7th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS
S u c e a v a, R o m a n i a, M a y 27 – 29, 2 0 0 4

PETRI NETS LIVENESS CONTROLLER FOR FLEXIBLE MANUFACTURING SYSTEMS

Călin CIUFUDEAN[1] , Alexandru LARIONESCU[2] , Constantin FILOTE[3] , George MAHALU[4]
"Stefan cel Mare" University of Suceava
Str. Universitatii nr.1, RO-5800 Suceava
[1)] calin@eed.usv.ro, [2)] lari@eed.usv.ro, [3)] filote@eed.usv.ro, [4)] mahalu@eed.usv.ro

Abstract. *This paper generalises some previous approaches for modelling flexible manufacturing systems (FMS) with shared resources. We consider a class of controlled discrete – event systems modelled as controlled Petri nets. Our goal is to model a live system, using the concept of synchronic distances in Petri nets, with a liveness controller that can be used for verifying some other parameters, beside the liveness of the Petri net model of the FMS, such as availability of the system's components. An example illustrates the given approach.*
Keywords: *controlled Petri nets, synchronic distances, liveness controller.*

Introduction

Research on discrete-event systems (DES's) has focused on the synthesis of controllers for achieving desired behaviour [1] – [3]. As with continuous systems, some specifications for DES's are more important than others. For continuous systems, stability must be guaranteed before optimal performance can be considered. For DES's, it is most important to be ensured that the system never enters a state in which equipment will be damaged or costly error-recovery procedures become prohibitive. Such operational constraints are usually referred as forbidden state specifications [4]. Translating these considerations in the modelling field, we notice that a major concern, when modelling DES's using Petri nets, is to check whether the Petri net model has desired qualitative properties such as liveness, boundedness and reversibility. These properties characterise the behaviour of a well-designed system. As long as manufacturing systems are concerned, the liveness ensures that blocking will never occur, the boundedness guarantees that the number of in-process parts is bounded, the reversibility enables the system to come back to its initial state from any state it reaches. Therefore, the reversibility property is related to the concept of error recovery in manufacturing [5] because in the presence of some significant error, the system may automatically be reinitialised through a recovery process. Because we use, for the modelling process, sure connected graphs (SCG), it results that boundedness property of the Petri net models is ensured. Since for this class of nets, the sufficient conditions for liveness are the same as those for reversibility [5], [6], the liveness-checking algorithm can be used to check reversibility, too. More, based on some properties related to the concept of synchronic distances in Petri nets we build a controller for liveness checking of the DES's models. This Petri net controller has been proven to be versatile, that is when checking the liveness of the net (liveness which is ensured by the related properties), it found that the net was not live. The role of the controller is to verify the availability of the equipment that composes the flexible manufacturing systems (FMS's) as exponents of the DES's. A FMS consists of a number of systems, such as process actions, material storage, material processing devices, raw and finite material transportation devices, control units etc. The material flows among the flexible manufacturing cells, machines and equipment are usually connected through an automated handling system. Production control units, including process information and control commands are routed via a communication system. The communication system can have computers, control units, local area networks. The FMS can manufacture diverse types of products in variable batch sizes and meet fast transition of customer requirements. Therefore we choose FMS's as an example of DES's that have the ability to cope with rapid market and demand changes.

Controlled Petri Nets

In this paper we consider controlled DES's which can be modelled with controlled Petri nets. Controlled Petri nets are an extension of standard Petri nets in which binary control inputs can be applied as external conditions for enabling transitions in the net. A controlled Petri net is a five-tuple: CPN = {P, T, E, C, B}, where P is the finite set of state places, T is the finite set of transitions, $E \subseteq (P \times T) \cup (T \times P)$ is the set of directed arcs connecting state places and transitions, C is the finite set of control places and $B \subseteq (C \times T)$ is the set of directed arcs associating control places with transitions.

The set of state (control) places which are inputs to a transition $t \in T$ is denoted by ${}^{(p)}t({}^{(c)}t)$ and the set of state places which are outputs of a transition $t \in T$ is denoted $t^{(p)}$. Similarly, for a state place $p \in P$, ${}^{(t)}p$ (respectively, $p^{(t)}$) represents the set of all transitions for which $p \in P$ is an output (input), and $C^{(t)}$ represents the set of all transitions for which control place $c \in C$ is an input. A controlled Petri net is strongly connected if there is a direct path between any two transitions in the graph.

A control $u : C \rightarrow \{0,1\}$ assigns a binary token count to each control place. The set of all controls will be represented as U. We notice that assuming a sure net for our models, we afford to assign a binary token count to control places. A transition $t \in T$ is said to be state enabled under a marking m if $m(p) \geq 1$ for all $p \in {}^{(p)}t$. A transition $t \in T$ is said to be enabled under a marking m and a control μ if it is state enabled under m and $\mu(c)=1$ for all $c \in {}^{(c)}t$. A transition set in a controlled Petri net is said to be enabled under a given marking m and control μ if all transitions in the set are enabled. A state transition notation is [2]: $m_0[u(\cdot), \sigma] \Rightarrow m$ in order to indicate that marking m results from the valid firing sequence σ of the initial marking m_0 under the control policy U. R(u, m) and $R(U(\cdot), m)$ denote the set of markings reachable under valid transition firing sequences of any length under the control u and the control policy U, respectively. The immediately reachable set $R_i(U(m), m)$ is the set of markings reachable under the firing of a single transition set. A partial ordering on the set of controls U is defined for two controls $u_i, u_j \in U$, $u_i \geq u_j$ signifies $u_i(c) \geq u_j(c)$ for all $c \in C$, and is said that u_i is more permissive than u_j. Given a marking m, and two controls u_i and u_j, if $u_i \geq u_j$, any transition set, which is enabled under the control u_j is also enabled under the control u_i. The forbidden state control problem [4], [5] is to determine a control policy U for which $R(U(\cdot),m_0) \subseteq M_F$ for all $m_0 \in M_F$, where M_F represents the set of forbidden markings. In [5], the set of forbidden markings is represented in terms of forbidden conditions. A marking $m \in M$ satisfies a forbidden set condition $F \subseteq P$ if $m(p)=1$ for all place conditions $p \in F$, where M is the initial marking set of the controlled Petri net. The method presented in [5] identifies a set of paths for the place conditions. Given a place $p \in P$, a precedence path π for p is defined as a sequence of directed places $(p_1, p_2, \dots p_n)$ so that

1) $p_n = p$;
2) ${}^{(c)}_{(t)}({}^{(t)}p_1) \neq \varnothing$;
3) for each $t = {}^{(t)}p_i$, for $1 < i \leq n$, ${}^{(c)}t = \varnothing$.

The notation $t_\pi = {}^{(t)}p_1$ denotes the controlled transition leading to the precedence path π. The set of all precedence paths for a place condition p is represented by Π_p.

Liveness properties of controlled Petri nets

In an ordinary Petri net, a transition is said to be live if from any reachable marking and for any transition, there is a reachable marking under which the transition is fireable. A controlled Petri net is live under a control policy U $(\cdot)$ and an initial marking m_0 if and only if for every marking $m \in R(U(\cdot), m_o)$ and every $t \in T$, there is a marking $m' \in R(U(\cdot), m)$ and a few controls $u \in U$ so that the transition t is enabled under the pair m' and u. We notice that a control policy can be so restrictive that the resulting controlled system is not live. Therefore, in order to have enough conditions for the controlled Petri nets to be live we define the synchronic distances [7]: For any two transitions t_i and t_j in a controlled Petri net, with an initial marking m_0, the synchronic distance $d(t_i, t_j)$ is:

$$d(t_i, t_j) = \max_{\sigma \in \Sigma} \left| \sigma^*(t_i) - \sigma^*(t_j) \right| \qquad (1)$$

Where Σ is the set of firing sequences starting at markings $m \in R$, and $\sigma^*(t)$ is the number of times the transition t fires in the firing sequence σ.

The following propositions represent results given in [4]-[7] about the properties concerning liveness in controlled Petri nets:

Proposition 1: Given a strongly connected controlled Petri net, an initial marking $m_0 \in M$, and a control law U, there is a finite number k so that if σ is a valid firing sequence of length $l>k$, then $\sigma^*(t) \geq 1$ for all $t \in T$.

This proposition states that if there is an allowable transition firing sequence of length greater than some constant, in a strongly connected controlled Petri net, then all transitions in the graph will have fired at least once in the sequence. Therefore, the length of a firing sequence is calculated as $\sum_{t \in T} \sigma^*(t)$.

Proposition 2: Given a strongly connected controlled Petri net, with a forbidden class condition F and an initial marking $m_0 \in M$, the following conditions hold:

a) ${}^{(p)}(p^{(t)}) = \{p\}$ for every place $p \in F$;

b) $\left({}^{(c)}t_{\pi_1} - {}^{(c)}\left(p^{(t)}\right)\right) \neq \varnothing$ for all $p \in F$ and $\pi_1 \in \Pi_{p_1}$,

where $p_1 \in \bigcup_{F_1 \in F} F$ and $p_1 \neq p$;

c) there are places $p_i, p_j \in F$ so that $p \in F$ and that ${}^{(c)}t_{\pi_i} \not\subset {}^{(c)}t_{\pi_j}$ for any $\pi_i \in \Pi_{\pi_i}$ and $\pi_j \in \Pi_{\pi_j}$;

Then, the controlled Petri net is live under the maximal admissible policy U_F. The first two conditions of proposition 2 state that the output transition from any marked place p in a forbidden set condition can always be enabled. Condition a) states that place p is the only state place input to the transition, and this transition is state enabled when p is marked. Condition b) states that there is a control so that the input transition to any path leading to a place in the forbidden state can be disabled without disabling the output transition for other forbidden places. Condition c) states that there is a control for the transition t_π, where the unmarked path π leads to the place $p \in F$, so that t_π is enabled to fire, while disabling $t_{\pi 1}$ for other path leading to $p_1 \in F$. Therefore, in a strongly connected controlled Petri net, with a forbidden class condition F and an initial marking $m_0 \in M$, if the proposition 2 holds for all F, then there is a fireable transition under some control $\mu \in U_F(m_0)$, and the net is live.

The given propositions allow us to build a controller, which we believe to be novel, so that under a control law $U(m_0)$ and in the presence of the $U_F(m)$ control, it ensures the liveness of the strongly connected controlled Petri net model of the FMS's.

Liveness controller for controlled Petri nets

Quality control and management systems must be analysed using collected data, and then the result is used for controlling the process and for preventing damages. When process conditions change, the process parameters must be adjusted according to process variability. Liveness can be used to investigate and control the process, as shown in Fig.1.

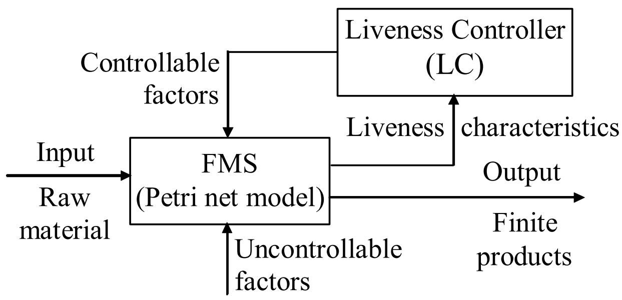

Fig. 1. Liveness controller for a FMS

Our approach for the liveness significance may be different from the classical ones, but we see this as a controller for the modelling process.

The liveness controller (LC) activities, in this paper, are constructed in terms of Petri nets formalisms: these activities can be refined into a Petri net, as shown in Fig. 2.

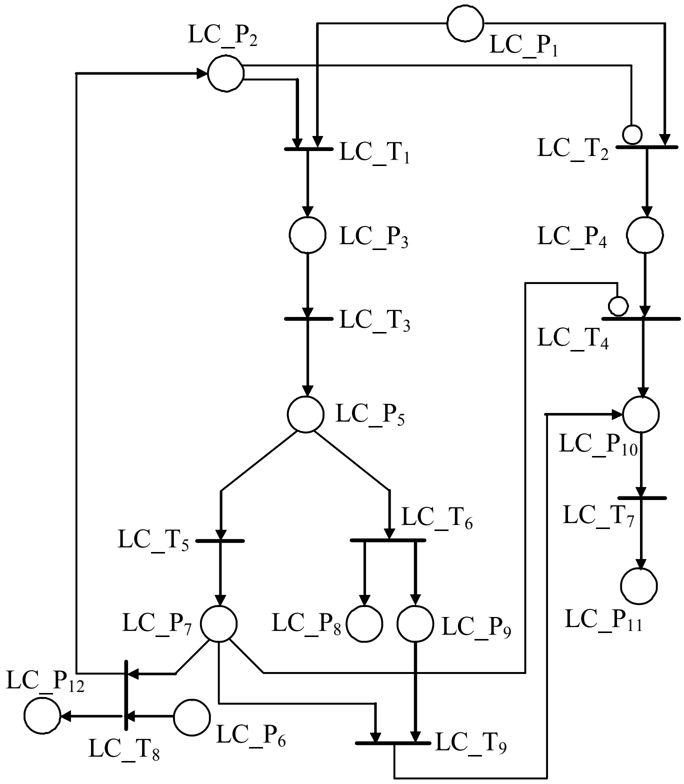

Fig. 2. Petri net model of a LC

The Petri net elements of the LC are described as follows. LC_P_1 represents the data collected from the shop floor. LC_T_1 indicates the initial data set and LC_T_2 indicates the subsequent data set. LC_T_1 and LC_T_2 are controlled and mutually excluded by LC_P_2. The number of tokens in LC_P_2 indicates the required initial number for analysing the capability of the process. For us, the process capability can be evaluated in terms of the liveness capability. LC_P_5 contains the data analysed by the process capability. The marking of LC_P_5 controls LC_T_5 and LC_T_6, which represent the live and the non-live process, respectively. If a process results to be non-live, then LC_P_6 restarts the capability analysis of the process after adjusting the process parameters in concordance with the conditions imposed by proposition 1 and proposition 2, as discussed above.

If the Petri net in Fig.2 is live, then LC_T_4 is not inhibited and the process is considered to be capable, where location LC_P_{11} stores the results of the control analysis. The inhibitor arc connected to LC_T_2 is designed for controlling the number of initial data simulations for analysing the process liveness. The inhibitor arc connected to LC_T_3 indicates when the initial data (measurements) are complete, and the process can be executed to generate the liveness of the system. When LC_P_{10} is marked by firing LC_T_4 it indicates that the process is live under the initial parameters, and when LC_P_{10} is marked by firing LC_T_9 it indicates that the process became live under adjusted parameters. The inhibitor arc connected to LC_T_4 indicates if the process liveness is not acceptable. In this situation LC_P_7 is marked and inhibits the firing of transition LC_T_4. Locations LC_P_8, LC_P_{11} and LC_P_{12} indicate to another (eventually) process the status of the controlled process. The following example of controlling the traffic in a railway system will highlight the above given approach.

An example of traffic coordination in a railway system

We consider an example of coordinating departures of railway vehicles in a railway system [8]. The goal of our Petri net is to model the layout of the transport system. When a vehicle needs to move from the current stop to the next adjacent stop, it needs to receive a "ticket" of movement first to know its destination. Then, the vehicle acquires the control right of the next adjacent stop to make sure that stop is free at the moment. If both of these conditions are satisfied, it can start its travelling to the next adjacent stop (station). In the same time, the control right of the current stop will be released to allow another vehicle to use it as a destination or pass-by stop. The controlled Petri net model of the railway system consists of a number of elementary controlled nets, as shown in Fig.3.

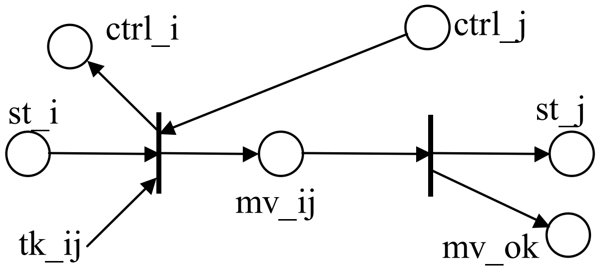

Fig. 3. Elementary controlled Petri net for modelling the railway system

The notations of places in Fig. 3 are explained as follows:

1) st_i, i∈N represents the stop at a workstation. A token in st_i means that a vehicle is currently stationing at stop i;
2) ctrl_i, i∈N represents the control right of stop i. When ctrl_i is marked, it means that stop i is freed now and all vehicles are allowed to move to stop i; otherwise, it means that there is a vehicle at the stop i, therefore no other vehicle can move to that stop;
3) mv_ij, i, j∈N, with an arc connecting stop i and stop j in the net, represents the status of the vehicle movement from stop i to stop j. A token in place mv_ij means that a vehicle is currently moving from stop i to stop j;
4) tk_ij, i, j∈N, with an arc connecting stop i and stop j in the controlled Petri net represents the "ticket" of the path from stop i to stop j. A token in place tk_ij means that a vehicle wants to move from stop i to stop j;
5) mv_ok represents the status of completing the vehicle movement along a path. A token in place mv_ok means that the vehicle has completed the path movement.

For example, we consider five workstations as shown in Fig. 4.
The process flow for the directed graph in Fig. 4 is the following: A convoy of railway trucks leaves the garage line (stop_0) and is sorted in order to load/unload some trucks at line number 1 (stop_1) and then to load/unload the rest of the trucks at the line number 2 (stop_2). The next operation is to form the convoy with the loaded trucks at the manoeuvre lines of the shunt board (stop_3) and then the convoy is sent back to the garage lines, when the trucks correspond for the traffic security laws, or to the line 1, in order to unload the trucks when these do not correspond for the traffic security. This final verification is made at the expedition lines (stop_4) of the shunt board.

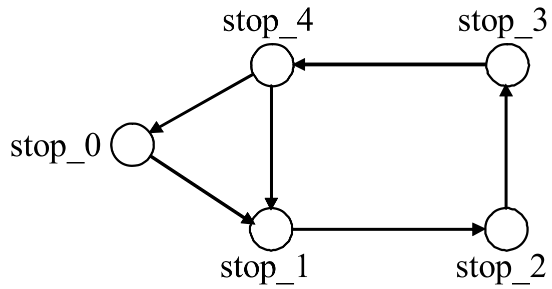

Fig. 4. Layout directed graph for five workstations example

The controlled Petri net for the layout directed graph in Fig. 4 is given in Fig. 5, in which stops st_i, where i = 0,…, 4, represent potential collision regions through which railway vehicles must pass. A collision can occur if two vehicles simultaneously occupy a zone (i.e., there are two tokens in the set of places representing a zone). Because the transition following each forbidden place (each place in a forbidden zone) has only a single input, place conditions a), and b) of proposition 2, given in section 3, are satisfied. Furthermore, since no two controlled transitions share simultaneously the same control places, then condition c) is also satisfied. It results the net in Fig. 5 is live under the initial parameters. In Fig. 2, in the liveness controller, we observe that transition LC_T_{11} is not inhibited.

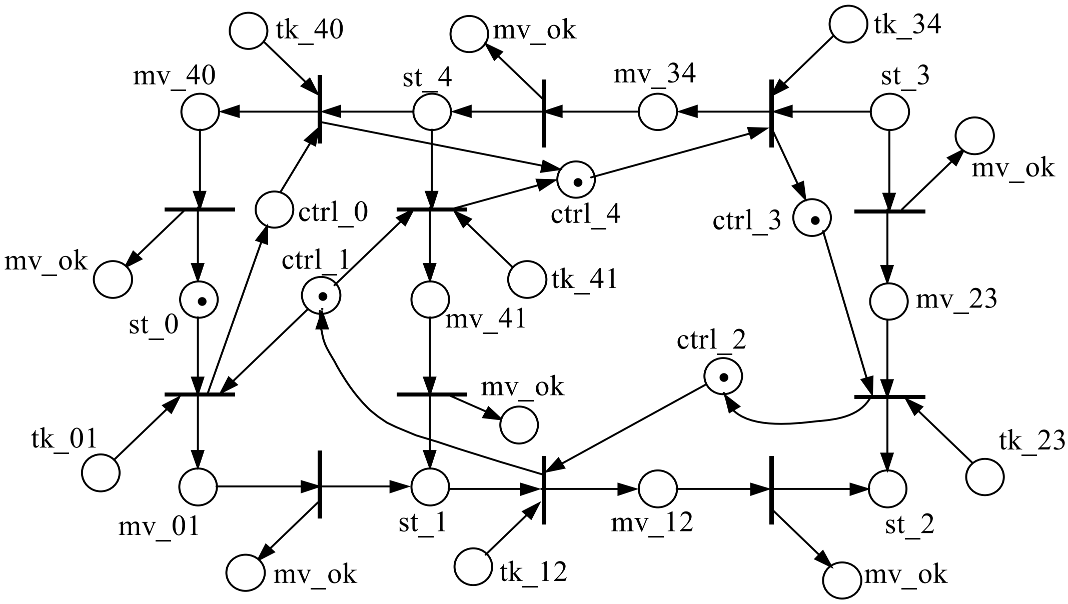

Fig. 5. Controlled Petri net for modelling the railway system with five workstations

Conclusions

In this paper we presented a set of sufficient conditions for a class of controlled Petri nets which will ensure liveness under a control policy for avoiding a set of forbidden states. Liveness is an important property of systems, which will ensure that the system continues to operate while avoiding undesirable states. In order to construct a live system we introduced an algorithm, which can be used in a unified modelling technology. Such research increases the integrability of models with different behaviours. An example for traffic coordination in a railway system was chosen for exemplifying the given approach for modelling FMS's. Further researches will increase this method, by applying differential Petri nets in the structure of the liveness controller.

References

[1] Jeng, M.D., DiCesare, F., (1995) *Synthesis using resource control nets for modelling shared-resource systems*, IEEE Trans. On Rob. And Autom., vol. 11, no. 3, pp. 317-327.

[2] Cieslak, J. et al., (1988) *Supervisory control of discrete-event processes with partial observations*, IEEE Trans. On Autom. Contr., vol. 33, no. 3, pp. 459-463.

[3] Murata, T., (1989) *Petri nets: properties, analysis and applications*, Proc. IEEE, vol. 77, pp. 541-580.

[4] Holloway, L.E., Krogh, B.H., (1990) *Synthesis of feedback control logic for a class of controlled Petri nets*, IEEE Trans. On Autom. Contr., vol. 35, no. 5, pp. 514-523.

[5] Jeng, M.D., (1997) *Petri nets for modelling automated manufacturing systems with error recovery*, IEEE Trans. on Rob. And Autom., vol. 13, no. 5, pp. 752-760.

[6] Zhou, M.C., Venkatesh, K., (1999) *Modell., Simul. Contr. of FMS – A Petri net approach*, River Edge, N.J: World Scientific.

[7] Holloway, L.E., Krogh, B.H., (1997) *On closed-loop liveness of discrete event systems under maximally permissive control*, IEEE Trans. on Autom. Contr., vol. 37, no. 5, pp. 692-697.

[8] Ciufudean, C., (2001) *Petri nets in railway traffic systems*, Advances in Electrical and Computer Engineering, University of Suceava, vol. 1(8), no. 1(15), pp. 15-20.