Real-time online Model Predictive Control

Colin Jones, Melanie Zeilinger, Stefan Richter

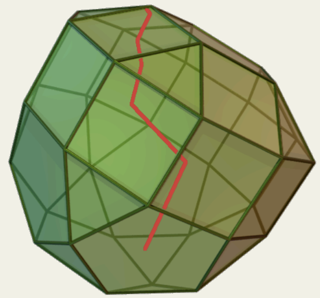
Automatic Control Laboratory, EPFL

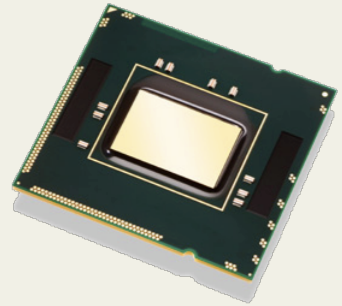
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Real-time synthesis : Complexity as a specification

MPC problem + Computational method + Embedded Processor = Real time!

$$
\begin{aligned}
J^\star(x_0) = \min_{u_i} \; & V_N(x_N) + \sum_{i=0}^{N-1} l(x_i, u_i) \\
\text{s.t. } & x_{i+1} = f(x_i, u_i) \\
& (x_i, u_i) \in \mathcal{X} \times \mathcal{U} \\
& x_N \in \mathcal{X}_N
\end{aligned}
$$





- Hardware platform bounds computation *time* and *storage*
- Current real-time explicit methods are limited to small problem dimensions

→ Online MPC can be applied to all problem dimensions

This talk: Real-time online MPC for high-speed large-scale systems

→ Fast online optimization

→ Satisfaction of real-time constraint

Fast online optimization

Many methods available:

CVX
Matlab Software for Disciplined Convex Programming
http://cvxr.com/cvx/

CVXMOD
Convex optimization software in Python
http://cvxmod.net/

CVXGEN
Code Generation for Convex Optimization
http://cvxgen.net/

qpOases
Online Active Set Strategy
http://www.kuleuven.be/optec/software/qpOASES

OOQP
Object-oriented software for quadratic programming
http://pages.cs.wisc.edu/~swright/ooqp/

QPSchur
A dual, active-set, Schur-complement method for large-scale and structured convex quadratic programming
[Bartlett et al., '06]

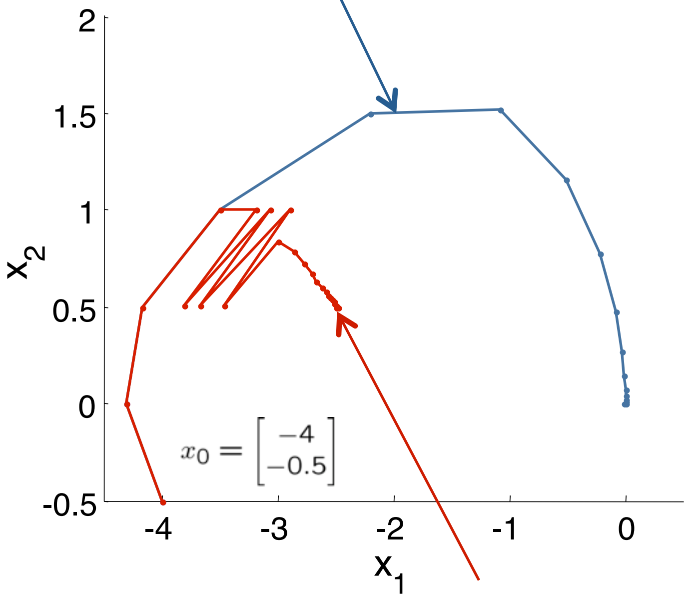
...many more

Online optimization can be applied to control high-speed systems

No guarantees on system theoretic properties when applied to MPC in a real-time setting.

Example:
Effect of limited computation time

Closed loop trajectory:
Optimal control law



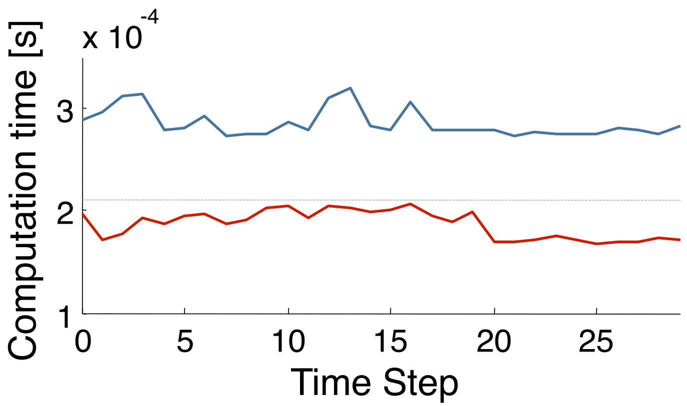
Closed loop trajectory:
Optimization stopped after 4 iterations
= max computation time of 21ms

Unstable example

$$x^+ = \begin{bmatrix} 1.2 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

$$|x_1| \le 5, -5 \le x_2 \le 1$$

$$|u| \le 1, N = 5, Q = I, R = 1$$



Limited computation time → No stability properties

Real-time online MPC: Goals

Real-time online MPC:

Guarantee that
- within the real-time constraint
- a feasible solution
- satisfying stability and performance criteria
- for any admissible initial state

is found.

NOTE: Optimality not required

We present two methods for linear systems:

Setting	Linear state and input constraints	'Simple' input constraints (e.g. box constraints)
Time scale	Milliseconds	Microseconds
Idea	Provide guarantees for any time constraint	Compute a priori bounds on the required online computation time
Approach	**Robust MPC with stability constraints**	**Fast gradient method**
	[M.N. Zeilinger et al., CDC 2009]	*[S. Richter et al., CDC 2009]*

Real-time online MPC: Goals

Real-time online MPC:

Guarantee that
- within the real-time constraint
- a feasible solution
- satisfying stability and performance criteria
- for any admissible initial state

is found.

NOTE: Optimality not required

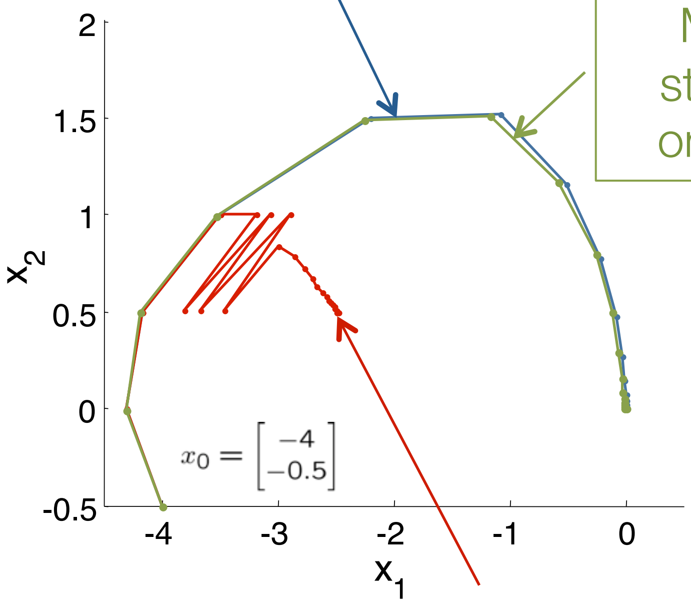
We present two methods for linear systems:

Setting	Linear state and input constraints	'Simple' input constraints (e.g. box constraints)
Time scale	Milliseconds	Microseconds
Idea	Provide guarantees for any time constraint	Compute a priori bounds on the required online computation time
Approach	**Robust MPC with stability constraints**	**Fast gradient method**
	[M.N. Zeilinger et al., CDC 2009]	*[S. Richter et al., CDC 2009]*

Example:
Stability under proposed real-time method

Closed loop trajectory:
Optimal control law

Proposed real-time MPC method stopped after 4 online iterations



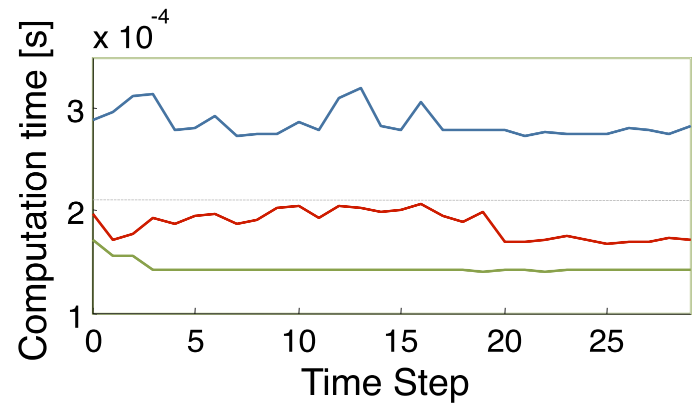
Closed loop trajectory:
Optimization stopped after 4 iterations
= max computation time of 21ms

Unstable example

$$x^+ = \begin{bmatrix} 1.2 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

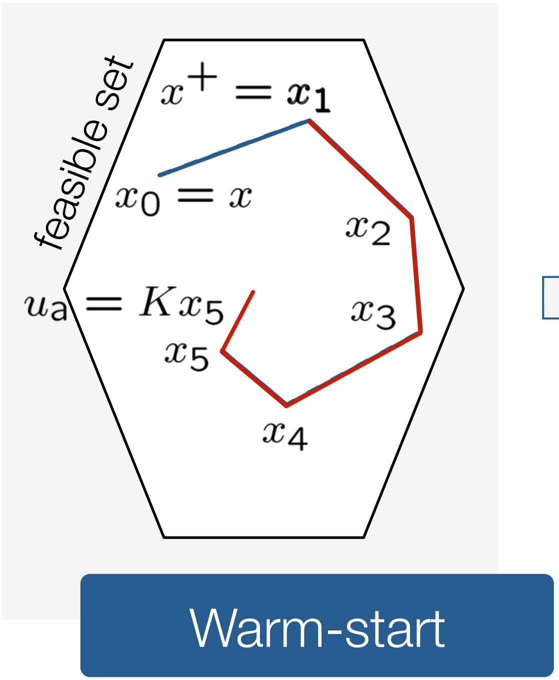
$$|x_1| \leq 5, -5 \leq x_2 \leq 1$$

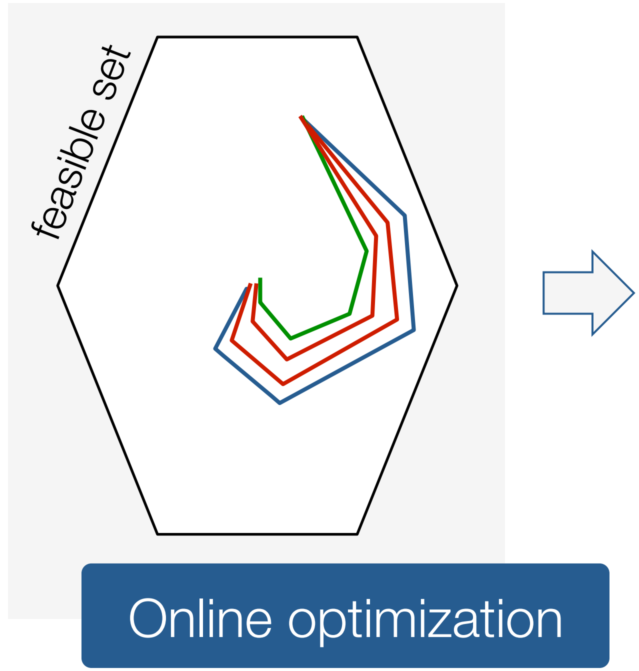
$$|u| \leq 1, N = 5, Q = I, R = 1$$



Real-time robust MPC : Nearly optimal and satisfies time constraints

Optimal MPC scheme





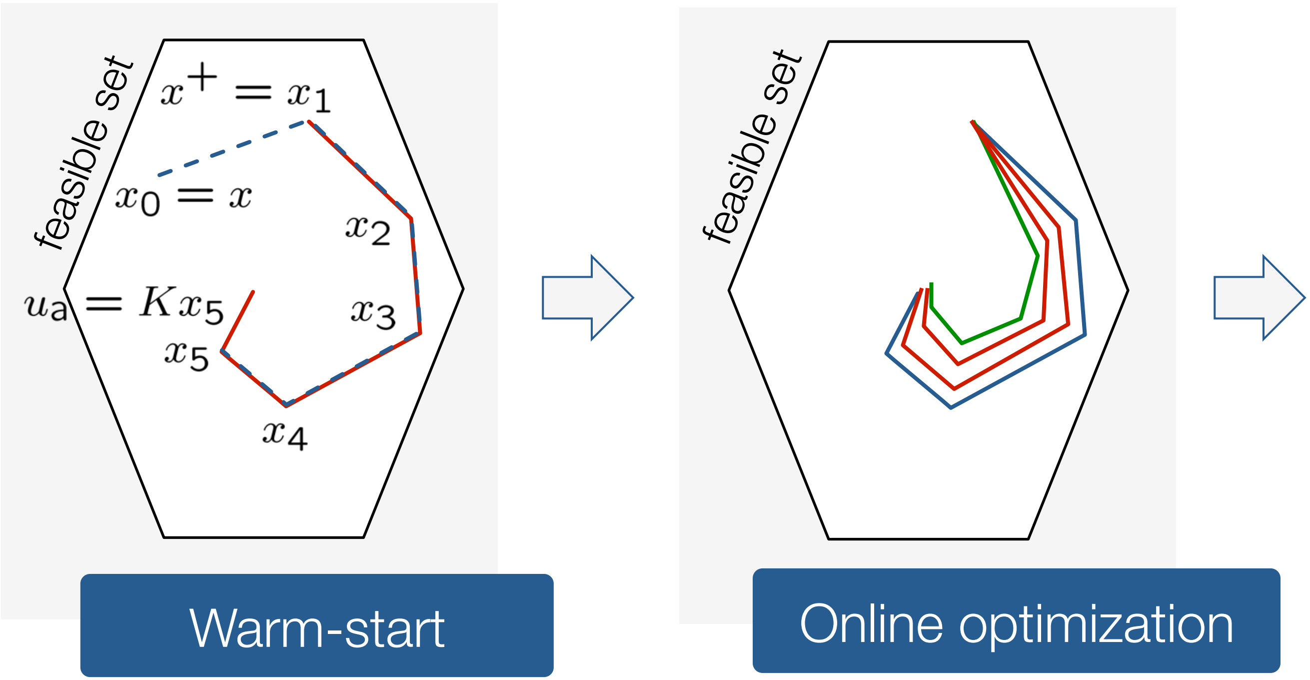
Optimal solution

Common warm-start: Shifted sequence

$$\boldsymbol{u}(x) = [u_0, \ldots, u_{N\text{-}1}]$$

$$\boldsymbol{u}_{shift}(x) = [u_1, \ldots, u_{N\text{-}1}, Kx_N]$$

Optimal MPC scheme

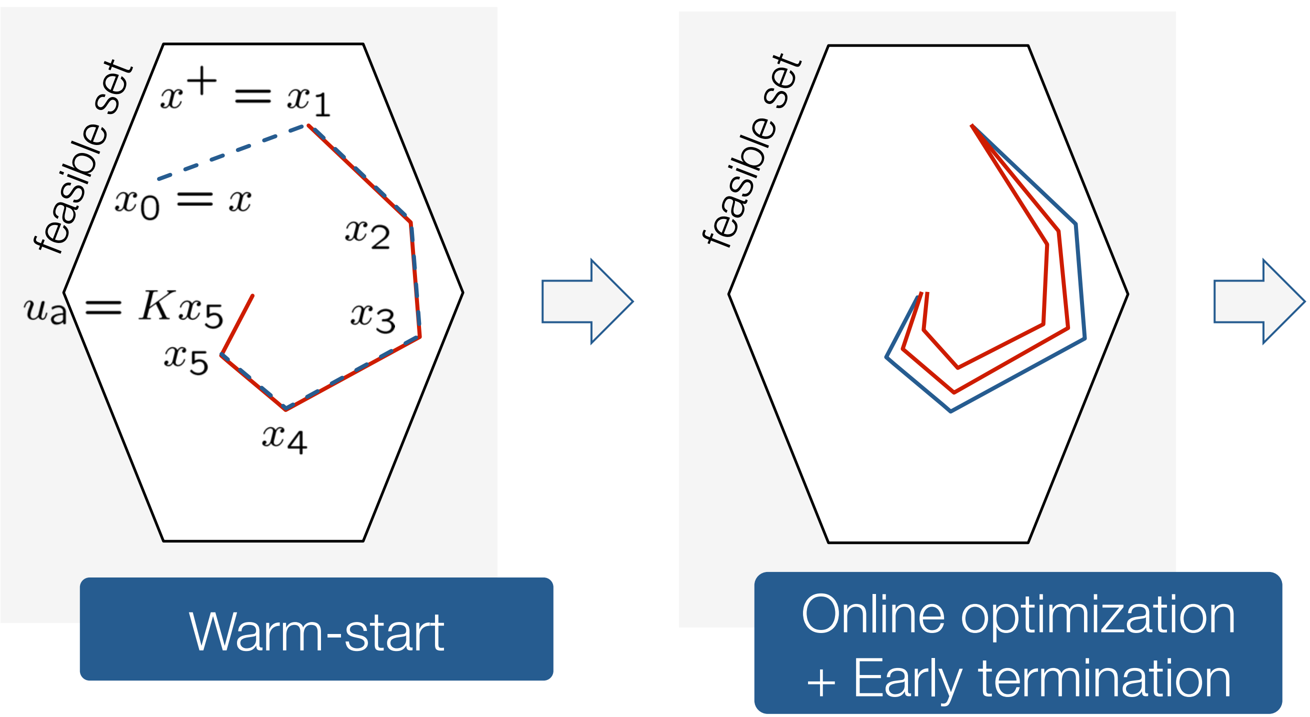


Optimal solution

Optimal MPC:

- Recursively feasible
- Stabilizing
- Unknown computation time…

Real-time MPC scheme – General idea



Suboptimal solution

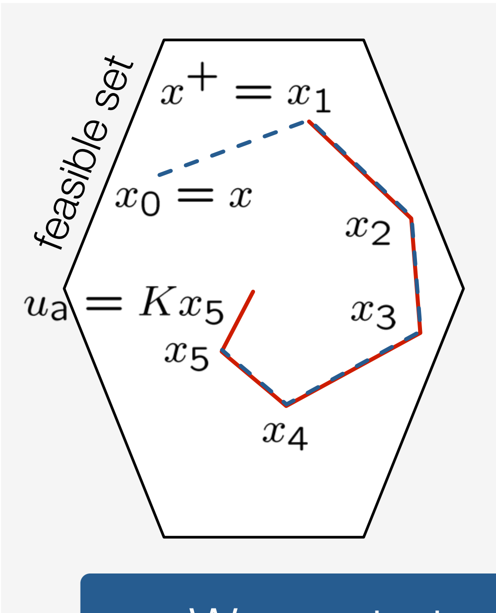
→ **Properties?**

General approach for real-time MPC:

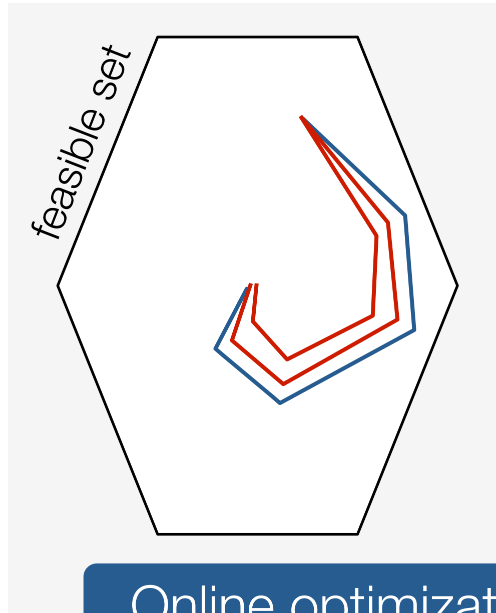
- Use of warm-start method
- Exploitation of structure inherent in MPC problems
- Early termination of the online optimization

*[Wang & Boyd 2008; Ferreau et al., 2008; Schofield, 2008; Cannon et al., 2007; .. **Many** more]*

Real-time MPC scheme - Current methods



Warm-start



Online optimization
+ Early termination

Suboptimal solution

→ **Properties?**

Suboptimal solution during online optimization steps

- can be infeasible
- can destabilize the system
- can cause steady-state offset

Problem definition

MPC problem:

$$
\begin{aligned}
J^*(x) = \min_{x,u} V_N(x, \mathbf{u}) \quad & \triangleq \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i + x_N^T P x_N \\
\text{s.t.} \quad x_{i+1} & = A x_i + B u_i \\
C x_i + D u_i & \leq b \\
x_N & \in \mathcal{X}_f \\
x_0 & = x
\end{aligned}
$$

$\downarrow$

Parametric Quadratic Program

Assumption: $\mathcal{X}_f$ is a polytope, but approach can be equivalently applied to ellipsoidal constraints.

Two QP formulations

Vectorized notation: $\mathbf{x} = [x_0^T, x_1^T, \ldots, x_N^T]^T, \;\; \mathbf{u} = [u_0^T, u_1^T, \ldots, u_{N-1}^T]^T$

Formulation 1:

- The predicted states can be expressed as $\mathbf{x} = \mathcal{A}x + \mathcal{B}\mathbf{u}$
- The MPC problem can be written using only the optimization variable $\mathbf{u}$:

$$\begin{aligned} \min_{\mathrm{u}} \quad & \mathbf{u}^T H_d \mathbf{u} \\ \text{s.t.} \quad & G_d \mathbf{u} \leq f_d + E_d x \end{aligned}$$

Matrices are dense

Formulation 2:

- Optimize over sequence of states and inputs $z = [\mathbf{x}^T, \mathbf{u}^T]^T$:
- Introduce equality constraints relating the states and inputs:

$$\begin{aligned} \min_{z} \quad & z^T H z \\ \text{s.t.} \quad & Gz \leq f \\ & Fz = Ex \end{aligned}$$

Matrices are sparse

Problem definition

MPC problem:

$$J^*(x) = \min_{\mathbf{x},\mathbf{u}} V_N(x, \mathbf{u}) \triangleq \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i + x_N^T P x_N$$

$$\begin{aligned} \text{s.t.} \quad x_{i+1} &= Ax_i + Bu_i \\ Cx_i + Du_i &\leq b \\ x_N &\in \mathcal{X}_f \\ x_0 &= x \end{aligned}$$

Assumption: $\mathcal{X}_f$ is a polytope, but approach can be equivalently applied to ellipsoidal constraints.

↓

Parametric Quadratic Program:

$$\begin{aligned} J^*(x) = \min_z \quad & z^T H z \\ \text{s.t.} \quad & Gz \leq d \\ & Fz = Ex \end{aligned}$$

Current real-time MPC methods
- Loss of feasibility

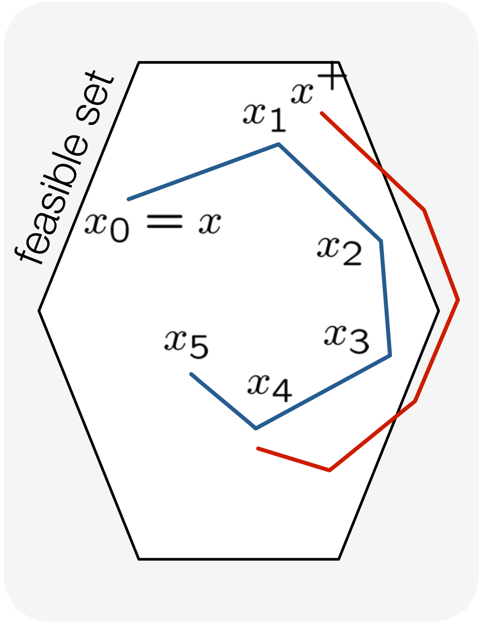
In practice: System will be subject to disturbances

Consider uncertain system: $x^+ = Ax + Bu + w$

where $w \in \mathcal{W}$ is a bounded disturbance.

Problem: Disturbances cause loss of feasibility of the warm-start solution

→ Recovery of feasibility not guaranteed in real-time



Current real-time MPC methods
- Loss of stability

Requirement for stability: Lyapunov function

→ Use of MPC cost as Lyapunov function

→ MPC cost has to decrease at every time step: $V_N(x, \mathbf{u}(x)) < V_N(x_{\text{prev}}, \mathbf{u}(x_{\text{prev}}))$

In a real-time approach this condition can be violated even when initializing with the shifted sequence

Interior-point methods:

- Efficient optimization method for a wide range of optimization problems

Background: Primal barrier interior-point method

Optimization problem:

$$\begin{aligned} \min_z \quad & z^T H z \\ \text{s.t.} \quad & Gz \leq d \\ & Fz = Ex \end{aligned}$$

Note: here QP, but general nonlinear program possible

Background: Primal barrier interior-point method

Optimization problem:

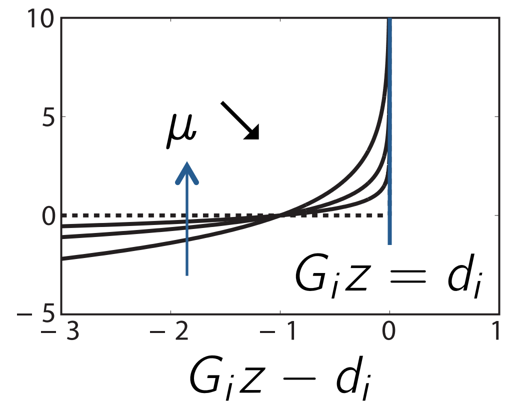
$$\begin{aligned}\min_z \quad & z^T H z \\ \text{s.t.} \quad & Gz \le d \\ & Fz = Ex\end{aligned}$$

Barrier method:

$$\begin{aligned}\min_z \quad & z^T H z - \mu \sum_{i=1}^{m} \log(-G_i z + d_i) \\ \text{s.t.} \quad & Fz = Ex\end{aligned}$$

barrier term with barrier parameter $\mu > 0$

- Equality constrained problem
- Approximation improves as $\mu \to 0$



[Boyd & Vandenberghe, 2004]

Background: Primal barrier interior-point method

Optimization problem:

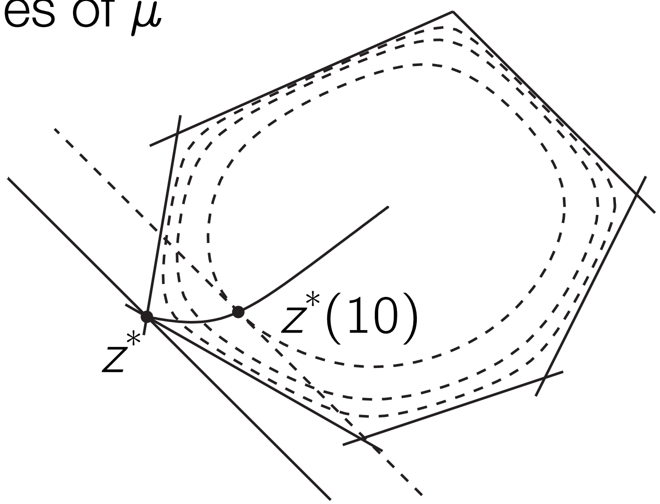
$$\begin{aligned} \min_z \quad & z^T H z \\ \text{s.t.} \quad & Gz \leq d \\ & Fz = Ex \end{aligned}$$

Barrier method:

$$\begin{aligned} \min_z \quad & z^T H z - \mu \sum_{i=1}^{m} \log(-G_i z + d_i) \\ \text{s.t.} \quad & Fz = Ex \end{aligned}$$

barrier term with barrier parameter $\mu > 0$

– Solve augmented problem for decreasing values of μ
  → $z^*(\mu)$ (central path)
– Convergence to the optimal solution of the original optimization problem for $\mu \rightarrow 0$



[Boyd & Vandenberghe, 2004]

Current real-time MPC methods - Loss of stability

Requirement for stability: Lyapunov function

→ Use of MPC cost as Lyapunov function

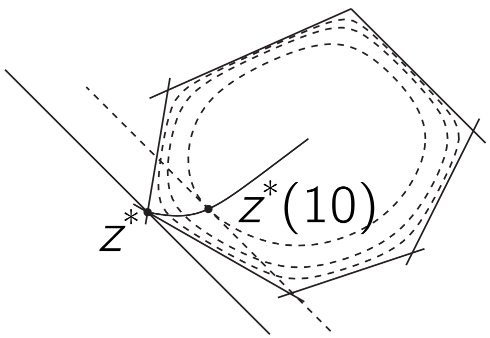
→ MPC cost has to decrease at every time step: $V_N(x, \mathbf{u}) < V_N(x_{\text{prev}}, \mathbf{u}_{\text{prev}})$

In a real-time approach this condition can be violated even when initializing with the shifted sequence

Interior-point methods:

- Efficient optimization method for a wide range of optimization problems
- Minimize augmented cost

$$\begin{aligned} \min_z \quad & z^T H z - \mu \sum_{i=1}^{m} \log(-G_i z + d_i) \\ \text{s.t.} \quad & Fz = Ex \end{aligned}$$



→ Decrease in cost does not enforce a decrease in MPC cost $z^T H z$

→ Steady-state offset for $\mu \neq 0$

Proposed real-time MPC method

Real-time online MPC:

Guarantee that

- within the real-time constraint
- a feasible solution
- satisfying stability criteria
- for any admissible initial state

is found.

Proposed real-time MPC method

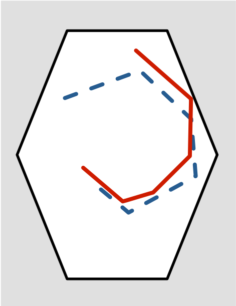
Real-time online MPC:

Guarantee that

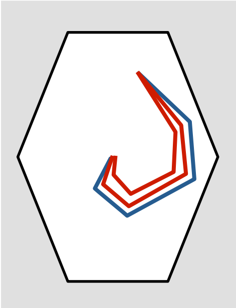
- within the real-time constraint $\Leftarrow$ Early termination
- a feasible solution $\Leftarrow$ Robust MPC
- satisfying stability criteria
- for any admissible initial state

is found.

- Robust MPC method provides feasibility of the warm-start solution by considering all possible disturbance sequences



- Use of primal feasible optimization method provides feasibility of the suboptimal solution obtained during online optimization



Proposed real-time MPC method

Real-time online MPC:

Guarantee that

- within the real-time constraint ⇐ Early termination
- a feasible solution ⇐ Robust MPC
- satisfying stability criteria ⇐ Lyapunov constraint
- for any admissible initial state

is found.

Introduce 'Lyapunov constraint':

Enforces decrease in suboptimal MPC cost *at each iteration*

$V_N(x, \mathbf{u}) < V_N(x_{\text{prev}}, \mathbf{u}_{\text{prev}})$ or $z^T H z \leq z_{\text{prev}}^T H z_{\text{prev}}$ → Quadratic constraint

→ (Input-to-state) Stability for *any* real-time constraint

→ Convergence to desired steady state

Extension to reference tracking: Extend tracking approach *in [Limon et al., 2008]*

Real-time robust MPC - Fast implementation
Interior point optimization

- Standard Newton step computation:

$$\begin{bmatrix} \nabla^2_{zz}\mathcal{L} + \mu G^T S^{-2} G & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \triangle z \\ \triangle \nu \end{bmatrix} = - \begin{bmatrix} r_d \\ r_p \end{bmatrix}$$

Speed of optimization
$\propto$
Time to solve linear system

- Tracking formulation and Lyapunov constraint
  → Modified Newton step matrix structure

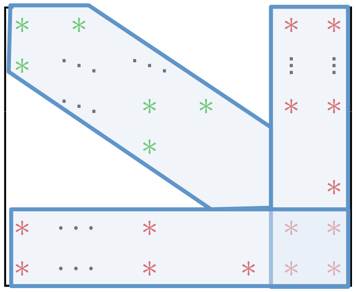
Real-time robust MPC - Fast implementation
Interior point optimization

- Standard Newton step computation:

$$\begin{bmatrix} \nabla^2_{zz}\mathcal{L} + \mu G^T S^{-2} G & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \triangle z \\ \triangle \nu \end{bmatrix} = - \begin{bmatrix} r_d \\ r_p \end{bmatrix}$$

Speed of optimization
$\propto$
Time to solve linear system

- Tracking formulation and Lyapunov constraint
  → Modified Newton step matrix structure
- Matrices can be transformed into arrow structure
  → Solved as efficiently as standard MPC problems
- Custom solver in C++ was developed
  → Extending *[Rao et al., 1998, Hansson, 2000 and Wang et al., 2008]*



Real-time robust MPC - Fast implementation
Interior point optimization

- Standard Newton step computation:

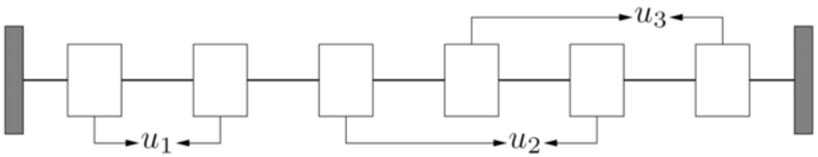
$$\begin{bmatrix} \nabla^2_{zz}\mathcal{L} + \mu G^T S^{-2} G & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \triangle z \\ \triangle \nu \end{bmatrix} = - \begin{bmatrix} r_d \\ r_p \end{bmatrix}$$

Speed of optimization
$\propto$
Time to solve linear system

- Tracking formulation and Lyapunov constraint
  → Modified Newton step matrix structure
- Matrices can be transformed into arrow structure
  → Solved as efficiently as standard MPC problems
- Custom solver in C++ was developed
  → Extending *[Rao et al., 1998, Hansson, 2000 and Wang et al., 2008]*

→ Fast solution of tracking problem
→ Guaranteed stability for time constraints!
→ Computation times faster than methods with no guarantees

Numerical examples



Oscillating masses example

- Problem: 12 states, 3 inputs
- Fast MPC with guarantees: horizon N=10

  → Computation of 5 Newton steps in **2 msec**

Comparison: CPLEX **26 msec**, SEDUMI **252 msec**

Closed loop performance loss in % for varying iteration numbers

$k_{\max}$	1	2	3	4	5	6	7	8
$\triangle J_{cl}$	1.39	1.32	1.10	0.88	0.70	0.55	0.44	0.33

→Optimal ~44 iterations

→ 2.5kHz sampling rate with stability guarantee

Random example

- Problem: 30 states, 8 inputs, horizon N=10

→ QCQP with 410 optimization variables and 1002 constraints

→ Computation of 5 Newton steps in **10 msec**

Real-time online MPC: Goals

Real-time online MPC:

Guarantee that
- within the real-time constraint
- a feasible solution
- satisfying stability and performance criteria
- for any admissible initial state

is found.

We present two methods for linear systems:

Target	Linear state and input constraints	'Simple' input constraints (e.g. box constraints)
Time scale	Milliseconds	Microseconds
Idea	Provide guarantees for any time constraint	Compute a priori bounds on the required online computation time
Approach	**Robust MPC with stability constraints**	**Fast gradient method**
	[M.N. Zeilinger et al., CDC 2009]	*[S. Richter et al., CDC 2009]*

Structured Optimization: Gradient Method for input constrained MPC

- Fast gradient method
  - Very simple
  - Easy to parallelize
  - Fast for large number of states (using dense problem formulation)

Require: $U_0 \in \mathbb{U}^N,\ V_0 = U_0$

for $i = 1$ to i_{min} **do**

$\quad U_i = \pi_{\mathbb{U}^N}\left(V_{i-1} - \frac{1}{L}\nabla J_N(V_{i-1}; x)\right)$

$\quad V_i = U_i + \beta_i(U_i - U_{i-1})$

end for

Work per iteration

- 1 matrix-vector product
- 2 vector sums
- 1 projection

Key result: Can compute *a priori* bound on required number of iterations i_{min}

[Y. Nesterov, 1983]
[S. Richter et al., CDC 2009]

Fast Gradient Method : Time bound to ε-optimality

- Solution with approximation error ε in $i_{\min}$ steps:

$$i_{\min} \geq \left\lceil \frac{\ln \frac{\epsilon}{\delta}}{\ln\left(1 - \sqrt{\frac{1}{\kappa}}\right)} \right\rceil$$

- κ condition number
- δ measure of *initial residual*

Cold start:

$$\delta \leq LR^2/2$$

- $\boldsymbol{u}_{ws} = 0$
- R : radius of feasible set
- Easy to compute

Warm start:

$$\delta \leq 2 \max_{x \in \mathbb{X}_N} J_N(\mathbf{u}_{\text{ws}}; x) - J_N^*(x)$$

- $\boldsymbol{u}_{ws}$: Warm start sequence
- Worst distance measured in terms of initial cost
- Hard to compute

[S. Richter et al., CDC 2009]

Fast Gradient Method : Time bound to ε-optimality

- Solution with approximation error ε in $\boldsymbol{i}_{\min}$ steps:

$$i_{\min} \geq \left\lceil \frac{\ln \frac{\epsilon}{\delta}}{\ln\left(1 - \sqrt{\frac{1}{\kappa}}\right)} \right\rceil$$

- κ condition number
- δ measure of *initial residual*

NOTE: Extension to state and input constraints possible using Lagrangian relaxation
[S. Richter et al., CDC 2011]

Cold start:

$$\delta \leq LR^2/2$$

- $\boldsymbol{u}_{ws} = 0$
- R : radius of feasible set
- Easy to compute

Warm start:

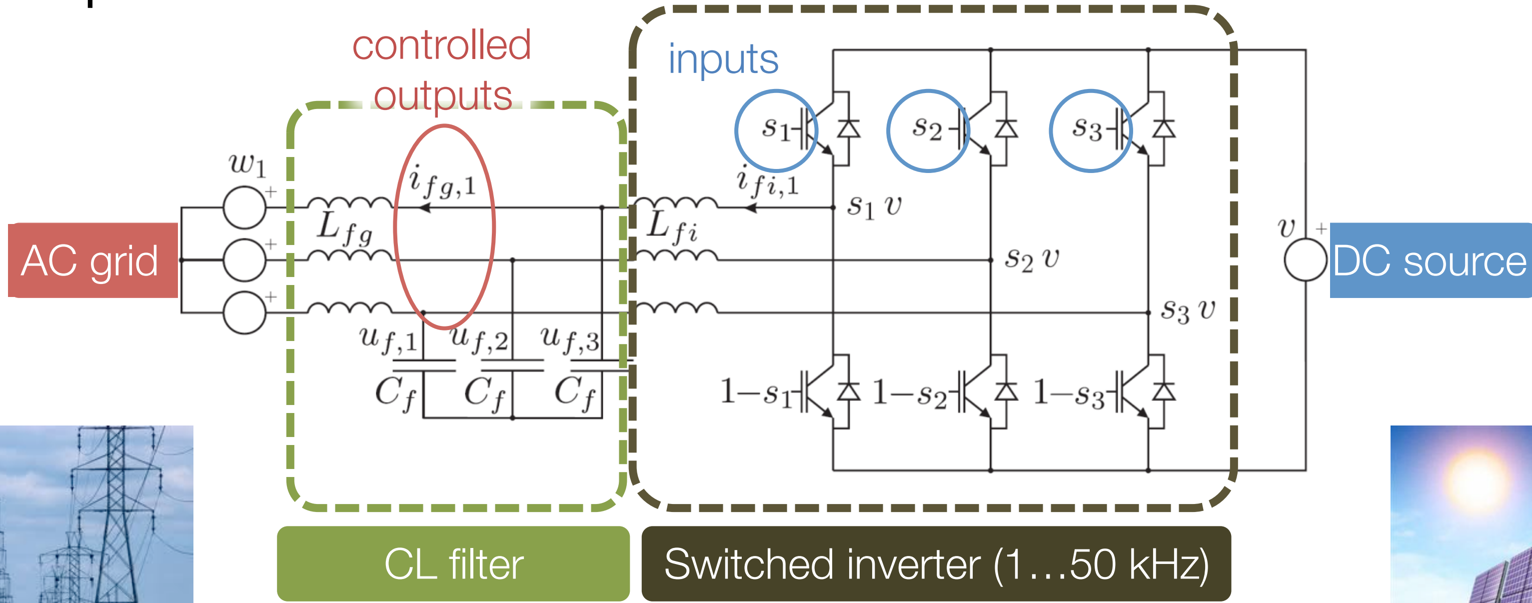
$$\delta \leq 2 \max_{x \in \mathbb{X}_N} J_N(\mathbf{u}_{\text{ws}}; x) - J_N^*(x)$$

- $\boldsymbol{u}_{ws}$: Warm start sequence
- Worst distance measured in terms of initial cost
- Hard to compute

[S. Richter et al., CDC 2009]

Example: Control of an AC-DC Power Converter

2-level 3-phase inverter:



Control objectives:

- Track currents $i_{fg,1}$, $i_{fg,2}$, $i_{fg,3}$
- Actively dampen CL filter dynamics

Model: Marginally stable system in d-q coordinates:

6 states / 2 inputs / 2 disturbances / 2 controlled outputs

[S. Richter et al., ACC 2010]

Example: Control of an AC-DC Power Converter

MPC Tracking Problem:

$$J_N^*(q) = \min \|\delta x_N\|_P^2 + \sum_{i=0}^{N-1} + \|\delta x_i\|_Q^2 + \|\delta u_i\|_R^2$$

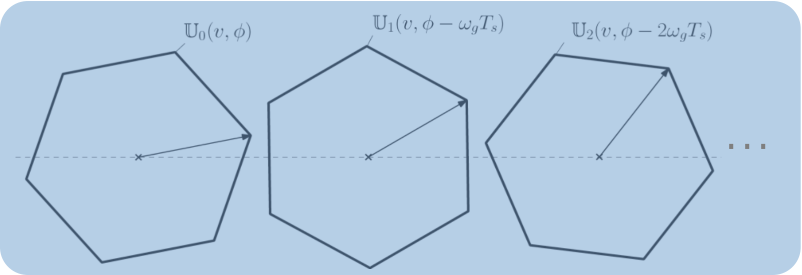
$$\text{s.t. } \delta x_i = x_i - x_{ss}$$

$$\delta u_i = u_i - u_{ss}$$

$$x_{i+1} = Ax_i + Bu_i + B_w w$$

$$u_i \in \mathbb{U}(v, \phi - i\omega_g T_s)$$

Rotating/Scaling Feasible Set:



Implementation environment:

- 16-bit native fixed-point DSP BF-533 from Analog Devices (≈10$)
- C code (integer arithmetic) + standard C-compiler

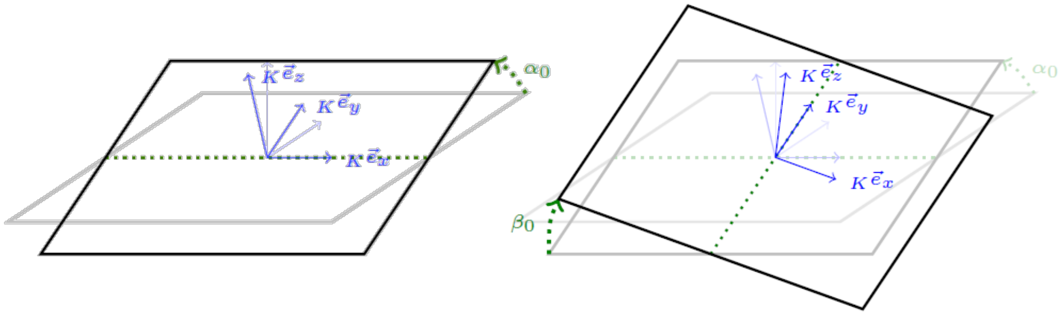
Main results:

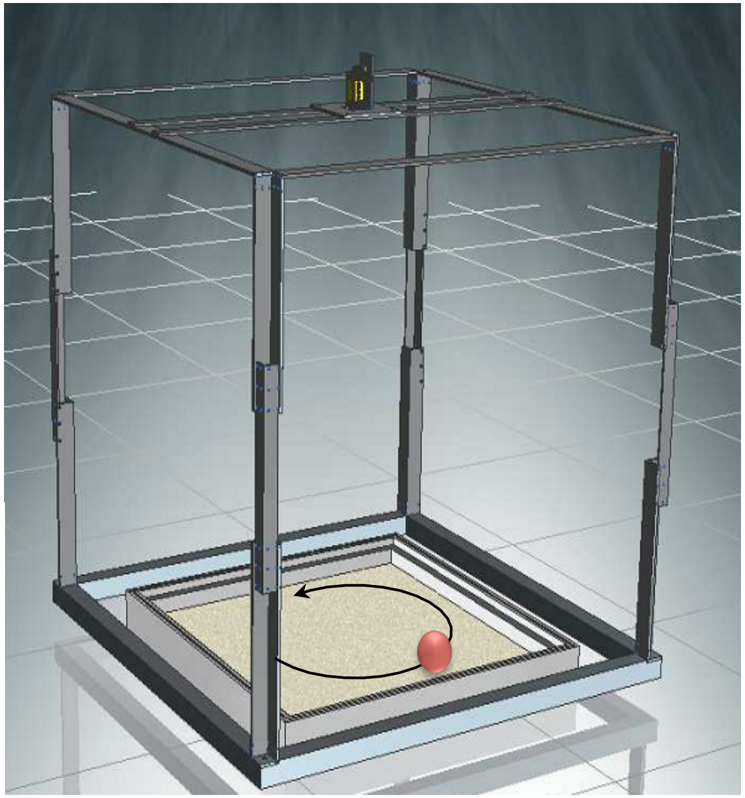
Bound:	*125 µs*
Solution Time:	*< 50 µs*
Memory:	*< 1kB*
Relative accuracy:	*< 1e-3*

[S. Richter et al., ACC 2010]

Example: Ball on Plate System

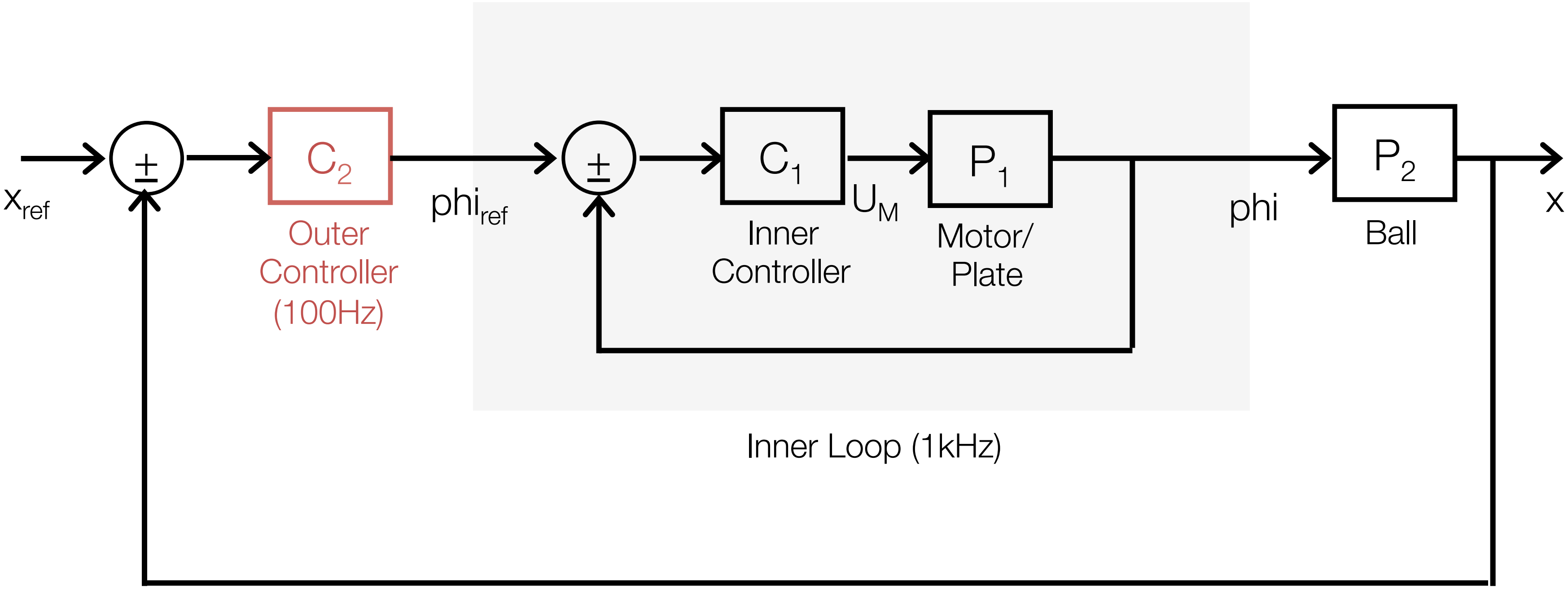
- Movable plate (0.66m x 0.66m)
- Can be revolved around two axis [+17° ; -17°] by two DC motors
- Angle is measured by potentiometers
- Linearized dynamics: 4 states, 2 inputs
- Position of the ball is measured by a camera





[Master thesis by R. Waldvogel, 2011]

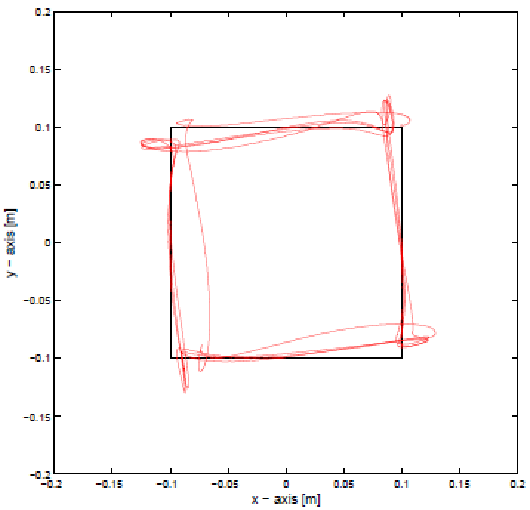
Example: Ball on Plate System
Cascaded Control Structure



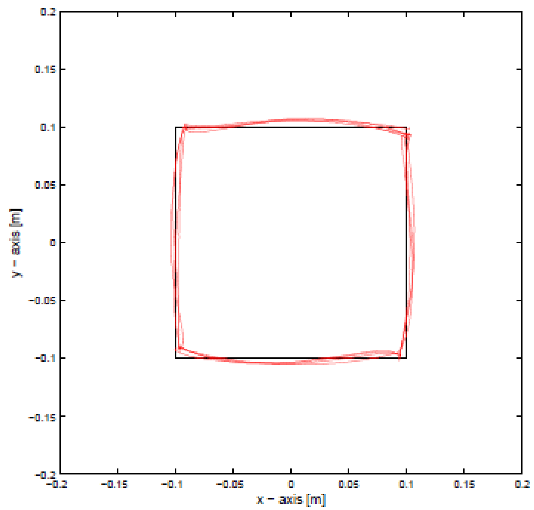
[Master thesis by R. Waldvogel, 2011]

Example: Ball on Plate System
Controller comparison

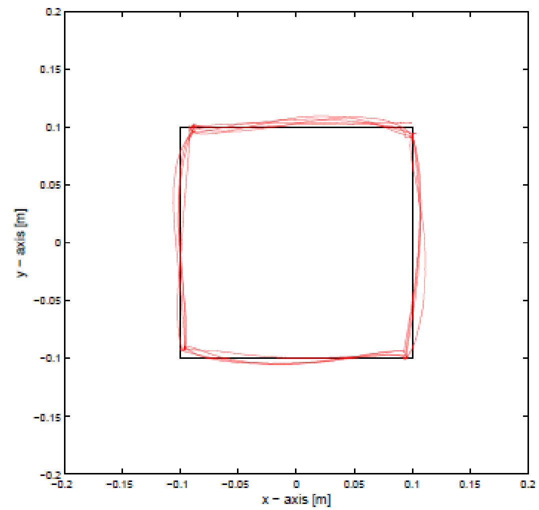
- Ball Control: PID vs. LQR vs. MPC Controller



(a) PID - Controller



(b) LQR - Controller

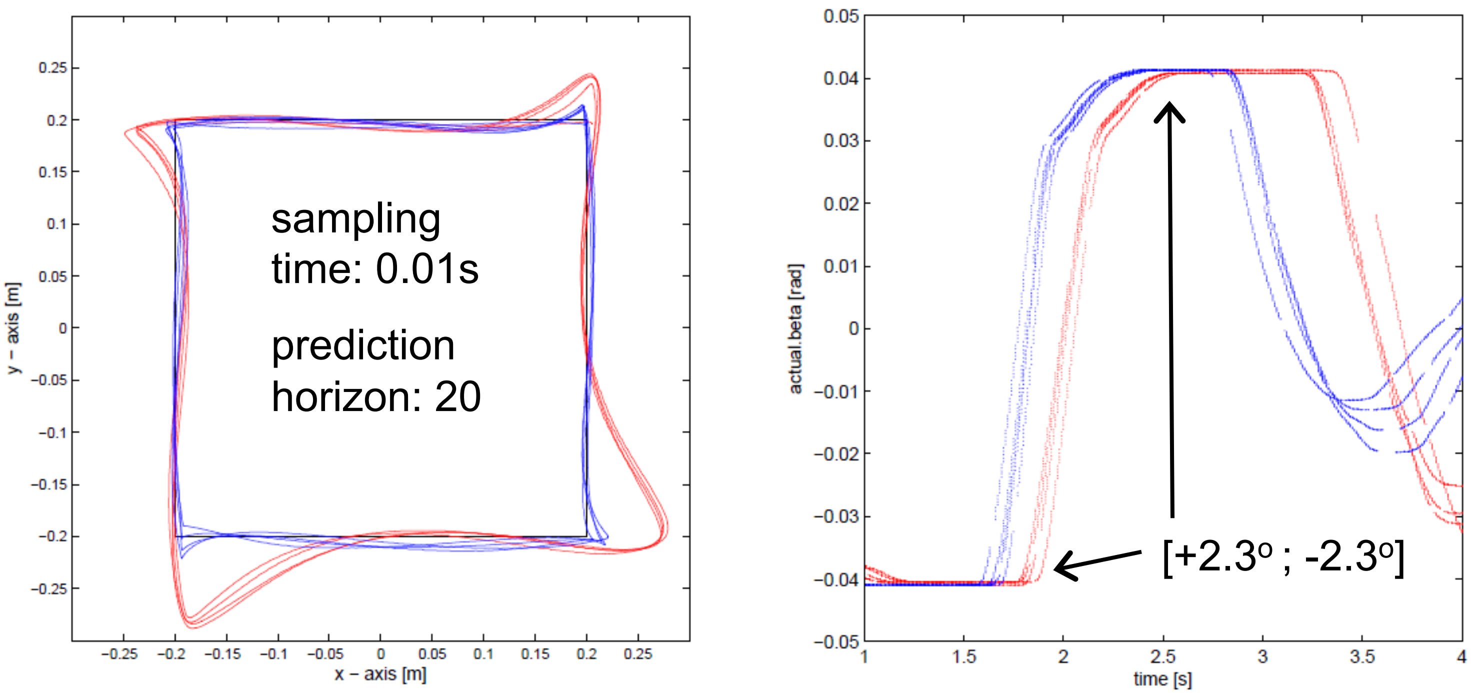


(c) MPC - Controller

[Master thesis by R. Waldvogel, 2011]

Example: Ball on Plate System
Controller comparison

- Ball Control: LQR vs. MPC control with Input Constraints



(a) LQR (red) vs MPC Controller (blue)   (b) Input β for the upper left corner

[Master thesis by R. Waldvogel, 2011]

Example: Ball on Plate System
Video

[Master thesis by R. Waldvogel, 2011]

Fast Gradient Toolbox

FiOrdOs

First Order Optimisation

Fabian Ullmann, Stefan Richter and Colin Jones

- Matlab Toolbox for Real-time First Order Optimization
  - C-Code Generation
  - Real-time code generation for embedded platforms
  - First release autumn'11

Real-time online MPC: Goals

Real-time online MPC:

Guarantee that
- within the real-time constraint
- a feasible solution
- satisfying stability and performance criteria
- for any admissible initial state

is found.

NOTE: Optimality not required

We present two methods for linear systems:

Setting	Linear state and input constraints	'Simple' input constraints (e.g. box constraints)
Time scale	Milliseconds	Microseconds
Idea	Provide guarantees for any time constraint	Compute a priori bounds on the required online computation time
Approach	**Robust MPC with stability constraints**	**Fast gradient method**
	[M.N. Zeilinger et al., CDC 2009]	*[S. Richter et al., CDC 2009]*