DISCRETE SUPERVISOR SYNTHESIS FOR A CLASS OF CONTINUOUS SYSTEMS

Jean-Marie Flaus

Laboratoire d’Automatique de Grenoble
ENSIIEG, BP 46, F-38402 St Martin d’Hères, FRANCE
Email : flaus@lag.grenet.fr

Abstract: In this paper, we deal with the problem of discrete supervisor synthesis for continuous systems. We consider a special class of systems, namely those that can be modelled by a hybrid flow net. The main idea for the synthesis is based on the notion of invariants in hybrid flow nets. The proposed approach is illustrated on a batch process example.

1. INTRODUCTION

This work deals with the synthesis of a discrete supervisor for a class of continuous processes, namely the transformation process. The notion of supervision has different meanings. In the continuous world, supervision stands for a set of techniques that allows monitoring of a controlled process in order to detect and prevent unexpected behavior of the process. In the discrete event world, a formal definition of supervision has been proposed [RAM87] and the aim of a supervisor is to prevent some events in order to keep the system in a given subset of the discrete state space. In this work, we propose a method for the synthesis of a discrete supervisor, that is to say a supervisor acting on discrete input of the continuous process in order to force the process to obey some inequality constraints. When such a supervisor is in closed loop with a continuous process, we obtain what is known as a hybrid dynamical system [STI92] [GRO93] [QUE94] [ANT95] [BRA95] [IMP95].

In the sequel, we present the class of systems we are interested in, namely the transformation systems, which can be modelled by hybrid flow nets whose definition we will give. Then we will present the synthesis method which is based on the notion of feedback invariants and is an extension of the idea presented in [YAM96] for DES. Finally, we illustrate this approach on a batch process example.

2. TRANSFORMATION PROCESSES

2.1. Presentation

General hybrid dynamical system models are very complex. First, the continuous part is often non linear and hence is already difficult to analyze alone. If you add the complexity provided by the DES part and by the interaction, you obtain something for which no general synthesis method can reasonably be considered.

So, we have chosen a less ambitious approach and consider a special class of continuous systems, called transformation systems [WAL82]. These systems have some interesting properties while at the same time being general enough to represent a large class of real systems, such as chemical kinetics, biotechnological processes or food processes.

The general form of the model of such systems is as follows:

\[ \frac{dX(t)}{dt} = D \Phi(X(t)), \quad X(0) = X_0 \]

where \( X(t) \in \mathbb{R}^n \), \( \Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m \) and \( D \) is \( n \times m \) matrix. We denote \( \phi_i(X(t)) \), \( x_i(t) \) the i-th component of \( \Phi(X(t)) \), \( X(t) \) respectively.

The following properties must be fullfiled:

1. \( \forall i \in \{1, \ldots, n\} \quad x_i(0) \geq 0 \)
2. \( \forall i \in \{1, \ldots, m\} \quad \phi_i(X(t)) \geq 0 \)
3. \( \forall i, j \quad d_{ij} < 0 \quad and \quad x_i(t) = 0 \Rightarrow \phi_j(X(t)) = 0 \)

Such a model can be represented by a graph structure which highlights the flow transformation. This graph structure is called a continuous flow net [FLA96] and is very similar to the Petri net one, where vertices are places and transitions. In a continuous flow net, places will be used to represent extensive variables such as volume, quantity of components or energy whose value is the marking of the place. Each place is connected to transitions (Figure 1). The flow through this transition is a continuous function of the marking that depends on the one hand on the value of the place, and on the other hand, on a function of the marking of the net which must be positive and bounded.
More formally, a continuous flow net is defined as follows.

**Definition**

A continuous flow net (CFN) is defined by a n-plet
\[ C = \{P, T, I, O, \Phi, X_i\} \]
where:
- \( P \) is a set of \( n \) places represented graphically by a rounded square. Each place is assigned a real number, positive, which is called marking or value of the place, and denoted \( x_i \). The set of values makes a vector \( X_0 \) where \( X_0 \) is the marking of the i-th place. \( X_0 \) is the initial marking.
- \( T \) is a set of \( m \) transitions (or gates), represented graphically by a specific rectangle (Figure 1).
- \( I : \{[0,\ldots,n]\} \times [0,\ldots,m] \rightarrow \mathbb{R} \) is the input function that specifies the arcs directed from places to transitions.
- \( O : \{[0,\ldots,n]\} \times [0,\ldots,m] \rightarrow \mathbb{R} \) is the output function that specifies the arcs directed from transitions to places.
- \( \Phi : \{[0,\ldots,m]\} \rightarrow F(\mathbb{R}^m, \mathbb{R}) \) assigns each transition a continuous function \( \varphi_j(X_i) \) defined from \( \mathbb{R}^m \) to \( \mathbb{R} \), that is such that
  1) \( \forall i, \varphi_j(X_i) \geq 0 \)
  2) \( \forall i, I_j(i,j) \neq 0 \) and \( X_{i,j} = 0 \implies \varphi_j(X_i) = 0 \)
which means that the flow must always be positive and that if the marking of at least one input place of a transition is zero, then the flow through the transition is zero.

2.2. Hybrid Flow nets

In order to model continuous and discrete aspects of a system, we introduce what we call the Hybrid Flow Net (HFN). This modelling tool is made of a continuous flow net interacting with a Petri net (PN) according to a control interaction, that is to say the PN controls the CFN and vice versa. It can be considered as an extension of Hybrid Petri Nets which are limited to linear systems [DAV92], but the interaction rules are very similar. As we will see, the overall philosophy of PNs is once again preserved: the validation of transition implies that all the input places are not empty and the evolution rule is similar. We will distinguish two kinds of interaction: a control interaction and a flow interaction.

In the first case, the marking of the place of one net is used to control the flow of another net, but the value of this control place is not modified by the firing of the transition. Graphically, the control place is connected to the transition on the small side.

In the second case, the marking of the place is used for the control and is modified by the firing of the transition: we have a conversion of the continuous flow into a discrete flow and vice versa.

The control of the continuous flow via a discrete place (figure 2) is performed via the modification of the flow as follows:

\[ \varphi'_j(X) = X_{di} \varphi_j(X_c) \]

where \( X_{ci} \) is the marking of the discrete place used for the control and \( \varphi_j(X_c) \) is the continuous flow rate defined above.

The control of a discrete transition with a continuous place is performed via a threshold: when the marking of the place is greater or equal, then the discrete transition may be enabled if it is enabled by the rest of the net. An inhibition influence can also be defined (figure 3): in this case the marking of the continuous place must be less than the threshold.
\[
\frac{dX_c(t)}{dt} = D_c \Phi(X_c(t))
\]
where \( \Phi(X) \) is the continuous flow vector, and the matrix \( D_c \) is the incidence matrix.

If we denote \( I_{cd} \) the input function of continuous flow net to the Petri net, \( O_{cd} \) the input function of continuous flow net to the Petri net and \( I_{dc} \) the same functions from the PN to the CFN, we can define the cross incidence matrices:

\[
D_q = [O_{dc}(i,j) - I_{cd}(i,j)]
\]
\[
D_q = [I_{cd}(i,j) - I_{dc}(i,j)]
\]

These matrices describe the conversion of continuous flow into discrete flow and vice versa. Let us denote the state vector of the HFN:

\[
X = \begin{bmatrix} X_c \\ X_d \end{bmatrix}
\]

We can write the incidence matrix of the HFN as follows:

\[
D = \begin{bmatrix} D_c & D_q \\ D_q & D_d \end{bmatrix}
\]

A place invariant of the HFN is defined as a vector \( x \) of length \( n_s+n_t \) whose component 1..\( n_c \) are real values and component \( n_s+1..n_d \) are integers and such that:

\[
x^T D = 0
\]

### 3. SUPERVISOR SYNTHESIS

#### 3.1. Notion of supervisor

Let us first recall the definition of a supervisor for a discrete event system. Let us consider a discrete event system, called the plant and let \( \Sigma \) denote a finite set of events that occur in this DES, where \( E \) is the set of states and \( \delta \) the transition function partially defined as \( \delta: \Sigma \times E \rightarrow E \). The plant can be modelled by an automaton \( G = (\Sigma, E, \delta, X_0) \) where \( X_0 \) is the initial value of the state.

The set of events is partitioned as the set of controllable events \( \Sigma_c \) and the set of uncontrollable events \( \Sigma_u \). A supervisor restricts the behavior of the plant by dynamically disabling some events of the controllable events upon execution of each event. If we denote \( K \) the language generated by the plant, defined as a set of sequences of events, a supervisor can be defined as a map: \( f: K \rightarrow 2^{\Sigma} \) and for each \( s \in K \), \( f(s) \) is the set of events that are disabled after the execution of \( s \). Such a supervisor will be designed so that the language under supervision lies in a given subset of \( K \).

For a DES modelled by a Petri net, the problem of designing a maximally permissive supervisor has been studied by [HOL90],[LI93],[GIU94] and [YAM96]. A survey is presented in [HKG97].

#### 3.2. Feedback Control of PN based on P-Invariants

The approach that we have used for the synthesis of our discrete supervisor is an extension of the supervisor synthesis for Petri nets based on place invariants proposed in [YAM96]. Let us recall the main ideas of this method. First, the control goal is to force the process to obey constraints of the form

\[
\sum_{i=1}^{n} l_i X_{di} \leq \beta
\]

where \( X_i \) is the marking of the discrete place \( p_i \) and \( l \) and \( \beta \) are integer constants. This inequality constraint can be transformed into an equality by introducing a non negative slack variable \( s \) into it to obtain

\[
\sum_{i=1}^{n} l_i X_{di} + s = \beta
\]

The slack variable represents a new place which holds the extra tokens required to meet the equality. This new place is part of a separate net called the controller net.

In order to design a controller for a set of constraints, they are written in matrix form for the extended Petri net including one control place for each constraint

\[
[L \ I] \begin{bmatrix} X_d \\ S \end{bmatrix} = B
\]

where \( S \) and \( B \) are respectively the vector of the slack variables (controller places) and the vector of \( \beta \) while \( L \) is an \( n_s \times n_t \) integer matrix and \( I \) is an \( n_s \times n_t \) identity matrix.

If we denote \( D_c \) the incidence matrix of the process model and \( D_d \) the incidence matrix of the controller, each place invariant defined by the relation above must satisfy

\[
[L \ I] \begin{bmatrix} D_p \\ D_c \end{bmatrix} = 0
\]

The matrix \( D_c \) defines the arcs that connect the controller places to the transitions of the process net. It can be computed as follows

\[
D_c = -LD_p
\]

and the initial marking of the controller is defined by

\[
S_0 = B - LX_{d0}
\]

This control method can be shown to be maximally permissive [YAM96].

#### 3.3. Extension to Hybrid Flow nets

As the hybrid flow nets have the same structural property about place invariant as Petri nets, it is possible to use the same approach for designing a
hybrid supervisor. In a Petri net, a supervisor defines the set of transitions that cannot be fired (which is equivalent to the set of disabled events). For a hybrid flow net, the supervisor modifies the flow in order to satisfy the invariant. Some continuous transitions are enabled or not in an adequate manner. These continuous transitions are controlled through a discrete place or through a continuous one, depending on the nature of the invariant: for an invariant involving at least a continuous place, we will use a continuous place in the supervisor.

We define the behavior of the continuous places so that the output of the supervisor is discrete: if the marking of a continuous control place is zero then the flow of the transitions to which the place is connected is turned off while if the marking is greater than zero, then the flow is left unchanged:

$$\varphi_{j\text{/supervision}}(X) = \delta(x_c) \varphi_j(X)$$

where $x_c$ is the marking of the control place and $$\delta(x) = 1 \text{ if } x > 0 \text{ and } \delta(x) = 0 \text{ otherwise}.$$ In both cases, the outputs of the supervisor are discrete.

The structure of the supervisor is then obtained using the same approach as the Petri net one which has been developed by [YAM96]. So, if we denote $D_p$, the incidence matrix of the hybrid model of the process, we can compute the incidence matrix $D_c$ of the controller as follows

$$D_c = -LD_p$$

The control places are connected to the continuous and discrete transitions of the process as indicated by the incidence matrix $D_c$. We illustrate this on a batch process example in the sequel.

We assumed that all the transitions are controllable and observable. Some methods have been proposed in [MOO95][MA96] in order to remove this limitation, but we have not investigated this point and we do not know if this can be useful for hybrid flow nets.

4. APPLICATION TO A BATCH PROCESS EXAMPLE

4.1. Presentation of the example

In order to illustrate the proposed method for the synthesis of a hybrid supervisor, we will apply it to an batch process example. For this process, the aim is to design a supervisor that prevents some undesired actions from being performed, such as using the same pipe for two transfers of different products at the same time or overfilling a tank. However, the supervisor does not implement the recipe which is executed by the controller within the limits that the supervisor allows.

This batch process is made up of six tanks and one stock (figure 4). Two products, A and B from tank 1 and 2 respectively are mixed in tanks 3, 4 or 5 in which a reaction occurs. Then the obtained product is sent to tank 6 which is used to fill the bottles, stocked in stock 6. The production is performed in parallel in tanks 3, 4 and 5.

The aim of the supervisor is to avoid:
- (r0) overfilling of stock 7,
- (r1) overfilling of tank 6,
- (r2) two transfers from tanks 1 or 2 in tanks 3,4 or 5 at the same time,
- (r3) two transfers from tanks 3,4 or 5 in tank 6 at the same time,
- (r4) to start a transfer from tanks 1 or 2 to tanks 3,4 or 5 when there is not enough product A or B in tanks 1 or 2 respectively.
The hybrid flow net of this system is given on figure 5. We can observe that the transition t10 transforms a continuous flow into a discrete one (10 bottles per unit of volume). The incidence matrix can be split into 3 parts, denoted D1, D2 and D3. The first one describes the structure of the physical process, the second is for the input valves and the last one describes the behaviour of the output valves. They are given in the following tables 1, 2 and 3.

<table>
<thead>
<tr>
<th></th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>t5</th>
<th>t6</th>
<th>t7</th>
<th>t8</th>
<th>t9</th>
<th>t10</th>
<th>t11</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>p2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>p3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>p4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>p5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>p6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>p7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t12</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t13</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t14</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t15</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t16</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t17</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t18</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t19</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t20</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t21</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t22</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t23</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. Part 1 of the incidence matrix (D1)

<table>
<thead>
<tr>
<th></th>
<th>t24</th>
<th>t25</th>
<th>t26</th>
<th>t27</th>
<th>t28</th>
<th>t29</th>
</tr>
</thead>
<tbody>
<tr>
<td>p20</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p21</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p22</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p23</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p24</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p25</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Part 2 of the incidence matrix (D2)

The full incidence matrix is then given as follows:

\[
D = \begin{bmatrix}
  D1 & 0 & 0 \\
  0 & D2 & 0 \\
  0 & 0 & D3
\end{bmatrix}
\]

4.2. Design of the Supervisor

First, we must express the requirements (r0)-(r4) in the adequate form of the constraints presented above:

\[
\sum_{i=1}^{n} l_i X_{di} \leq \beta
\]

For each requirement, we obtain one or several conditions:

(c0) \[ x_6 + 0.1 d_7 \leq 0.1 d_{7\max} \]

(c1a) \[ x_3 + x_6 \leq x_{6\max} \]

(c1b) \[ x_4 + x_6 \leq x_{6\max} \]

(c1c) \[ x_5 + x_6 \leq x_{6\max} \]

(c2) \[ d_9 + d_{11} + d_{13} + d_{15} + d_{17} + d_{19} \leq 1 \]

(c3) \[ d_{21} + d_{23} + d_{25} \leq 1 \]

(c4a) \[ \frac{x_1}{20} + d_8 \geq 1 \]

(c4b) \[ \frac{x_1}{20} + d_{12} \geq 1 \]

(c4c) \[ \frac{x_1}{20} + d_{16} \geq 1 \]

(c4d) \[ \frac{x_1}{20} + d_{10} \geq 1 \]

(c4e) \[ \frac{x_1}{20} + d_{14} \geq 1 \]

(c4f) \[ \frac{x_1}{20} + d_{18} \geq 1 \]

In these relations, \( d_{7\max} \) denotes the maximum size of the stock, \( x_{6\max} \) denotes the maximum size of the tank 6 and the minimum value for enabling a transfer from tanks 1 or 2 to tank 3, 4 or 5 is taken equal at 20.

They all have the right form of an inequality « less than » constraint except the conditions (c4x) which are « greater than » inequalities. In this case, the same approach can be applied for the design of the supervisor with a slight modification: the slack variable is an excess variable [YAM96] and we have

\[
\sum_{i=1}^{n} l_i X_{i} - s = \beta
\]

which leads to the following relation for computing the incidence matrix of the controller:

\[
[L - I] \begin{bmatrix}
  D_p \\
  D_c
\end{bmatrix} = 0
\]

and

\[ D_c = LD_p \]

The first condition (c0) is a hybrid condition because it involves a continuous and a discrete place in the same invariant. The associated vector \( L_0 \) is equal to:

\[ L_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ \ldots \ 0] \]

One controller place is needed to satisfy the condition (c0) and the arcs connected to this place
are given by $D_{c0} = -L_0 \cdot D$ which is equal to a vector of 29 components equal to 0 except

$$D_{c0}(7) = D_{c0}(8) = D_{c0}(9) = -1 \text{ and } D_{c0}(11) = 0.1$$

This means that the control place is connected to the transition $t_7$, $t_8$ and $t_9$ with an arc of weight 1 and from the transition $t_{11}$ with an arc of weight 0.1. This place, denoted $x_{c0}$, is represented on figure 6. Of course, as the invariant includes continuous places, then the control place is continuous.

Let us now consider the conditions (c1x). These conditions are purely continuous. The associated vectors $L_{c_{x_1}}$ can be put in a matrix form equal to:

$$L_4 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & \cdots & 0 \end{bmatrix}$$

We can compute the incidence matrix of the 3 control places to obtain:

$$D_{c_{1}} = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & -1 & -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

Let us consider the control place for the condition (c1a). From the matrix $D_{c_{1}}$, we can see that this place is connected to the transitions $t_{1}$, $t_{2}$, $t_{8}$ and $t_{9}$ from transition $t_{10}$. This control place is represented on figure 6. The two other places have similar connections.

The conditions (c2) and (c3) are purely discrete. So the control places are discrete. To obtain the connections to these places, we build the weight vectors and we compute the following incidence vectors:

$$D_{c_{2}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 1 & -1 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_{c_{3}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

The control place for $c_{3}$ is represented on figure 6. As we can see, this control place keeps one token when the three valves are closed and gives that token to the opened valve, which prevents any other valve from opening.

For the last conditions (c4x), we only consider in detail (c4a) which has the following associated vector:

$$L_{4_{a}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & \cdots \end{bmatrix}$$

The relation for computing the matrix of the controller is a little bit different as explained above $D_{c_{4a}} = L_{4_{a}} \cdot D$ and we obtain:

$$D_{c_{4a}} = \begin{bmatrix} -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 20 & -20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

So the control place (continuous as the condition is hybrid) is connected to the transitions $t_{1}$, $t_{3}$, $t_{5}$ and $t_{13}$ with a weight 10 and from $t_{12}$ with the same weight. Intuitively, this means that in order to open the valve v1, the marking of the control place must be greater than 20.

By using all the conditions, we get a controller with 12 places, two of which are discrete and the others continuous. The role of the arcs connecting the control places to the process model can be explained intuitively, however this method of design is an efficient way of finding all of them easily. As we can see, the supervisor is hybrid since it acts on the process with continuous and discrete inputs.

![Feedback Controller for the batch process](image)

5. CONCLUSION

In this paper, we have presented an approach for designing a discrete supervisor for the class of transformation systems which can be modelled by a hybrid flow net. This method is an extension of the idea proposed by [YAM96] for the design of DES supervisors based on feedback invariants. We have shown on a batch process example that this method is an initial interesting stage for designing a supervisor. The same method could be applied to hybrid Petri nets which have the same structure. The next stage will be to consider the case when some transitions are not controllable or not observable.

7. REFERENCES


[MA96] J.O.Moody and P.J. Ansaklis, Supervisory Control of Petri Nets with uncontrollable/unobservable transitions, 35th CDC, Kobe, Japan, 96


