Convex and Non-Convex Embedded Optimization Algorithms and Applications

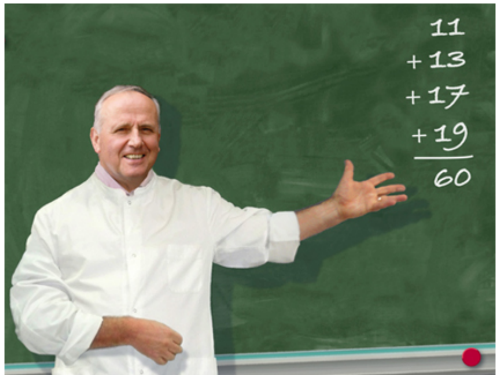
Moritz Diehl

Optimization in Engineering Center OPTEC

& Electrical Engineering Department ESAT-SCD

K.U. Leuven

Belgium



Colloque JBHU 2010, Oct 25-27, Bayonne,

en l'honneur de Jean-Baptiste Hiriart-Urruty

Overview

- Idea of Embedded Optimization
- Perception-based Clipping of Audio Signals using Convex Optimization
- Control of Tethered Airplanes by Auto-Generated Real-Time Iterations

Embedded Optimization = CPU Intensive, Nonlinear Map

x

Embedded Parametric Optimization

$$u = \arg\min_{u} \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} g \\ h \end{bmatrix} \text{ s.t. } Au + Bx \leq b$$

u

Very powerful concept!

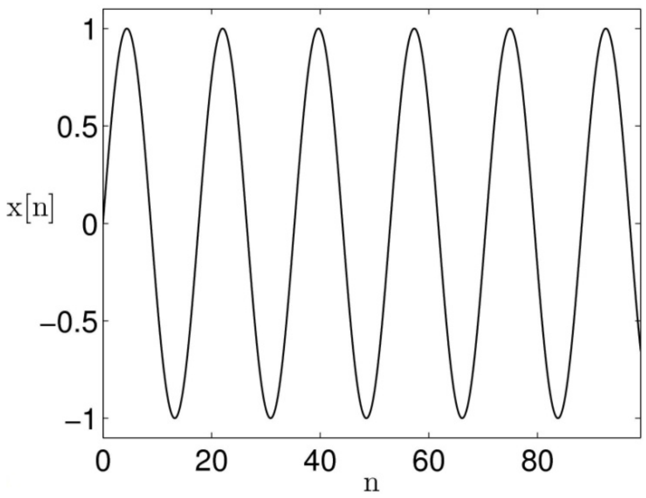
We can prove [Baes, D., Necoara 2008]:
“**Every continuous map** can be generated as solution map of a parametric convex program”

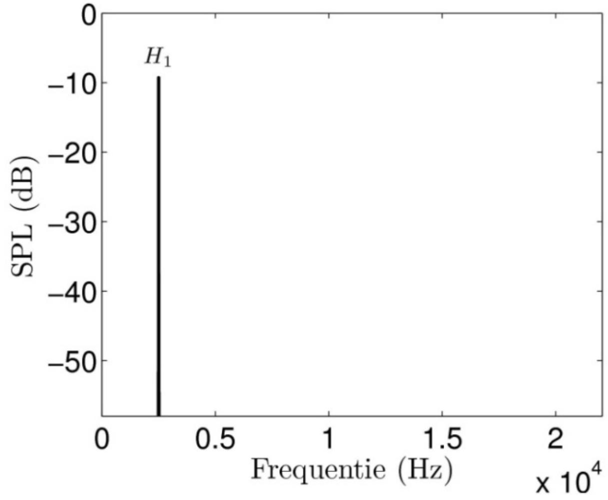
Real-time perception-based clipping of audio signals using convex optimization

Bruno Defraene, Toon van Waterschoot, Hans Joachim Ferreau,
Marc Moonen & M.D.

Clipping Problem Statement

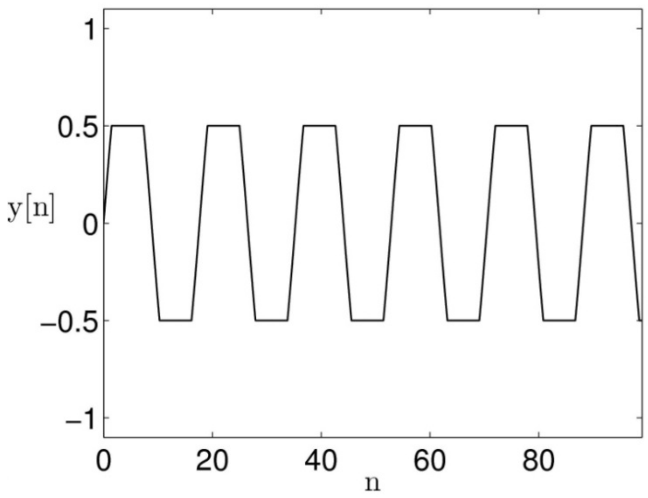
- Clipping = limit amplitude of digital audio signal to range [L,U]
- Real time audio applications (mobile phones, hearing aids…)
- Hard clipping has a large negative effect on perceptual sound quality (distortion)



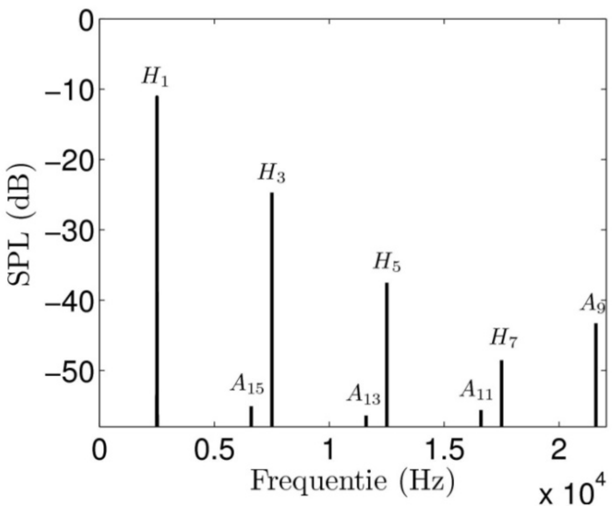


Clipping Problem Statement

- Clipping = limit amplitude of digital audio signal to range [L,U]
- Real time audio applications (mobile phones, hearing aids…)
- Hard clipping has a large negative effect on perceptual sound quality (distortion)

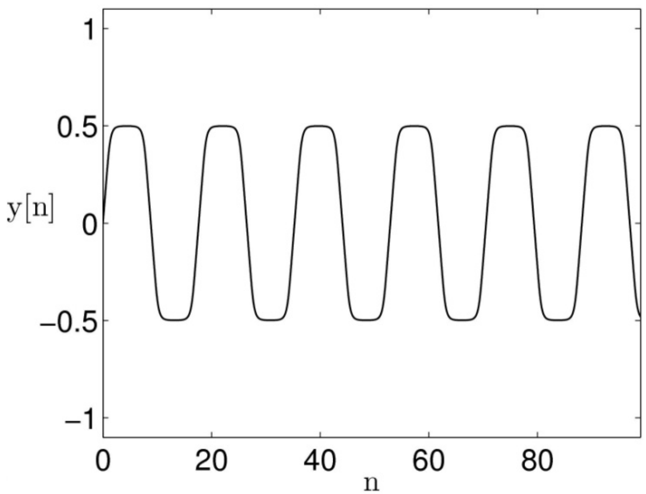


HARD CLIPPING

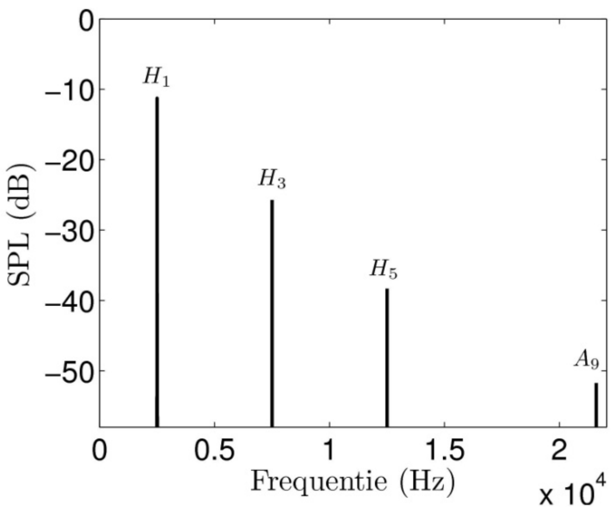


Clipping Problem Statement

- Clipping = limit amplitude of digital audio signal to range [L,U]
- Real time audio applications (mobile phones, hearing aids…)
- Hard clipping has a large negative effect on perceptual sound quality (distortion)
- Soft clipping does not help much

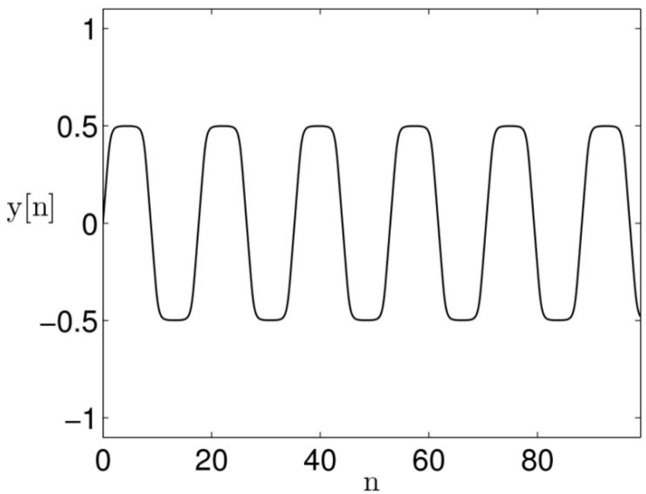


SOFT CLIPPING

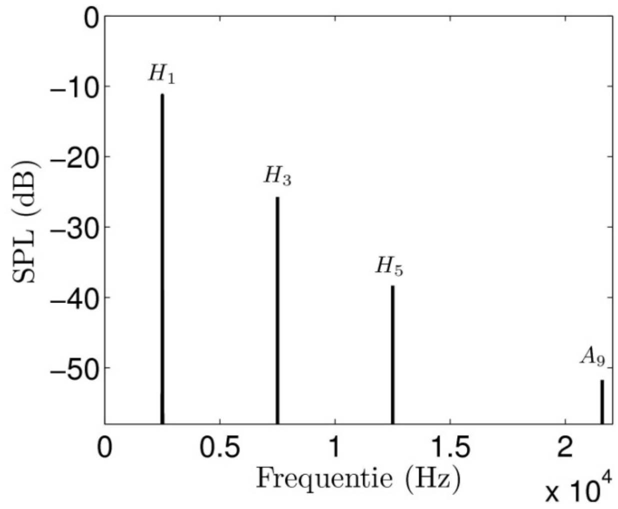


Clipping Problem Statement

- Clipping = limit amplitude of digital audio signal to range [L,U]
- Real time audio applications (mobile phones, hearing aids…)
- Hard clipping has a large negative effect on perceptual sound quality (distortion)
- Soft clipping does not help much



SOFT CLIPPING



What is the optimal way of clipping?

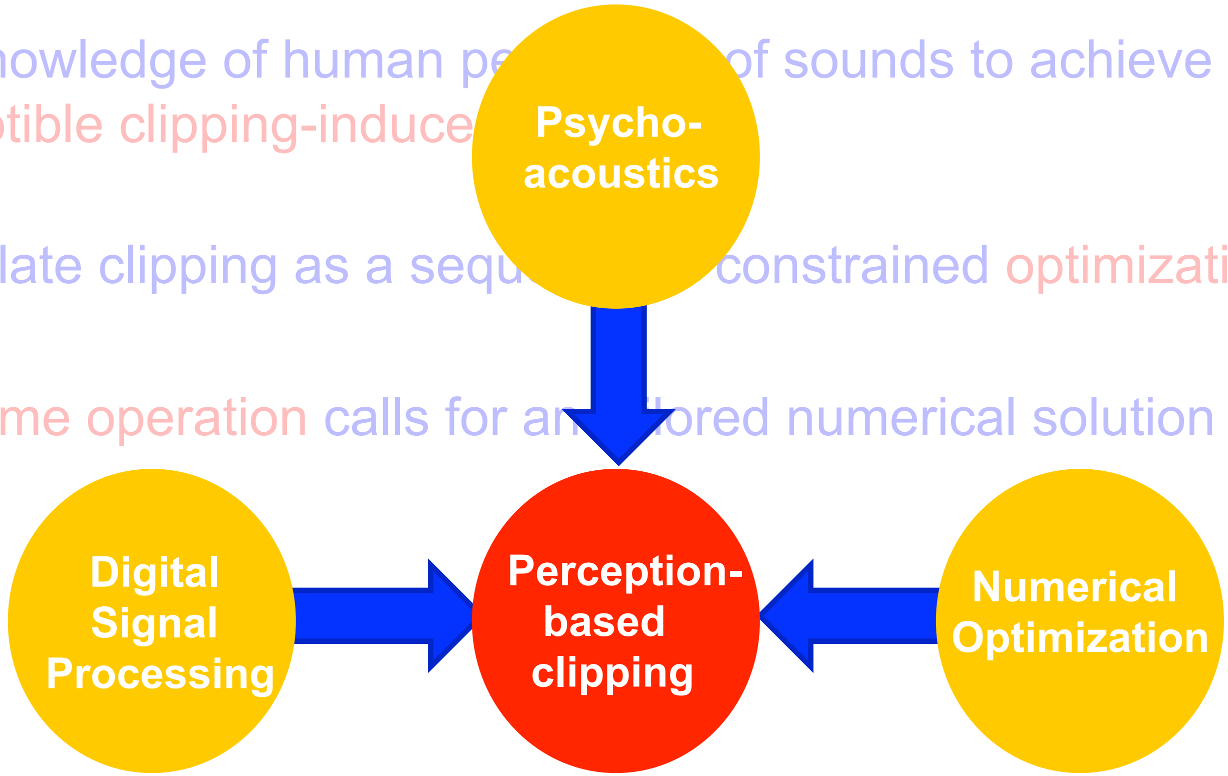
Perception-based clipping - a novel approach

- Flexible: adapt to instantaneous properties of the input signal
- Use knowledge of human perception of sounds to achieve minimal perceptible clipping-induced distortion
- Formulate clipping as a sequence of constrained optimization problems
- Real-time operation calls for an tailored numerical solution strategy

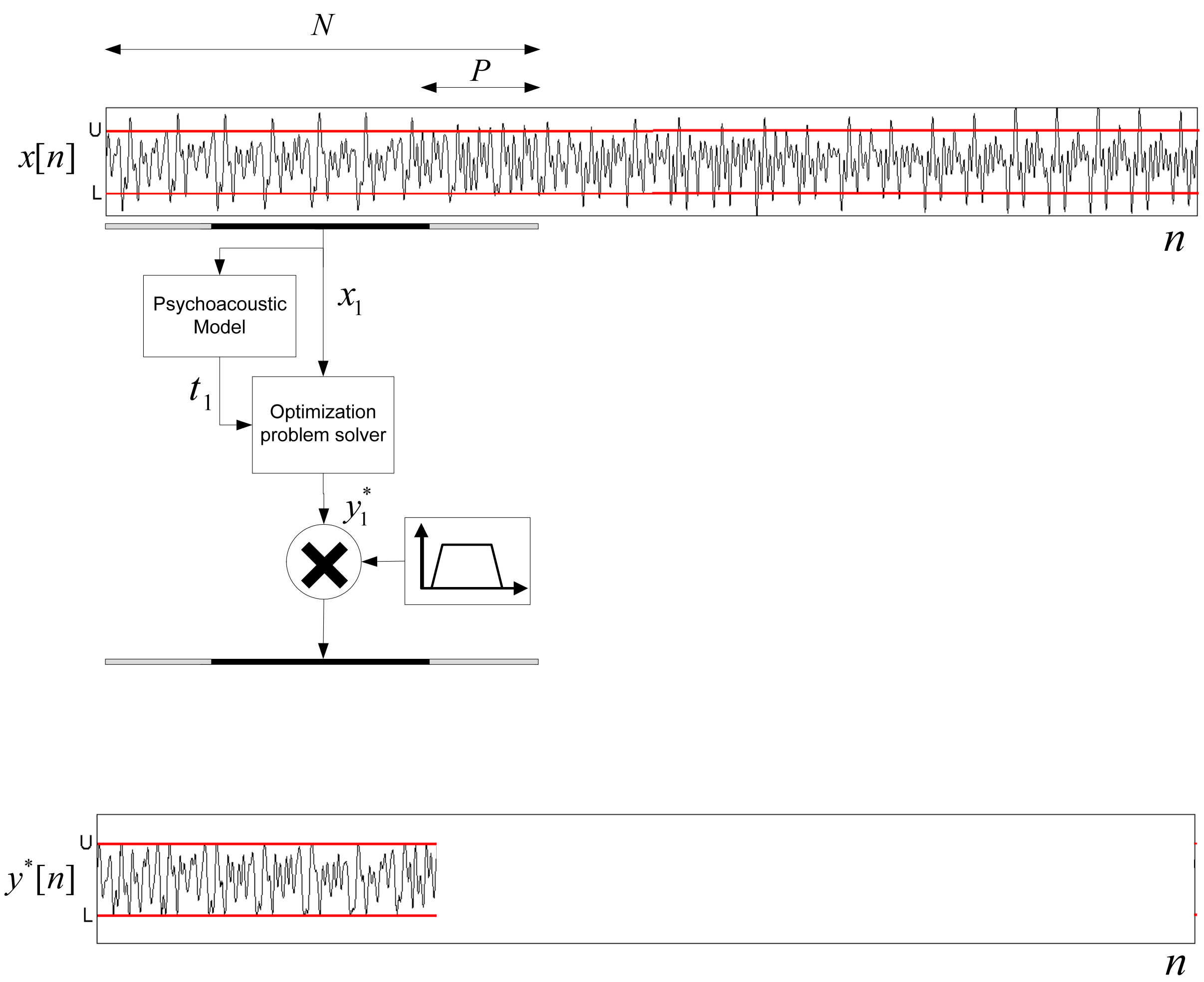
Perception-based clipping - a novel approach

Multidisciplinary approach

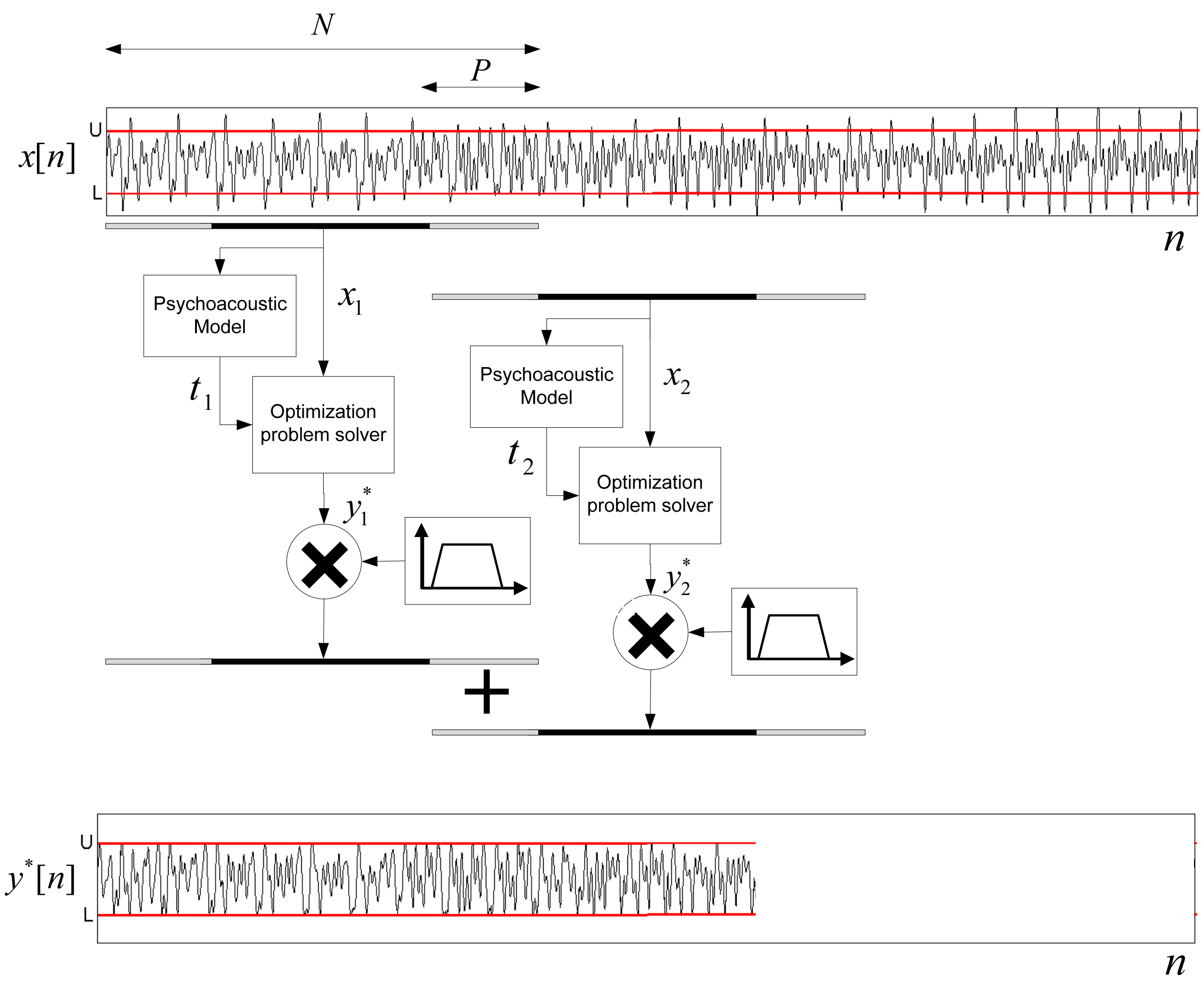
- Flexible: adapt to instantaneous properties of the input signal
- Use knowledge of human pe of sounds to achieve minimal perceptible clipping-induce
- Formulate clipping as a sequ constrained optimization problems
- Real-time operation calls for an lored numerical solution strategy



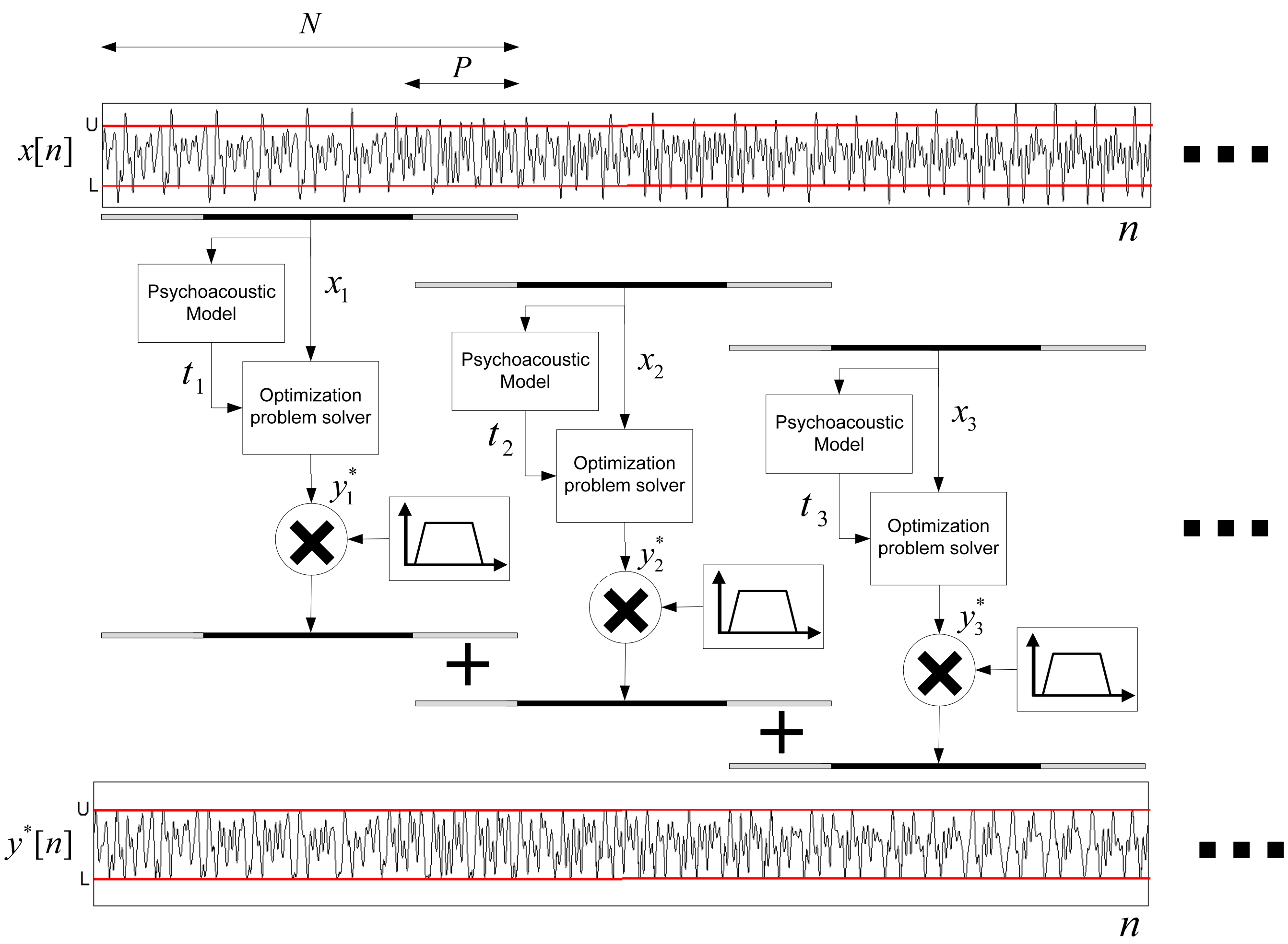
Perception-based clipping algorithm



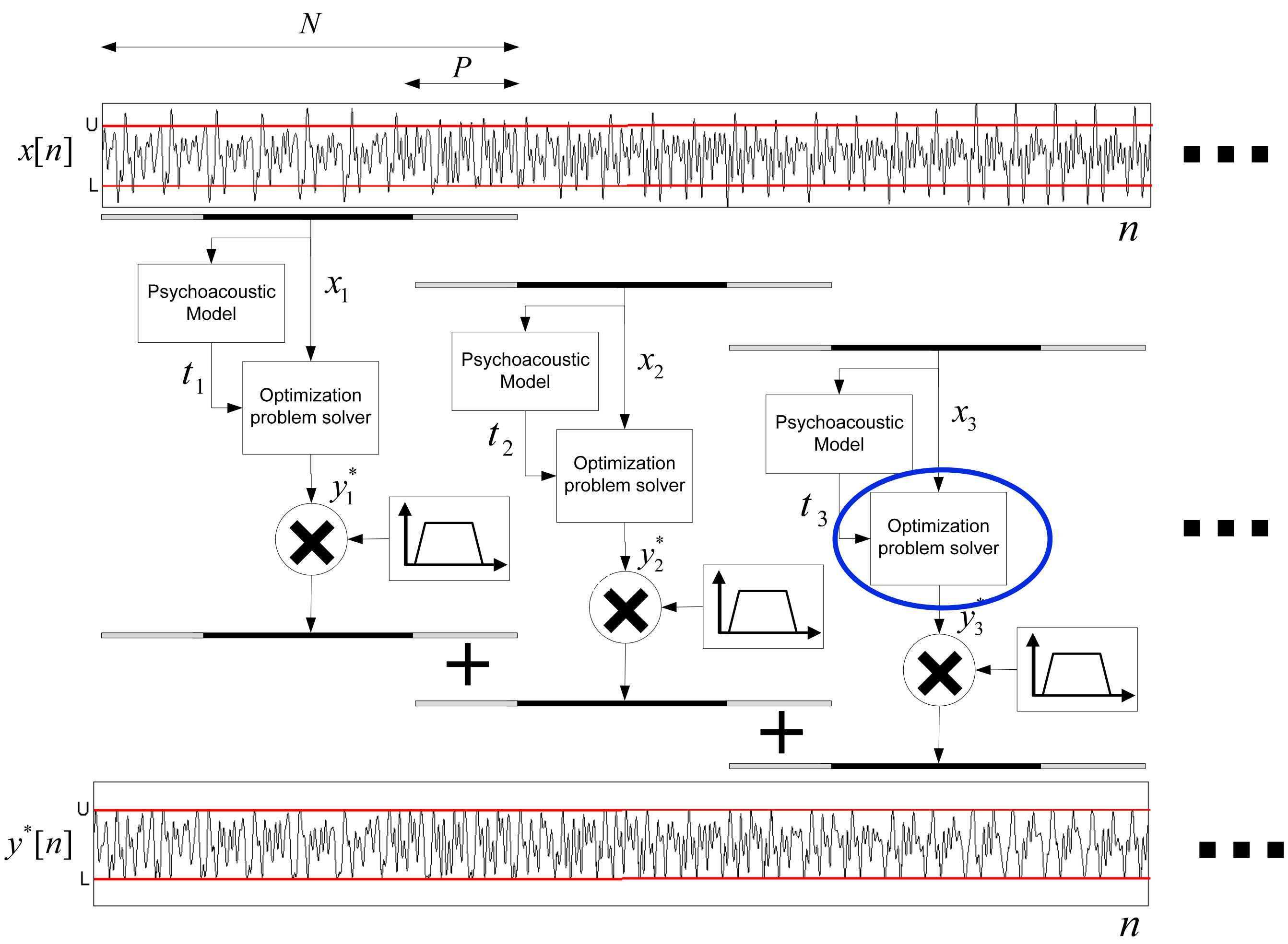
Perception-based clipping algorithm



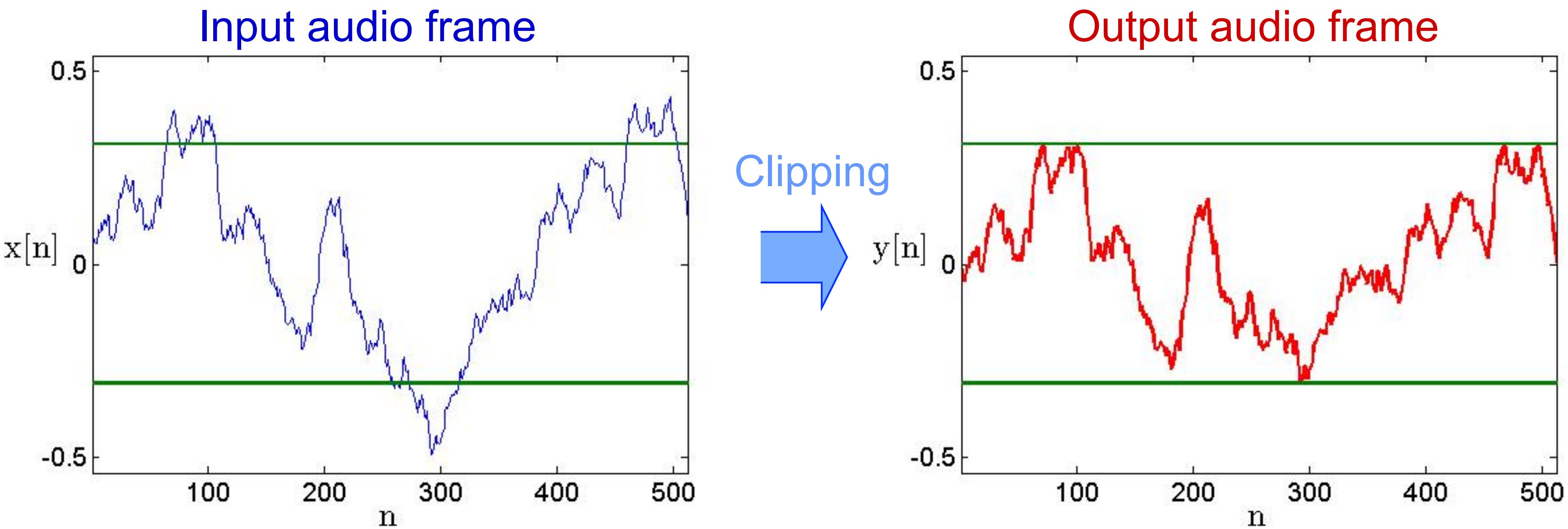
Perception-based clipping algorithm



Perception-based clipping algorithm



Embedded optimization problems = QPs



Minimize perceptual difference in frequency space subject to clipping constraint:

$$\boxed{\min_{y \in \mathbb{R}^N} \frac{1}{2}(y-x)^T D^H W D(y-x) \quad \text{s.t.} \quad L \leq y \leq U}$$

dense **Fourier Matrix** D and diagonal weighting matrix $W = \begin{bmatrix} w_1 & & \\ & w_2 & \\ & & \ddots \end{bmatrix}$

w_i = inverse of *perceptual masking threshold* of current audio frame (not today's topic)

How to solve the QPs fast enough ?

$$\min_{y \in \mathbb{R}^N} \frac{1}{2}(y-x)^T D^H W D(y-x) \quad \text{s.t.} \quad L \leq y \leq U$$

- QP solution time using a general purpose QP solver : +/- 500 ms

[Intel CPU 2.8 GHz]

$\Updownarrow$

- Real-time objective for N = 512 sample frame in CD-Quality: 8.7 ms

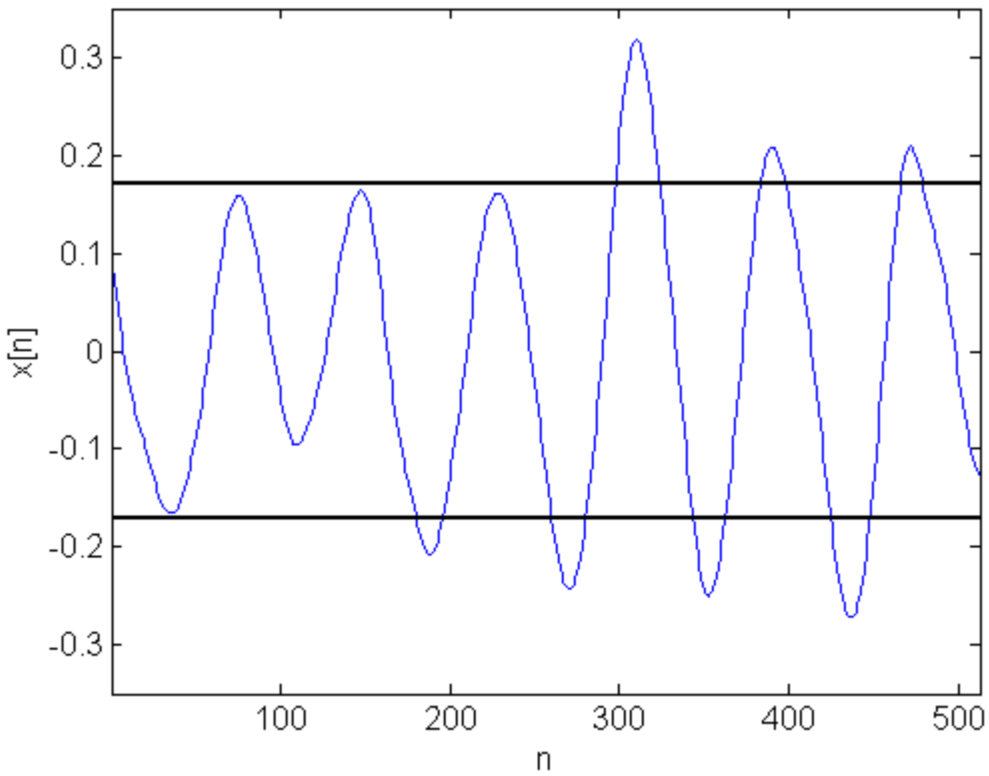
Three Tailored QP Solution Methods

- Method 1: External Active Set Strategy with Small Dual QPs
- Method 2: Projected Gradient
- Method 3: Nesterov's Optimal Gradient Scheme

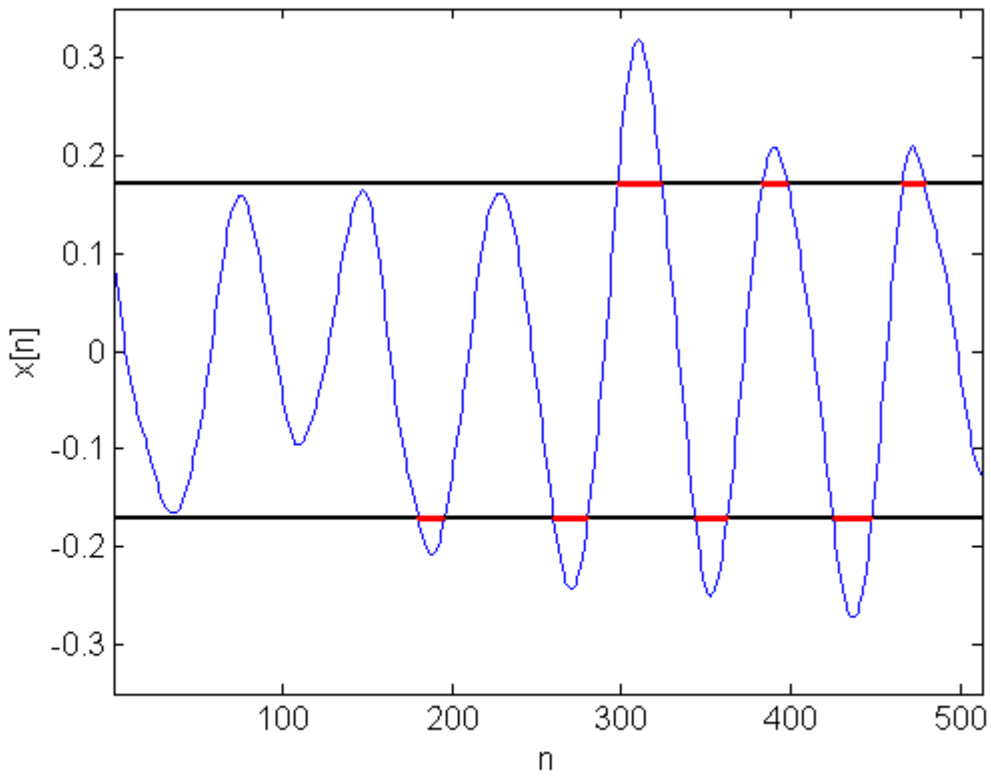
Three Tailored QP Solution Methods

- **Method 1: External Active Set Strategy with Small Dual QPs**
- Method 2: Projected Gradient
- Method 3: Nesterov’s Optimal Gradient Scheme

External Active Set Strategy: Original Signal

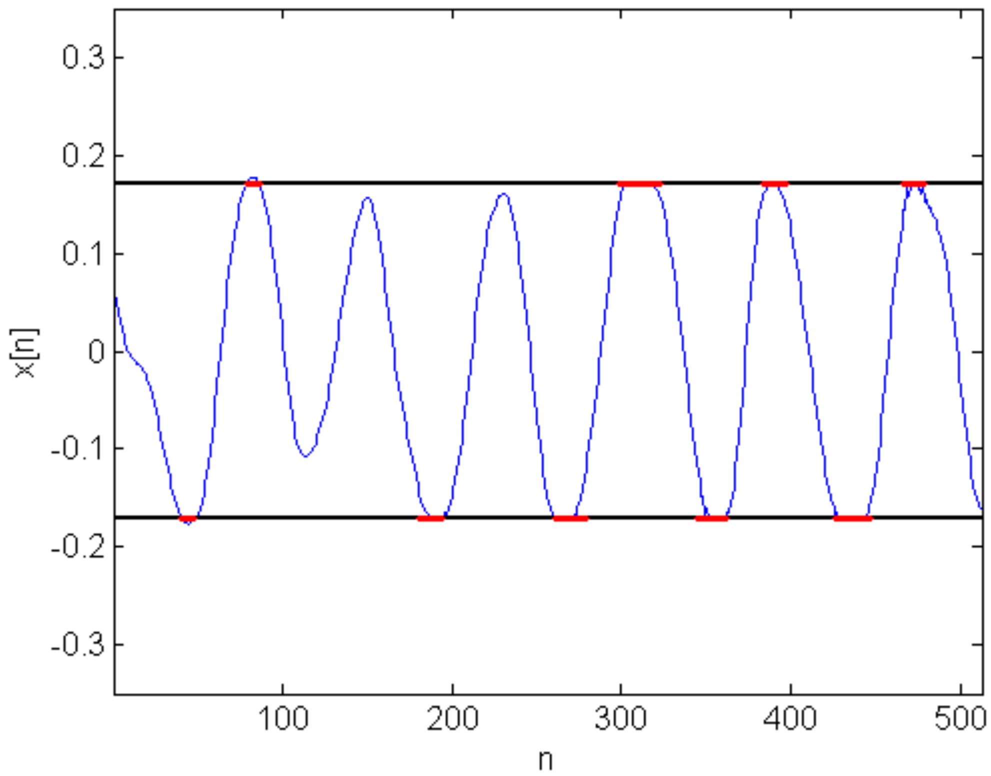


External Active Set Strategy: Original Signal

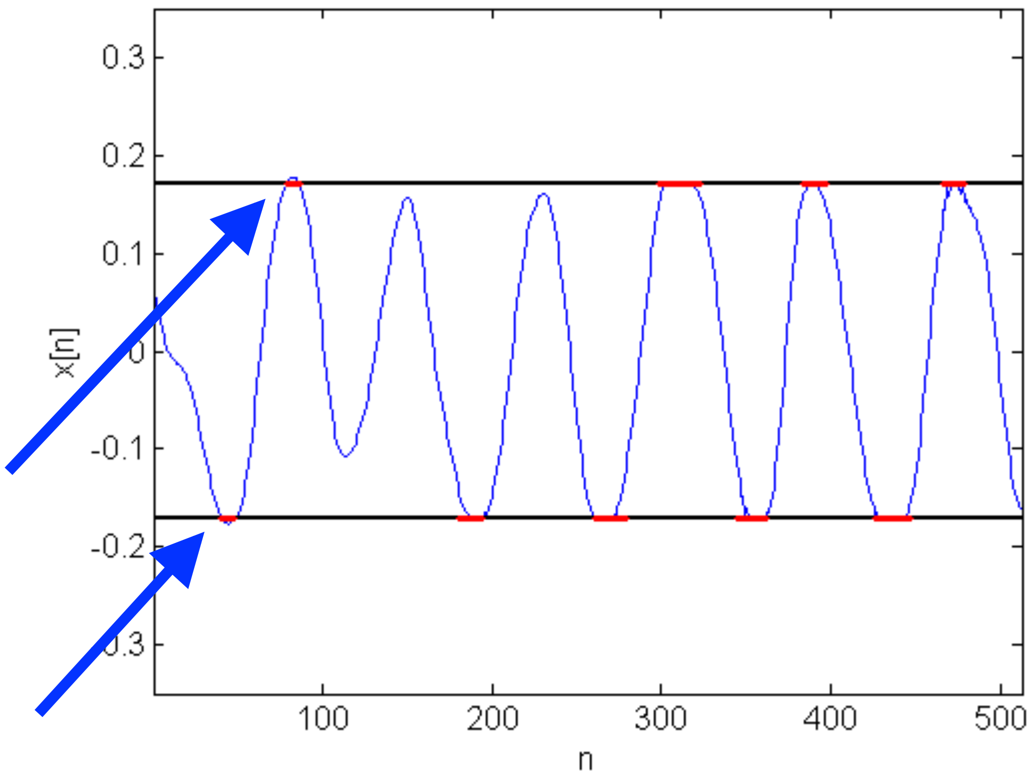


violated constraint indices = nonzero multipliers in small scale dual QP

External Active Set Strategy: 1st Iteration Result

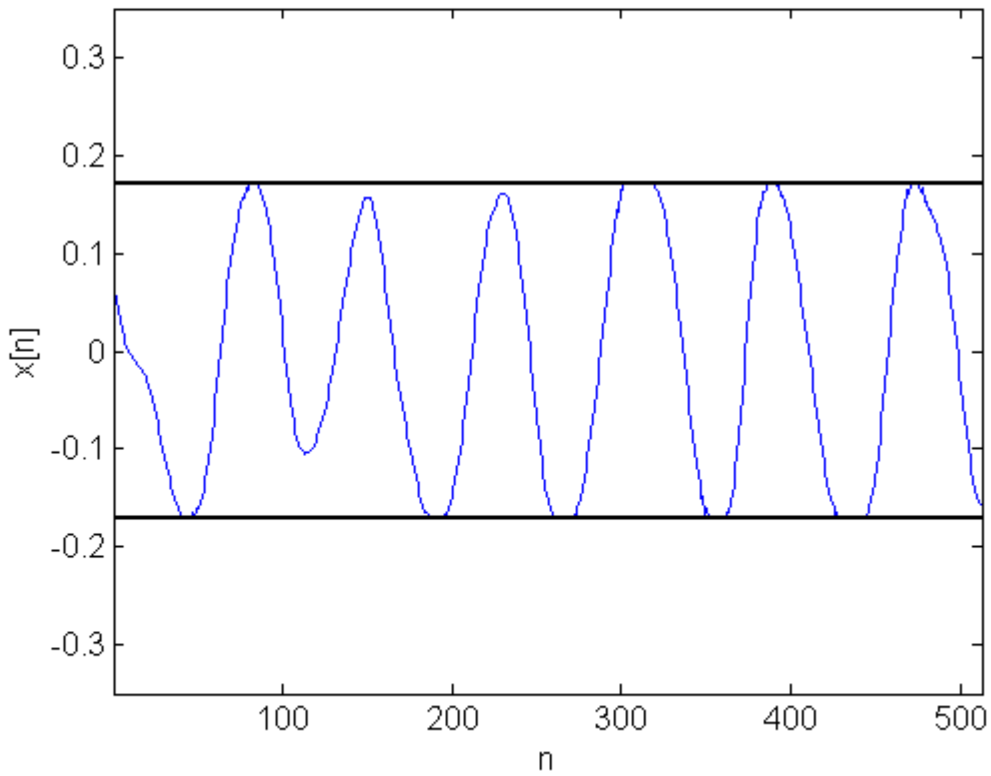


External Active Set Strategy: 1st Iteration Result

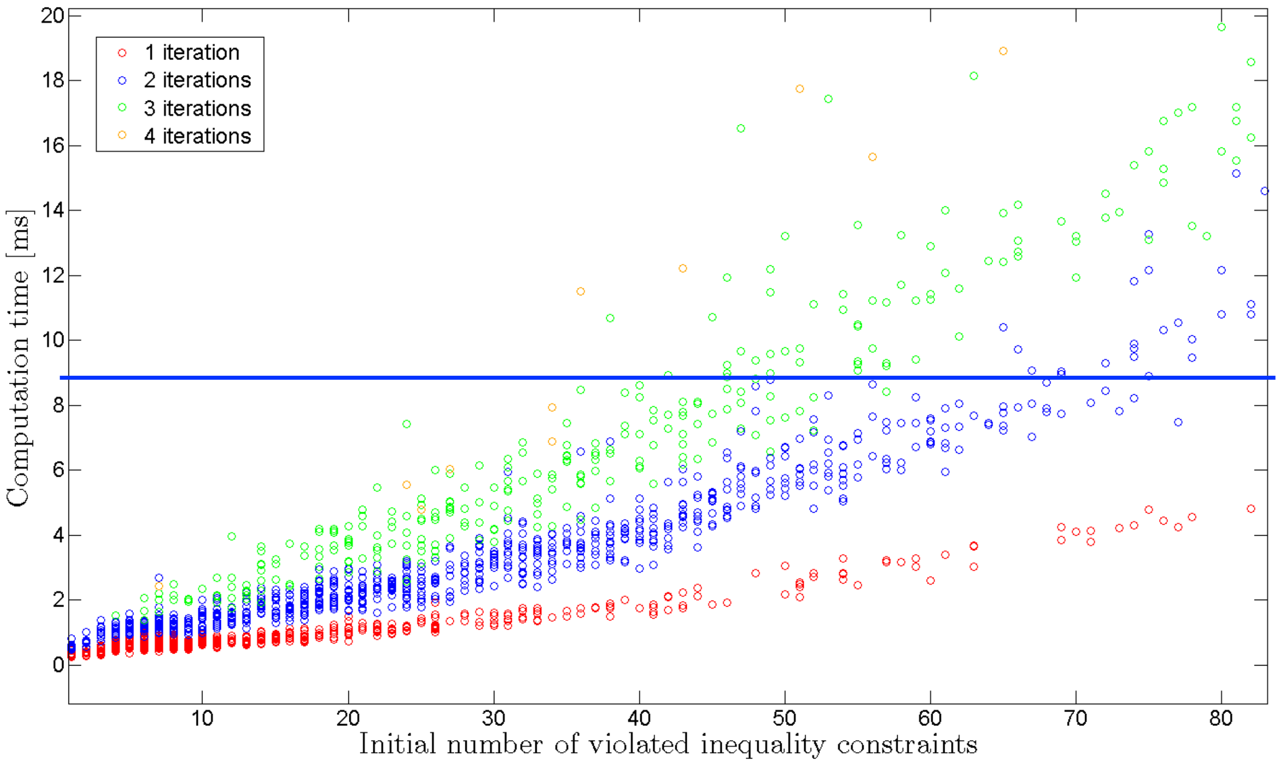


add the very few newly violated constraint indices to dual QP, solve again

External Active Set Strategy: 2nd Iteration = Solution



CPU Time Tests with External Active Set Strategy



512 variables

1024 inequality constraints

Real-time

[Intel CPU ~2.8 GHz]

- 40 x faster than standard QP solver, but not always real-time feasible

Three Tailored QP Solution Methods

- Method 1: External Active Set Strategy with Small Dual QPs
- **Method 2: Projected Gradient**
- Method 3: Nesterov's Optimal Gradient Scheme

Method 2: Projected Gradient

Algorithm 1 Projected gradient descent

Input $x \in \mathbb{R}^n$, $y^0 = rand$, Lipschitz constant L
Output $y^* \in \mathbb{R}^n$

1: $k = 0$
2: **while** stopping criterion is not met **do**
3: $\quad \boxed{\tilde{y}^{k+1} = y^k - \frac{1}{L}\nabla f(y^k)}$ Gradient step
4: $\quad y^{k+1} = \Pi_Q(\tilde{y}^{k+1})$
5: $\quad k = k + 1$
6: **end while**

Method 2: Projected Gradient

Algorithm 1 Projected gradient descent

Input $x \in \mathbb{R}^n$, $y^0 = rand$, Lipschitz constant L

Output $y^* \in \mathbb{R}^n$

1: $k = 0$

2: **while** stopping criterion is not met **do**

3: $\quad \tilde{y}^{k+1} = y^k - \frac{1}{L}\nabla f(y^k)$

4: $\quad \boxed{y^{k+1} = \Pi_Q(\tilde{y}^{k+1})}$ Projection on feasible set

5: $\quad k = k + 1$

6: **end while**

Method 2: Projected Gradient

Algorithm 1 Projected gradient descent

Input $x \in \mathbb{R}^n$, $y^0 = rand$, Lipschitz constant L
Output $y^* \in \mathbb{R}^n$

1: $k = 0$
2: **while** stopping criterion is not met **do**
3: $\quad \tilde{y}^{k+1} = y^k - \frac{1}{L}\nabla f(y^k)$
4: $\quad y^{k+1} = \Pi_Q(\tilde{y}^{k+1})$
5: $\quad k = k + 1$
6: **end while**

- Calculating the gradient is extremely cheap !

$$\nabla f(y) = D^H W D(y - x)$$ **= FFT – weighting - IFFT**

Method 2: Projected Gradient

Algorithm 1 Projected gradient descent

Input $x \in \mathbb{R}^n$, $y^0 = rand$, Lipschitz constant L
Output $y^* \in \mathbb{R}^n$

1: $k = 0$
2: **while** stopping criterion is not met **do**
3: $\quad \tilde{y}^{k+1} = y^k - \frac{1}{L} \nabla f(y^k)$
4: $\quad y^{k+1} = \Pi_Q(\tilde{y}^{k+1})$
5: $\quad k = k + 1$
6: **end while**

- Calculating the gradient is extremely cheap !

$$\nabla f(y) = D^H W D(y - x)$$ **= FFT – weighting - IFFT**

- Projecting onto feasible set is also extremely cheap !

$$\Pi_Q(y) := \arg \min_{y \in \mathbb{R}^N} \frac{1}{2} \|y - x\|_2^2 \quad \text{s.t.} \quad L \leq y \leq U$$ **= Hard clipping**

Three Tailored QP Solution Methods

- Method 1: External Active Set Strategy with Small Dual QPs
- Method 2: Projected Gradient
- **Method 3: Nesterov's Optimal Gradient Scheme**

Method 3 - Nesterov's Optimal Scheme

Algorithm 1 Projected gradient descent using Nesterov's optimal method

Input $x \in \mathbb{R}^n$, $y^{-1} = c^0 = rand$, $a^0 = 1$, Lipschitz constant L

Output $y^* \in \mathbb{R}^n$

$k = 0$

while stopping criterion is not met **do**

  $\tilde{y}^{k+1} = c^k - \frac{1}{L}\nabla f(c^k)$

  $y^{k+1} = \Pi_Q(\tilde{y}^{k+1})$

  $a^{k+1} = \frac{(1+\sqrt{4(a^k)^2+1})}{2}$

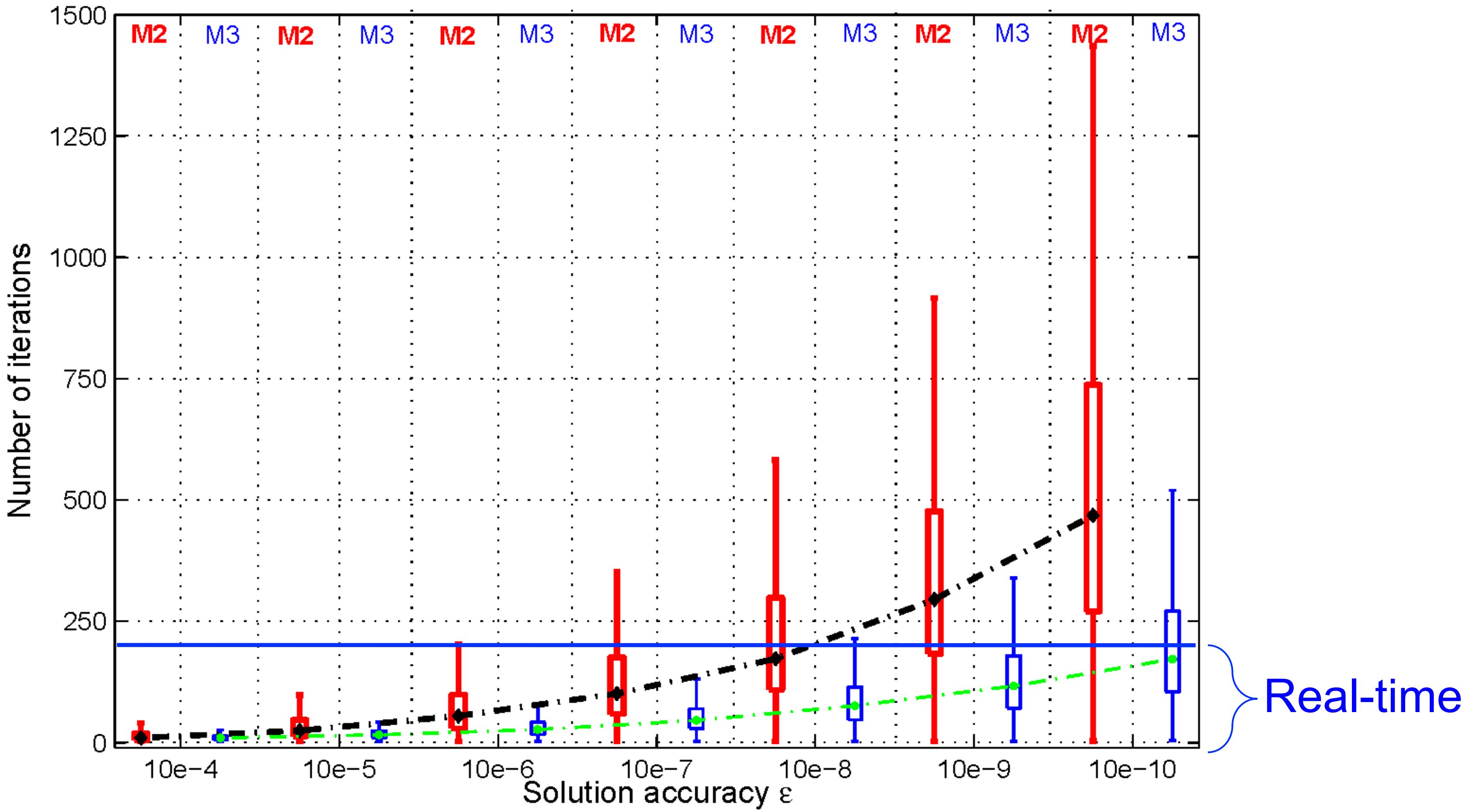
  $c^{k+1} = y^k + \frac{a^k-1}{a^{k+1}}(y^k - y^{k-1})$

  $k = k+1$

end while

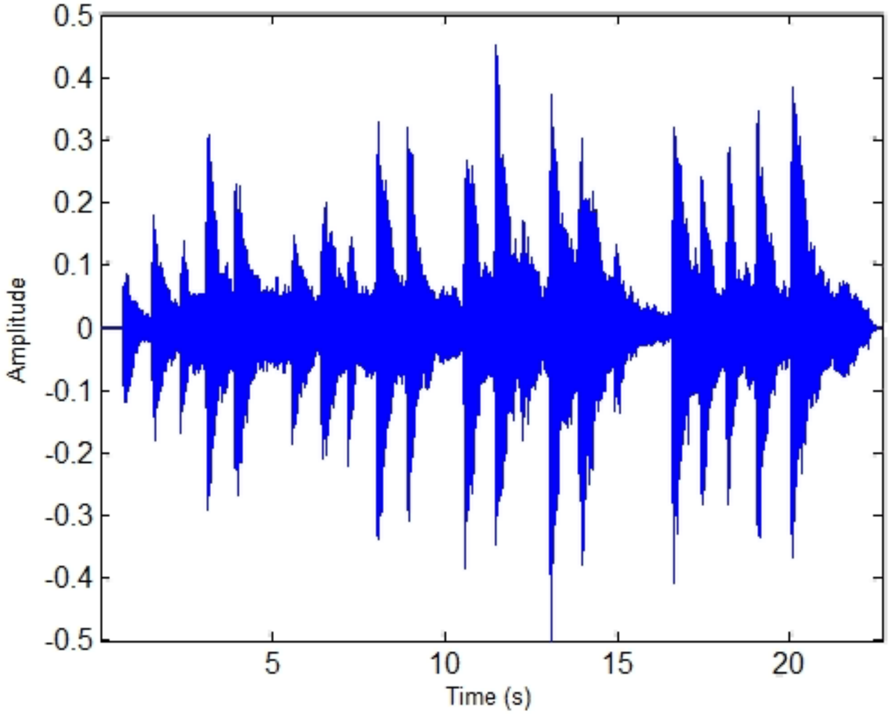
- Minor code modifications to standard projected gradient
- one extra vector addition: negligible extra cost per iteration
- faster convergence, provably with optimal rate $O(\frac{1}{\sqrt{\varepsilon}})$ [Nesterov 1983]

Audio CPU Test: Gradient (M2) vs. Nesterov (M3)

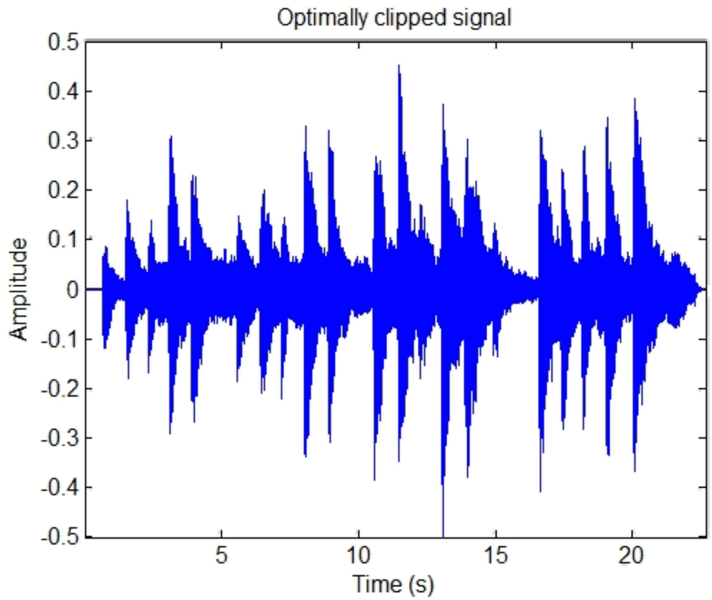


- Nesterov's scheme real-time feasible below accuracy 10^{-8}
- Already 10^{-6} delivers no perceptual difference to exact solution

Hard Clipped Signal (pour Jean-Baptiste)



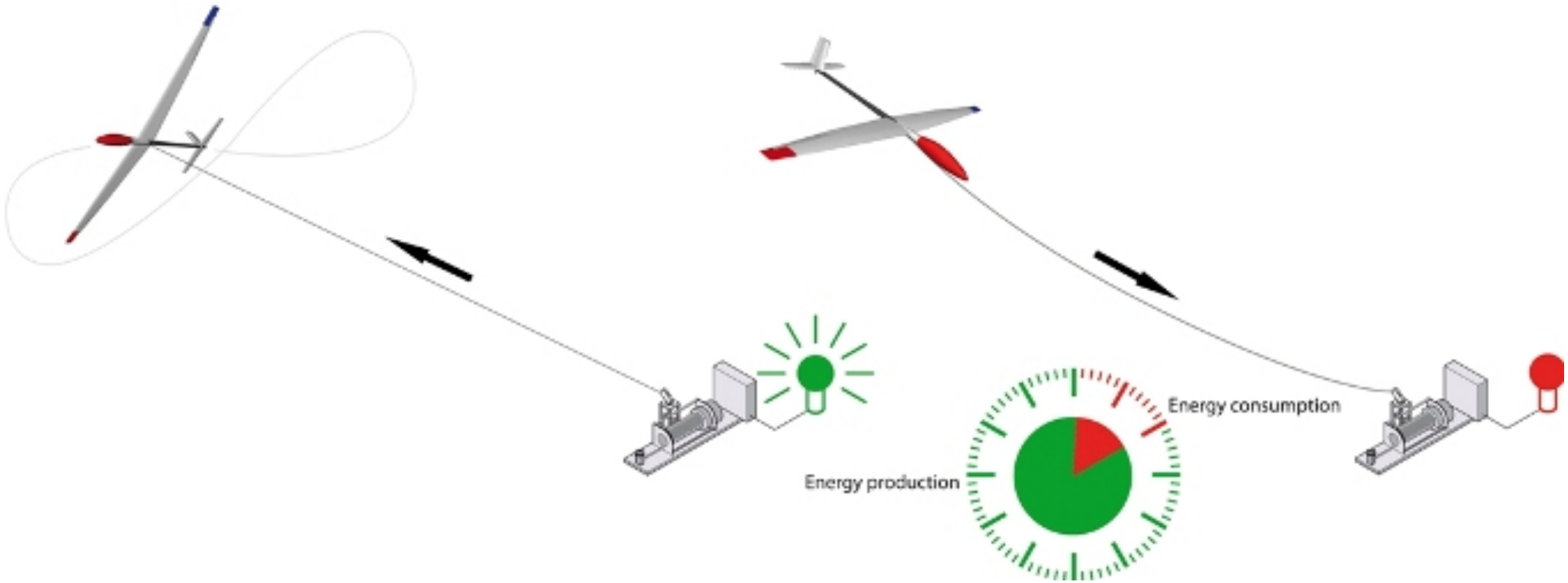
Optimally Clipped by Nesterov's Gradient Scheme



Control of Tethered Airplanes by Auto-Generated Real-Time Iterations

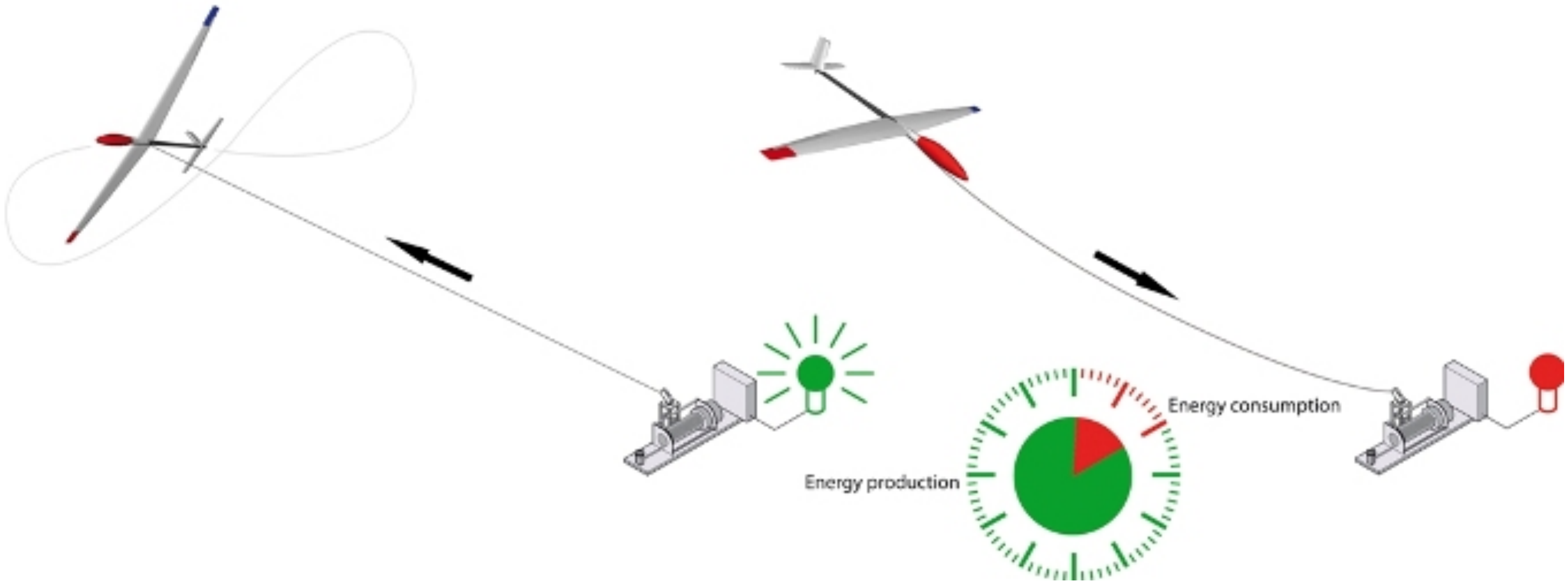
Boris Houska, Hans Joachim Ferreau, Kurt Geebelen, Reinhart Paelinck, Joris Gillis, Jan Swevers, Dirk Vandepitte, M.D.

Idea: Wind Power by Tethered Planes



Enormous potential (e.g. 5 MW for 500 m^2 wing) .

Idea: Wind Power by Tethered Planes



Enormous potential (e.g. 5 MW for 500 m^2 wing) .

Two major questions:

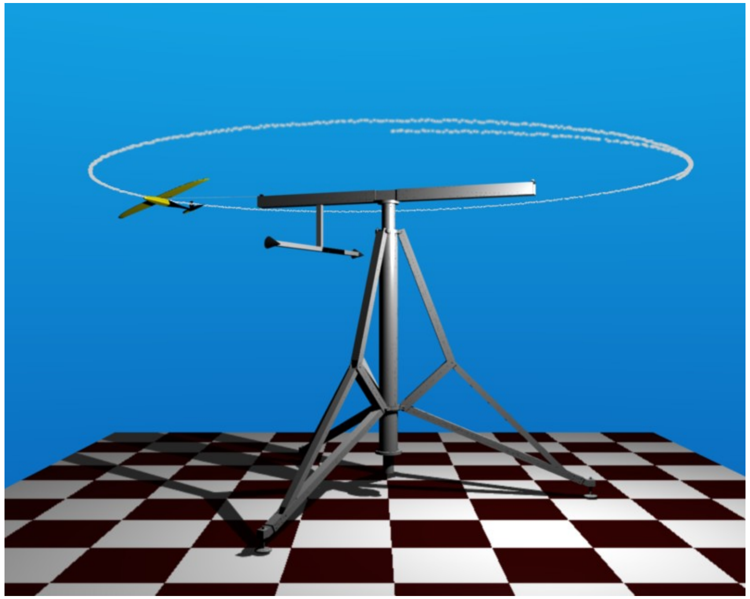
- How to start up and land ?
- How to control automatically ?

ERC Starting Grant “HIGHWIND” for 2011-2015

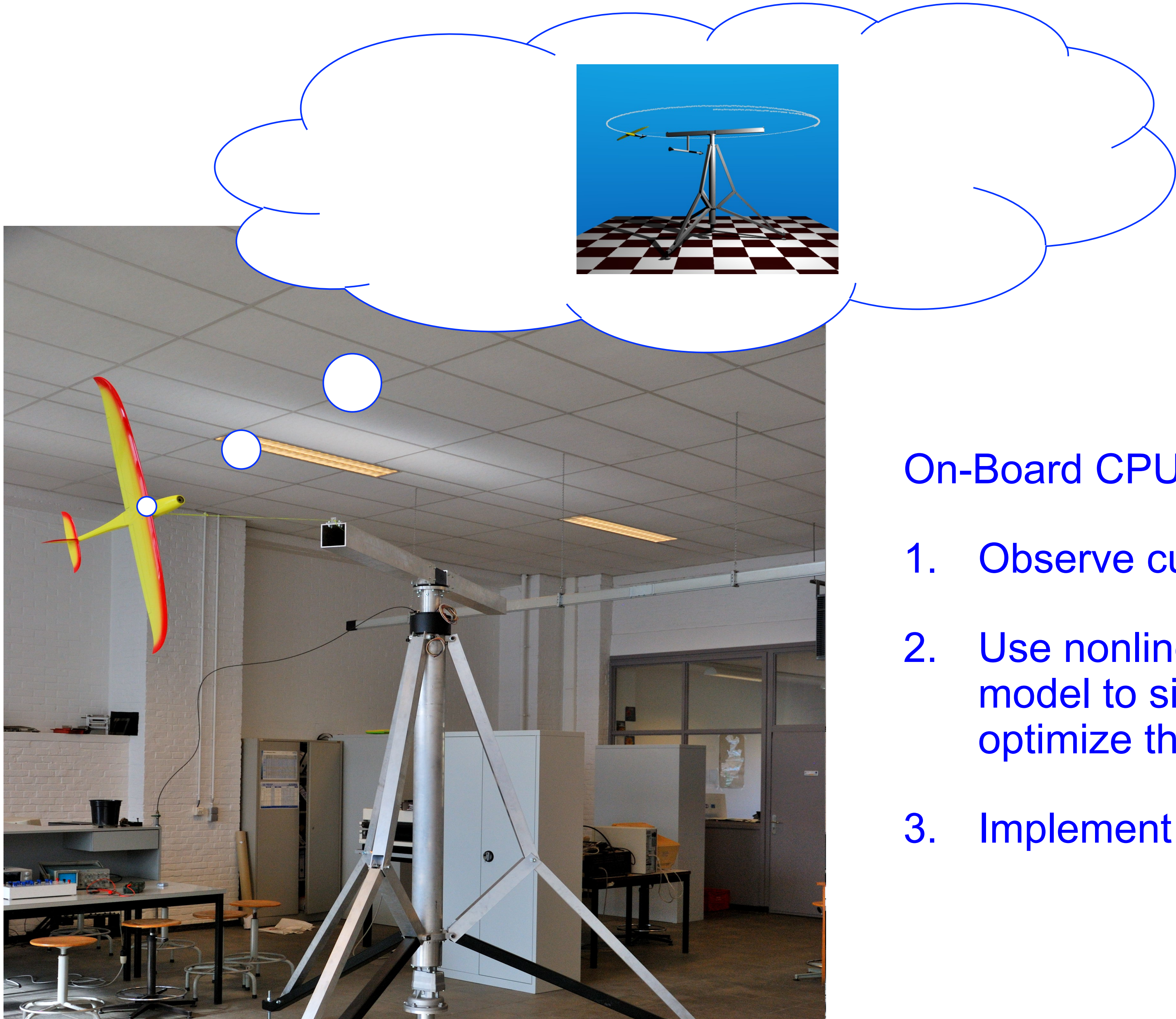
HIGHWIND

Modelling, Optimization, and Control of High Altitude Wind Power Generators

Aim: Guide the development of high altitude wind power, focus on *modeling, optimization, and control*, plus small scale experiments.



Nonlinear Model Predictive Control (NMPC)



On-Board CPU repeats:

1. Observe current state
2. Use nonlinear ODE model to simulate and optimize the future
3. Implement first control.

NMPC Embedded Optimal Control Problem

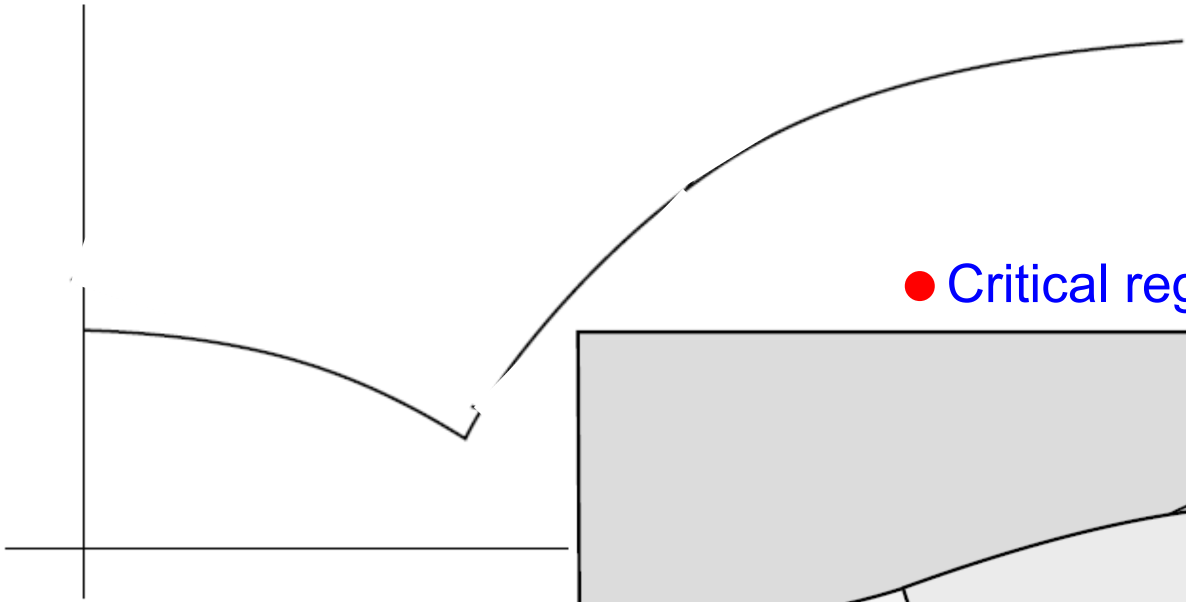
$$
\begin{array}{rrcll}
\underset{x,\,z,\,u}{\text{minimize}} & \displaystyle\sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) & + & E\,(x_N) & \\
\text{subject to} & x_0 - \bar{x}_0 & = & 0, & \\
 & x_{i+1} - f_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
 & g_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
 & h_i(x_i, z_i, u_i) & \leq & 0, & i = 0, \ldots, N-1, \\
 & r\,(x_N) & \leq & 0. &
\end{array}
$$

Structured “parametric Nonlinear Program (p-NLP)”

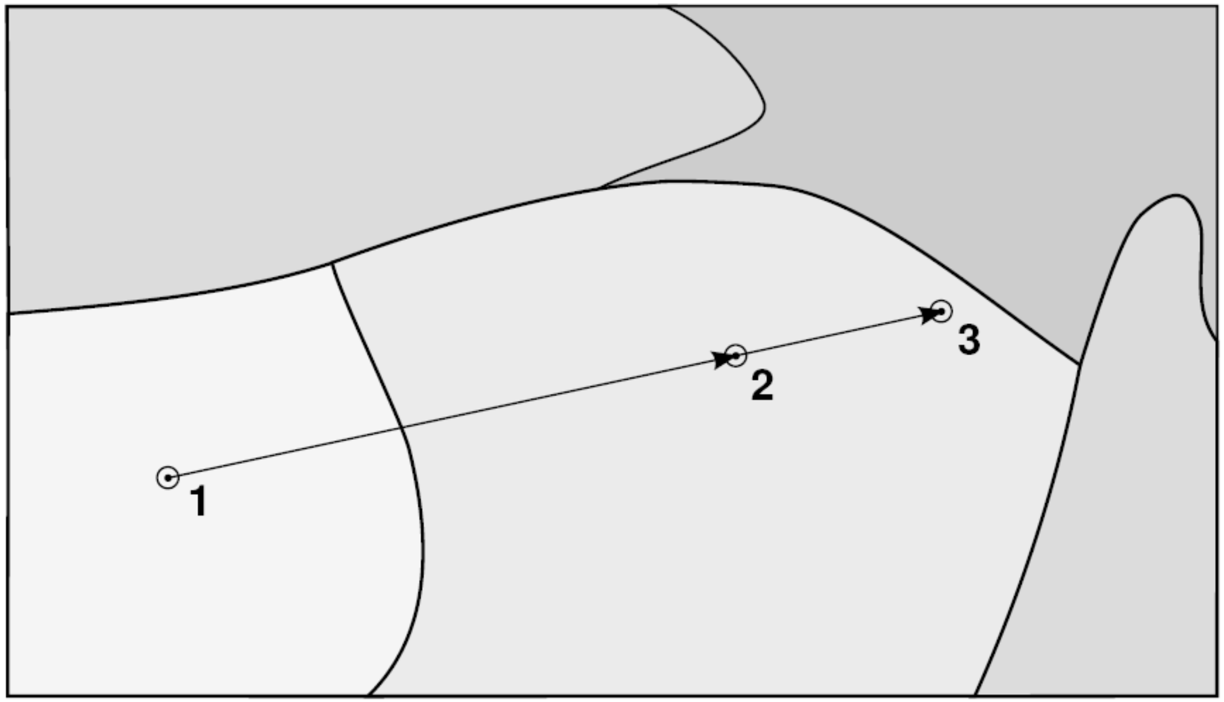
- Initial Value $\bar{x}_0$ is not known beforehand (“online data”)
- Discrete time dynamics come from ODE simulation in “direct multiple shooting” [Bock & Plitt 1984]

NMPC = parametric NLP

- Solution manifold is piecewise differentiable (kinks at active set changes)



- Critical regions are non-polyhedral

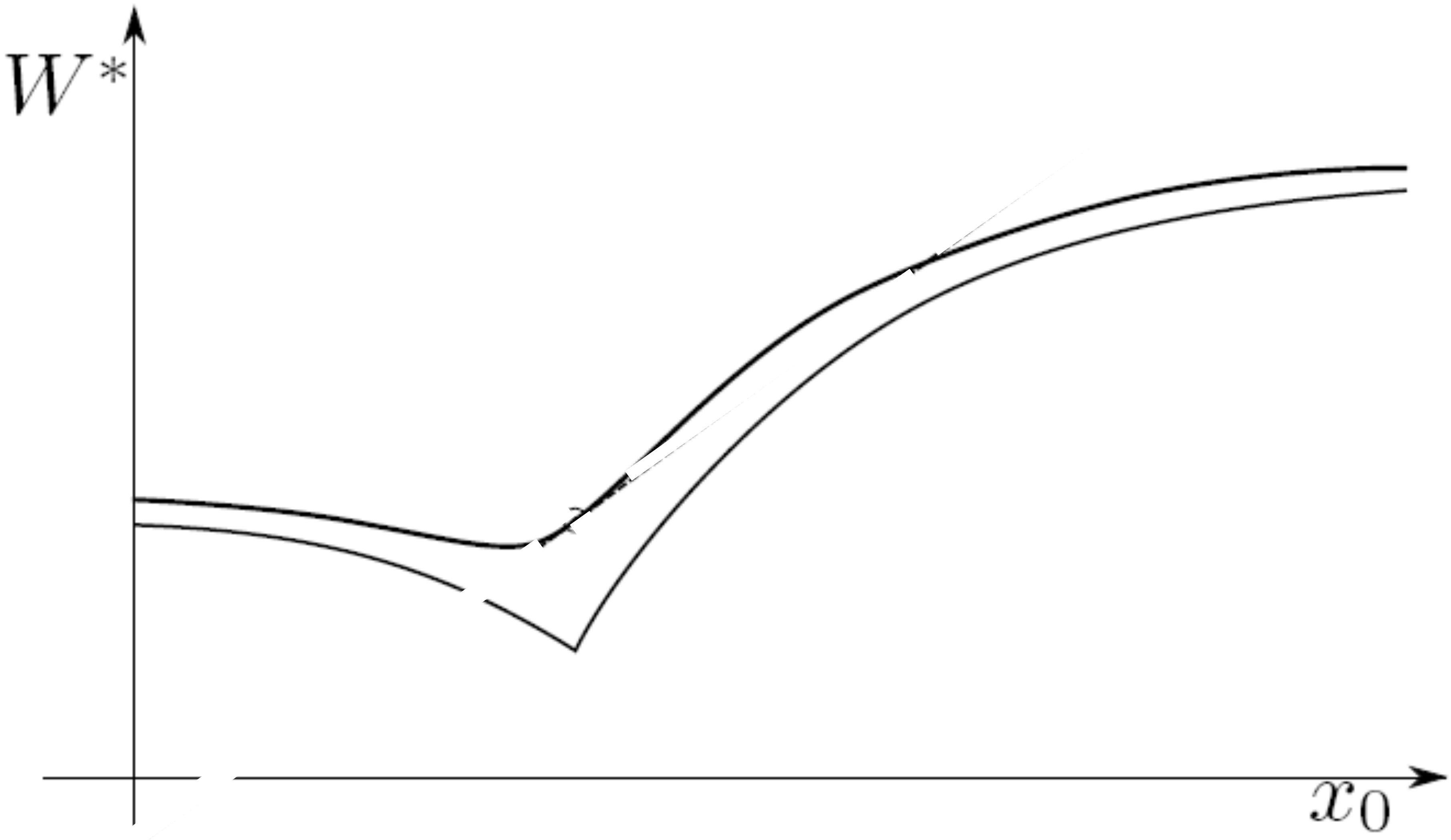


Two Approaches for p-NLP Pathfollowing

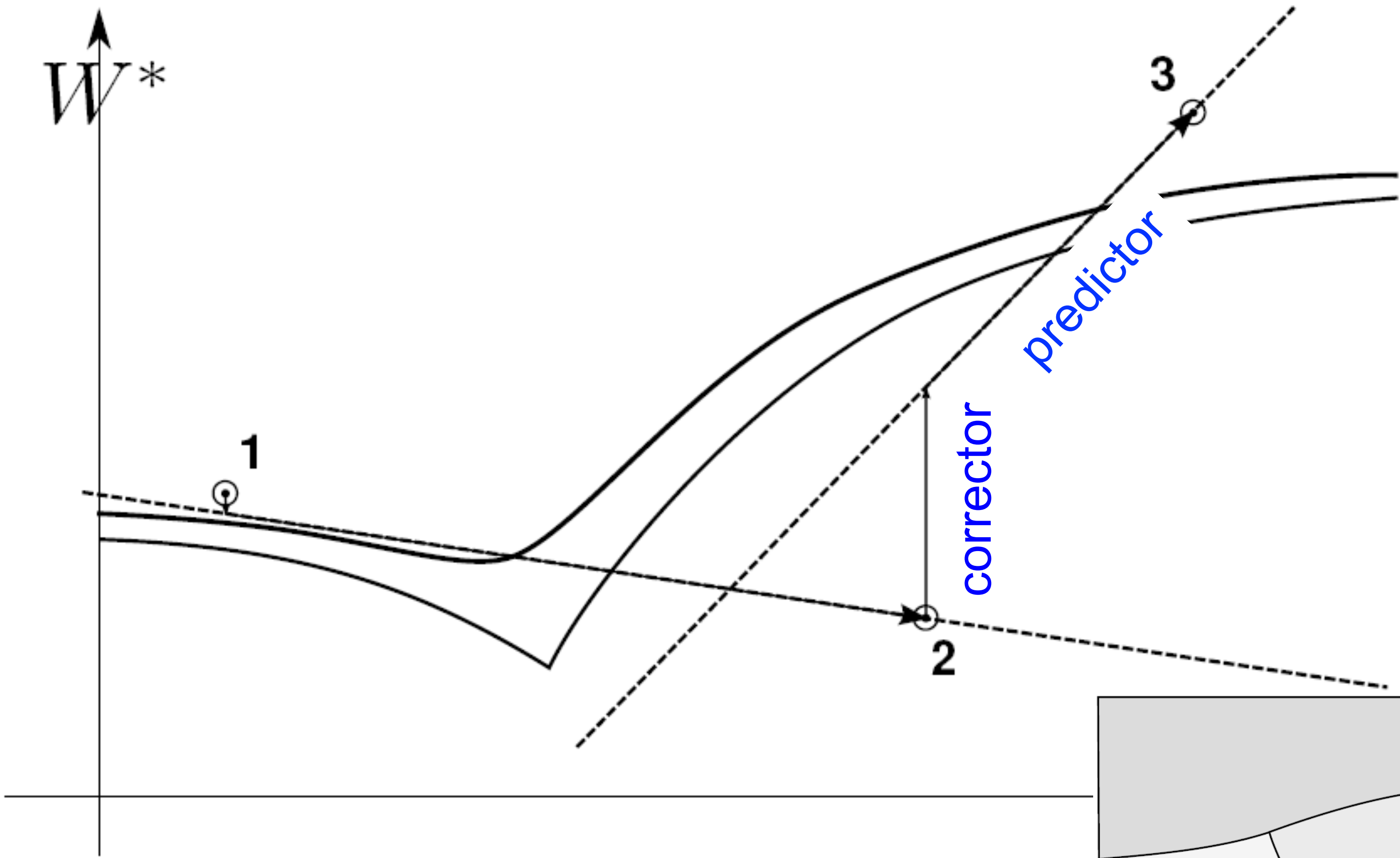
1. Interior Point (IP): use self-concordant barrier to make p-NLP problem smooth, use predictor-corrector path-following scheme [Ohtsuka 2004, Boyd & Wang 2009]

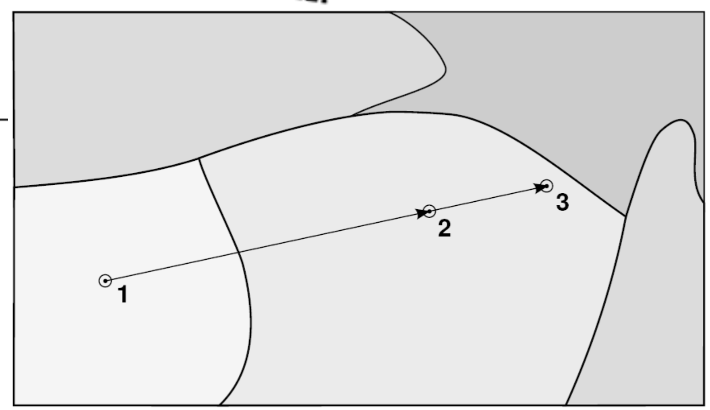
2. Sequential Quadratic Programming (SQP): solve a sequence of parametrically changing QPs [Li & Biegler 1991, D. 2001]

Note: IP with fixed barrier makes p-NLP smooth



Approach 1: IP pathfollowing for p-NLP [Ohtsuka 2004]





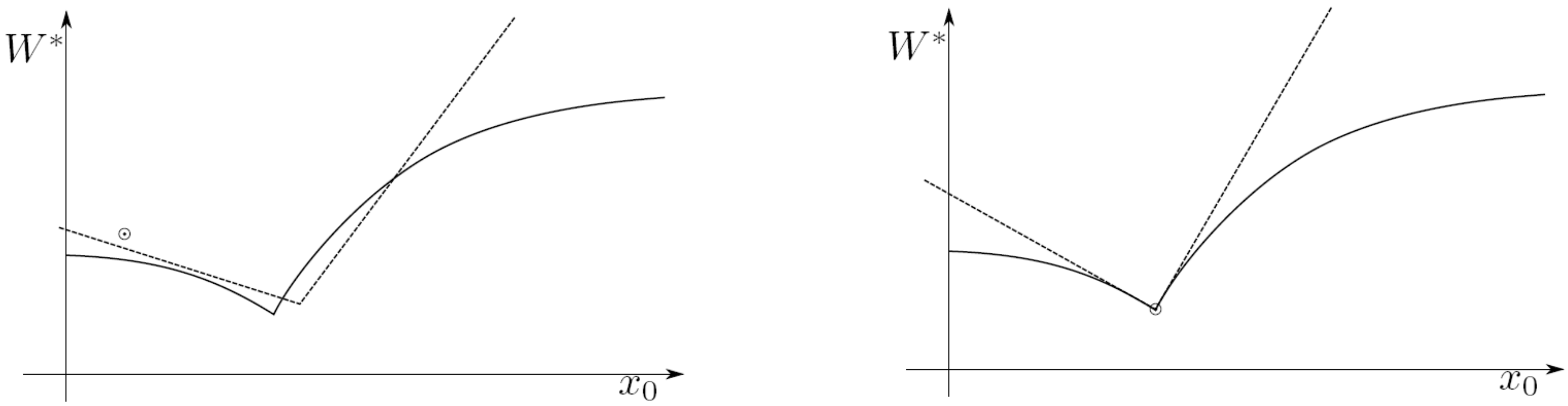
Problem: overshoot at active set changes
Can we do better?

Approach 2: Sequential Quadratic Programming for p-NLP

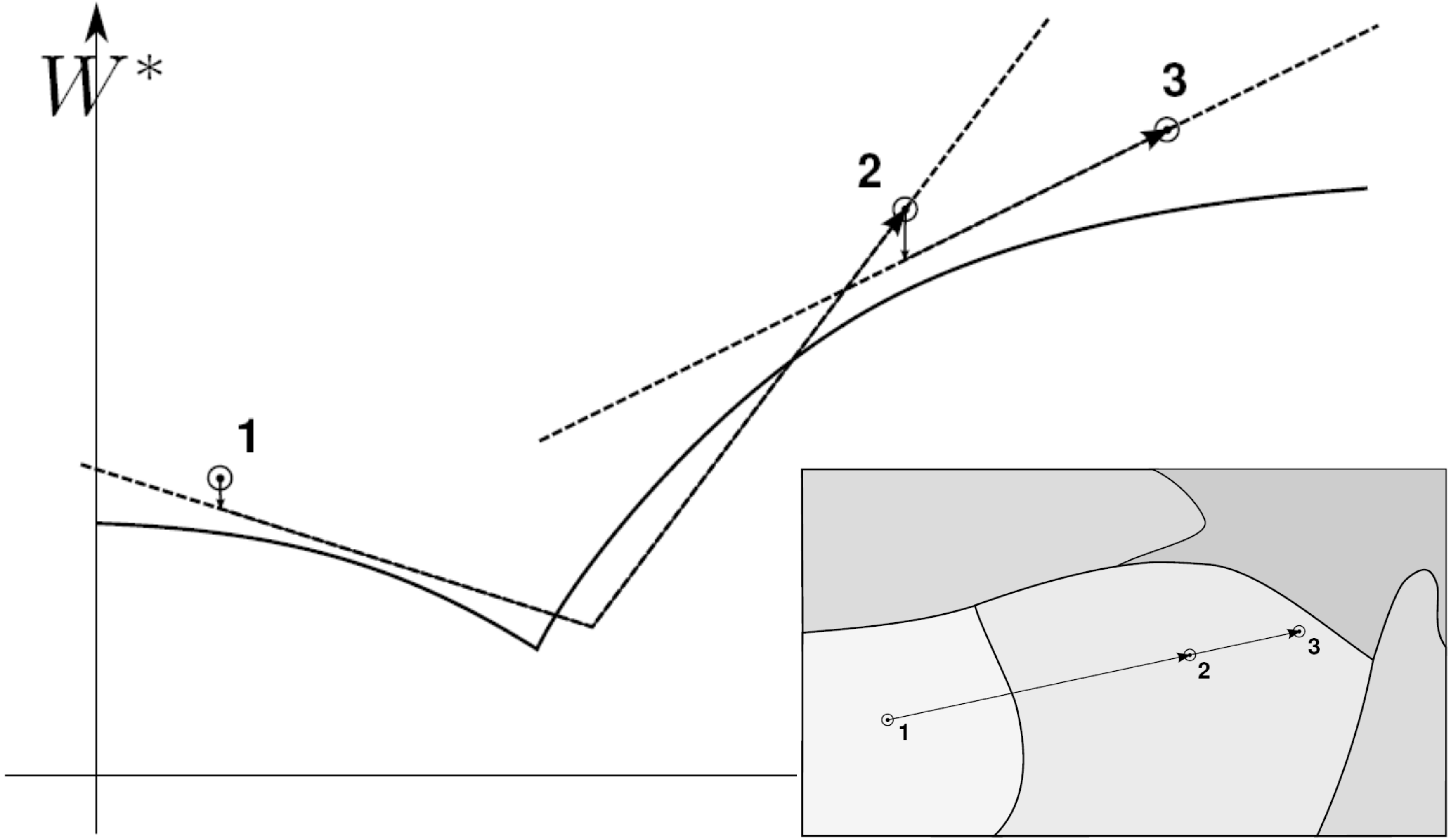
- In each iteration, linearize and solve parametric QP with inequalities

$$
\boxed{
\begin{array}{rrcll}
\underset{x,\,u}{\text{minimize}} & \displaystyle\sum_{i=0}^{N-1} L_{\text{redQP},i}(x_i, u_i) & + & E_{\text{QP}}(x_N) & \\
\text{subject to} & x_0 - \bar{x}_0 & = & 0, & \\
 & x_{i+1} - c_i - A_i x_i - B_i u_i & = & 0, & i = 0, \ldots, N-1, \\
 & \bar{h}_i + \bar{H}_i^x x_i + \bar{H}_i^u u_i & \leq & 0, & i = 0, \ldots, N-1, \\
 & r' + R x_N & \leq & 0. &
\end{array}
}
$$

- This "Initial Value Embedding" delivers first order prediction also at active set changes [D. 2001].



SQP Real-Time Iteration [D. 2001]



- long “preparation phase” for linearization
- fast “feedback phase” (QP solution once $\bar{x}_0$ is known)

Real-Time Iteration Algorithm

Initialization: Choose initial values for x and u at all multiple shooting nodes.

Repeat Online:

1) Evaluate objective function and dynamic system equations and their derivatives at current iterate.
2) Condense resulting large-scale QP into a smaller-scale dense QP.
3) Wait for the measurement x_0.
4) Compute initial value embedding.
5) Solve dense QP.
6) Send first control immediately to the process.
7) Update current iterate and shift the time.

Automatic Code Generation with ACADO Toolkit

- A Toolkit for „Automatic Control and Dynamic Optimization“
- Open-source software (LGPL 3)
- Implements direct single and multiple shooting
- Developed at OPTEC by Boris Houska & Hans Joachim Ferreau

- Uses symbolic user syntax
  - to generate derivative code by automatic differentiation
  - to detect model sparsity
  - to **auto-generate C-code** for NMPC Real-Time Iterations...

Automatic Code Generation with ACADO Toolkit

- A Toolkit for „Automatic Control and Dynamic Optimization“
- Open-source software (LGPL 3)
- Implements direct single and multiple shooting
- Developed at OPTEC by Boris Houska & Hans Joachim Ferreau

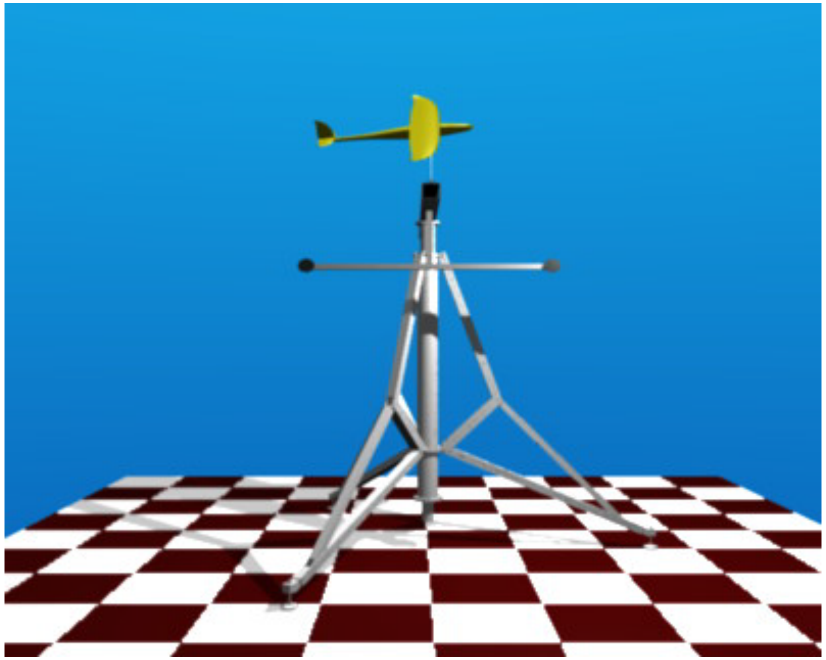
- Uses symbolic user syntax
  - to generate derivative code by automatic differentiation
  - to detect model sparsity
  - to **auto-generate C-code** for NMPC Real-Time Iterations...

```
params.g[0] = acadoWorkspace.g[4] +
acadoWorkspace.H[4]*acadoWorkspace.deltaY[0] +
acadoWorkspace.H[18]*acadoWorkspace.deltaY[1] +
```

CPU time per real-time iteration

In each embedded optimization problem:

- ODE model with 4 states & 2 controls
- 30 Runge-Kutta steps of RK4
- 10 multiple shooting intervals
- 60 QP variables



	CPU time	Percentage
Integration & sensitivities	711 μs	84 %
Condensing	71 μs	8 %
QP solution (with `qpOASES`)	34 μs	4 %
Remaining operations	27 μs	$<$ 4 %
A complete real-time iteration	843 μs	100 %

Auto-generated C-code 300 times faster than standard real-time iterations [Ilzhoefer et al. 2007]

1 kHz feedback possible!

First Indoors Test Flights (not yet controlled)

Summary

- Embedded Optimization promises to revolutionize all aspects of control engineering and signal processing
- It needs sophisticated numerical methods
- OPTEC develops open source software for embedded optimization
- Powerful tool in applications:
  - Optimal clipping for hearing aids (convex, large, 100 Hz)
  - Predictive control of tethered airplanes (non-convex, small, 1000 Hz)

- Open postdoc and PhD positions in Highwind ERC project, on „Mathematical modelling and optimal control of tethered airplanes“

OPTEC QP Workshop, November 25-26

OPTEC Workshop on Large Scale Convex Quadratic Programming - Algorithms, Software, and Applications

Leuven, November 25 and 26, 2010

Thursday:

- **Yurii Nesterov**: Fast gradient methods for large-scale optimization problems
- **Philippe Toint**: Inexact range-space Krylov solvers for linear systems arising from inverse problems
- **Eric Kerrigan**: A Well-conditioned Interior Point Method for Quadratic Programming with an Application to Optimal Control
- **Nick Gould**: CQP: a fortran 90 module for large-scale convex quadratic programming
- **Stephen Wright**: Efficient methods for structured quadratic programs
- **Joachim Dahl**: Solving large-scale convex QPs with MOSEK

OPTEC QP Workshop, November 25-26

Friday

- **Michael Saunders**: A Regularized Active-set Method for Sparse Convex Quadratic Programming
- **Christian Kirches**: A Block Structured Active Set Method for Mixed--Integer Optimal Control Problems
- **Daniel Axehill**: A Dual Active Set-Like Quadratic Programming Algorithm Tailored for Model Predictive Control
- **John Bagterp Jørgensen**: Convex QP Algorithms for Linear MPC with Soft Output Constraints
- **Oleg Burdakov**: Monotonicity recovering QP-based methods for postprocessing finite element solutions

→ **Register by Nov 1**, google “optec qp”, fee EUR 120


Merci !
B
A