# Some issues concerning definition and expression of state for hybrid systems under supervision

O. Naud

# Some issues concerning definition and expression of state for hybrid systems under supervision

**Olivier NAUD**
UMR ITAP
Cemagref
P.O Box 5095, 34196 Montpellier Cedex 5, France
Phone: +33-(0)4 67 04 63 70, Fax: +33-(0)4 67 63 57 95,
E-mail: firstname.last@cemagref.montpellier.fr

**Abstract - A hybrid model can be seen as a set of Differential Algebraic Equations (DAEs) with a set of switches or as a Discrete Event System (DES) where DAEs are assigned to each discrete state. The first form, that we will call Switched-DAEs, is convenient for simulation and physical analysis. It can be based on a power port approach like bond-graphs. The second form, hybrid automaton, allows for analysis on an abstract level and evaluation of complexity in regards to system supervision. Supervision also often requires to handle qualitative information, as knowledge inserted in the model, and as generated information towards a Man-Machine Interface (MMI). We consider here the case where qualitative information is added on Switched-DAEs with labelled intervals defined on the continuous variables. We also choose the hybrid automaton formalism for a supervision-oriented model. It is calculated from an original Switched-DAEs system. We discuss here the semantics of discrete transitions for the automaton, and under which conditions the discrete automaton is an appropriate abstraction of the dynamics of the hybrid system.**

*Keywords*— **causal graph 'hybrid systems' qualitative intervals**

## I. INTRODUCTION

Among the tasks which a supervision system performs, we can distinguish 'observation', 'communication' and 'correction'. This 'observation' function will be defined here as "identifying current working state from current information and previous state in order that 'communication' and 'correction' tasks can be performed with this state as an input". If we consider the case of mechatronic systems, the 'correction' task requires a model including a set of corrective actions, criteria of choice and control of continuous dynamics. The working state has then to be expressed in a hybrid system framework (such as defined in [ALU 95] or [SÖD 95]). The 'communication' task, when there is a man-machine interface, requires symbolic, qualitative, information. A common strategy to attach symbolic information to continuous variables is to use intervals [TRA 97], [DAN 95]. We have also benefited from Kuipers' ideas about qualitative reasoning about physical systems [KUI 94] and we have chosen a 'qualitative hybrid system' framework for supervision. The rest of the paper will concern only the 'observation' part of supervision and focus on semantics of states and transitions.

A hybrid model can be seen as a set of Differential Algebraic Equations (DAEs) with a set of switches. Let us call this form *Switched-DAEs*. Switches include external logic inputs and logical conditions based on values of continuous variables. The value of switches determine which equations are currently valid. Switched-DAEs are convenient for simulation and physical analysis. They can be based on a power port approach like bond-graphs [STR 94], [NAU 05]. It is worth to mention that the generic physical mecanisms provided by bond-graphs have also been found well suited to the Artificial Intelligence's field of Knowledge-based problem solving [TOP 91]. In our case, qualitative information will be attached to variables thanks to labelled intervals, that we call *modalities* [NAU 05]. The hybrid system can also be seen as a Discrete Event System (DES) where DAEs are assigned to each discrete state, as a hybrid automaton or other hybrid extensions of DES paradigms [GUÉ 01]. For a large range of supervision systems, it is desirable that the DES can be used as an abstraction of the detailed hybrid system. In [LUN 01], *supervisory control* requires a complete hybrid model, but the 'global view' is included in the DES part (in this case a Petri net) of the model. We will discuss in the following how to compute an automaton from *Switched-DAEs with modalities*, and which semantics should be attached to automaton state and transitions. We will before briefly introduce our formalism called *half-causal graph*. This graph and a set of annotations have been designed to represent Switched-DAEs.

## II. THE HALF-CAUSAL GRAPH

The half-causal graph [NAU 05] is essentially an alternative graph to bond-graph [KAR 90] for power-port based physical analysis. A power-port connection is based on conservation of power with two variables, generalised flow and generalised effort. Such connection does not impose causal direction : port inputs and outputs are not fixed *a priori*. There is nevertheless a constraint on causal ordering : a mechanism or component cannot impose to another both flow and effort. The half-causal graph was created to make explicit name of variables on a power-port based graph. The choice of variables is guided by a decomposition in components. We recall here the definition of component models given in [NAU 03].

We define in equation 1 the 'model instance' $\bar{\boldsymbol{C}}$ of a component $\boldsymbol{C}$:

$$\bar{\boldsymbol{C}} \models \boldsymbol{C} \quad \text{with} \quad \bar{\boldsymbol{C}} = \{\nu_e, \nu_i, \mathcal{F}, \mathcal{Z}\} \tag{1}$$

where $\bar{\boldsymbol{C}} \models \boldsymbol{C}$ reads "$\bar{\boldsymbol{C}}$ models $\boldsymbol{C}$", and the terms of $\bar{\boldsymbol{C}}$ designate :

$\nu_e$ : *external* variables
$\nu_i$ : *internal* variables
$\mathcal{F}$ : numerical and/or logical constraints
$\mathcal{Z}$ : causal relations

In the sequel, we will shorten 'model instance of a component' to 'component'.

Energy component. : A component $\bar{\boldsymbol{C}} \models \boldsymbol{C}$ is said an *energy component* if :

- all its external variables are generalized efforts, flows or also displacements if these are related to component flows
- all *'main'* internal variables are efforts, flows, displacements as well as momentums where required
- derivatives of efforts and flows which are necessary to the writing of the constraints $\mathcal{F}$ constitute the exhaustive set of *'intermediate'* internal variable
- each internal momentum variable that can mathematically be replaced by a flow will be replaced by the latter
- Each flow variable, be it internal or external, will cause the definition of the corresponding displacement as an internal variable unless the variable in question was already defined as an external variable

An energy component $\bar{\boldsymbol{C}} \models \boldsymbol{C}$ is said *elementary* if it has only one external flow, one external effort and at most one other external or main internal variable. Intermediate internal variables can be included. It follows that capacitors, inertias and resistors of bond-graphs are elementary energy components. Example of components are given in [NAU 05]. Like in [HAU 96], we name *'causal relations'* (elements of $\mathcal{Z}$) such relations where the value of the 'influenced' variable is deduced from current and past values of the 'influent' variable. Causal relations can be either integrative or derivative in a HCG. Other relations are named 'constraints' or 'rigid relations'.

Once a component analysis of the system has been made, the components can be assembled in a HCG. The HCG has the following definition [NAU 03], [NAU 05] :

$$\begin{array}{rll}
\mathcal{G} & = \{S, S', R, \mathcal{L}_\mathcal{C}, \mathcal{L}_\mathcal{D}, \mathcal{L}_\mathcal{R}\} & \\
S & \quad\text{nodes for efforts, flows,} & \\
 & \quad\text{displacements, momentums} & \\
S' & \quad\text{nodes for derivative of efforts} & \\
 & \quad\text{and flows} & \\
R & \quad\text{nodes to label rigid relations} & (2)\\
\mathcal{L}_\mathcal{C} & \quad\text{bipolar integrative causal relations} & \\
\mathcal{L}_\mathcal{D} & \quad\text{bipolar derivative causal relations} & \\
\mathcal{L}_\mathcal{R} & \quad\text{arcs of rigid relations} & \\
\mathcal{C}omp & \leftrightarrow \mathcal{L}_\mathcal{R} \quad \text{Computational information} & \\
 & \quad\text{on arcs of rigid relations} &
\end{array}$$

$$\begin{array}{rll}
S & = E \cup F \cup D \cup M & \\
E & \quad\text{effort variables} & \\
F & \quad\text{flow variables} & (3)\\
D & \quad\text{displacement variables} & \\
F & \quad\text{momentum variables} &
\end{array}$$

$$\begin{array}{rll}
S' & = E' \cup F' & \\
E' & \quad\text{variables for derivatives of efforts} & (4)\\
F' & \quad\text{variables for derivatives of flows} &
\end{array}$$

$$\begin{array}{c}
\text{please note:}\\
\forall x \in D, \dot{x} \in F\\
\forall x \in M, \dot{x} \in E
\end{array}$$

$$\begin{array}{rll}
\mathcal{L}_\mathcal{C} = & \{R_i, i \in [1,n],\ \forall i\ R_i \in (E \times F) \cup (F \times E) & \\
 & \text{or } R_i \in (E' \times E) \cup (F' \times F) & \\
 & \quad \cup (E \times M) \cup (F \times D)\} & \\
\mathcal{L}_\mathcal{R} = & \{R_i = \{a_i, b_i\},\ i \in [1,m], & (5)\\
 & \quad a_i \in R\ et\ b_i \in S \cup S'\} & \\
\exists f_c : & \mathcal{L}_\mathcal{R} \leftrightarrow \mathcal{C}omp \quad \forall i \in [1,m], f_c(R_i) = C_i & \\
 & \quad C_i \in \{':=' ,' =:' ,' ?',\ ' :\neq:'\} &
\end{array}$$

The symbol := assigned to arc $R_i = \{a_i, b_i\}$ with $b_i \in S \cup S'$ means $b_i$ comes out a calculation using the member variables of the constraint $a_i$.
The symbol =: assigned to arc $R_i = \{a_i, b_i\}$ with $b_i \in S \cup S'$ means that $b_i$ is an input to the calculation of variables of the constraint $a_i$.
The symbol ? assigned to arc $R_i = \{a_i, b_i\}$ with $b_i \in S \cup S'$ means that there is no knowledge yet on how to calculate $b_i$.
The symbol :≠: assigned to arc $R_i = \{a_i, b_i\}$ with $b_i \in S \cup S'$ means that $b_i$ cannot be determined with symbolic calculus on the constraint $a_i$.
A constraint $A \in R$ is said *solved* if $\forall i$ such that $R_i \in \mathcal{L}_\mathcal{R}$ and $R_i = \{A, b_i\}$, $f_c(R_I) \neq\ '?'$ and $f_c(R_I) \neq\ ' :\neq:'$.

Modalities. The *Annotated HCG* (AHCG) is augmented by qualitative information expressed as 'modalities'. A modality is defined relatively to a variable and can also be related to arcs of the HCG. We distinguish between *sign* modalities, *structural* modalities and *functional* modalities. A modality is basically an interval and a label.
For a scalar variable $X$, an interval $[a,b]$ defined on $R$ and the interval label '$name$', we will define the symbol $\vdash$ such that :

$$X \in [a,b] \quad\Longleftrightarrow\quad X \vdash \{name\}$$

A *functional* modality characterise a variable and not an equation. With functional modalities can be integrated qualitative knowledge from various sources, from functional specifications to quality control.
*Structural* modalities represent limits of validity of algebraic differential equation sets. Each characterise a variable and one or several relations. An AHCG with structural modalities is a hybrid system. Rigid relations can be disabled temporarily ('*suspended*') or definitively ('*broken*'). An example of the first case is a compression-only spring. An electrical fuse corresponds to the second case. Structural modalities can also affect causal relations for '*saturations*' of dynamics. Examples of such cases include mechanical stops, and reaching pressure of a relief valve. Switches that are direct inputs from outside the system without any power port variable involved are called here *command switches*.
The schematics for such structural modalities are given in table I (taken from [NAU 05]).

| Change in structure | schematics |
|---|---|
| Break of a rigid relation | $I_A$ {fuse passing} $I_B$ {fuse melted} 0 |
| Suspend a rigid relation | {¬action} 0 A {action} B |
| Saturation of dynamics | $\dot{X} \xrightarrow{+\vee 0[\|[]\|} X$ $\{free\}[\|stop_{left}\ stop_{right}\|]$ |
| Distant change (suspend) | →A{¬action} 0 C B |
| Command switch | A 0 B |

TABLE I
STRUCTURAL MODALITIES AND COMMAND SWITCH ANNOTATION

Sign modalities are defined in [NAU 05] and won't be detailed here.
The notation for interval bounds deviates from the standard due to the physical meaning of the interval. When $X \in [|min \quad max|]$, it means that $X \in ]min \quad max[$ or $X = min_+$ ($X$ moving away from $min$ on the right side) or $X = max_-$ ($X$ moving away from $max$ on the left side).
Formal definition of these intervals with logical predicates could be supported by the power-port concept (bond of the bond-graph) and be inspired by the ground construction of modal intervals (for an example on modal intervals, see

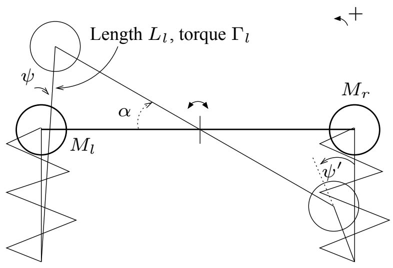


Fig. 1. Pivoting bar with traction elastics (ex. 1)

[SAI 02]). Such definition is beyond the scope of this paper. However, our intent can be specified by the following. 'Cutting up points' allows us to build a partition of $X$ for any structural modality. If several modalities apply to this variable, the final partition for $X$ will be obtained by intersecting all modalities [NAU 03]. We call the intervals of this partition '*qualifying intervals*'. By using naming conventions, it is possible to give a unique label to each of them. It follows that qualifying intervals are also modalities.
We will use the subsequent notation for modalities. For a scalar variable $X$, an interval $[a,b]$ defined on $R$ and its label '$name$', the symbol $\vdash$ is defined as follows:

$$X \in [a,b] \quad\Longleftrightarrow\quad X \vdash \{name\}$$

For each time instant, there is a unique modality $M$ such that $X \vdash M$. If we consider the case of a mechanical device between two stops with positions $min$ and $max$, then the range of $X$ is as follows:

$$[min \quad max] = |[min \ \cup\ [|min \quad max|] \ \cup\ max]|$$

We use the symbol $\prec$ to indicate contiguity and order of intervals. We have :

$$|[min \ \prec\ [|min \quad max|] \ \prec\ max]|$$

## III. Discrete state semantics

We will use in this paper some didactic models. Example 1 is depicted in figure 1. It is composed of an inertial element (bar with 2 masses) mounted on a pivot and 2 elastics (traction-only springs that can break). The ground acts as mechanical stops for masses $M_l$ and $M_r$.
From equations given below, the AHCG can be drawn as in figure 2.

$$J.\ddot{\alpha} = \Gamma_l + \Gamma_r \tag{6}$$

$$L_l = (h_0 - l_0.sin(\alpha)/cos(\psi) \tag{7}$$

with conditions (9)

$$L_r = (h_0 + l_0.sin(\alpha)/cos(\psi') \tag{8}$$

with conditions (10)

$$tan(\psi) = \frac{-l_0(1-cos(\alpha))}{(h_0 - l_0.sin(\alpha))} \quad (9)$$

$$-\pi/2 < \psi \leq 0 \quad \text{and} \quad \alpha < arcsin(h_0/l_0)$$

$$tan(\psi') = \frac{l_0(1-cos(\alpha))}{(h_0 + l_0.sin(\alpha))} \quad (10)$$

$$0 \leq \psi' < \pi/2 \quad \text{and} \quad \alpha > -arcsin(h_0/l_0)$$

$$\begin{aligned} F_l &= k_l(L_l - L_{l0}) \quad \text{if} \quad L_l \geq L_{l0} & (11)\\ F_l &= 0 \quad \text{otherwise} & \\ F_r &= k_r(L_r - L_{r0}) \quad \text{if} \quad L_r \geq L_{r0} & (12)\\ F_r &= 0 \quad \text{otherwise} & \end{aligned}$$

$$\begin{aligned} \Gamma_l &= F_l.l_0.cos(\alpha - \psi) & (13)\\ \Gamma_r &= -F_r.l_0.cos(\alpha - \psi') & (14) \end{aligned}$$

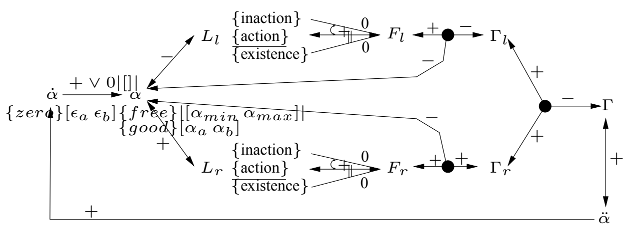


Fig. 2. Pivoting bar : annotated half-causal graph (AHCG)

The partition on $L_l$ and $L_r$ results from two structural modalities $\{existence\}$ and $\{action\}$ for rigid relations 6 and 8 respectively. $\{existence\}$ defines a break of the elastic and of the rigid relation if length exceeds the upper bound of the interval. $\{action\}$ means that the same rigid relation is suspended if length is outside this second interval, actually smaller than its lower bound. The partition that is used in AHCG of figure 2 is shown on bottom of figure 3. $L_{l0}$ is the free length of the spring and $\delta l_e$ is the maximum elastic deformation. We have $\{inaction\} \prec \{action\} \prec \{\neg existence\}$

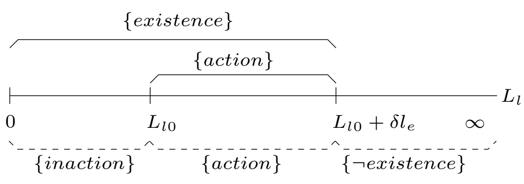


Fig. 3. Modalities and partition into qualifying intervals

When we consider the case of example 1, it appears that the knowledge of current modality for each variable of the system is not sufficient to determine which equations are currently valid. Let us define $L_l$ more precisely as the measured distance between left extremity of the bar and the elastic attachment point on the ground. The modality of $L_l$ can be determined according to equation 6 and figure 3. Let us also make the hypothesis that each elastic deterministically breaks when its length exceeds the plastic limit $l_p = L_{l0} + \delta l_e$. If at some date t, $L_l(t) > l_p$, then the elastic is broken from that time on and equation 11 never applies afterwards. It is anyway possible that $l_l(t+1) \in [L_{l0}\ l_p]$ which has label $action$.

Structural modalities introduce changes in the set of valid equations, which we have called switches. The vector of all switch values defines what we call a *commutation*. The vector $M$ of modalities for all annotated variables in the system can be regarded as the 'discrete state of the variables', and the commutation $C$ can be regarded as the 'discrete state of the equations' graph'. The state $S$ of the system can be characterized as follows:

$$S = (M, C) \quad (15)$$

$C$ is a vector with $r$ elements. $r$ is the number of HCG annotations with structural modalities and command switches. Each switch element of $C$ can be called *elementary* commutation. Part of the information that is included in $C$ can be deduced from $M$, this the case for elementary commutations such as suspensions of rigid relations and saturations. It is then possible to obtain a more compact representation and express $S$ as:

$$S = (M, C_e) \quad (16)$$

Elements of $C_e$ are called *explicit* unitary commutations. They include in particular command switches and breaks of rigid relations (as given in table I). Other elementary commutations are said *implicit*.

The vector modality $M$ for a vector $X$ of annotated variables is defined as follows:

$$X(t) \vdash M \iff \forall i, X_i(t) \vdash M_i$$

$$\text{with} \quad X(t) = \begin{pmatrix} \vdots \\ X_i(t) \\ \vdots \end{pmatrix} \quad \text{and} \quad M = \begin{pmatrix} \vdots \\ M_i \\ \vdots \end{pmatrix}$$

## IV. CONVERTING A SET OF SWITCHED-DAES TO A HYBRID AUTOMATON

We have proposed in [NAU 03] a method to convert switched-DAEs in the shape of an AHCG into a finite state automaton. The finite state automaton is considered an abstraction of the hybrid system. The states are defined according to equation 15. There are several ways to build such an automaton.

The most obvious one would be to simulate the hybrid system, and detect for each simulation step which state $S = (M, C)$ is active. The commutation $C$ would be obtained thanks to switch values. The modality $M$ would be determined by finding for each annotated variable which qualifying interval includes the current real value.
The main drawback is that the set of states is in this case incomplete : it depends on the initial conditions (initial values for continuous variables of the AHCG and initial value of switches). Building a useful automaton then requires to merge the discrete state trajectories obtained for many various initial conditions. There is no guarantee that the resulting automaton would be complete, because the number of initial cases is infinite. Furthermore, the automaton would not be deterministic (no label on transitions). In the particular case of linear continuous-variables systems, Lunze clearly stated in [LUN 99] that an abstract discrete event trajectory obtained by quantisation of the continuous model is not deterministic. For a known initial interval of a state variable $x$, which initial real value is unknown, there can be several quantised trajectories.

One alternative is to *envision* the possible discrete state dynamics of the hybrid system by constraint propagation techniques. In the artificial intelligence field of qualitative simulation, Kuipers [KUI 94] applied such methods to his QSIM models. The result is called *attainable envisionment*, when it is computed from an initial qualitative state information. When all possible states are computed independently of initial condition, the result is called *total envisionment*. We have followed Kuipers' guidelines for constraint propagation [KUI 94] and adapted the Qfilter algorithm to our own state formulation. We will as in [NAU 03] restrict our presentation to hybrid systems without command switches. The explicit elementary commutations considered are breaks in rigid relations, which are *irreversible*. We will furthermore restrict our case to rigid relations which are monotonous in respect to each variable. A rigid relation such as $f(\dots, x_i, \dots) = 0$ is said *monotonous with respect to* $x_i$ if for all considered values of other variables $x_j$, an increase in $x_i$ always provokes an increase of $f$, or a decrease in $x_i$ always provokes a decrease of $f$. In this case, signs of influence do not depend on the variable value, and a sign annotation can be attached to corresponding arc on an AHCG.
We start from an initial state $S = (M, C_e)$ and check which possible next states can be attained with *minimal change* in the system. Our criterion for this minimal change is that only the modality of one variable should change by one increment each time. Then the label of a state to state transition would be either $x_i \nearrow$ or $x_i \searrow$. $x_i \nearrow$ reads 'the current modality for $x_i$ goes to next higher modality'. The abstraction method has nevertheless to assure that all states that are physically attainable from the initial state are included. Because the number of different initial discrete state conditions is finite, the completeness of attainable envisionment ensures that a total envisionment can be computed from the merging of the automata. When many continuous variables are annotated with modalities, there are cases when the rule of 'one modality change only' would abusively lead to reject states. We call theses cases '*events on linked variables*'. The minimal change criterion has then to be expressed as 'transition with minimum number of variables changing modality'.

Algorithmic details of our constraint propagation method is beyond the scope of this paper, but we will give hereunder its outline.

For a state $S = (M, C_e)$ with $M = \begin{pmatrix} \vdots \\ M_i \\ \vdots \end{pmatrix}$, state

$S' = (M', C'_e)$ is said a '*topological neighbour*' (or more simply neighbour) of $S$ if:

$$\exists j, \quad M_j \prec M'_j \quad \vee \quad M'_j \prec M_j$$
$$\forall i,\ i \neq j, \quad M_j = M'_j$$

From current state, topological neighbours are generated. Not all neighbours are considered but only those that are compatible with information on signs of variables. This first 'filtering' is made using the causal relations. If $x_j = \int x_i$ and $M_i$ is such that $x_i > 0$ then the modality $M_j$ of $x_j$ cannot decrease. The topological neighbours that go through this first filtering are candidates for successor states of current state. These can be either states that have already been created in the automaton, or 'new' states that have to be 'filtered' by constraint propagation methods. The modalities of all variables have to be checked against the rigid relations. Interval analysis techniques are used (a brief review of such techniques is given in [TRA 97]). The monotony properties of rigid relations also allows to check consistency of modalities even without solving computational causality for the variable for which modality is changing.
The checking of states and transitions is analogous to the browsing of a tree 'depth by depth'. From initial state, all successors, or childs, are checked and possibly created. Then, all grand childs are created. An automaton is created instead of a tree because states that have already been created are not created again when encoutered : a transition is simply added.
The outline of the algorithm is as follows [NAU 03] :

- For all node states $S_j$ of current depth
  - Create a temporary list of neighbours $S_j^k$ of $S_j$ and apply to these the following 'transition-filtering' procedure
    - apply implicit commutation resulting from transition $S_j^k \rightarrow S_j$
    - check the transition $S_j^k \rightarrow S_j$ against evolutions per-

mitted by causal relations

  * if transition not rejected look for an ‘existing’ state similar to $S_j^k$
  * if there was one, create the appropriate transition
  * if there was none : follow sub-procedure ($r$)
    · check modalities of $S_j^k$ against rigid relations
    · if state not rejected, apply explicit irreversible commutations
    · if state not rejected, create $S_j^k$ as child of current node $S_j$

  If candidate state $S_j^k$ was rejected during sub-procedure ($r$), because of a rigid relation $\mathcal{C}\int\sqcup_{||}$, erase

  $S_j^k$ from temporary list, and replace it by a sublist

  of neighbours of $S_j^k$ according to other variables included in $\mathcal{C}\int\sqcup_{||}$. Apply ‘transition-filtering’ procedure to this sublist

– Go to next brother state of $S_j$ and repeat previous step

- If no child was created, stop

Because of the ‘events with linked variables’ problematic, the algorithm contains a recursive call. The algorithm is built so that all physically possible states are included in the automaton (completeness). The counterpart is that some unrealistic states might go through the filtering process and never be encountered in practice. Hybrid simulations with stochastic choices of initial conditions can be a practical solution to check the automaton against this problem. To give maximum performance to the filtering and limit unwanted states to a minimum, a practical algorithm has to include a *refinement* procedure. This refinement consists in narrowing interval ranges of tested variables instead of using only original ranges of modalities, and can be achieved by constraint propagation methods as for the checking of rigid relations. Selective refinement should be performed before testing of evolutions against causal relations. A practical algorithm also has to check variables in an appropriate order so that inconsistent states are rejected with shortest computing time. The litterature on optimisation techniques for constraint propagation is abundant [TRA 97].

## V. Transitions and time abstraction

We have proposed a design method for a supervision-oriented hybrid automaton. Its focus is on exhaustivness of possible qualitative behaviours of the system considered. One drawback is its lack of compactness for systems with many variables and annotations. This problem can be partially adressed by a hierarchical analysis and inclusion of automata within another like in statecharts [SCH 98]. We have found more crucial to focus on another property of possible trajectories. When comparing different trajectories towards or from a common point, the notion of ‘time

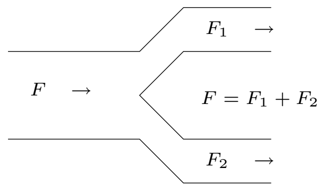


Fig. 4. Pipe ‘Y’ junction (ex. 2)

distance’ should be kept consistent between original hybrid system and abstract discrete event system.
Our abstract system is an automaton which does not bear any explicit time information. One common interpretation of automata that model physical systems is that some time may be spent in states and that transitions are instantaneous. This is not inconsistent with the hypothesis that the number of transitions from a given state towards another is somewhat related to an amount of time. It is desirable that the minimum number of transitions between two states in the abstract automaton of a mechatronic hybrid system can be used as a ‘qualitative distance’, which allows to compare trajectories. This comparison is a ‘first guess’ as such ‘qualitative time distance’ does not constitute an exact order of events in the real time-line. We have met this requirement by the notion of minimal change in the system during transitions. Transitions are preferably changes on the modality of one variable only or ‘events on linked variables’ as defined in section IV.
Example 2 depicted in figure 4 and table II is intended to illustrate the need for the ‘event on linked variables’ mechanism.It is composed of a ‘Y’ junction out of a main pipe. The underlined modality names indicate the initial state $S_0$ considered.

| Variable | Modality | Type | Bounds |
|---|---|---|---|
| $F_1$ | low | functionnal | [0 5[ |
| $F_1$ | <u>nominal</u> | functionnal | [5 6[ |
| $F_1$ | high | functionnal | [6 9[ |
| $F_1$ | danger | functionnal | [9 10] |
| $F_2$ | low | functionnal | [0 5[ |
| $F_2$ | <u>nominal</u> | functionnal | [5 6[ |
| $F_2$ | high | functionnal | [6 10] |
| $F$ | low | functionnal | [0 10[ |
| $F$ | <u>nominal</u> | functionnal | [10 12[ |
| $F$ | high | functionnal | [12 20] |

TABLE II
Modalities for pipe junction example

Let us test the transition $S_0$ to candidate state $S_0^1$, where $S_0^1$ is defined by $F \vdash \{high\}$, $F_1 \vdash \{nominal\}$ and $F_2 \vdash \{nominal\}$. $S_0^1$ is to be rejected because the intervals do not satisfy the constraint $F = F_1 + F_2$. In the contrary, $S_0^2$, where $F \vdash \{nominal\}$, $F_1 \vdash \{high\}$ and $F_2 \vdash \{nominal\}$, is valid.
Let us set $S_1 = S_0^2$.
$S_1^1$, where $F \vdash \{high\}$, $F_1 \vdash \{high\}$ and $F_2 \vdash \{nominal\}$, is valid. We obtain a valid trajectory:

$$S_0 \rightarrow S_1 \rightarrow S_1^1 \quad (17)$$

In this example, filtering out $S_0^1$ does not make it impossible to reach a state in which $F \vdash \{high\}$, as shown by equation 17. Completeness is not called into question. Should we consider that the 'distance' from a $S_0$ to $S_1^1$ is 2 while 'distance' from $S_0$ to $S_1$ is 1? In the first case, $F$ is moving while in the second case, $F_1$ is moving. There is no physical reason to consider a change for $F$ more 'distant' than a change for $F_1$.

Because we have defined 'events with linked variables', the algorithm that builds the automaton performs the following procedure: because $S_0^1$ is checked out, other variables of the constraint are checked against a change. Both $F_1$ and $F_2$ are tested. $S_{1,0}^1$, corresponds to the test on $F_1$ and has same modalities that previously discussed $S_1^1$. $S_{1,0}^1$ should be created as $S_3$. Same reasoning is applied to $F_2$. Finally, we obtain 4 sons with a modality increase from the initial state $S_0$, which are given in table III. Sons with a modality decrease should of course also be added.

| Modality | $S_0$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
|---|---|---|---|---|---|
| $F_1 \vdash \{low\}$ | | | | | |
| $F_1 \vdash \{nominal\}$ | 1 | | 1 | | 1 |
| $F_1 \vdash \{high\}$ | | 1 | | 1 | |
| $F_1 \vdash \{danger\}$ | | | | | |
| $F_2 \vdash \{low\}$ | | | | | |
| $F_2 \vdash \{nominal\}$ | 1 | 1 | | 1 | |
| $F_2 \vdash \{high\}$ | | | 1 | | 1 |
| $F \vdash \{low\}$ | | | | | |
| $F \vdash \{nominal\}$ | 1 | 1 | 1 | | |
| $F \vdash \{high\}$ | | | | 1 | 1 |

TABLE III
SUCCESSORS OF $S_0$ (MODALITY INCREASE) FOR PIPE JUNCTION EXAMPLE

## VI. CONCLUSION

A file format for switched-DAEs annotated with qualitative information has been defined and implemented in a software that we called DDS (Discrete Description for Supervision). Such mathematical description can be prepared with the AHCG methodology [NAU 05]. In this paper, the type of switches is discussed on the ground of the AHCG. It is argued that a supervision-oriented hybrid model of mechatronic systems should be built in such a way that the discrete part of the model can be used as an abstraction. An expression of states is proposed, and some principles to compute an automaton as such discrete abstract model are given. The practical expression of states does take into account modalities and only the switches that are either irreversible or command switches. The soundness of transition labels is discussed against a criterion of 'qualitative distance'. This criterion is important when it is considered to use the discrete automaton in a stand-alone mode for certain parts of supervision control. Because our automaton does not bear explicit time information, this led to the definition of 'event with linked variables'.

The practicality of principles and of DDS software were tested on a realistic mechanical system, which was a model of a bogie of a forest machine [NAU 03]. A model of the complete forest machine was created on a commercial simulation software. A simulation of an accident with the machine overturning while driving on a slope was created. The abstract model of the bogie was computed with DDS. Qualitative information (functional modalities) were defined according to load specifications for the machine. Structural modalities resulted from contact between tires and the ground. Results of the realistic accident simulation were mapped against the discrete states of the abstract model. It followed that the number of transitions from a given state to the accident state was a consistent indication of dangerosity level. Our criterion of minimal qualitative change for transitions, such as explained in section IV and discussed in section V, was shown a sound criterion in this example. It is possible to use the abstract automaton as a help to choose strategy and sensors for an anti-overtuning monitoring device.

## REFERENCES


[ALU 95] ALUR R., COURCOUBETIS C., HALBWACHS N., HENZINGER T. A., HO P. H., NICOLLIN X., OLIVERO A., SIFAKIS J., YOVINE S., *The Algorithmic Analysis of Hybrid Systems*, *Theoretical Computer Science*, vol. 138, n1, p. 3-34, 1995.

[DAN 95] DANES P., AGUILAR-MARTIN J., *The Symbolic-Numeric Interface - a Zosteric Approach*, *Applied Artificial Intelligence*, vol. 9, n5, p. 451-478, 1995.

[GUÉ 01] GUÉGUEN H., LEFEBVRE M.-A., *A comparison of mixed specification formalisms*, *Journal Européen des Systèmes Automatisés*, vol. 35, n4, p. 381-393, 2001.

[HAU 96] HAUTIER J., J. F., *Le graphe informationnel causal*, *Bulletin de l'Union des Physiciens*, vol. 90, njuin 96, p. 167-189, 1996.

[KAR 90] KARNOPP D. C., MARGOLIS D. L., ROSENBERG R. C., *System dynamics : a unified approach*, Wiley-Interscience, Wiley, New York, 2nd dition, 1990.

[KUI 94] KUIPERS B. J., *Qualitative Reasoning : Modeling and simulation with incomplete knowledge*, vol. 59, MIT Press, Cambridge, MA, USA, 1994.

[LUN 99] LUNZE J., NIXDORF B., SCHRODER J., *Deterministic discrete-event representations of linear continuous-variable systems*, *Automatica*, vol. 35, n3, p. 395-406, 1999.

[LUN 01] LUNZE J., NIXDORF B., RICHTER H., *Process supervision by means of a hybrid model*, *Journal of Process Control*, vol. 11, n1, p. 89-104, 2001.

[NAU 03] NAUD O., Modélisation hybride pour la supervision de systèmes mécatroniques : application à la stabilité en pente de machines mobiles *Hybrid modeling for supervision of mechatronic systems : application to stability of mobile machinery on slopes - in french*, PhD thesis, INSA, Toulouse, 2003.

[NAU 05] NAUD O., *The half-causal graph : power-port based schematics with annotations for computational analysis of mechatronic and hybrid systems*, *Submitted to Journal of Franklin Institute. See also [NAU 03]*, 2005.

[SAI 02] SAINZ M. Á., ARMENGOL J., VEHÍ J., *Fault detection and isolation of the three-tank system using the modal interval analysis*, *Journal of Process Control*, vol. 12, n2, p. 325-338, February 2002.

[SCH 98] SCHOLZ P., Design of Reactive Systems and their Distributed Implementation with Statecharts, PhD Thesis, Technischen Universitt (TUM), München, 1998.

[SÖD 95] SÖDERMAN U., STRÖMBERG J.-E., Switched Bond Graphs: multiport switches, mathematical characterization and systematic composition of computational models, Rapport nLiTH-IDA-R-95-07, Department of Computer and Information Science, Linköping University, Sweden, 1995.

[STR 94] STRÖMBERG J.-E., A Mode Switching Modelling Philosophy, PhD thesis, Linköping University, Linköping, 1994.

[TOP 91] TOP J. L., AKKERMANS J. M., BREEDVELD P. C., *Qualitative reasoning about physical systems: an artificial intelligence perspective*, *Journal of the Franklin Institute*, vol. 328, n5-6, p. 1047-1065, 1991.

[TRA 97] TRAVÉ-MASSUYÈS L., GUERRIN F., DAGUE P., *Le raisonnement qualitatif : pour les sciences de l'ingénieur*, Diagnostic et maintenance, Hermès, Paris, 1997.