Pergamon

Computers chem. Engng Vol. 22, Suppl., pp. S221–S228, 1998

Printed in Great Britain

PII: S0098-1354(98)00058-1

0098-1354/98 $19.00 + 0.00

Modelling and Optimisation of General Hybrid Systems in the Continuous Time Domain

M.P. Avraam, N. Shah* and C.C. Pantelides

Centre for Process Systems Engineering
Imperial College of Science, Technology and Medicine
London SW7 2BY, United Kingdom

Abstract. This paper presents a mathematical formulation for the dynamic optimisation of hybrid processes described by general state-transition networks. In each state, the system behaviour is described by a set of differential and algebraic equations (DAEs). Transitions occur from one state to another whenever certain logical conditions are satisfied. The time horizon of interest is divided into a number of periods of variable duration, with the system potentially being in a different state in each period. Neither the initial nor the final state of the system, nor indeed any of its intermediate states, need to be known *a priori*. The resulting dynamic optimisation problem involves both continuous and discrete variables and is solved using a complete discretisation approach. Two examples of the application of the proposed methodology are presented. © 1998 Elsevier Science Ltd. All rights reserved.

INTRODUCTION

The vast majority of process systems of practical interest can be viewed as exhibiting hybrid behaviour, combining periods of continuous operation interspersed with discrete changes. For instance, thermodynamic phases appear and disappear, flow regimes switch from laminar to turbulent and vice versa, fluids in pipes change direction of motion, and processing equipment fails.

Recently, Pantelides (1995) described a modelling framework for such discrete/continuous processing systems based on the concept of *state transition networks*. At any given time, the system finds itself in exactly one of a number of states; its mathematical behaviour is then described by a given set of evolution DAE equations.

A *transition* from one state to another is triggered when a given logical condition is satisfied. In addition to the logical condition that triggers it, each transition is characterised by a set of constraints that determine the condition of the system in its new state just after the transition in terms of the condition in its previous state just before the transition.

This paper is concerned with the optimisation of hybrid systems described by general state-transition networks. This is a dynamic optimisation problem that aims to determine a set of time-dependent *controls* and time invariant *parameters* that optimise some performance index of the system under consideration over a certain time horizon. The duration of the horizon may be fixed or may be allowed to vary within given bounds.

The optimisation of hybrid systems may be viewed as a class of *multistage dynamic optimisation* problems in which the system goes through a number of "stages" (analogous to the "states" in the state-transition networks), each described by a different set of equations. In the process engineering literature, such problems have been considered by, for instance, Morison and Sargent (1984) and Vassiliadis *et al.* (1994a; 1994b). There is, however, an important difference between classical multistage systems and the hybrid processes of interest to this work. In our case, neither the actual number of stages that the system will go through during the time horizon of interest nor the identities of the individual stages are known *a priori*.

In the following section, we review the representation of hybrid processes in terms of state-transition networks. The subsequent section then introduces a novel mathematical formulation for the optimisation of such systems based on the division of time into a number of periods of varying duration, with the system being in a unique (albeit *a priori* unknown) state throughout each period. The next section outlines a method for the solution of the resulting mathematical problem. Finally, two examples are presented to illustrate some of the features of the hybrid process optimisation problem and its solution.

STATE-TRANSITION NETWORK REPRESENTATION OF HYBRID SYSTEMS

We consider a hybrid system subjected to a set $u(t)$ of time-varying external inputs and described by a set of time-invariant parameters ν (*e.g.* characterising the equipment geometry). At any one time, the system can exist in only one of a finite number

* Author to whom all correspondence should be addressed, e-mail: n.shah@ic.ac.uk fax: ++44-171-5946606

of states. In each state, the system is described by a set of continuously varying variables and a set of equations that relate these variables.

Many models have a unique state, *i.e.* the composition of the variable and equation sets remains unchanged throughout the operation of the system. On the other hand, discontinuities in the behaviour of a system will, in general, result in changes to either or both of these sets, giving rise to different states for the corresponding model. A *transition* from one state to another is triggered when logical conditions, involving the variables that describe the system in the former state, are satisfied.

State Characterisation

If we neglect any spatial or other non-temporal variation of properties in our system, each of the states $s \in \mathcal{S}$ will be characterised by:

(1) A set of continuous variables $x^{(s)}(t) \in \mathcal{X}^{(s)}$ and $y^{(s)}(t) \in \mathcal{Y}^{(s)}$.

(2) A set of ordinary differential and algebraic equations:

$$f^{(s)}(x^{(s)}(t), \dot{x}^{(s)}(t), y^{(s)}(t), u(t), \nu, t) = 0 \quad (1)$$

that determines the behaviour of the system when it is in state s. It is assumed that equations (1) determine unique values of $\dot{x}^{(s)}(t)$ and $y^{(s)}(t)$ for any given values $x^{(s)}(t) \in \mathcal{X}^{(s)}$, $u(t) \in \mathcal{U}$ and $\nu \in \mathcal{V}$.

(3) A set of additional equality and inequality constraints:

$$g_{eq}^{(s)}(x^{(s)}(t), \dot{x}^{(s)}(t), y^{(s)}(t), u(t), \nu, t) = 0 \quad (2)$$

$$h_{in}^{(s)}(x^{(s)}(t), \dot{x}^{(s)}(t), y^{(s)}(t), u(t), \nu, t) \leq 0 \quad (3)$$

that must hold at all times while the system is in state s.

(4) A set of possible transitions $\mathcal{T}_{ss'}$ to the same or other states $s' \in \mathcal{S}$.

We note that the set of time-varying external inputs $u(t)$ to which the system is subjected is the same for all its internal states s although it may be subject to different sets of constraints. The same is true for the set of time-invariant parameters ν.

Transition Characterisation

Each transition is characterised by:

(1) An initial state, s.

(2) A final state, s'.

(3) A set of $\mathcal{N}^{ss'}$ algebraic expressions of the form:

$$l^{(ss'n)}(x^{(s)}, \dot{x}^{(s)}, y^{(s)}, u, \nu, t), \forall n = 1, .., \mathcal{N}^{ss'} \quad (4)$$

The transition from state s to state s' is triggered if one or more of these expressions attains a negative value, *i.e.* if the disjunctive logical condition:

$$\bigvee_{n=1}^{\mathcal{N}^{ss'}} [l^{(ss'n)}(\cdot) < 0] \quad (5)$$

becomes true.

(4) A set of relations determining the condition of the system in state s' at time t immediately after the transition, from the conditions of the system in state s at time t^- immediately before it:

$$I^{(ss')}\,(\,x^{(s')}(t), \dot{x}^{(s')}(t), y^{(s')}(t), x^{(s)}(t^-), \dot{x}^{(s)}(t^-), y^{(s)}(t^-), u(t), \nu\,) = 0 \quad (6)$$

MATHEMATICAL FORMULATION

Given the representation of the hybrid system in terms of a state-transition network, we proceed to construct a mathematical model that expresses the behaviour in terms of differential and algebraic equalities and inequality constraints. In contrast to Dimitriadis *et al.* (1997), here we adopt a continuous representation of time. We define a set of periods ($k = 1, .., \mathcal{K}$) of variable duration. The start of the horizon is denoted as time τ_0 and the end of period k as τ_k. Transitions may occur *only* at the boundaries between periods within the time horizon.

The following binary variables are introduced to characterise the system behaviour:

$$X_{sk} = \begin{cases} 1 & \text{if the system is in state } s \text{ over period } k, \\ & \forall s \in \mathcal{S}, k = 1, .., \mathcal{K} \\ 0 & \text{otherwise} \end{cases}$$

$$L_{ss'kn} = \begin{cases} 1 & \text{if } l^{(ss'n)}(x^{(s)}, \dot{x}^{(s)}, y^{(s)}, u, \nu, \tau_k) < 0 \\ & \text{at time } \tau_k \\ & \forall s \in \mathcal{S}, (s, s') \in \mathcal{T}_{ss'}, n = 1, .., \mathcal{N}^{ss'}, \\ & k = 1, .., \mathcal{K} - 1 \\ 0 & \text{otherwise} \end{cases}$$

$$L_{ss'k} = \begin{cases} 1 & \text{if the transition from state } s \text{ to state } s' \\ & \text{takes place at time } \tau_k \\ & \forall s \in \mathcal{S}, (s, s') \in \mathcal{T}_{ss'}, k = 1, .., \mathcal{K} - 1 \\ 0 & \text{otherwise} \end{cases}$$

System Behaviour

The system can be in only one state over any period k. This is expressed mathematically as:

$$\sum_{s \in \mathcal{S}} X_{sk} = 1, \quad \forall k = 1, .., \mathcal{K} \quad (7)$$

While the system is in state s, the corresponding equations characterising the behaviour of the system hold in that state. This can be written mathematically as:

$$\mathcal{L}_F^{(s)}(1 - X_{sk}) \leq f^{(s)}(x^{(s)}(t), \dot{x}^{(s)}(t), y^{(s)}(t), u(t), \nu, t) \leq \mathcal{U}_F^{(s)}(1 - X_{sk}) \;,\; \forall s \in \mathcal{S}, k = 1, .., \mathcal{K} \quad (8)$$

where $\mathcal{L}_F^{(s)}$ and $\mathcal{U}_F^{(s)}$ are lower and upper bounds on the functions $f^{(s)}(\cdot)$. We note that, if X_{sk}=1, then the above reduces to $f^{(s)}(\cdot)$=0 and the equation holds. On the other hand, (8) is not constraining if X_{sk}=0.

Path Constraints

We now turn our attention to handling the equality and inequality path constraints (2) and (3) that may be associated with each state. We can express these as:

$$\mathcal{L}_G^{(s)}(1-X_{sk}) \le g_{eq}^{(s)}(x^{(s)}(t),\dot{x}^{(s)}(t),y^{(s)}(t),u(t),\nu,t) \le \mathcal{U}_G^{(s)}(1-X_{sk}),\quad \forall s\in\mathcal{S},k=1,..,\mathcal{K},t\in[\tau_{k-1},\tau_k] \quad (9)$$

and

$$h_{in}^{(s)}(x^{(s)}(t),\dot{x}^{(s)}(t),y^{(s)}(t),u(t),\nu,t) \le \mathcal{U}_H^{(s)}(1-X_{sk}),\ \forall s\in\mathcal{S},k=1,..,\mathcal{K},t\in[\tau_{k-1},\tau_k] \quad (10)$$

Here $\mathcal{L}_G^{(s)}$ and $\mathcal{U}_G^{(s)}$ are lower and upper bounds on the functions $g_{eq}^{(s)}(\cdot)$ while $\mathcal{U}_H^{(s)}$ is an upper bound on $h_{in}^{(s)}(\cdot)$. We note that, if $X_{sk}=1$, then the above constraints reduce to $g_{eq}^{(s)}(\cdot)=0$ and $h_{in}^{(s)}(\cdot)\le 0$ respectively. On the other hand, they are not constraining if X_{sk}=0.

Transition Triggering Conditions

The binary variable $L_{ss'kn}$ must take a value of one if the value of the algebraic expression $l^{(ss'n)}(\cdot)$ is negative at time τ_k; otherwise, it must be zero. This is achieved via the constraints:

$$\mathcal{L}_L^{(ss'n)}L_{ss'kn}+\epsilon \le l^{(ss'n)}(x^{(s)}(\tau_k),\dot{x}^{(s)}(\tau_k),y^{(s)}(\tau_k),u(\tau_k),\nu,\tau_k) \le \mathcal{U}_L^{(ss'n)}(1-L_{ss'kn}),\quad \forall k,s\in\mathcal{S},(s,s')\in\mathcal{T}_{ss'},n=1,..,\mathcal{N}^{ss'} \quad (11)$$

where $\mathcal{L}_L^{(ss'n)}$ and $\mathcal{U}_L^{(ss'n)}$ are lower and upper bounds on the functions $l^{(ss'n)}(\cdot)$, and ϵ a small positive tolerance. Note that, if $l^{(ss'n)}(\cdot)<0$, then $L_{ss'kn}$ must take a value of one in order not to violate the first inequality in (11). On the other hand, if $l^{(ss'n)}(\cdot)>0$, then $L_{ss'kn}$ is forced to zero so as not to violate the second inequality in (11).

Occurrence of Transitions

A transition will take place from state s to state s' at time $t=\tau_k$ if and only if the system is at state s *and* at least one of the logical conditions that may trigger the transition $s\longrightarrow s'$ is true. This can be expressed mathematically as:

$$L_{ss'k} \le \sum_{n=1}^{\mathcal{N}^{ss'}} L_{ss'kn},\quad \forall s\in\mathcal{S},(s,s')\in\mathcal{T}_{ss'},k=1,..,\mathcal{K}-1 \quad (12)$$

$$L_{ss'k} \le X_{sk},\quad \forall s\in\mathcal{S},(s,s')\in\mathcal{T}_{ss'},k=1,..,\mathcal{K}-1 \quad (13)$$

$$L_{ss'k} \ge L_{ss'kn}+X_{sk}-1,\ \forall s\in\mathcal{S},(s,s')\in\mathcal{T}_{ss'},\ n=1,..,\mathcal{N}^{ss'},k=1,..,\mathcal{K}-1 \quad (14)$$

Constraints (12) and (13) ensure that $L_{ss'k}$ will be zero (*i.e.* the transition $s\longrightarrow s'$ will *not* take place at time τ_k), *either* if all $L_{ss'kn}$, $n=1,..N^{ss'}$ are zero (*i.e.* none of the corresponding logical conditions is satisfied) *or* if X_{sk} is zero (*i.e.* the system is not in state s in the first place). On the other hand, if at least one of the $L_{ss'kn}$ variables is one *and* X_{sk} is also one, then (14) ensures that $L_{ss'k}=1$ (*i.e.* the transition must take place).

If transition $s\longrightarrow s'$ does take place at time τ_k, the system will find itself in state s' during period $k+1$. This is ensured by the constraint:

$$X_{s',k+1} \ge L_{ss'k},\ \forall s\in\mathcal{S},(s,s')\in\mathcal{T}_{ss'},k=1,..,\mathcal{K}-1 \quad (15)$$

For the mathematical model to describe the system behaviour in a unique and unambiguous manner, we must define what happens if *none* of the transitions associated with the current state s of the system is triggered. In this case, the system state during period $k+1$ is the same as that during period k. This is enforced by:

$$X_{s,k+1} \ge X_{sk}-\sum_{(s,s')\in\mathcal{T}_{ss'}} L_{ss'k},\ \forall s\in\mathcal{S},k=1,..,\mathcal{K}-1 \quad (16)$$

We assume that the expressions $l^{(ss'n)}$ triggering the transitions from each state s are such that at most one of the variables $L_{ss'k}$ can take a value of 1 for any given k; otherwise, the system behaviour described by our state-transition network ceases to be deterministic.

An additional constraint is required in order to prevent any transition from taking place *during* period k, thus ensuring that transitions occur *only* at the boundaries between periods. Thus, none of the logical conditions that trigger a transition can be satisfied during period k. This can be written mathematically as:

$$-\int_{\tau_{k-1}}^{\tau_k}\min\left(0,l^{(ss'n)}(\cdot)\right)dt \le M_k(1-X_{sk}) \quad (17)$$

where the "large" quantity M_k is given by $-\mathcal{L}_L^{(ss'n)}(\tau_k-\tau_{k-1})$. We note that if the system is in state s during period k (*i.e.* $X_{sk}=1$), the above constraint will ensure that the triggering function $l^{(ss'n)}$ will not take a negative value over any finite duration of time during period k.

System Condition Following a Transition

If a transition from state s to state s' takes place at time τ_k, the conditions in state s' have to be calculated based on the final values of the conditions in s. This is done by enforcing equations (6) at time τ_k:

$$\mathcal{L}_I^{(ss')}(1-L_{ss'k}) \le I^{(ss')}\left(x^{(s')}(\tau_k),\dot{x}^{(s')}(\tau_k),y^{(s')}(\tau_k),x^{(s)}(\tau_k),\dot{x}^{(s)}(\tau_k),y^{(s)}(\tau_k),u(\tau_k),\nu\right) \le \mathcal{U}_I^{(ss')}(1-L_{ss'k}),\ \forall s\in\mathcal{S},k=1,..,\mathcal{K}-1 \quad (18)$$

where $\mathcal{L}_I^{(ss')}$ and $\mathcal{U}_I^{(ss')}$ are lower and upper bounds on the functions $I^{(ss')}(\cdot)$. Constraint (18) ensures

that $I^{(ss')}(\cdot) = 0$ if $L_{ss'k} = 1$ (*i.e.* if the transition from state s to s' actually takes place), otherwise it is not constraining.

Bound Constraints

The control variables $u(t)$ of the system may be subject to upper and lower bounds. Both of these may depend on the current state s of the system:

$$\sum_{s\in\mathcal{S}} X_{sk}U_s^{min} \le u(t) \le \sum_{s\in\mathcal{S}} X_{sk}U_s^{max} \quad \forall t \in [\tau_{k-1}, \tau_k), \quad k = 1,..,\mathcal{K} \tag{19}$$

Bound constraints of the form:

$$\nu^L \le \nu \le \nu^U \tag{20}$$

may also be imposed on the time invariant parameters ν.

To avoid numerical problems arising from the duration of one or more periods vanishing to zero, a minimum non-zero duration is imposed on all periods. An upper bound on the period duration may also be imposed, if necessary:

$$\Delta_k^L \le \tau_{k+1} - \tau_k \le \Delta_k^U, \quad \forall k = 1,..,\mathcal{K}-1 \tag{21}$$

The total length of the time horizon (corresponding to time $\tau_{\mathcal{K}}$) may be subject to given lower and upper bounds:

$$\mathcal{H}^L \le \tau_{\mathcal{K}} \le \mathcal{H}^U \tag{22}$$

Initial Conditions

In general, the initial state of the system may not be known *a priori* but may be allowed to be any one of a subset $\mathcal{S}^0$ of all system states $\mathcal{S}$. This can be expressed mathematically as:

$$\sum_{s\in\mathcal{S}^0} X_{s1} = 1 \tag{23}$$

If the system is initially in a state $s \in \mathcal{S}^0$, the initial values of the system variables will generally have to satisfy a set of initial conditions which may take the form of a general set of equations $C^{(s)}(\cdot) = 0$. This leads to the constraints:

$$\mathcal{L}_C^{(s)}(1 - X_{s1}) \le C^{(s)}\left(x^{(s)}(0), \dot{x}^{(s)}(0), y^{(s)}(0), u(0), \nu\right) \le \mathcal{U}_C^{(s)}(1 - X_{s1}), \ \forall s \in \mathcal{S}^0 \tag{24}$$

where $\mathcal{L}_C^{(s)}$ and $\mathcal{U}_C^{(s)}$ are lower and upper bounds on the functions $C^{(s)}(\cdot)$. We note that, if X_{s1}=1, then the above reduces to $C^{(s)}(\cdot)$=0 and the equation holds. On the other hand, (24) is not constraining if X_{s1}=0.

End Point Conditions

The final state of the system may be constrained to be one of a subset $\mathcal{S}^F$ of all system states $\mathcal{S}$. This can be expressed mathematically as:

$$\sum_{s\in\mathcal{S}^F} X_{s\mathcal{K}} = 1 \tag{25}$$

If the system finds itself in a state s at the end of the time horizon, the final values of the corresponding variables may generally have to satisfy end-point inequality constraints of the form $E^{(s)}(\cdot) \le 0$. This leads to the constraint:

$$E^{(s)}(x^{(s)}(\tau_{\mathcal{K}}), \dot{x}^{(s)}(\tau_{\mathcal{K}}), y^{(s)}(\tau_{\mathcal{K}}), u(\tau_{\mathcal{K}}), \nu, \tau_{\mathcal{K}}) \le \mathcal{U}_E^{(s)}(1 - X_{s\mathcal{K}}) \ , \ \forall s \in \mathcal{S}^F \tag{26}$$

where $\mathcal{U}_E^{(s)}$ are upper bounds on the functions $E^{(s)}(\cdot)$. We note that, if $X_{s\mathcal{K}} = 1$, then the above reduces to $E^{(s)}(\cdot) \le 0$. On the other hand, (26) is not constraining if $X_{s\mathcal{K}} = 0$.

Objective Function

We assume that the objective function to be minimised generally comprises contributions from the operation of the system over each period, plus a fixed contribution that is a function $\Psi(\nu)$ of the time invariant parameters ν only.

The operating cost incurred over a period k during which the system is in state s can be expressed as a function $\Phi(.)$ of the final values of the variables in that period[1].

The overall objective function can be written as:

$$\min_{\tau_k, u(t), \nu} \Psi(\nu) + \sum_{k=1}^{\mathcal{K}} \sum_{s\in\mathcal{S}} X_{sk}\Phi^{(s)}\left(x^{(s)}(\tau_k), \dot{x}^{(s)}(\tau_k), y^{(s)}(\tau_k), u(\tau_k), \nu\right) \tag{27}$$

SOLUTION METHOD

The mathematical formulation developed in the previous section is a dynamic optimisation problem involving both continuous and discrete variables. In this section, we consider some issues regarding the solution of this type of problem.

Complete Discretisation Approach

As with all dynamic optimisation problems, the first issue that has to be addressed is the infinite dimensionality of some of the constraints and unknowns. For the purposes of this paper, we choose to do this using a complete discretisation approach (Cuthrell and Biegler, 1987).

In particular, we divide each period $k = 1,..,\mathcal{K}$ into $\mathcal{D}$ finite elements. These $\mathcal{K}\mathcal{D}$ elements define a grid of points $0,..,\mathcal{K}\mathcal{D}$. We then construct piecewise polynomial approximations of degree m for each of the variables $x^{(s)}(t)$ and $y^{(s)}(t)$, $\forall s \in \mathcal{S}$ and of degree m' for variables $u(t)$ over these elements.

The discretised constraints are obtained as follows:

- The constraints (8), (9), (10) and (19) for each period k are enforced at:
  - the grid points $(k-1)\mathcal{D},..,k\mathcal{D}-1$ for $k = 1,..,\mathcal{K}-1$, and points $(\mathcal{K}-1)\mathcal{D},..,\mathcal{K}\mathcal{D}$ for the last period;

[1]This may necessitate the introduction of new variables that are the integrals of the instantaneous operating cost over each period.

– at m collocation points within elements $(k-1)\mathcal{D}, .., k\mathcal{D}-1$.

- The constraints (11) and (18) relating to transitions between period k and $k+1$ are enforced at grid point $k\mathcal{D}$.
- The integrals in constraints (17) prohibiting transitions during period k are approximated in terms of the variable values at grid points $(k-1)\mathcal{D}, .., k\mathcal{D}$, and at the collocation points within the corresponding elements.
- The initial condition constraints (24) are enforced at the first point (point 0) on the grid.
- The end-point constraints (26) are enforced at the last point (point $\mathcal{K}\mathcal{D}$) on the grid.
- Continuity on the differential variables $x^{(s)}$ is enforced at points $(k-1)\mathcal{D}+1, .., k\mathcal{D}-1$ within period k.

The remaining constraints in the formulation are already finite dimensional and, therefore, do not need to be discretised.

The durations of the $\mathcal{D}$ finite elements in each period are allowed to vary (subject to their total duration being equal to the duration of the period) so as to provide some control of the approximation error. To this end, the differential equations are not enforced at the right boundary of each element, and their residual there is used to provide an error estimate (Vasantharajan and Biegler, 1990).

Dealing with Problem Degeneracy

One problem faced by our formulation is that, in general, the number $\mathcal{K}$ of periods in the optimal solution cannot be determined *a priori*. This is entirely analogous to already well-known problems in other areas such as process scheduling formulations using continuous representations of time (Zhang, 1995).

The standard approach with dealing with the above problem is to set $\mathcal{K}$ to a value that is likely to exceed the number of transitions taking place in the optimal solution. If the optimal solution obtained with a given $\mathcal{K}$ actually makes use of all $\mathcal{K}$ periods allowed, then the computation may be repeated with a higher value of $\mathcal{K}$.

The problem with the above approach is that having redundant periods in the problem causes degeneracy as there is an infinite number of ways in which a period can be divided into two or more consecutive periods with no transition taking place at the boundaries between them.

One way of dealing with the above difficulty is to force all inactive periods to the end of the time horizon. This is achieved by the additional constraint:

$$\sum_{(s,s')\in\mathcal{T}} L_{ss',k+1} \leq \sum_{(s,s')\in\mathcal{T}} L_{ss'k}, \quad \forall k = 1, .., \mathcal{K}-1 \tag{28}$$

and the following modification of constraint (21):

$$\begin{aligned} \Delta^L \sum_{(s,s')\in\mathcal{T}} L_{ss'k} &\leq \tau_k - \tau_{k-1} \\ \leq \quad \Delta^U \sum_{(s,s')\in\mathcal{T}} L_{ss'k}, &\quad \forall k = 2, .., \mathcal{K} \end{aligned} \tag{29}$$

where $\mathcal{T}$ is the set of all transitions in the state-transition network:

$$\mathcal{T} \equiv \bigcup_{s,s'\in\mathcal{S}} \mathcal{T}_{ss'} \tag{30}$$

We note that the quantity $\sum_{(s,s')\in\mathcal{T}} L_{ss'k}$ takes a value of 1 if a transition actually takes place at the end of period k, and is zero otherwise. Hence, constraint (28) simply states that, if no transition takes place at the end of period k, then no transition will be allowed to take place thereafter either. Moreover, constraint (29) will force the duration of periods $k+1, .., \mathcal{K}$ to zero.

System Decomposition

In most process systems of practical interest, most of the equations $f^{(s)}(\cdot)$ describing the transient behaviour of the system are, in fact, common for all states s. The same is true for most (or even all) of the describing variables $x^{(s)}(t)$ and $y^{(s)}(t)$. In such cases, we can take the subset of common equations $\bar{f}$ outside the general constraint (8) and simply write:

$$\bar{f}\left(\bar{x}(t), \dot{\bar{x}}(t), \bar{y}(t), u(t), \nu, t\right) = 0, \quad \forall k = 1, .., \mathcal{K} \tag{31}$$

where $\bar{x}$ and $\bar{y}$ are the subsets of the differential and algebraic variables that are common to all states in the problem. This results in substantial reduction in the number of constraints in comparison with (8) since it is written once rather than for each state $s \in \mathcal{S}$.

Furthermore, most practical hybrid processes can be viewed as comprising a number of independent state-transition sub-networks (Pantelides, 1995). For instance, in a process comprising Λ flash units, we could have a separate sub-network for each unit, including only the states and transitions relevant to that unit. The set of states $\mathcal{S}$ in the overall state-transition network is the cartesian product of the sets of states $\mathcal{S}_\lambda$ in the sub-networks $\lambda = 1, .., \Lambda$. This would give rise to[2] $\prod_{\lambda=1}^{\Lambda} |\mathcal{S}_\lambda|$ distinct states (*i.e.*, 3^Λ for the above example).

It is, in fact, more efficient in such cases to characterise the state of the system directly in terms of the states of the individual sub-networks. For instance, the variables X_{sk} would be replaced by variables $X_{s\lambda k}$ defined to take a value of 1 if sub-system λ is in its state $s \in \mathcal{S}_\lambda$ during period k, and 0 otherwise. The advantage of this is that the total number of variables is now proportional to $\sum_{\lambda=1}^{\Lambda} |\mathcal{S}_\lambda|$ rather than to $\prod_{\lambda=1}^{\Lambda} |\mathcal{S}_\lambda|$. The various constraints may also

[2]Here $|A|$ denotes the cardinality of a set A.

be disaggregated in this manner; for instance, the system behaviour constraints (8) can be written as:

$$\begin{aligned} \mathcal{L}_F^{(s\lambda)}(1 - X_{s\lambda k}) \quad &\leq \\ f^{(s\lambda)}(x^{(s\lambda)}(t), \dot{x}^{(s\lambda)}(t), y^{(s\lambda)}(t), u(t), \nu, t) \quad &\\ \leq \quad \mathcal{U}_F^{(s\lambda)}(1 - X_{s\lambda k}), \quad &\\ \forall \lambda = 1, .., \Lambda, s \quad \in \quad \mathcal{S}_\lambda, k = 1, .., \mathcal{K} \quad &(32) \end{aligned}$$

Implementation

The complete discretisation method described above converts the original infinite dimensional dynamic optimisation problem into a mixed integer nonlinear programming (MINLP) problem. In the examples described below, this is solved using an outer approximation algorithm implemented within the DICOPT++ (1992) code interfaced to GAMS.

NUMERICAL EXAMPLES

This section presents two examples of the application of our algorithm. The first one is a hybrid variant of a standard continuous test problem used widely in the optimal control literature. Despite its simplicity, it demonstrates clearly that, in hybrid systems, the sequence of states that the optimal solution trajectory follows cannot be predicted *a priori*, and that, in fact, it may even depend on the numerical values of the problem parameters. The second problem is taken from a process engineering application.

Example 1: Optimal Car Motion

We consider a car equipped with a turbo accelerator. The latter is activated if the car exceeds a certain critical velocity V_1 and is deactivated automatically if the velocity falls below another critical value V_2 ($< V_1$).

The state transition network representation of this system is shown in figure 1. It can be seen that the position $x(t)$ and velocity $v(t)$ are governed by Newton's law of motion. The effect of the turbo accelerator is to make the actual acceleration three times its nominal value $a(t)$ The variable $c(t)$ represents the cumulative cost of the car operation up to and including time t. The cost of operation per unit time is P_n and P_t in the normal and turbo modes respectively.

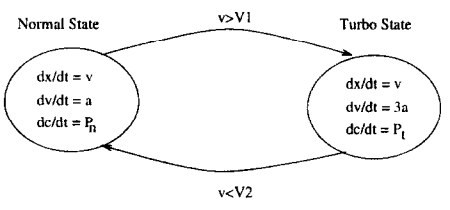

Figure 1: Car system state transition representation

The car has to cover a distance of 2000 m starting and finishing at zero velocity. We, therefore, have the initial conditions:

$$x(0) = 0; \quad v(0) = 0; \quad c(0) = 0 \qquad (33)$$

and the end-point constraints:

$$x(t_f) = 2000; \quad v(t_f) = 0 \qquad (34)$$

where t_f denotes the final time.

The velocity must never exceed 50 m/s at any time during the travel. This leads to the inequality path constraint:

$$0 \leq v(t) \leq 50, \quad \forall t \in [0, t_f] \qquad (35)$$

The only control available is the nominal acceleration $a(t)$ which is bounded as follows:

$$-5 \leq a(t) \leq 5, \quad \forall t \in [0, t_f] \qquad (36)$$

Finally, the optimisation objective function is:

$$\min \; t_f + c(t_f) \qquad (37)$$

We consider two different cases for this problem. In case 1, the running costs in both states are the same ($P_n = P_t = 1$) while in case 2, operating in the turbo state is much more expensive than in the normal one ($P_n = 1$, $P_t = 10$). In both cases, the critical switching velocities are taken as:

$$V_1 = 30 \; m/s; \quad V_2 = 20 \; m/s.$$

We also note that the equations (6) that determine the system condition following a transition from the normal to the turbo regime and vice-versa are provided by the continuity of the differential variables $x(t)$ and $v(t)$.

The problem is solved using the complete parameterisation approach described earlier. We set the number of periods $\mathcal{K}$ to 10, each divided into 5 finite elements of variable duration. Two interior collocation points are used in each element ($m = 2$) and a piecewise constant profile is assumed for the acceleration control variable (m'=0). In the optimal solution of case 1, the turbo regime is entered; once the velocity reaches its upper bound of 50 m/s, the acceleration is set to zero maintaining this maximum velocity until it is necessary to start decelerating to bring the car back to a stop at its final destination. This is to be expected as there is no financial penalty associated with operating in the turbo mode relative to the normal one. The system therefore follows the transition sequence Normal $\rightarrow$ Turbo $\rightarrow$ Normal.

In case 2, the optimal solution maintains the velocity just under the value of 30 m/s at which the switching to the turbo mode takes place. This is preferable since the cost of operating in the turbo mode far outweighs the savings in travel time that can be achieved.

It is worth noting that the number of periods used (10) in these calculations far exceeds the necessary minimum values for the two cases (3 and 1 respectively). However, this does not appear to cause any problems for the solution algorithm.

Example 2: Safety analysis of a surge drum

Dimitriadis *et al.* (1997) demonstrate how process safety verification can be viewed as a dynamic optimisation problem in which the various disturbances are manipulated to bring the system from any point in a given initial region to any point in one of the dangerous (or otherwise undesirable) regions in the minimum possible time. We apply this approach to a safety analysis problem, first studied by Srinivasan *et al.* (1997), using our hybrid system optimisation algorithm.

The system comprises a surge drum (see figure 2) in a sour water stripper plant. Sour water containing some hydrocarbon oil enters the main reservoir where the lighter oil separates from the aqueous phase. The top oil layer flows over the weir and is collected in the side oil reservoir.

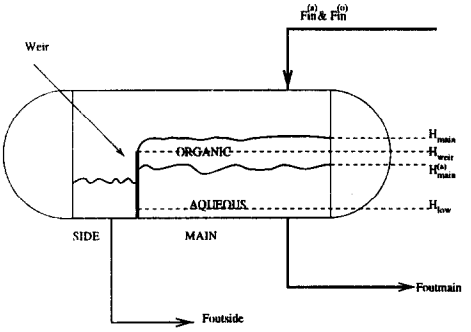

Figure 2: Schematic of the surge drum

The state transition network model is shown in figure 3. At any time, the system can be in one of the following four states:

- *High*
  The interface between aqueous and organic phases in the main reservoir is above the weir height H_{weir} (1.55m).
- *Normal*
  The interface between aqueous and organic phases in the main reservoir is below the weir height H_{weir} but the total height is above H_{weir}.
- *Low*
  The total height of liquid in the main reservoir is below H_{weir} but the aqueous phase height is above H_{Low}.
- *Empty*
  The aqueous phase height in the main reservoir is below H_{Low}.

The hazards that may occur in this system arise as follows:

- Reduced or no flow into the drum can cause the interface level to fall. When the total liquid height (H_{main}) falls below H_{Low}, the slop oil content in the main outflow increases sharply. This can eventually lead to build-up of the oil in the stripper, gumming up the trays.

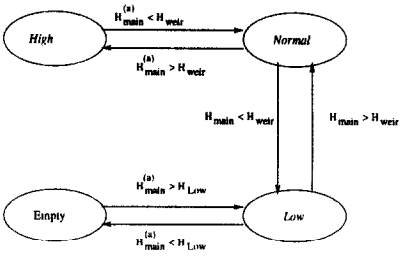

Figure 3: State transition representation of the surge drum

- If the aqueous height in the main reservoir rises above the top of the weir, then aqueous solution will flow into the organic outlet via the side reservoir.

Therefore hazards are associated with the system being in the states *Empty* and *High*.

The system disturbances to be treated as optimisation variables are the initial heights of the two liquid phases, $H^{(o)}_{main}(0)$ and $H^{(a)}_{main}(0)$, and the inlet flowrates of the two liquid phases, $F^{(o)}_{in}(t)$ and $F^{(a)}_{in}(t)$.

The behaviour of the system in each state is described by material balances and gravity driven outflows which assume that the phases are immiscible. There are some qualitative differences to the models in each state. In the *Normal* state, there is no organic phase in the main outflow and no aqueous phase in the side outflow. In the *Low* phase, there is no flow in the side outlet. In the *Empty* state, it is assumed that the flow in the main outlet is entirely organic. Finally, in the *High* state, there is an aqueous flow over the weir and in the side outlet.

The system is initially assumed to be in the *Low* state. The initial conditions of the system are not specified precisely but are bounded as follows:

$$1.00 \leq H^{(o)}_{side}(0) \leq 1.25\ m \tag{38}$$

$$0.20 \leq H^{(o)}_{main}(0) \leq 0.34\ m \tag{39}$$

$$0.50 \leq H^{(a)}_{main}(0) \leq 0.70\ m \tag{40}$$

The potential disturbances entering the system are bounded as follows:

$$0 \leq F^{(a)}_{in}(t) \leq 10^{-2}\ m^3/s \tag{41}$$

$$0 \leq F^{(o)}_{in}(t) \leq 10^{-4} m^3/s \tag{42}$$

The objective is to determine whether, under the influence of the disturbances, the system can enter an unsafe state, and if so to determine the minimum time required to do so.

The problem is formulated using ten periods, five intervals per period and three collocation points per interval. The inlet flowrates were assumed to be of piecewise constant form. The problem is again solved

using DICOPT++, requiring about 1 CPU min on a Sun Ultra workstation.

The results indicate that the fastest route to a hazardous situation occurs when the aqueous inlet flowrate is at its upper bound. The system passes through the transition sequence *Low* → *Normal* → *High*, arriving at the unsafe state after 2713 seconds of operation. The initial conditions are selected such that $H^{(o)}_{main}(0)$ and $H^{(a)}_{main}(0)$ are at their upper bounds. The value of $H^{(o)}_{side}(0)$ is less important in this solution; it is set at 1.003 m.

The plot in figure 4 shows the variation of the height of the aqueous and organic phases in the main reservoir with time.

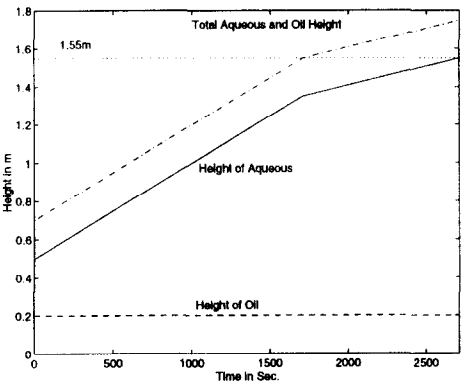

Figure 4: Time profles of heights in main reservoir

CONCLUDING REMARKS

This paper has considered the formulation and solution of dynamic optimisation problems for hybrid processes described by general state-transition networks. Since the macroscopic transient behaviour of most processes of practical interest is described by such networks, the effective solution of this problem is of paramount importance to process design and operation.

To our knowledge, this is the first time that the optimisation of such problems has formally been considered in the literature. The major technical difficulty is that the sequence of states that the system trajectory follows is generally not known *as priori* and is, in fact, determined by the values of the optimisation decision variables. We have shown that this complication can be handled effectively through the introduction of discrete variables.

In this paper, we have adopted a complete discretisation approach for the solution of the underlying mathematical problem. Notwithstanding the simplicity of its implementation, this type of technique also has well-known deficiencies, especially with respect to the size of problem that can be solved. On the other hand, it is worth noting that a naive approach based on the direct application of control vector parameterisation (see, for instance, Vassiliadis *et al.* (1994a)) where the sequence of states in the system trajectory for given values of the optimisation parameters is determined by the integration of the underlying hybrid dynamic model does not consistently lead to satisfactory results. The reason is that this integration does not provide the optimiser with information regarding any *other* possible trajectory. Our current work is, therefore, concerned with the development of alternative solution methods.

REFERENCES

Gams Development Corp., *GAMS/DICOPT - A solver for Mixed Integer Nonlinear Programming.* GAMS Development Corporation (1992).

Cuthrell, J. E. and L. T. Biegler, On the optimization of differential-algebraic process systems. *AIChE*, **33**, 1257–1270 (1987).

Dimitriadis, V. D., N. Shah, and C. C. Pantelides, Modelling and safety verification of discrete/continuous processing systems using discrete time domain models. *AIChE*, **43**, 1041–1059 (1997).

Morison, K. R. and R. W. H. Sargent, Optimization of multistage processes described by differential-algebraic equations. In 4^{th} *IIMAS Workshop on "Numerical Analysis"*. Guanajuato, Mexico (1984).

Pantelides, C. C., Modelling, simulation and optimization of hybrid processes. In *Workshop on "Analysis and Design of Event-Driven Operations in Process Systems"*. London (1995).

Srinivasan, R., V. D. Dimitriadis, N. Shah, and V. Venkatasubramanian, Integrating knowledge-based and mathematical programming approaches for process safety verification. *Computers chem. Engng.*, **S21**, S905–S910 (1997).

Vasantharajan, S. and L. T. Biegler, Simultaneous strategies for optimization of differential-algebraic systems with enforcement of error criteria. *Computers chem. Engng*, **14**, 1083–1100 (1990).

Vassiliadis, V. S., R. W. H. Sargent, and C. C. Pantelides, Solution of a class of multistage optimization problems. 1. problems without path constraints. *Ind. Eng. Chem. Res.*, **33**, 2111–2122 (1994a).

Vassiliadis, V. S., R. W. H. Sargent, and C. C. Pantelides, Solution of a class of multistage optimization problems. 2. problems with path constraints. *Ind. Eng. Chem. Res.*, **33**, 2123–2133 (1994b).

Zhang, X., *Algorithms for optimal process scheduling using nonlinear models.* PhD thesis, University of London (1995).