A distributed NMPC scheme without stabilizing terminal constraints

Lars Grüne

Mathematical Institute, University of Bayreuth, Germany

research partially supported by

DFG Priority Research Program 1305
"Control theory for digitally networked dynamical systems"

LCCC Workshop on
"Distributed Model Predictive Control and Supply Chains"
Lund, May 18–21, 2010

Setup

We consider $m \geq 1$ nonlinear discrete time control systems

$$x_k(n+1) = f(x_k(n), u_k(n)), \quad k = 1, \ldots, m$$

with $x_k(n) \in X_k$, $u_k(n) \in U_k$, $\quad X_k$, U_k metric spaces

Setup

We consider $m \geq 1$ nonlinear discrete time control systems

$$x_k(n+1) = f(x_k(n), u_k(n)), \quad k = 1, \ldots, m$$

with $x_k(n) \in X_k$, $u_k(n) \in U_k$, $\quad X_k$, U_k metric spaces

Notation: $x_k^u(n)$: open loop solution for some $u_k(\cdot)$

$x_k(n)$: closed loop solution for some feedback law

Setup

We consider $m \geq 1$ nonlinear discrete time control systems

$$x_k(n+1) = f(x_k(n), u_k(n)), \quad k = 1, \ldots, m$$

with $x_k(n) \in X_k$, $u_k(n) \in U_k$, X_k, U_k metric spaces

Notation: $x_k^u(n)$: open loop solution for some $u_k(\cdot)$

$x_k(n)$: closed loop solution for some feedback law

Goal: stabilize each subsystem while maintaining a common state constraint

Setup

We consider $m \geq 1$ nonlinear discrete time control systems

$$x_k(n+1) = f(x_k(n), u_k(n)), \quad k = 1, \dots, m$$

with $x_k(n) \in X_k$, $u_k(n) \in U_k$, $\quad X_k$, U_k metric spaces

Notation: $x_k^u(n)$: open loop solution for some $u_k(\cdot)$

$x_k(n)$: closed loop solution for some feedback law

Goal: stabilize each subsystem while maintaining a common state constraint

Subsystems are allowed to transmit data to each other once per sampling period

Setup

We consider $m \geq 1$ nonlinear discrete time control systems

$$x_k(n+1) = f(x_k(n), u_k(n)), \quad k = 1, \ldots, m$$

with $x_k(n) \in X_k$, $u_k(n) \in U_k$, $\quad X_k$, U_k metric spaces

Notation: $x_k^u(n)$: open loop solution for some $u_k(\cdot)$

$x_k(n)$: closed loop solution for some feedback law

Goal: stabilize each subsystem while maintaining a common state constraint

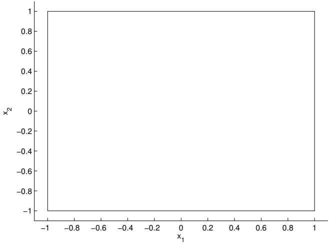
Subsystems are allowed to transmit data to each other once per sampling period

Before we give the precise problem formulation we illustrate the problem by a simple example

Example: simple mobile robots in the plane

position: $x_k = (x_{k,1}, x_{k,2}) \in [-1,1]^2$

velocity: $u_k = (u_{k,1}, u_{k,2}) \in [-\frac{1}{4}, \frac{1}{4}]^2$



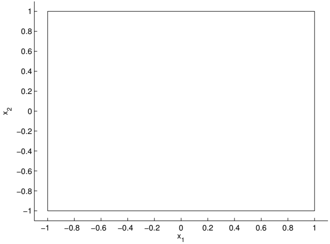
Example: simple mobile robots in the plane

position: $x_k = (x_{k,1}, x_{k,2}) \in [-1,1]^2$

velocity: $u_k = (u_{k,1}, u_{k,2}) \in [-\frac{1}{4}, \frac{1}{4}]^2$

sampling time: $T > 0$

$$\begin{aligned} x_{k,1}(n+1) &= x_{k,1}(n) + T u_{k,1}(n) \\ x_{k,2}(n+1) &= x_{k,2}(n) + T u_{k,2}(n) \end{aligned}$$

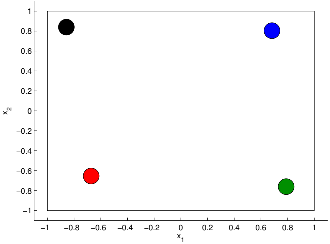


Example: simple mobile robots in the plane

position: $x_k = (x_{k,1}, x_{k,2}) \in [-1, 1]^2$

velocity: $u_k = (u_{k,1}, u_{k,2}) \in [-\frac{1}{4}, \frac{1}{4}]^2$

sampling time: $T > 0$



$$
\begin{aligned}
x_{k,1}(n+1) &= x_{k,1}(n) + T u_{k,1}(n) \\
x_{k,2}(n+1) &= x_{k,2}(n) + T u_{k,2}(n)
\end{aligned}
$$

Given: Initial values $x_k(0)$ (●)

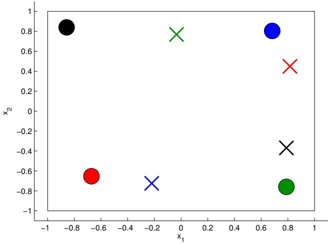
UNIVERSITÄT BAYREUTH

Example: simple mobile robots in the plane

position: $x_k = (x_{k,1}, x_{k,2}) \in [-1, 1]^2$

velocity: $u_k = (u_{k,1}, u_{k,2}) \in [-\frac{1}{4}, \frac{1}{4}]^2$

sampling time: $T > 0$



$$x_{k,1}(n+1) = x_{k,1}(n) + T u_{k,1}(n)$$
$$x_{k,2}(n+1) = x_{k,2}(n) + T u_{k,2}(n)$$

Given: Initial values $x_k(0)$ (●) and equilibria $x_k^\star$ (×)

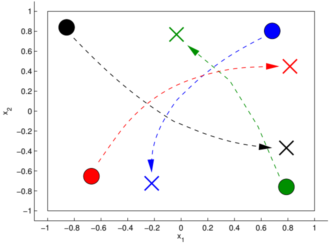
UNIVERSITÄT BAYREUTH

Example: simple mobile robots in the plane

position: $x_k = (x_{k,1}, x_{k,2}) \in [-1, 1]^2$

velocity : $u_k = (u_{k,1}, u_{k,2}) \in [-\frac{1}{4}, \frac{1}{4}]^2$

sampling time: $T > 0$



$$
\begin{aligned}
x_{k,1}(n+1) &= x_{k,1}(n) + T u_{k,1}(n) \\
x_{k,2}(n+1) &= x_{k,2}(n) + T u_{k,2}(n)
\end{aligned}
$$

Given: Initial values $x_k(0)$ ($\bullet$) and equilibria $x_k^\star$ ($\times$)

Goal: find feedback controllers which

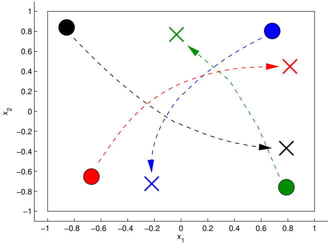
- control each robot from $x_k(0)$ to $x_k^\star$ (stabilization)

Example: simple mobile robots in the plane

position: $x_k = (x_{k,1}, x_{k,2}) \in [-1, 1]^2$

velocity: $u_k = (u_{k,1}, u_{k,2}) \in [-\frac{1}{4}, \frac{1}{4}]^2$

sampling time: $T > 0$



$$
\begin{aligned}
x_{k,1}(n+1) &= x_{k,1}(n) + T u_{k,1}(n) \\
x_{k,2}(n+1) &= x_{k,2}(n) + T u_{k,2}(n)
\end{aligned}
$$

Given: Initial values $x_k(0)$ ($\bullet$) and equilibria $x_k^\star$ ($\times$)

Goal: find feedback controllers which

- control each robot from $x_k(0)$ to $x_k^\star$ (stabilization)
- while staying in $[-1, 1]^2$ and avoiding collisions (state constraints)

Formal problem formulation

Goal: given equilibria $x_k^\star \in X_k$ and a state constraint set

$$\mathbb{X} \subset X_1 \times X_2 \times \ldots \times X_m$$

Formal problem formulation

Goal: given equilibria $x_k^\star \in X_k$ and a state constraint set

$$\mathbb{X} \subset X_1 \times X_2 \times \ldots \times X_m$$

find feedback controllers $F_k : X_k \times Y_k \to U_k$

Formal problem formulation

Goal: given equilibria $x_k^\star \in X_k$ and a state constraint set

$$\mathbb{X} \subset X_1 \times X_2 \times \ldots \times X_m$$

find feedback controllers $F_k : X_k \times Y_k \to U_k$ such that

- $x_k^\star$ is asymptotically stable for the k-th subsystem

$$x_k(n+1) = f(x_k(n), F_k(x_k(n), y_k(n)))$$

- $(x_1(0), \ldots, x_m(0)) \in \mathbb{X}$ implies $(x_1(n), \ldots, x_m(n)) \in \mathbb{X}$ for all $n \geq 0$

Formal problem formulation

Goal: given equilibria $x_k^\star \in X_k$ and a state constraint set

$$\mathbb{X} \subset X_1 \times X_2 \times \ldots \times X_m$$

find feedback controllers $F_k : X_k \times Y_k \to U_k$ such that

- $x_k^\star$ is asymptotically stable for the k-th subsystem

$$x_k(n+1) = f(x_k(n), F_k(x_k(n), y_k(n)))$$

- $(x_1(0), \ldots, x_m(0)) \in \mathbb{X}$ implies $(x_1(n), \ldots, x_m(n)) \in \mathbb{X}$ for all $n \geq 0$

Here $y_k(n) \in Y_k$ is data transmitted from the other subsystems

Formal problem formulation

Goal: given equilibria $x_k^\star \in X_k$ and a state constraint set

$$\mathbb{X} \subset X_1 \times X_2 \times \ldots \times X_m$$

find feedback controllers $F_k : X_k \times Y_k \to U_k$ such that

- $x_k^\star$ is asymptotically stable for the k-th subsystem

$$x_k(n+1) = f(x_k(n), F_k(x_k(n), y_k(n)))$$

- $(x_1(0), \ldots, x_m(0)) \in \mathbb{X}$ implies $(x_1(n), \ldots, x_m(n)) \in \mathbb{X}$ for all $n \geq 0$

Here $y_k(n) \in Y_k$ is data transmitted from the other subsystems

Example: state constraints for mobile robots

$$\mathbb{X} = \{(x_1, \ldots, x_m) \in [-1,1]^{2m} \mid \|x_k - x_l\| \geq \delta \text{ for } k \neq l\}$$

Formal problem formulation

Goal: given equilibria $x_k^\star \in X_k$ and a state constraint set

$$\mathbb{X} \subset X_1 \times X_2 \times \ldots \times X_m$$

find feedback controllers $F_k : X_k \times Y_k \to U_k$ such that

- $x_k^\star$ is asymptotically stable for the k-th subsystem

$$x_k(n+1) = f(x_k(n), F_k(x_k(n), y_k(n)))$$

- $(x_1(0), \ldots, x_m(0)) \in \mathbb{X}$ implies $(x_1(n), \ldots, x_m(n)) \in \mathbb{X}$ for all $n \geq 0$

Here $y_k(n) \in Y_k$ is data transmitted from the other subsystems

Example: state constraints for mobile robots

$$\mathbb{X} = \{(x_1, \ldots, x_m) \in [-1, 1]^{2m} \mid \|x_k - x_l\| \geq \delta \text{ for } k \neq l\}$$

Idea: use model predictive control (MPC)

The basic MPC concept

We recall basic MPC concepts for one subsystem, i.e., $m = 1$.

The basic MPC concept

We recall basic MPC concepts for one subsystem, i.e., $m = 1$.

At each time instant n solve for the current state $x = x(n)$

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

where ℓ penalizes the distance to the equilibrium $x^\star$

UNIVERSITÄT BAYREUTH

The basic MPC concept

We recall basic MPC concepts for one subsystem, i.e., $m = 1$.

At each time instant n solve for the current state $x = x(n)$

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

where ℓ penalizes the distance to the equilibrium $x^\star$

$\rightsquigarrow$ optimal trajectory $\quad x^{opt}(0), \ldots, x^{opt}(N-1)$

The basic MPC concept

We recall basic MPC concepts for one subsystem, i.e., $m = 1$.

At each time instant n solve for the current state $x = x(n)$

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

where ℓ penalizes the distance to the equilibrium $x^\star$

$\rightsquigarrow$ optimal trajectory $x^{opt}(0), \ldots, x^{opt}(N-1)$

with optimal control $u^{opt}(0), \ldots, u^{opt}(N-1)$

UNIVERSITÄT BAYREUTH

The basic MPC concept

We recall basic MPC concepts for one subsystem, i.e., $m = 1$.

At each time instant n solve for the current state $x = x(n)$

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

where ℓ penalizes the distance to the equilibrium $x^\star$

$\rightsquigarrow$ optimal trajectory $x^{opt}(0), \ldots, x^{opt}(N-1)$

with optimal control $u^{opt}(0), \ldots, u^{opt}(N-1)$

$\rightsquigarrow$ MPC feedback law $F(x(n)) := u^{opt}(0)$

The basic MPC concept

We recall basic MPC concepts for one subsystem, i.e., $m = 1$.

At each time instant n solve for the current state $x = x(n)$

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

where ℓ penalizes the distance to the equilibrium $x^\star$

$\rightsquigarrow$ optimal trajectory $\quad x^{opt}(0), \dots, x^{opt}(N-1)$

with optimal control $u^{opt}(0), \dots, u^{opt}(N-1)$

$\rightsquigarrow$ MPC feedback law $F(x(n)) := u^{opt}(0)$

$\rightsquigarrow$ closed loop

$$x(n+1) = f(x(n), F(x(n))) = f(x^{opt}(0), u^{opt}(0)) = x^{opt}(1)$$

The basic MPC concept

We recall basic MPC concepts for one subsystem, i.e., $m = 1$.

At each time instant n solve for the current state $x = x(n)$

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

where ℓ penalizes the distance to the equilibrium $x^\star$

$\rightsquigarrow$ optimal trajectory $x^{opt}(0), \dots, x^{opt}(N-1)$

with optimal control $u^{opt}(0), \dots, u^{opt}(N-1)$

$\rightsquigarrow$ MPC feedback law $F(x(n)) := u^{opt}(0)$

$\rightsquigarrow$ closed loop

$$x(n+1) = f(x(n), F(x(n))) = f(x^{opt}(0), u^{opt}(0)) = x^{opt}(1)$$

How can we guarantee asymptotic stability of the closed loop?

Stabilizing terminal constraints

At time n solve for the current state $x = x(n)$

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

UNIVERSITÄT BAYREUTH

Stabilizing terminal constraints

At time n solve for the current state $x = x(n)$

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

Stability can be guaranteed by adding the terminal constraint

$$x^u(N) = x^\star$$

(point constraint, [Keerthi/Gilbert '88, ...])

UNIVERSITÄT BAYREUTH

Stabilizing terminal constraints

At time n solve for the current state $x = x(n)$

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

Stability can be guaranteed by adding the terminal constraint

$$x^u(N) = x^\star$$

(point constraint, [Keerthi/Gilbert '88, ...])

or variants like $x^u(N) \in \mathcal{N}(x^\star)$ plus terminal costs

(regional constraint, [Chen/Allgöwer '98, ...])

Stabilizing terminal constraints

Typical stability result with stabilizing terminal constraints:

Stabilizing terminal constraints

Typical stability result with stabilizing terminal constraints:

Theorem: Assume that each $x \in \mathbb{X}$ is feasible, i.e., there exists $x^u(\cdot)$ with $x^u(0) = x$, $x^u(N)$ satisfying the terminal constraints and

$$x^u(n) \in \mathbb{X}, \quad n = 1, \ldots, N-1.$$

Then F stabilizes the system maintaining the state constraints $\mathbb{X}$

Stabilizing terminal constraints

Typical stability result with stabilizing terminal constraints:

Theorem: Assume that each $x \in \mathbb{X}$ is feasible, i.e., there exists $x^u(\cdot)$ with $x^u(0) = x$, $x^u(N)$ satisfying the terminal constraints and

$$x^u(n) \in \mathbb{X}, \quad n = 1, \dots, N-1.$$

Then F stabilizes the system maintaining the state constraints $\mathbb{X}$

In the robot problem with one robot and a static obstacle, in the simulation with point terminal constraint this is satisfied for $N = 18$

UNIVERSITÄT
BAYREUTH

Stabilizing terminal constraints

Typical stability result with stabilizing terminal constraints:

Theorem: Assume that each $x \in \mathbb{X}$ is feasible, i.e., there exists $x^u(\cdot)$ with $x^u(0) = x$, $x^u(N)$ satisfying the terminal constraints and

$$x^u(n) \in \mathbb{X}, \quad n = 1, \ldots, N-1.$$

Then F stabilizes the system maintaining the state constraints $\mathbb{X}$

In the robot problem with one robot and a static obstacle, in the simulation with point terminal constraint this is satisfied for $N = 18$

Regional terminal constraints may allow for smaller N but the construction of terminal costs becomes difficult if the terminal region contains an obstacle

No stabilizing terminal constraints

Without stabilizing terminal constraints, stability can be guaranteed under conditions like, e.g.,

No stabilizing terminal constraints

Without stabilizing terminal constraints, stability can be guaranteed under conditions like, e.g.,

- rank conditions [Alamir/Bornard '95]

No stabilizing terminal constraints

Without stabilizing terminal constraints, stability can be guaranteed under conditions like, e.g.,

- rank conditions [Alamir/Bornard '95]
- controllability conditions [Jadbabaie/Hauser '05, Gr. '09, Gr./Pannek/Seehafer/Worthmann '10]

No stabilizing terminal constraints

Without stabilizing terminal constraints, stability can be guaranteed under conditions like, e.g.,

- rank conditions [Alamir/Bornard '95]
- controllability conditions [Jadbabaie/Hauser '05, Gr. '09, Gr./Pannek/Seehafer/Worthmann '10]
- detectability conditions [Grimm/Messina/Teel/Tuna '05ff]

No stabilizing terminal constraints

Without stabilizing terminal constraints, stability can be guaranteed under conditions like, e.g.,

- rank conditions [Alamir/Bornard '95]
- controllability conditions [Jadbabaie/Hauser '05, Gr. '09, Gr./Pannek/Seehafer/Worthmann '10]
- detectability conditions [Grimm/Messina/Teel/Tuna '05ff]
- relaxed dynamic programming conditions [Gr./Rantzer '08]
- ...

UNIVERSITÄT BAYREUTH

No stabilizing terminal constraints

Without stabilizing terminal constraints, stability can be guaranteed under conditions like, e.g.,

- rank conditions [Alamir/Bornard '95]
- controllability conditions [Jadbabaie/Hauser '05, Gr. '09, Gr./Pannek/Seehafer/Worthmann '10]
- detectability conditions [Grimm/Messina/Teel/Tuna '05ff]
- relaxed dynamic programming conditions [Gr./Rantzer '08]
- ...

and sufficiently large optimization horizon N

No stabilizing terminal constraints

Without stabilizing terminal constraints, stability can be guaranteed under conditions like, e.g.,

- rank conditions [Alamir/Bornard '95]
- controllability conditions [Jadbabaie/Hauser '05, Gr. '09, ← Gr./Pannek/Seehafer/Worthmann '10]
- detectability conditions [Grimm/Messina/Teel/Tuna '05ff]
- relaxed dynamic programming conditions [Gr./Rantzer '08]
- ...

and sufficiently large optimization horizon N

No stabilizing terminal constraints

Typical stability result without stabilizing terminal constraints:

No stabilizing terminal constraints

Typical stability result without stabilizing terminal constraints:

Theorem: Assume that each $x \in \mathbb{X}$ is feasibly exponentially controllable through ℓ, i.e., there exists $u(\cdot)$ with $x^u(0) = x$,

$$x^u(n) \in \mathbb{X} \quad \text{and} \quad \ell(x^u(n), u(n)) \leq C\sigma^n \min_u \ell(x, u)$$

for $n = 0, \ldots, N-1$ with $C > 0$, $\sigma \in (0, 1)$ independent of x.

No stabilizing terminal constraints

Typical stability result without stabilizing terminal constraints:

Theorem: Assume that each $x \in \mathbb{X}$ is feasibly exponentially controllable through ℓ, i.e., there exists $u(\cdot)$ with $x^u(0) = x$,

$$x^u(n) \in \mathbb{X} \quad \text{and} \quad \ell(x^u(n), u(n)) \le C\sigma^n \min_u \ell(x, u)$$

for $n = 0, \ldots, N-1$ with $C > 0$, $\sigma \in (0,1)$ independent of x.

Then F stabilizes the system under the state constraints $\mathbb{X}$ if

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

UNIVERSITÄT BAYREUTH

No stabilizing terminal constraints

Typical stability result without stabilizing terminal constraints:

Theorem: Assume that each $x \in \mathbb{X}$ is feasibly exponentially controllable through ℓ, i.e., there exists $u(\cdot)$ with $x^u(0) = x$,

$$x^u(n) \in \mathbb{X} \quad \text{and} \quad \ell(x^u(n), u(n)) \leq C\sigma^n \min_u \ell(x,u)$$

for $n = 0, \ldots, N-1$ with $C > 0$, $\sigma \in (0,1)$ independent of x.

Then F stabilizes the system under the state constraints $\mathbb{X}$ if

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

Note: $\alpha \to 1$ as $N \to \infty$

UNIVERSITÄT BAYREUTH

No stabilizing terminal constraints

Typical stability result without stabilizing terminal constraints:

Theorem: Assume that each $x \in \mathbb{X}$ is feasibly exponentially controllable through ℓ, i.e., there exists $u(\cdot)$ with $x^u(0) = x$,

$$x^u(n) \in \mathbb{X} \quad \text{and} \quad \ell(x^u(n), u(n)) \leq C\sigma^n \min_u \ell(x, u)$$

for $n = 0, \ldots, N-1$ with $C > 0$, $\sigma \in (0, 1)$ independent of x.

Then F stabilizes the system under the state constraints $\mathbb{X}$ if

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

Note: $\alpha \to 1$ as $N \to \infty$

In the one-robot problem with $\ell(x, u) = \|x - x^\star\|^2 + \|u\|^2/10$, in simulations stability is obtained for $N = 3$

UNIVERSITÄT BAYREUTH

Common idea in the proofs

Both with and without stabilizing terminal constraints, the stability proof relies on establishing the inequality

$$V_N(x(n+1)) < V_N(x(n))$$

for the optimal value function $V_N(x) = \inf_{u \in \mathcal{U}} J_N(x,u)$ in a suitable uniform way

UNIVERSITÄT BAYREUTH

Common idea in the proofs

Both with and without stabilizing terminal constraints, the stability proof relies on establishing the inequality

$$V_N(x(n+1)) < V_N(x(n))$$

for the optimal value function $V_N(x) = \inf_{u \in \mathcal{U}} J_N(x,u)$ in a suitable uniform way $\rightsquigarrow$ V_N is a Lyapunov function

Common idea in the proofs

Both with and without stabilizing terminal constraints, the stability proof relies on establishing the inequality

$$V_N(x(n+1)) < V_N(x(n))$$

for the optimal value function $V_N(x) = \inf_{u \in \mathcal{U}} J_N(x,u)$ in a suitable uniform way $\rightsquigarrow$ V_N is a Lyapunov function

To this end, the proofs use the tail

$$x^{opt}(1), \ldots, x^{opt}(N)$$

of the optimal trajectory at time n with $x^{opt}(0) = x(n)$ to construct feasible trajectories at time $n+1$ for initial value $x(n+1) = x^{opt}(1)$

UNIVERSITÄT BAYREUTH

Common idea in the proofs

Both with and without stabilizing terminal constraints, the stability proof relies on establishing the inequality

$$V_N(x(n+1)) < V_N(x(n))$$

for the optimal value function $V_N(x) = \inf_{u \in \mathcal{U}} J_N(x,u)$ in a suitable uniform way $\rightsquigarrow$ V_N is a Lyapunov function

To this end, the proofs use the tail

$$x^{opt}(1), \ldots, x^{opt}(N)$$

of the optimal trajectory at time n with $x^{opt}(0) = x(n)$ to construct feasible trajectories at time $n+1$ for initial value $x(n+1) = x^{opt}(1)$ $\rightsquigarrow$ upper bound for $V_N(x(n+1))$

Common idea in the proofs

Both with and without stabilizing terminal constraints, the stability proof relies on establishing the inequality

$$V_N(x(n+1)) < V_N(x(n))$$

for the optimal value function $V_N(x) = \inf_{u \in \mathcal{U}} J_N(x,u)$ in a suitable uniform way $\rightsquigarrow$ V_N is a Lyapunov function

To this end, the proofs use the tail

$$x^{opt}(1), \ldots, x^{opt}(N)$$

of the optimal trajectory at time n with $x^{opt}(0) = x(n)$ to construct feasible trajectories at time $n+1$ for initial value $x(n+1) = x^{opt}(1)$ $\rightsquigarrow$ upper bound for $V_N(x(n+1))$

$\rightsquigarrow$ crucial for extension to the distributed context: $x_k^{opt}(1), \ldots, x_k^{opt}(N)$ for subsystem x_k must remain feasible when the other subsystems x_l, $l \neq k$, update their prediction

UNIVERSITÄT BAYREUTH

A hierarchical distributed MPC scheme

[Richards/How, ACC '04, IJC '07] propose the following hierarchical MPC scheme with terminal constraints:

A hierarchical distributed MPC scheme

[Richards/How, ACC '04, IJC '07] propose the following hierarchical MPC scheme with terminal constraints:

At each sampling instant n:

A hierarchical distributed MPC scheme

[Richards/How, ACC '04, IJC '07] propose the following hierarchical MPC scheme with terminal constraints:

At each sampling instant n:

for k from 1 to m

end of k-loop

A hierarchical distributed MPC scheme

[Richards/How, ACC '04, IJC '07] propose the following hierarchical MPC scheme with terminal constraints:

At each sampling instant n:

for k from 1 to m

subsystem k computes (and transmits) its optimal prediction $x_k^{opt,n}(0), \ldots, x_k^{opt,n}(N)$ taking into account its own position $x_k^{opt,n}(0) = x_k(n)$ at time n and

end of k-loop

UNIVERSITÄT BAYREUTH

A hierarchical distributed MPC scheme

[Richards/How, ACC '04, IJC '07] propose the following hierarchical MPC scheme with terminal constraints:

At each sampling instant n:

for k from 1 to m

subsystem k computes (and transmits) its optimal prediction $x_k^{opt,n}(0), \ldots, x_k^{opt,n}(N)$ taking into account its own position $x_k^{opt,n}(0) = x_k(n)$ at time n and

- the predictions $x_l^{opt,n}$ for $l = 1, \ldots, k-1$

end of k-loop

UNIVERSITÄT BAYREUTH

A hierarchical distributed MPC scheme

[Richards/How, ACC '04, IJC '07] propose the following hierarchical MPC scheme with terminal constraints:

At each sampling instant n:

for k from 1 to m

subsystem k computes (and transmits) its optimal prediction $x_k^{opt,n}(0), \ldots, x_k^{opt,n}(N)$ taking into account its own position $x_k^{opt,n}(0) = x_k(n)$ at time n and

- the predictions $x_l^{opt,n}$ for $l = 1, \ldots, k-1$
- the predictions $x_p^{opt,n-1}$ for $p = k+1, \ldots, m$

end of k-loop

A hierarchical distributed MPC scheme

[Richards/How, ACC '04, IJC '07] propose the following hierarchical MPC scheme with terminal constraints:

At each sampling instant n:

for k from 1 to m

subsystem k computes (and transmits) its optimal prediction $x_k^{opt,n}(0), \ldots, x_k^{opt,n}(N)$ taking into account its own position $x_k^{opt,n}(0) = x_k(n)$ at time n and

- the predictions $x_l^{opt,n}$ for $l = 1, \ldots, k-1$
- the predictions $x_p^{opt,n-1}$ for $p = k+1, \ldots, m$

such that for $j = 0, \ldots, N-1$:

$$(x_1^{opt,n}(j), \ldots, x_k^{opt,n}(j), x_{k+1}^{opt,n-1}(j+1), \ldots, x_m^{opt,n-1}(j+1)) \in \mathbb{X}$$

end of k-loop

A hierarchical distributed MPC scheme

[Richards/How, ACC '04, IJC '07] propose the following hierarchical MPC scheme with terminal constraints:

At each sampling instant n:

for k from 1 to m

subsystem k computes (and transmits) its optimal prediction $x_k^{opt,n}(0), \ldots, x_k^{opt,n}(N)$ taking into account its own position $x_k^{opt,n}(0) = x_k(n)$ at time n and

- the predictions $x_l^{opt,n}$ for $l = 1, \ldots, k-1$
- the predictions $x_p^{opt,n-1}$ for $p = k+1, \ldots, m$

such that for $j = 0, \ldots, N-1$:

$$(x_1^{opt,n}(j), \ldots, x_k^{opt,n}(j), x_{k+1}^{opt,n-1}(j+1), \ldots, x_m^{opt,n-1}(j+1)) \in \mathbb{X}$$

end of k-loop

all systems apply the resulting feedback control value

A hierarchical distributed MPC scheme

What is Y_k in this scheme?

A hierarchical distributed MPC scheme

What is Y_k in this scheme?

The feedback $F_k : X_k \times Y_k \to U_k$ depends on $x_k \in X_k$ and $(x_l^{opt,n}(\cdot), x_p^{opt,n-1}(\cdot))$ for $l = 1, \ldots, k-1$, $p = k+1, \ldots, m$

A hierarchical distributed MPC scheme

What is Y_k in this scheme?

The feedback $F_k : X_k \times Y_k \to U_k$ depends on $x_k \in X_k$ and

$\underbrace{(x_l^{opt,n}(\cdot), x_p^{opt,n-1}(\cdot))}_{=:y_k(n)}$ for $l = 1, \ldots, k-1$, $p = k+1, \ldots, m$

A hierarchical distributed MPC scheme

What is Y_k in this scheme?

The feedback $F_k : X_k \times Y_k \to U_k$ depends on $x_k \in X_k$ and

$\underbrace{(x_l^{opt,n}(\cdot), x_p^{opt,n-1}(\cdot))}_{=:y_k(n)}$ for $l = 1, \ldots, k-1$, $p = k+1, \ldots, m$

$\rightsquigarrow Y_k = X_1^{N+1} \times \ldots \times X_{k-1}^{N+1} \times X_{k+1}^{N+1} \times \ldots \times X_m^{N+1}$

A hierarchical distributed MPC scheme

What is Y_k in this scheme?

The feedback $F_k : X_k \times Y_k \to U_k$ depends on $x_k \in X_k$ and

$\underbrace{(x_l^{opt,n}(\cdot), x_p^{opt,n-1}(\cdot))}_{=:y_k(n)}$ for $l = 1, \ldots, k-1$, $p = k+1, \ldots, m$

$\rightsquigarrow Y_k = X_1^{N+1} \times \ldots \times X_{k-1}^{N+1} \times X_{k+1}^{N+1} \times \ldots \times X_m^{N+1}$

How is the scheme initialized?

A hierarchical distributed MPC scheme

What is Y_k in this scheme?

The feedback $F_k : X_k \times Y_k \to U_k$ depends on $x_k \in X_k$ and

$\underbrace{(x_l^{opt,n}(\cdot), x_p^{opt,n-1}(\cdot))}_{=:y_k(n)}$ for $l = 1, \ldots, k-1$, $p = k+1, \ldots, m$

$\leadsto Y_k = X_1^{N+1} \times \ldots \times X_{k-1}^{N+1} \times X_{k+1}^{N+1} \times \ldots \times X_m^{N+1}$

How is the scheme initialized?

At time $n = 0$ we start with arbitrary, i.e., not necessarily optimal feasible solutions. These can be found by optimization or any other method

Stability theorem with terminal constraints

Theorem [Richards/How]: Assume that for the initial values $x_k(0)$, $k = 1, \ldots, m$, we can find feasible solutions, i.e., $x_k^u(\cdot)$ with $x_k^u(0) = x_k(0)$, $x_k^u(N)$ satisfying the terminal constraints and

$$(x_1^u(n), \ldots, x_m^u(n)) \in \mathbb{X}, \quad n = 0, \ldots, N-1$$

Stability theorem with terminal constraints

Theorem [Richards/How]: Assume that for the initial values $x_k(0)$, $k = 1, \ldots, m$, we can find feasible solutions, i.e., $x_k^u(\cdot)$ with $x_k^u(0) = x_k(0)$, $x_k^u(N)$ satisfying the terminal constraints and

$$(x_1^u(n), \ldots, x_m^u(n)) \in \mathbb{X}, \quad n = 0, \ldots, N-1$$

Then, initializing the hierarchical scheme at $n = 0$ with the corresponding $u_k(\cdot)$ and $F_k(x_k(0)) := u_k(0)$, the resulting distributed MPC feedback laws F_k feasibly stabilize all subsystems

Stability theorem with terminal constraints

Theorem [Richards/How]: Assume that for the initial values $x_k(0)$, $k = 1, \dots, m$, we can find feasible solutions, i.e., $x_k^u(\cdot)$ with $x_k^u(0) = x_k(0)$, $x_k^u(N)$ satisfying the terminal constraints and

$$(x_1^u(n), \dots, x_m^u(n)) \in \mathbb{X}, \quad n = 0, \dots, N-1$$

Then, initializing the hierarchical scheme at $n = 0$ with the corresponding $u_k(\cdot)$ and $F_k(x_k(0)) := u_k(0)$, the resulting distributed MPC feedback laws F_k feasibly stabilize all subsystems

For the mobile robot example with 4 robots, in the simulation we need $N = 18$ to satisfy the initial feasibility assumption

Stability theorem with terminal constraints

Theorem [Richards/How]: Assume that for the initial values $x_k(0)$, $k = 1, \ldots, m$, we can find feasible solutions, i.e., $x_k^u(\cdot)$ with $x_k^u(0) = x_k(0)$, $x_k^u(N)$ satisfying the terminal constraints and

$$(x_1^u(n), \ldots, x_m^u(n)) \in \mathbb{X}, \quad n = 0, \ldots, N-1$$

Then, initializing the hierarchical scheme at $n = 0$ with the corresponding $u_k(\cdot)$ and $F_k(x_k(0)) := u_k(0)$, the resulting distributed MPC feedback laws F_k feasibly stabilize all subsystems

For the mobile robot example with 4 robots, in the simulation we need $N = 18$ to satisfy the initial feasibility assumption

Observations:

- rather long horizons N needed (as for one robot)

Stability theorem with terminal constraints

Theorem [Richards/How]: Assume that for the initial values $x_k(0)$, $k = 1, \ldots, m$, we can find feasible solutions, i.e., $x_k^u(\cdot)$ with $x_k^u(0) = x_k(0)$, $x_k^u(N)$ satisfying the terminal constraints and

$$(x_1^u(n), \ldots, x_m^u(n)) \in \mathbb{X}, \quad n = 0, \ldots, N-1$$

Then, initializing the hierarchical scheme at $n = 0$ with the corresponding $u_k(\cdot)$ and $F_k(x_k(0)) := u_k(0)$, the resulting distributed MPC feedback laws F_k feasibly stabilize all subsystems

For the mobile robot example with 4 robots, in the simulation we need $N = 18$ to satisfy the initial feasibility assumption

Observations:

- rather long horizons N needed (as for one robot)
- all “conflicts” are resolved in the initialization phase

Removing the terminal constraints

The proof of the preceding theorem uses the fact that due to the hierarchical structure each prediction remains feasible when the other subsystems update their predictions

Removing the terminal constraints

The proof of the preceding theorem uses the fact that due to the hierarchical structure each prediction remains feasible when the other subsystems update their predictions

$\rightsquigarrow$ usual argument in the stability proof applies

Removing the terminal constraints

The proof of the preceding theorem uses the fact that due to the hierarchical structure each prediction remains feasible when the other subsystems update their predictions

$\rightsquigarrow$ usual argument in the stability proof applies

Since our controllability based stability proof without terminal constraints uses a similar argument, removal of the terminal constraints should be possible

Removing the terminal constraints

The proof of the preceding theorem uses the fact that due to the hierarchical structure each prediction remains feasible when the other subsystems update their predictions
$\rightsquigarrow$ usual argument in the stability proof applies

Since our controllability based stability proof without terminal constraints uses a similar argument, removal of the terminal constraints should be possible

Question: What is a suitable "distributed" controllability condition?

Removing the terminal constraints

The proof of the preceding theorem uses the fact that due to the hierarchical structure each prediction remains feasible when the other subsystems update their predictions
$\rightsquigarrow$ usual argument in the stability proof applies

Since our controllability based stability proof without terminal constraints uses a similar argument, removal of the terminal constraints should be possible

Question: What is a suitable "distributed" controllability condition?

For defining such a condition, we need some more notation

Partial state constraints

Recall: the state space of the overall system is

$$X = X_1 \times X_2 \times \ldots \times X_m$$

Partial state constraints

Recall: the state space of the overall system is

$$X = X_1 \times X_2 \times \ldots \times X_m$$

For an index set $I = \{i_1, \ldots, i_p\} \subseteq M := \{1, \ldots, m\}$ define the partial state space

$$X_I := X_{i_1} \times X_{i_2} \times \ldots \times X_{i_p}$$

Partial state constraints

Recall: the state space of the overall system is

$$X = X_1 \times X_2 \times \ldots \times X_m$$

For an index set $I = \{i_1, \ldots, i_p\} \subseteq M := \{1, \ldots, m\}$ define the partial state space

$$X_I := X_{i_1} \times X_{i_2} \times \ldots \times X_{i_p}$$

Elements of X_I are called partial states and denoted by

$$x_I = (x_{i_1}, \ldots, x_{i_p})$$

Partial state constraints

Recall: the state space of the overall system is

$$X = X_1 \times X_2 \times \ldots \times X_m$$

For an index set $I = \{i_1, \ldots, i_p\} \subseteq M := \{1, \ldots, m\}$ define the partial state space

$$X_I := X_{i_1} \times X_{i_2} \times \ldots \times X_{i_p}$$

Elements of X_I are called partial states and denoted by

$$x_I = (x_{i_1}, \ldots, x_{i_p})$$

The partial state constraint set is defined as

$$\mathbb{X}_I := \{x_I \in X_I \,|\, \text{there is } x_{M\setminus I} \in X_{M\setminus I} \text{ with } (x_1, \ldots, x_m) \in \mathbb{X}\}$$

UNIVERSITÄT BAYREUTH

Distributed controllability

In words: "no matter what the others intend to do, the k-th subsystem can find a feasible way towards its equilibrium, provided it knows what the others intend to do"

Distributed controllability

In words: "no matter what the others intend to do, the k-th subsystem can find a feasible way towards its equilibrium, provided it knows what the others intend to do"

Formally: at each time $n \geq 0$, denote the other subsystems' predictions available to the k-th subsystem by $x_{I_j}^{opt}(j)$, where $I_j \subset M$ with $k \notin I_j$ and the time arguments are already appropriately shifted.

Then we assume that there are $C > 0$, $\sigma \in (0,1)$ such that for each $j = 0, \ldots, N-2$ there is $u_k(\cdot)$ with

$$(x_{I_j}^{opt}(j+j'), x_k^u(j')) \in \mathbb{X}_{I_{j+j'} \cup \{k\}}$$

and

$$\ell_k(x_k^u(j'), u_k(j')) \leq C\sigma^{j'} \min_u \ell_k(x_k^u(0), u)$$

for $j' = 0, \ldots, N-j-1$ where $x_k^u(0) = x_k^{opt,n-1}(j+1)$.

Stability theorem without terminal constraints

Assume that the distributed controllability assumption holds.

Then the F_k stabilize all subsystems under the state constraints $\mathbb{X}$ if

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

Stability theorem without terminal constraints

Assume that the distributed controllability assumption holds.

Then the F_k stabilize all subsystems under the state constraints $\mathbb{X}$ if

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

Note: again $\alpha \to 1$ as $N \to \infty$

Stability theorem without terminal constraints

Assume that the distributed controllability assumption holds.

Then the F_k stabilize all subsystems under the state constraints $\mathbb{X}$ if

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

Note: again $\alpha \to 1$ as $N \to \infty$

In the 4 robot problem with $\ell_k(x,u) = \|x - x_k^\star\|^2 + \|u\|^2/10$, in simulations stability is obtained for $N = 5$ up to $N = 8$, depending on the initial configuration

Conclusion and discussion

- The hierarchical scheme can be extended to NMPC without stabilizing terminal constraints

Conclusion and discussion

- The hierarchical scheme can be extended to NMPC without stabilizing terminal constraints
- This may lead to considerably shorter optimization horizons N

Conclusion and discussion

- The hierarchical scheme can be extended to NMPC without stabilizing terminal constraints
- This may lead to considerably shorter optimization horizons N
- Conflicts are resolved when they are detected, not necessarily in the initialization phase

Conclusion and discussion

- The hierarchical scheme can be extended to NMPC without stabilizing terminal constraints
- This may lead to considerably shorter optimization horizons N
- Conflicts are resolved when they are detected, not necessarily in the initialization phase
- However, there are many open questions regarding both the controllability assumption and the design of the scheme.

UNIVERSITÄT BAYREUTH

Conclusion and discussion

- The hierarchical scheme can be extended to NMPC without stabilizing terminal constraints
- This may lead to considerably shorter optimization horizons N
- Conflicts are resolved when they are detected, not necessarily in the initialization phase
- However, there are many open questions regarding both the controllability assumption and the design of the scheme. Some of these will be discussed on the remaining three slides — and maybe during and after this workshop

UNIVERSITÄT BAYREUTH

Discussion: distributed controllability

- The distributed controllability assumption is defined implicitly via the a priori unknown predictions
  $\rightsquigarrow$ difficult to check

Discussion: distributed controllability

- The distributed controllability assumption is defined implicitly via the a priori unknown predictions
  $\rightsquigarrow$ difficult to check

- a sufficient condition is that controllability holds for all possible trajectories — this is easier to check but very restrictive

Discussion: distributed controllability

- The distributed controllability assumption is defined implicitly via the a priori unknown predictions
  $\rightsquigarrow$ difficult to check

- a sufficient condition is that controllability holds for all possible trajectories — this is easier to check but very restrictive

- In the robot example, this sufficient condition holds if m is relatively small and there are no encounters at the boundary of the state space

Discussion: distributed controllability

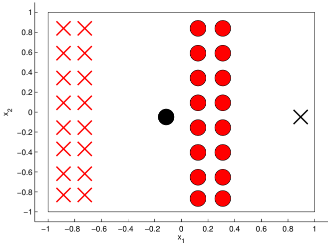
- The distributed controllability assumption is defined implicitly via the a priori unknown predictions
  $\rightsquigarrow$ difficult to check

- a sufficient condition is that controllability holds for all possible trajectories — this is easier to check but very restrictive

- In the robot example, this sufficient condition holds if m is relatively small and there are no encounters at the boundary of the state space

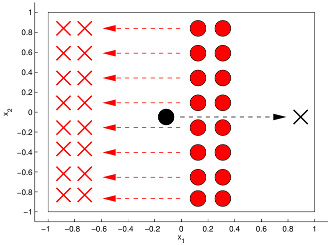
- However, even for large m, in simulations the scheme works without problems. A possible explanation is given by the following scenario.

Discussion: distributed controllability



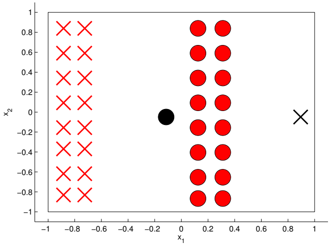
large m

Discussion: distributed controllability



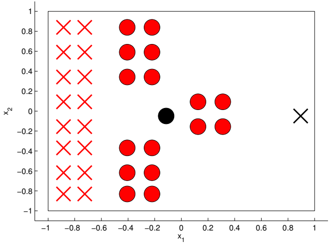
large m

Discussion: distributed controllability



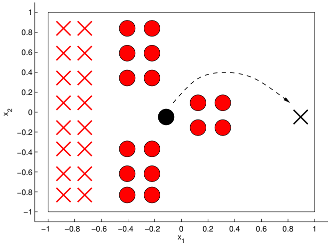
large m

Discussion: distributed controllability



large m

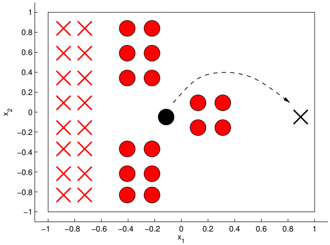
Discussion: distributed controllability



large m

- Even for large m, controllability may still be satisfied once (most of) the other subsystems are close to their equilibria

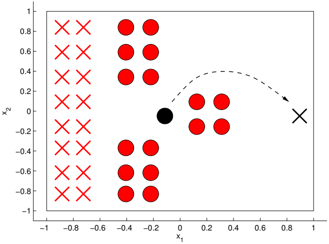
Discussion: distributed controllability



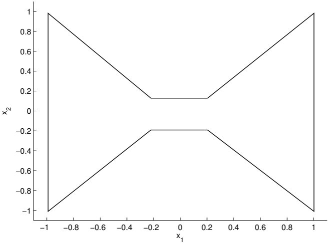
large m

- Even for large m, controllability may still be satisfied once (most of) the other subsystems are close to their equilibria
  $\rightsquigarrow$ application of small gain type arguments possible?

Discussion: distributed controllability



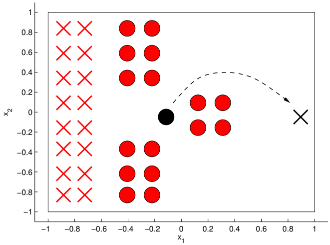
large m



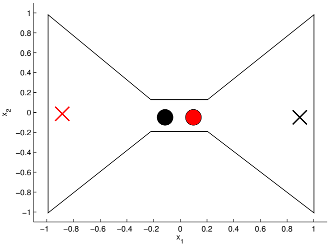
bottleneck

- Even for large m, controllability may still be satisfied once (most of) the other subsystems are close to their equilibria
  $\rightsquigarrow$ application of small gain type arguments possible?

Discussion: distributed controllability



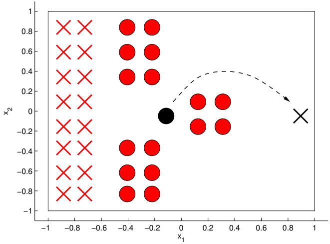
large m



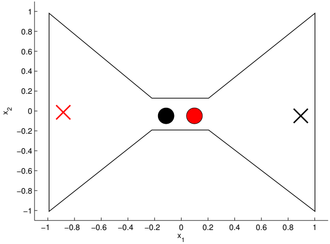
bottleneck

- Even for large m, controllability may still be satisfied once (most of) the other subsystems are close to their equilibria
  $\rightsquigarrow$ application of small gain type arguments possible?

Discussion: distributed controllability



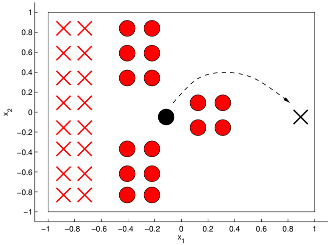
large m



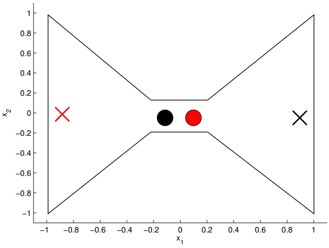
bottleneck

- Even for large m, controllability may still be satisfied once (most of) the other subsystems are close to their equilibria $\rightsquigarrow$ application of small gain type arguments possible?
- In the bottleneck case, cooperation instead of simply avoiding each other will be needed.

Discussion: distributed controllability



large m



bottleneck

- Even for large m, controllability may still be satisfied once (most of) the other subsystems are close to their equilibria $\rightsquigarrow$ application of small gain type arguments possible?
- In the bottleneck case, cooperation instead of simply avoiding each other will be needed. Can we tell self-resolvable from unresolvable deadlocks?

Discussion: hierarchical scheme

- Main drawback of the scheme: optimization must be performed sequentially in each sampling instant

Discussion: hierarchical scheme

- Main drawback of the scheme: optimization must be performed sequentially in each sampling instant
  $\rightsquigarrow$ inefficient for large m

Discussion: hierarchical scheme

- Main drawback of the scheme: optimization must be performed sequentially in each sampling instant
  $\rightsquigarrow$ inefficient for large m
  (although the overall computational complexity is lower than for centralized MPC, cf. [Richards/How '07])

Discussion: hierarchical scheme

- Main drawback of the scheme: optimization must be performed sequentially in each sampling instant
  $\rightsquigarrow$ inefficient for large m
  (although the overall computational complexity is lower than for centralized MPC, cf. [Richards/How '07])
- Simple (and provably stable) relaxation: only one subsystem optimizes at a time

Discussion: hierarchical scheme

- Main drawback of the scheme: optimization must be performed sequentially in each sampling instant
  $\rightsquigarrow$ inefficient for large m
  (although the overall computational complexity is lower than for centralized MPC, cf. [Richards/How '07])

- Simple (and provably stable) relaxation: only one subsystem optimizes at a time
  $\rightsquigarrow$ more efficient but still scales badly with growing m

Discussion: hierarchical scheme

- Main drawback of the scheme: optimization must be performed sequentially in each sampling instant
  $\rightsquigarrow$ inefficient for large m
  (although the overall computational complexity is lower than for centralized MPC, cf. [Richards/How '07])

- Simple (and provably stable) relaxation: only one subsystem optimizes at a time
  $\rightsquigarrow$ more efficient but still scales badly with growing m

- Is there a chance to obtain a truly parallel optimization with provable stability and feasibility???