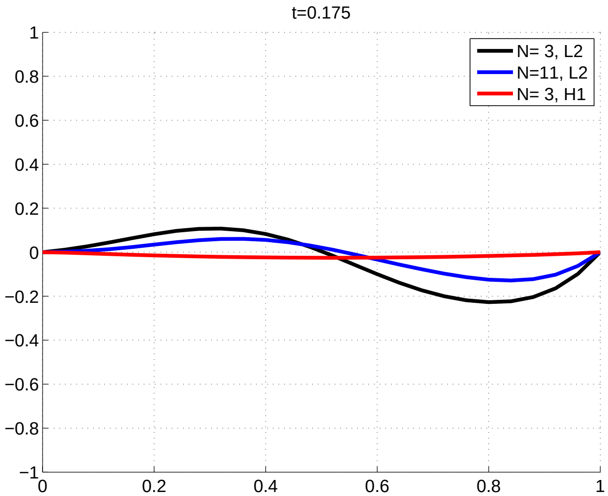
UNIVERSITÄT BAYREUTH

MPC with L_2 vs. H_1 cost



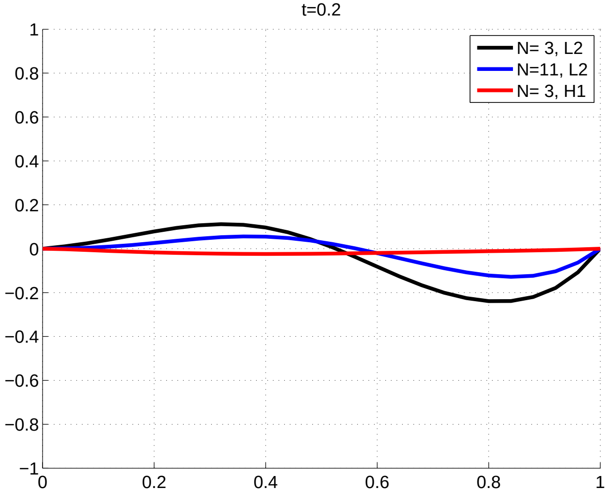
MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

MPC with L_2 vs. H_1 cost



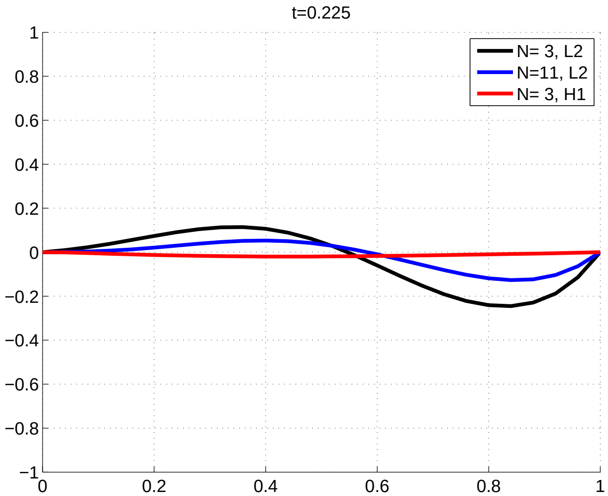
MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

MPC with L_2 vs. H_1 cost



MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

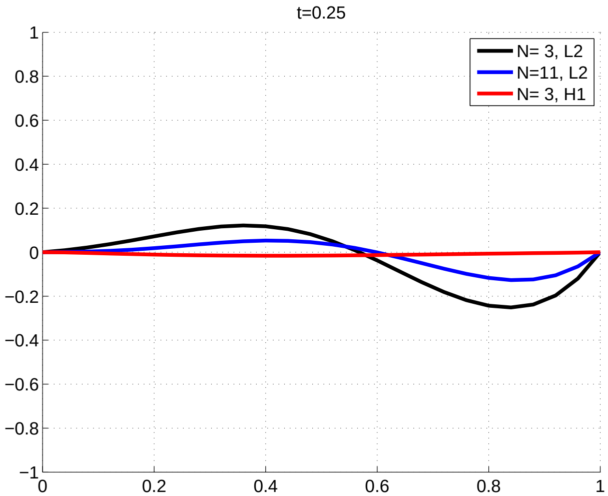
MPC with L_2 vs. H_1 cost



MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

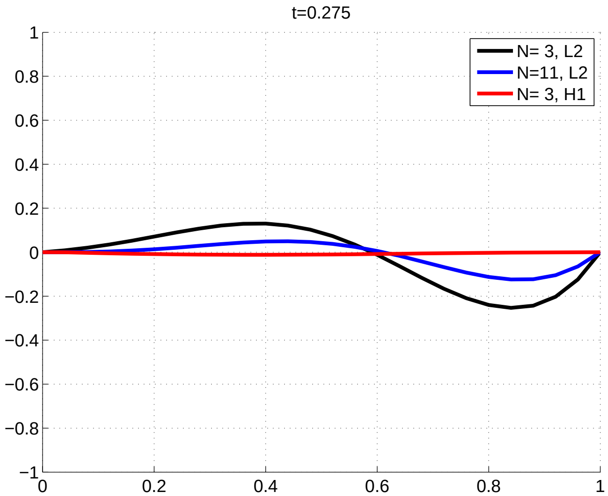
UNIVERSITÄT BAYREUTH

MPC with L_2 vs. H_1 cost



MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

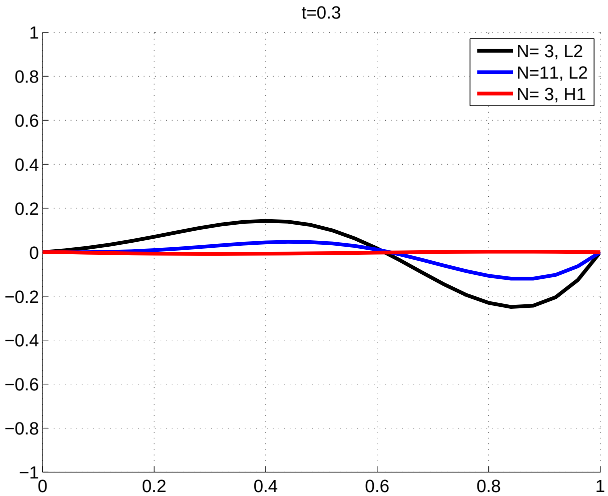
MPC with L_2 vs. H_1 cost



MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

MPC with L_2 vs. H_1 cost



MPC with L_2 and H_1 cost, $\lambda = 0.1$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

Boundary Control

Now we change our PDE from distributed to (Dirichlet-) boundary control, i.e.

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y)$$

with

domain $\Omega = [0,1]$

solution $y = y(t,x)$

boundary conditions $y(t,0) = u_0(t)$, $y(t,1) = u_1(t)$

parameters $\nu = 0.1$ and $\mu = 10$

Boundary Control

Now we change our PDE from distributed to (Dirichlet-) boundary control, i.e.

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y)$$

with

domain $\Omega = [0,1]$

solution $y = y(t,x)$

boundary conditions $y(t,0) = u_0(t)$, $y(t,1) = u_1(t)$

parameters $\nu = 0.1$ and $\mu = 10$

with boundary control, stability can only be achieved via large gradients in the transient phase

Boundary Control

Now we change our PDE from distributed to (Dirichlet-) boundary control, i.e.

$$y_t = y_x + \nu y_{xx} + \mu y(y+1)(1-y)$$

with

domain $\Omega = [0,1]$

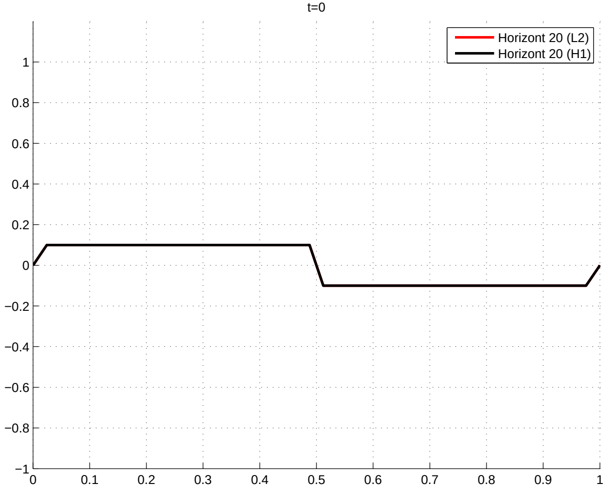
solution $y = y(t,x)$

boundary conditions $y(t,0) = u_0(t)$, $y(t,1) = u_1(t)$

parameters $\nu = 0.1$ and $\mu = 10$

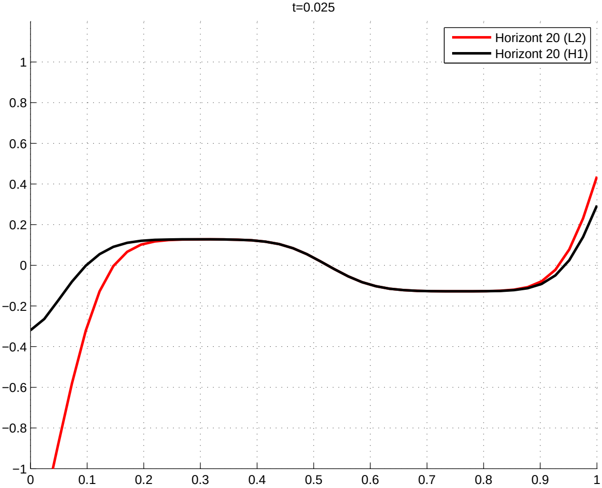
with boundary control, stability can only be achieved via large gradients in the transient phase
$\rightsquigarrow$ L_2 should perform better that H_1

Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

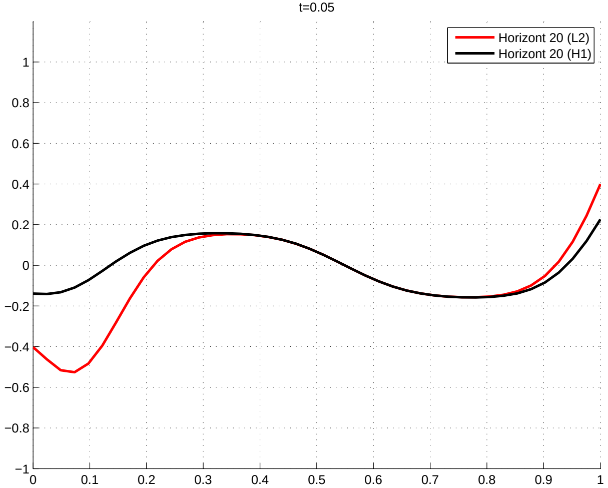
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

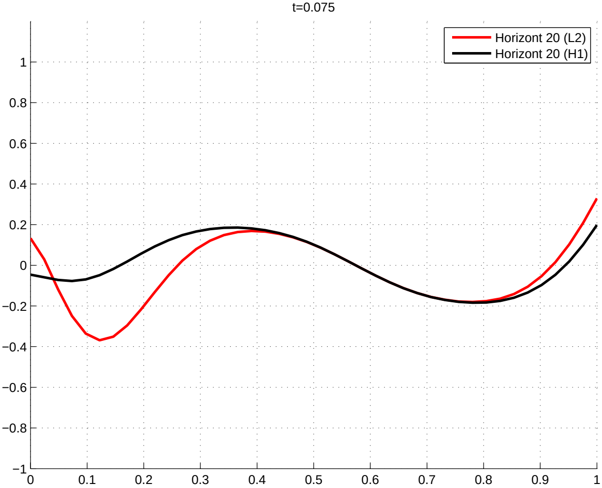
UNIVERSITÄT BAYREUTH

Boundary control, L_2 vs. H_1, $N = 20$



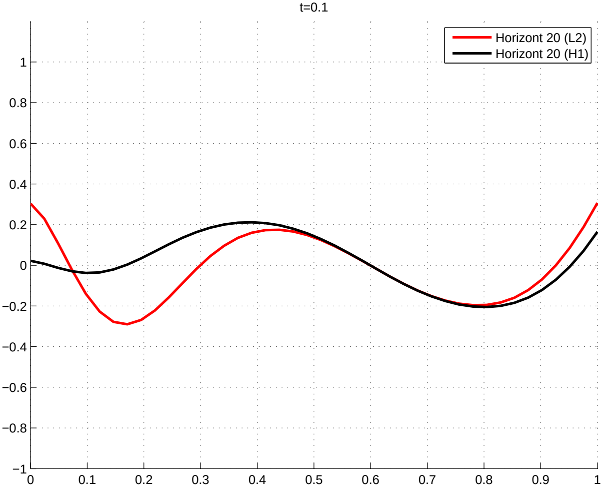
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2 vs. H_1, $N = 20$



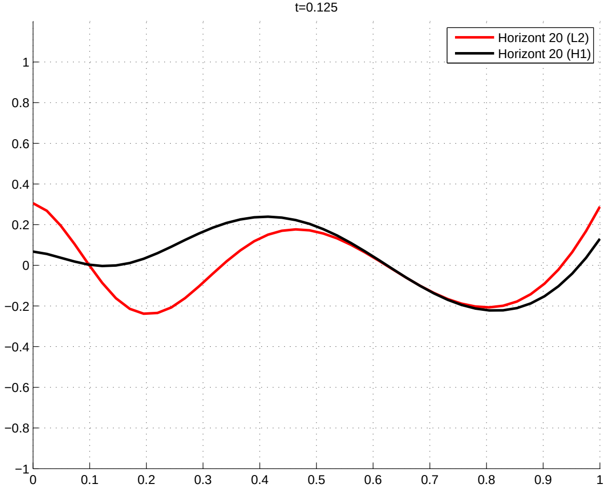
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2 vs. H_1, $N = 20$



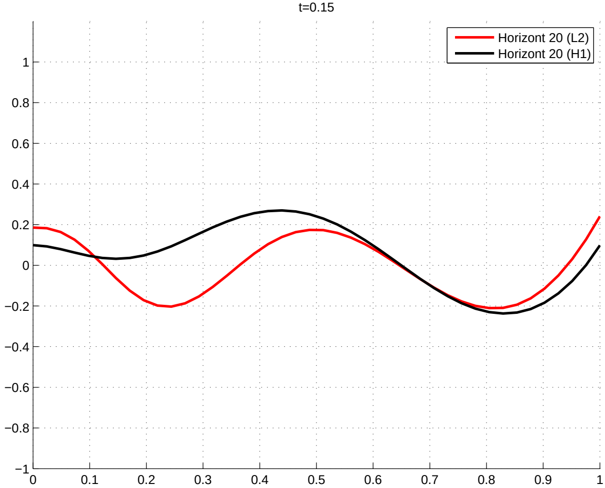
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

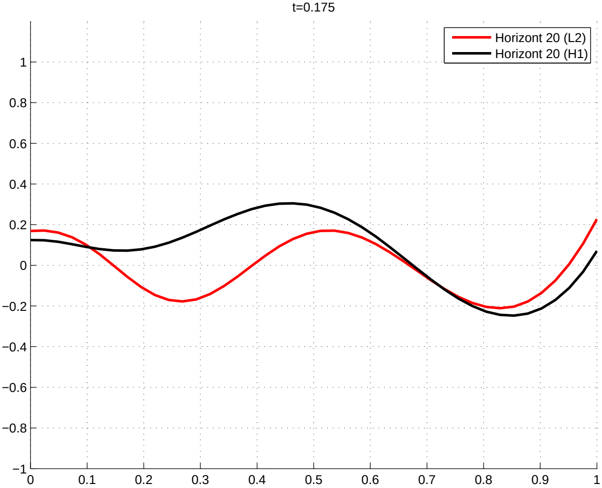
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

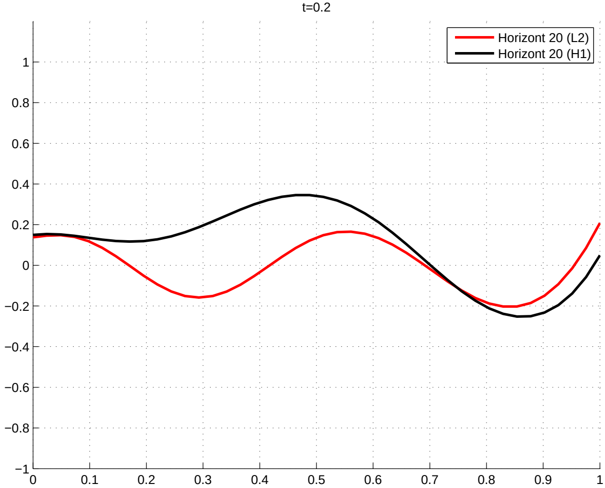
UNIVERSITÄT BAYREUTH

Boundary control, L_2 vs. H_1, $N = 20$



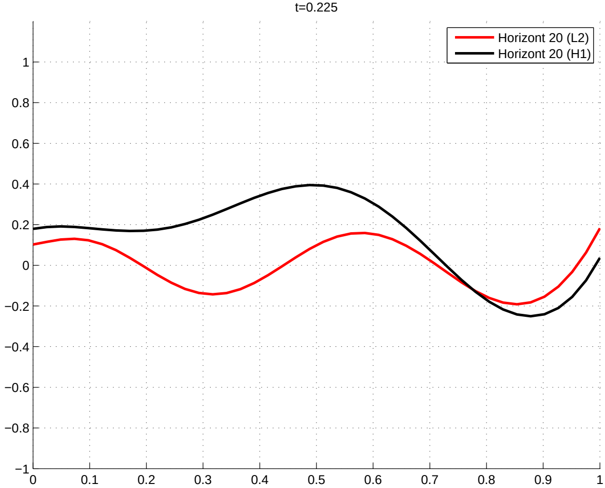
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

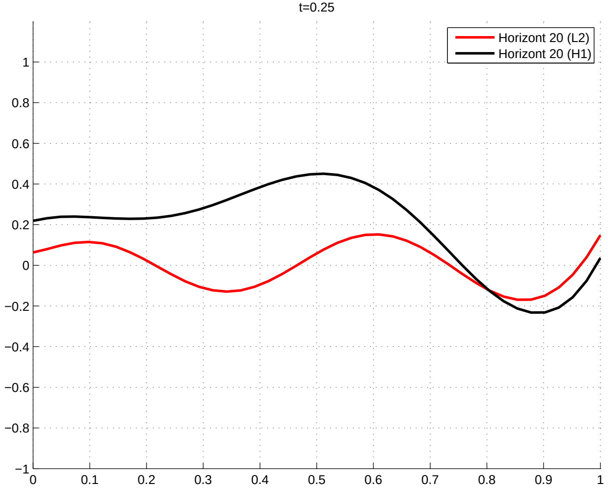
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

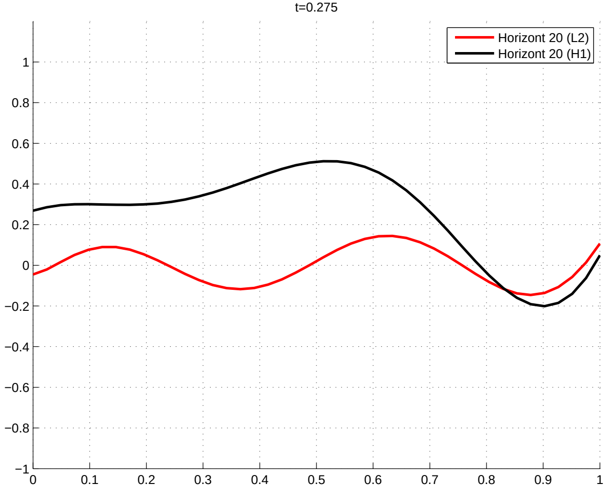
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

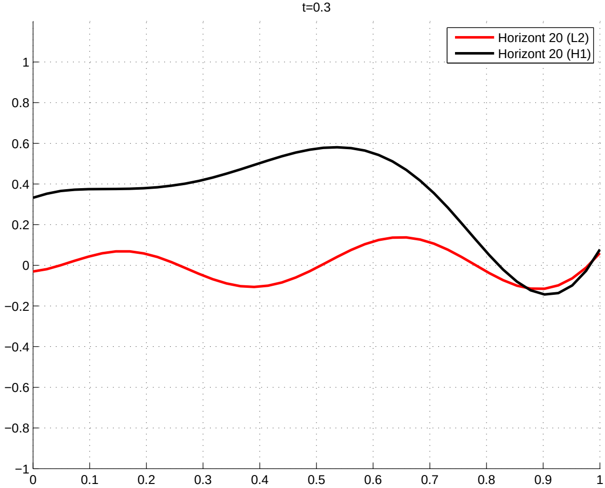
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

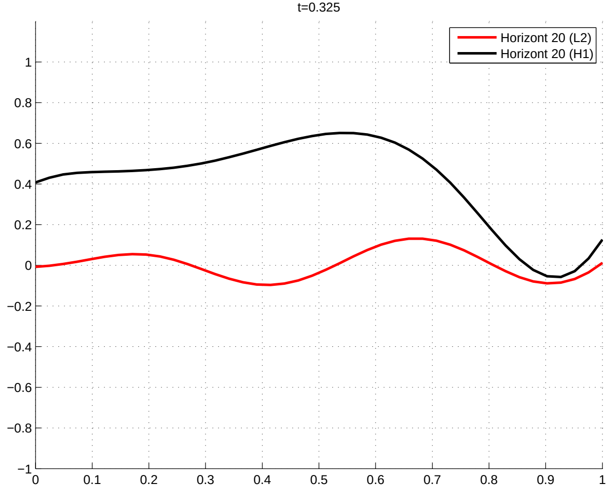
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

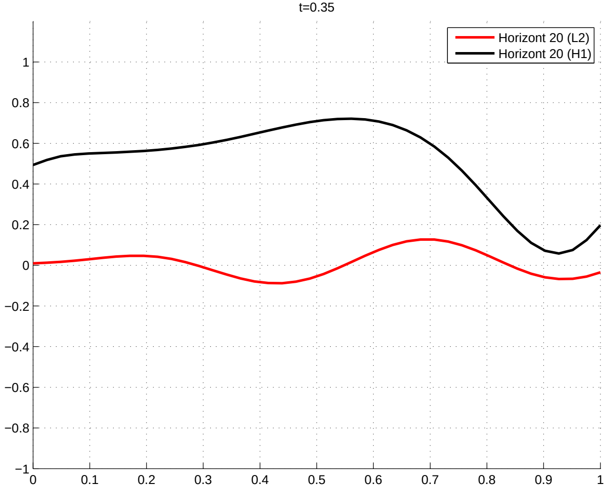
UNIVERSITÄT BAYREUTH

Boundary control, L_2 vs. H_1, $N = 20$



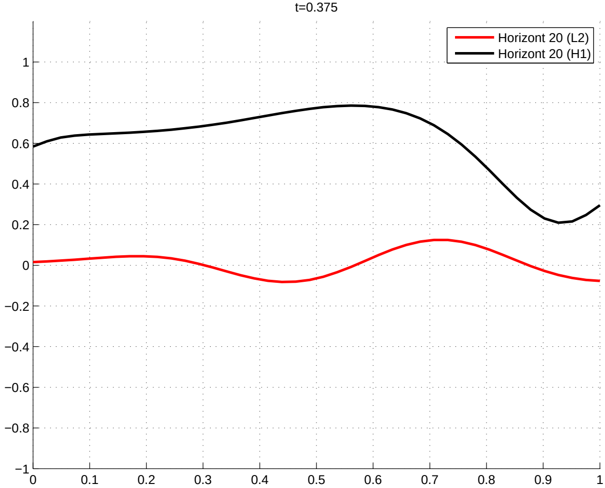
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2 vs. H_1, $N = 20$



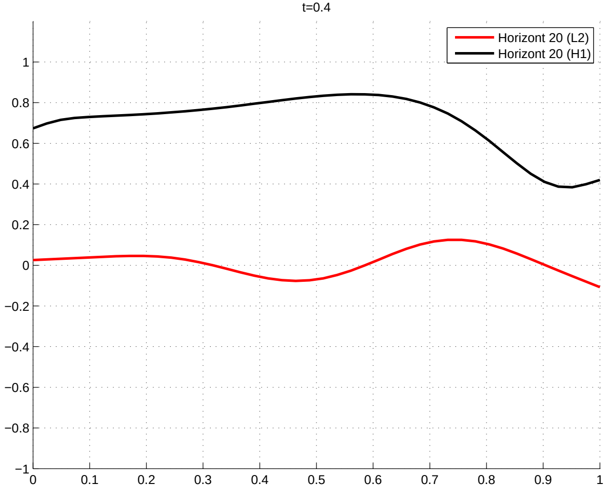
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

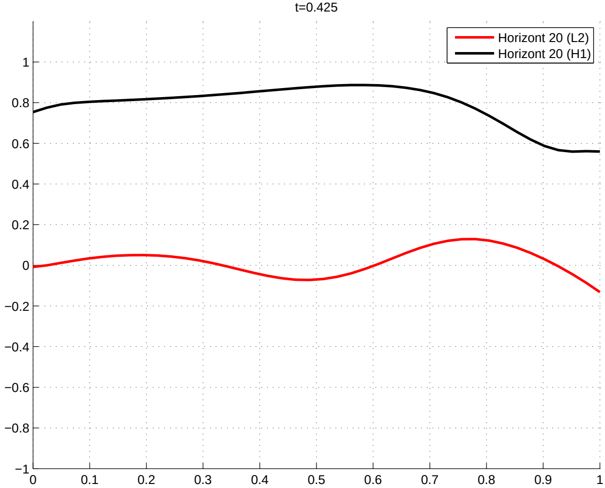
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

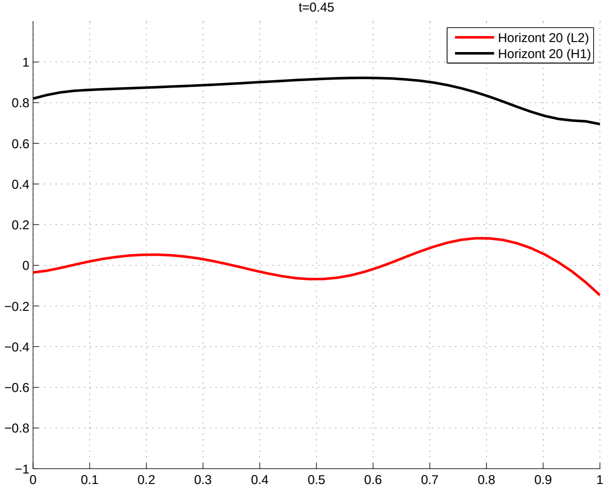
UNIVERSITÄT BAYREUTH

Boundary control, L_2 vs. H_1, $N = 20$



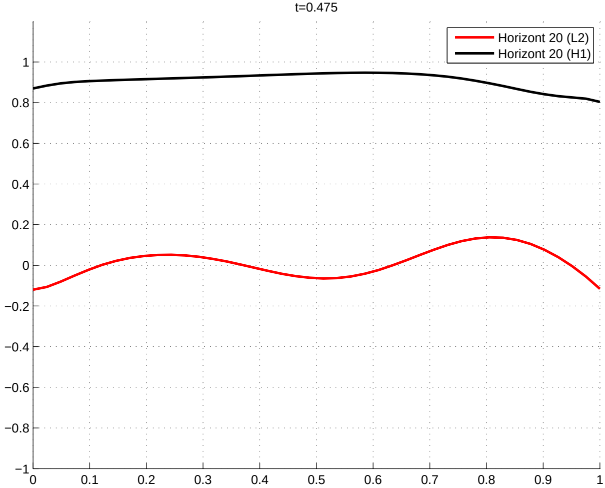
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

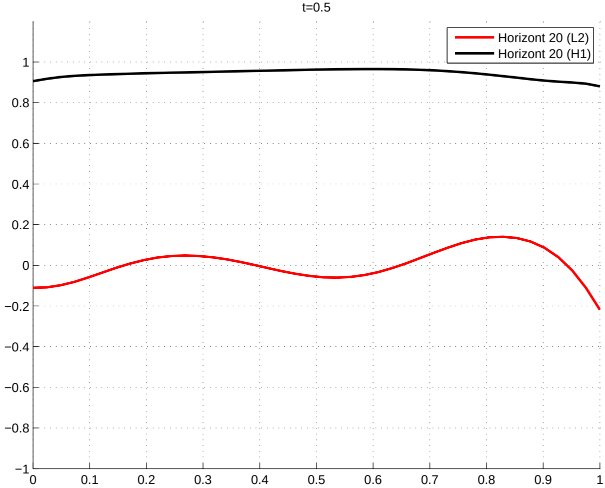
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

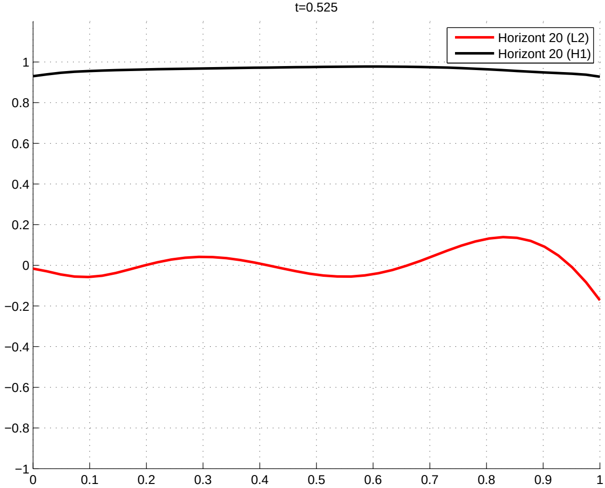
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

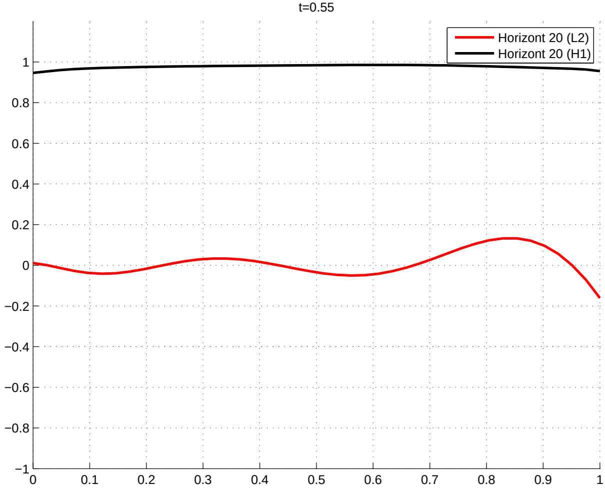
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

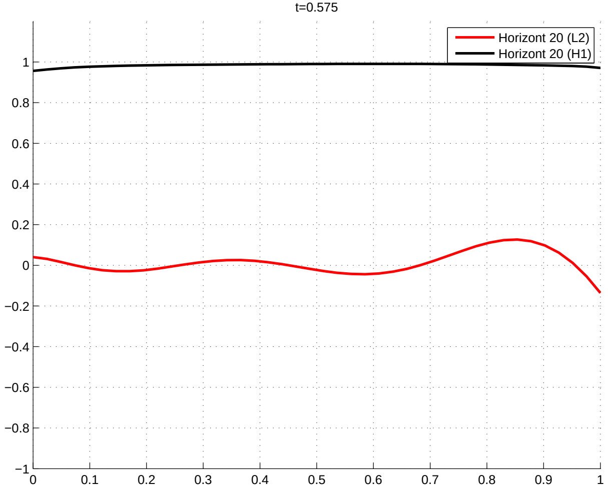
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

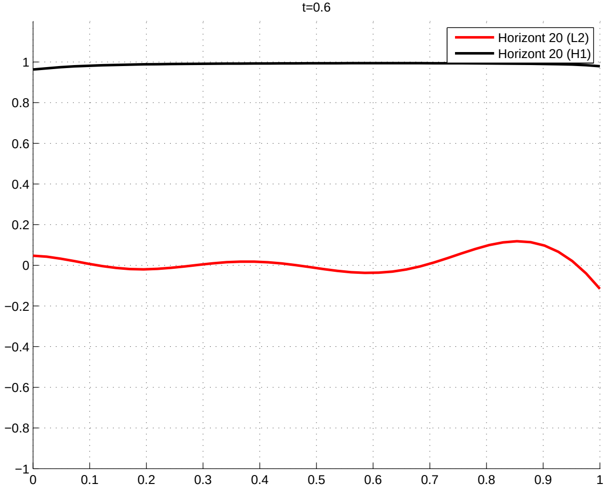
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

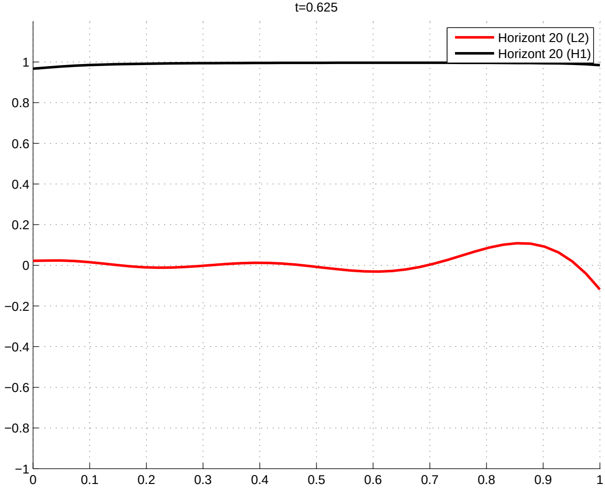
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

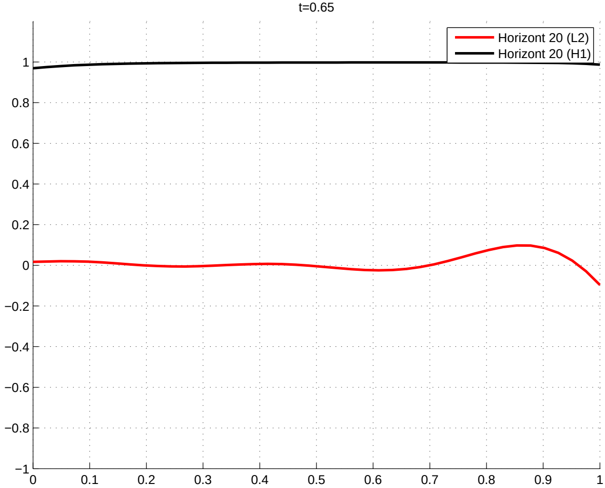
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

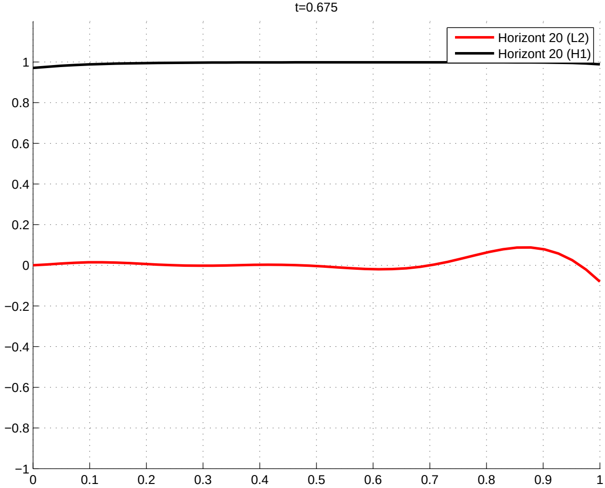
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

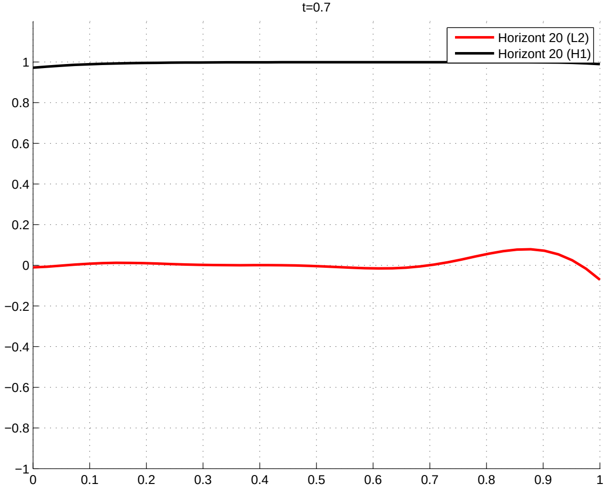
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

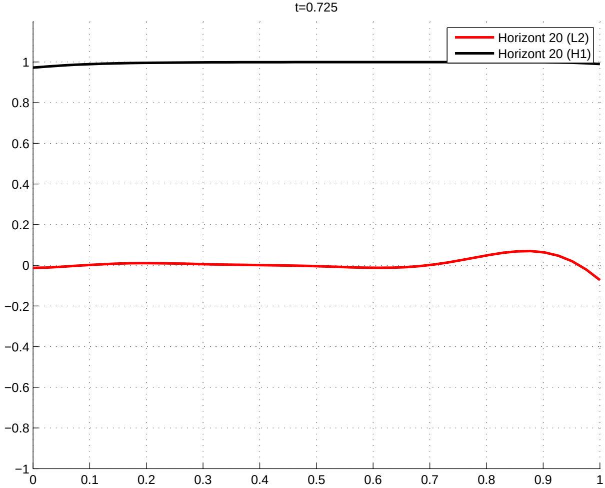
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

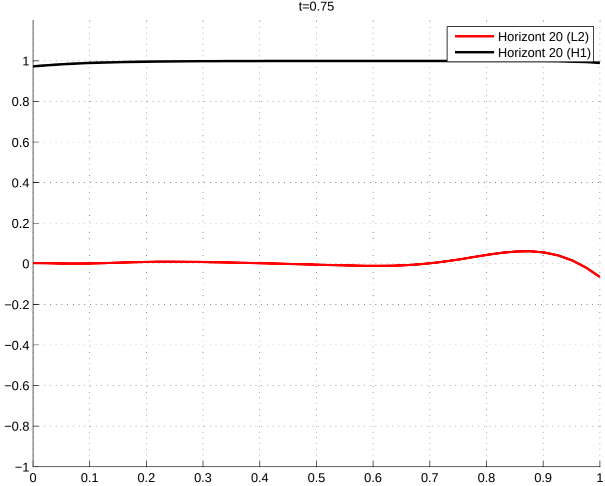
UNIVERSITÄT BAYREUTH

Boundary control, L_2 vs. H_1, $N = 20$



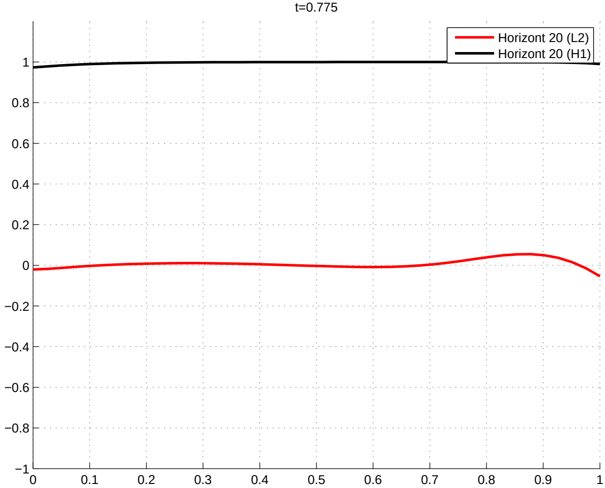
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

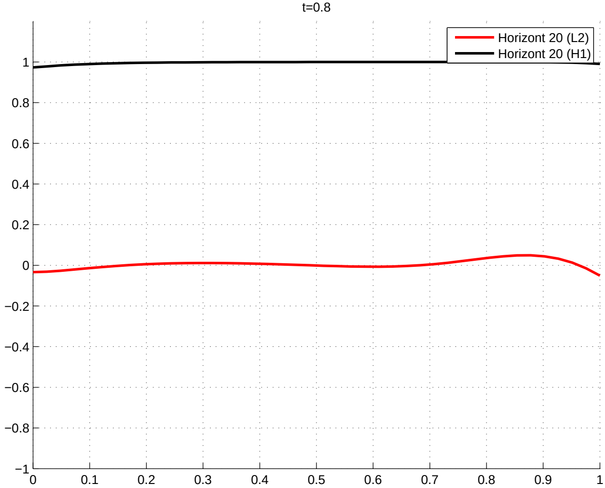
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

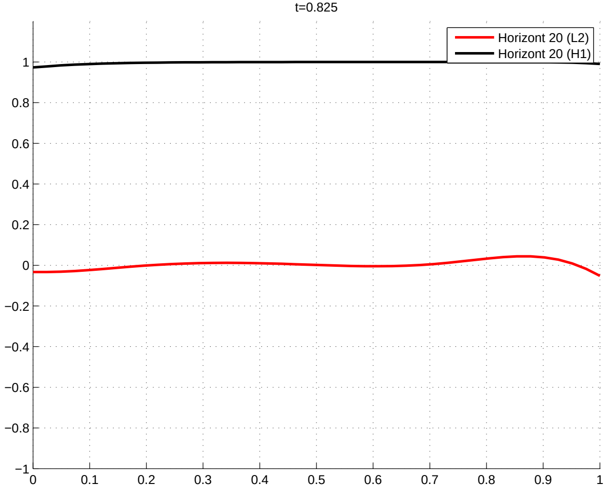
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

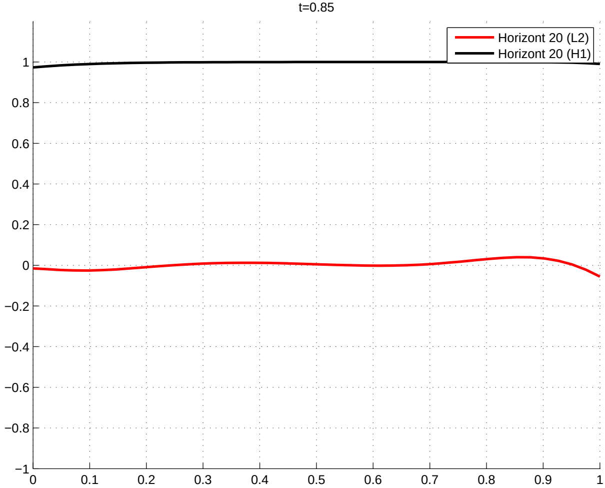
Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

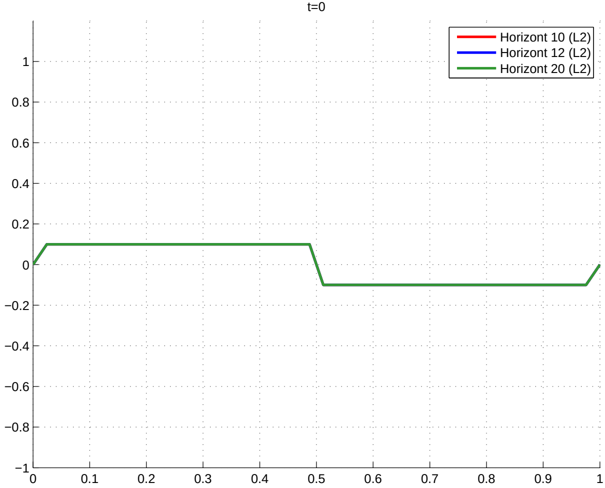
UNIVERSITÄT BAYREUTH

Boundary control, L_2 vs. H_1, $N = 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

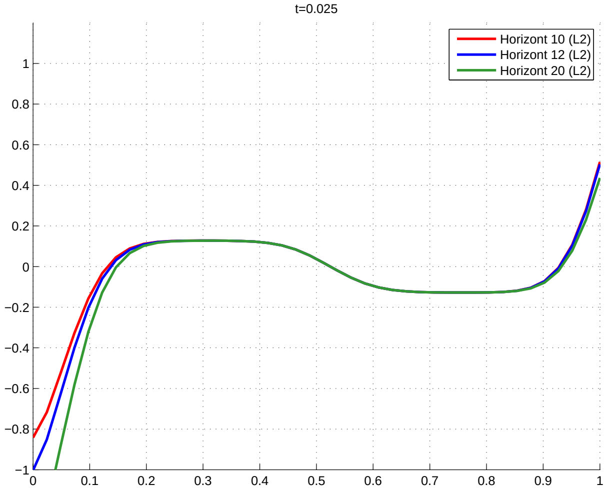
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

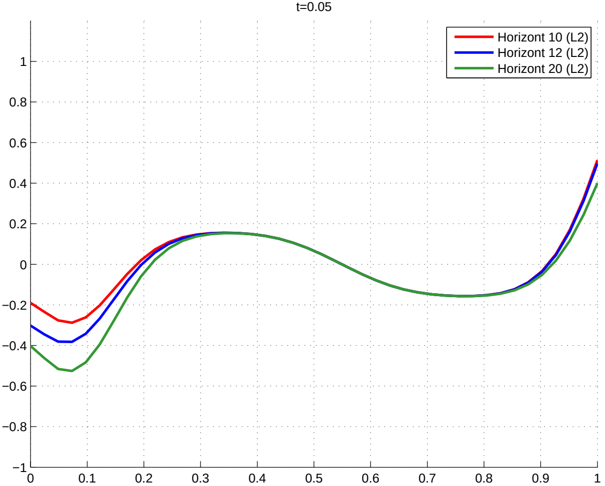
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

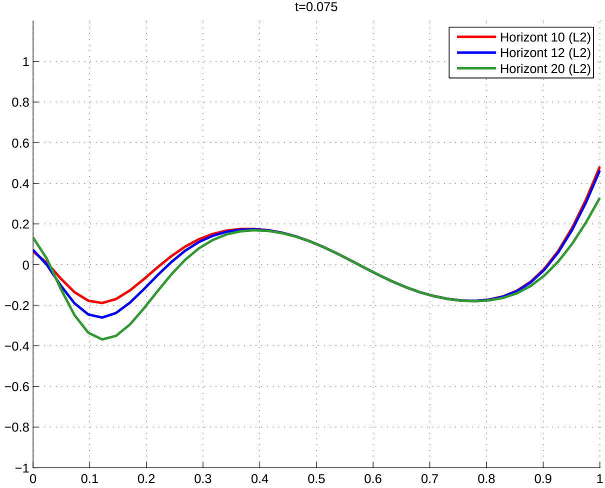
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

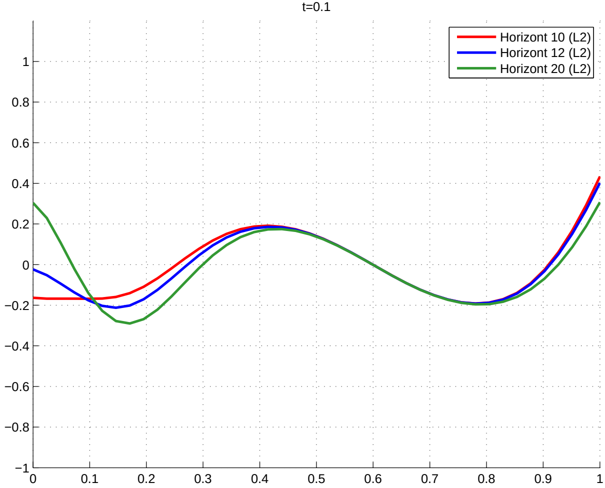
UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

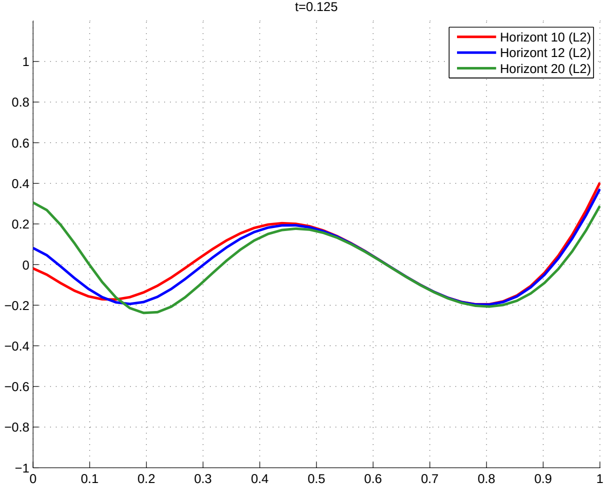
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

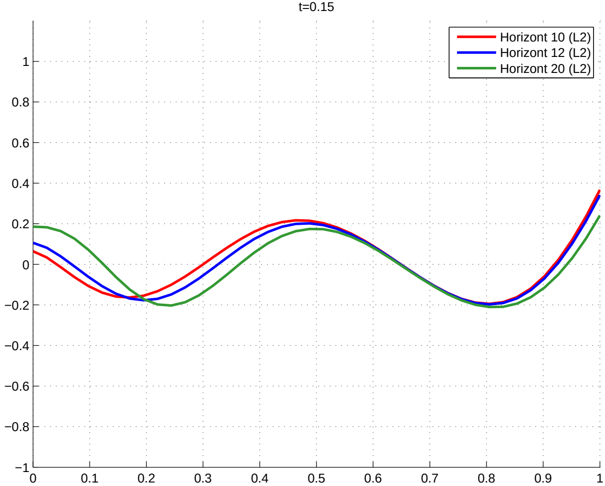
UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

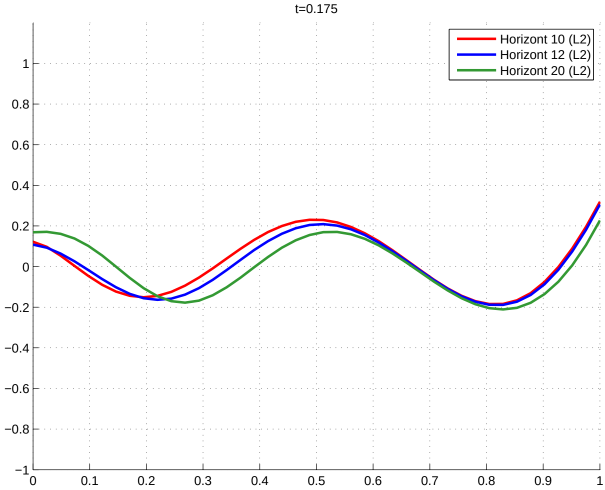
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

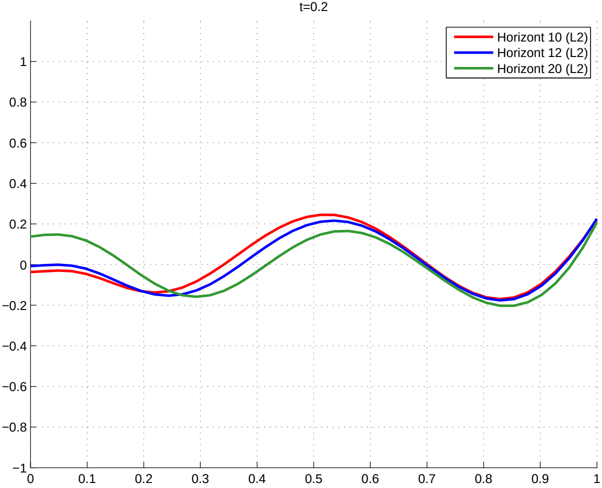
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

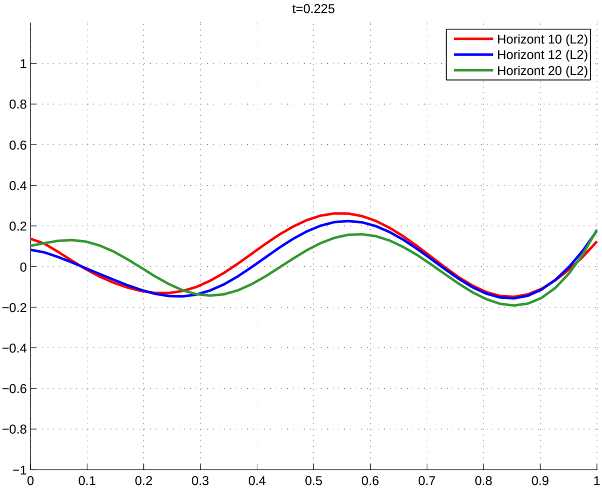
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

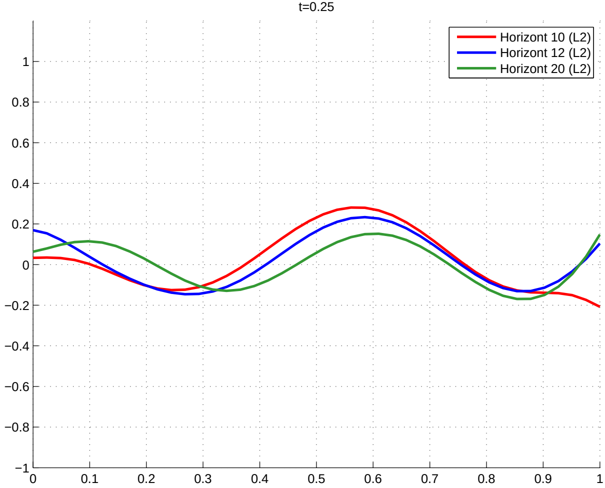
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

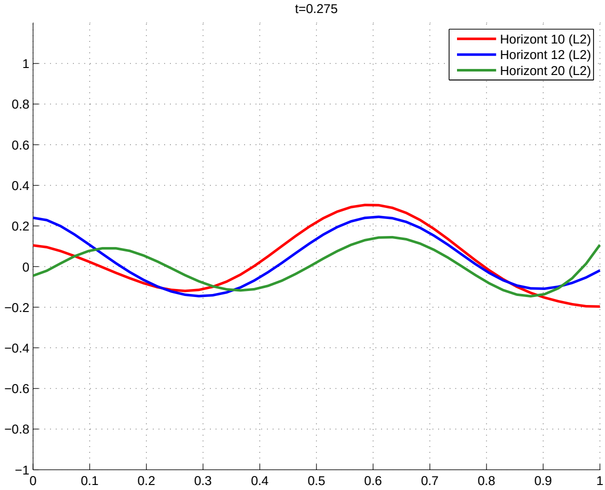
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

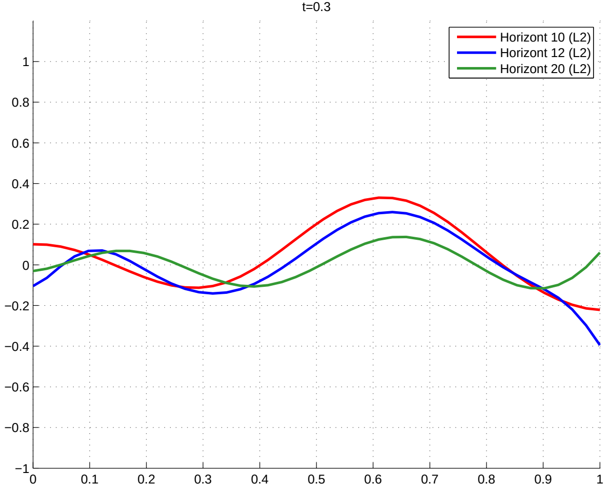
UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

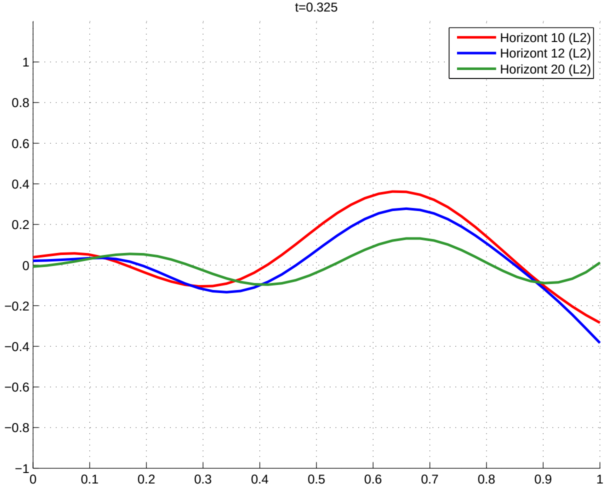
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

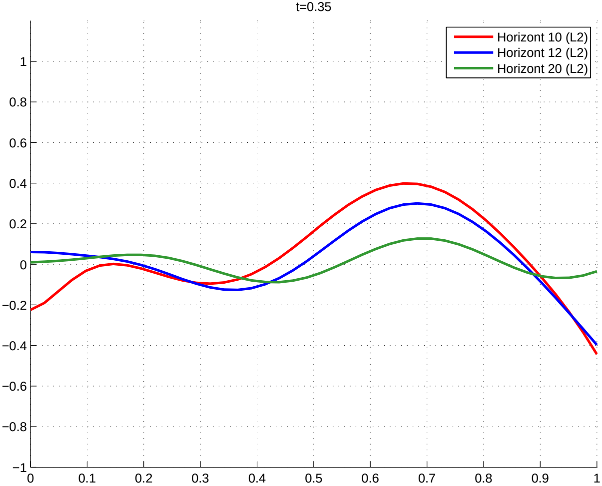
UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

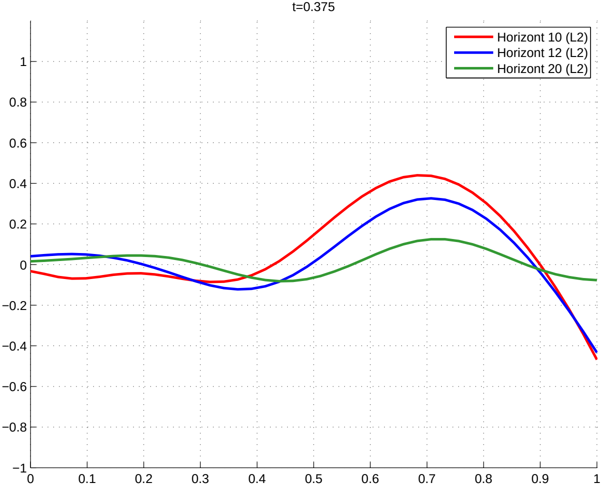
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

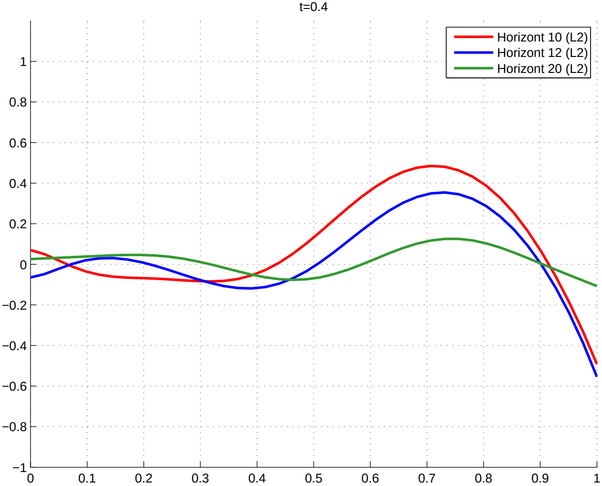
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

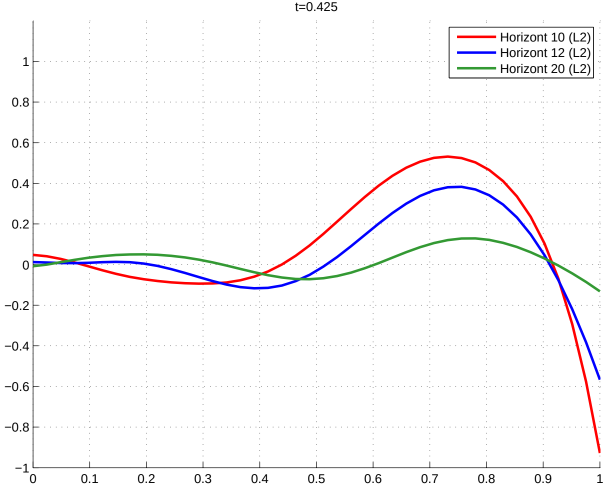
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

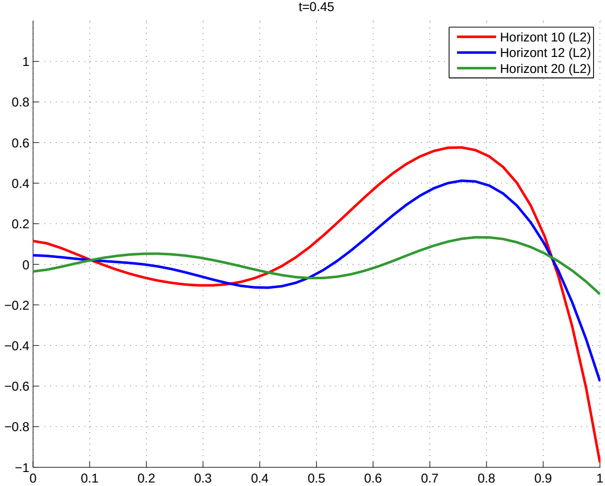
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

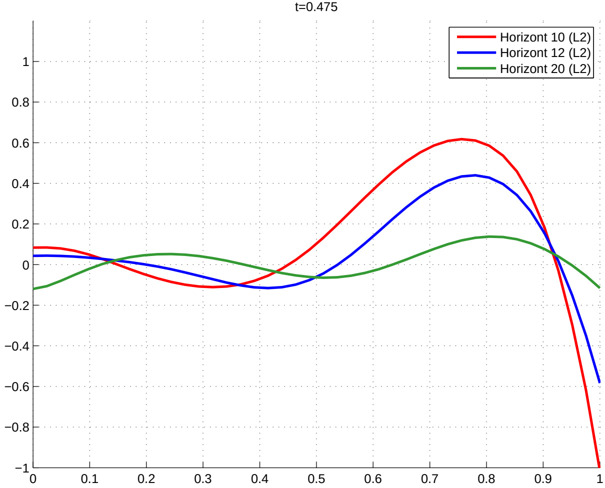
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

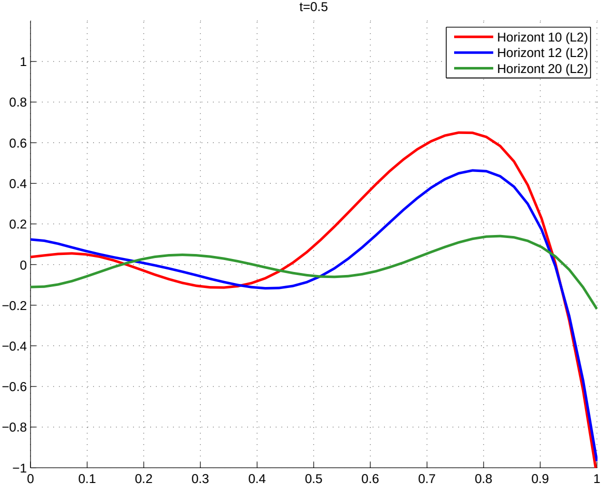
UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

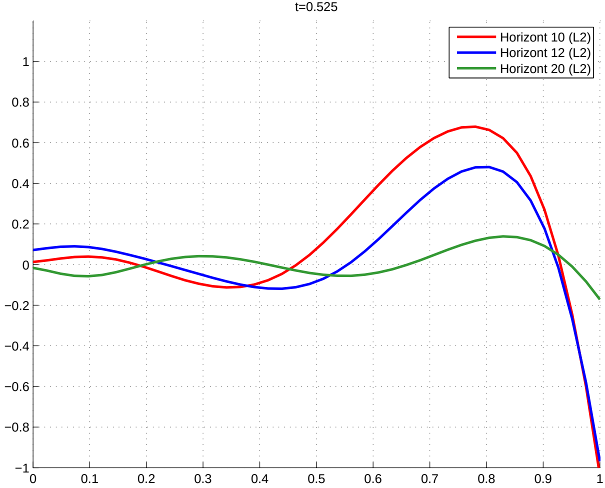
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

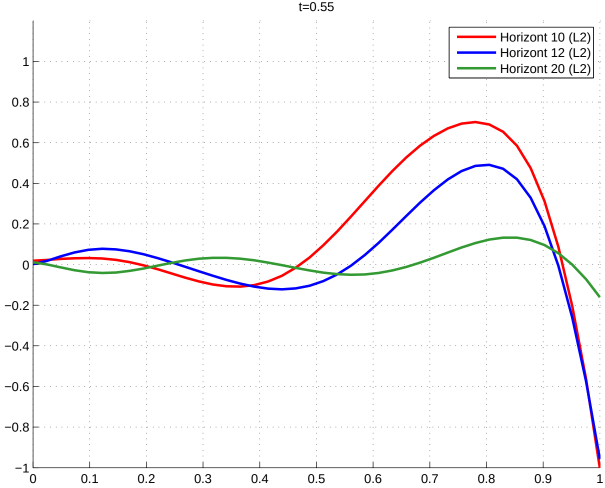
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

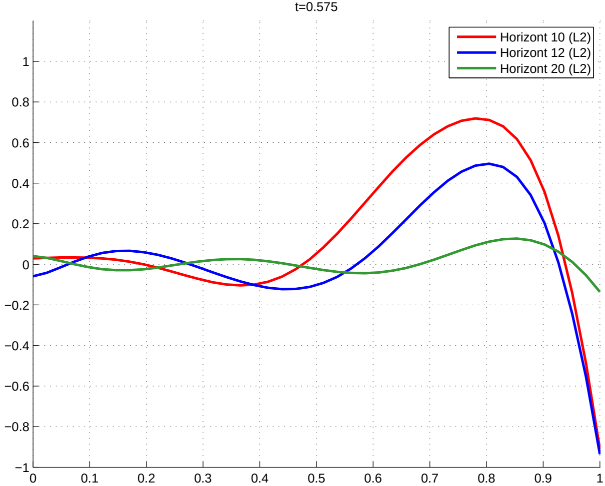
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

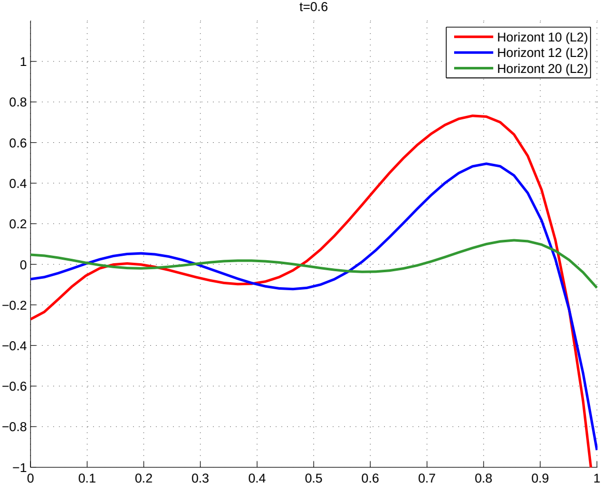
UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

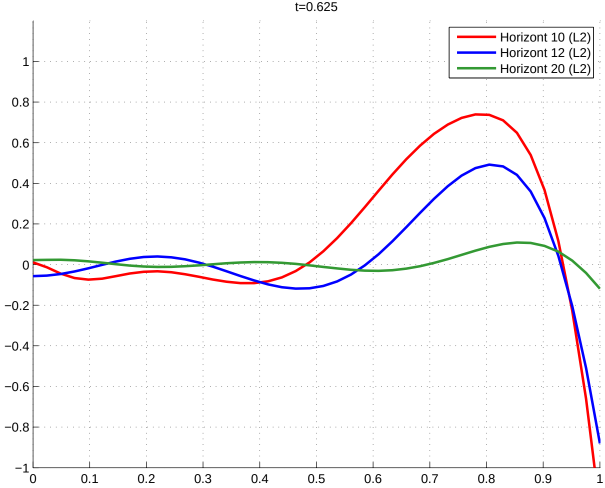
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

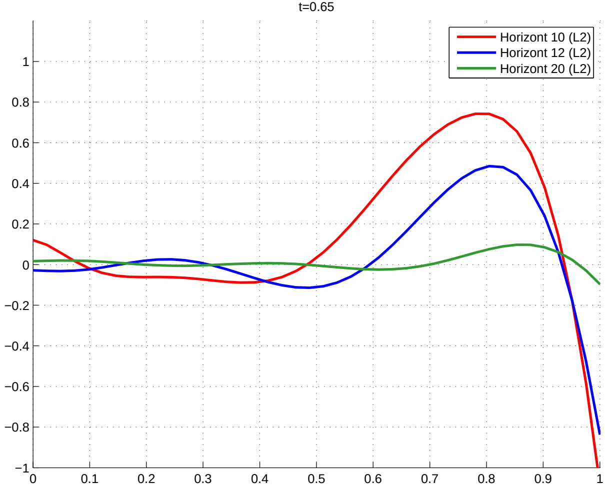
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

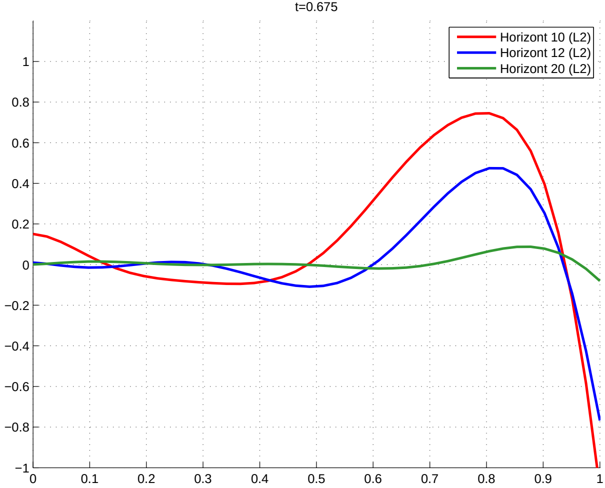
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

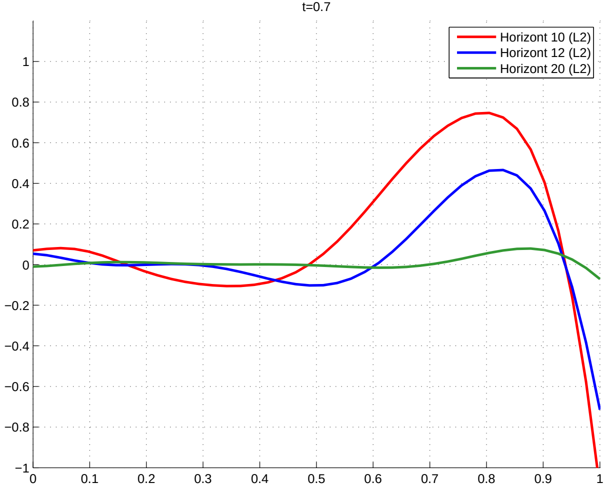
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

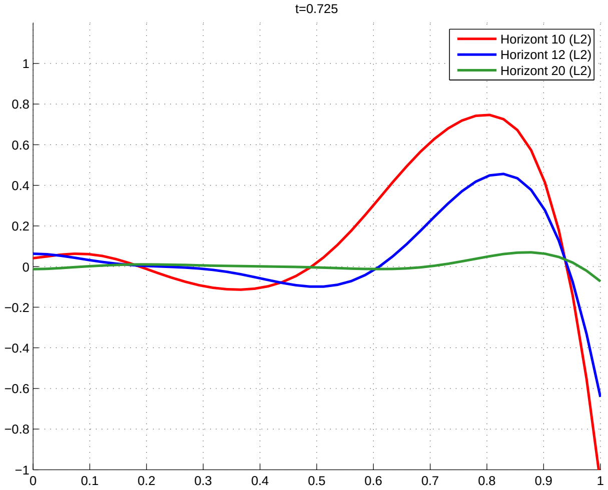
UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

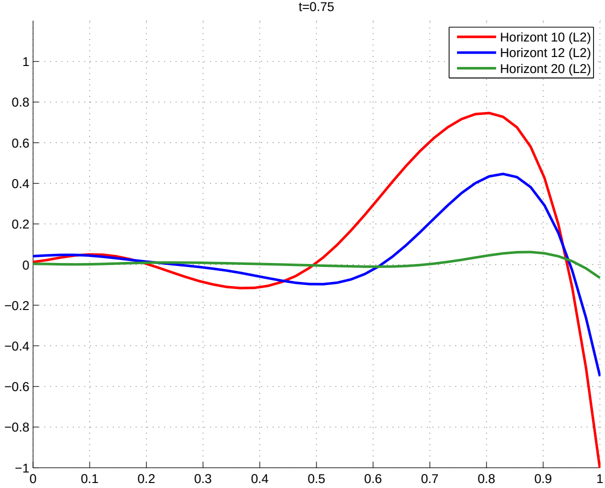
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

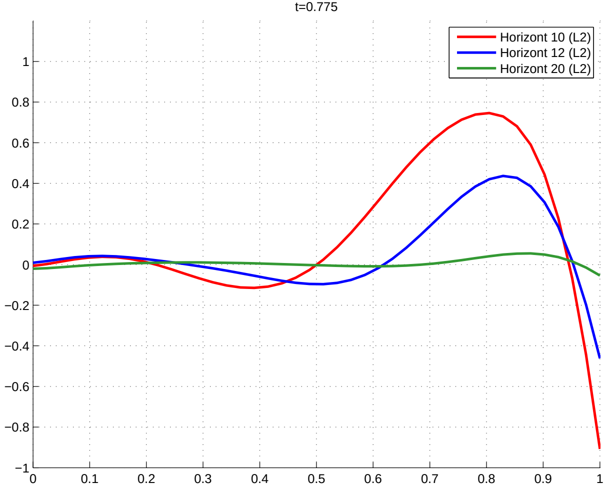
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

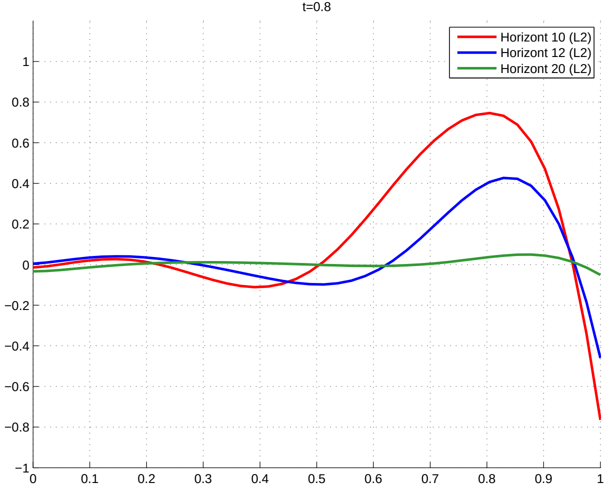
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

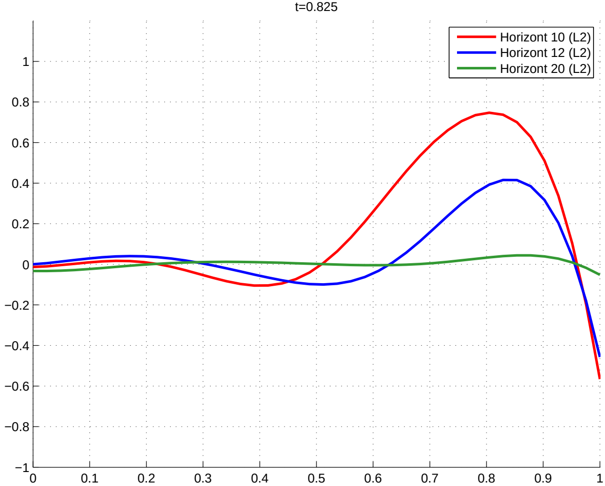
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

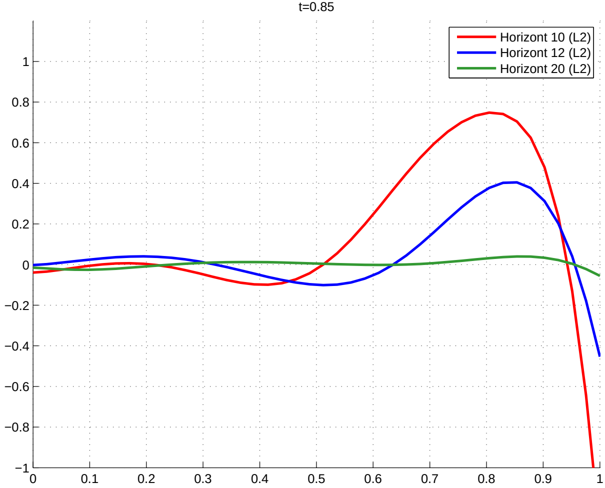
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

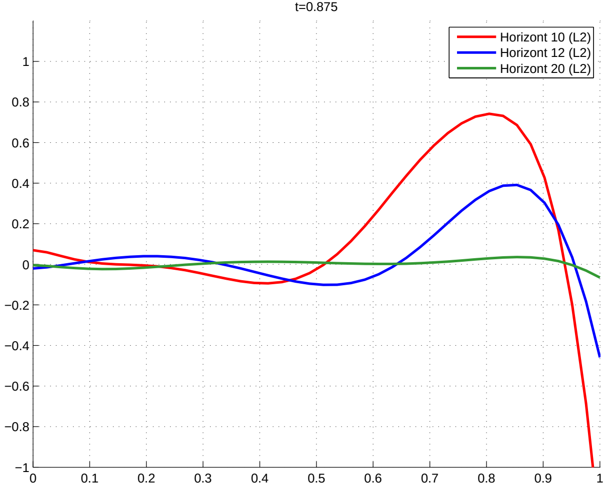
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

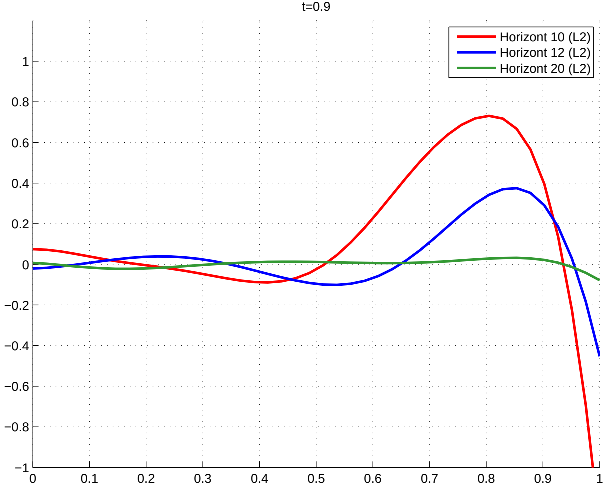
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

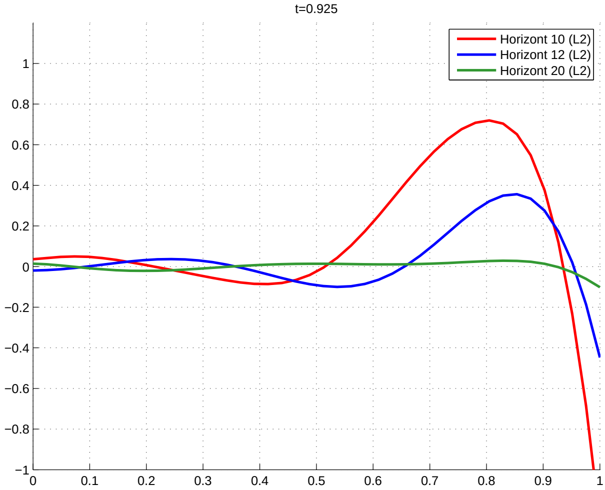
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

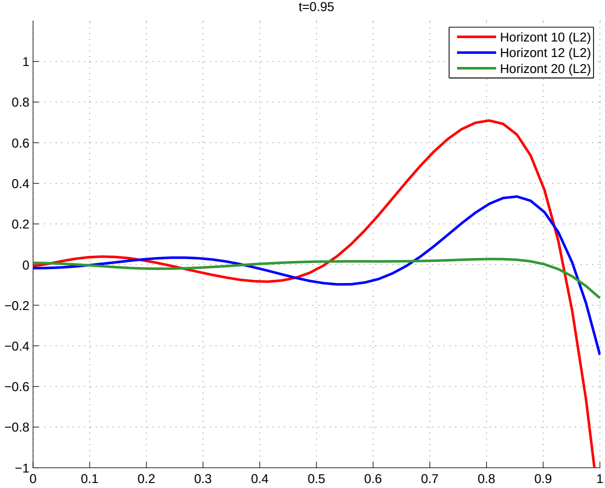
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

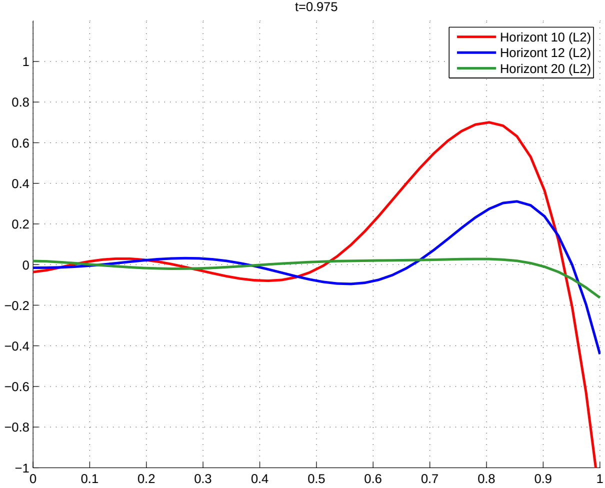
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

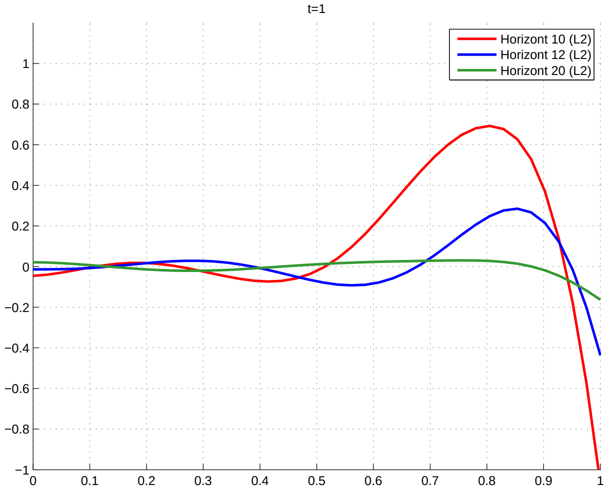
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

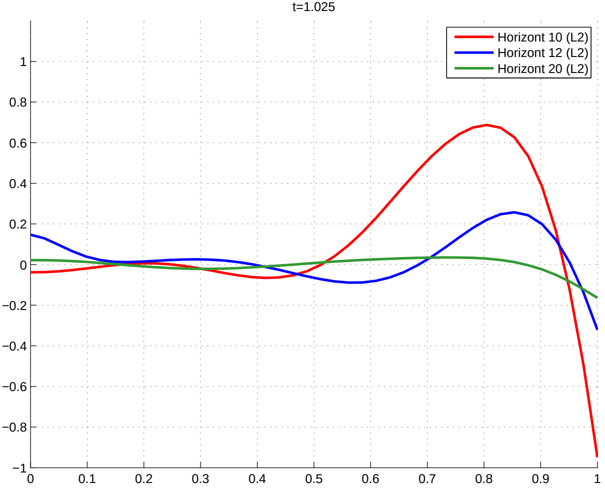
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

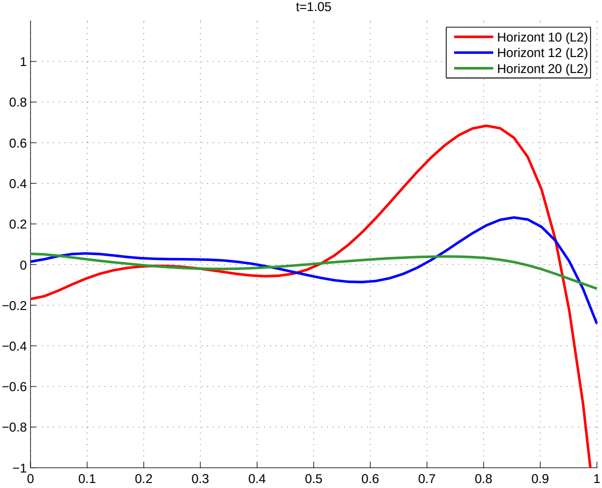
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

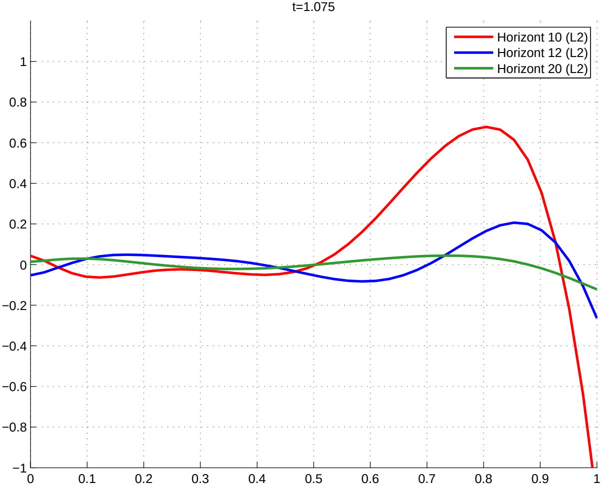
UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



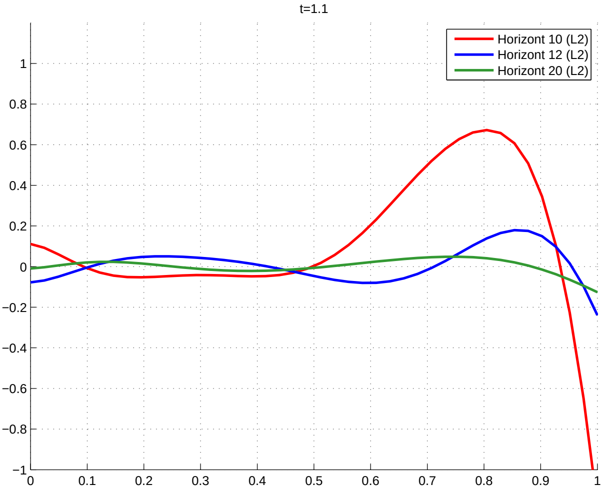
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2, $N = 10, 12, 20$



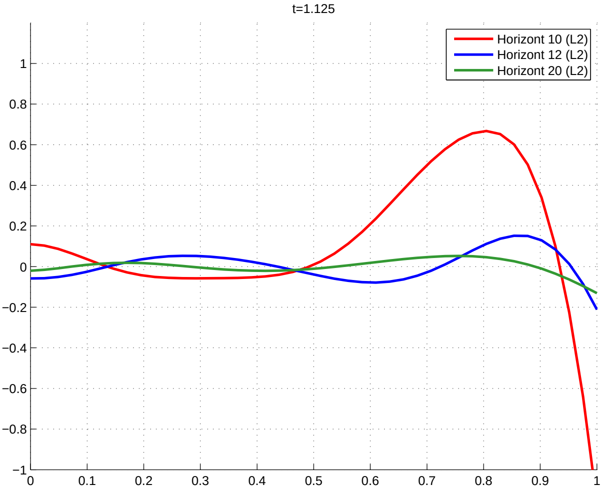
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

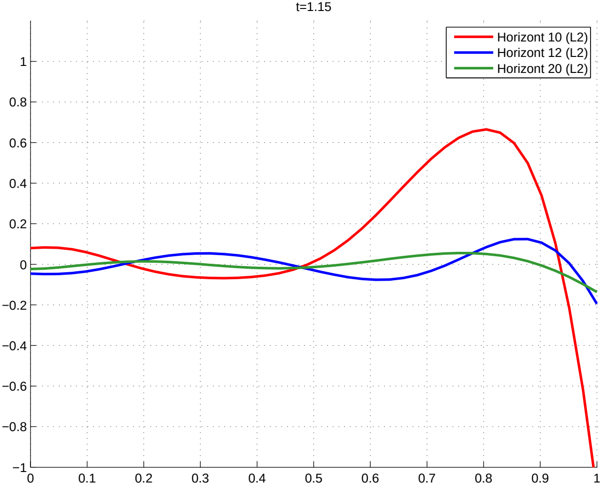
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

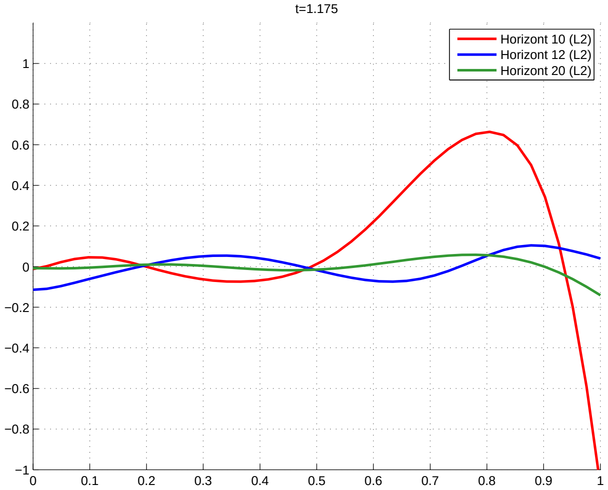
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

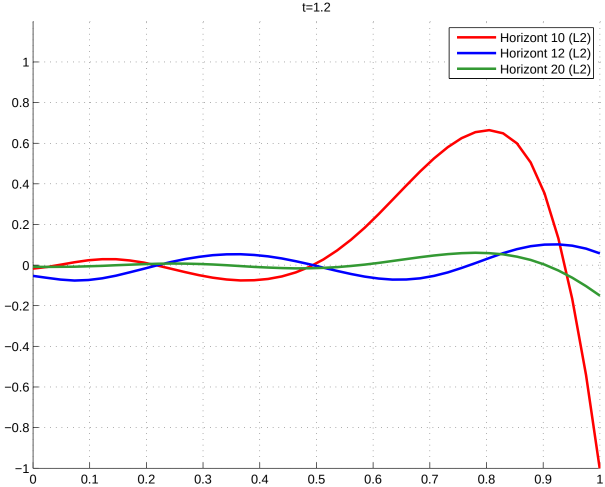
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

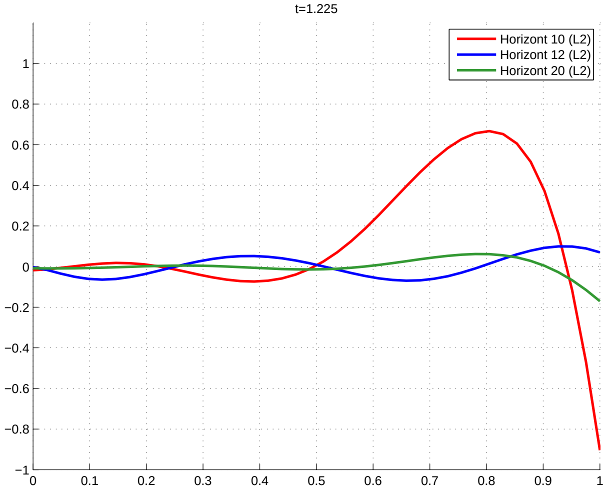
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

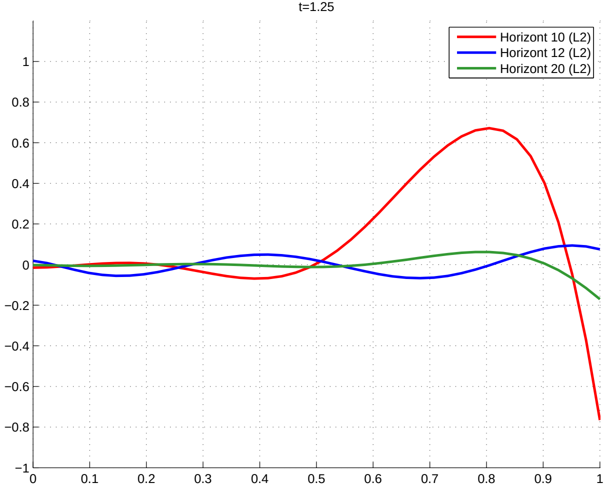
UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

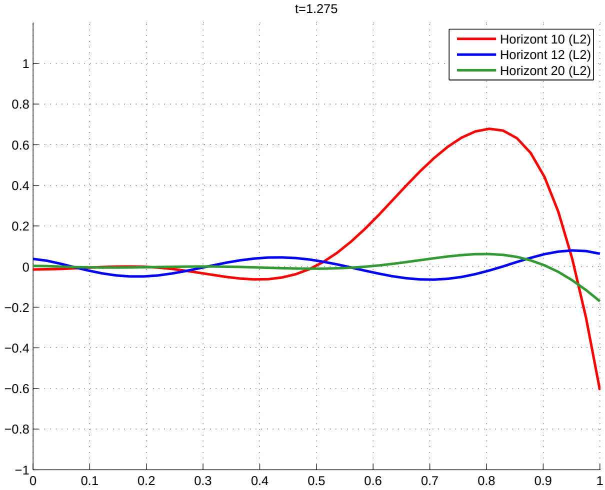
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

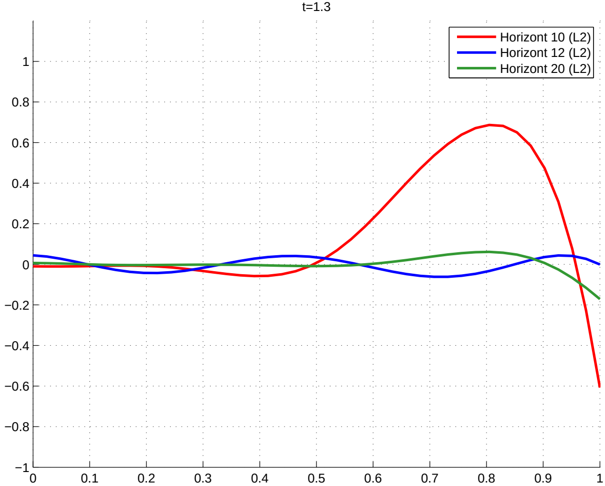
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

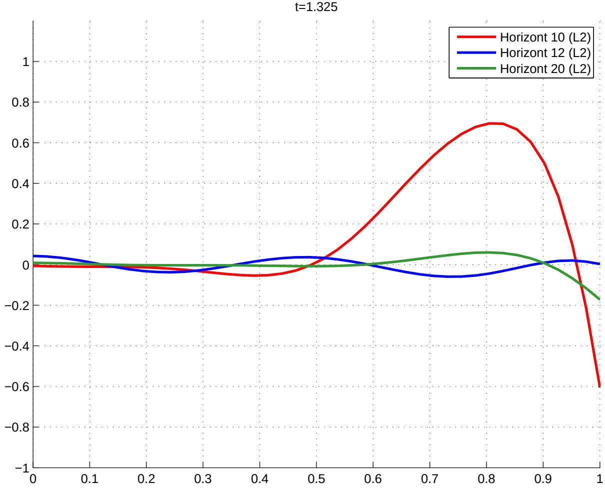
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

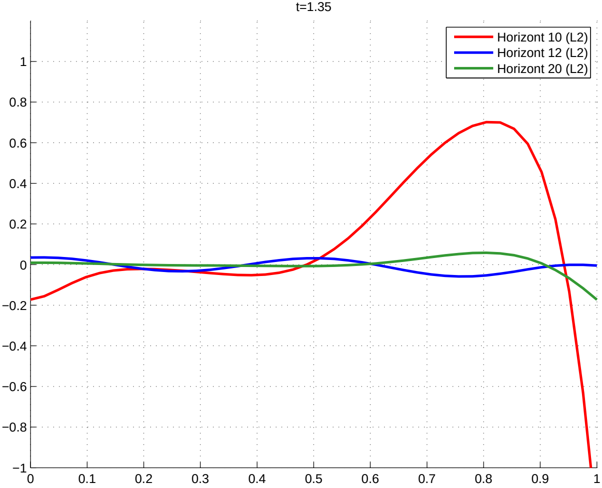
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

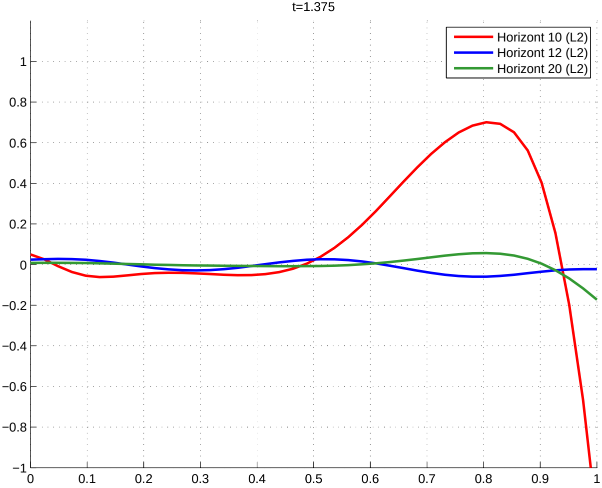
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

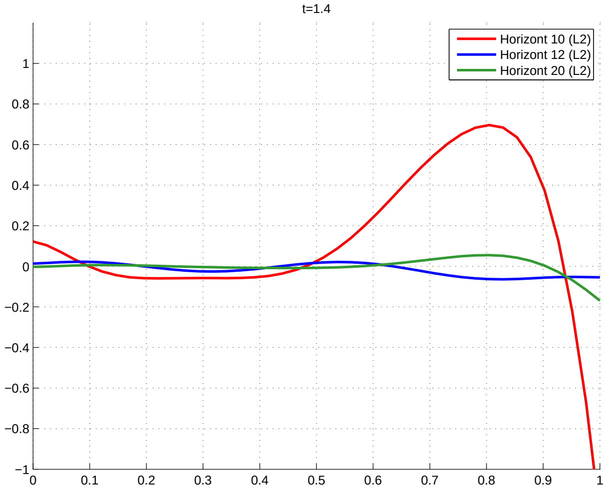
Boundary control, L_2, $N = 10, 12, 20$



Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

UNIVERSITÄT BAYREUTH

Boundary control, L_2, $N = 10, 12, 20$



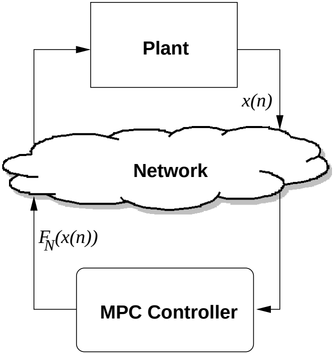
Boundary control, $\lambda = 0.001$, sampling time $T = 0.025$

(PDE computations: N. Altmüller, A. Grötsch, J. Pannek, S. Trenz, K. Worthmann)

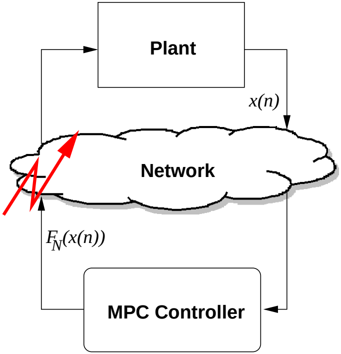
(7) Varying control horizon

[Grüne/Pannek/Worthmann '09]

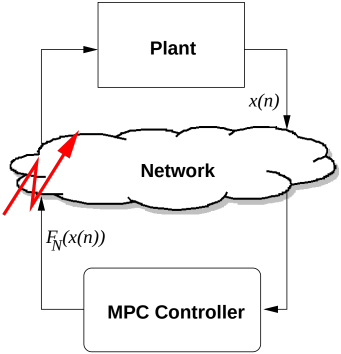
Packet loss



Packet loss

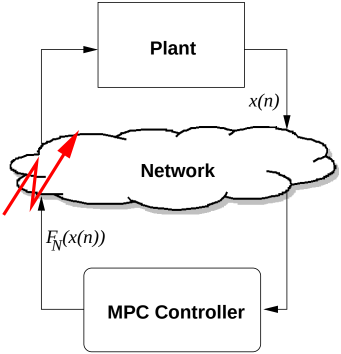


Packet loss



Idea: • send several values of optimal open loop control sequence (instead of just the first value)

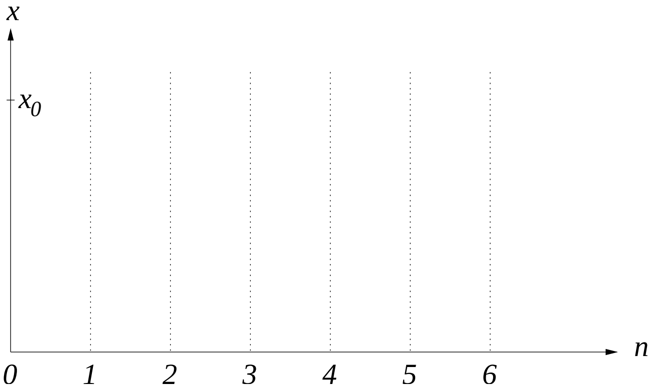
Packet loss



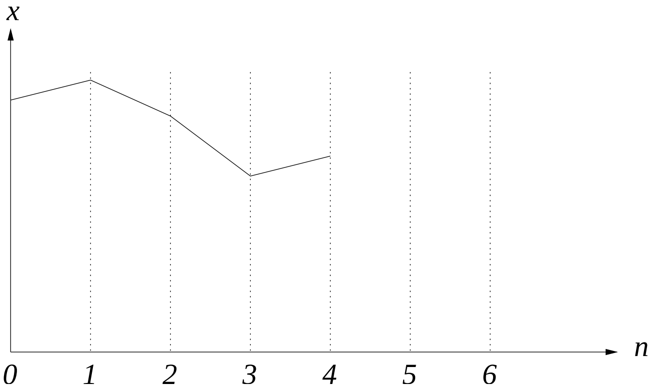
Idea: • send several values of optimal open loop control sequence (instead of just the first value)

• use these values until next values arrive

Schematic illustration of the idea

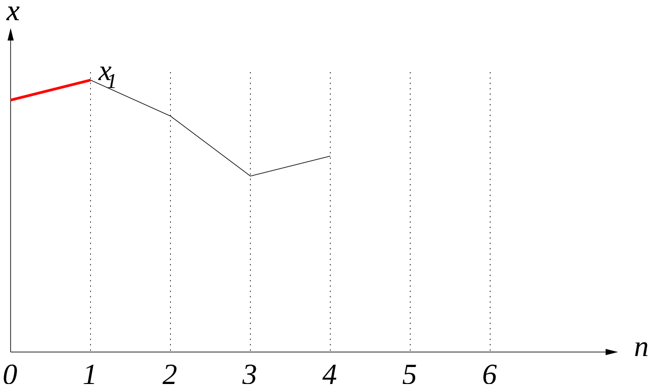


Schematic illustration of the idea



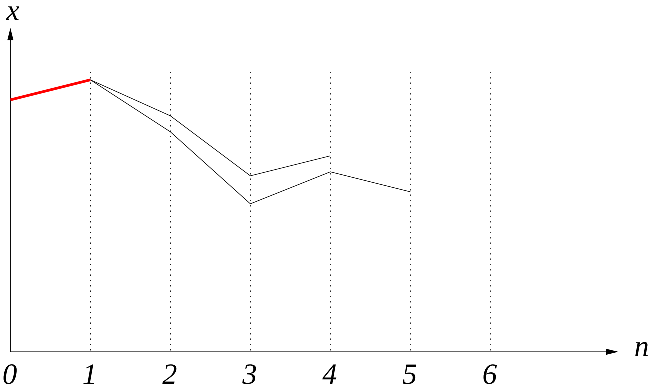
black = predictions (open loop optimization)

Schematic illustration of the idea



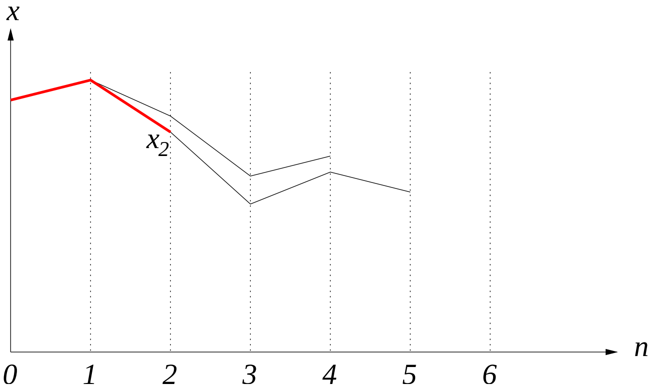
black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



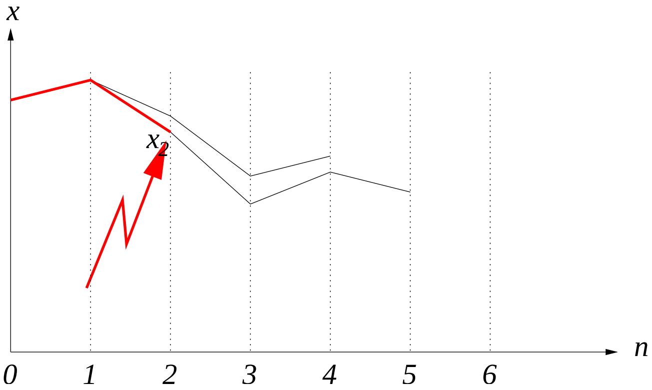
black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



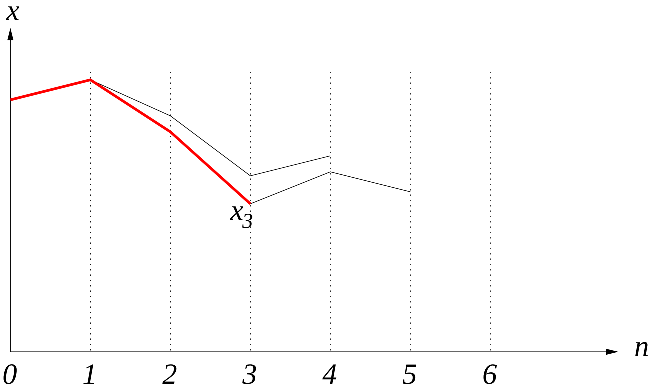
black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



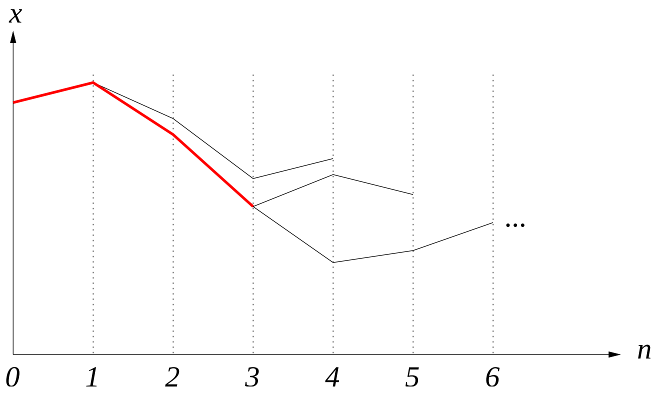
black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



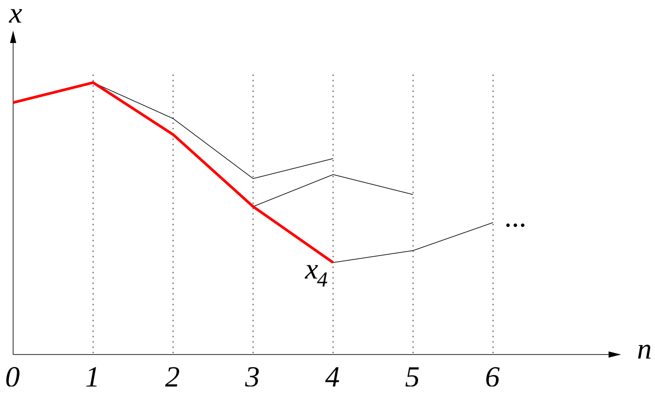
black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



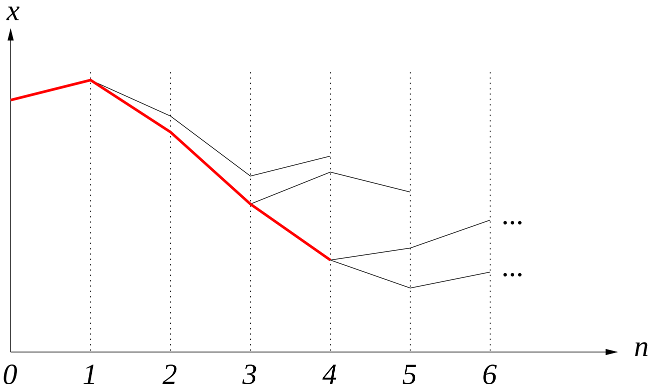
black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



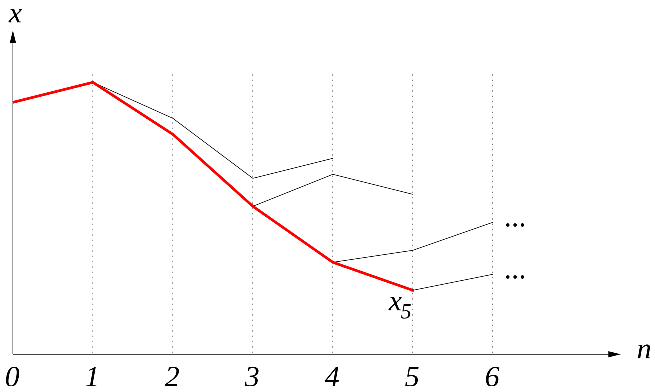
black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



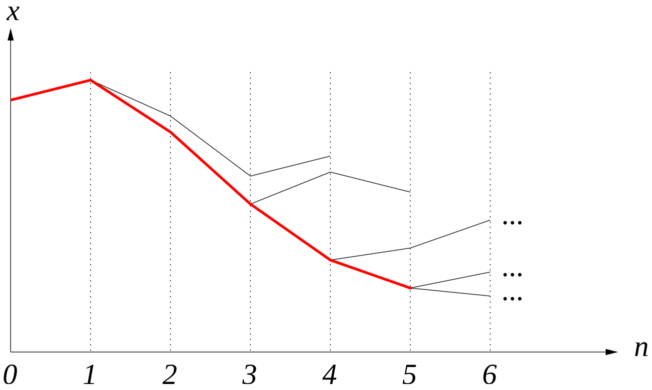
black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



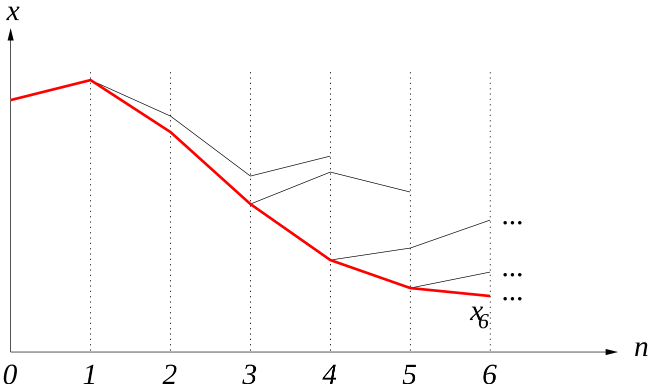
black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



black = predictions (open loop optimization)
red = MPC closed loop

Schematic illustration of the idea



black = predictions (open loop optimization)

red = MPC closed loop

Rigorous formulation

Denote successful transmission times by n_i, $i = 1, 2, \ldots$

Rigorous formulation

Denote successful transmission times by n_i, $i = 1, 2, \dots$

Define a buffer length $M \in \mathbb{N}$, $M \le N - 1$

Rigorous formulation

Denote successful transmission times by n_i, $i = 1, 2, \ldots$

Define a buffer length $M \in \mathbb{N}$, $M \leq N - 1$

At each transmission time n_i, the plant receives and buffers the feedback control sequence

$$F_N(x_{n_i}, k) = u^*(k), \quad k = 0, 1, 2, \ldots, M - 1$$

and implements

$$F_N(x_{n_i}, 0),\ F_N(x_{n_i}, 1),\ \ldots,\ F_N(x_{n_i}, m_i - 1)$$

on the control horizon $m_i = n_{i+1} - n_i \leq M$, i.e., until the next sequence arrives

UNIVERSITÄT BAYREUTH

Rigorous formulation

Denote successful transmission times by n_i, $i = 1, 2, \dots$

Define a buffer length $M \in \mathbb{N}$, $M \leq N - 1$

At each transmission time n_i, the plant receives and buffers the feedback control sequence

$$F_N(x_{n_i}, k) = u^*(k), \quad k = 0, 1, 2, \dots, M - 1$$

and implements

$$F_N(x_{n_i}, 0),\ F_N(x_{n_i}, 1),\ \dots,\ F_N(x_{n_i}, m_i - 1)$$

on the control horizon $m_i = n_{i+1} - n_i \leq M$, i.e., until the next sequence arrives

Note: m_i is unknown at time n_i

Stability theorem

Theorem: If there exists $\alpha \in (0,1]$ such that the relaxed Lyapunov inequality

$$V_N(x(m,x_0,u^*)) \le V_N(x) - \alpha \sum_{k=0}^{m-1} \ell(x(m,x_0,u^*),u^*(m))$$

holds for all $m = 1,\dots,M$, then asymptotic stability follows for the MPC closed loop with arbitrary transmission times n_i, $i \in \mathbb{N}$, satisfying $m_i = n_{i+1} - n_i \ge M$.

Stability theorem

Theorem: If there exists $\alpha \in (0, 1]$ such that the relaxed Lyapunov inequality

$$V_N(x(m, x_0, u^*)) \le V_N(x) - \alpha \sum_{k=0}^{m-1} \ell(x(m, x_0, u^*), u^*(m))$$

holds for all $m = 1, \ldots, M$, then asymptotic stability follows for the MPC closed loop with arbitrary transmission times n_i, $i \in \mathbb{N}$, satisfying $m_i = n_{i+1} - n_i \ge M$.

Furthermore, V_N is Lyapunov function at the transmission times n_i and we get the suboptimality estimate

$$J_\infty(x, F_N) \le V_\infty(x)/\alpha$$

Stability theorem

Theorem: If there exists $\alpha \in (0,1]$ such that the relaxed Lyapunov inequality

$$V_N(x(m,x_0,u^*)) \le V_N(x) - \alpha \sum_{k=0}^{m-1} \ell(x(m,x_0,u^*),u^*(m))$$

holds for all $m = 1,\ldots,M$, then asymptotic stability follows for the MPC closed loop with arbitrary transmission times n_i, $i \in \mathbb{N}$, satisfying $m_i = n_{i+1} - n_i \ge M$.

Furthermore, V_N is Lyapunov function at the transmission times n_i and we get the suboptimality estimate

$$J_\infty(x,F_N) \le V_\infty(x)/\alpha$$

Note: The stability for arbitrary but fixed m carries over to time varying m_i because V_N is a common Lyapunov function

Computation of $\alpha(N,m)$

We want $\alpha = \alpha(N,m)$ satisfying

$$V_N(x(m,x_0,u^*)) \le V_N(x) - \alpha \sum_{k=0}^{m-1} \ell(x(m,x_0,u^*),u^*(m)),$$

for all $m = 1,\dots,M$.

Computation of $\alpha(N,m)$

We want $\alpha = \alpha(N,m)$ satisfying

$$V_N(x(m,x_0,u^*)) \le V_N(x) - \alpha \sum_{k=0}^{m-1} \ell(x(m,x_0,u^*),u^*(m)),$$

for all $m = 1,\ldots,M$.

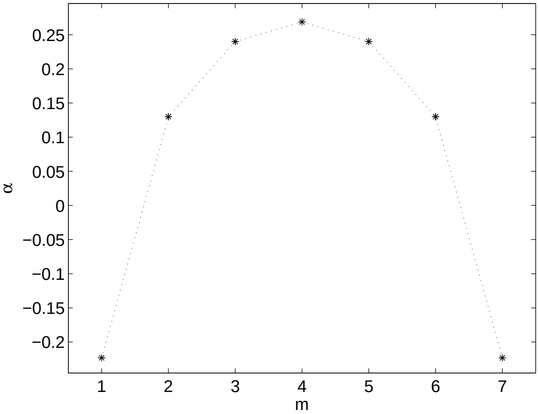
Again, for each m this can be computed via an explicitly solvable linear program which yields

$$\alpha = 1 - \frac{\prod_{i=m+1}^{N}(\gamma_i - 1) \prod_{i=N-m+1}^{N}(\gamma_i - 1)}{\left(\prod_{i=m+1}^{N}\gamma_i - \prod_{i=m+1}^{N}(\gamma_i - 1)\right)\left(\prod_{i=N-m+1}^{N}\gamma_i - \prod_{i=N-m+1}^{N}(\gamma_i - 1)\right)}$$

with $\gamma_i = \sum_{k=0}^{i-1} C\sigma^k$

UNIVERSITÄT BAYREUTH

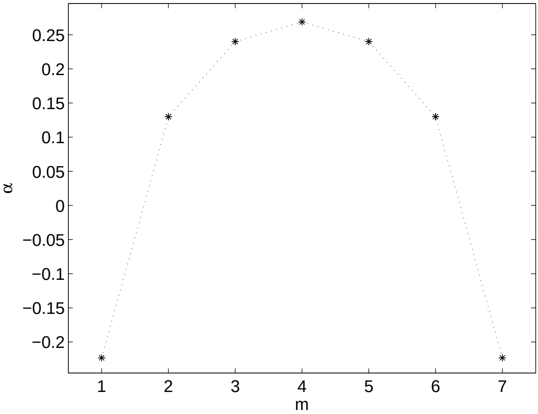
Example



$\alpha(N, m)$ for $C = 2$, $\sigma = 0.68$, $N = 8$, $m = 1, \ldots, 7$

UNIVERSITÄT BAYREUTH

Example



$\alpha(N,m)$ for $C=2$, $\sigma=0.68$, $N=8$, $m=1,\ldots,7$

This symmetry and monotonicity is not a coincidence

Property of $\alpha(N, m)$

Theorem: The values $\alpha(N, m)$ satisfy

$$\alpha(N, m) = \alpha(N, N - m), \;\; m = 1, \ldots, N - 1$$

and

$$\alpha(N, m) \leq \alpha(N, m + 1), \;\; m = 1, \ldots \lceil N/2 \rceil$$

UNIVERSITÄT BAYREUTH

Property of $\alpha(N, m)$

Theorem: The values $\alpha(N, m)$ satisfy

$$\alpha(N, m) = \alpha(N, N - m), \;\; m = 1, \ldots, N - 1$$

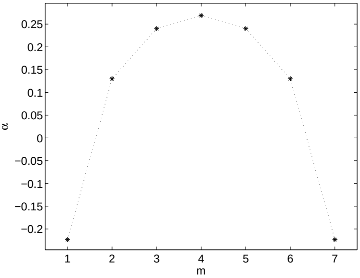
and

$$\alpha(N, m) \leq \alpha(N, m + 1), \;\; m = 1, \ldots \lceil N/2 \rceil$$

Corollary: If N is such that all C, σ-exponentially controllable systems are stabilized with "classical" MPC ($m = 1$), then they are stabilized for arbitrary varying control horizons $m_i \in \{1, \ldots, N - 1\}$

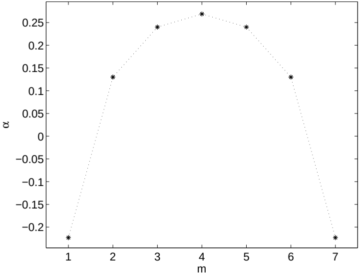
UNIVERSITÄT BAYREUTH

Conservatism of worst case analysis



The symmetry states that the worst case system for m behaves exactly as good as the worst case system for $N-m$.

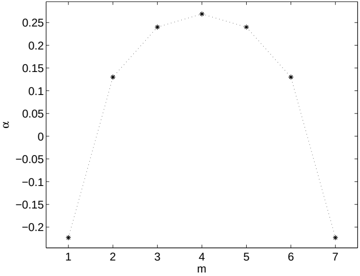
Conservatism of worst case analysis



The symmetry states that the worst case system for m behaves exactly as good as the worst case system for $N - m$.

However, in general these worst case systems do not coincide.

Conservatism of worst case analysis



The symmetry states that the worst case system for m behaves exactly as good as the worst case system for $N-m$.

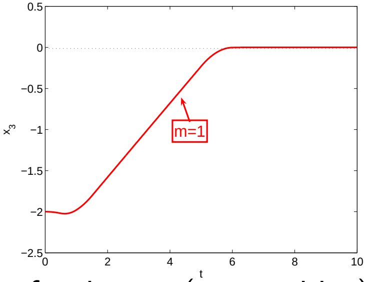
However, in general these worst case systems do not coincide.

How conservative is this worst case approach?

Example: linearized inverted pendulum

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} u, \qquad x_0 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 0 \end{pmatrix}$$

sampling time $T = 0.5$, $\ell(x,u) = 2\|x\|_1 + 4\|u\|_1$, $N = 11$

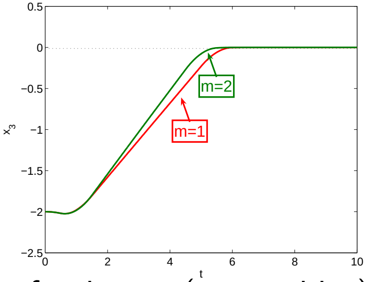


x_3 component of trajectory (cart position) for different m

Example: linearized inverted pendulum

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} u, \qquad x_0 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 0 \end{pmatrix}$$

sampling time $T = 0.5$, $\ell(x, u) = 2\|x\|_1 + 4\|u\|_1$, $N = 11$



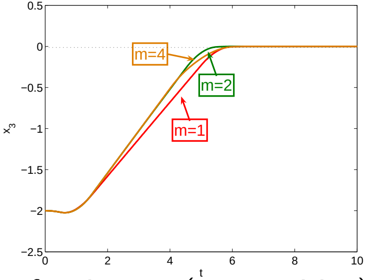
x_3 component of trajectory (cart position) for different m

UNIVERSITÄT BAYREUTH

Example: linearized inverted pendulum

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} u, \qquad x_0 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 0 \end{pmatrix}$$

sampling time $T = 0.5$, $\ell(x,u) = 2\|x\|_1 + 4\|u\|_1$, $N = 11$

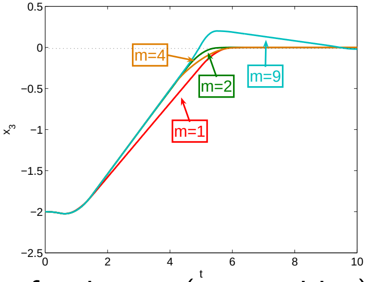


x_3 component of trajectory (cart position) for different m

Example: linearized inverted pendulum

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} u, \qquad x_0 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 0 \end{pmatrix}$$

sampling time $T = 0.5$, $\ell(x,u) = 2\|x\|_1 + 4\|u\|_1$, $N = 11$

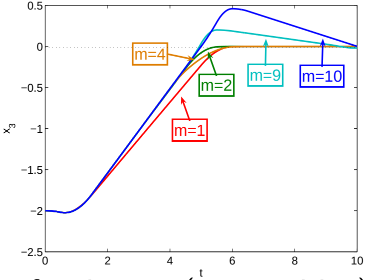


x_3 component of trajectory (cart position) for different m

Example: linearized inverted pendulum

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} u, \qquad x_0 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 0 \end{pmatrix}$$

sampling time $T = 0.5$, $\ell(x,u) = 2\|x\|_1 + 4\|u\|_1$, $N = 11$

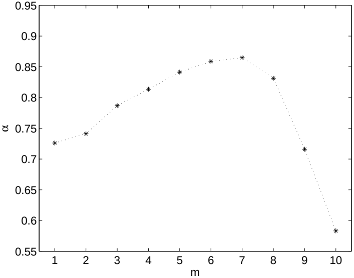


x_3 component of trajectory (cart position) for different m

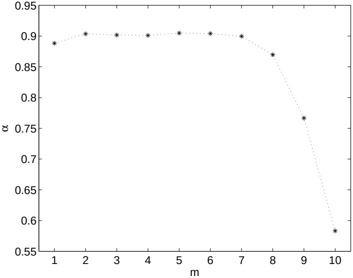
Example: linearized inverted pendulum

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} u, \qquad x_0 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 0 \end{pmatrix}$$

sampling time $T = 0.5$, $\ell(x,u) = 2\|x\|_1 + 4\|u\|_1$, $N = 11$



α after 1 MPC step



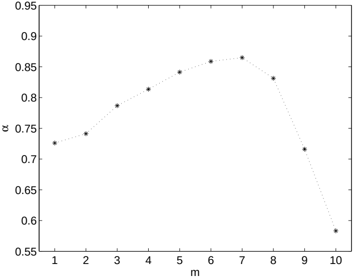
α at time $n = 20$

UNIVERSITÄT BAYREUTH

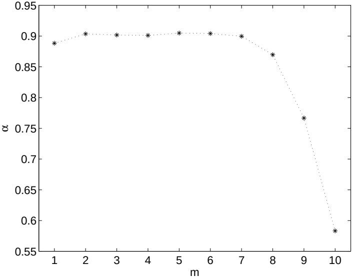
Example: linearized inverted pendulum

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} u, \qquad x_0 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 0 \end{pmatrix}$$

sampling time $T = 0.5$, $\ell(x,u) = 2\|x\|_1 + 4\|u\|_1$, $N = 11$



α after 1 MPC step



α at time $n = 20$

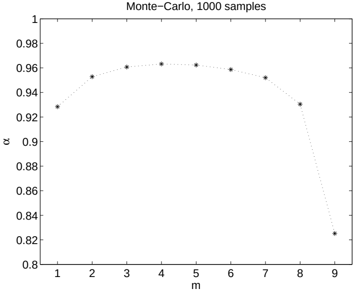
Symmetry is not present in this example

UNIVERSITÄT BAYREUTH

Monte Carlo simulation

Alternative to worst case approach: probabilistic analysis:

We generate random trajectories satisfying the LP-optimality conditions derived from the C, σ-exponential controllability condition and compute α by Monte Carlo simulation



This results are qualitatively similar to the numerical simulations

Summary of Part 2

- Stability of unconstrained MPC problems can be ensured using exponential controllability conditions

Summary of Part 2

- Stability of unconstrained MPC problems can be ensured using exponential controllability conditions
- First proofs used convergence $V_N \to V_\infty$ in order to establish stability

Summary of Part 2

- Stability of unconstrained MPC problems can be ensured using exponential controllability conditions
- First proofs used convergence $V_N \to V_\infty$ in order to establish stability
- Tighter and more useful estimates can be obtained by using optimality conditions for (pieces of) trajectories

Summary of Part 2

- Stability of unconstrained MPC problems can be ensured using exponential controllability conditions
- First proofs used convergence $V_N \to V_\infty$ in order to establish stability
- Tighter and more useful estimates can be obtained by using optimality conditions for (pieces of) trajectories
- The conditions lead to an explicitly solvable linear program

Summary of Part 2

- Stability of unconstrained MPC problems can be ensured using exponential controllability conditions
- First proofs used convergence $V_N \to V_\infty$ in order to establish stability
- Tighter and more useful estimates can be obtained by using optimality conditions for (pieces of) trajectories
- The conditions lead to an explicitly solvable linear program
- The knowledge obtained from this analysis can be used to design good MPC schemes by choosing suitable running costs ℓ

Summary of Part 2

- Stability of unconstrained MPC problems can be ensured using exponential controllability conditions
- First proofs used convergence $V_N \to V_\infty$ in order to establish stability
- Tighter and more useful estimates can be obtained by using optimality conditions for (pieces of) trajectories
- The conditions lead to an explicitly solvable linear program
- The knowledge obtained from this analysis can be used to design good MPC schemes by choosing suitable running costs ℓ
- The analysis can be extended to variable control horizons useful for networked control systems