Analysis of Manufacturing Lines using a Phase Space Algorithm with Discrete Control

T. El-Fouly - N. Zerhouni - M. Ferney- A. El Moudni
Université de Technologie Belfort-Montbéliard,
Site de Belfort
Laboratoire d'Automatique Mécatronique Productique et systèmique (L.A.M.P.S)
90010 Belfort Cedex
FRANCE

Abstract: - This paper presents an analysis of the dynamic behavior of manufacturing systems, as a basis for the control study. In this paper we present the special case of an open manufacturing line as an example. We present an analysis as well as an algorithm for minimizing the overall evolution time of a manufacturing line. The objective is to study the effect of changing the flow rate of the pieces arriving to the line on the overall performance of the manufacturing line (settling time, throughput) during the transient and the stationary regions. The system is modeled by a continuous Petri net receiving an input flow that changes discretely. The change of the flow rate is due to the control series applied to the model. IMACS/IEEE CSCC'99 Proceedings, Pages:2631-2639

Keywords: c

1. Introduction

This paper presents an intermediate step towards a control theory that could be applied first to manufacturing lines and which we aim to generalize to be applied to production systems. It presents a method for modeling manufacturing systems while conserving its continuity in order to study its dynamics. At the beginning the studied system is modeled by a continuous Petri Net which helps us to describe the dynamics of the system in the form of differential equations, each equation describes the evolution of the number of pieces stocked in the buffers accompanying each machine in the manufacturing line. Once the model is obtained and the differential equations are described we notice that the equations are nonlinear, which made us think in cutting the evolution or the lifecycle of the manufacturing process into different phases. A phase is a time interval where the dynamics of the system is constant and linear. The phase variation is marked by the variation of the state equations of the system, which could be translated in other words as a change in the dynamics of the system. The phase could be considered as a sub state space of the whole state space of the system. A stationary state is detected when the number of phases remains constant even with changing the studied system parameter after that. The use of our approach helps in applying a control approach after that. Our study aims to detect a series of source speeds in order to minimize the evolution time during a phase, to minimize the number of phase changes and to maximize the throughput. This paper presents also one of the steps taken to study the effect of changing the source speed on the transient and stationary states of the studied model. Another approach used a lot by other research groups is the hybrid automata [2],[3] and [4]. Section 2 presents an introduction to Petri nets and especially continuous Petri nets, which will be used in our study. Section 3 presents the application of Petri nets in the domain of modeling manufacturing lines. Section 4 presents an algorithm for constructing the phase space of the evolution of the system with varying the speed of the source. Section 5 presents the effect of speed profile variation on the evolution of the system. Different speed profiles are used and the maximum number of phases is obtained.

2. Petri Nets

The Petri Net is a graphical utility for describing the relation between events and conditions [7]. It permits to model the behavior taking into consideration the sequence of activities, the parallelism of activities in time, the synchronization of the activities, and the resource sharing between activities. A Petri Net is an oriented graph consisting of places [5], transitions and arcs. The arcs connect between a place and a transition or vice versa. These arcs indicate the direction of flow of marks. For a manufacturing system, the different places of the Petri net model the different buffers of the system while the transitions model the different machine [1].

A Petri Net is defined by 4 variables, 2 sets and 2 applications [5] PN = < P, T, Pre, Post>

Where:

$P = \{P_1, P_2, ... P_n\}$ is a set of places, $T = \{T_1, T_2, ... T_n\}$is a set of Transitions.
Pre: $P \times T \rightarrow N$ $(P_i, T_j) \rightarrow$ Pre (P_i, T_j)=arc weight between P_i and T_j.
Post: $P \times T \rightarrow N$ $(P_i, T_j) \rightarrow$ Post (P_i, T_j)=arc weight between T_j and P_i.

Incidence Matrix: W = [Post (P_i, T_j)]-[Pre (P_i, T_j)]

There are many types of Petri nets, each one models a particular class of systems, i.e. the timed Petri nets and the continuous Petri nets. There are two ways of assigning timing to the timed Petri nets, either by assigning the timing to the transitions or by assigning the timing to the places. The timed Petri nets model a system having the number of marks circulating in the system is not important. But if this number of marks is important (explodes) the continuous Petri nets are used in this case. A continuous Petri Net is a model where the marking in the places is real and the transitions are continuously fired [5]. There are two types of continuous Petri nets. The first is called constant speed continuous Petri net which is characterized by having all of its transitions having a constant firing speed $V = \frac{1}{d}$, while the second one is called variable speed continuous Petri net which is characterized by having a variable firing speed $V = \frac{1}{d} min(1, m_1, ..., m_i)$ associated to its transitions.

3 Modeling A Manufacturing Line using Continuous PN

In this section we will make use of the continuous Petri Net model presented in section 3 in the domain of manufacturing systems. Applying this to an open manufacturing line having *n* working benches, a source, and an output buffer as presented in the following figure.

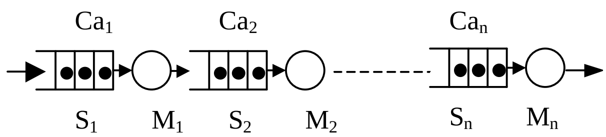

Fig. 1 Open Manufacturing Line

Each bench consists of a buffer and a machine. The buffers are considered to have finite capacities. The following Petri Net model models this manufacturing line:

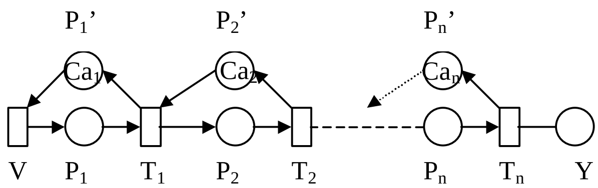

Fig. 2 Petri Net Model

The Petri net model presented in Fig 2. models the open manufacturing line of Fig 1. Where a machine is modeled by a continuous transition having the same speed while a buffer is modeled by a continuous place because as noted before we consider in our system finite capacity buffers, a buffer i will have the capacity Ca_i. An equation for the system's evolution is needed to be established. This equation must be function in the initial marking. Initially we consider all places are empty while the speed vector of the n transitions is given by:

$$U = [U_1, U_2, ... U_n].$$
$$M_0 = [0, 0,0...0]$$

The flow of pieces entering the system to be manufactured is defined by the speed of the source machine given by symbol *V*. The number of pieces produced by this system is given by *Y*.

The differential equations that describes the evolution of the system are:

$$\begin{cases} \dot{m}_1 = V - U_1 \min(1, m_1) \\ \dot{m}_2 = U_1 \min(1, m_1) - U_2 \min(1, m_2) \\ \bullet \\ \bullet \\ \dot{m}_n = U_{n-1} \min(1, m_{n-1}) - U_n \min(1, m_n) \end{cases}$$

And:

$$\dot{Y} = U_n \min(1, m_n)$$

We could notice from the previous equation that it is nonlinear because of the presence of the function **min**, this non linearity led us to define the phase concept. For the initial state we have all places initially empty without any marking, we will have the equations:

$$\begin{cases} \dot{m}_1 = V - U_1 \cdot m_1 \\ \dot{m}_2 = U_1 \cdot m_1 - U_2 \cdot m_2 \\ \vdots \\ \dot{m}_n = U_{n-1} \cdot m_{n-1} - U_n \cdot m_n \end{cases} \quad \text{If } M_0 = [0, 0, 0... 0]$$

And:

$$\dot{Y} = U_n \cdot m_n$$

Or in matrix form:

$$\dot{M} = A \cdot M + B \cdot V$$

And the output:

$$\dot{Y} = C \cdot M + D \cdot V$$

Where:

$$A = \begin{pmatrix} -U_1 & 0 & 0 & \cdot & \cdot & 0 \\ U_1 & -U_2 & \cdot & \cdot & \cdot & 0 \\ 0 & U_2 & -U_3 & \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & -U_{n-1} & 0 \\ 0 & 0 & \cdot & \cdot & U_{n-1} & -U_n \end{pmatrix}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ 1 \end{pmatrix}^T \quad \text{And } D = 0$$

The analysis presented study the effect of changing *V* on the system's behavior. The simple analytical methods can't solve the previous system of equations because the matrix A could be a singular matrix depending on the simulated phase. The Maple V5 R5 software allows us to solve numerically such systems of equations based on the values of A, B,C and D. But the numerical solution of this system of equations is not our goal because what we need is an analytical expression describing the vector M(t). This expression will be used to define the dynamics of the system during the phase. Decomposing the function that caused the nonlinearly which is the **min** in our case to its components the result are different equations, each could be considered to define the dynamics during a phase. In the next section we will describe the concept of creating the different phases because the choice is not that simple because at the same time of choosing the phases the evolution time during a phase is to be minimized and the output is to be maximized. So the goal is not just to cut the lifecycle of the manufacturing process into different phases but it is also to minimize these phases duration. Using the expressions describing the marking of the different places, the different evolution times are calculated and then the smallest one is chosen. Choosing the minimum time of evolution for each phase allows us after that to construct the control sequence to be applied to the system to minimize the evolution time required to reach the desired state. A tool has been developed using the software tool Maple V5 R5 to calculate the different evolution times. The tool gives after that the different phases of the system.

4 The Modeling Algorithm

In this section we present an algorithm for modeling the manufacturing lines using the phases concept. The algorithm is as shown in Fig. 3. This algorithm constructs the phase space for the studied model.

The algorithm starts with constructing the Petri Net model for the manufacturing line taking into consideration that a buffer is modeled by a continuous place and a machine is modeled by a continuous transition having the same speed. After constructing the PN model the initial state is defined. Using the previously defined initial speed a value for the speed of the source V is chosen. With this value of V, the evolution time during the first phase is calculated and the place that reaches the unity value is chosen. Assuming that V is a variable, the minimum evolution time could be calculated depending on V. The obtained minimum time is called t_{min} and the corresponding source speed is V_{opt}. This procedure is repeated until a stationary state or the desired state is reached. When the desired state is reached the phase schema is constructed in order to predict the evolution of the studied system. This algorithm assures the existence of a minimum evolution time. It could also be considered as a step towards an optimal control with respect to the evolution time or with respect to the time to reach a desired state. Something must be noted here that the speed of the different transitions is bounded between two given values noted as V_{min} and V_{max}. These boundaries or values depend on the capabilities of the machines. The range of variation of the speed V also depends on the used source, and the maximum and minimum production rate produced from it.

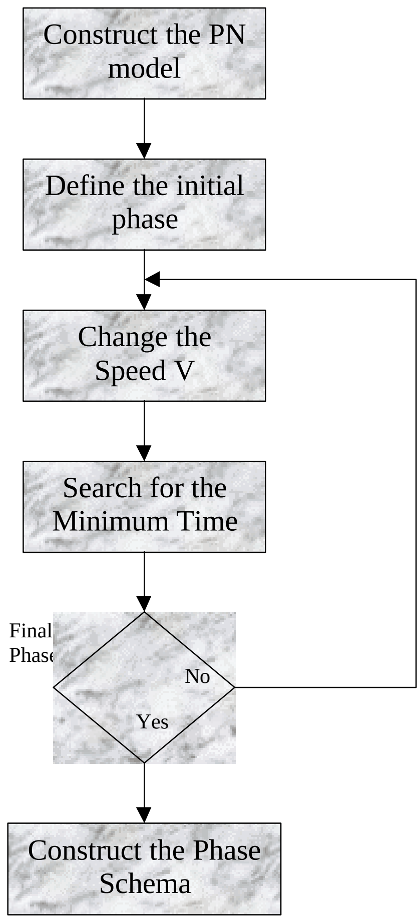

Fig. 3 The Modeling Algorithm

The previous algorithm uses the differential equation described in the previous section to construct the marking evolution equations during the studied phase. Although the speed V is constant during the phase, the equations describing the evolution of the marking during the phase will be defined as function in V, so as to be applied to the algorithm that will calculate the appropriate value of V to be applied to this phase.

$$m = f(t,V)$$

And since a phase variation is signaled by a place having its marking passing the unity value, the minimum evolution time is obtained for every place in the PN model by substituting:

$$m = f(t,V) = 1$$

Then:

$$t = g(V)$$

To get the minimum of all the cases over the range of speed provided by the source [V_{min}, V_{max}]:

$$t_{\min} = \min_{V = V_{\min} \to V_{\max}} \left(g(V) \right)$$

Comparing the different evolution times for the different places of the PN model, the minimum is chosen and the corresponding source speed V is added to the control series.

Applying the same concept to all different phases, the evolution of the system during a certain interval of time could be known. The control series to be applied to the system could also be known during this interval.

Before applying the algorithm there was a very important step to perform which was to choose which speed profile is to be applied to the different transitions. This profile could affect to a certain extent the performance of the system. In the next section we will present a study to show the effect of changing the speeds profile on the maximum number of phases.

5 Effect Of Speed Variation.

The effect of varying the speeds of the different transitions on the number of phases[6] is presented in this section. The effect of this variation will be also directed towards the effect of reaching the desired state in a controlled manufacturing line.

A maple tool had been developed to help in obtaining the desired results, this tool helps in constructing the different phases with respect to variations in the speeds. The system used in this section is an opened manufacturing system that is modeled by a Petri Net as shown in Fig.2.

The importance of this study is to choose the most appropriate speed profile to be applied to the system. In the next example we will choose the worst case and we will try to apply the algorithm on this case so if a solution is found, then all the other cases by consequence are most probably solved. The study resulted in the conclusion that when the speed profile $U_1>U_2>\ldots>U_n$ is chosen, the maximum number of phases is obtained. This case had been chosen to assume the worst case when the system passes by the maximum number of phases before settling.

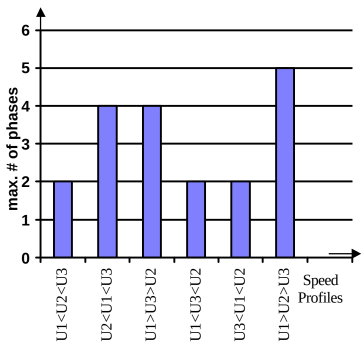

Fig. 4 Max. # Of phases variation with respect to different speed profiles.

Fig.4. presents the results obtained for the case of 3 working benches example (when n = 3). The speed profile that gave the maximum number of phases is then $U_1>U_2>U_3$. With the obtained result we chose the previous profile to perform the study.

All the results obtained in the previous part is with respect to variations in the transitions speeds of all the transitions except the source transition that we fixed in this section which is considered in our algorithm as a control parameter to minimize the evolution time to reach the steady state.

6 The Phase-Control Algorithm

The phase-control algorithm is the algorithm which helps us to construct the phase schema and at the same time to construct the control sequence to be applied to the system during the real evolution of the studied system.

The inputs to the algorithm are the initial state and the control variable to be applied to the system. The initial state consists of the initial marking of the different places of the Petri net model, and the speed vector defining the speeds of the different transitions of the Petri Net model. The control variable in our case is the flow of pieces supplied to the system or in other words the speed of the source.

After providing the initial state and the control variable to the Petri Net model, the first phase could be constructed without any problems. This is due to the fact that the phase depends a lot on its initial state. For each studied phase there are 2 inputs and an output. The 2 inputs are the initial marking of the different places of the Petri Net model for this phase which is by its turn the final marking for the previous phase. The Second input is the optimal evolution time for the previous phase. The output from a phase is the equations defining the evolution of the marking of the places of the Petri Net model.

The second step in this algorithm is providing the outputs of the first phase to the controller. The controller is the module that will calculate the optimal evolution time for the first phase and at the same time it will calculate the corresponding source speed. This is done as shown in the previous algorithm, where the equations describing the evolution of the marking during the studied phase is applied to the controller.

After that the desired state to be reached is checked if it belongs to this phase, if it belongs the algorithm stops and the system has reached the desired state, if not the algorithm continues with the next step. The equation describing the fact of choosing the best time is as follows:

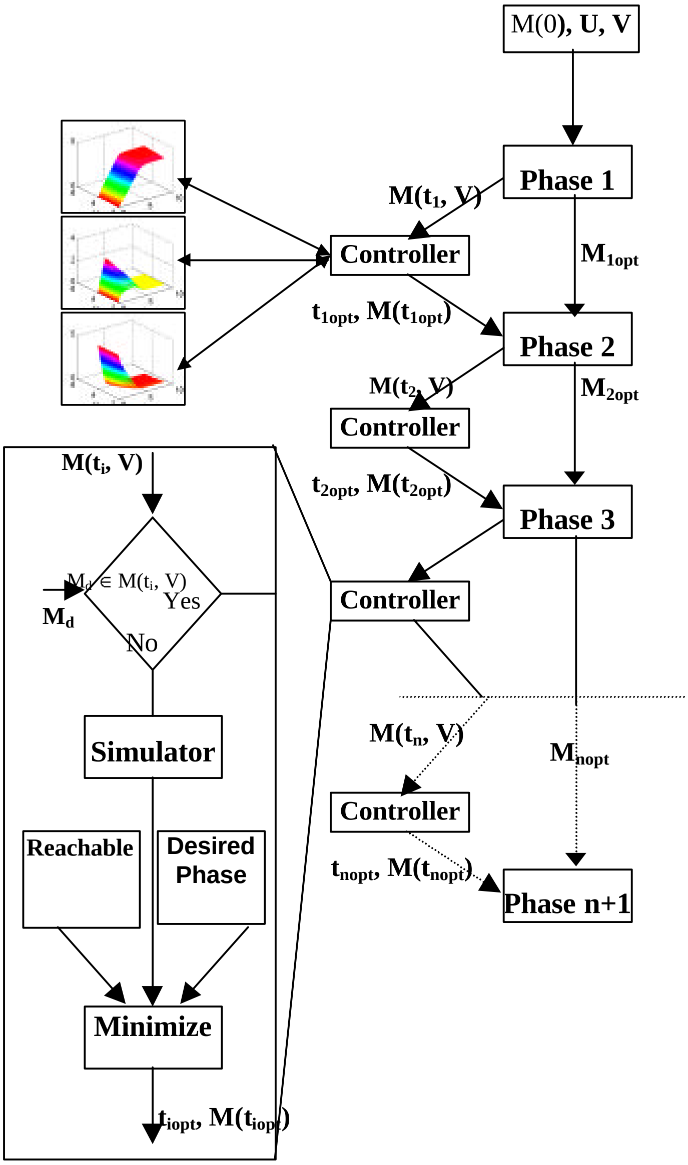

Fig. 5. The Phase-Control Algorithm

The Controller performs a three-dimensional simulation to determine the place that will be responsible for the next phase variation. There are many reasons to have a phase variation, for example if the marking of a place reaches the unity value or if the marking in a place reaches its maximum capacity. This phase is checked after that with the list of reachable phases by the studied system and also with the desired phase which contains the desired state. Using these simulations the minimum evolution time and the corresponding marking of the places are calculated.

The output of the controller is after that applied to the next phase. This algorithm continues until a stationary state is reached or the desired state is reached in the case of searching for a one.

7 Illustrative Example

The system presented in this example is an open manufacturing line consisting of 3 working benches each consists of a machine and an accompanying buffer. This line is presented as in Fig. 5 :

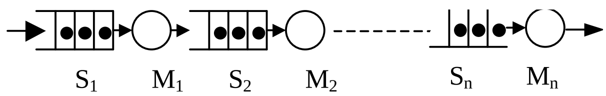

Fig. 6 Open Manufacturing Line with 3 working benches

This open manufacturing line is modeled by the Petri net model of Fig. 7, where a buffer is modeled by a continuous place and a machine is modeled by a continuous transition. The speed range that could be provided by the source is [0,10].

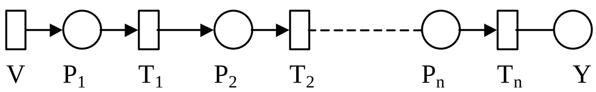

Fig. 7 Corresponding Petri Net model

The equations that describes the marking evolution in the different places of the Petri Net model:

$$\dot{m}_1(t) = V - U_1 \cdot \mathbf{min}(1, m_3(t))$$
$$\dot{m}_2(t) = U_1 \cdot \mathbf{min}(1, m_1(t)) - U_2 \cdot \mathbf{min}(1, m_2(t))$$
$$\dot{m}_3(t) = U_2 \cdot \mathbf{min}(1, m_2(t)) - U_3 \cdot \mathbf{min}(1, m_3(t))$$

And

$$\dot{Y}(t) = U_3 \cdot \mathbf{min}(1, m_3(t))$$

The speed vector is defined by $U = \begin{pmatrix} 4 & 3 & 2 \end{pmatrix}$

The initial marking vector is defined by $M_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

And $Y(0) = 0$.

And since we work with limited capacity buffers, then the vector presenting the maximum capacity of the corresponding places is given by Ca = (10, 3, 2.3)

Using the defined initial values, the equations that defines the marking evolution in the different places of the Petri net could be given by:

$$\dot{m}_1(t) = V - 4 \cdot \mathbf{min}(1, m_1(t))$$
$$\dot{m}_2(t) = 4 \cdot \mathbf{min}(1, m_1(t)) - 3 \cdot \mathbf{min}(1, m_2(t))$$
$$\dot{m}_3(t) = 3 \cdot \mathbf{min}(1, m_2(t)) - 2 \cdot \mathbf{min}(1, m_3(t))$$

And

$$\dot{Y}(t) = 2 \cdot \mathbf{min}(1, m_3(t))$$

Then the equations describing the marking in the different places of the PN model during the first phase could be given by:

$$m_1(t) = V * \left(\frac{1}{4} - \frac{1}{4} e^{-4t} \right)$$
$$m_2(t) = 4 \cdot V \cdot \left(\frac{1}{12} - \frac{1}{3} e^{-3t} + \frac{1}{4} e^{-4t} \right)$$
$$m_3(t) = 12 \cdot V \cdot \left(\frac{1}{24} - \frac{1}{8} e^{-4t} + \frac{1}{3} e^{-3t} - \frac{1}{4} e^{-2t} \right)$$

And the output:

$$Y(t) = 24 * V * \left(\frac{1}{24} t - \frac{13}{288} + \frac{1}{32} e^{-4t} - \frac{1}{9} e^{-3t} + \frac{1}{8} e^{-2t} \right)$$

Then the marking in the different places could be simulated with respect to the velocity of the source V and the time t. As an example the simulation of the marking in the first place and the output place is in Fig.8 and Fig. 9.

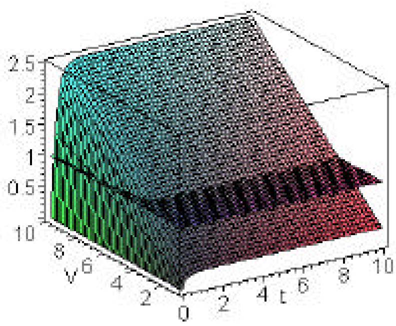

Fig. 8 The marking M_1(t)

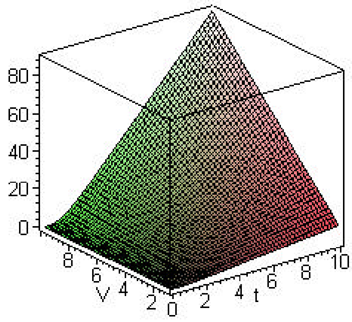

Fig. 9 The marking Y(t)

Fig.8 and Fig.9 show the marking in the first place and in the output place during the first phase. It could be observed from the simulations that the marking in the different places during a certain phase depend on the source velocity V. The algorithm used for the simulation is then the algorithm defined in section 5. So if the evolution time is to be minimized during a certain phase then the velocity V chosen during that phase must be appropriately chosen. The curve shown in Fig.10 for example shows the marking evolution in the first place with respect to a wide range of chosen velocities for the source. To observe a phase variation the marking must pass the unity value so focusing more on the relation between the velocity and the marking the following curve is obtained:

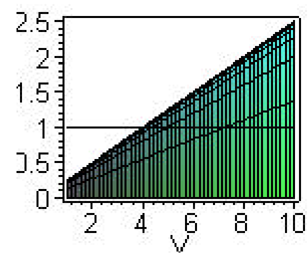

Fig. 10 Variation of M with respect to V

Fig.10 is the projection of the curve drawn in Fig.8 on the M,V plane. Estimating the results from this curve only could lead to a false result. This is because the result obtained from a very fast glance to this curve is that the most appropriate speed would be V = 4 but this is not the case when taking into consideration the following curve also:

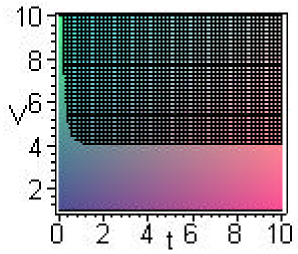

Fig. 11 Relation between V and t

Fig.11 shows that the time depends inversely on the speed V during the first phase.

This conclusion will lead to choosing the source speed to be the maximum possible speed allowed by the source V_{max}.

The overall curve describing the relation between the marking in the different places and the evolution time is as follows:

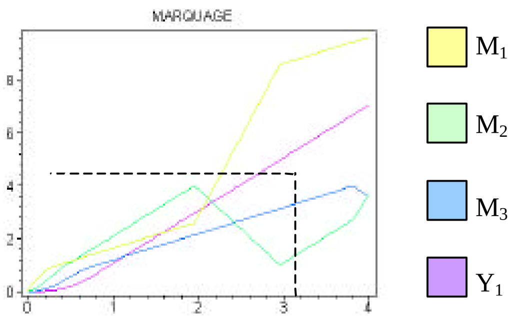

Fig. 12 Evolution without applying the algorithm

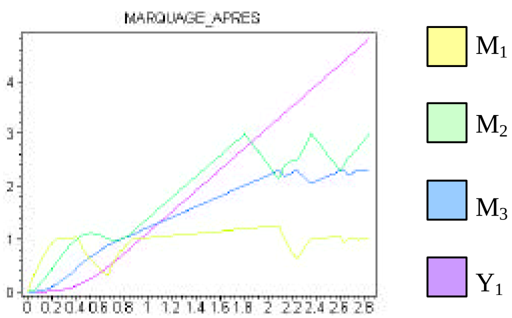

Fig. 13 Evolution with applying the algorithm

The results presented in the previous curves shows the effect of applying the algorithm to the overall performance of the system. The first remark is obtained by comparing the level of pieces in the first stock with the original one. The level or number of pieces ranges to a certain limit between 2 boundaries while in the original case the number of pieces could reach infinity. In the original case we need to use an infinite capacity buffer or the source of pieces supplying the system must be stopped which is not logical. But in our case the number of pieces is limited which is considered as a great advantage. The second remark is the throughput of the system which could be seen higher than the original case without applying the algorithm and which could be better also by adjusting the initial conditions and the maximum and minimum allowable speed for the source.

The algorithm presented in section 4 calculates the velocity that will minimize the evolution time. At the same time this algorithm will determine the place that will be responsible for reaching the second phase and in this example it is the first place. The algorithm continues to calculate the different places that will make the system reaches the next phase, the evolution time is calculated at the same time.

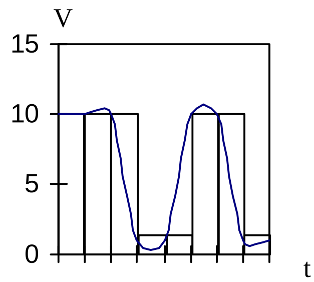

Fig. 14 The Control Series

After applying this algorithm to construct all the phases the control series could be constructed as shown in the previous curve. This series is in fact the different values for the source speed to be used during the manufacturing process.

We conclude from the results obtained from the previous example that using the modeling procedure presented in this paper gives a helpful way to minimize the evolution time and at the same time the algorithm used to minimize the evolution time is a good algorithm for open manufacturing systems. The tools developed and presented in this paper helps at the same time to predict if the desired state is reachable using the provided initial state, and this could help after that to define an initial state that will make the desired state reachable.

7 Conclusion

This paper present a method developed for modeling manufacturing systems, this method is based on a new concept of cutting the evolution of the system on different phases. Each phase is characterized by different dynamics. But at the same time this method saves the continuity of the system and it also saves the dynamics of the system during a certain phase and this could be recognized in the equations describing the marking of the different places of the PN model.

The study performed on the system for studying the effect of changing the speed profile on the performance of the system led to the importance of choosing an appropriate speed profile.

The performed study calculates the minimum evolution time with respect to each phase and the corresponding source speed. These speeds that were calculated during the simulation phase could be applied to the real system during the manufacturing process.

The developed algorithm not only tries to choose the appropriate speeds to minimize the time, but it also maximizes the throughput of the system. This is also important to guarantee the performance of the system while applying the algorithm during the simulation phase or while applying the control series during the manufacturing phase.

Using the same concept, the minimum time for reaching a desired state could be calculated. This could be considered as a step towards the control algorithm.

The developed algorithm was applied to manufacturing lines having infinite capacity buffers, as well as to manufacturing line having finite capacity buffers.

Our future work concerns also performing a study on comparing the performance of using the approximated continuous model and the initial discrete model.

References:

[1] H. Alla, J.B.Cavaille, J. Le Bail, G. Bel, *Modélisation par réseaux de Petri Hybrides et évaluation de Performances de systèmes de production par lots*, (Rapport Intermédiaire, 1992)

[2] M. Allam, H. Alla, Modeling Production Systems by Hybrid Automata and Hybrid Petri Nets, *Control of industrial systems, IFAC-IFIP-IMACS, 1* ,1997 ,463-468.

[3] R. Alur, C. Courcoubetis, T. A. Henzinger, And P. H. Ho, Hybrid automata: an algorithmic approach to the specification and analysis of hybrid systems, In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Workshop on theory of Hybrid systems, Lecture Notes in Computer Science* 736, 1993, 209-229.

[4] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, S. Yovine, The algorithmic Analysis of Hybrid systems, *Theoretical Computer Science* 138, 1995, 3-34.

[5] R. David, H. Alla, *Du Grafcet aux Réseaux de Petri*. Hermès:Paris.

[6] T. El-Fouly, N. Zerhouni, M. Ferney, A. El Moudni, Modeling and analysis of manufacturing systems using a hybrid approach, *3ème conference Internationale sur l'Automatisation des Processus Mixtes, ADPM'98 Reims*, 1998, P 79 – 85.

[7] J. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, 1981.

[8] J. A. Stiver, P. J. Antsaklis, M. D. Lemmon, A logical DES approach to the design of hybrid systems, *Technical Report of the ISIS Group (Interdisciplinary Studies of Intelligent Systems)* ISIS-94-011, University of Notre Dame, October 1994.