Combinatorial Integral Approximation for Mixed-Integer Nonlinear Optimal Control

Sebastian Sager

Junior Research Group
Mathematical and Computational Optimization

Interdisciplinary Center for Scientific Computing
University of Heidelberg

October, 2010
Banff


DO NOT
FEED
THE MOOSE

Outline

MIOC introduction

Methods for Mixed-Integer Optimal Control

...based on HJB / Dynamic Programming

...based on Direct Methods

MINLP

MS MINTOC

Switching Costs and Combinatorial Constraints

MathOpt Heidelberg

Applications I: Valves and ports

- Watergates and on/off pumps in water networks

  [Burgschweiger, Deuerlein, Gugat, Hante, Leugering, Steinbach, . . .]

Images: Wikipedia, BASF SE

Applications I: Valves and ports

- Watergates and on/off pumps in water networks

  [Burgschweiger, Deuerlein, Gugat, Hante, Leugering, Steinbach, . . .]

- Valves, (de)compressors in gas networks

  [Leugering, Martin, Schultz, Steinbach, S. Ulbrich, . . .]

Images: Wikipedia, BASF SE

Applications I: Valves and ports

- Watergates and on/off pumps in water networks

  [Burgschweiger, Deuerlein, Gugat, Hante, Leugering, Steinbach, . . .]
- Valves, (de)compressors in gas networks

  [Leugering, Martin, Schultz, Steinbach, S. Ulbrich, . . .]
- Valves in separation processes

  [Biegler, Engell, Findeisen, Grossmann, Kienle, Marquardt, Swartz, . . .]
  - Slop cut recycling in batch distillation

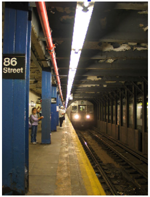
    [Bock, Diehl, S., . . .]
  - Evaporator operation [Sonntag, Engell, . . .]
  - Simulated Moving Bed superstructure

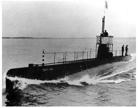
    [Biegler, Engell, Potschka, S., Weismantel, Westerlund, . . .]

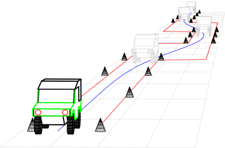
Images: Wikipedia, BASF SE

Applications II: optimal gear shifts in transport

- New York subway: energy optimal rides (?), (?)
- Submarine control (?)
- Look-ahead control in heavy duty trucks (?), (?), (?)
  - Feedback control, GPS data, $\geq$ 16 gears
- Elchtest benchmark: automobile testdriving (??), (?), [Borelli, ...]
- Periodic time-optimal automobile driving, (?)
- Formula 1 racing, (?)







MathOpt Heidelberg

Images: Wikipedia, youtube (Hellström), Pesch, Gerdts in 'Mathematics — Key Technology for the Future'

iwr

Applications III: Others

- Traffic lights in Traffic flow [Aw, Leugering, Rascle, . . .], (?), (?)
- Biology / Medicine: inhibition, treatment [Engelhart, Lebiedz, S., . . .]
- Economics: yes/no decisions (part time, 2nd job?) [Kübler, Kuhn]
- Robot swarm movement and communication (?)
- Optimum experimental design [Bock, Körkel, Hoffmann, Kostina, Lebiedz, Schittkowski, Schlöder, Seidel-Morgenstern, . . .]
  - Variables: is measurement done or not, $w(x, t) \in \{0, 1\}$
- . . .
- benchmark library: `http://mintoc.de`

Problem class

$$
\begin{aligned}
\min_{x,v,u}\ & \phi(x(t_f)) \\
\text{s.t.}\quad \dot{x}(t) &= f(x(t), v(t), u(t)), \\
0 &\leq c(x(t), u(t)), \\
0 &\leq r_{\text{in}}(x(t_0), \dots, x(t_f)), \\
0 &= r_{\text{eq}}(x(t_0), \dots, x(t_f)), \\
v(t) &\in \Omega := \{v^1, v^2, \dots, v^{n_\omega}\}, \quad t \in [t_0, t_f].
\end{aligned}
$$

- x differential states, u, v control functions
- All functions sufficiently smooth
- Generalizations: PDEs, algebraic variables, constant-in-time control values, multi-stage, ...
- Focus of talk: controls $v(t)$ from finite set $v^i \in \Omega \subseteq \mathbb{R}^{n_v}$.

MathOpt Heidelberg

Why is this problem class difficult?

$$\begin{array}{llcl} \min\limits_{x,v,u} & \phi(x(t_f)) & & \\ \text{s.t.} & \dot{x}(t) & = & f(x(t), v(t), u(t)), \\ & 0 & \leq & c(x(t), u(t)), \\ & 0 & \leq & r_{\text{in}}(x(t_0), \ldots, x(t_{\text{f}})), \\ & 0 & = & r_{\text{eq}}(x(t_0), \ldots, x(t_{\text{f}})), \\ & v(t) & \in & \Omega := \{v^1, v^2, \ldots, v^{n_\omega}\}, \quad t \in [t_0, t_{\text{f}}]. \end{array}$$

- Nonlinear
- Differential equations
- Path and point constraints
- Integer decisions (in function space)

MathOpt Heidelberg

iwr

Outline

MIOC introduction

Methods for Mixed-Integer Optimal Control

...based on HJB / Dynamic Programming

...based on Direct Methods

MINLP

MS MINTOC

Switching Costs and Combinatorial Constraints

Alternative Methods

- HJB / Dynamic Programming
  - based on Bellman's Principle of Optimality
  - tabulated enumeration
  - easy to extend to integer controls
  - but: curse of dimensionality

MathOpt Heidelberg

Alternative Methods

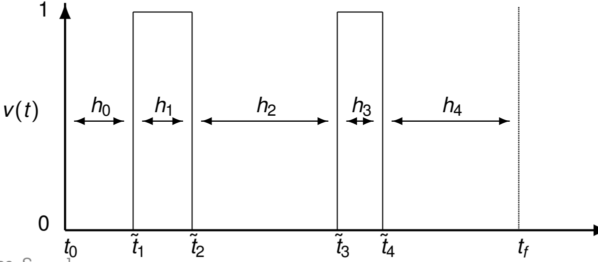

- HJB / Dynamic Programming
  - based on Bellman's Principle of Optimality
  - tabulated enumeration
  - easy to extend to integer controls
  - but: curse of dimensionality
- Pontryagin's Maximum Principle
  - use optimality conditions in function space
  - solve resulting boundary value problem
  - $u^*(t)$ is the pointwise maximum of the Hamiltonian

$$u^*(t) = \arg \max_{u \in U} H(x(t), u, \lambda(t))$$

  U may also be a discrete set!
  - practice: use switching functions to detect changes
  - difficult for singular and path-constrained arcs
  - analytical work, needs to be well initialized

Switching time optimization

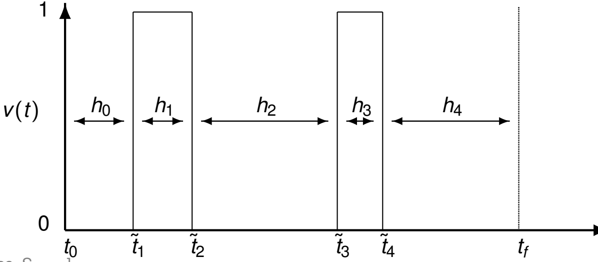
Take given switching order with piecewise fixed controls $v(\cdot)$, optimize interval lengths (after standard time transformation)



[Gerdts, Kaya, Leineweber, Maurer, Noakes, S., . . .]

Switching time optimization

Take given switching order with piecewise fixed controls $v(\cdot)$, optimize interval lengths (after standard time transformation)



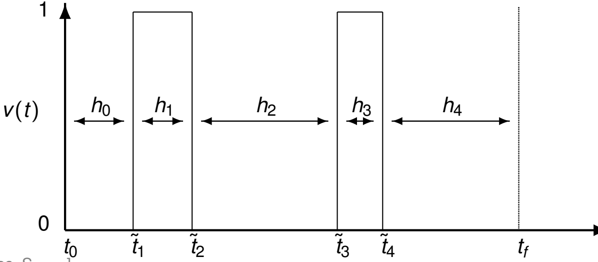
[Gerdts, Kaya, Leineweber, Maurer, Noakes, S., ...]

$$\min_{x,v,u} \phi(x(t_f))$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t), v(t), u(t)), \quad t \in [t_0, t_f]$$

Switching time optimization

Take given switching order with piecewise fixed controls $v(\cdot)$, optimize interval lengths (after standard time transformation)



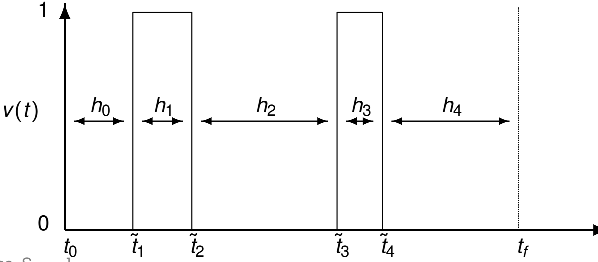
[Gerdts, Kaya, Leineweber, Maurer, Noakes, S., ...]

$$\min_{x,v,u} \phi(x(t_f))$$
$$\text{s.t.} \quad \dot{x}(t) = f(x(t), v(t), u(t)), \quad t \in [t_0, t_\mathrm{f}]$$

$$\min_{x,h,u} \phi(x(t_f))$$
$$\text{s.t.} \quad \dot{x}(t) = f(x(t), v^{i_j}, u(t)), \quad t \in [\tilde{t}_j, \tilde{t}_{j+1}]$$
$$\ldots$$

Switching time optimization

Take given switching order with piecewise fixed controls $v(\cdot)$, optimize interval lengths (after standard time transformation)



[Gerdts, Kaya, Leineweber, Maurer, Noakes, S., ...]

$$\min_{x,v,u} \phi(x(t_f))$$
$$\text{s.t.} \quad \dot{x}(t) = f(x(t), v(t), u(t)), \quad t \in [t_0, t_f]$$

$$\min_{x,h,u} \phi(x(t_f))$$
$$\text{s.t.} \quad \dot{x}(t) = f(x(t), v^{i_j}, u(t)), \quad t \in [\tilde{t}_j, \tilde{t}_{j+1}]$$
$$\dots$$

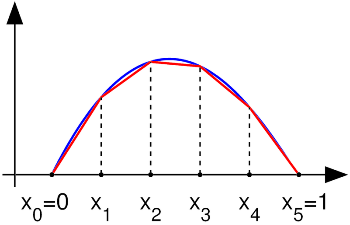
- Main issue: initialization

MathOpt Heidelberg

iwr

Linear approximations

- MILP way faster than MINLP solvers
- Idea: approximate nonlinearities and obtain MILP
  - A) Use linearizations in NMPC context [Jones, Morari, Borrelli, . . .]
  - B) Use piecewise linear approximation (?)



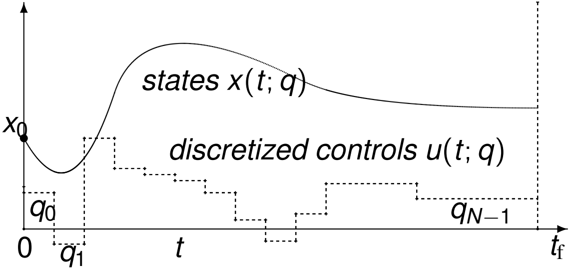
- Main issues: problem size, adaptivity, restarts

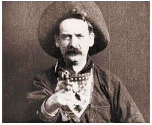
MathOpt Heidelberg

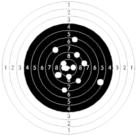
Image: Wikipedia

Direct Single Shooting [Hicks, Ray 1971; Sargent, Sullivan 1977]

Discretize controls $u(t)$ on fixed grid $0 = t_0 < t_1 < \ldots < t_N = t_\mathrm{f}$.

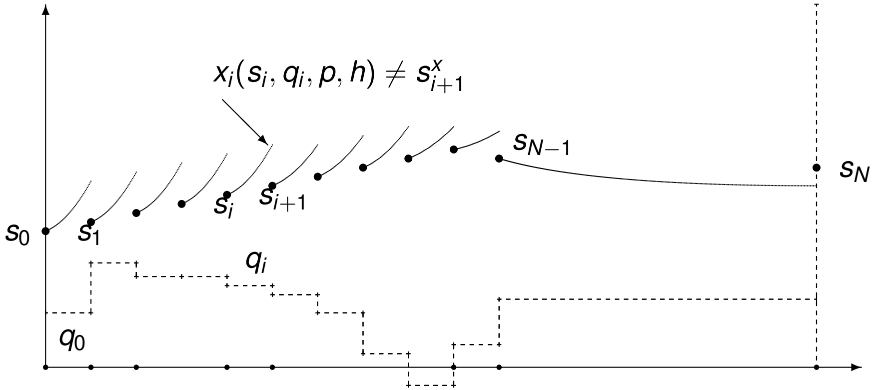






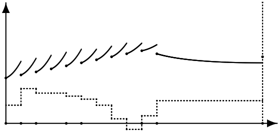
Regard states $x(t)$ on $[t_0, t_\mathrm{f}]$ as dependent variables.
Use numerical integration to obtain state as function $x(t; q, x_0)$ of finitely many control parameters $q = (q_0, q_1, \ldots, q_{N-1})$ and the initial value x_0.

Direct Multiple Shooting [Bock and Plitt, 1981, 1984, ...]



- Idea: Decouple intervals, add extra continuity constraints.
- Denote each interval's variables by $w_i := (s^x_i, s^z_i, q_i)$.
- Summarize all in large vector $w := (w_0, \ldots, w_N)$.

NLP in Direct Multiple Shooting [Bock and Plitt, 1981, 1984, . . .]



$$\min_{w} \sum_{i=0}^{N} \phi_i(w_i) \;\; \text{s.t.}$$

$$\left\{ \begin{array}{rcll} s^x_{i+1} - x_i(w_i) & = & 0 & \text{(continuity)} \\ g_i(w_i) & = & 0 & \text{(algebraic consistency)} \\ c_i(w_i) & \geq & 0 & \text{(path constraints)} \\ \sum_{i=0}^{N} r_{\mathrm{i}}(w_i) & \geq & 0 & \text{(multipoint inequality constraints)} \\ \sum_{i=0}^{N} r_{\mathrm{e}}(w_i) & \geq & 0 & \text{(multipoint equality constraints)} \end{array} \right.$$

Intermediate summary

- Large problem class with many applications
- Different approaches to solve control problems
  - Dynamic Programming
  - Maximum Principle
  - Direct Methods: discretize controls, solve NLP
- Direct Methods: different ways to parameterize states

MathOpt
Heidelberg

MINLP approach to solve MIOCPs

$$\begin{array}{llll} \min\limits_{x,v,u} \phi(x(t_f)) & & & \\ \text{s.t.} & \dot{x}(t) & = & f(t, x(t), v(t), u(t)), \quad t \in [t_0, t_f], \\ & 0 & \leq & c(t, x(t), v(t), u(t)), \\ & 0 & \leq & r_i(x(t_0), \dots, x(t_f)), \\ & 0 & = & r_e(x(t_0), \dots, x(t_f)), \\ & v(t) & \in & \Omega. \end{array}$$

- Consider the infinite dimensional problem
- Discretize

MINLP approach to solve MIOCPs

$$
\begin{aligned}
\min_{x,v,u} \; & \phi(x(t_f)) \\
\text{s.t.} \quad \dot{x}(t) &= f(t, x(t), v(t), u(t)), \quad t \in [t_0, t_f], \\
0 &\leq c(t, x(t), v(t), u(t)), \\
0 &\leq r_i(x(t_0), \ldots, x(t_f)), \\
0 &= r_e(x(t_0), \ldots, x(t_f)), \\
v(t) &\in \Omega.
\end{aligned}
$$

- Consider the infinite dimensional problem
- Discretize
- Let variables inherit integer constraint

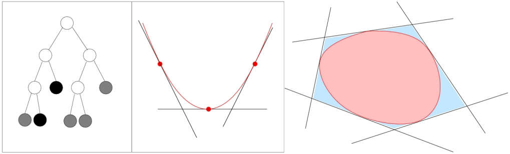
$$
\begin{aligned}
\min_{\gamma,\beta} \quad & F(\gamma, \beta) \\
\text{s.t.} \quad 0 &= G(\gamma, \beta) \\
0 &\leq H(\gamma, \beta) \\
\beta_i &\in \Omega, i = 1..N
\end{aligned}
$$

MathOpt Heidelberg

iwr

Mixed–Integer Nonlinear Programming

- Generic algorithms: Nonlinear Branch & Bound, Outer Approximation, Disjunctive Programming, ...



- Active research area
  - [Bonami, Wächter, ...] (Bonmin)
  - [Leyffer, Linderoth, ...] (FilMint)
  - [Biegler, Floudas, Grossmann, Lee, Marquardt, Oldenburg, Sahinidis, Weismantel, ...]
  - ...
- But: extremly expensive for control problems

New Approach

$$
\begin{aligned}
\min_{x,v,u} \ & \phi(x(t_f)) \\
\text{s.t.} \quad \dot{x}(t) &= f(x(t), v(t), u(t)), \\
0 &\le c(x(t), u(t)), \\
0 &\le r_{\text{in}}(x(t_0), \dots, x(t_f)), \\
0 &= r_{\text{eq}}(x(t_0), \dots, x(t_f)), \\
v(t) &\in \Omega := \{v^1, v^2, \dots, v^{n_\omega}\}, \quad t \in [t_0, t_f].
\end{aligned}
$$

- Important: lower bounds
- Relaxation $v \in \Omega \longrightarrow v \in \text{conv}\ \Omega$ is too weak!
- Is there a better relaxation?

(Partial) Outer convexification [Sager, 2005]

Easy to show:

$$\begin{aligned} v(\cdot) &\in \Omega := \{v^1, v^2, \dots, v^{n_\omega}\} \\ \dot{x}(t) &= f(x(t), v(t), u(t)) \end{aligned}$$

$$\Longleftrightarrow$$

MathOpt Heidelberg

(Partial) Outer convexification [Sager, 2005]

Easy to show:

$$\begin{aligned} v(\cdot) &\in \Omega := \{v^1, v^2, \dots, v^{n_\omega}\} \\ \dot{x}(t) &= f(x(t), v(t), u(t)) \end{aligned}$$

$$\iff$$

$$\begin{aligned} \omega(\cdot) &\in \{0,1\}^{n_\omega} \\ \dot{x}(t) &= \sum_{i=1}^{n_\omega} f(x(t), v^i, u(t))\, \omega_i(t) \\ \sum_{i=1}^{n_\omega} \omega_i(t) &= 1, \quad t \in [t_0, t_\mathrm{f}] \end{aligned}$$

MathOpt Heidelberg

Problem class after Outer Convexification

$$
\begin{array}{llll}
\min\limits_{x,u,\omega} \phi(x(t_f)) & & & \\
\text{s.t.} & \dot{x}(t) & = & \sum_{i=1}^{n_\omega} f(x(t), v^i, u(t))\, \omega_i(t), \\
& 0 & \leq & c(x(t), u(t)), \\
& 0 & \leq & r_i(x(t_0), \ldots, x(t_f)), \\
& 0 & = & r_e(x(t_0), \ldots, x(t_f)), \\
& 1 & = & \sum_{i=1}^{n_\omega} \omega_i(t), \\
& \omega(t) & \in & \{0,1\}^{n_\omega}, \qquad t \in [t_0, t_f].
\end{array}
$$

- Important: lower bounds
- Is relaxation $\omega \in \{0,1\}^{n_\omega} \longrightarrow \alpha \in [0,1]^{n_\omega}$ good?

Approximating the state [S., Bock, Diehl, Mathematical Programming, to appear]

LEMMA. Let $x(\cdot)$ and $y(\cdot)$ be solutions of

$$\begin{aligned} \dot{x}(t) &= A(t, x(t)) \cdot \alpha(t), \quad x(0) = x_0, \\ \dot{y}(t) &= A(t, y(t)) \cdot \omega(t), \quad y(0) = y_0, \end{aligned}$$

with $t \in [0, t_\mathrm{f}]$, for given measurable functions $\alpha, \omega : [0, t_\mathrm{f}] \to [0, 1]^{n_\omega}$ and $A : \mathbb{R}^{n_x+1} \mapsto \mathbb{R}^{n_x \times n_\omega}$.

MathOpt Heidelberg

Approximating the state [S., Bock, Diehl, Mathematical Programming, to appear]

LEMMA. Let $x(\cdot)$ and $y(\cdot)$ be solutions of

$$\begin{aligned} \dot{x}(t) &= A(t, x(t)) \cdot \alpha(t), \quad x(0) = x_0, \\ \dot{y}(t) &= A(t, y(t)) \cdot \omega(t), \quad y(0) = y_0, \end{aligned}$$

with $t \in [0, t_\mathrm{f}]$, for given measurable functions $\alpha, \omega : [0, t_\mathrm{f}] \to [0, 1]^{n_\omega}$ and $A : \mathbb{R}^{n_x+1} \mapsto \mathbb{R}^{n_x \times n_\omega}$. If $C, L, M \in \mathbb{R}^+$ exist such that

$$\begin{aligned} \left\| \frac{\mathrm{d}}{\mathrm{d}t} A(t, x(t)) \right\| &\leq C, \\ \| A(t, y(t)) - A(t, x(t)) \| &\leq L \| y(t) - x(t) \|, \end{aligned}$$

for $t \in [0, t_\mathrm{f}]$ almost everywhere and $A(\cdot, x(\cdot))$ essentially bounded by M on $[0, t_\mathrm{f}]$,

MathOpt Heidelberg

iwr

Approximating the state [S., Bock, Diehl, Mathematical Programming, to appear]

LEMMA. Let $x(\cdot)$ and $y(\cdot)$ be solutions of

$$\begin{aligned}\dot{x}(t) &= A(t, x(t)) \cdot \alpha(t), \quad x(0) = x_0, \\ \dot{y}(t) &= A(t, y(t)) \cdot \omega(t), \quad y(0) = y_0,\end{aligned}$$

with $t \in [0, t_\mathrm{f}]$, for given measurable functions $\alpha, \omega : [0, t_\mathrm{f}] \to [0, 1]^{n_\omega}$ and $A : \mathbb{R}^{n_x+1} \mapsto \mathbb{R}^{n_x \times n_\omega}$. If $C, L, M \in \mathbb{R}^+$ exist such that

$$\begin{aligned}\left\| \frac{\mathrm{d}}{\mathrm{d}t} A(t, x(t)) \right\| &\leq C, \\ \| A(t, y(t)) - A(t, x(t)) \| &\leq L \, \| y(t) - x(t) \|,\end{aligned}$$

for $t \in [0, t_\mathrm{f}]$ almost everywhere and $A(\cdot, x(\cdot))$ essentially bounded by M on $[0, t_\mathrm{f}]$, and it exists $\epsilon \in \mathbb{R}^+$ such that

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\| \leq \epsilon$$

for all $t \in [0, t_\mathrm{f}]$,

Approximating the state [S., Bock, Diehl, Mathematical Programming, to appear]

LEMMA. Let $x(\cdot)$ and $y(\cdot)$ be solutions of

$$\begin{aligned} \dot{x}(t) &= A(t, x(t)) \cdot \alpha(t), \quad x(0) = x_0, \\ \dot{y}(t) &= A(t, y(t)) \cdot \omega(t), \quad y(0) = y_0, \end{aligned}$$

with $t \in [0, t_f]$, for given measurable functions $\alpha, \omega : [0, t_f] \to [0, 1]^{n_\omega}$ and $A : \mathbb{R}^{n_x+1} \mapsto \mathbb{R}^{n_x \times n_\omega}$. If $C, L, M \in \mathbb{R}^+$ exist such that

$$\begin{aligned} \left\| \frac{\mathrm{d}}{\mathrm{d}t} A(t, x(t)) \right\| &\leq C, \\ \| A(t, y(t)) - A(t, x(t)) \| &\leq L \| y(t) - x(t) \|, \end{aligned}$$

for $t \in [0, t_f]$ almost everywhere and $A(\cdot, x(\cdot))$ essentially bounded by M on $[0, t_f]$, and it exists $\epsilon \in \mathbb{R}^+$ such that

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\| \leq \epsilon$$

for all $t \in [0, t_f]$, then for all $t \in [0, t_f]$ it holds

$$\| y(t) - x(t) \| \leq (\| x_0 - y_0 \| + (M + C) t \epsilon) \, e^{Lt}$$

MathOpt Heidelberg

iwr

Approximating the state [S., Bock, Diehl, Mathematical Programming, to appear]

LEMMA. Let $x(\cdot)$ and $y(\cdot)$ be solutions of

$$\begin{aligned} \dot{x}(t) &= A(t, x(t)) \cdot \alpha(t), \quad x(0) = x_0, \\ \dot{y}(t) &= A(t, y(t)) \cdot \omega(t), \quad y(0) = y_0, \end{aligned}$$

with $t \in [0, t_\mathrm{f}]$, for given measurable functions $\alpha, \omega : [0, t_\mathrm{f}] \to [0, 1]^{n_\omega}$ and $A : \mathbb{R}^{n_x+1} \mapsto \mathbb{R}^{n_x \times n_\omega}$. If $C, L, M \in \mathbb{R}^+$ exist such that

$$\begin{aligned} \left\| \frac{\mathrm{d}}{\mathrm{d}t} A(t, x(t)) \right\| &\leq C, \\ \| A(t, y(t)) - A(t, x(t)) \| &\leq L \| y(t) - x(t) \|, \end{aligned}$$

for $t \in [0, t_\mathrm{f}]$ almost everywhere and $A(\cdot, x(\cdot))$ essentially bounded by M on $[0, t_\mathrm{f}]$, and it exists $\epsilon \in \mathbb{R}^+$ such that

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\| \leq \epsilon$$

for all $t \in [0, t_\mathrm{f}]$, then for all $t \in [0, t_\mathrm{f}]$ it holds

$$\| y(t) - x(t) \| \leq (\| x_0 - y_0 \| + (M + C) t \epsilon) \, e^{Lt}$$

MathOpt Heidelberg

Related results: [Gonzalez, Häckl, Pietrus, Tidball, Veliov]

Approximating the state [S., Bock, Diehl, Mathematical Programming, to appear]

In other words:

$$\| \, y(t) - x(t) \, \|$$

will be small for all t, if

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\|$$

is small for all times $t \in [0, t_\mathrm{f}]$.

How to get binary variables $\omega(\tau)$
from continuous variables $\alpha(\cdot)$?

Sum Up Rounding [Sager, 2005]

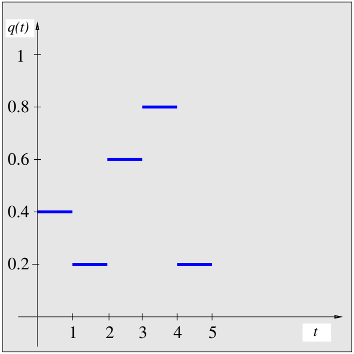
Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} .$$

Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

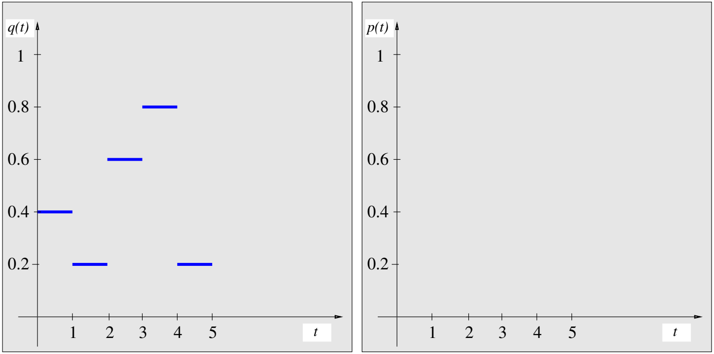
$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} .$$



Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega$, $i = 1 \dots n_{int}$ set

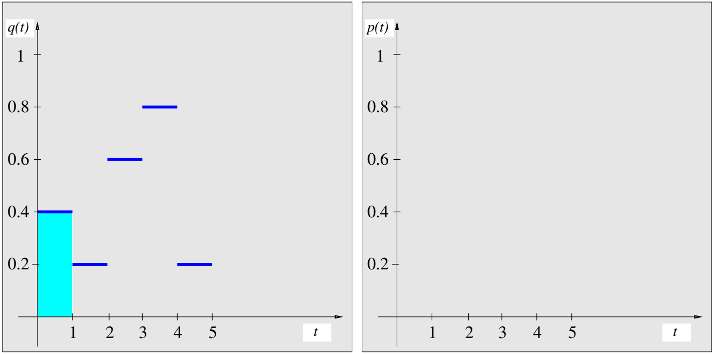
$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} .$$



Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

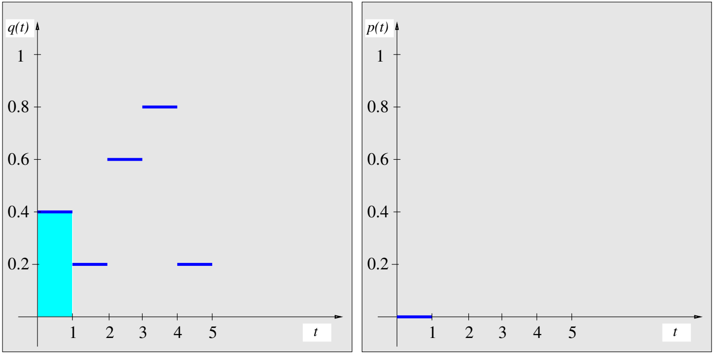
$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k}\Delta t_k - \sum_{k=0}^{i-1} p_{j,k}\Delta t_k \geq 0.5\Delta t_i \\ 0 & \text{else} \end{cases} .$$



Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} .$$

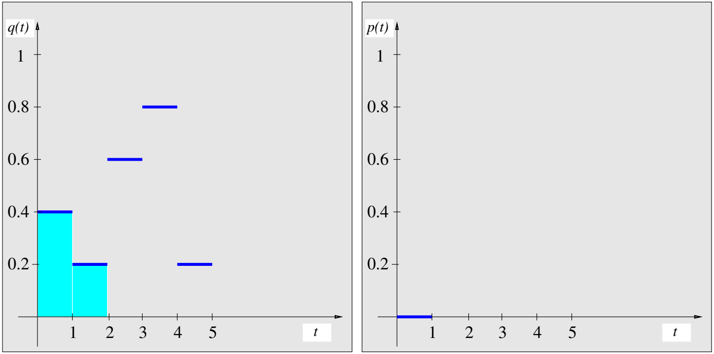


MathOpt Heidelberg

Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega$, $i = 1 \dots n_{int}$ set

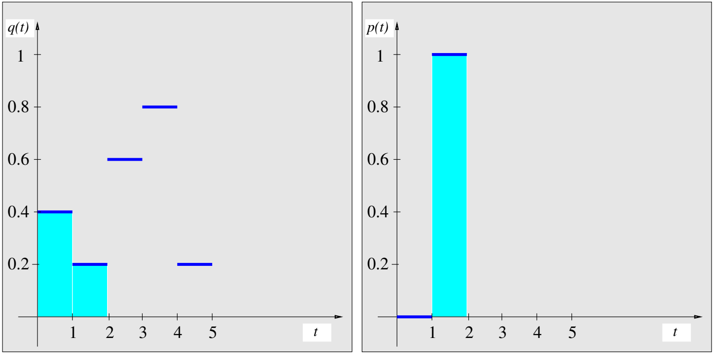
$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} .$$



Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega$, $i = 1 \dots n_{int}$ set

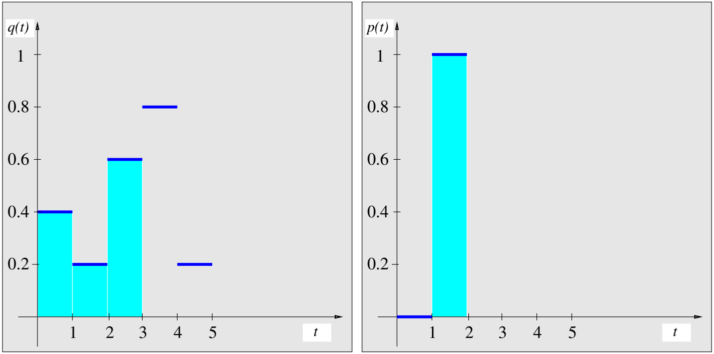
$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} .$$



Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

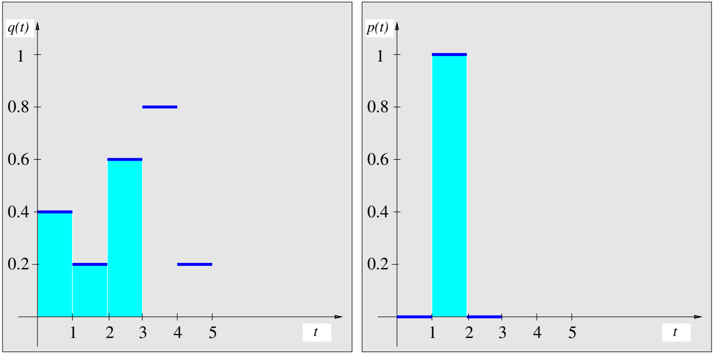
$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k}\Delta t_k - \sum_{k=0}^{i-1} p_{j,k}\Delta t_k \geq 0.5\Delta t_i \\ 0 & \text{else} \end{cases} .$$



Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k}\Delta t_k - \sum_{k=0}^{i-1} p_{j,k}\Delta t_k \geq 0.5\Delta t_i \\ 0 & \text{else} \end{cases} .$$



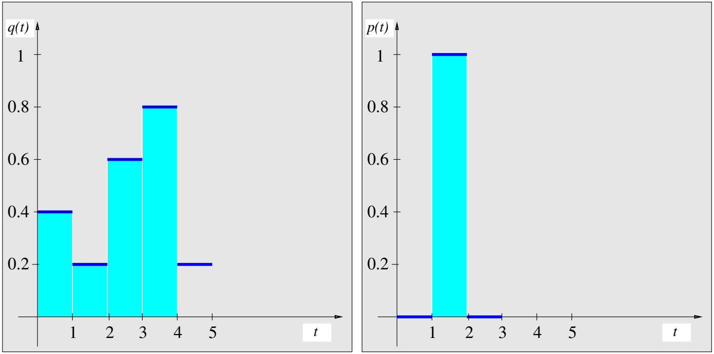
MathOpt Heidelberg

iwr

Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k}\Delta t_k - \sum_{k=0}^{i-1} p_{j,k}\Delta t_k \geq 0.5\Delta t_i \\ 0 & \text{else} \end{cases}.$$



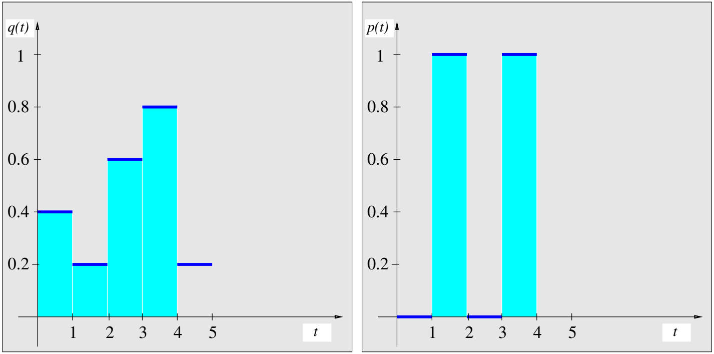
MathOpt Heidelberg

iwr

Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k}\Delta t_k - \sum_{k=0}^{i-1} p_{j,k}\Delta t_k \geq 0.5\Delta t_i \\ 0 & \text{else} \end{cases} .$$



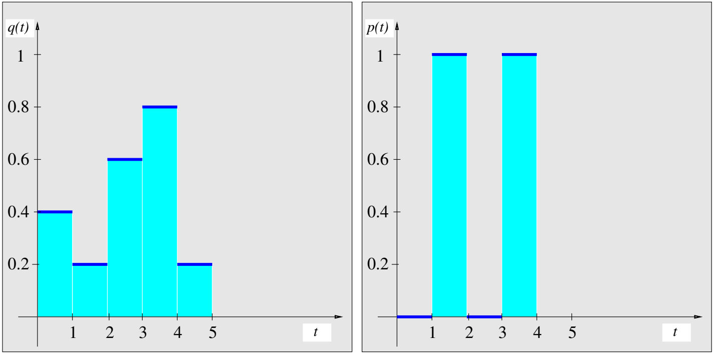
MathOpt Heidelberg

iwr

Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega$, $i = 1 \dots n_{int}$ set

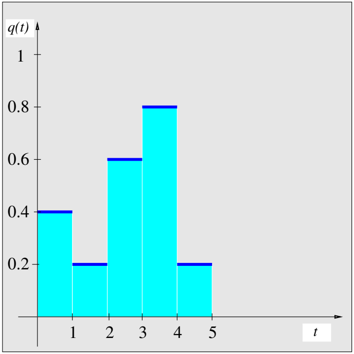
$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} .$$

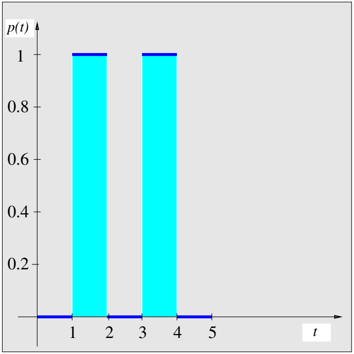


Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} .$$

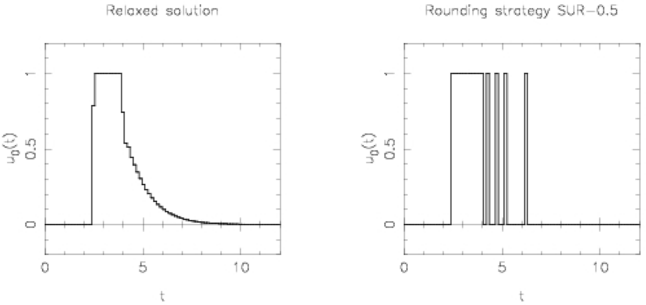




Sum Up Rounding [Sager, 2005]

Assume continuous solution $q_{j,i} \in [0,1]$. Goal: $p_{j,i} \in \{0,1\}$.
For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_\omega, i = 1 \dots n_{int}$ set

$$p_{j,i} = \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} .$$



Approximating the control [S., Bock, Diehl, Mathematical Programming, to appear]

LEMMA. Let functions $\alpha : [0, t_\mathrm{f}] \mapsto [0, 1]^{n_\omega}$,

$$\alpha_j(t) = q_{j,i}, \quad t \in [t_i, t_{i+1}]$$

and $\omega : [0, t_\mathrm{f}] \mapsto \{0, 1\}^{n_\omega}$,

$$\begin{aligned} \omega_j(t) &= p_{j,i}, \quad t \in [t_i, t_{i+1}] \\ p_{j,i} &= \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} . \end{aligned}$$

be given. Then it holds

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\| \leq 0.5 \max_i \Delta t_i$$

(Similar strategy and lemma allow to include SOS1 constraints for nonlinear case)

Implications

- 2 lemmata together: if SUR is used then

$$\| y(t) - x(t) \| \leq C\Delta t$$

  for any (relaxed) solution $x(\cdot)$, control grid size Δt
- Objective function is a continuous function, thus

$$|\Phi(x(t_\mathrm{f})) - \Phi(y(t_\mathrm{f}))| \leq \gamma\Delta t$$

MathOpt Heidelberg

iwr

Implications

- 2 lemmata together: if SUR is used then

$$\| y(t) - x(t) \| \leq C\Delta t$$

for any (relaxed) solution $x(\cdot)$, control grid size Δt
- Objective function is a continuous function, thus

$$|\Phi(x(t_\mathrm{f})) - \Phi(y(t_\mathrm{f}))| \leq \gamma\Delta t$$

- Extension to continuous constraints straightforward

Implications

- 2 lemmata together: if SUR is used then

$$\| \, y(t) - x(t) \, \| \leq C\Delta t$$

  for any (relaxed) solution $x(\cdot)$, control grid size Δt
- Objective function is a continuous function, thus

$$|\Phi(x(t_f)) - \Phi(y(t_f))| \leq \gamma \Delta t$$

- Extension to continuous constraints straightforward
- We know the exact lower bound from $\alpha(\cdot)$ solution
- Sum Up Rounding constructive way to get integer solution

MS MINTOC algorithm [Sager 2005; Sager, J Proc Control, 2009]

1. $k = 0$. Input: control discretization grid $\mathcal{G}^0$, $TOL \in \mathbb{R}^+$.
2. If necessary, reformulate and convexify problem. Relax this problem to $\alpha(\cdot) \in [0,1]^{n_\omega}$.
3. REPEAT
   - 3.1 Solve relaxed problem on $\mathcal{G}^k$. Obtain $\mathcal{T}^k = (x^k(\cdot), u^k(\cdot), \alpha^k(\cdot))$ and the grid–dependent optimal value ϕ^{RC}.
   - 3.2 If $\mathcal{T}^k$ on $\mathcal{G}^k$ fulfills $\omega^k(\cdot) := \alpha^k(\cdot) \in \{0,1\}^{n_\omega}$ then STOP.
   - 3.3 Apply Sum Up Rounding to $\alpha^k(\cdot)$. Fix $u^k(\cdot)$. Obtain $y^k(\cdot)$ and upper bound ϕ^{STO} by simulation.
   - 3.4 If $\phi^{\text{STO}} < \phi^{\text{RC}} + TOL$ then STOP.
   - 3.5 Refine the control grid $\mathcal{G}^k$.
   - 3.6 $k = k + 1$.
4. Bijection to obtain solution $\Phi^* = \phi^{\text{STO}}$ for original problem.

Algorithm terminates in finite number of iterations [S., 2005; S. 2009]

Intermediate summary

- Main trick: binary controls propagated to continuous states
- Most applications: no explicit constraints on binary controls

Intermediate summary

- Main trick: binary controls propagated to continuous states
- Most applications: no explicit constraints on binary controls
- At the price of chattering, relaxed solution can always be reached $\rightarrow$ no integer gap!!!
- However: for any fixed control grid there may be a gap
  $\rightarrow$ black box MINLP performs badly!
- Sum Up Rounding constructive way to get integer solution
- Needs to be combined with iterative refinement of grid

- Is Outer Convexification crucial?

MathOpt Heidelberg

Example: MIOCP nonlinear in v

$$
\begin{aligned}
\min_{x,v} \quad & x_2(t_\mathrm{f}) \\
\text{s.t.} \quad & \dot{x}_0 = -\frac{x_0 \ \sin(v_1)}{\sin(1)} + (x_0 + x_1)\ {v_2}^2 + (x_0 - x_1)\ {v_3}^3 \\
& \dot{x}_1 = (x_0 + 2x_1)\ v_1 + (x_0 - 2x_1)\ v_2 + (x_0 + x_1)\ v_3 \\
& \qquad + (x_0 x_1 - x_2)\ {v_2}^2 - (x_0 x_1 - x_2)\ {v_2}^3, \qquad (1) \\
& \dot{x}_2 = x_0^2 + x_1^2, \\
& x(0) = (0.5, 0.5, 0)^T, \\
& x_1 \geq 0.4, \\
& v \in \{(1,0,0), (0,1,0), (0,0,1)\}
\end{aligned}
$$

with $t \in [t_0, t_\mathrm{f}] = [0, 1]$. This problem can be relaxed by requiring

$$
v_1, v_2, v_3 \in [0, 1], \quad \sum_{i=1}^{3} v_i = 1
$$

Example: partially outer convexified with ω

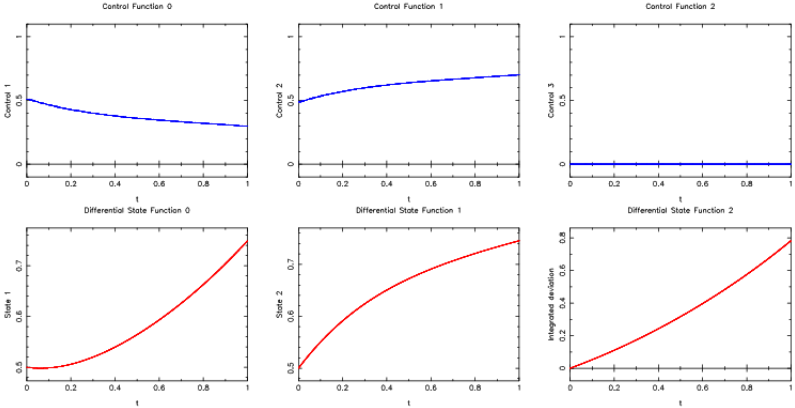
$$
\begin{aligned}
\min_{x,\omega} \quad & x_2(t_\mathrm{f}) \\
\text{s.t.} \quad & \dot{x}_0 = -x_0\,\omega_1 + (x_0 + x_1)\,\omega_2 + (x_0 - x_1)\,\omega_3, \\
& \dot{x}_1 = (x_0 + 2x_1)\,\omega_1 + (x_0 - 2x_1)\,\omega_2 + (x_0 + x_1)\,\omega_3, \\
& \dot{x}_2 = x_0^2 + x_1^2, \\
& x(0) = (0.5, 0.5, 0)^T, \\
& x_1 \geq 0.4, \\
& \omega_i \in \{0,1\}, \quad \sum_{i=1}^{3} \omega_i = 1
\end{aligned}
\tag{2}
$$

with $t \in [t_0, t_\mathrm{f}] = [0, 1]$. Relaxation: $\alpha_i \in [0, 1]$

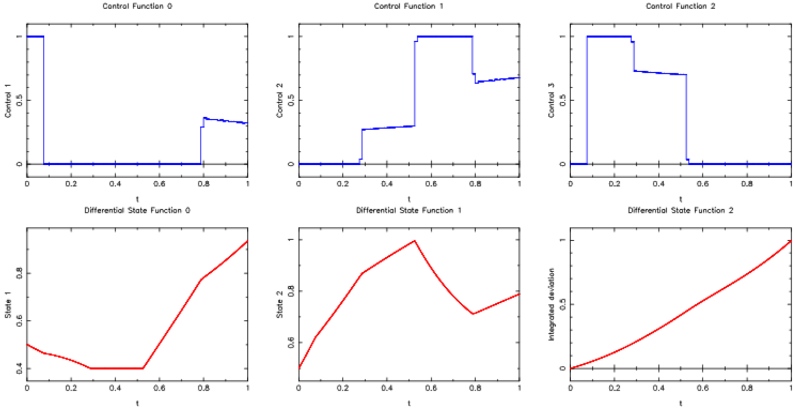
Is (by construction) identical to the one in (?) and (?).

MathOpt Heidelberg

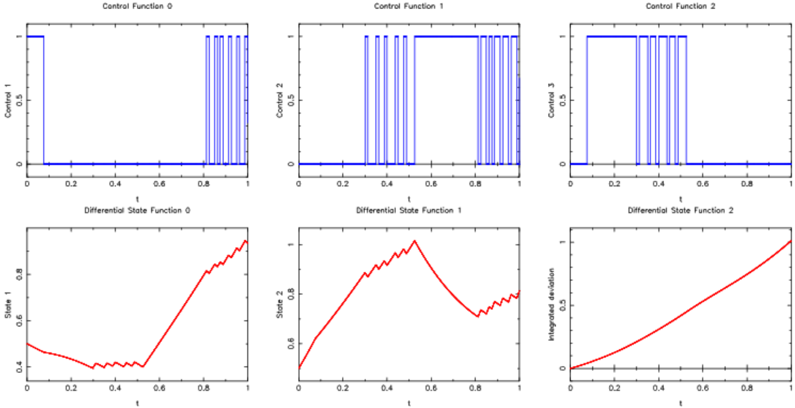
OCP (1) - nonlinear relaxed: $\Phi = 0.782$



OCP (2) - outer convexification relaxed: $\Phi = 0.996$



MIOCPs (1) and (2) - Sum Up Rounding: $\Phi = 1.012$



Elchtest benchmark problem [Gerdts, OCAM, 2005]

Elchtest in min. time with gear shifts, *N* discretization points.

N	t_f	CPU Time	t_f	CPU Time
20	6.779751	00:23:52	6.779035	00:00:24
40	6.786781	232:25:31	6.786730	00:00:46
80	–	–	6.789513	00:04:19

Left: Branch and Bound, Inner Convexification [Gerdts, OCAM, 2005]
Pentium III with 750 MHz

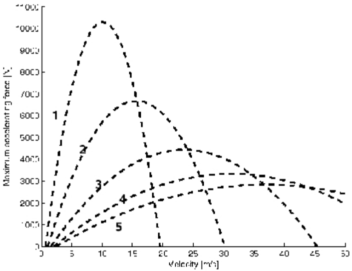
Right: MS MINTOC, Outer Convexific. [Kirches, Bock, Schlöder, S., OCAM, 2009]
AMD Athlon XP 3000+ with 2166 MHz

Similar speedup for free switching times [Gerdts OCAM 2006, Kirches et al. 2009]

Why is this formulation so beneficial?!?

$$\dot{v}(t) = \frac{1}{m}\Big(\big(F_{\mathrm{lr}}^{\mu} - F_{\mathrm{Ax}}\big)\ \cos\beta(t) + F_{\mathrm{lf}}\ \cos\big(\delta(t) + \beta(t)\big) \\ - \big(F_{\mathrm{sr}} - F_{\mathrm{Ay}}\big)\ \sin\beta(t) - F_{\mathrm{sf}}\ \sin\big(\delta(t) + \beta(t)\big)\Big).$$

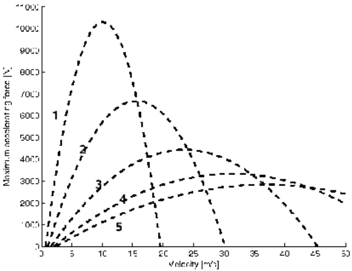
F_{lr}^{μ} is a function of ϕ, F_{B}, v, and i_{g}^{μ}. Maximum acceleration $\to \phi := 1, F_{\mathrm{B}} := 0$.

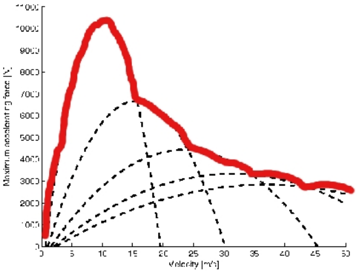


Why is this formulation so beneficial?!?

$$\dot{v}(t) = \frac{1}{m}\Big(\big(F_{\text{lr}}^{\mu} - F_{\text{Ax}}\big)\ \cos\beta(t) + F_{\text{lf}}\ \cos\big(\delta(t)+\beta(t)\big)$$
$$- \big(F_{\text{sr}} - F_{\text{Ay}}\big)\ \sin\beta(t) - F_{\text{sf}}\ \sin\big(\delta(t)+\beta(t)\big)\Big).$$

F_{lr}^{μ} is a function of ϕ, F_{B}, v, and i_{g}^{μ}. Maximum acceleration $\rightarrow \phi := 1, F_{\text{B}} := 0$.



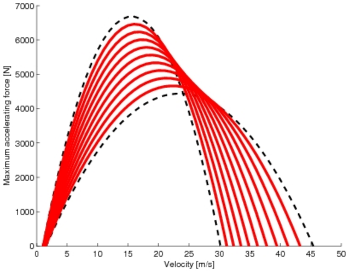


Compare maximum acceleration

Compare maximum acceleration for convex combinations, $a = 0.1 \ldots 0.9$

Inner convexification, use
$F_{\text{lr}} = F_{\text{lr}}(a \cdot i_{\text{g}}^2 + (1-a) \cdot i_{\text{g}}^3)$

where i_{g}^2 and i_{g}^3 are the transmission ratios of second resp. third gear



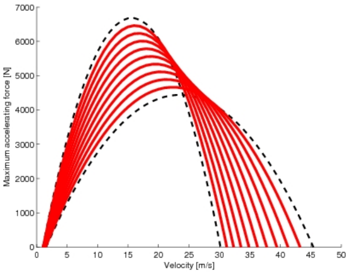
Compare maximum acceleration

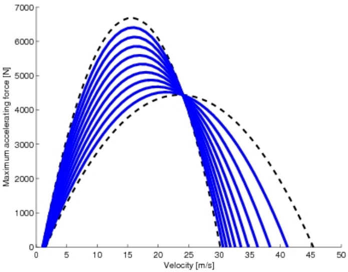
Compare maximum acceleration for convex combinations, $a = 0.1 \ldots 0.9$

Inner convexification, use
$F_{\mathrm{lr}} = F_{\mathrm{lr}}(a \cdot i_g^2 + (1-a) \cdot i_g^3)$

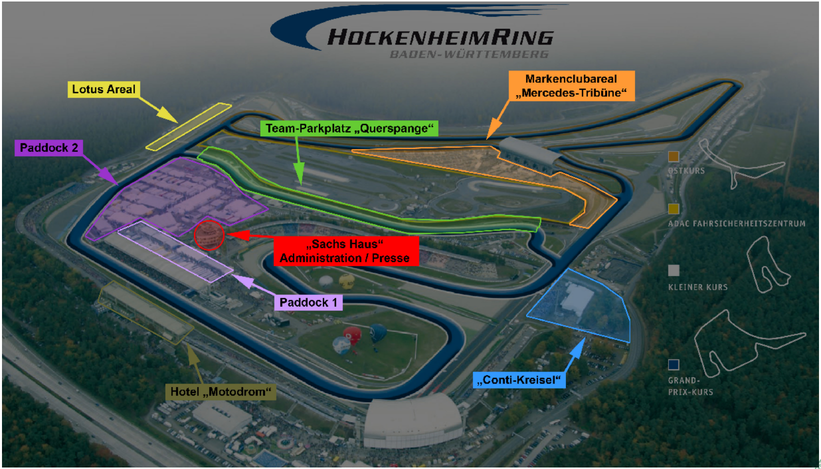
Outer convexification, use
$F_{\mathrm{lr}} = a \cdot F_{\mathrm{lr}}(i_g^2) + (1-a) \cdot F_{\mathrm{lr}}(i_g^3)$

where i_g^2 and i_g^3 are the transmission ratios of second resp. third gear



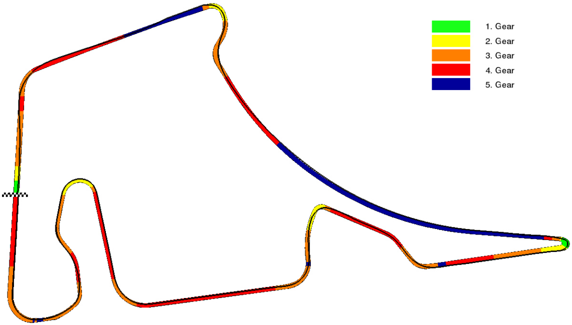


Diploma thesis F. Kehrle



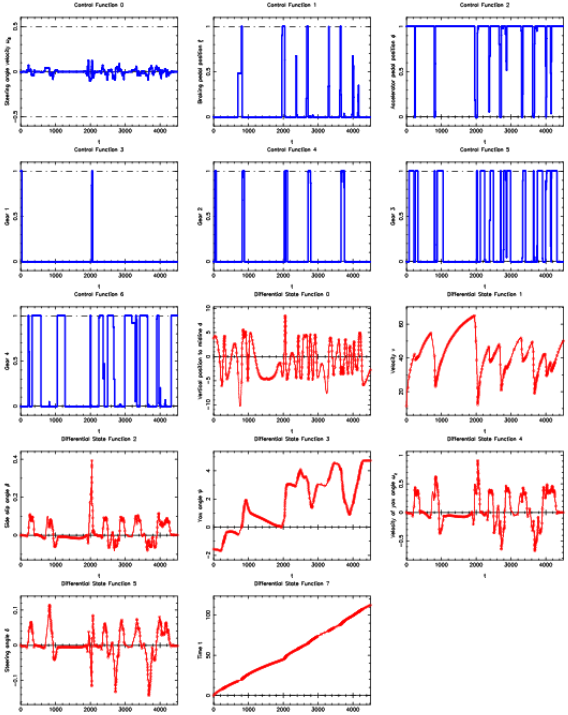
Problem characteristics

- Long optimization horizon, unstable $\Longrightarrow$ use Bock's direct multiple shooting method, 350 intervals
- $350 \cdot 5 =$ 1750 binary control variables
- $350 \cdot 3 =$ 1050 continuous control variables
- $\approx 7 \cdot 4300 \Rightarrow \approx$ 30.000 of discretized state variables
- Nonlinear dynamics, nasty engine speed + path constraints



MathOpt Heidelberg

NMPO iwr


Control Function 0
Control Function 1
Control Function 2
Control Function 3
Control Function 4
Control Function 5
Control Function 6
Differential State Function 0
Differential State Function 1
Differential State Function 2
Differential State Function 3
Differential State Function 4
Differential State Function 5
Differential State Function 7

Outline

MIOC introduction

Methods for Mixed-Integer Optimal Control

...based on HJB / Dynamic Programming

...based on Direct Methods

MINLP

MS MINTOC

Switching Costs and Combinatorial Constraints

Switching Costs and Combinatorial Constraints

- What if constraints depend explicitly on integer variables?

$$0 \leq c(x(t), v(t), u(t))$$

- Example: switching restrictions / costs on fixed control grid
- Problem: Sum Up Rounding solution infeasible!

Switching Costs and Combinatorial Constraints

- What if constraints depend explicitly on integer variables?

$$0 \leq c(x(t), v(t), u(t))$$

- Example: switching restrictions / costs on fixed control grid
- Problem: Sum Up Rounding solution infeasible!

- One option: use MINLP solver

Switching Costs and Combinatorial Constraints

- What if constraints depend explicitly on integer variables?

$$0 \leq c(x(t), v(t), u(t))$$

- Example: switching restrictions / costs on fixed control grid
- Problem: Sum Up Rounding solution infeasible!

- One option: use MINLP solver
- Alternative: decompose problem into NLP and MILP, making use of approximation theorem!

Combinatorial Integral Approximation [S., Jung, Kirches, MMOR, to appear]

Idea: minimize $\max_t \left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\|_\infty$ with binary controls $\omega(\cdot) \approx p$ that fulfill combinatorial constraints.

Combinatorial Integral Approximation [S., Jung, Kirches, MMOR, to appear]

Idea: minimize $\max_t \left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\|_\infty$ with
binary controls $\omega(\cdot) \approx p$ that fulfill combinatorial constraints.

$$\begin{aligned}
&\min_{p} \quad \max_{k=1..n_\omega} \max_{i=1..n_t} \left| \sum_{j=1}^{i} (q_{k,j} - p_{k,j}) \Delta t_j \right| \\
&\text{subject to} \\
&\quad \sigma_{k,\max} \geq \sum_{i=1}^{n_t - 1} |p_{k,i} - p_{k,i+1}|, \quad k = 1..n_\omega, \\
&\quad p_{k,i} \in \{0, 1\}, \quad k = 1..n_\omega, \; i = 1..n_t.
\end{aligned} \tag{3}$$

Given are: control discretization grid Δt_j,
maximum numbers of switches $\sigma_{k,\max}$, and relaxed solution $\alpha(\cdot) \approx q$

Combinatorial Integral Approximation [S., Jung, Kirches, MMOR, to appear]

- MINLP solution (Bonmin) $>>$ 1800 sec
- NLP + MILP solution (CPlex) $\approx$ 360 sec

Combinatorial Integral Approximation [S., Jung, Kirches, MMOR, to appear]

- MINLP solution (Bonmin) $>>$ 1800 sec
- NLP + MILP solution (CPlex) $\approx$ 360 sec
- NLP + MILP solution (own code) $\approx$ 1.5 sec

How do we get this MILP solution so fast?

Cutting Planes?

- Number of facets of feasible polytope [PORTA – Christof, Löbel, Reinelt]

n_t	Control values $q_{1,j}$ fixed to: 0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
4	16	18	23	33	29	33	23	18	16
5	21	31	87	189	54	189	87	31	21
6	30	60	745	612	248	612	745	60	30
7	47	150	4838	4840	922	4840	4838	150	47
8	83	899	37470	29884	4212	29884	37470	899	83

- depends heavily on relaxed solution q
- Hence cutting planes of limited use

MathOpt Heidelberg

Cutting Planes?

- Number of facets of feasible polytope [PORTA – Christof, Löbel, Reinelt]

n_t	Control values $q_{1,j}$ fixed to: 0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
4	16	18	23	33	29	33	23	18	16
5	21	31	87	189	54	189	87	31	21
6	30	60	745	612	248	612	745	60	30
7	47	150	4838	4840	922	4840	4838	150	47
8	83	899	37470	29884	4212	29884	37470	899	83

- depends heavily on relaxed solution q
- Hence cutting planes of limited use
- Instead use Branch&Bound without LP relaxations, making use of specific structure of inequalities

MathOpt Heidelberg

iwr

Algorithm 1: Combinatorial Branch and Bound

```
Input  : Relaxed controls q, time grid {t_i}, i = 1..n_t, max. numbers of switches
         σ_{k,max}, k = 1..n_ω.
Result : Optimal solution (η*, s*, p*) of (3).
begin
    Create empty priority queue Q ordered by a.η (non-decreasing), if equal by a.d
    (non-increasing).
    Push an empty node (0, {}, {}, 0.0) into the queue.
    while Q is not empty do
        a = top node of Q and remove the node from Q.
        /* 1st solution found is optimal since best-first search
           is used                                                  */
        if a.d = n_t then
            Return optimal solution (a.η, a.s, a.p).
        /* Create child nodes, use strong branching.               */
        else
            forall possible permutations φ of {0,1}^{n_ω} do
                Create new node n with n.d = d + 1, n.p = a.p, n.s = a.s.
                Set n.p_{k,d+1} = φ_k, calculate n.s_{k,d+1}.
                if n.s fulfills switching constraint until time d + 1 then
                    n.η = max{ a.η, max_{k=1}^{n_ω} {± Σ_{j=1}^{d+1} (q_{k,j} − p_{k,j}) Δt_j} }
                    Push n into Q.
end
```

Where the listing uses plain-text math:

- Input: q, $\{t_i\}$, $i = 1..n_t$, $\sigma_{k,\max}$, $k = 1..n_\omega$
- Result: (η^*, s^*, p^*)
- $n.\eta = \max\left\{a.\eta, \max_{k=1}^{n_\omega}\{\pm \sum_{j=1}^{d+1}(q_{k,j} - p_{k,j})\Delta t_j\}\right\}$

Combinatorial Integral Approximation [S., Jung, Kirches, MMOR, to appear]

THEOREM. Assume p^{SUR} to be the solution obtained by Sum Up Rounding. The following claims hold for the optimal solution (η^*, s^*, p^*) of the MILP (3):

$$
\begin{aligned}
& (a)\ \eta^* < 0.5\ \delta t = 0.5 \min_{i=1..n_t} \Delta t_i \\
\Rightarrow\quad & (b)\ p^* = p^{\text{SUR}} \\
\Rightarrow\quad & (c)\ \sigma_{k,\max} \geq \sigma_k^{\text{SUR}} \quad \forall\ k = 1..n_\omega \\
\Rightarrow\quad & (d)\ \eta^* \leq 0.5\ \Delta t = 0.5 \max_{i=1..n_t} \Delta t_i
\end{aligned}
$$

where the solution $p^* = p^{\text{SUR}}$ in (b) is unique.

MathOpt Heidelberg

iwr

Combinatorial Integral Approximation [S., Jung, Kirches, MMOR, to appear]

THEOREM. Assume p^{SUR} to be the solution obtained by Sum Up Rounding. The following claims hold for the optimal solution (η^*, s^*, p^*) of the MILP (3):

$$
\begin{aligned}
& (a)\ \eta^* < 0.5\ \delta t = 0.5 \min_{i=1..n_t} \Delta t_i \\
\Rightarrow \quad & (b)\ p^* = p^{\text{SUR}} \\
\Rightarrow \quad & (c)\ \sigma_{k,\max} \geq \sigma_k^{\text{SUR}} \quad \forall\ k = 1..n_\omega \\
\Rightarrow \quad & (d)\ \eta^* \leq 0.5\ \Delta t = 0.5 \max_{i=1..n_t} \Delta t_i
\end{aligned}
$$

where the solution $p^* = p^{\text{SUR}}$ in (b) is unique.

Hence: MILP solution approximates MINLP solution when $n_t \to \infty$ and if $\sigma_{k,\max}$ are large enough

CIA summary

- replace 1 MINLP by 1 NLP + 1 MILP
- generic approach (not only for switching constraint)
- for given grid Δt the solutions may be different,
  but asymptotic convergence as $\Delta t \to 0$
- MILP solution can be used as fast UB in MINLP
- MILP solution can be used to initialize Switching Time Optimization
- MILP structure (e.g., max switching) may allow for ultrafast strategies

MathOpt
Heidelberg

Conclusions

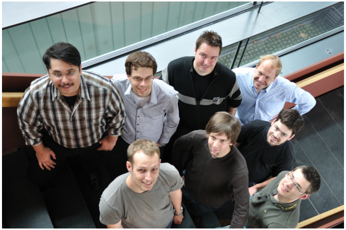
- Many applications (with optimization potential) are MIOCPs
- Open benchmark library: http://mintoc.de

- Many different approaches compete
- Speaker's personal favorite: MS MINTOC
  - exact bounds and error estimates
  - can be combined with state-of-the-art numerics
  - extensions towards NMPC, global, robust, multi-objective, DAE, PDE, combinatorial constraints, . . . are possible

- Survey article [S. Sager. Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control. Journal of Process Control, 2009.]
- Literature: http://mathopt.de

Acknowledgements

- Hans Georg Bock
- Moritz Diehl
- Matthias Gerdts
- Johannes P. Schlöder
- Graduate School MathComp Heidelberg
- MathOpt group



Thank you for your attention!

Questions?

MathOpt Heidelberg