Reformulations and Numerical Methods for Control Problems with Integer-valued Functions

Sebastian Sager

joint work with C. Kirches, J.P. Schlöder, H.G. Bock
Interdisciplinary Center for Scientific Computing, University of Heidelberg,
and with M. Diehl, OPTEC, K.U. Leuven

May 14, 2008

Hybrid Systems Workshop, Istanbul

Outline

Outline

Motivation, problem definition

Discretization of MIOCPs in ODE

Reformulations

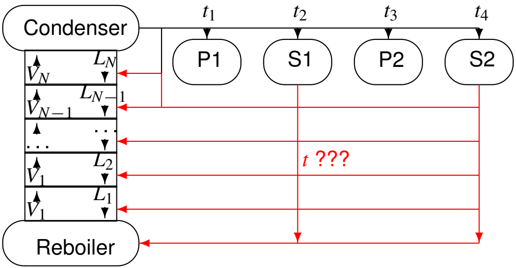
MS MINTOC algorithm

Applications, summary

Motivation: Switched systems

Dynamic processes with discrete (time–dependent) decisions.
Often multi-scale modeling of fast transient behavior.

- Valves (open or closed)
- Gears or disjoint operation modes
- On-off devices (engines, generators, ...)



Problem class

- Optimal control problem arising from a complex dynamic process, e.g., in engineering, economics, cell biology, ...

Problem class

- Optimal control problem arising from a complex dynamic process, e.g., in engineering, economics, cell biology, ...

$$
\begin{array}{llcl}
\min_{x,u,p,t_f} & \phi(t_f, x(t_f), p) & & \\
\text{s.t.} & \dot{x}(t) & = & f(t, x(t), u(t), p), \\
 & 0 & \leq & c(t, x(t), u(t), p), \\
 & & & \\
 & 0 & \leq & r_i(x(t_0), \ldots, x(t_f), p), \\
 & 0 & = & r_e(x(t_0), \ldots, x(t_f), p)
\end{array}
$$

- $x(\cdot)$ differential states, $u(\cdot)$ control functions
- Minor importance in this talk: p time-independent controls

Problem class

- Optimal control problem arising from a complex dynamic process, e.g., in engineering, economics, cell biology, ...

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

- Additional controls $\omega(t)$ from finite set $\omega^i \in \Omega \subseteq \mathbb{R}^{n_\omega}$.
- Ex. 1: $\omega^1 = (i_T^1, \eta_T^1)^T$, Ex. 2: $\omega^1 = (0,0,1)^T, \omega^2 = (0,1,1)^T$

Direct methods for optimal control

$$
\begin{aligned}
&\min_{x,u,p} \phi(t_f, x(t_f), p) \\
&\text{s.t.} \quad \dot{x}(t) = f(t, x(t), u(t), p), \\
&\qquad\quad 0 \leq c(t, x(t), u(t), p), \\
&\qquad\quad 0 \leq r_{\mathrm{i}}(x(t_0), \ldots, x(t_f), p), \\
&\qquad\quad 0 = r_{\mathrm{e}}(x(t_0), \ldots, x(t_f), p)
\end{aligned}
$$

- Consider the infinite dimensional optimization problem

Direct methods for optimal control

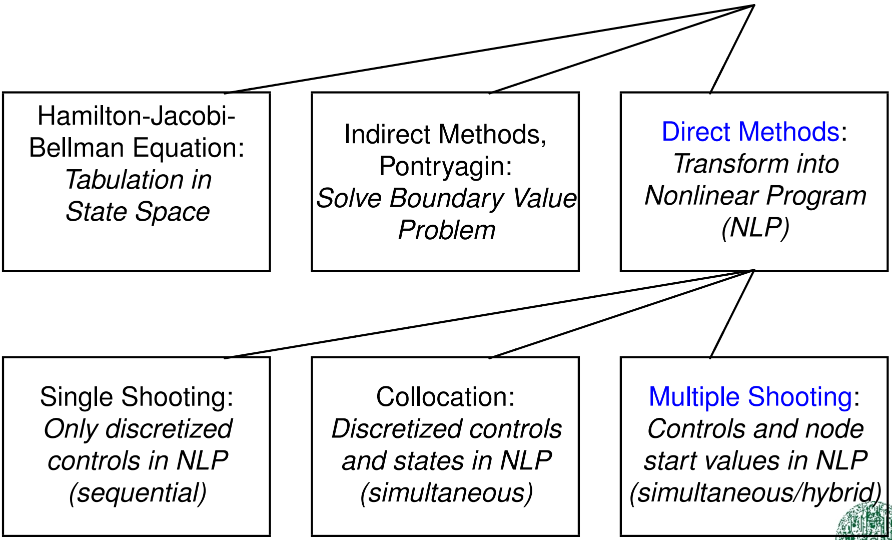
$$
\begin{array}{llll}
\min\limits_{x,u,p} & \phi(t_f, x(t_f), p) & & \\
\text{s.t.} & \dot{x}(t) & = & f(t, x(t), u(t), p), \\
 & 0 & \leq & c(t, x(t), u(t), p), \\
 & & & \\
 & 0 & \leq & r_{\text{i}}(x(t_0), \ldots, x(t_f), p), \\
 & 0 & = & r_{\text{e}}(x(t_0), \ldots, x(t_f), p)
\end{array}
$$

- Consider the infinite dimensional optimization problem
- Apply Direct Multiple Shooting

  [Bock et al. 1981 ff.]

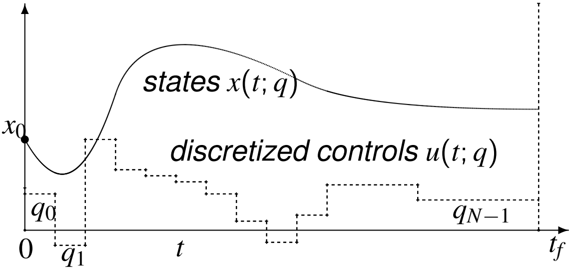
$$
\begin{array}{llll}
\min\limits_{\alpha} & F(\alpha) & & \\
\text{s.t.} & 0 & = & G(\alpha), \\
 & 0 & \leq & H(\alpha)
\end{array}
$$

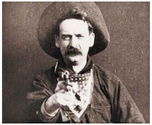
Overview: Optimal Control Family Tree

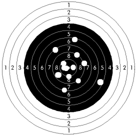


Direct Single Shooting [Hicks, Ray 1971; Sargent, Sullivan 1977]

Discretize controls $u(t)$ on fixed grid $0 = t_0 < t_1 < \ldots < t_N = t_f$.

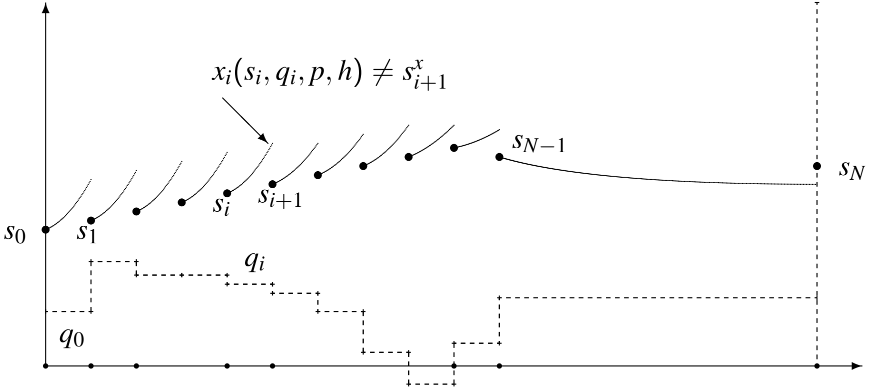






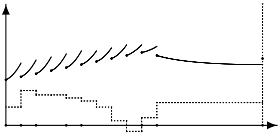
Regard states $x(t)$ on $[t_0, t_f]$ as dependent variables.
Use numerical integration to obtain state as function $x(t; q, x_0)$ of finitely many control parameters $q = (q_0, q_1, \ldots, q_{N-1})$ and the initial value x_0.

Direct Multiple Shooting [Bock et al. 1981 ff.]



Main idea: Decouple intervals and add extra continuity constraints.
Denote each interval's variables by $w_i := (s_i^x, s_i^z, q_i)$.
Summarize all in large vector $w := (w_0, \ldots, w_N)$.

NLP in Direct Multiple Shooting



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

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

Summary

Bock's Direct Multiple Shooting method [Bock et al. 1981 ff.]

- is **simultaneous** optimal control method
- uses **adaptive** integrators, but NLP has **fixed dimensions**
- can treat **nonlinear** and **unstable** systems with **state constraints**
- can use knowledge of x in **initialization** (e.g., in tracking problems), impact on convergence region of Newton type method.
- easy to **parallelize**
- NLP structure exploited by **high rank updates**, **partial reduction technique**, **condensing**, ...
- Replaces Checkpointing in **Adjoint Calculation**

- Implemented in modular software package MUSCOD-II

  [Leineweber, Schäfer, Diehl, Sager, Potschka, Kirches, Albersmeyer, Wirsching, Hoffmann, ...]

Back to the point

Important for the following:

- Direct approach: finitely many optimization variables
- Can calculate functions and derivatives

MINLP approach to solve MIOCPs

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

$$\begin{aligned}
\text{s.t.} \quad \dot{x}(t) &= f(t, x(t), \omega(t), u(t), p), \quad t \in [t_0, t_f], \\
0 &\leq c(t, x(t), \omega(t), u(t), p), \\
0 &\leq r_{\text{i}}(x(t_0), \dots, x(t_f), p), \\
0 &= r_{\text{e}}(x(t_0), \dots, x(t_f), p), \\
\omega(t) &\in \Omega.
\end{aligned}$$

- Consider the infinite dimensional problem
- Direct Multiple Shooting

MINLP approach to solve MIOCPs

$$
\begin{aligned}
\min_{x,\omega,u,p} \quad & \phi(t_f, x(t_f), p) \\
\text{s.t.} \quad \dot{x}(t) &= f(t, x(t), \omega(t), u(t), p), \quad t \in [t_0, t_f], \\
0 &\leq c(t, x(t), \omega(t), u(t), p), \\
0 &\leq r_{\mathrm{i}}(x(t_0), \ldots, x(t_f), p), \\
0 &= r_{\mathrm{e}}(x(t_0), \ldots, x(t_f), p), \\
\omega(t) &\in \Omega.
\end{aligned}
$$

- Consider the infinite dimensional problem
- Direct Multiple Shooting
- Let variables inherit integer constraint

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

Mixed–integer Nonlinear Program

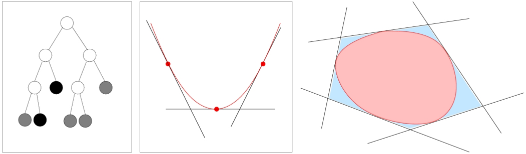
$$\begin{array}{lccl} \min\limits_{\alpha,\beta} & F(\alpha,\beta) & & \\ \text{s.t.} & 0 & = & G(\alpha,\beta), \\ & 0 & \leq & H(\alpha,\beta), \\ & \beta_i & \in & \Omega, i = 1..N \end{array}$$

- Obtain a Mixed–integer Nonlinear Program (MINLP)

Mixed–integer Nonlinear Program

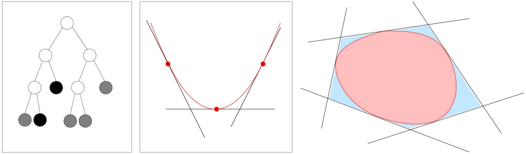
$$
\begin{aligned}
\min_{\alpha,\beta} \quad & F(\alpha, \beta) \\
\text{s.t.} \quad & 0 = G(\alpha, \beta), \\
& 0 \leq H(\alpha, \beta), \\
& \beta_i \in \Omega, i = 1..N
\end{aligned}
$$

- Obtain a Mixed–integer Nonlinear Program (MINLP)
- Apply generic algorithms as Nonlinear Branch & Bound, Outer Approximation, ...



Mixed–integer Nonlinear Programming

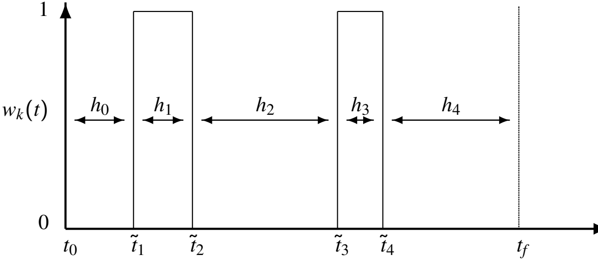
- Apply generic algorithms as Nonlinear Branch & Bound, Outer Approximation, . . .



- Active field of research
  - Biegler, Bonami, Grossmann, Wächter, . . . (Bonmin)
  - Leyffer, Linderoth, . . . (FilMint)
  - . . .
- But: **very** costly, can't we avoid enumeration?

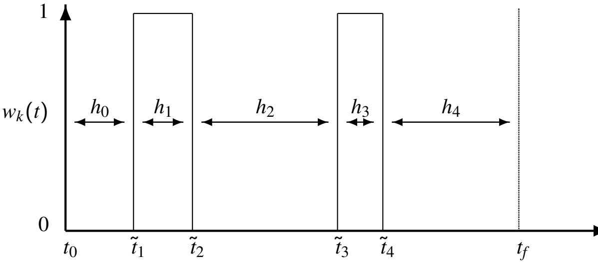
Switching time optimization

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



Switching time optimization

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

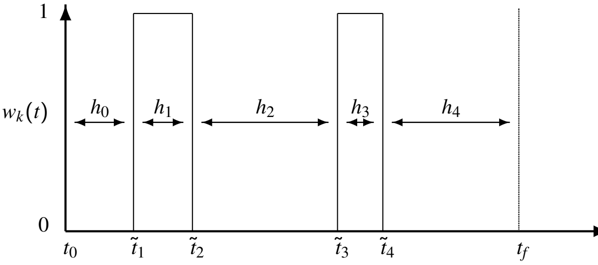


$$\min_{x,\omega,u,p} \phi(t_f, x(t_f), p)$$

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

Switching time optimization

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



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

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

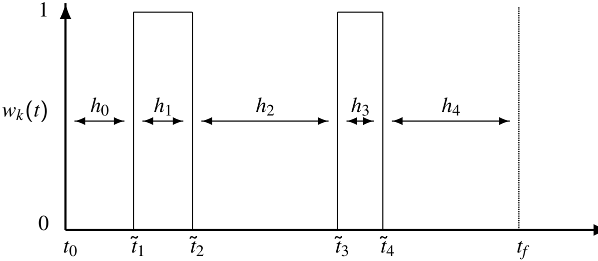
$$\min_{x,h,u,p} \phi(t_f, x(t_f), p)$$

$$\text{s.t.} \quad \dot{x}(t) = f(t, x(t), \omega^{i_j}, u(t), p), \quad t \in [\tilde{t}_j, \tilde{t}_{j+1}]$$

$$\ldots$$

Switching time optimization

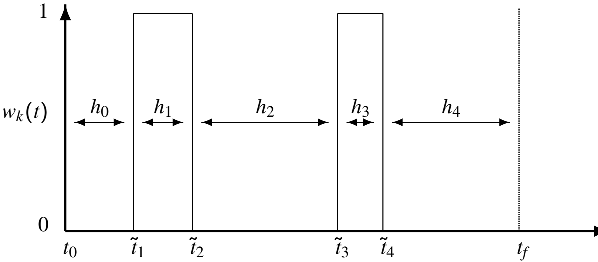
Fixed switching structure with fixed controls, optimize continuous interval lengths (after standard time transformation)



Concept old and well known:

Switching time optimization

Fixed switching structure with fixed controls, optimize continuous interval lengths (after standard time transformation)

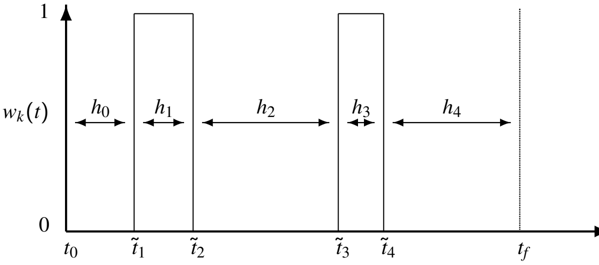


Concept old and well known:

- Indirect approaches, switching function to determine $\tilde{t}_i$

Switching time optimization

Fixed switching structure with fixed controls, optimize continuous interval lengths (after standard time transformation)

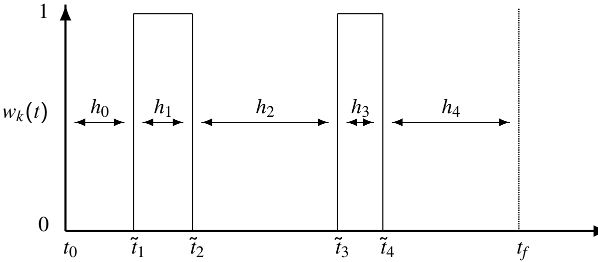


Concept old and well known:

- Indirect approaches, switching function to determine $\tilde{t}_i$
- Hybrid systems, switching function to determine phase transitions

Switching time optimization

Fixed switching structure with fixed controls, optimize continuous interval lengths (after standard time transformation)



Concept old and well known:

- Indirect approaches, switching function to determine $\tilde{t}_i$
- Hybrid systems, switching function to determine phase transitions
- Multi-stage processes: batch processes

Switching Time Optimization

Advantage:

- Only continuous variables $\rightarrow$ very fast

Disadvantages:

Switching Time Optimization

Advantage:

- Only continuous variables → very fast

Disadvantages:

- Still discrete decision (switching structure)

Switching Time Optimization

Advantage:

- Only continuous variables → very fast

Disadvantages:

- Still discrete decision (switching structure)
- Numerical stability when stage lengths become zero

Switching Time Optimization

Advantage:

- Only continuous variables $\rightarrow$ very fast

Disadvantages:

- Still discrete decision (switching structure)
- Numerical stability when stage lengths become zero
- No measure for quality of solution

Switching Time Optimization

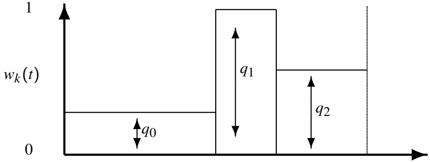
Advantage:

- Only continuous variables $\rightarrow$ very fast

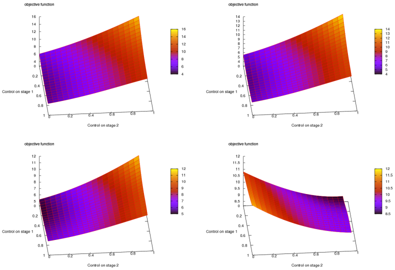
Disadvantages:

- Still discrete decision (switching structure)
- Numerical stability when stage lengths become zero
- No measure for quality of solution
- Initialization is crucial, many local minima

Example

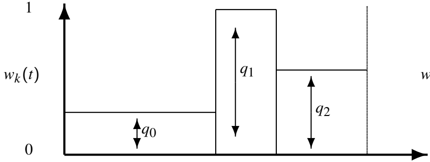


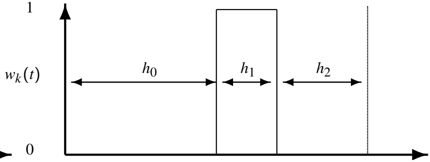
1
$w_k(t)$
0
q_0
q_1
q_2

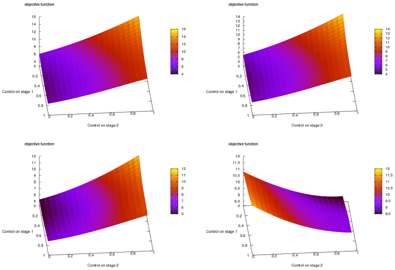


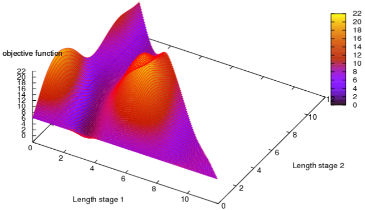
objective function
Control on stage 1
Control on stage 2
objective function
Control on stage 1
Control on stage 2
objective function
Control on stage 1
Control on stage 2
objective function
Control on stage 1
Control on stage 2

Example


1
0
$w_k(t)$
q_0
q_1
q_2


1
0
$w_k(t)$
h_0
h_1
h_2


objective function
Control on stage 1
Control on stage 2


objective function
Length stage 1
Length stage 2

Novel idea: Outer convexification

Reformulate nonlinear problem such that

- The relaxed problem has bang-bang solutions,
  no need to enumerate!

If this is not the case, then

- Obtain best lower bound from purely continuous problem
- Obtain an integer solution by adaptivity and rounding strategy
- Refine solution using switching time optimization

Relaxation and Outer Convexification

Problem (B) resp. (R)

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

subject to

$$\begin{aligned} x(t_0) &= x_0 \\ \omega(\cdot) &\in \Omega \text{ resp. } \mathrm{conv}\ \Omega \\ \dot{x}(t) &= f(x(t), \omega(t), u(t), p) \end{aligned}$$

Often $\phi^{\mathrm{R}} \ll \phi^{\mathrm{B}}$. Better bound?

Relaxation and Outer Convexification

Problem (B) resp. (R)

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

subject to

$$\begin{aligned} x(t_0) &= x_0 \\ \omega(\cdot) &\in \Omega \text{ resp. conv } \Omega \\ \dot{x}(t) &= f(x(t), \omega(t), u(t), p) \end{aligned}$$

Problem (BC) resp. (RC)

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

subject to

$$\begin{aligned} x(t_0) &= x_0 \\ w(\cdot) &\in \{0,1\}^{n_w} \text{ resp. } [0,1]^{n_w} \\ \dot{x}(t) &= \sum_{i=1}^{n_w} f(x(t), \omega^i, u(t), p)\, w_i(t) \\ \sum_{i=1}^{n_w} w_i(t) &= 1, \quad t \in [t_0, t_f] \end{aligned}$$

Bijection between binary solutions, $\phi^{\text{B}} = \phi^{\text{BC}}$.
What can we say about relation between ϕ^{RC} and ϕ^{BC}???

Approximating the state [S., Bock, Diehl, 2008]

THEOREM. Let $x(\cdot)$ and $y(\cdot)$ be solutions of the initial value problems

$$\begin{aligned} \dot{x} &= A(x)\, w, \quad x(0) = x_0, \\ \dot{y} &= A(y)\, v, \quad y(0) = y_0 \end{aligned}$$

with $t \in [0, t_f]$. If for all $t \in [0, t_f]$ it holds that

$$\begin{aligned} \| w \| &\leq 1, \quad \| v \| \leq 1, \\ \| A(x) \| &\leq M \quad \forall\, x \in \mathbb{R}^{n_x}, \\ \| A(y) - A(x) \| &\leq L \| y - x \| \quad \forall\, x, y \in \mathbb{R}^{n_x}, \\ \left\| \int_0^t v(\tau) - w(\tau)\, \mathrm{d}\tau \right\| &\leq \epsilon \end{aligned}$$

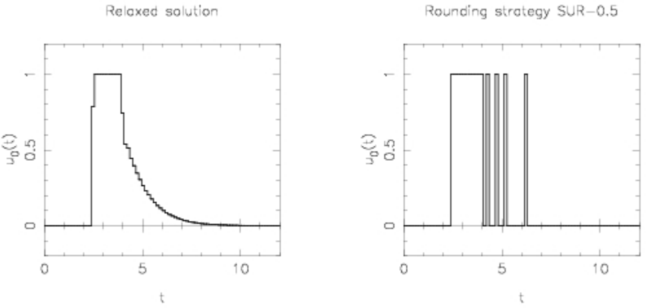
then for all $t \in [0, t_f]$ it holds

$$\| y(t) - x(t) \| \leq (2M\epsilon + \| y_0 - x_0 \|)\, e^{Lt}.$$

Sum Up Rounding

For $\Delta t_i = t_{i+1} - t_i$ and all $j = 1 \dots n_w, i = 1 \dots n_{int}$ 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, 2008]

THEOREM. Let functions $v : [0, t_f] \mapsto [0, 1]^{n_w}$,

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

and $w : [0, t_f] \mapsto \{0, 1\}^{n_w}$,

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

be given. Then it holds

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

Implications

- $\forall\epsilon$: If grid fine enough $\rightarrow \phi^{\text{RC}} \leq \phi^{\text{BC}} + \epsilon = \phi^{\text{B}} + \epsilon$
- Extension to path constraints straightforward, as functions continuous in states $x(\cdot)$
- Sum Up Rounding constructive way to get integer solution

- At the price of more control functions, we can calculate ϕ^{B} by solution of purely continuous problem!

Implications

- $\forall\epsilon$: If grid fine enough $\rightarrow \phi^{\text{RC}} \leq \phi^{\text{BC}} + \epsilon = \phi^{\text{B}} + \epsilon$
- Extension to path constraints straightforward, as functions continuous in states $x(\cdot)$
- Sum Up Rounding constructive way to get integer solution

- At the price of more control functions, we can calculate ϕ^{B} by solution of purely continuous problem!

- Note: if control constraints depend explicitely on $w(\cdot)$, problem specific analysis! [Kawajiri, Biegler, S., SMB: in progress]

MS MINTOC algorithm

- $k = 0$. Provide initial control discretization grid $\mathcal{G}^k$.
- Convexify problem (B). Relax this problem to $\tilde{w}(\cdot) \in [0, 1]^{n_w}$.
- REPEAT
  - Solve relaxed problem for control discretization $\mathcal{G}^k$, obtain the grid–dependent optimal value ${\phi^{\text{RC}}}_{\mathcal{G}^k}$ of the trajectory $\mathcal{T}^k$.
  - If the optimal trajectory on $\mathcal{G}^k$ is bang-bang then STOP.
  - Apply Sum Up Rounding to $\mathcal{T}^k$.
  - Use switching time optimization, initialized with this solution. Obtain upper bound ϕ^{STO}.
  - If feasible and $\phi^{\text{STO}} < \phi^{\text{RC}} + \varepsilon$ then STOP.
  - Refine the control grid $\mathcal{G}^k$, based on control values of trajectory $\mathcal{T}^k$. $k = k + 1$.

Testdrive benchmark problem

	Branch & Bound		MS Mintoc	
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

N = # discretization intervals $\approx$ # integer variables
CPU times in [hh:min:sec]

Left: *Gerdts, Optimal Control Applications and Methods, 2005*,
CPU times for Pentium III machine with 750 MHz

Right: *Kirches, S., Schlöder, Bock, submitted*,
AMD Athlon XP 3000+ with 2.166 GHz and 1024 MB of RAM

Applications I

Valves and ports

- SMB superstructure [S., Engell et. al., 2007], [Kawajiri, Biegler, S., ?]
- Slop cut recycling in batch distillation [S., Diehl, Bock, 2005]
  - For the first time considering variable reflux to single trays
  - 23% more profit compared to solution in the literature

Applications I

Valves and ports

- SMB superstructure [S., Engell et. al., 2007], [Kawajiri, Biegler, S., ?]
- Slop cut recycling in batch distillation [S., Diehl, Bock, 2005]
  - For the first time considering variable reflux to single trays
  - 23% more profit compared to solution in the literature

Yes/no decisions in biological systems

- Determination of phase resetting stimuli in calcium signalling pathway [Lebiedz, S. et. al., 2005]
- Manipulating circadian rhythms by light stimuli [Shaik, S. et. al, 2008]
- Benchmark population dynamics problem [S., Schlöder et al., 2006]

Applications II

Gears in transport

- New York subway control problem [S., Bock, Reinelt, 2008]
  - For the first time considering path constraints
- Gear choice in heavy duty trucks [ongoing thesis C. Kirches]
  - Feedback control, using available GPS data, 16 gears
- Benchmark problem MINLP: automobile testdriving
  [Gerdts, 05/06], [Kirches, S., et al., submitted]
- Periodic time-optimal automobile driving,
  [S., Kirches, et al., submitted]

Conclusions

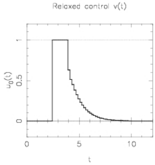
- Use outer convexification for modeling!!!
- Do not use standard MINLP methods,
  but Sum Up Rounding and Switching Time Optimization
  for integer control functions

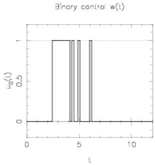
- Many problems solvable for the first time
  (or orders of magnitude faster)
- Allows to use more detailed models
- Real-time capable computing times!

Thank you very much for your attention!

Questions?

Example





m	Δt	$\int_0^{12} v - w$	$x_2^R(12)$	$x_2^B(12)$	$x_2^{\text{STO}}(12)$
2	6	2.87924	5.40278	8.51376	7.05533
4	3	0.631402	2.75402	5.08501	2.49896
8	1.5	0.585463	1.46812	1.90096	1.38276
16	0.75	0.0827811	1.35597	1.73284	1.38276
32	0.375	0.00718493	1.34881	1.61399	1.34972
64	0.1875	0.00151131	1.34406	1.34680	1.34569
128	0.09375	0.00135644	1.34405	1.34511	1.34479
256	0.046875	0.00130135	1.34402	1.34430	1.34424

- Here: bisection for illustrative purposes, in practice more advanced adaptivity strategies
- Integer solution for $m = 256$ switches 30 times!

NLP in Direct Single Shooting

After control and constraint discretization obtain NLP in variables q and x_0.
Note: calculate numerical DAE solution for each new (x_0, q)!

$$\underset{q,\,x_0}{\text{minimize}} \qquad \phi[x(t;q), z(t;q), u(t;q)]$$

subject to

$$\begin{aligned} c(t_i, x(t_i;q), z(t_i;q), u(t_i;q)) &\geq 0, \quad i = 0, \ldots, N, \\ r_\mathrm{i}\left(x_0, x(t_1;q), z(t_1;q), \ldots, x(t_f;q), z(t_f;q)\right) &\geq 0, \\ r_\mathrm{e}\left(x_0, x(t_1;q), z(t_1;q), \ldots, x(t_f;q), z(t_f;q)\right) &= 0. \end{aligned}$$

Solve with finite dimensional optimization solver, e.g., Sequential Quadratic Programming (SQP).

Direct Single Shooting: Pros and Cons

- **Sequential** simulation and optimization.
- \+ Can use state-of-the-art ODE/DAE solvers.
- \+ Few degrees of freedom even for large ODE/DAE systems.
- o Need only initial guess for controls q.
- \- Cannot use knowledge of x in initialization (e.g., in tracking problems), impact on convergence region of Newton type method.
- \- DAE solution $x(t;q)$ can depend very nonlinearly on q, existence not guaranteed.
- \- Unstable systems difficult to treat.
- Often used in engineering applications.

Direct Collocation (Sketch) [Tsang et al. 1975]

- Discretize controls and states on **fine** grid with node values $s_i \approx x(t_i)$.
- Replace infinite dimensional ODE

$$0 = \dot{x}(t) - f(x(t), u(t)), \quad t \in [t_0, t_f]$$

by finitely many equality constraints

$$\begin{aligned} c_i(q_i, s_i, s_{i+1}) &= 0, \quad i = 0, \ldots, N-1, \\ \text{e.g.,} \quad c_i(q_i, s_i, s_{i+1}) &:= \frac{s_{i+1} - s_i}{t_{i+1} - t_i} - f\left(\frac{s_i + s_{i+1}}{2}, q_i\right) \end{aligned}$$

- Approximate also integrals, e.g.,

$$\int_{t_i}^{t_{i+1}} L(x(t), u(t))dt \approx L\left(\frac{s_i + s_{i+1}}{2}, q_i\right)(t_{i+1} - t_i)$$

NLP in Direct Collocation

After discretization obtain large scale, but sparse NLP:

$$\underset{q,\,s}{\text{minimize}} \quad \phi[s,q]$$

subject to

$$\begin{aligned}
c_i(q_i, s_i, s_{i+1}) &= 0, \quad i = 0, \ldots, N-1, && \text{(discretized ODE/DAE model)} \\
c(s_i, q_i) &\geq 0, \quad i = 0, \ldots, N, && \text{(discretized path constraints)} \\
r_{\mathrm{i}}\,(s_0, \ldots, s_N) &\geq 0, && \text{(multipoint inequality constraints)} \\
r_{\mathrm{e}}\,(s_0, \ldots, s_N) &= 0. && \text{(multipoint equality constraints)}
\end{aligned}$$

Solve by SQP or interior point method for sparse problems.

Direct Collocation: Pros and Cons

- **Simultaneous** simulation and optimization.

+ Large scale, but very sparse NLP.
+ Can use knowledge of x in initialization.
+ Can treat unstable systems well.
+ Robust handling of path and terminal constraints.

- Adaptivity needs new grid, changes NLP dimensions.

- Successfully used for practical optimal control by, e.g., Biegler and Wächter (IPOPT), Betts, Bock/Schulz (OCPRSQP), v. Stryk (DIRCOL), ...