INSTABILITIES AND DETERMINISTIC CHAOS IN JUST IN TIME PRODUCTION SYSTEMS
Comparison between neural networks simulation and continuous simulation

Daniel Thiel
E.N.I.T.I.A.A., *Laboratory of Economics and Management*
Domaine de la Géraudière 44072 Nantes cedex 03 (France) e-mail : thiel@enitiaa-nantes.fr

Introduction

This paper tackles on instabilities and chaos which can occur in production systems whose organisation is based on the just-in-time philosophy. This could be a *Kanban* system which controls the production flows. In this case, cards or other manual and visual devices, accompanied by parts containers, signalise transfer and/or manufacturing operations thus acting as production orders. For instance, a worker, from an assembly line, needing more components, attaches a transportation *Kanban* to an empty parts container, that is moved to the previous work centre (according to routing procedure) where is replenished (with new manufactured parts) and moved back to the assembly line.

At first, our research focused on one elementary cell of an integrated production line which has many dysfunctions, like stoppages of work or rejected parts. In such circumstances, different regulation mechanisms are acting in case of fluctuations in the work-in-process inventory. We built a system dynamics model whose objective consisted in studying the inventory stability according to different productivity values. The simulation results led us observe chaotic behaviour for some values of this parameter (many other authors have explored the deterministic chaos phenomenon in system dynamics, see for instance : Mosekilde, Mosekilde, Aracil and Allen, 1988). These results were also observed by simulation of an interactive automata network, hence the originality of this paper.

The System Dynamics Model

The causal diagram

The following influence diagram (see figure 1) shows the dynamics relations inside a basic production cell like a workstation (cf. the model of a production unit proposed by Massotte from IBM in 1993). The principle of the production policy consists in continuous refurnishing of the Work-In-Process (WIP) inventory. The production lot size (production rate) is equal to the level of WIP multiply by the productivity rate. The output rate (production completion rate) is equal to the WIP level which is weighted by the ratio of the real level of WIP to the desired level of WIP. Three loops are present in this model as shown in the figure 1.

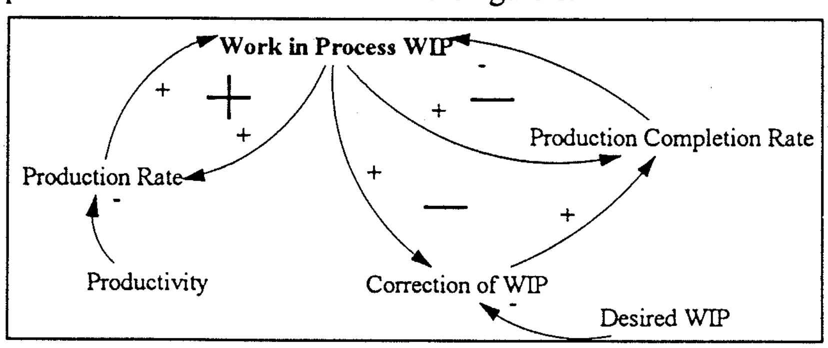

Figure 1 - *Influence diagram of the Working Centre Control*

The flow diagram

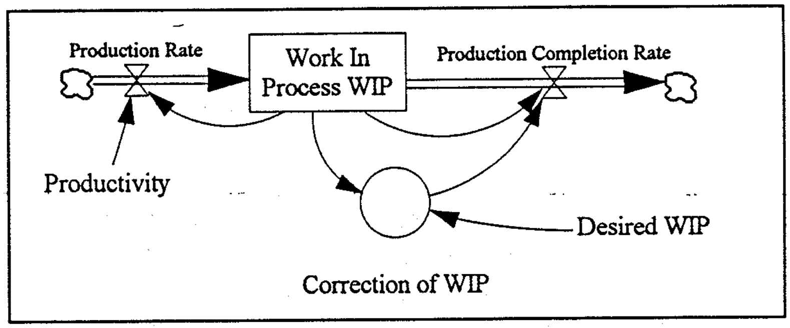

Figure 2 - *Flow Diagram*

The corresponding level and flows equations are :
*Work_In_Process_WIP(t) = Work_In_Process_WIP(t - dt) + (Production_Rate - Production_Completion_Rate) * dt*
INIT Work_In_Process_WIP = 1
Production_Rate = Work_In_Process_WIP/Productivity/DT
*Production_Completion_Rate = Work_In_Process_WIP*Correction_of_WIP/DT*
Correction_of_WIP = Work_In_Process_WIP/Desired_WIP
Desired_WIP = 2
Productivity = variable contained in 0 to 1.

Remark : *The production and output rates are both divided by* dt *because the input and output flows are discrete and correspond on a set of many products which are laying on a container.*

Simulation results

By using exclusively the Euler's method, we simulated the previous model and we studied the sensitivity of the work-in-process level to the productivity rate (see results in figure 3). By very low productivity for instance 33% which could correspond on an important rejected part rate of the previous production line or on a low capacity, the work-in-process level can not reach a stable equilibrium. This evolution is chaotic. In case of 50% productivity rate, we again observed an unstable behaviour of WIP but at this time, it is a periodic evolution and the attractor is a second order limit cycle. The last value of the productivity as showing in the figure 3, is a more current value (66% and over). In this situation, the work-in-process level is stable and strives towards a fixed point attractor (see also the next figure 4).

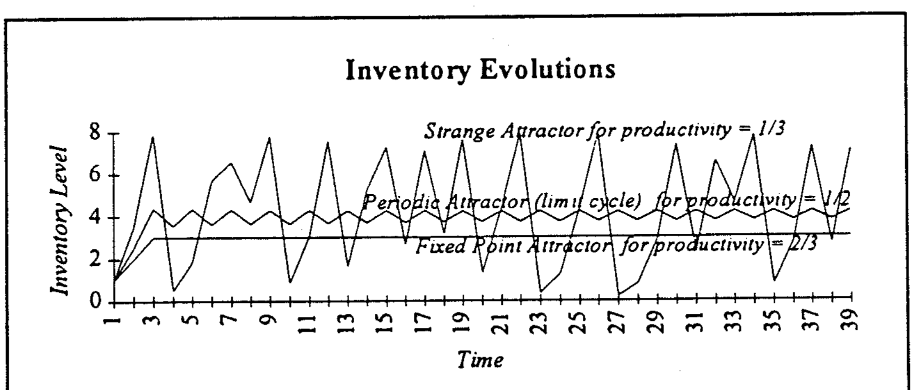

Figure 3 - *Inventory Sensitivity on the productivity*

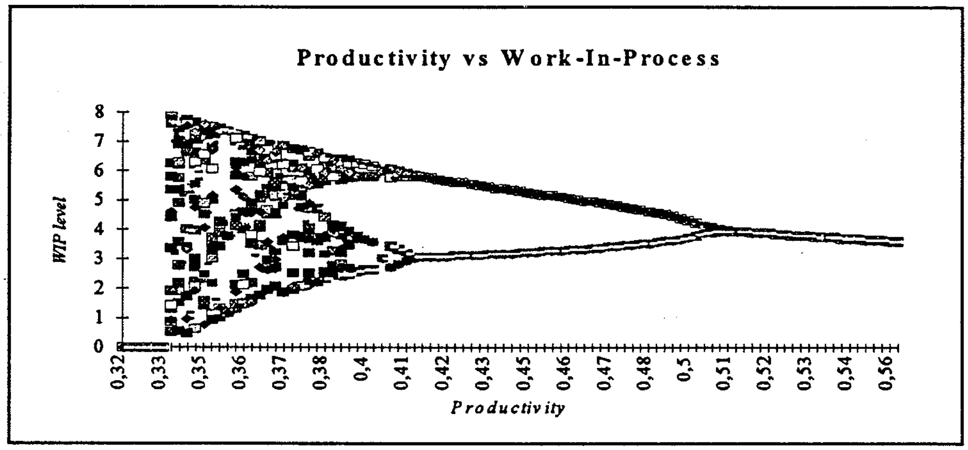

Figure 4 - *Bifurcations and chaos in the model*

The Equivalent Automata Network Model

The following step consists in comparing the previous dynamic simulation results to the simulation of an equivalent automata network. The principle of the transformation of our previous model structure by levels and flows into an automata network is showing in the figure 5. This type of neural network is a discrete dynamical model (cf. Fogelman Soulie et al., 1987). It is an iterative model where space and time are discrete; where elements involved in the system, called cells, are finite automata, and where the interactions between these elements are moreover purely local. As is now well known, this type of systems go back to Von Neuman, in the early 1950s (Von Neuman, 1966).

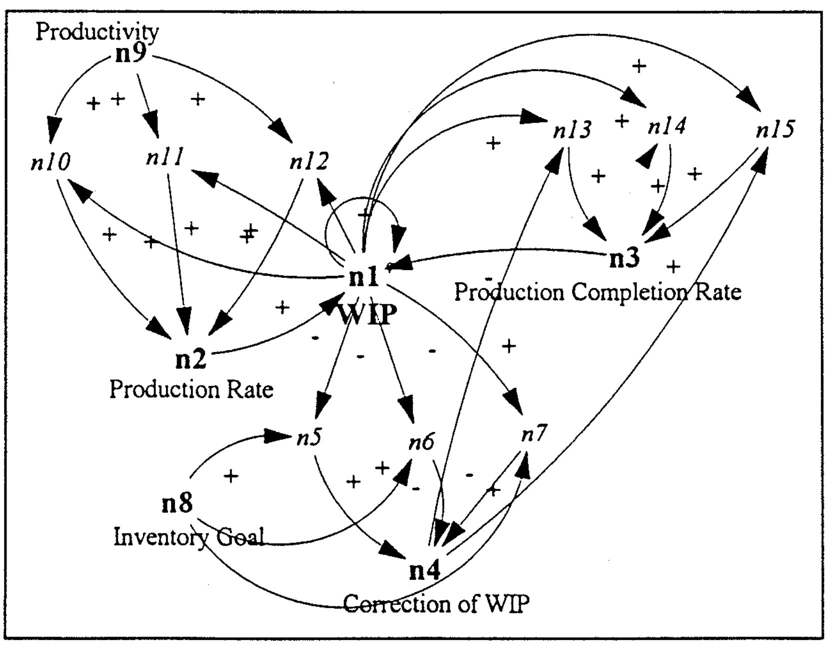

Figure 5 - *The equivalent Neural Network*

To transform the previous model into a neural network, the different continuous variables were changing into discrete variables as shown below. It was also necessary to transform the non-linear relations of the model by using hidden layers of units. This type of simulation has permitted to bring in light the same attractors as previously : fixed points, limit cycles and chaotic attractors (see also the work of Reggiani et al., 1995).

Neurones of the model :

n_1 = Work-In-Process WIP (values 0, 1, 2)
n_2 = Production_Rate (values 0, 1, 2, 3, 4, 5, 6)
n_3 =Production_Completion_Rate (values 0, 1, 2)
n_4 = Correction of WIP (values 0, 1, 2)
n_8 = Desired WIP (values 0, 1, 2)
n_9 = Productivity (values 0, 1, 2, 3)
n_5 to n_7 and n_{10} to n_{15} = hidden layers of neurones (values 0, 1, 2, 3, 4).

Threshold Equations :

If $(n_1 + n_2 - n_3) > 0$ then $n_1 = (n_1 + n_2 - n_3)$ else $n_1 = 0$
If $(0.8n_{10} + 0.96n_{11} + n_{12} - 10) > 0$ then $n_2 = (0.8n_{10} + 0.96n_{11} + n_{12} - 10)$ else $n_2 = 0$
If $(0.8n_{10} + 0.96n_{11} + n_{12} - 10) > 0$ then $n_3 = (n_{13} + 0.3n_{14} + n15 - 6.1)$ else $n_3 = 0$
If $(n5 - 2n_6 - n_7 + 0.99) > 0$ then $n_4 = (n5 - 2n_6 - n_7 + 0.99)$ else $n_4 = 0$
If $(n_1 + n_8 - 1) > 0$ then $n_5 = (n_1 + n_8 - 1)$ else $n_5 = 0$
If $(-n_1 + n_8 - 1) > 0$ then $n_6 = (-n_1 + n_8 - 1)$ else $n_6 = 0$
If $(n_1 + n_8 + 1) > 0$ then $n_7 = (n_1 + n_8 + 1)$ else $n_7 = 0$
If $(n_1 + 0.7n_9 + 1) > 0$ then $n_{10} = (n_1 + 0.7n_9 + 1)$ else $n_{10} = 0$
If $(1.2n_1 + n_9 + 1) > 0$ then $n_{11} = (1.2n_1 + n_9 + 1)$ else $n_{11} = 0$
If $(1.2n_1 + 0.7n_9 + 1) > 0$ then $n_{12} = (1.2n_1 + 0.7n_9 + 1) > 0$ else $n_{12} = 0$
If $(2n1 + n4) > 0$ then $n13 = (2n1 + n4)$ else $n13 = 0$
If $(2n1 + n4) > 0$ then $n14 = (2n1 + n4)$ else $n14 = 0$
If $(2n1 + n4) > 0$ then $n15 = (2n1 + n4)$ else $n15 = 0$

Conclusion

A further analysis of these results has been realised with two objectives :

- one more methodological in term of complementarity between theses two approaches (see Thiel 1995)

- the second was more qualitative and treated the dynamic proprieties of the system.

New research proposals have been elaborated in the objective to improve the understanding of more complex production systems which generate non-linear behaviour and deterministic chaos.

References

Mc Culloch, Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biophysics*. 5, 115-133.

Fogelman Soulié F., Y. Robert, M. Tchuente, 1987. *Automata networks in computer science* (Princeton University Press, New Jersey).

Mosekilde E., J. Aracil, P.M. Allen. 1988. Instabilities and chaos in non-linear dynamic systems. *System Dynamics Review*, 4, 1-2, 14-55.

Massotte, P. 1993. Behavioural Analysis of complex system. *Proceeding of the 1er congrès biennal de l'AFCET Systémique et Cognition*, Versailles, France, 71-82.

Neuman Von, J. 1966. *Theory of self Reproducing Automata*. A.W. Burks (ed.), (University of Illinois Press, Urbana).

Reggiani A., P. Nijkamp. 1995. Competition and complexity in spatially connected networks. *System Dynamics Review*. 11, 1-2, 51-66.

Thiel D. 1995. Detecting attractors in production systems using system dynamics and neural networks. *1995 INRIA/IEEE Symposium on Emerging Technology and Factory Automation*. Paris, 677-684.