Performance evaluation of a controlled flow-shop system with a timed petri net model

Loic Plassart, Philippe Le Parc, Frank Singhoff, Lionel Marcé

PERFORMANCE EVALUATION
OF A CONTROLLED FLOW-SHOP SYSTEM
WITH A TIMED PETRI NET MODEL

Loïc Plassart, Philippe Le Parc, Frank Singhoff, Lionel Marcé

Université de Bretagne Occidentale
Laboratoire d'Informatique des Systèmes Complexes (LISyC)
20, avenue Le Gorgeu – 29285 Brest Cedex, France
{loic.plassart—philippe.le-parc—frank.singhoff—lionel.marce}@univ-brest.fr

Abstract: This paper presents an original performance analysis applied to a flow-shop system driven by a set of local command units and a central controller. The performance evaluation is done with a timed coloured Petri net model. Simulation results show needs for bounding the controller response time in order to meet production targets.

1   INTRODUCTION

During the last few decades the performance increasing of the production systems was mainly based on an improvement of the device acting directly on the product flow. However, the behavior of a production system is also conditioned by the characteristics of the equipment which ensures its control (Grieco et al., 2001) .

The control system design of an automated production system poses a large amount of hardware and software problems. They are often complex and bring into play a more and more consequent amount of data. This evolution usually implies taking into account the specific requirements applied to the reactivity of various equipments.

This article presents an industrial case study applied to a flow-shop system whose control is jointly ensured by a set of local command units (programmable logic controllers) and a central controller. Each local command unit is located at the production machine level of the flow-shop and drives an operative part.

Message exchanges between the local command units and the controller are carried out several times during the operating cycle of the production machine. They are supported by a fieldbus.

In literature, many manufacturing control architectures are identified (Dilts et al., 1991). They are often declined in three main types from centralized over hierarchical to heterarchical control. Our control architecture is based on a typical hierachical structure in which an upper level device coordinates the activities of a group of lower level devices in a master-slave manner (Jones et al., 1989). The control flow is typically top-down and the feedback flow is bottom-up.

In the form of hierarchical architecture, the exchanges are always triggered by the upper level device. But in the present case study, they are initiated by the local command units. The exchanges are operated according to a request transmission and a response reception. The stimulus is then bottom-up and more than one exchange can be running at the same time.

Message routing, waiting and processing cumulated delays constitue a response time that is necessary to bound in order to ensure the control system does not influence production rates significantly.

This study aims to analyze the controller reactivity to requests sent by local command units and to evaluate its impact on the production system performances.

The article begins by a description of the production system and the control architecture in section 2. The modelling of the flow-shop system and its control devices are detailed in section 3. It is based on the formalism of timed coloured Petri nets. The model performance analysis relates to the message reception buffer occupancy and controller activity rates. Several configurations are evaluated. The simulation results summary, presented in section 4, allows to measure model performances and to analyse the correlation degree between real production systems and their
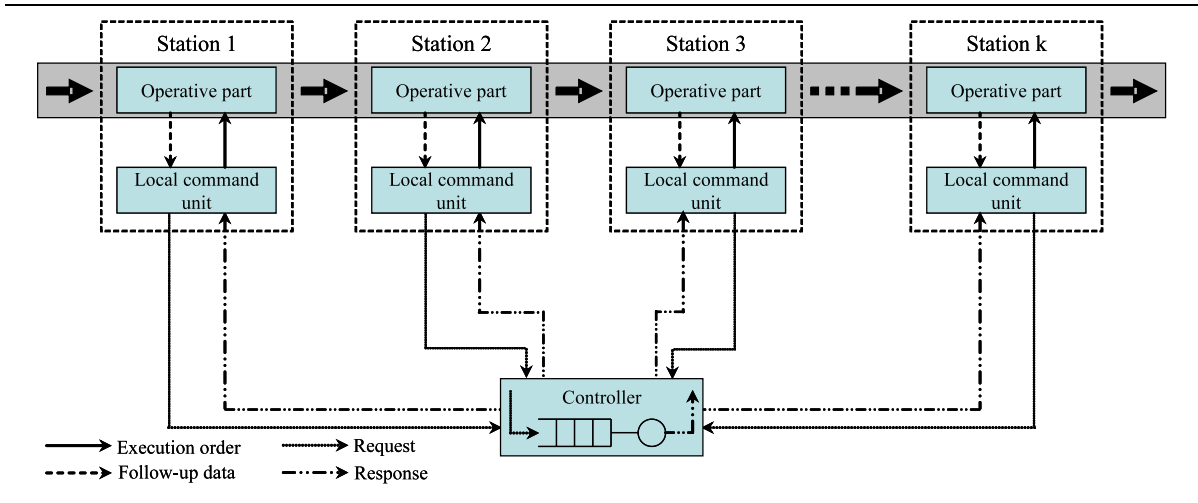
Figure 1: System architecture

model. Lastly, section 5 presents the conclusions of our work and a brief description of our future research in this area.

2 SYSTEM DESCRIPTION

The considered production system is an automated assembly process with several machines called stations organized in line. This stucture corresponds to a flow-shop system in which there is one station at each stage. The stations work in an independent way from each other and execute individually their operating cycle. A station cannot retain and operate more than one part at a given moment. When a station is available (no assembly in progress) and a part is present at its entry, it starts its operating cycle.

The control of the assembly line is ensured by a central controller which coordinates the various stations. Thus, it has to be considered as a shared resource of the flow-shop system [Fig. 1].

2.1 Operating cycle of the stations

During its operating cycle, each station performs many common operations. For most of them, each occurence is of constant duration. Only the assembly processing is of specific duration. The mean duration of common operations [Tab. 1] is obtained by measurements done on real production systems.

An operating cycle also includes two waiting phases of a variable duration due to exchanges with the controller. A waiting phase corresponds to the controller response time when it is requested by a local command unit. On existing assembly lines, this delay is variable and depends of controller load. The

Operation	Duration$_{(ms)}$
Identification	550
Status request	60
Waiting	Variable
Status analysis	60
Assembly processing	Specific
Data reporting	60
Waiting	Variable
Part relaxation	60

Table 1: Operating cycle of a station

mean waiting time is close to 600 ms.

Parts are transfered from a station to the next one by a conveyor belt. The conveying time between two stations is always about 3 000 ms.

2.2 Coordination function

The local command units request and inform the controller in a regular way to condition and coordonate their actions. These exchanges proceed at the production rate. The controller has an information centralization function it enriches by data transmitted by the local command units and that it can give back.

During an exchange, the local command unit sends a request to the controller. The message supporting the request is placed in a communication buffer. The controller manages the message selection by a FIFO policy and executes the processing in adequacy with the message parameters. The latency time in the communication buffer is obviously variable. At the end of the processing, the controller establishes a response message that it transmits to the local command unit.

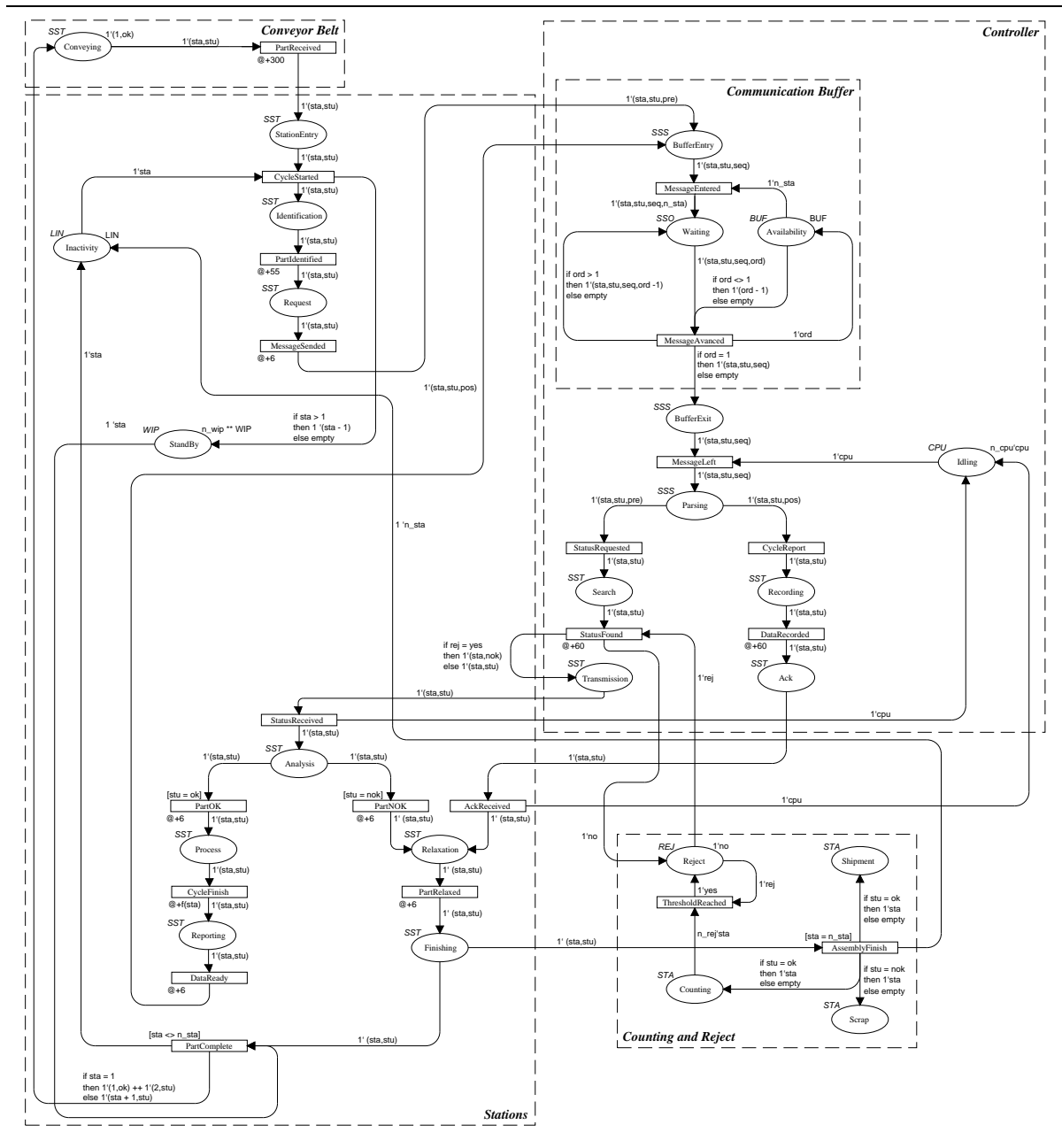A request sent by a local command unit causes a

Figure 2: Petri net of the studied system

freezing situation of its cycle. The activity restart is triggered by the response arrival coming from the controller.

3 SYSTEM MODELLING

The Petri nets interest for the design and the analysis of production systems has often been empha-sized (Barad, 1998; Desrochers and Al-Jaar, 1995; DiCesare et al., 1993). Our model was designed with that formalism. The timed and coloured extensions have also been used.

The behaviour of the data processing systems used in computer-integrated manufacturing and robotics is often based on timed sequences making it possible to define the duration of activities. The timed extensions of Petri nets are highly suitable models for the for-mal study of these mechanisms (Juanole et al., 2000).

Although timed coloured Petri nets are usually used to investigate the logical correctness of systems, they can also be used to evaluate the performances of them.

The Petri net colouring allows to reduce the size of a model to get a more compact representation of the systems including dictinct components having similar behaviours. This is the case here with the station operating cycle.

The model was designed using the software Design/CPN (Christensen et al., 1997). Our choice has been guided by the integration of performance facilities in this computer tool (Linstrøm and Wells, 1998). Design/CPN[1] allows to collect data during the model execution and create reports containing statistical values.

3.1 Modelled behaviour

This work focuses on the study of interactions within the control system of an automated assembly line. Thus, the model describes the architecture supporting the exchanges between the local command units and the controller. It is primarily founded on the independent activities of the various stations and their access to a shared resource which is the controller.

The model [Fig. 2] implements three main elements that are the station set, the communication buffer and the controller. A counting function of produced parts is also implemented.

Station operations are integrated into the model in the form of places. The described sequence takes into account the operations of routing (place *Conveying*), part identification (place *Identification*), status request sent to the controller (place *Request*), response analysis emitted by the controller (place *Analysis*), part assembly processing (place *Process*), production data reporting (place *Reporting*) and part relaxation (place *Relaxation*).

Transitions correspond to events that make the token circulation symbolizing the system evolution. Temporization of some transitions (notation @+*delay*) makes it possible to keep the presence of tokens in the places located upstream during the specified time. It allows to specify the execution time of operations.

Colouring is primarily used to distinguish various stations as well as various positions of messages in the communication buffer.

For the stations, colours are applied according to their number and their order on the assembly line. Part moving from a station to another one is modelled by incrementing the token colour number, except in case of a processing by the last station (end of assembly). Station availability is managed by a specific place (place *Inactivity*) guaranteeing exclusivity

---

[1]Design/CPN has been replaced by CPN/Tools

---

of its execution to the processing of only one part. The presence of a token of color x indicates that the station x is inactive. The place *Inactivity* is initialized according to the number of declared stations.

The communication buffer is modelled by a place (*Waiting*) in which the token represents the situation of a station having emitted a request. Colors are affected according to the order in the buffer. Each token move models a message advance and its color number is decreased.

The controller disponibility is done by a single place (place *Idling*) and a token in this place indicates the controller availability. A message arrival in the communication buffer implies the token leaves the place *Idling*. The controller stops its activity when it completes the message processing and when the communication buffer is empty.

3.2 Evaluated criteria

Performance of the studied production systems is mainly evaluated by the analysis of the communication buffer occupancy and by the controller load.

Communication buffer contents and the message origin make it possible to know the number and the list of stations waiting for a response from the controller.

The message waiting time and the controller response time can be deduced from the simulation data. The analysis of these criteria allows to evaluate production system performances and to obtain indications leading to a good knowledge in the way in which the exchanges between the local command units and the controller are held.

Product inter-departure times are also measured. They are defined as the time between two part complete assemblies and they can be easily compared with the results of modelled real systems. Product inter-departure times are a simple and effective performance indicator of the production systems.

3.3 Model settings

The size of the communication buffer and the fieldbus speed are two modelling assumptions integrated into the model.

The size of the communication buffer is defined according to the number of stations declared in the model. It makes it possible to have a sufficient capacity for all evaluated configurations in order to avoid a saturation in accordance with the real systems.

The message propagation on the fieldbus is very fast compared with the modelled system. Only the latency delays in the communication buffer and message processing are significant. So, within the framework of our study, we consider that the message transmission is instantaneous.

The temporization of the Petri net places is mainly defined with mean measurements done on several existing assembly lines. So, the model is primarily intended for real production system evaluation.

However, the simulation work may also implement configurations that are not effective in the workshops. The assembly processing time cannot be given by measurements. Then, these durations were generated using a pseudo-random generator. The generation takes into account of the maximal and the minimal bounds measured on the existing assembly lines where the mechanical processing times are between 1 100 and 8 500 ms.

Consequently, the model settings are mainly based on the number of declared stations and the message processing time by the controller. The evaluated configurations differ by the number of stations. The various cases go from 2 to 30 stations.

The time sampling rate is generally selected quite higher than the main time-constant of the controlled process (Ogata, 1987). The time quanta appointed for the modelled activities temporization is 10 ms. This period is defined according to the characteristics of the local command units whose cycle time is around 60 ms.

Finally, the work-in-progress between two stations is limited to three parts.

4 Simulation results

The simulation results are declined in two parts. The first phase of our work related to the study of existing production systems (from 2 to 15 stations). With these results, we can check the model by comparing the simulation results with measurements done on real assembly lines. The second phase is intended to evaluate the extension possibilities of the station number and the potential profits brought by the reduction of the message processing time.

The model execution time corresponds to a 75 mn manufacturing period sampled every 10 ms (450 000 clock ticks). It covers the transient and stationary regimes. The switch from the transient to the stationary regime is triggered by the assembly end of the first product. However, for homogeneity reason, we study the stationary regime on the 360 000 last clock ticks matching with 60 mn of manufacturing.

4.1 Real systems

Simulation results [Tab. 2] for existing assembly lines show that the controller load lies between 28 % and 100 %. The saturation point is reached with the 13-station configuration.

The model execution gives product inter-departure mean times τ_s close to those τ_e noted on existing assembly lines.

Conf.	Contr. load$_{(\%)}$	Buff.	$\tau_{s(s)}$	$\tau_{e(s)}$
2	28.4	0.001	8.52	8.65
3	42.5	0.069	8.51	
4	56.5	0.104	8.54	8.69
5	67.4	0.264	8.93	
7	89.8	0.685	9.48	
10	99.7	2.759	12.6	
13	100	5.199	16.85	16.28
15	100	6.724	19.6	

Table 2: Results for existing systems

For configurations up to 7 stations, the mean buffer occupancy remains quite lower than one message.

4.2 Prospect

The prospective simulation results show a permanent controller activity [Tab. 3].

Conf.	Contr. load$_{(\%)}$	Buff.	$\tau_{s(s)}$
20	100	10.87	27.36
25	100	13.141	36.68
30	100	16.651	45.64

Table 3: Prospective results

The report also indicates that the mean buffer occupancy is between 11 and 17 messages.

4.3 Production rate bound

Our simulation results show that product inter-departure mean time τ_s grows with the number of stations (Little law). By comparing this indicator with the cumulated time of message processing needed for assembling one part, we can give a production rate bound τ_c for the modelled assembly lines. In this formula, k is the number of stations and m_i is the number of messages proceed for the station i:

$$\tau_c = \sum_{i=1}^{k} \sum_{j=1}^{m_i} t_{ij}$$

The production rate τ_s obtained by the simulation approaches but never meets this bound [Fig. 3].

For a classical flow-shop system without controller, the productivity rate τ_m is typically fixed by the slowest station (Cohen et al., 1983):
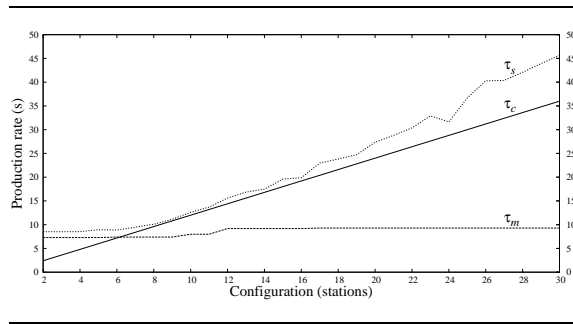
$$\tau_m = \max\{t_1, t_2, ..., t_k\}$$

Figure 3: Production rate evolution

By analyzing the curves, we can give a resulting production rate bound τ_r:

$$\tau_r = \max\{t_1, t_2, ..., t_k, \sum_{i=1}^{k} \sum_{j=1}^{m_i} t_{ij}\}$$

It corresponds to the worst bound between the one established by the controller implementation τ_c and the one defined for flow-shop systems τ_m.

5 CONCLUSIONS AND FUTURE WORKS

The study presented in this article applies to the control system of automated assembly lines. The aim is to estimate the impact of the controller implementation on the production rates. The described analysis enables to evaluate the behaviour and the performances of the considered production systems.

The modelling and simulation phases show that the efficiency of the assembly lines is strongly linked to the controller implementation. The production rate bound fixed by the controller is defined by the message processing time. In the case of the cumulated time of message processing is longer than the operating cycle time of the slowest machine, the controller must be seen as the bottleneck station. The need for limitation of the message processing time to reach the production targets is then shown.

Our future works are based on two distinct angles.

Firstly, we have to analyze the system with real network considerations and take into account hybrid flow-shop systems in which parallel stations are implemented.

Afterwards, we will study specific message scheduling policies. Message processing in a FIFO manner don't probably correspond to the best way. We will also evaluate improvements got by message dispatching on several processors.

REFERENCES

Barad, M. (1998). Timed Petri nets as a verification tool. *30th Winter Simulation Conference*, pages 547–554.

Christensen, S., Jørgensen, J., and Kristensen, L. (1997). Design/CPN – A computer tool for coloured Petri nets. *3rd International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, pages 209–223.

Cohen, G., Dubois, G., Quadrat, J., and Viot, M. (1983). Analyse du comportement périodique de systèmes de production par la théorie des dioïdes. *Rapport de Recherche n 191 – Institut National de Recherche en Informatique et Automatique (France)*.

Desrochers, A. and Al-Jaar, R. (1995). *Applications of Petri nets in manufacturing systems: modelling, control and performance analysis*. IEEE Press.

DiCesare, F., Harhalakis, G., Proth, J., Silva, M., and Vernadat, F. (1993). *Practise of Petri nets in manufacturing*. Chapman & Hall.

Dilts, D., Boyd, N., and Whorms, H. (1991). The evolution of control architectures for automated manufacturing systems. *Journal of Manufacturing Systems*, 10:79–93.

Grieco, A., Semeraro, Q., and Tolio, T. (2001). A review of different approaches to the FMS loading problem. *The International Journal of Flexible Manufacturing Systems*, 13:361–364.

Jones, A., Barkmeyer, E., and Davis, W. (1989). Issues in the design and implementation of a system architecture for computer integrated manufacturing. *Journal of Computer Integrated Manufacturing*, 2:65–76.

Juanole, G., Abdellatif, S., and Gallon, L. (2000). Nouveaux concepts sur les transitions du modèle RdPTS : Propriétés dynamiques et attributs temporels dynamiques. *1ère Conférence Internationale Francophone d'Automatique (CIFA'2000)*, pages 976–981.

Linstrøm, B. and Wells, L. (1998). Simulation based performance analysis in Design/CPN. *Workshop on Practical Use of Coloured Petri nets and Design (CPN'98)*, pages 117–130.

Ogata, K. (1987). *Discrete-time control systems*. Prentice-Hall International Editions.