A "Hybrid" Solver for Optimal Control of Hybrid Systems

Nicolò Giorgetti and Alberto Bemporad

Amsterdam, June 16-17

Dept. of Information Engineering
University of Siena, Italy
giorgetti@dii.unisi.it

Limitations of MIP Approaches

Current approach for solving **optimal control** of hybrid systems: **M**ixed-Integer (linear/quadratic) **P**rogramming.

1) The discrete/logic part of the hybrid system must be converted into linear MI inequalities (e.g.: by HYSDEL)

➡ **LOSS of the original discrete structure**
Introduction of auxiliary binary variables

2) The efficiency of the MIP solver relies upon the tightness of the continuous LP/QP **relaxations.**

➡ **Poor performance (=many LP/QPs) if relaxations are not tight**

We need better solution techniques

"Hybrid" Solvers

Combine MIP and Constraint Logic Programming (CLP) to overcome the previous difficulties

Why CLP ?

- More flexible modeling than MIP
- Structure is kept and exploited to direct the search.

Why MIP ?

- Specialized techniques for highly structured problems (e.g. LP problems);
- A wide range of tight relaxations are available

Why a combined approach ?

Performance increase already shown in other application domains (Harjunkoski, Jain, Grossmann, 2000)

Constraint Logic Programming

CLP is a set of techniques for solving a **finite domain problem** (=set of constraints over a set of integer finite domain variables)

Example: $X \neq 7 \quad Z \neq 2 \quad X = Z + 3Y$

$X \in [1\#8] \quad Y \in [1\#10] \quad Z \in [1\#10]$

GOAL: Find all feasible assignments for X,Y, and Z

CLP alternates two techniques for solving the problem:

- Constraint Propagation: efficient inference mechanism used to reduce the domains of the variables.
- Constraint Distribution: splits a problem into complementary cases once constraint propagation cannot advance further.

Constraint Logic Programming

Example (continued):

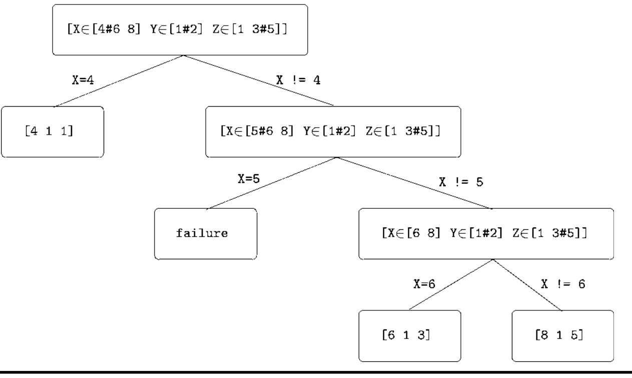
$X \neq 7 \quad Z \neq 2 \quad X = Z + 3Y$

$X \in [1\#8] \quad Y \in [1\#10] \quad Z \in [1\#10]$

A pre-phase of constraint propagation produces

$X \in [4\#6\ 8] \quad Y \in [1\#2] \quad Z \in [1\ 3\#5]$

By iterating between propagation and distribution the solutions of the problem are determined:



Logic-Based Branch&Bound

The basic modeling framework has the following form:
(Bockmayr, Kasper,1998)

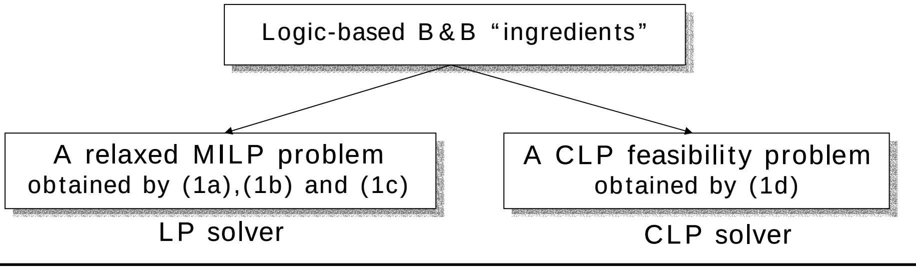
$$\min_v f(v) \quad \text{(1a)}$$

$$\text{s.t. } Gv \leq d,\ \overline{G}v = \overline{d} \quad \text{(1b) Continuous constraints}$$

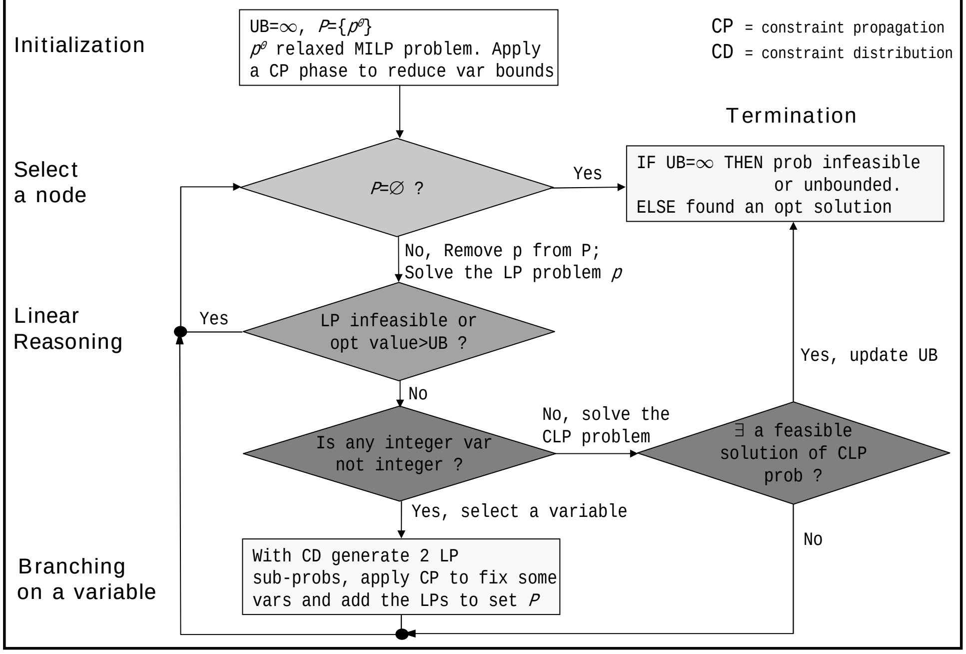
$$G'v + D'w' \leq E',\ \overline{G'}v + \overline{D'}w' = \overline{E'} \quad \text{(1c) Mixed constraints}$$

$$g(w, w') \quad \text{(1d) Logic constraints}$$

$$v \in \mathcal{R}^{n_v},\ w \in \{0,1\}^{n_w},\ w' \in \{0,1\}^{n'_w}$$

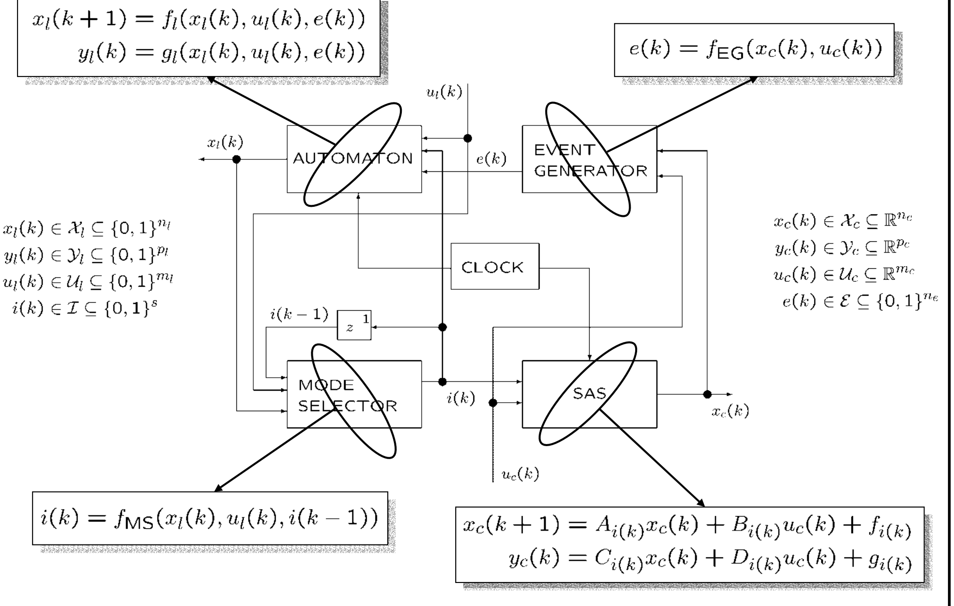


Logic-Based B&B algorithm



Discrete-Time Hybrid Systems

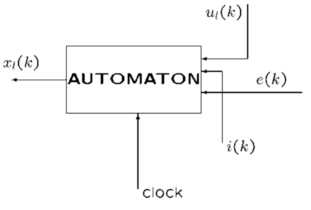
(Torrisi, Bemporad, 2003)



Hybrid Models for the Hybrid Solver

Continuous constraints	$Gv \leq d,\ \overline{G}v = \overline{d}$
Mixed constraints	$G'v + D'w' \leq E',\ \overline{G'}v + \overline{D'}w' = \overline{E'}$
Logic constraints	$g(w, w')$

Automaton

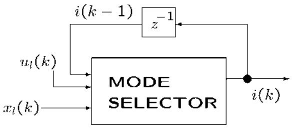


$$x_l(k+1) = f_l(x_l(k), u_l(k), e(k))$$
$$y_l(k) = g_l(x_l(k), u_l(k), e(k))$$

No transformation

Logic constraints

Mode Selector



$$i(k) = f_{\mathsf{MS}}(x_l(k), u_l(k), i(k-1))$$

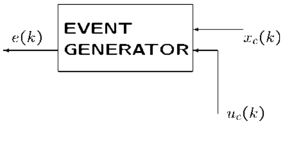
No transformation

Logic constraints

Hybrid Models for the Hybrid Solver

Continuous constraints	$Gv \leq d,\ \overline{G}v = \overline{d}$
Mixed constraints	$G'v + D'w' \leq E',\ \overline{G'}v + \overline{D'}w' = \overline{E'}$
Logic constraints	$g(w, w')$

Event Generator

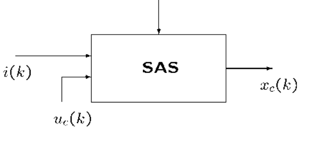


$$e(k) = f_{\mathsf{EG}}(x_c(k), u_c(k))$$

Big-M technique

Mixed constraints

Switched Affine System



$$x_c(k+1) = A_i x_c(k) + B_i u_c(k) + f_i$$
$$y_c(k) = C_i x_c(k) + D_i u_c(k) + g_i$$

Big-M technique

Mixed constraints
+
Continuous constraints

Optimal Control of Hybrid Systems

We define the following optimal control problem:

$$\min_{\{x(k),u(k)\}_{k=0}^{T-1}} \sum_{t=0}^{T-1} \|Q_x(x(k)-x_e(k))\|_\infty + \|Q_u(u(k)-u_e(k))\|_\infty + \|Q_y(y(k)-y_e(k))\|_\infty$$

s.to

$$x_l(k+1) = f_l(x_l(k), u_l(k), e(k)) \quad (1)$$
$$y_l(k) = g_l(x_l(k), u_l(k), e(k)) \quad (2)$$
$$x_c(k+1) = A_{i(k)} x_c(k) + B_{i(k)} u_c(k) + f_{i(k)} \quad (3)$$
$$y_c(k) = C_{i(k)} x_c(k) + D_{i(k)} u_c(k) + g_{i(k)} \quad (4)$$
$$e(k) = f_{\mathrm{EG}}(x_c(k), u_c(k)) \quad (5)$$
$$i(k) = f_{\mathrm{MS}}(x_l(k), u_l(k), i(k-1)) \quad (6)$$
$$h_D(k)(\{x,u,e,i\}_0^{T-1}) \leq 0 \quad (7)$$
$$h_A(k)(\{x,u,e,i\}_0^{T-1}) \leq 0 \quad (8)$$

where:

(1-6) are dynamical constraints

(7) are design constraints (e.g.: input/state/logic constraints)

(8) are ancillary constraints (don't change the solution, only help the solver. E.g.: reachability constraints)

This problem can be solved by the hybrid solver

Motorbike Example

MODEL: a simplified hybrid model of a motorbike with three "semi-automatic" gears.

GOAL: solve an optimal control problem in order to track a desired speed profile

SAS - Discrete-time continuous dynamics

Speed (Km/h) $v(k+1) = (1-\alpha_i)v(k) + \beta_i \omega(k) - c u_b$

Engine speed (rpm) $\omega(k+1) = \omega(k) + d u_t - f u_b,$

2 continuous inputs:

u_t engine torque (Nm)

u_b brake force (N)

2 continuous states:

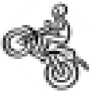
v and **ω**

Event Generator – Thresholds events for automatic gear shift

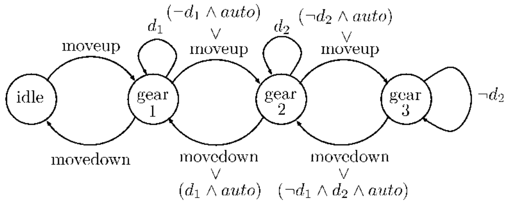
$$[d_1(k) = 1] \longleftrightarrow [\omega(k) \leq t_1]$$
$$[d_2(k) = 1] \longleftrightarrow [\omega(k) \leq t_2]$$

where t_1 and t_2, $t_1 \leq t_2$, are constant thresholds.

Motorbike Example



Automaton - Gear automaton



3 binary inputs:

auto: automatic gear shifts enabled

moveup, **movedown:** manual gear shift commands

4 binary states:

idle, gear1, gear2, gear3

Mode selector – Current gear of the motorbike (≡ logic state)

Optimal Control Problem: minimize $\sum_{k=0}^{T} |v(k) - v_e|$ subject to the hybrid dynamics and the following additional constraints:

Continuous Constraints

$$0 \leq u_t(k) \leq 50$$
$$0 \leq u_b(k) \leq 50.$$

Logic condition on torque and brakes

$$[d_t = 0] \longleftrightarrow [u_t(k) \leq 0]$$
$$[d_b = 0] \longleftrightarrow [u_b(k) \leq 0]$$
$$\neg(d_t \wedge d_b)$$

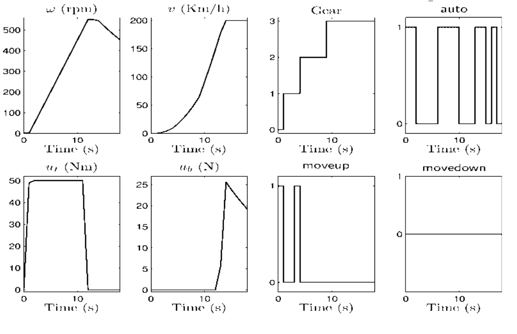
An exclusive-or condition on the gear commands

$$\text{moveup}(k) \oplus \text{movedown}(k)$$

Motorbike Example

Initial conditions:
$v(0) = 0$ Km/h
$\omega(0) = 0$ rad/s
gear = idle

$v_e = 200$ Km/h
$T = 20$ s
$t_1 = 100$ rpm
$t_2 = 400$ rpm



Same results both with MILP and with lb-B&B

CPU time

(Pentium IV 1.8GHz
Ilog CPLEX 8.1
+Solver 5.3)

Horizon	Integer vars	MILP	lb-B&B	Reduction(%)
10	110	0.221	0.12	45.70
20	220	8.091	1.992	75.38
30	330	26.318	7.798	70.37
40	440	69.961	20.128	71.23

LPs solved

Horizon	MILP	lb-B&B
10	309	6
20	30645	49
30	71070	126
40	146101	229

Conclusions

Current state:

✓ A unifying framework for MIP and CLP techniques for solving optimal control problems for hybrid systems.

✓ A more expressive modeling language for hybrid systems (due to CLP).

✓ A superior computation time in comparison to more standard mixed-integer programming techniques.

Ongoing and Future Research

- MIQP logic-based solvers for quadratic performance indices
- Embed MILP solvers in CLP as logic constraints, and solve satisfiability problems
- Alternative relaxations tighter than big-M (particularly for SAS and EG parts)
- Applications

References

[1] A. Bemporad and N. Giorgetti. Logic-based hybrid solvers for optimal control of hybrid systems. 2003. Submitted CDC 2003.

Paper download: http://www.dii.unisi.it/~bemporad