Using Interval Petri Nets and Timed Automata for Diagnosis of Discrete Event Systems (DES)

L. Mhamdi*a, H. Dhouibi*a, Z. Simeu-Abazi b, N. Liouanea

a UR ATSI D. Génie Electrique de l’ENIM, Monastir, Tunisia
b Grenoble-INP / UJF-Grenoble 1 / CNRS, G-SCOP, Grenoble, 46, avenue Félix Viallet - 38031- France

PAPER INFO

Paper history:
Received 10 June 2013
Received in revised form 27 August 2013
Accepted 14 September 2013

Keywords:
Modeling
Petri Nets
Interval Constraint Petri Nets
Timed Automata
Robustness

ABSTRACT

A discrete event system (DES) is a dynamic system that evolves in accordance with the abrupt occurrence, at possibly unknown irregular intervals, of physical events. Because of the special nature of these systems, different tools are currently used for their analysis, design and modeling. The main focus of this paper is the presentation of a new modeling approach of Discrete Event Systems. The proposed approach is based on hybrid model which combines Interval Constrained Petri Nets (ICPN) and Timed Automata. These tools allow us to evaluate, respectively, the quality variations and to manage the flow type disturbance. An example analysis illustrates our approach.

1. INTRODUCTION

Discrete event systems is an area of research that is developing within the interstices of computer, control and communication sciences [1]. The basic direction of research addresses issues in the analysis and design of distributed intelligent systems operating within real time constraints. In view of the relatively long history of prior approaches to discrete event control design (notably discrete event system simulation, and analysis via Petri nets, starting in the 1960s, and investigations via stochastic models, including perturbation analysis from the early 1970s), Consequently a suitable model, rich in analytical properties, is necessary to synthesize the needed controller. Petri net is being widely accepted by the research community for modelling and simulation of DES. The modeling formalism of discrete event systems is suitable to represent man-made systems such as manufacturing, telecommunication, transportation and logistics systems [2-5]. The literature suggests different modeling techniques for DES such as automata [6], Petri-nets [7], or algebraic state space models [8-11].

The aim of this paper is to explain the use of hybrid tool which combines Interval Constrained Petri Nets (ICPN) and timed Automata. ICPN is a sub-class of High Level Petri Nets with Abstract Marking (AM-HLPN) [12-14]. ICPN model allows us to model and guarantee a constraint on any parameter of a manufacturing process. In our case, it has been used to model the flux of material flow for regulating the quality of tobacco. Timed automata is a tool for the modelling and verification of real time systems [15, 16].

A timed automata is essentially a finite automata (FSM), extended with real-valued variables. Such an automata may be considered as an abstract model of timed systems. This expressive modelling tool offers the possibilities of model analysis like verification, controller synthesis and also faults detection and isolation to model manufacturing systems whose activity times are included between a minimum and a maximum value. We use it for modelling the discrete parts of system command. Both tools are applied to a robustness control for regulation systems and for description of dynamical system.

*Corresponding Author Email: lotfi.mim@yahoo.fr (Mhamdi, L)
In this paper, first, the process is presented. The following section describes ICPN and gives some basic notions on the timed automata which is used in the modeling step. Then, the ICPN mode which describes the material flow is built. This tool presents a complement to the P-temporal Petri nets [12]. Therefore, the robust control laws of this model are proven to use production data information of manufacturing production systems. Then, we use the timed automata model to describe the control law. Finally, when the global model of the process is completely defined, we present an application to tobacco manufacturing to illustrate our approach. Finally, a conclusion is presented with some perspectives.

2. MATHEMATICAL MODEL

The problem which lays in the cigarette transformation and production systems is dependent on the weight of the manufactured units.

Indeed, from a quality point of view, a cigarette which is too heavy has a difficult drawing, and a light cigarette gives the consumer the impression of bad quality, because its extremities are not well filled and can easily fray. From a point of view of cost, the excess quantity of tobacco in a cigarette is considered as a loss and can cause blockage due to stuffing in the circuit of the tube formation.

For a given cigarette, the weight depends on six parameters: the compactness (C) of the tobacco, the level (H) of the supplying conveyor, the trimmer setting (E), the tobacco density (d), the module (m), and the length (l) of the cigarette. The evolution of these parameters is proper to the typology of the considered cigarette fabric.

The compactness (C) of the tobacco is a parameter of the system entry to be considered. Its value depends essentially on the raw material. Its variation is random and its distribution can be modeled by a normal law (according to the production statistical data). The level of tobacco varies depending on the compactness and has to be in a given range to guarantee the good functioning of the system. This level is a variable that depends on the tobacco flow, a parameter which must be supervised in order to maintain an optimal density.

This flow is controlled in two steps: at the moment of filling the distributor by a control C₁ and at the moment of driving tobacco on the conveyor by a control C₂.

The control C₁ maintains the level of tobacco in the girdle (tobacco reservoir). The level must be kept between a minimum n₁ under which the machine stops, locking the tobacco, and a maximum level n₂ over which there is a need for supply. The control C₂ regulates the level of tobacco (H) on the conveyor.

It must be kept between a minimum level n₃ under which the tobacco driving cylinder stops, locking tobacco, and a maximum level n₄ over which the tobacco driving cylinder stops. The trimming is adjustable by a mechanical system which the operator can manipulate in order to adjust the quantity of tobacco in the cigarettes. In conclusion, the cigarette transformation consists of two stages (Figure 1).

- **Flow Regulation:** To obtain a homogeneous density of the tobacco. This density depends on the compactness of the tobacco (C), the level (H) of tobacco on the conveyor, and the quantity of tobacco trimmer (E) such as:

\[ d = \alpha \times C \times H \times E \]  

(1)

were:
- C: compactness in g/mm³, where C ∈ [Cₘᵢₙ, Cₘₐₓ];
- H₁: level of tobacco in the reservoir in mm, where H₁ ∈ [H₁ₘᵢₙ, H₁ₘₐₓ];
- H₂: level of tobacco on the conveyor in mm, where H₂ ∈ [H₂ₘᵢₙ, H₂ₘₐₓ];
- E: trimmer of tobacco flow, where E ∈ [Eₘᵢₙ, Eₘₐₓ];
- \( \alpha \): is a constant value.

- **Cigarette Formation:** this stage consists of enveloping the bundles of tobacco obtained, by the cigarettes paper. The tobacco band is obtained with a diameter (m) and is cut into sections of length (l) of a cigarette to get the consumption unit. Consequently, the cigarette weight can be written:

\[ P = km \cdot l \cdot d \]  

(2)

where:
- m: diameter of the cigarettes in mm.
- l: length in mm.
- d: density in g/mm³.
k: constant value (π/4).

Obviously, the variation of one of these parameters provides a variation of the weight. In this work we are interested on the first stage. Our goal is to maintain the density within the specified limits by changing the setting parameters (C, H and E), while they have to remain in a valid range.

3. MATRX GEOMETRIC SOLUTION

Several classes Petri networks have been developed, each trying to describe a “view” of production systems, their design and conduct. Among those classes, we include models that integrate the dimension of time [17-21]. The extension covers the modeling systems whose behaviour depends on an explicit time values. The resulting model can treat problems related to the analysis and evaluation of performance through the analytical methods. Consequently, Petri nets have been used to model various kinds of dynamic event-driven systems like computers networks [22], communication systems [23, 24], manufacturing plants [25-27], command and control systems [28], real-time computing systems [29, 30], logistic networks [31] and workflows [32-34] to mention only a few important examples. The study of the workshops with temporal or non temporal constraints contains a singular problem which occurs when one is in the presence of a synchronization mechanism. We choose the Petri nets (Intervals Constrained Petri Nets) and the timed automata as modelling tools. In fact, these tools are known as being powerful tools of modelling of Discrete Event Systems.

3. 1. Interval Constrained Petri Nets  ICPN are introduced to extend the application [35, 36] field of P-Time PN by proceeding to a functional abstraction of the parameter associated places. Furthermore, introducing a new formalism is an opportunity to review the initial definition. Thus, we present in an unequivocal manner the marking as a multi-set. In the same way, the transmission of a quantity conveyed by a token is represented explicitly.

3. 1. 1. Definition  An ICPN is a t-uple < R, m, IS, D, Val, Val0, X, X0 > where:

- R is a marked PN
- m is an application associating token to places:

Let Va be a set of rational variables.

Let V be a non empty set of formulas to use a variables of Va.

Let μV be a multiset defined on V. m: P → μV

p ∈ P → m(p), where m(p) is a place marking. We note M the application: M: P → N (set of positive integer)

p → Card(m(p))

IS: P → R \{[−∞, +∞] × R ∪ (Q’ ∪ [−∞, +∞])

defines the intervals associated to places. R is the set of real numbers.

p → IS = [a, b] with 0 ≤ a ≤ b

D is an application that associates to each pair (place, token) a rational variable q(q ≤ h). This variable corresponds to a modification of the associated value of a token in a place.

D: m(p)×P → Va

∀i, 1 ≤ i ≤ n, n = Card(P). Let k be a token, k ∈ m(p) .

k → q|a ≤ q ≤ b, where a, b are rational values fixed by IS

X is an application that assigns to each variable a value. X: Va → Q; va → u ∈ Q: X sets the qi.

X0 defines the initial values of variables.

Val associates to each token a formula of values in Q. Val is an application of set of the tokens m(p) in V: m(p) → V ( (k ∈ m(p)) → v ∈ V ), where k is a given token.

Val0 defines to initial formulas associated to tokens.

A mark in the place p, is taken into account in transition validations when it has reached a defined value between a and b. When the value is greater than b, the mark is said to be dead. Logically, in the firing of an upstream transition, tokens are generated in the output places and their associated variables are equal to:

Val(K) + q(K)

The signification of q and Val(k) are intentionally not fixed in order to provide a general model. As an example with P-time PN there is the following relation:

Val(K) + q(k) ; Where t represent time.

In ICPN the application, X is not mathematically imposed. We will meet, for example, applications where q parameters represent weight variations of cigarettes. In this case, parameter values associated to pairs (place, token) are independent.

3. 1. 2. State Definition  A state E is defined by a t-uple < m, D, Val, X > where:

m, D, Val and X are the above defined applications D and m assign a variable q(k) to each token k in a place pi.
A token \( k \) of the place \( p_i \) can take a part in the validation of output transitions if:

\[
q_i(k) \in [a, b],
\]

where \([a, b]\) is the static interval associated to the place \( p_i \). This token \( k \) dies when:

\[
q_i(k) > b
\]

\( X \) is an application which provides a value for each variable of \( V \). Actually, \( X \) defines the real value of each \( q \).

When \( X \) is not defined, there exists a way to make the model evolve. Furthermore, some mathematical properties may be outlined. It is the mathematical abstraction.

3.1.3. Computing the Next Step There are two different ways of reaching a state from a given one. The first solution is to use the evolution of associated variables. The second one is the transition firings. The following two definitions correspond to these two possibilities of evolution.

**Definition 1**: A state \( E'(m', D', Val', X') \) is accessible from another state \( E(m, D, Val, X) \) according to associated variable evolution if and only if:

1- \( m' = m \), and
2- \( \forall j \) a token in \( p_j \); \( q'_j(j) = q_j(j) + \Delta q_j(j), a_j \leq q'_j(j) \leq b_j \)

where \([a, b]\) is the static interval of the place \( p_j \). The possibility of reaching \( q'_j(j) \) depends generally on the coupling with other \( q \) evolutions. This particular aspect is not presented here.

**Definition 2**: A state \( E'(m', D', Val', X') \) is accessible from another state \( E(m, D, Val, X) \) by the firing transition \( t \), if and only if:

1- \( t \) is validated from \( E \),
2- \( p \in P \), \( m'(p) = m(p) - PrE(p, t) + Post(p, t) \)

\( PrE(p, t) \) corresponds to the weight of the output arcs from \( p \) to \( t \). \( Post(p, t) \) corresponds to the weight of the input arcs from \( t \) to \( p \).

3- Tokens that remain in the same place keep the same associated value between \( E \) and \( E' \).

The newly created tokens take null values for the \( q \) counter associated to their new places. The value allocated to the token \( k' \) by \( Val \) is:

\[
Val(k') = Val(k) + q(k).
\]

Where \( k \) is a token that is in an input place \( p_i \) of \( t \) and consumed to fire \( t \).

The previous firing rule allows computing states and accessibility-relationships. The set of the firing sequences from an initial state specifies the PN behavior as well as sets of accessible markings or validated firing sequences in the case of Autonomous PN.

3.2. The Timed Automata The fault diagnosis problem in DES is generally solved by using the model-based approach. In other words, an algorithm for fault detection and isolation will analyze and compare the model with the observed behavior. The timed automata is a well suited tool for modelling all the evolutions of a DES.

Timed automata is a tool for modelling and verification of real time systems [15, 16]. A timed automata is essentially a finite automata (FSM), extended with real-valued variables. Such an automata may be considered as an abstract model of timed systems. This expressive modelling tool offers the possibilities of model analysis like verification, controller synthesis and also faults detection and isolation. In the original theory of timed automata, it is a finite state extended with a set of real-valued variables modelling clocks. Constraints on the clock variables are used to restrict the behavior of timed automata, and conditions are used to enforce progress properties. A simplified version, namely Timed Safety Automata, is introduced in to specify progress properties using local invariant conditions. Due to its simplicity, a Timed Safety Automata has been adopted in several verification tools for timed automata. Using timed automata, this system is described in a qualitative and quantitative way. This is illustrated by an example on the Figure 2, where the qualitative parameters represent the sequence of events while quantitative ones relate to temporal parameters.

The problem of automata analysis is considerably more difficult in the timed case compare to the discrete case; in the discrete case, one deals with classical regular languages which have robust closure properties. The timed automata tool [37, 38] is defined as a finite state machine with a set of continuous variables that are named clock. These variables evolve continuously in each location of the automata, according to an associated evolution function. As long as the system is in one state \( Li \), the clock \( xi \) is continuously incremented.

Its evolution is described by \( X = 1 \). The clocks are synchonized and change with the same step. An invariant is associated to each state. It corresponds to the conditions needed to remain in the state. The number of clocks depends on the parallelism in the system. The automata can stay in one state as long as the invariant condition is checked.

3.2. The Timed Automata

Figure 2. Example of Timed Automata
Each transition of an automata is conditioned by an event or temporization called “guard” and its execution determines the discrete evolution of the variables according to its associated assignment.

Let us consider the timed automata given in Figure 2. This automata has two clocks $x_1$ and $x_2$. The continuous evolution of time in this model is represented by $\dot{X}=1$ and the labeled arcs in the graph represent the model of discrete evolution. The guard in each arc is a transition labelling labeling function that assigns firing conditions with the transitions of the automata. The affectation is a function that associates with each transition of the automata one relation that allows actualizing the value of continuous state space variables after the firing of a transition. The invariant in the state $L_0$ variables after the firing of a transition. The affectation is a function that associates with each transition of the automata one relation that allows actualizing the value of continuous state space variables after the firing of a transition. The invariant in the state $L_0$ and $L_1$ are $x_0 \leq 2$ and $x_2 \leq 3$, respectively. The initial state of this system is represented by an input arc in the origin state ($L_0$). In the dynamic model, active clocks are found in each state. A graphical interpretation of the timed automata is the automata graph (Figure 2).

3.3. Modelling Process with ICPN

It is possible to construct a RdP model of a shop in order to study the different laws of order while simulating the statistical distribution of the orders on the diameter [39].

However, it is necessary to assume that the synthesis of the model is subject to mistakes and approximations. In the best of the cases, we can recover on the model the robustness of the parameters directly on statistics of the shops are available. Then it is possible to calculate the tolerances of the parameters directly on statistics of the shop. These last results will be logically more correct than those that integrate the imperfections ensuing of the modelling phase.

Note that these parameters in Equation (1) ($d$, $C$, $E$, $H$) are related to each others. Obviously, the variation of one of these parameters provides a variation of the weight. When it is outside the validity range, the production has to be rejected, or the machine blocks. Our objective is to make sure that the permitted tolerance concerning the weight of the cigarettes will be respected by controlling $H_1$, $H_2$ and $E$ parameters. It must belong to a predefined range The aim of the controller is to maintain the density specification by changing the setting levels $H_1$ and $H_2$, whereas they have to remain in a valid range.

We consider the variations of a parameter are always very small in contrast to its setting value. Consequently, the above relation may be approximated by the following one doing a first order linearization [40]. A first step develops nonlinear relation (1). This gives the relations (5) which describes the behavior of the process around a reference state.

$$\Delta d = \alpha_1 \Delta C + \alpha_2 \Delta H_1 + \alpha_3 \Delta H_2 + \alpha_4 \Delta E$$  \hspace{1cm} (5)
using timed automata. Then we use the approach developed by Lunze [42]. These approaches are based on principle of state and events by integrating the time. At first, we consider that the production sequence is as following:

The system starts from a known initial state. Firstly, the shaft-off flap open; the tobacco flows into the reservoir. After duration time of \( t_1 \), the level \( n_2 \) is reached, then the drive motor rotates at low speed and the cylinder begins to rotate and the tobacco flows on the conveyor belt. After duration time of \( t_2 \) the level \( n_4 \) is reached and the machine goes to the nominal speed. As the level in the reservoir and on the conveyor are respectively greater than \( n_1 \) and \( n_3 \) the system is functioning normally.

The control \( C_1 \) maintains the level \( H_1 \) of tobacco in the girdle (tobacco reservoir). The level must be kept between a minimum \( n_1 \) under which the machine stops, locking tobacco, and a maximum level \( n_2 \) over which there is a need for supply.

The control \( C_2 \) regulates the level of tobacco \( (H_2) \) on the conveyor. It must be comprised between a minimum level \( n_3 \) under which the tobacco-driving cylinder stops, locking tobacco, and a maximum level \( n_4 \) over which the tobacco-driving cylinder stops.

The control \( C_3 \) regulates the trimmer (E). A trimmer motor (M) consists of an electrodynamics motor whose output shaft is mechanically linked to the trimmer disks through a quadrant gear. By driving the trimmer disks, the tobacco rod weight can be adjusted.

This kind of motor allows carrying out rapid and precise movements. Due to a position transducer (T) integrated in the motor, the trimmer disks position is controlled in closed control loop. Data from the scanning unit is processed in the electronic rack, which sends a correction command to the trimmer if the cigarette is outside of the user’s defined tolerances. The control sequence is then the following:

- **S_0**: is the initial position when the machine is initialized.
- **S_1**: the shaft-off flap open the tobacco flows into reservoir.
- **S_2**: After time \( t_1 \) the cylinder begins to rotate and the tobacco flows on the conveyor belt.
- **S_3**: After an additional time \( t_2 \) the machine reaches its first speed.
- **S_4**: After an additional time \( t_3 \) the machine reaches its nominal speed.

The proposed approach deals with time analysis of timed automata. The principle of modelling task is to follow the control sequence. Then, let us start building the state space model of the system. We use the levels in reservoir and on the conveyor (two states for each: LOW and HIGH). This leads to four states for the timed automation, which correspond to all the possibilities for the levels of tobacco.

- State \( L_1L_2 \ (n_1n_3) \) denotes LOW level in both parts.
- State \( L_1H_2 \ (n_1n_4) \) denotes LOW level in reservoir and HIGH level on the conveyor.
- \( H_1L_2 \ (n_3n_1) \) denotes HIGH level in reservoir and LOW level on the conveyor.
- \( H_1H_2 \ (n_3n_4) \) denotes HIGH level in reservoir and HIGH level on the conveyor.

4. APPLICATION: DESIGNS OF EXPERIMENTS

Figure 4 describes the global model of the process of Figure 1. To construct this global model with validity intervals, a new modelling tool has used ICPN in order to evaluate the variations of the quality of tobacco and Timed Automata to manage the flow type disturbance.

Thus, when the model is built, it is possible to describe the constraints on the quality parameters which are required for manufacturing of products in accordance with the specifications.

This model allows setting the system functioning around a target state. Another policy is to control the workshop while following the evolution of the parameters in the course of time in order to compensate for the fluctuations. Note that in this global model we added places representing optimal states "SM_1, SM_2 and SM_3" process operating.

![Figure 4. Global ICPN model with command (flow-quality)](image-url)
4. 1. Control Maintaining Constant Density  The effective value of parameters can be calculated with polynomial algorithms [43]. This can be done because the above algorithm is only based on the structural properties of P-time Petri Net. In this case, it has been shown that under some particular assumptions, the property may be extended to ICPN [44].

4. 2. Computing the Robust Control  Finally, when the ICPN model of the process is completely defined, it is possible to analyze the structural properties. It has been proven that most of the structural properties of P-time PN can be extended to ICPN [45, 46]. An internal robustness analysis of the ICPN model of the presented process is published in [13].

Finally, using the information from production data, a computing methodology has been applied in order to build the valid intervals of the ICPN model.

This approach uses only a sub-part of the information, because we only want to find critical tests which are needed for the designs of experiments applied in the production data.

An observation of the tobacco processing by different units during one month has resulted in picking out the variations of the output measures: compacity, trimmer and the tobacco density.

Figures 5 and 6 represent, respectively, the variation of trimmer tobacco and density. It has plotted control limits that present regulation boundaries, which are managing boundaries, and a centreline gained by calculating average arithmetic value of the measurement samples. In our case, the measured values are within control limits and then the process is under control.

A centreline (CL) represents the mathematical average of all the samples plotted. UCL and LCL present respectively the upper and lower statistical control limits that define the constraints of common cause variations.

5. CONCLUSION

In this paper, a methodology of design and modelling of control laws is adopted. We have modelled using the ICPN tool which presents a functional abstraction of the P-temporal Petri Nets, constraints subjected on flow and quality parameters while integrating the margins of robustness. The goal is to satisfy qualitative and quantitative needs of the market.

It is to be noted that the presented data are real production data of an existing workshop. The proposed methodology is therefore validated by a large set of data, and it provides an interesting industrial efficiency for the considered case study.

The next step in our work is the model checker. It means to verify if all faulty states in the dynamic model are reachable, or it is necessary to add some other sensors to isolate the faults.

6. REFERENCES


Using Interval Petri Nets and Timed Automata for Diagnosis of Discrete Event Systems (DES)

L. Mhamdi, H. Dhouibi, Z. Simeu-Abazi, N. Liouane

* UR ATSI D. Génie Electrique de L’ENIM, Monastir - Tunisie
a Grenoble-INP / UJF-Grenoble 1 / CNRS, G-SCOP, Grenoble, 46, avenue Félix Viallet - 38031- France

PAPER INFO

Paper history:
Received 10 June 2013
Received in revised form 27 August 2013
Accepted 14 September 2013

Keywords:
Modeling
Petri Nets
Interval Constraint Petri Nets
Timed Automata
Robustness

doi: 10.5829/idosi.ije.2014.27.01a.14