15th Triennial World Congress, Barcelona, Spain

OPTIMAL CONTROL OF SWITCHED CONTINUOUS SYSTEMS USING MIXED-INTEGER PROGRAMMING

Olaf Stursberg [*,**] and **Sebastian Engell** [**]

[*] *currently at: Carnegie Mellon University, ECE Dept., 5000 Forbes Ave, Pittsburgh, PA 15213; olaf@ece.cmu.edu*
[**] *Process Control Lab (CT-AST), Chemical Eng. Dept., University of Dortmund, D-44221 Dortmund (Germany)*

Abstract: The paper presents an optimization-based approach to compute controllers for a class of hybrid systems with switched dynamics. The starting point is a representation as a hybrid automaton which models autonomous switching between different nonlinear dynamics and includes discrete as well as continuous control inputs. The automaton is transformed into a linear discrete-time model in equation-based form. The task of generating an optimal control law to drive the system from an initial state into a target region (while avoiding forbidden states) is solved by mixed-integer programming performed in a moving-horizon setting with variable time steps.

Keywords: Automata, Hybrid Dynamics, Integer Programming, Optimization, Predictive Control, Switched Systems.

1. INTRODUCTION

Two different discrete phenomena are usually distinguished in hybrid system behavior: the switching between different dynamics and jumps in the state trajectory (Branicky *et al.*, 1998). Particularly switching has been identified as a suitable approximation of several phenomena in a large variety of technical applications, as for example logic controllers in processing systems. This contribution considers a class of hybrid systems with a behavior that is determined by switching inputs (besides continuous controls) and autonomous changes between different sets of ODEs. For such a system, the problem of determining the control trajectories that drive it optimally from an initial state into a new target point or region is investigated. Unlike many other approaches in the context of optimal control and optimization of hybrid systems (see e. g. (Broucke *et al.*, 2000), (Gokbayrak and Cassandras, 2000), (Hedlund and Rantzer, 1999), (Slupphaug *et al.*, 1997), (Branicky *et al.*, 1998)) the method proposed in this paper transforms the control task into a set of discrete-time equations and inequalities. The latter contain continuous and discrete variables such that mixed-integer programming (MIP) is used for the solution.

In difference to the methods published by (Buss *et al.*, 2000), (Galan and Barton, 1998), (Schweiger and Floudas, 1998), the optimization is performed repeatedly on a moving horizon in order to enable the use for large time horizons. This MPC-like strategy can be found also in the method introduced by (Bemporad and Morari, 1999) for so-called MLDs. While the MIP solution on moving horizons and some parts of the logic formulation are similar in both approaches, the work in this paper starts from a different type of model (hybrid automata), uses a different objective function, explicitly considers the exclusion of forbidden regions, and treats the switching times as optimization parameters (i.e., variable time steps are used). The latter allows using relatively large time horizons even if only a small number of sampling points can be considered for optimization due to complexity reasons. As shown for an example, this can be important in order to find a feasible path into the target while avoiding to move into forbidden regions.

2. OPTIMAL CONTROL OF HYBRID AUTOMATA WITH SWITCHED DYNAMICS

The following definition of hybrid automata formulates the dynamics considered in this paper. The definition is based on the one in (Stursberg and Engell, 2001) but it is slightly modified with respect to the state set and the state transitions:

Definition 2.1. Hybrid Automaton $A_{N,C}$
The automaton with switched non-linear continuous time dynamics is given by:

$$A_{N,C} = (\boldsymbol{X}, \boldsymbol{U}, V, E, Z, f, \phi) \qquad (1)$$

with the following components: The state vector $\boldsymbol{x}(t)$, $dim(\boldsymbol{x}) = n$ is defined on the convex state space $\boldsymbol{X} := \{\boldsymbol{x} \in \boldsymbol{X} \mid \boldsymbol{C} \cdot \boldsymbol{x} \leq \boldsymbol{d}, \boldsymbol{C} \in \mathbb{R}^{q\times n}, \boldsymbol{d} \in \mathbb{R}^{q\times 1}, q \in \mathbb{N}\}$ and the continuous inputs $\boldsymbol{u}(t)$, $dim(\boldsymbol{u}) = m_u$ on $\boldsymbol{U} = [u_1^-, u_1^+] \times \ldots \times [u_{m_u}^-, u_{m_u}^+]$ with $u_j^-, u_j^+ \in \mathbb{R}$. The discrete input $\boldsymbol{v}(t) \in \boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r\}$, $dim(\boldsymbol{v}) = m_v$ switches between finitely many options and only finitely often at times t_k in a time interval $[t_0, t_f]$. A set $E = \{E_1, \ldots, E_{n_E}\}$ of hyperplanes $E_j := \{\boldsymbol{x} \in \boldsymbol{X} \mid \boldsymbol{c}_j \cdot \boldsymbol{x} = d_j, \boldsymbol{c}_j \in \mathbb{R}^{1\times n}, d_j \in \mathbb{R}\}$ partitions the state space into a set $R = \{R_1, \ldots, R_{n_R}\}$ of convex and disjunct regions $R_i := \{\boldsymbol{x} \in \boldsymbol{X}, E_i \in E \mid J = \{1, 2, \ldots, n_E\}, H \subseteq J, \forall h \in H : \boldsymbol{c}_h \cdot \boldsymbol{x} \sim_h d_h, \sim_h \in \{<, \leq\}\}$ such that $\bigcup_{i=1}^{n_R} R_i = \boldsymbol{X}$. The set of discrete states $Z = \{z_1, \ldots, z_{n_Z}\}$ is formed by a mapping $\rho : R \to Z$ that assigns one discrete state $z_i \in Z$ to each region R_i.

The continuous dynamics is given such that $\boldsymbol{f} : \boldsymbol{X} \times \boldsymbol{U} \times \boldsymbol{V} \times Z \to \mathbb{R}^n$ defines $\boldsymbol{x}(t)$ as the unique solution of an ODE $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_z(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t))$ for a time interval $[t_k, t_{k+1}[$ between two switching events. At each point of time t that dynamics $\boldsymbol{f}_z$ is valid which is assigned to the discrete state z_i with $R_i = \rho^{-1}(z_i)$, $\boldsymbol{x}(t) \in R_i$. According to the transition function $\phi = Z \times \boldsymbol{X} \times \boldsymbol{X} \times R \to Z$, a transition $z_1 \to z_2$ $(z_1, z_2 \in Z)$ occurs at t_k, if: $\boldsymbol{x}(t_k^-) \in R_1$, $\rho(R_1) = z_1$ and $\boldsymbol{x}(t_k) \in R_2$, $\boldsymbol{x}(t_k) \notin R_1$, $\rho(R_2) = z_2$. Hence, if R_1 is left across E_j, the transition guard is: $\boldsymbol{c}_j \cdot \boldsymbol{x}(t_k^-) = d_j \wedge \boldsymbol{c}_j \cdot \boldsymbol{x}(t_k) > d_j$ (if E_j belongs to R_1), or $\boldsymbol{c}_j \cdot \boldsymbol{x}(t_k^-) < d_j \wedge \boldsymbol{c}_j \cdot \boldsymbol{x}(t_k) = d_j$ (if E_j is assigned to R_2) respectively. It is required that $\boldsymbol{x}(t)$, $t \in [t_0, t_f]$ fulfills the continuity condition $\boldsymbol{x}^+ = \boldsymbol{x}(t^*)$ for all discrete transitions and switching events at a time t^*. ($\boldsymbol{x}^+$ is the time successor of $\boldsymbol{x}(t^*)$.)

Semantics: Let $T = \{t_0, t_1, t_2, \ldots, t_N, t_f\}$ contain the initial time t_0, the final time t_f and all points of time $t_k \in]t_0, t_f[$, at which a transition according to ϕ or a switching in $\boldsymbol{v}(t)$ occurs. A valid *run* $r : T \to Z \times \boldsymbol{X}$ of $A_{N,C}$ is then the finite sequence $r(t_0), r(t_1), r(t_2), \ldots, r(t_N), r(t_f)$ of *hybrid states* $r(t_k) = (z(t_k), \boldsymbol{x}(t_k))$ such that:

(a) *Initialization*: $r(t_0) = (z(t_0), \boldsymbol{x}_0)$ with $z(t_0) \in Z$, $\boldsymbol{x}_0 = \boldsymbol{x}(t_0) \in \boldsymbol{X}$ with $\boldsymbol{x}_0 \in R^* \in R$, $\rho(R^*) = z(t_0)$.
(b) *Progress*: $r(t_{k+1}) = (z(t_{k+1}), \boldsymbol{x}(t_{k+1}))$ follows from $r(t_k)$ by the assignments:
$z(t_{k+1}) = \phi(z(t_k), \boldsymbol{x}(t_{k+1}^-), \boldsymbol{x}(t_{k+1}), R_j)$
with $\boldsymbol{x}(t_{k+1}^-) \in R_j$, $\rho(R_j) = z(t_k)$ and
$$\boldsymbol{x}(t_{k+1}) = \int_{t_k}^{t_{k+1}^-} \boldsymbol{f}_{z_k}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t))dt$$
with $\boldsymbol{v}(t) = \boldsymbol{v}_j \in \boldsymbol{V}$ for $t \in [t_k, t_{k+1}[$. ◇

For this type of automaton, the following optimal control problem is posed: Assume that

- a hyperrectangular *target region* $R_T \subset \boldsymbol{X}$ with $R_T = [x_{T,1}^-, x_{T,1}^+] \times \ldots \times [x_{T,n}^-, x_{T,n}^+]$ and $x_{T,j}^-, x_{T,j}^+ \in \mathbb{R}$,
- a set $R_F = \{R_{F,1}, \ldots, R_{F,n_F}\}$ of polyhedral *forbidden regions* $R_{F,i} \subset \boldsymbol{X}$, and $R_{F,i} := \{\boldsymbol{x} \in R_{F,i} \mid \boldsymbol{C}_F \cdot \boldsymbol{x} \leq \boldsymbol{d}_F, \boldsymbol{C}_F \in \mathbb{R}^{q_F\times n}, \boldsymbol{d}_F \in \mathbb{R}^{q_F\times 1}\}$, $R_{F,i} \cap R_T = \emptyset$,
- and an initial state $\boldsymbol{x}_0 = \boldsymbol{x}(t_0) \in \boldsymbol{X}$ with $\boldsymbol{x}_0 \notin R_T$ and $\boldsymbol{x}_0 \notin R_{F,i}$ are given.

The aim is to compute those input trajectories $\boldsymbol{u}(t)$, $\boldsymbol{v}(t)$ which optimally move the system from $\boldsymbol{x}_0$ into R_T while none of regions in R_F is crossed:

$$\min_{\boldsymbol{u}(t),\boldsymbol{v}(t)} \Omega(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)), \qquad (2)$$

$$\Omega = \int_{t_0}^{t_f} (\alpha(t, \boldsymbol{x}) + \beta(t, \boldsymbol{u}) + \gamma(t, \boldsymbol{v}) + \delta(t))dt,$$

$$s.t. \quad \boldsymbol{x}_0 = \boldsymbol{x}(t_0), \quad \boldsymbol{x}(t) \in \boldsymbol{X}, \quad \boldsymbol{x}(t_f) \in R_T$$
$$\boldsymbol{x}(t) \notin R_{F,i} \quad \forall t \in [t_0, t_f] \quad \forall R_{F,i} \in R_F,$$
and subject to the dynamics of $A_{N,C}$.

The terms of the objective function Ω have the following meaning:

- $\alpha(t, \boldsymbol{x}) = \mu_1(t) \cdot \|\boldsymbol{w}_1 \cdot (\boldsymbol{x}(t) - R_T)\|_1$ denotes the distance between the current state and (the nearest boundary) of the target region; $\mu_1 (> 0)$ and $\boldsymbol{w}_1$ are appropriate weights;
- $\beta(t, \boldsymbol{u}) = \mu_2(t) \cdot \|\boldsymbol{w}_2 \cdot (\boldsymbol{u}(t) - \boldsymbol{u}_S)\|_1$ measures the deviation of $\boldsymbol{u}(t)$ from a reference vector $\boldsymbol{u}_s$ (again with weights μ_2 and $\boldsymbol{w}_2$);
- $\gamma(t, \boldsymbol{v}) = \mu_3(t) \cdot \begin{cases} w_3 : & \text{if } \boldsymbol{v}(t^-) \neq \boldsymbol{v}(t) \\ 0 : & \text{else} \end{cases}$ adds the amount w_3 to the costs (weighted by μ_3) if input switching occurs;
- $\delta(t) = \begin{cases} \mu_4(t) : & \text{if } \boldsymbol{x}(t) \neq R_T \\ 0 : & \text{else} \end{cases}$ is the weighted sum of the time required to reach the target.

The different weights (all non-negative) are design parameters that are chosen such that specific optimization objectives are met.

3. MODEL TRANSFORMATION AND SOLUTION BY MIP

The control problem posed in the previous section is clearly non-trivial since it involves the optimization of nonlinear equations depending on logical decisions. The strategy proposed here transforms the problem into a linear, discrete-time formulation and uses branch-and-bound optimization to solve the resulting mixed-integer problem.

3.1 *A Discrete-Time Hybrid Automaton*

First, a hybrid automaton with linear discrete-time dynamics is defined. This automaton is then used to approximate the behavior of $A_{N,C}$.

Definition 3.1. Hybrid Automaton $A_{L,D}$
A hybrid automaton with *switched linear discrete time dynamics*:

$$A_{L,D} = (\boldsymbol{X}, \boldsymbol{U}, V, E, Z, \boldsymbol{f}^D, \phi^D) \tag{3}$$

consists of the state space $\boldsymbol{X}$, the continuous input space $\boldsymbol{U}$, and the set of discrete inputs V as for $A_{N,C}$. The partitioning of $\boldsymbol{X}$ into a set of polyhedral regions $R = \{R_1, \dots, R_{n_R}\}$ by the set E of switching planes is also the same. But the trajectories are now defined on a discrete time domain $t_k \in T = \{t_0, t_1, \dots, t_f\}$, i.e., the variables are constant on each time interval $[t_k, t_{k+1}[$. The discrete state set $Z = \{z_1, \dots, z_{n_z}\}$ again results from an assignment $\rho : R \to Z$ of one discrete state to each region $R_i \in R$.

The continuous state transfer function $\boldsymbol{f}^D : \boldsymbol{X} \times \boldsymbol{U} \times V \times \boldsymbol{Z} \to \mathbb{R}^n$ determines a new continuous state according to the linear, discrete-time equation $\boldsymbol{x}_{k+1} := \boldsymbol{x}(t_{k+1}) = \boldsymbol{A}_{z,\boldsymbol{v}_k}\boldsymbol{x}(t_k) + \boldsymbol{B}_{z,\boldsymbol{v}_k}\boldsymbol{u}(t_k) + \boldsymbol{L}_{z,\boldsymbol{v}_k}$ with matrices $\boldsymbol{A}_{z,\boldsymbol{v}_k} \in \mathbb{R}^{n \times n}$, $\boldsymbol{B}_{z,\boldsymbol{v}_k} \in \mathbb{R}^{n \times p}$, and $\boldsymbol{L}_{z,\boldsymbol{v}_k} \in \mathbb{R}^{n \times 1}$ depending on the discrete state $z_k = z(t_k) \in Z$ and the current discrete input $\boldsymbol{v}_k = \boldsymbol{v}(t_k)$.

The transition function $\phi^D : Z \times \boldsymbol{X} \times \boldsymbol{X} \times R \to Z$ specifies the current discrete state: For two regions $R_a, R_b \in R$, a transition $z_k \to z_{k+1}$ occurs at t_{k+1}, if: $\boldsymbol{x}_k \in R_a$, $\rho(R_a) = z_k$ and $\boldsymbol{x}_{k+1} \in R_b$, $\rho(R_b) = z_{k+1}$, $\boldsymbol{x}_{k+1} \notin R_a$. If R_a is left across E_j the transition guard is $(\boldsymbol{c}_j \cdot \boldsymbol{x}_k \leq d_j) \wedge (\boldsymbol{c}_j \cdot \boldsymbol{x}_{k+1} > d_j)$ if E_j belongs to R_a, or $(\boldsymbol{c}_j \cdot \boldsymbol{x}_k < d_j) \wedge (\boldsymbol{c}_j \cdot \boldsymbol{x}_{k+1} \geq d_j)$ if E_j is assigned to R_b.

Semantics: Transitions and changes in $\boldsymbol{v}(t)$ and $\boldsymbol{u}(t)$ can occur only at the points of time in T. A valid *run* of $A_{L,D}$ is defined by $r : T \to \boldsymbol{X} \times Z$ as the sequence $r(t_0), r(t_1), \dots, r(t_f)$ of hybrid states $r(t) = (\boldsymbol{x}(t), z(t))$ according to:

(a) *Initialization*: $r(t_0) = (\boldsymbol{x}_0, z(t_0))$ with $\boldsymbol{x}_0 = \boldsymbol{x}(t_0) \in R^* \in R$, $\rho(R^*) = z(t_0) \in Z$.
(b) *Progress*: $r(t_k) = (\boldsymbol{x}_k, z_k)$ for $t_k \in T \setminus t_0$ results from:
   1. continuous evolution: $\boldsymbol{x}_{k+1} = \boldsymbol{f}^D(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{v}_k, z_k)$;
   2. discrete transitions: $z_{k+1} = \phi(z_k, \boldsymbol{x}_k, \boldsymbol{x}_{k+1}, R_j)$ with $x_k \in R_j$, $x_{k+1} \notin R_j$. ◇

3.2 *Reformulation of the Control Problem*

The transformation of the $A_{N,C}$-model into $A_{L,D}$ comprises the following steps: First, the nonlinear dynamics is linearized: In the simplest form, the dynamics $\boldsymbol{f}$ of $A_{N,C}$ is linearized at the center-point $\boldsymbol{x}_i^c$ of each region R_i for a fixed value $\boldsymbol{u}^c$ and each discrete input vector $\boldsymbol{v} \in \boldsymbol{V}$: $\boldsymbol{x}_{k+1} = \boldsymbol{A}^l_{z,\boldsymbol{v}_k}\boldsymbol{x}^l_k + \boldsymbol{B}^l_{z,\boldsymbol{v}_k}\boldsymbol{u}^l_k + \boldsymbol{L}^l_{z,\boldsymbol{v}_k}$ with $\boldsymbol{x}^l_k = \boldsymbol{x}_k - \boldsymbol{x}^c_i$, $\boldsymbol{u}^l_k = \boldsymbol{u}_k - \boldsymbol{u}^c$. A more accurate solution is used for the moving horizon strategy described in the next subsection: In this case the nonlinear dynamics is linearized at the initial state $\boldsymbol{x}_k$ of each optimization.

The second step of the transformation is time discretization: Variable time steps are used in order to treat the switching times as optimization parameters and to achieve that the optimization horizon can cover a larger time range. The time step $\triangle t_k = t_k - t_{k-1}$, $t_k \in T \setminus \{t_0\}$ is defined as:

$$\triangle t_k = \sum_{i=1}^{p} \delta_i \cdot \triangle t_{min} \cdot b_{i,k} \in \triangle T \tag{4}$$

with a minimum time step $\triangle t_{min} \in \mathbb{R}$, a set of constant parameters δ_i, and finitely many binary variables $b_{i,k} \in \{0, 1\}$. The values of the latter is computed by the optimization and determines the progress in time.

Since $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$, and $\boldsymbol{v}(t)$ are constant on the interval $[t_k, t_{k+1}[$, the dynamics obtained from linearization can be rewritten into:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= e^{\boldsymbol{A}^l_{z,\boldsymbol{v}_k} \cdot \triangle t_k} \cdot \boldsymbol{x}^l_{k+1} + \\ &\int_0^{\triangle t_k} e^{\boldsymbol{A}^l_{z,\boldsymbol{v}_k} \cdot \triangle t_k - \tau} d\tau \cdot (\boldsymbol{B}^l_{z,\boldsymbol{v}_k} \cdot \boldsymbol{u}^l_k + \boldsymbol{L}^l_{z,\boldsymbol{v}_k}) \\ &=: \boldsymbol{A}_{z,\boldsymbol{v}_k} \cdot \boldsymbol{x}^l_k + \boldsymbol{B}_{z,\boldsymbol{v}_k} \cdot \boldsymbol{u}^l_k + \boldsymbol{L}_{z,\boldsymbol{v}_k}. \end{aligned} \tag{5}$$

Hence, the matrices in the dynamics of $A_{L,D}$ do not only depend on R_i and $\boldsymbol{v}_k$ but also on the time step $\triangle t_k$. The choice between these different discrete options makes the formulation nonlinear. In order to obtain a linear representation in equation-based form the following formulations are introduced:

- Rewriting the model in (5) into:

$$\lambda_{z,\boldsymbol{v}_k,\triangle t_k} := \boldsymbol{x}_{k+1} - \boldsymbol{A}_{z,\boldsymbol{v}_k} \cdot \boldsymbol{x}^l_k - \boldsymbol{B}_{z,\boldsymbol{v}_k} \cdot \boldsymbol{u}^l_k - \boldsymbol{L}_{z,\boldsymbol{v}_k}, \tag{6}$$

the activation of a specific model follows from:

$$\sum_{z \in Z} \sum_{\boldsymbol{v}_k \in \boldsymbol{V}} \sum_{\triangle t_k \in \triangle T} b_{z_k, \boldsymbol{v}_k, \triangle t_k} \cdot \lambda_{z, \boldsymbol{v}_k, \triangle t_k} = 0,$$
$$\sum_{z \in Z} \sum_{\boldsymbol{v}_k \in \boldsymbol{V}} \sum_{\triangle t_k \in \triangle T} b_{z_k, \boldsymbol{v}_k, \triangle t_k} = 1 \qquad (7)$$

where the binary variable $b_{z_k, \boldsymbol{v}_k, \triangle t_k}$ stands for a specific combination of region, discrete input, and time step.

- To linearize the products of binary and continuous variables (contained in (7)), the so-called *Big-M-approach* (Glover, 1975; Williams, 1978) is employed. The product $r = b \cdot x$ between a binary variable b and a continuous variable x is replaced by:

$$r \geq b \cdot x_{min}, r \leq b \cdot x_{max}, \qquad (8)$$
$$r \leq x - (1-b) \cdot x_{min}, r \geq x - (1-b) \cdot x_{max}.$$

  The constants $x_{min}, x_{max} \in \mathbb{R}$ denote the minimum and maximum values of x. (The same equations are chosen for $b \cdot u$.)
- In each point of time, it has to be checked in which region R_i the current state $\boldsymbol{x}_k$ lies, i.e., if $\boldsymbol{C}_i \boldsymbol{x}_k \leq \boldsymbol{d}_i$ is fulfilled. Similar as in (Williams, 1978; Bemporad and Morari, 1998), the following formulation is used:

$$\boldsymbol{C}_i \cdot \boldsymbol{x}(t_k) \leq \boldsymbol{d}_i + (1-b) \cdot \boldsymbol{s}_{max}, \qquad (9)$$
$$\boldsymbol{C}_i \cdot \boldsymbol{x}(t_k) > \boldsymbol{d}_i + b \cdot \boldsymbol{s}_{min}$$

  where the binary variable $b \in \{0, 1\}$ is 1 if $\boldsymbol{x}_k$ is in R_i, and 0 otherwise. The vectors $\boldsymbol{s}_{min}, \boldsymbol{s}_{max} \in \mathbb{R}^{q \times 1}$ denote the bounds of $\boldsymbol{C}_i \boldsymbol{x}(t_k) - \boldsymbol{d}_i$ in $\boldsymbol{X}$. While (9) applies for the case that all bounding hyperplanes belong to R_i, the symbols $<$ and $\geq$ have to be used for hyperplanes that do not belong to R_i. The test, whether a forbidden region is reached, can be modelled in the same manner. If the binary variable assigned to a forbidden region is explicitly set to zero, control inputs that would lead into this region are eliminated from the set of feasible solutions of the optimization problem.
- The objective function Ω in (2) has to be rewritten in a discrete time form, i.e. the integral is replaced by a sum over the discrete points of time. Furthermore, the term $\alpha(\boldsymbol{x})$ contains the logical decision whether $\boldsymbol{x}_k$ is in R_T or not. This can be implemented according to (9) also.

The linearization of (7) and the test on region containment (9) can alternatively be modelled by *disjunctive formulations*, which reduce the required number of binary variables (Stursberg and Panek, 2002).

3.3 *Optimization on Moving Horizons*

The reformulation gives a linear representation of the optimal control problem for $A_{L,D}$. This allows using mixed integer linear programming (MILP) to compute the control trajectories. The simplest way is to perform the optimization at once for the complete horizon T. But even with very efficient MILP-solvers this approach is only applicable for short horizons (wrt. the number of points in T), since the complexity grows exponentially with $|T|$. Instead the moving horizon approach illustrated in Fig. 1 is used: The original automaton $A_{N,C}$ and the approximating automaton $A_{L,D}$ are provided, the first for simulation and the second as the optimization model. After initialization, $A_{L,D}$ is optimized for a prediction horizon with only a small number of points $P = \{t_1, t_2, \ldots, t_{n_P}\}$, $P \subset T$ with usually $n_P << |T|$. The first time step $\triangle t_1$ within P is set to $\triangle t_{min}$ and, in order to cover a large time period for the prediction, the time steps increase over P.

The optimization result for the first step is applied to the simulation model which is then simulated for the interval $\triangle t_1$. The *exact* new state $\boldsymbol{x}_{k+1}$ is returned to the optimizer and the optimization is repeated with a horizon that is shifted one time step ahead. This alternating procedure is repeated until the target region R_T (or a maximum number of steps) is reached. Besides the applicability to larger time horizons, the approach has the advantage that deviations between $A_{L,D}$ and $A_{N,C}$ are corrected in each step.

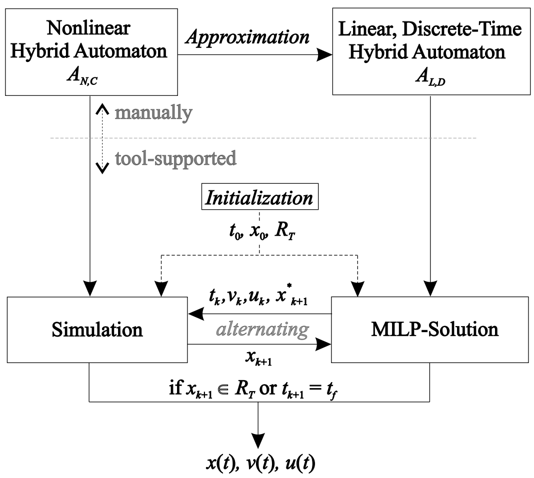

Fig. 1. Scheme of the moving horizon approach.

4. EXAMPLE: TANK SYSTEM

The approach is illustrated using a laboratory plant consisting of two tanks (Fig. 2): Tank T1 is filled by the flow F_1 that can be controlled through the valve V_1. It represents a scalar continuous control which can be varied within the interval $u \in [F_{1,min}, F_{1,max}]$. The liquid is transferred from T1 into a second tank T2 through a connecting pipe. The valve V_2, which can be switched between two settings $\{V_{2,1}, V_{2,2}\}$, allows to adjust the flow

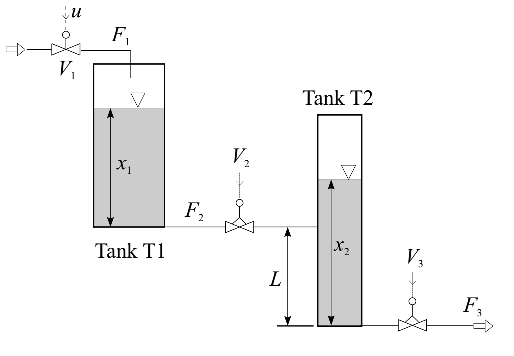

Fig. 2. The tank system.

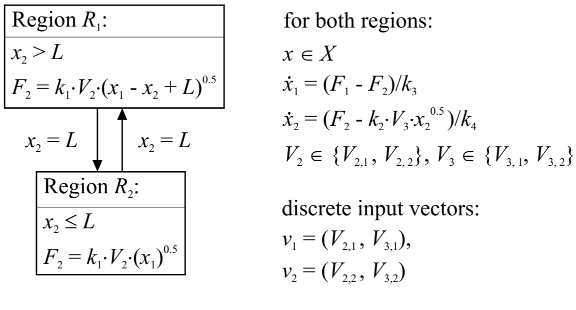

Fig. 3. The nonlinear hybrid model of the example

F_2 discretely. In the same way, the outflow F_3 of tank T2 is discretely controlled by another valve with two settings $V_3 \in \{V_{3,1}, V_{3,2}\}$. In the following it is assumed that only the combinations $\boldsymbol{v}_1 = (V_{2,1}, V_{3,1})$ and $\boldsymbol{v}_2 = (V_{2,2}, V_{3,2})$ can occur, i. e., the discrete input set is $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2\}$. When modelling the behavior of the system, it has to be considered that the dynamics of the liquid levels x_1 and x_2 in both tanks changes autonomously when x_2 crosses the value L (the height of the connecting pipe). Fig. 3 shows a corresponding $A_{N,C}$ model with distinct flows F_2 for two regions R_1, R_2 and constant parameters k_1 to k_4 . (The dynamics changes also at $x_2 = L + x_1$ because of a reversed flow direction through the pipe – this fact is neglected since the region $x_2 \geq L + x_1$ is completely contained in a forbidden region, see below).

The following control problem is investigated: Those trajectories $\boldsymbol{v}(t)$ and $\boldsymbol{u}(t)$ have to be determined which drive the tank system from an initial state $\boldsymbol{x}_0 = (0.01, 0.01)$ to a target region R_T such that an objective function Ω is minimized and that forbidden regions $R_{F,1}$ and $R_{F,2}$ are not reached, i.e.:

$$\min_{\boldsymbol{v}(t),\boldsymbol{u}(t)} \Omega(t, \boldsymbol{x}(t)), \qquad (10)$$

$$\Omega = \int_0^{t_f} \mu_1(t) \cdot \|\boldsymbol{x}(t) - R_T\|_1 \cdot dt + \mu_4 \cdot t_f,$$

$$\begin{aligned} s.t.\quad & \boldsymbol{x}_0 = (0.01, 0.01), \\ & \boldsymbol{x}(t_f) \in R_T = [0.3, 0.5] \times [0.5, 0.6], \\ & \boldsymbol{x}(t) \in \boldsymbol{X} = [0, 0.6] \times [0, 0.6], \forall t \in [t_0, t_f], \\ & \boldsymbol{x}(t) \notin R_{F,1} = [0.2, 0.6] \times [0, 0.2], \\ & \boldsymbol{x}(t) \notin R_{F,2} = [0, 0.3] \times [0.3, 0.6], \\ & \text{for the dynamics given in Fig. 3.} \end{aligned}$$

For simplicity, the chosen function Ω does not involve switching costs but only terms for the final time t_f and for the distance between $\boldsymbol{x}(t)$ and the target region. To assign a higher priority to the distance term, $\mu_1(t) >> \mu_4$ is chosen. The transformation of $A_{N,C}$ into an approximating automaton $A_{L,D}$ is based on linearizations in one (centered) point of R_1 and R_2, and the time discretization considers the following set of time-steps: $\triangle T = \{2, 4, 6, 10, 14\}$. Fig. 4 shows the trajectory obtained for a single optimization of $A_{L,D}$ over the full horizon $\{t_1, \ldots, t_H\}$. The case I refers to a test with a horizon $|T| = 5$, for which the sequence $\triangle t_{1,\ldots,4} = (14, 10, 10, 10)$ of time-steps was determined to be optimal (computed in 24 CPU-sec on a SPARC II workstation). The switching points for $A_{L,D}$ lie close to $R_{F,1}$ and $R_{F,2}$ such that the simulation of $A_{N,C}$ with the optimal input trajectories crosses the forbidden regions. To avoid this problem, two of the time-steps were manually fixed to $\triangle t_1 = \triangle t_2 = 10$ and the horizon increased to $|T| = 6$. The resulting state trajectory for $A_{L,D}$, denoted by $\boldsymbol{x}_{\mathtt{II}}(t)$ in Fig. 4, is far enough away from the forbidden regions to avoid that the corresponding state trajectory of $A_{N,C}$ crosses $R_{F,1}$ or $R_{F,2}$ [1] .

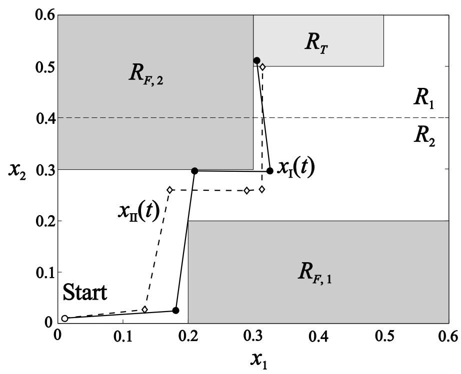

Fig. 4. Optimal state trajectories of $A_{L,D}$ for two cases (I: $|T| = 5$, II: $|T| = 6$).

The trajectories generated with the moving horizon approach are shown in Fig. 5: The state trajectory $\boldsymbol{x}(t)$ is the simulation result for $A_{N,C}$ obtained from the alternating procedure explained in Sec. 3.3. In each step the optimization was performed with a relatively short prediction horizon $n_P = 3$, and the target region is reached in 12 steps overall. The first time-step for each optimization was fixed to $\triangle t_1 = 2$ while the parameters $\triangle t_2$ and $\triangle t_3$ are chosen by the optimiza-

[1] A more sophisticated approach to ensure that the trajectory of $A_{N,C}$ lies outside of R_F is to determine an upper bound on the model deviation $\hat{f}_i = ||f_i - f_i^D||_\infty$ ($i \in \{1, \ldots, n\}$) and to enlarge the forbidden regions by moving their hyperplanes outside by $\hat{\boldsymbol{f}} \cdot \triangle t_{max}$.

tion algorithm from the set $\{2, 4, 6, 10, 14\}$. Due to the distance term in Ω, large values are assigned to $\triangle t_2$ and $\triangle t_3$ in most of the steps such that the prediction covers a relatively large time range despite of the small value of n_P. This enables that the optimizer can find a way around the forbidden regions, and in the last seven steps the target region R_T is reached within the prediction. The computation for this setting took 106 CPU-sec for the total number of 12 steps.

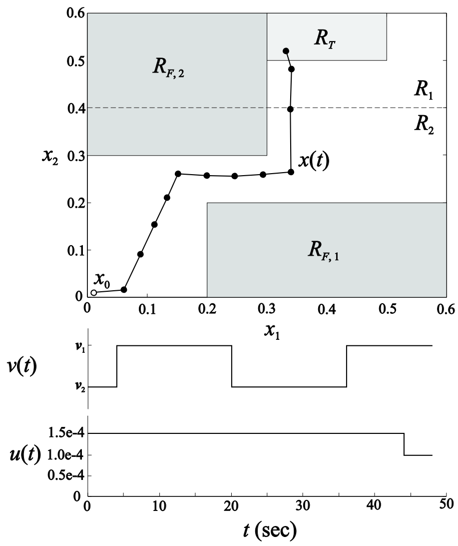

Fig. 5. State and input trajectories of $A_{N,C}$ for the moving horizon setting.

5. CONCLUSION

The paper presents a pragmatic approach to solve optimal control problems for systems with switching dynamics and discrete as well as continuous controls. Of course the reformulation of $A_{N,C}$ into $A_{L,D}$ is an approximation, but often the optimal input trajectories obtained for $A_{L,D}$ lead to state trajectories of $A_{N,C}$ which are sufficiently close to the optimal one. A drawback of the equation-based form of $A_{L,D}$ is the fact that the logical part of the model leads to a large number of binary variables, and the number of discrete options grows exponentially with the number of considered time points. While for a moving horizon strategy with a 'small' horizon n_P the complexity grows only linear with $|T|$, the short look-ahead horizon can lead to large deviations from the optimal solution. The approach of using variable time steps (which grow within the horizon P) allows covering larger time periods and thus enhances the chance to find a path around forbidden regions.

Recent work is focused on developing a tool support for the part marked as manual procedure in Fig. 1, and on investigating alternative methods to formulate the optimization model in order to improve the efficiency of the MIP solution.

The work presented is financially supported by the German Research foundation (DFG).

6. REFERENCES

Bemporad, A. and M. Morari (1998). Predictive control of constrained hybrid systems. In: *Preprints Int. Symp. on Nonlinear Model Predictive Control (Ascona)*. pp. 108–127.

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica* **35**(3), 407–427.

Branicky, M. S., V. S. Borkar and S. K. Mitter (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Trans. Automatic Control* **43**(1), 31–45.

Broucke, M., M.D. Di Benedetto, S. Di Gennaro and A. Sangiovanni-Vincentelli (2000). Theory of optimal control using bisimulations. In: *Proc. 3^rd^ Int. Workshop of Hybrid Systems (HSCC2000)*. Vol. 1790 of *LNCS*. Springer. pp. 89–102.

Buss, M., O. von Stryk, R. Bulirsch and G. Schmidt (2000). Towards hybrid optimal control. *Automatisierungstechnik* **9**, 448–459.

Galan, S. and P. I. Barton (1998). Dynamic optimization of hybrid systems. *Computers and Chemical Engineering* **22 (Suppl.)**, 183–190.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Managem. Science* **22**(4), 455–460.

Gokbayrak, K. and C. G. Cassandras (2000). Hybrid controllers for hierarchically decomposed systems. In: *Proc. 3rd Int. Workshop of Hybrid Systems (HSCC2000)*. Vol. 1790 of *LNCS*. Springer. pp. 117–129.

Hedlund, S. and A. Rantzer (1999). Optimal control of hybrid systems. In: *Proc. 38th Conf. Decision and Control (Phoenix)*.

Schweiger, C. A. and C. A. Floudas (1998). Process synthesis, design and control: A mixed integer optimal control framework. In: *Proc. 5th IFAC Symp. on Dynamics and Control of Process Systems*. pp. 189–194.

Slupphaug, O., J. Vada and B. A. Foss (1997). Model predictive control in systems with continuous and discrete control inputs. In: *Proc. American Control Conference*.

Stursberg, O. and S. Engell (2001). Optimized startup-procedures of processing systems. In: *Proc. 6th IFAC Symp. on Dynamics and Control of Process Systems*. pp. 231–236.

Stursberg, O. and S. Panek (2002). Control of switched hybrid systems based on disjunctive formulations. In: *Proc. 5th Int. Workshop of Hybrid Systems (HSCC2002), Stanford (CA)*.

Williams, H. P. (1978). *Model Building in Mathematical Programming*. 1st ed.. J. Wiley P.