Preprints of the 2nd IFAC Conf. on Analysis and Design of Hybrid Systems (Alghero, Italy), 7-9 June 2006

DYNAMIC OPTIMIZATION OF AN INDUSTRIAL EVAPORATOR USING GRAPH SEARCH WITH EMBEDDED NONLINEAR PROGRAMMING

Christian Sonntag [*,1] **Olaf Stursberg** [**]
Sebastian Engell [*]

[*] *Process Control Laboratory (BCI-AST)*
University of Dortmund, 44221 Dortmund, Germany.
[**] *Industrial Automation Systems (EI-LSR)*
Technical University of Munich, 80290 Munich, Germany.

Abstract: In this paper, we consider the task to start the operation of an industrial evaporation system. Rigorous modelling gives rise to a hybrid automaton with large nonlinear DAE-models that describe the continuous evolution in the discrete locations. The optimization problem is solved by a hierarchical procedure that consists of a branch-and-bound algorithm with embedded nonlinear dynamic optimization over a finite look-ahead horizon. Important elements of the algorithm are the introduction of a dynamic choice of the time intervals over which the controls are constant and of tailored penalty functions in order to obtain solutions which are close to infeasible trajectories.

Keywords: Automaton, hybrid systems, large-scale models, nonlinear programming, optimal control.

1. INTRODUCTION

Chemical processing systems usually exhibit nonlinear and, in case of changes of the physical state of the substances (e.g. from liquid to vapour), switched dynamics. In addition, the states and the inputs are constrained, and a considerable number of actuated inputs usually is of discrete nature, e.g. valves which can be either open or closed. Besides continuous feedback controllers, there are logic (switching) controls that establish sequential procedures and initiate exception routines if malfunctions of sensors or actuators or serious disturbances occur. An important function of logic controls is for example the safe shut-down of the plant in case of dangerous situations. In order to describe such automated processing plants, hybrid models with nonlinear dynamics are appropriate. However, the computation of optimal controls for such models is very challenging due to the usually large number of discrete degrees of freedom in connection with the nonlinear dynamics. As an example, the hybrid model of a simplified industrial evaporation system is described in this paper, and the task of optimal start-up of this plant is formulated. The evaporator is a simplified version of a benchmark example that was developed within the EU Network of Excellence HyCon in cooperation with an industrial partner.

Due to the complexity of the hybrid model, the application of techniques which evaluate the optimality conditions, see e.g. (Sussmann, 1999; Bran-

[1] Corresponding author: c.sonntag@bci.uni-dortmund.de. This work is supported by the EU-funded NoE HyCon and the Graduate School of *Production Engineering and Logistics* at University of Dortmund.

icky *et al.*, 1998; Shaikh and Caines, 2003), seems not very promising. It is preferable instead to either employ techniques based on simpler approximating models , e.g. (Bemporad *et al.*, 2000; Lincoln and Rantzer, 2001; Stein *et al.*, 2004), or to use numeric methods for optimizing the original dynamics such that fast convergence to a good solution is obtained, e.g. (Buss *et al.*, 2000; Barton and Lee, 2002). The approach for solving the optimal control problem for the evaporator presented in this paper is based on the technique introduced in (Stursberg, 2004*a*; Stursberg, 2004*b*). It consists of a graph search algorithm that fixes discrete degrees of freedom, and embedded nonlinear programming (NLP) is used to select locally optimal continuous inputs. The hybrid dynamics is evaluated within the NLP by hybrid simulation. For the evaporator, however, this algorithm does not produce satisfying result without further tuning – hence, two modifications are introduced here: the first determines a time discretization for selecting the inputs based on the progress in the state space, and the second introduces specific penalty functions that allow for trajectories of low costs which are close to the boundary of the feasible state set.

2. A HYBRID MODEL OF THE EVAPORATOR

In the processing industries, liquids are often concentrated by evaporating volatile solvents such that non-volatile components (often the products) are enriched in the liquid phase. Fig. 1 shows a scheme of the industrial-scale evaporation system considered in this paper (as a particular instance of the multi-stage system in (Sonntag and Stursberg, 2005)). During start-up, the initially empty evaporation vessel (A) is filled with cold liquid feed consisting of a non-volatile product, water, and alcohol. The heat supply to the vessel is realized by condensing hot steam in a heat exchanger (B). When the liquid starts to boil, vapor can be drained from the top of the vessel. If the product concentration meets desired purity requirements, the product can be continuously drained from the bottom of the vessel. The operation is controlled using four valves and two pumps (C). While the vapor drain and the liquid feed can be adjusted continuously by V_{V1} and V_1, the valves controlling the inflow of hot steam (V_{V2}) and the product drain (V_2) are switched between two discrete settings.

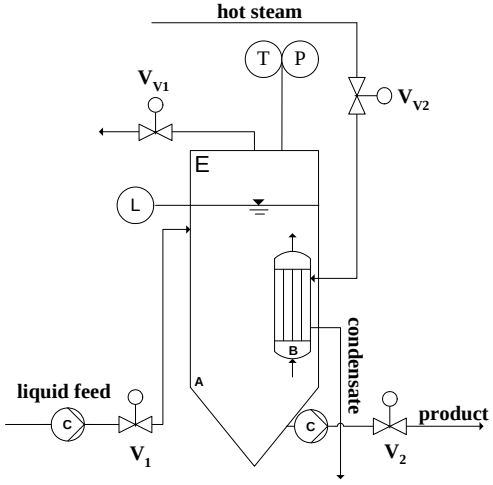

Fig. 1. Flowchart of the Evaporator.

2.1 Hybrid Automaton with DAE Dynamics

To model the evaporator, we extend the hybrid automaton specified in (Stursberg, 2004*a*) by an input availability mapping and continuous dynamics given as DAEs:

Definition 1. A hybrid automaton with DAE dynamics $HA = (Z, z_0, X, x_0, Y, U, V, \psi, inv, \Theta, g, j, f, l)$ consists of:

- the finite set of *locations* $Z = \{z_1, \ldots, z_{n_z}\}$;
- an *initial location* $z_0 \in Z$;
- the *continuous state space* $X \subseteq \mathbb{R}^n$ on which the vector of (differential and algebraic) *continuous variables* $x(t) \in X$ is defined (for simplicity of notation, we do not distinguish between variables and their evaluations here);
- an initialization of the continuous variables $x_0 \subseteq X$;
- the *continuous output space* $Y \subseteq \mathbb{R}^m$ on which the vector of *continuous output variables* $y(t) \in Y$ is defined;
- the *space of continuous inputs* $U = [u_1^-, u_1^+] \times \ldots \times [u_{n_u}^-, u_{n_u}^+]$ with $u_j^-, u_j^+ \in \mathbb{R}$, and the continuous inputs are denoted by $u(t) \in U$;
- a finite set of discrete inputs $V = \{v_1, \ldots, v_{n_d}\}$ with discrete inputs $v(t) \in V$ for which $v_j \in \mathbb{R}^{n_v}$;
- the *input availability mapping* $\psi : Z \rightarrow \mathcal{U} \times \mathcal{V}$, $\mathcal{U} \subseteq U$, $\mathcal{V} \subseteq V$ that assigns sets of available continuous and discrete inputs to each location $z \in Z$;
- the invariant function $inv : Z \rightarrow 2^X$ which assigns an *invariant set* $inv(z) \subseteq X$ to each $z \in Z$; we require that $x_0 \in inv(z_0)$;
- the set of *discrete transitions* $\Theta \subseteq Z \times Z$, and each transition is a pair $\theta = (z_i, z_j) \in \Theta$;
- the function $g : \Theta \rightarrow 2^X$ that associates a *guard set* $g(\theta) \subseteq X$ with each transition $\theta \in \Theta$; for each $z \in Z$, the guard sets of all transitions $\theta = (z, \bullet)$ are required to be pairwise disjoint;
- the *jump function* $j : \Theta \times X \rightarrow X$ which assigns an update $j(\theta, x) \in X$ of the continuous variables to each $\theta \in \Theta$ and $x \in g(\theta)$;
- the *flow function* $f : Z \times X \times U \times V \rightarrow \mathbb{R}^n$ that determines the continuous evolution in each

location $z \in Z$. For given input trajectories $u(t)$ and $v(t)$ and a location $z \in Z$, we assume that a unique solution of $M\dot{x}(t) = f(z, x(t), u(t), v(t))$ exists for an initialization $x(0) \in inv(z)$, $t \in \mathbb{R}^{\geq 0}$, and $M \in \mathbb{R}^{n \times n}$, $M_{ij} = 0 \; \forall \; i \neq j$; $i, j \in \{1, \ldots, n\}$; note that f constitutes a system of semi-explicit differential-algebraic equations (DAEs) if M is not regular;

- and the *output function* $l : X \rightarrow Y$ which uniquely maps the evaluation of the continuous variables to output variables $y(t) \in Y$. $\diamond$

Let $\Sigma = \bigcup_{z \in Z} \bigcup_{x \in inv(z)} (z, x)$ denote the set of valid *hybrid states* $\sigma = (z, x)$ of HA with $z \in Z$, $x \in inv(z)$. $T = \{t_0, t_1, t_2, \ldots\}$ is the ordered set of time points $t_k \in \mathbb{R}^{\geq 0}$ which contains the initial time $t_0 = 0$ and all points of time at which an input changes or a transition occurs. For $t_k \in T$, the hybrid states, inputs, and outputs are written as: $\sigma_k := (z_k, x_k)$, $u_k := u(t_k)$, $v_k := v(t_k)$ with $(u_k, v_k) \in \psi(z_k)$, and $y_k := y(t_k)$.

A continuous input trajectory defined on T is a sequence $\phi_u = (u_0, u_1, u_2, \ldots)$, and a discrete input trajectory is given by $\phi_v = (v_0, v_1, v_2, \ldots)$, meaning that u_k and v_k are piecewise constant on $[t_k, t_{k+1}[$. For given ϕ_u and ϕ_v, a sequence $\phi_\sigma = (\sigma_0, \sigma_1, \sigma_2, \ldots)$ of hybrid states $\sigma_k = \sigma(t_k) = (z_k, x_k)$ is a deterministic *feasible run* of HA iff:

- $\sigma_0 = (z_0, x_0)$, with $x_0 \in inv(z_0)$, and
- $\sigma_{i+1} = (z_{i+1}, x_{i+1}) \in \Sigma$ follows from $\sigma_i = (z_i, x_i) \in \Sigma$ according to:
  - $\overline{x}_{i+1} = x(t_{i+1})$ is the continuous state obtained from solving $M\dot{x}(t) = f(z_i, x(t), u(t), v(t))$ for $t \in [t_i, t_{i+1}]$ starting from $x_i = x(t_i)$, and $x(t) \in inv(z_i)$ must apply for all $t \in [t_i, t_{i+1}]$, and $x(t) \notin g(\theta)$ for $t \in [t_i, t_{i+1}[$ and for all $\theta = (z_i, \bullet)$.
  - if $\overline{x}_{i+1} \in g(\theta)$, $\theta = (z_i, z')$: $z_{i+1} := z'$, $x_{i+1} := j(\theta, \overline{x}_{i+1}) \in inv(z_{i+1})$;
    else $x_{i+1} := \overline{x}_{i+1}$, $z_{i+1} := z_i$. $\diamond$

2.2 HA Model for the Evaporator

Since product can be drained from the vessel only if the target region (see Sec. 3) is reached, it is sufficient to introduce four locations, i.e. $Z = \{z_{NE}, z_{NET}, z_E, z_{ET}\}$, where 'E' stands for evaporating, 'NE' for not evaporating, and 'T' for target. The continuous dynamics of all locations is modeled by a DAE system with four differential equations (modeling the the total masses of the three components m_1 (product), m_2 (water), and m_3 (alcohol), and the total inner energy U), and 13 algebraic equations that were derived using the assumption that the system is in thermodynamical equilibrium during evaporation. The algebraic variables are the mass fractions of the three components in the liquid phase (w_1, w_2, w_3) and in the vapor phase (ξ_1, ξ_2, ξ_3), the temperature in the evaporator T, the total mass of the liquid m_{liq}, the volume of the vapor phase V_{vap}, the energy transfer between the heat exchanger and the evaporator $\dot{Q}$, the mean temperature T_m and the pressure P_{HE} in the heat exchanger, and the flow rate of hot steam into the heat exchanger $\dot{F}_{HE}$. If the system is in location z_{NE}, the variables ξ_1, ξ_2, ξ_3, and V_{vap} are set to zero since a vapor phase does not exist. As an example for the high degree of complexity and nonlinearity of the DAE-system, the equation describing the equilibrium conditions for the mass of the alcoholic component looks like:

$$m_3 = m_{liq} \cdot w_3 + \frac{w_3 \cdot P_3^0(T) \cdot V_{vap}}{R \cdot T \cdot \left[\frac{w_1}{M_1} + \frac{w_2}{M_2} + \frac{w_3}{M_3}\right]} \quad (1)$$

with the universal gas constant R, the molecular weights M_1, M_2, and M_3, and $P_3^0(T)$ is a third-order polynomial in T.

Using the justifiable assumption that the liquid mass m_{liq} and the mass fractions w_1, w_2, and w_3 remain constant for any transition θ, all hybrid states $\sigma = (z, x)$ of the evaporator are uniquely determined by a reduced vector:

$$x_{red} = [w_1, w_2, T, L, P]. \quad (2)$$

The output variables L and P, which can be uniquely determined from the state variables, represent the liquid level and the pressure in the evaporator [2]. While L follows from the mass and density of the liquid, P is given by:

$$P = \frac{\frac{w_1}{M_1} \cdot P_1^0(T) + \frac{w_2}{M_2} \cdot P_2^0(T) + \frac{w_3}{M_3} \cdot P_3^0(T)}{\frac{w_1}{M_1} + \frac{w_2}{M_2} + \frac{w_3}{M_3}}, \quad (3)$$

where $P_1^0(T)$, $P_2^0(T)$, and $P_3^0(T)$ are third-order polynomials in T.

A bounded and reduced continuous state space can be defined as:

$$X_{red} = [0, 0.98] \times [0, 1] \times [300, 440] \times [0, 100] \times [0, 5], \quad (4)$$

and $x_{red} \in X_{red}$. (Units are omitted for abbreviation.) During start-up, the upper bounds given in Eq. 4 must not be exceeded for safety reasons.

The continuous inputs u_1 and u_2 represent the settings of the valves V_{V1} and V_1, and are defined on the range [0 %, 100 %], where 0 % means completely closed. For all $z \in Z$, both continuous inputs are available. A discrete input is defined as the vector $v_j = (V_2, V_{V2})$ and the discrete input space by $V = \{(0\,\%, 80\,\%), (0\,\%, 100\,\%), (11.5\,\%, 80\,\%), (11.5\,\%, 100\,\%)\}$. If $z \in \{z_{NE}, z_E\}$, the product drain valve must remain closed, such

[2] The dynamics of the process is nevertheless determined by the higher-order DAE-system since the latter cannot be solved explicitly for x_{red}, i.e. the use of x_{red} does not reduce the model complexity.

that the available discrete inputs are reduced to $\mathcal{V}_{z_{NE},z_E} = \{(0\,\%, 80\,\%), (0\,\%, 100\,\%)\}$. Otherwise, all discrete inputs are available.

Between all pairs of locations, transitions in both directions can occur, i.e. $|\Theta| = 12$. The transitions can be divided into three classes: (a) transitions occurring if the product concentration reaches or leaves the target region (see Sec. 3), (b) transitions occurring when the liquid begins or stops to evaporate, and (c) transitions for which both is true. While the guard condition for (a) only depends on the current value of w_1, the guard set for (b) is assumed to be given by $P = 0.4$, with P defined according to Eq. 3. The guard for (c) is the conjunction of the two previous ones. The invariants of the locations are bounded by the set of states defined by the guard conditions and the boundaries in Eq. 4. Although the state variables ξ_1, ξ_2, ξ_3, and V_{vap} are reset with discrete transitions of the hybrid model, the variables in x_{red} are not affected due to the modeling assumptions described above.

3. THE OPTIMAL CONTROL PROBLEM

The following type of optimal control problem is considered in this paper: Given are an initial state $\sigma_0 \in \Sigma$ and a target set $\Sigma_t \subset \Sigma$ with $\Sigma_t = \{(z_t, x) | \exists_1 z_t \in Z : x \in X_t \subset inv(z_t)\}$. It is assumed that the ordered set of time points $T = \{t_0, t_1, t_2, \dots, t_f\}$ is finite, and that the continuous and discrete inputs can only be changed at $t_k \in T_s \subset T$, while ϕ_σ remains defined on T. The set $\Phi_{u,s}$ contains all possible continuous input trajectories $\phi_u = (u_0, u_1, u_2, \dots)$ defined on T_s, and $\Phi_{v,s}$ contains all possible $\phi_v = (v_0, v_1, v_2, \dots)$. The control task is then to determine input trajectories ϕ_u^* and ϕ_v^* that lead to a feasible run ϕ_σ^* of HA from σ_0 into Σ_t such that a cost function Ω is minimized:

$$\min_{\phi_u \in \Phi_{u,s}, \phi_v \in \Phi_{v,s}} \Omega\left(t_f, \phi_\sigma\right) \tag{5}$$
$$s.t.\ \phi_\sigma = (\sigma_0, \dots, \sigma_f) \text{ with } \sigma_0 = (z_0, x_0),$$
$$\sigma_f := (z(t_f), x(t_f)) \in \Sigma_t.$$

The initial state of the evaporator is given by $x_{0,red} = [0.12, 0.85, 327, 1, 0.282]$, denoting a state with very low product concentration, low level, and no evaporation ($z_0 := z_{NE}$). The control task is to drive the system into the location $z_t = z_{ET}$ and the target region $X_{t,red} = [0.8, 0.84] \times [0.16, 0.2] \times [370, 420] \times [60, 64] \times [0.5, 4]$ in a time-optimal fashion.

Using the cost function $\Omega = t_f$ in (5) and optimizing over the complete time horizon $[t_0, t_f]$ is often computationally intractable since $|T_s|$ choices for u and v lead to an exponential growth of the solution space with increasing $|T_s|$. In this case, a substitute for Ω may be chosen which allows for an appropriate cost evaluation of trajectories also over shorter time horizons. A possible choice is a cost function which combines t_f with a notion of distance of any intermediate state σ_k to the target region Σ_t. It was found for the evaporator, however, that this choice for Ω does provide only solutions for which t_f is somewhat worse than the optimal value. Reasoning about the expected behavior of the system led to the following heuristically-chosen cost function:

$$\Omega(x_{s,red}) = \begin{cases} \alpha \cdot |w_{s,1} - w_{s,1,t}| + \beta \cdot |L_s - L_{s,max}| \\ \text{if } |w_{s,1} - w_{s,1,t}| > 0.09, \\ \beta \cdot |w_{s,1} - w_{s,1,t}| + \alpha \cdot |L_s - L_{s,t}| \\ \text{if } |w_{s,1} - w_{s,1,t}| \leq 0.09. \end{cases} \tag{6}$$

Note that before evaluating $\Omega(x_{s,red})$, all variables of x_{red} are scaled to the range $[0, 1]$ according to $x_{s,red} = D^{-1} \cdot x_{red} - c$, with $D = \text{diag}(0.98, 1, 140, 100, 5)$ and $c = [0, 0, 2.1429, 0, 0]^T$. In Eq. 6, $w_{s,1}$ is the scaled product concentration, $w_{s,1,t} = 0.804$ the scaled target concentration, L_s the scaled liquid level, $L_{s,max} = 1$ the upper bound for L_s, $L_{s,t} = 0.62$ is the scaled target level, and α and β denote weighting constants. The motivation for this choice of $\Omega(x_{s,red})$ is that the fastest way to increase $w_{s,1}$ should be to maximize the energy transfer $\dot{Q}$ between the heat exchanger and the evaporation vessel (and thus, the evaporation rate of the volatile components). The energy transfer increases with the difference between the liquid temperature T and the temperature of the hot steam. If the liquid level L is held close to its maximum value, a minimal temperature of the liquid is achieved. After the product concentration has reached a threshold which is close to its target value, the level has to be driven into the target region. In order to push the increase of $w_{s,1}$ which exhibits slow dynamics, the constant weights were chosen according to $\frac{\alpha}{\beta} = 4$ if $|w_{s,1} - w_{s,1,t}| > 0.09$. For $|w_{s,1} - w_{s,1,t}| \leq 0.09$, the weights were chosen according to $\frac{\beta}{\alpha} = 4$ to put the focus on driving L_s into the target.

4. THE OPTIMIZATION APPROACH

The approach used here for optimizing the start-up of the evaporator is based on the graph search algorithm presented in (Stursberg, 2004*a*; Stursberg, 2004*b*). This section first reviews the basic principle, and then describes necessary modifications to solve the case study.

4.1 Graph Search with Embedded NLP

The main idea of the approach is to separate the optimization of the discrete and continuous degrees of freedom by encoding the discrete choices

in an acyclic graph. For each node of the graph, optimal values for the continuous degrees of freedom are determined using nonlinear programming in which numerical simulation of the hybrid model over a constant time horizon is employed to evaluate the cost for the corresponding evolution of the system. Each node of the graph is characterized by a structure $n = (\phi_\sigma, \phi_u, \phi_v, c_a, c_p)$ which contains the state (ϕ_σ) and input (ϕ_u, ϕ_v) trajectories by which the hybrid state $\sigma_k = (z_k, x_k)$ assigned to the node was reached, the cost c_a that was accumulated on the path into σ_k, and, to determine the most promising nodes for further investigation, a prediction c_p of the cost for the remaining path into the target region. Two different techniques are used to prune the search graph: upper bounds of the accumulated cost are iteratively determined to remove branches that will lead to provably inferior solutions. Furthermore, if the hybrid states of two nodes are in close neighborhood, a cost comparison criterion is applied to prune the locally inferior node. The neighborhood of the hybrid state (z_k, x_k) of a node is modeled using an ellipsoidal set defined by

$$(x - x_k)^T \cdot P_v \cdot (x - x_k) \leq \epsilon, \qquad (7)$$

with a small ϵ.

4.2 Progress-Dependent Simulation Times

Due to the large variations of the gradients over X, and in particular due to the fact that $|\dot{L}| \gg |\dot{w}_1|$, the use of a constant simulation time is a bad choice for the evaporator. For example, large time periods drive L outside the permitted range for most inputs, while such steps are necessary to obtain the required change in w_1. Rather than using a fixed time to advance between nodes, we employ a criterion for terminating the simulation that is bound to the progress in X. If $\sigma_k = (z_k, x_k)$ is the initial hybrid state, the simulation is terminated if the continuous state trajectory reaches the boundary of a hyper-ellipsoid[3] defined by $(x - x_k)^T \cdot P_p \cdot (x - x_k) = r^2$.

4.3 Exclusion of Infeasible State Trajectories

As discussed in Sec. 3, $L_s(t)$ should be held close to 100 % for parts of the start-up procedure. In order to permit this behavior while preventing the generation of infeasible state trajectories ($L_s(t) > 1$ would terminate the simulation), the NLP step is modified as sketched in Fig. 2: It shows two state trajectories leading into the neighborhood of an invariant boundary. If $x_{red,k}$ is reached

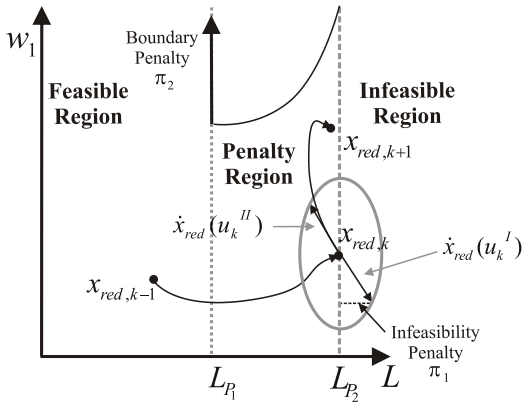

Fig. 2. Penalty computation for hybrid states near to invariant boundaries.

from $x_{red,k-1}$ during simulation, the latter is stopped and a new node is created. When the NLP step is then carried out for $x_{red,k}$, it is likely that the NLP solver uses a guess for u_k which leads into the infeasible region (marked as u_k^I in Fig. 2). In this case, the simulation would immediately stop again, and the solver would lack the information about the performance of this guess. To avoid this problem, the point $x_{red,k}$ is first projected onto an ellipsoid surrounding $x_{red,k}$ by using a numerical approximation of the gradient $\dot{x}_{red}(z, x_{red,k}, u_k^I, v_k)$. The distance of the resulting point to the boundary of the invariant then constitutes an infeasibility penalty π_1. As a result, the NLP solver would prefer a choice for u_k, for which $\dot{x}_{red}$ points in direction of the feasible region; then $\pi_1 := 0$. In Fig. 2, the choice u_k^{II} leads to $\dot{x}_{red}(z, x_{red,k}, u_k^{II}, v_k)$ and a feasible next state $x_{red,k+1}$. As an additional penalty, states close to the boundary of $inv(z_k)$ are penalized by a function $\pi_2 = a \cdot (\frac{L(t) - L_{P_1}}{L_{P_2} - L_{P_1}})^b$ with positive constants a and b (see Fig. 2). The two penalties π_1 and π_2 are only used within the NLP step, but not to assess the nodes within the search tree.

5. OPTIMIZATION RESULTS

Fig. 3 shows a projection of the optimization result for the start-up of the evaporator into the (w_1, T, L)-space. Nodes that have been explored are marked by an 'x', and pruned nodes are shown as a '+' (neighborhood) or an 'o' (cost). The solid line represents the state trajectory of the best solution found. The following parameterization of the search algorithm was used: The neighborhood ellipsoids (according to Eq. 7) were parameterized as $P_v = \mathrm{diag}(10^6, 10^6, 1, 100, 4 \cdot 10^4)$, $\epsilon = 1$, and the ellipsoid for the progress criterion were chosen small in the directions of w_1 and w_2, i.e. $P_p = \mathrm{diag}(1600, 1600, 0.01, 0.0025, 100)$, $r = 2$. Fig. 4 shows the continuous input trajectory corresponding to the best found solutions: The inputs are determined by the algorithm such that the

[3] In the general case, this criterion is used only within $inv(z_k)$; for the evaporator, however, it can be applied even for transitions occurring within $[t_k, t_{k+1}]$ for $t_k, t_{k+1} \in T_s$.

vapor drain through V_{V1} is maximized while L is held at 100 % using appropriate settings for V_1. Towards the end, V_1 is closed to drive the liquid level into the target region. V_2 is always closed since w_1 is the last variable to reach the target region, and V_{V2} remains open. During the optimization, 600 nodes were investigated, and 96 solutions were found. The complete optimization run took around 6 hours on a PC with Pentium-IV 2.8 GHz. The best solution was determined after 746 seconds, and using the input trajectories defined by this solution, the target region is reached after 13937 seconds.

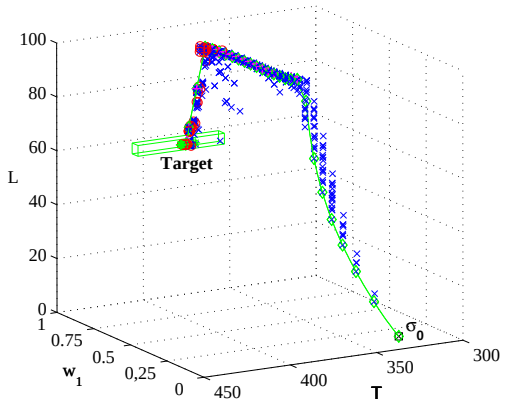

Fig. 3. Explored nodes and the state trajectory of the best solution.

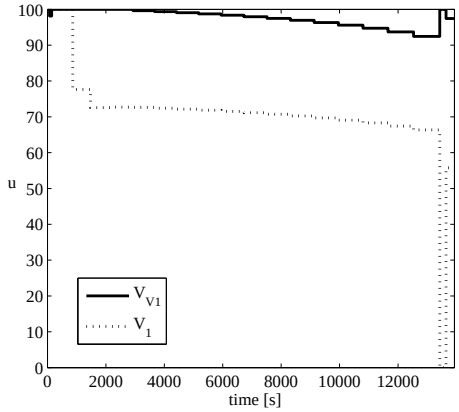

Fig. 4. Continuous input trajectories for the best solution.

6. CONCLUSIONS

The problem of optimizing the start-up of a hybrid model of an industrial-scale evaporator with complex nonlinear continuous dynamics was tackled using a graph search algorithm with embedded nonlinear programming. To obtain feasible and good solutions with the algorithm, two modifications had to be introduced: (a) a state-dependent termination criterion for the embedded hybrid simulation, and (b) a modified cost function for the NLP step to rule out infeasible evolutions while allowing optimization near or on the boundary of invariants.

The next steps of this research are as follows: We try to develop schemes that are able to find feasible solutions for similarly challenging problems without the requirement of first determining specifically tuned cost functions (as described in Sec. 3). In addition, we aim at extending the solution concept presented here to the multi-stage evaporator as described in (Sonntag and Stursberg, 2005).

REFERENCES

Barton, P.I. and C.K. Lee (2002). Modeling, simulation, sensitivity analysis and optimization of hybrid systems. *ACM Trans. Modeling and Comp. Simulation* **12**(4), 256–289.

Bemporad, A., N. A. Bozinis, V. Dua, M. Morari and E. N. Pistikopoulos (2000). Model predictive control: A multiparametric programming approach. In: 10^{th} *ESCAPE Symposium*. pp. 301–306.

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.

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

Lincoln, B. and A. Rantzer (2001). Optimizing linear system switching. In: 40^{th} *IEEE CDC*. pp. 2063–2068.

Shaikh, M.S. and P.E. Caines (2003). On the optimal control of hybrid systems. In: *Hybrid Systems: Comp. and Control*. Vol. 2623 of *LNCS*. Springer. pp. 466–481.

Sonntag, C. and O. Stursberg (2005). Optimally controlled start-up of a multi-stage evaporation system. Technical report for the network of excellence HyCon. University of Dortmund.

Stein, O., J. Oldenburg and W. Marquardt (2004). Continuous reformulations of discrete - continuous optimization problems. *Comp. and Chemical Eng.* **28**(10), 1951–1966.

Stursberg, O. (2004*a*). Dynamic optimization of processing systems with mixed degrees of freedom. In: *Proc.* 7^{th} *Int. IFAC Symp. Dyn. and Control of Process Systems*. number 164.

Stursberg, O. (2004*b*). A graph-search algorithm for optimal control of hybrid systems. In: 43^{rd} *IEEE CDC*. pp. 1412–1417.

Sussmann, H.J. (1999). A maximum principle for hybrid optimal control problems. In: 38^{th} *IEEE CDC*. pp. 425–430.