A neural network for the reduction of a Product Driven System emulation model

Philippe Thomas, André Thomas, Marie-Christine Suhner

A neural network for the reduction of a Product Driven System emulation model

Philippe THOMAS, André THOMAS, Marie-Christine SUHNER

Centre de Recherche en Automatique de Nancy (CRAN-UMR 7039), Nancy-University, CNRS

ENSTIB 27 rue du Merle Blanc, B.P. 1041, 88051 Epinal cedex 9, France
Tel. +33(0) 3 29 29 61 73. Fax +33(0) 3 83 68 44 37
philippe.thomas@cran.uhp-nancy.fr

In new Intelligent Manufacturing Systems, Product Driven Systems (PDS) architecture require emulation tool (Thomas *et al*. 2008) to be developed. Discrete events simulation is often used to build such emulation tool, nevertheless this remains complex because of large scale problems. The goal of this paper is to propose a way to design a simulation model by reducing its complexity. According to theory of constraints, we build reduced models composed exclusively of bottlenecks and a neural network. In Particular, a multilayer perceptron is used. The structure of the network is determined by using a pruning procedure. This work highlights the impact of discrete data on the computational results. An application to a sawmill internal supply chain is suggested to validate the proposed approach.

Key words: multilayer perceptron, reduced model, simulation, neural network, supply chain

1. Introduction

In classical centralized manufacturing systems, planning and control processes simulation is essential for evaluation of planning and scheduling scenarios to make better and faster decisions. Indeed, simulation allows to describe dynamically the behaviors of machines, where WIP (work in process), and queues are easily modeled. So, simulation models would be useful in order to perform a “Predictive scheduling” (Lopez and Roubellat 2001) or a rescheduling in case or disturbance.

On the other hand, in Product Driven processes (distributed way to control physical flow in a Supply Chain), dedicated architectures are implemented. These architectures consist of a control system and an emulation system which are very useful for Product Driven Systems (PDS) design. Moreover, it is used in order to validate such systems by one hand and making decisions by scenario evaluation by the other hand. So, the PDS architectures require

emulation models which should be sufficiently precise to represent the most closely the real system by maintaining a reasonable size in order to decrease computation running time.

Discrete event simulation is also often used to build such emulation system, but emulation model design, which is not a trivial task, relies on reusability, modularity and genericity concepts (Thomas *et al*. 2008). Moreover, for emulation models, the number of "objects" and the number of events can be very large. Consequently, the problem relies on the time to build the model. Moreover, the simulation running time could be too much high which makes the models not operational in practice. Thus, it could be necessary to reduce the model size (Thierry *et al*. 2008).

The real time systems performing manufacturing follow up (production reporting) transmit information very quickly to the management system (Khouja 1998). However, it is difficult to use this large amount of information to make decisions (Prisker and Snyder 1994, Roder 1994). At these levels of planning and control, to estimate how the whole physical system behaves, the "management of critical resources" (bottlenecks) is often efficient (Vollmann *et al*. 1992). Goldratt and Cox, in "The Goal" (1992) put through the Theory of Constraints (TOC), which proposes to manage the whole supply chain by bottlenecks control. Dynamic discrete events simulation of material flow permits this management (Thomas and Charpentier 2005). In fact, build an emulation model is a complex task which could a lot of time. Moreover, emulation models which aims to represent real industrial cases are often complex because of the problems large scale (Page *et al*. 1999). Thus, numerous authors have expressed interest in using the simplest (reduced/aggregated) models of simulation (Brooks and Tobias 2000, Chwif *et al*. 2006, Ward 1989). In Thomas and Charpentier (2001), the authors have shown that an interesting method would be to reduce the model according to the TOC.

In addition, neural networks have proved their abilities to extract performing models from experimental data (Thomas *et al*. 1999). So, the use of neural networks appears recently

as an interesting approach within the framework of the supply chain (Thomas and Thomas 2008). In this context, we associate a queuing model with a neural network, respectively, to model both the bottlenecks and, works centers.

However, neural networks are generally used in order to perform a mapping between continuous spaces, and, in the considered cases, continuous variables (as length, speed…) are mixed with discrete ones (as category, color…).

Thus, the main goal of this paper is to investigate the impact of these discrete data on the learning process and on the quality of neural model used in order to reduce simulation models according to the TOC, i.e. to maximize the bottleneck utilization rate. This is studied with one industrial example which is a sawmill flow shop case. In the next part, a brief bibliography overview is presented. The third part describes the proposed approach of reduction model and the multilayer perceptron. The fourth part presents the construction of an emulation model applied to the sawmill internal supply chain case. Part five focuses on one step of this approach which is the neural network design. The validation of the approach and the impact of the discrete data on the results are highlighted in the last part before to conclude.

2. Bibliography overview on model reduction

The two main difficulties encountered during the design step in a supply chain simulation model are related to the size of the system and the complexity of the control system. The problem could be seen at the supply chain level which is composed of a group of enterprises and composed in turn of a group of factories, or at the shop floor level which is composed of a group of work centers, etc. Moreover, modeling the behavior of the leading policies of each enterprise and the relationships between them is needed (Thierry *et al*. 2008). This fact implies that the duration of one simulation may become unacceptably long to be usable.

Therefore, it may be useful to reduce the size of the model. Different ways can be used to perform the model reduction (Zeigler 1976):

- abstraction, which allows the complexity of the model to be reduced and preserves the validity of the results (Frantz 1995),
- aggregation, which is a form of abstraction where a group of data or variables with common characteristics can be replaced by aggregated data or variables (Mercé 1987),
- reduction of the number of events, where a part of discrete event system is replaced by a variable or a formula (Zeigler 1976).

In addition, Innis and Rexstad (1983) have listed 15 simplification techniques for general modeling. Their approach is composed of four steps: hypotheses (identify the important parts of the system), formulation (specify the model), coding (build the model), and experiments. Based on these works, different approaches have been proposed.

Brooks and Tobias (2000) suggest a 'simplification of models' approach for cases where the indicators to be followed are the average throughput rates. They suggest an eight-stage procedure. The reduced model can be very simple and then an analytical solution becomes feasible and the dynamic simulation redundant. Their work is interesting, but is valid in cases where the required results are averaged and where the aim is to measure throughput. It is not interesting to follow the various events taking place in the work center (WC).

Leachman (1986) has proposed a model for use in the semiconductor industry, which uses cycle time as an indicator. This model has been improved by Hung and Leachman (1999). They propose a technique for model reduction to be applied in large wafer fabrication facilities. They use 'total cycle time' and 'equipment utilization' as decision-making indicators to do away with the WC. In their case, these WCs have a low utilization rate and a fixed service level (they use the standard deviation of batch waiting time as a decision-making criterion).

Tseng *et al*. (1999) compare the regression techniques applied to an 'aggregate model' (macro) by using the 'flow time' indicator. They suggest reducing the model by mixing the

‘macro’ and ‘micro’ approaches, so as to minimize errors in complex models. Here again, for the ‘macro’ view, they deal only with the estimation of flow time as a whole. For the ‘micro’ approach, they construct an individual regression model for each stage of the operation to estimate its individual flow time. The cumulative order of flow time estimates is then the sum of the individual flow times. They, then, try to mix the macro and micro approaches.

These different approaches simplify the model by using a macroscopic view of the system and by optimizing a macroscopic indicator (total cycle time, flow time…)

Li *et al*. (2009) propose a reduction model approach based on the aggregation of machines on the production line. They build a complete model of the production line and, if the last two machines correspond to a serial line, they aggregate them. The same is performed with the first two machines if they correspond to a serial line. These aggregation steps may be performed recursively and they are denoted backward and forward aggregation, respectively. If the two machines to be aggregated follow a Bernoulli model or an exponential model, an analytical investigation allows the production rate of the new aggregated machine to be determined. If not, a simulation phase must be performed to determine an empirical formula for the production rate.

Some works (Doumeing 1989, Hwang *et al.* 1999) use Petri nets as tool in order to simplify network structures by using macro-places which represent complex activities associated with function groups.

To simplify models, some works have studied the use of a continuous flow model based on gradient estimation for stochastic systems in order to approximate discrete manufacturing environments (Ho 1987, Suri and Fu 1994). Other authors use metamodels (linear regression, splines, Kriging, etc.) to perform a simulation model (Kleijnen and Sargent 2000). Neural networks can be viewed as a type of metamodel (Barton 1994, Pierreval 1996, Kleijnen and Sargent 2000). In addition, neural networks have proved their abilities to extract

models from experimental data (Thomas *et al*. 1999). Therefore, the use of neural networks has emerged recently as an interesting approach within the framework of the supply chain (Shervais *et al*. 2003, Chiu and Lin 2004).

3. The proposed model reduction process

a. The algorithm

The proposed approach is based on the association of discrete event models and continuous metamodels (neural network) in order to design a simulation model. Our previously described objective was to maximize the bottleneck utilization rate and, at the same time, simplify simulation model construction for modellers. The reduction algorithm proposed is an extension of those presented by Thomas and Charpentier (2005). The main goal of this algorithm is to reduce the number of simulation blocks. For its understanding, three concepts must be defined:

- 'conjunctural bottleneck' (current bottleneck) is a WC that is saturated for the MPS in the predictive scheduling in question. This means that it uses all of its available capacity,
- 'structural bottleneck', we mean a WC that has often been or is in such a condition. Effectively, production managers know only where the regular overloaded WC(s) is (are),
- 'synchronization work centers' are resources used jointly with bottlenecks for at least one MO and are used for the planning of different MOs that do not use a bottleneck. To minimize the number of these 'synchronization work centers', only those which are often associated with bottlenecks in MOs must be considered.

The main algorithm steps are recalled and explained below:

1) Identify the work center (WC) which is the structural bottleneck. This one has been the main capacity constraint for several years (according to the experience of production manager).
2) Identify the conjunctural bottleneck for the bundle of manufacturing orders (MOs) of the Master Production Schedule (MPS) under consideration (see explanation later in the text).
3) Among the WCs not listed in 1 and 2, identify the one (synchronization WC) that satisfies the following two conditions:
    - presents at least in one of the MOs using a bottleneck, and

- widely used considering the whole body of MOs.

4) If all MOs have been considered, go to 5; if not, go to 3.
5) Use neural networks to model the intervals between the WCs that have been found during the preceding steps (figure 1).

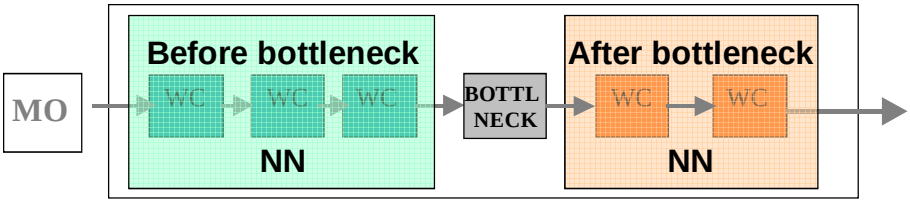

Figure 1. Reduction model algorithm

Hence, the WCs remaining in the model are either conjunctural or structural bottlenecks, or are WCs that are vital to the synchronization of the MOs. All other WCs are incorporated in 'aggregated blocks' upstream or downstream of the bottlenecks.

The main benefits of this algorithm are:

- modellers can focus on the description of the bottlenecks,
- noncrucial parts of the system are modelled with a learning approach (automatization of this modelling step),
- the resulting model is less complex than a complete one, and
- simulation time is shorter than with a complete model.

This paper focuses on step 5 of the reduction algorithm which uses a neural model. Here, the bottlenecks are considered as known.

b. The multilayer perceptron (MLP)

The works of Cybenko (1989) and Funahashi (1989) have proved that a multilayer neural network with only one hidden layer using a sigmoidal activation function and an output layer using a linear activation function can approximate all nonlinear functions with the desired accuracy. This result explains the great interest of this type of neural network, which is called 'multilayer perceptron.' In this work, it was assumed that a part of the modelled production system could be approximated with a nonlinear function obtained with a MLP. The objectives

of this nonlinear function are to model the material flow behaviour (in our case, processing time).

The structure of the multilayer perceptron is recalled here. Its structure is shown in figure 2. The neurons of the first (or input) layer distribute just the n_0 inputs $\{x_1^0, \cdots, x_{n_0}^0\}$ of the MLP to the neurons of the next (hidden) layer. A special input neuron (depicted by a square in figure 2) represents a constant input equal to 1, and it is used to represent the biases or thresholds of the hidden layer.

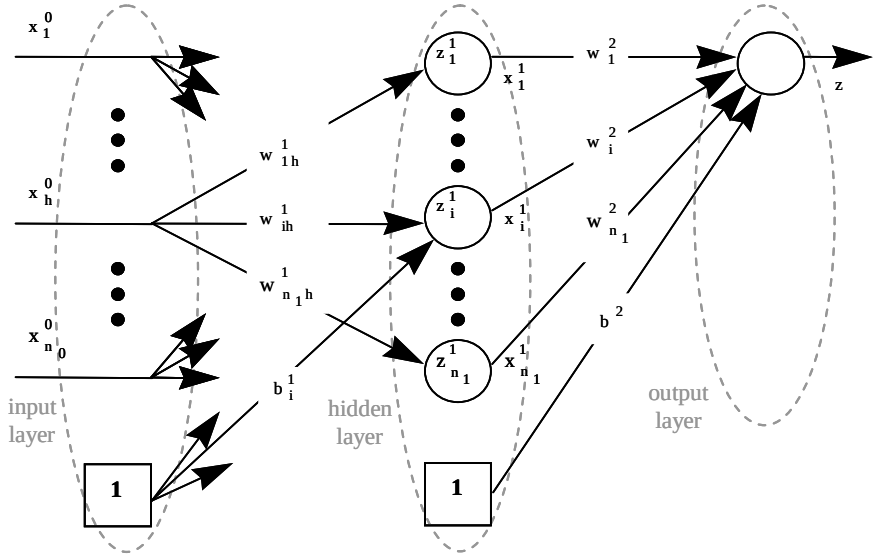

Figure 2. Structure of the multilayer perceptron

The i^{th} neuron ($i = 1, \ldots n_1$) in the hidden layer receives the n_0 inputs $\{x_1^0, \cdots, x_{n_0}^0\}$ from the input layer with the associated weights $\{w_{i1}^1, \cdots, w_{in_0}^1\}$. This neuron first computes the weighted sum of the n_0 inputs:

$$z_i^1 = \sum_{h=1}^{n_0} w_{ih}^1 . x_h^0 + b_i^1, \qquad (1)$$

where b_i^1 is the bias or threshold term of the i^{th} hidden neuron. The output of this neuron is given by a so-called 'activation function' of the sum in (1):

$$x_i^1 = g\left(z_i^1\right), \quad (2)$$

where g(.) is chosen as the hyperbolic tangent:

$$g(x) = \frac{2}{1+e^{-2x}} - 1 = \frac{1-e^{-2x}}{1+e^{-2x}}. \quad (3)$$

Lastly, the outputs of the hidden neurons $\left\{x_1^1, \cdots, x_{n_1}^1\right\}$ are distributed with associated weights $\left\{w_1^2, \cdots, w_{n_1}^2\right\}$ to the unique neuron of the last (or output) layer. As for the input layer, a particular hidden neuron (depicted by a square in figure 2) represents a constant input equal to 1, which is used to represent the bias or threshold of the output layer.

The neuron of the last layer simply performs the following sum, with its activation function being chosen as linear:

$$z = \sum_{i=1}^{n_1} w_i^2 . x_i^1 + b, \quad (4)$$

where w_i^2 are the weights connecting the outputs of the hidden neurons with the output neuron and b is the threshold of the output neuron.

The number of hidden neurons must be determined. For this, the learning starts from an over parameterized structure. A weight elimination method is used to remove spurious parameters, and the pruning algorithm used here is the one proposed by Setiono and Leow. (2000).

In our case, the input neurons are processing system variables (number of parts, inventories...). As previously said output neuron is throughput time.

The learning of the MLP is performed in three steps:

1. initialization of the weights and biases of an oversized structure by using a modification of the Nguyen–Widrow algorithm (Thomas and Bloch 1997),
2. learning of the parameters by using the Levenberg–Marquard algorithm with a robust criterion (Thomas and Bloch 1996), and
3. weights elimination by using the Neural Network Pruning for Function Approximation (N2PFA) algorithm (Setiono and Leow 2000).

Kleijnen and Sargent (2000) have proposed a metamodeling process that can be subdivided into 10 steps:

- determine the goal of the metamodel,
- identify the inputs and their characteristics,
- specify the domain of applicability,
- identify the output variable and its characteristics,
- specify the accuracy required of the metamodel,
- specify the validity of metamodel measures and their required values,
- specify the metamodel and review this specification,
- specify a design,
- fit the metamodel, and
- determine the validity of the metamodel.
  In this work, these different steps were used to design the neural network.

4. Design of emulation models

For validation, we used the proposed approach to build a simulation model of a sawmill. In this actual case, managers needed a tool to help them in their weekly MPS decision-making process. In this process, their decision variables are number of logs, product demand..., their objectives are throughput time, MPS respect (backorders)... The industrial example considered here is limited to the shop floor level. However, this approach can be deployed to all levels of the supply chain.

a. Overview of the sawmill

At the time of the study, the sawmill had a capacity of 270,000 m^3/year, a turnover of €52 million and 300 employees.

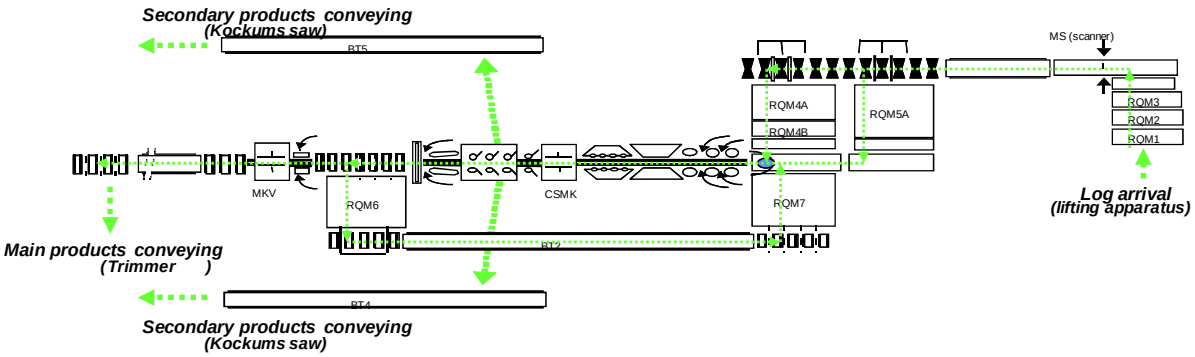

Figure 3. First part of the process: the Canter line

The sawmill objective is to transform logs into main and secondary products according to a cutting plan. The physical industrial production system is composed of sequential work centers (kockums saw, trimmer, sorter…) and queues or conveyors (named respectively RQM4, RQM5, RQM7…). It is subdivided into three main parts. The first one is the canter line presented figure 3. In this subsystem, the log enters the system in RQM1 then it is steered to RQM4 or 5 according to its characteristics. After that, it passes to the cutting machine (Canter). It then enters the edger. After this phase, the log is transformed into main and secondary products. The final operation is the cross cutting which consists in cutting up products to length.

Two important steps occur during this process. The first one is the choice of the conveyors RQM4 or RQM5 in order to store the arrival log. In function of this choice, the time spending by the log to wait the Canter saw may be very different. The second one is the type of product considered. When the cutting plan is considered, two types of products appear: main and secondary ones.

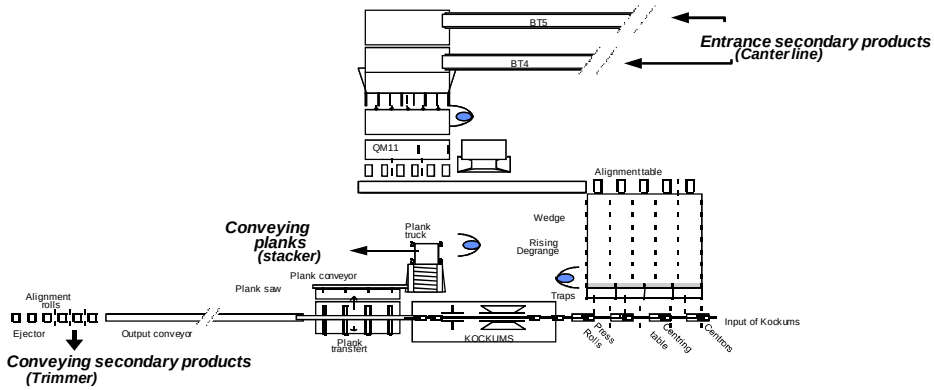

Figure 4. Second part of the process: the Kockums line

Figure 4 shows the second part of the process, where the main machine is the Kockums saw. Only secondary products are driven on this part. The secondary products are taken in the line by the BT4 and BT5 conveyors. They are cut by the QM11 saw, after which they reach the Kockums saw, which optimizes the plank according to the products needed. The alignment table is used as the input inventory of the Kockums saw. The secondary products are finally sent to the third part of the process by the exit conveyor.

The third part of the process is the trimmer line, which is presented in figure 5. This line performs the final operation of cross cutting. This operation consists in cutting up products to length. The input of the line is from collectors 1 and 2, which collect the secondary and main products from Kockums and Canter lines respectively. Saw 1 is used to perform default bleeding and Saw 2 cuts up products to length. A previous work (Thomas and Charpentier, 2005) has shown that this machine, the trimmer saw, is the bottleneck of the entire process.

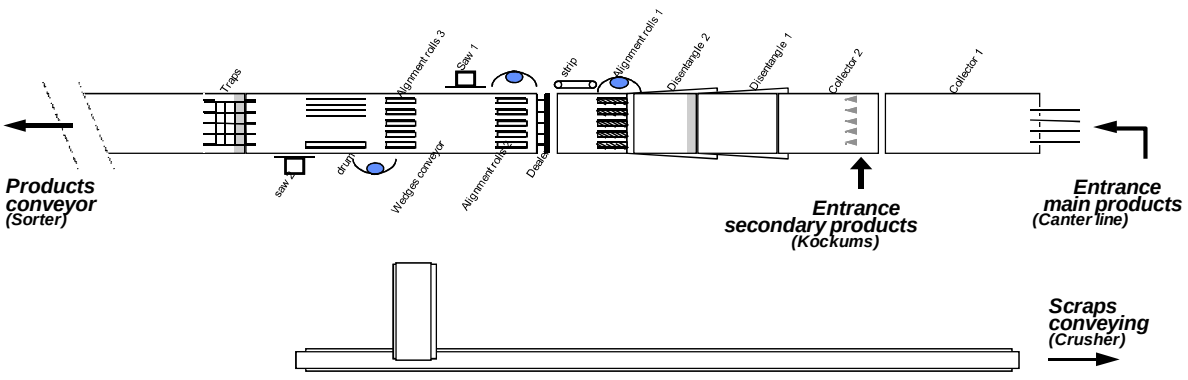

Figure 5. Third part of the process: the Trimmer line

However, when the physical industrial system is considered, three types of products have to be considered. In fact the Cutting machine Canter works into three steps. First, one saw (CSMK) cuts two faces of the considered log and produces two secondary products. These two products are driven to kockums saw in order to be finished. Next the log is rotated of 90° and stored into conveyor RQM7. After that, the log is driven once again to the Canter machine. The saw (CSMK) cuts the two other faces of the log, and produces the two other secondary products which are driven to kockums saw. At this time, a parallelepiped is obtained which is divided into three main products by another saw (MKV). The main products are finally driven to the trimmer.

b. Application of the reduction model approach

In order to produce the sawmill emulation model, described in the preceding part, the procedure proposed in part 3 is applied. The model is designed with the Arena® software and the inclusion of neural network is performed by using a module VBA.

The first step of the procedure is to identify the structural bottleneck. Preceding studies of the sawmill have shown that the structural bottleneck is the trimmer (Thomas and Charpentier 2005). The second and the third steps are respectively to determine the conjunctural bottleneck and the synchronization WC. In the considered case, no conjunctural bottleneck or synchronization WC is present. This fact allows us to focus on the step five which is the core of this paper.

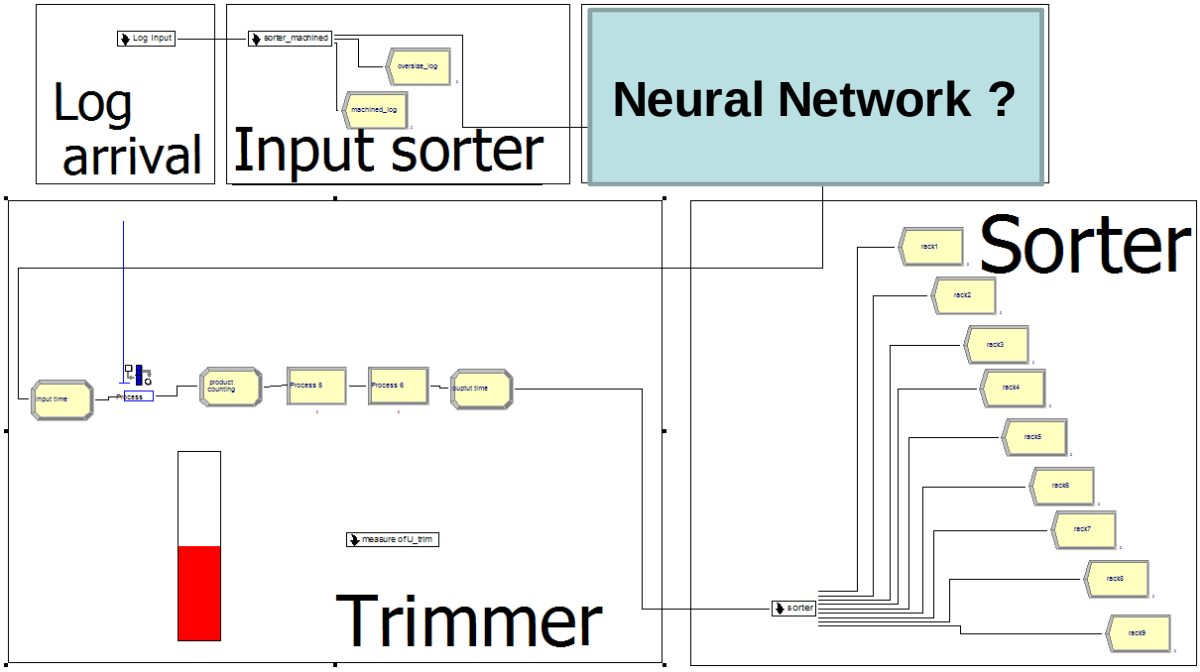

Figure 6. The reduced model

The step five specifies that all the WC which are not bottleneck (structural or conjectural) or synchronization ones must be modeled by using a neural model, consequently and within this framework, the functioning of inventories RQM4, RQM5 and RQM7, the canter and the kockums lines must be modeled by a neural network. The discrete part of the model describes the functioning of the bottleneck (the trimmer) and the log arrival. The emulation model can be described by figure 6. The structure of the neural network (input and output layers) is constrained by the information that must be transmitted to the bottleneck (output) and by the information given by the log arrival part (input). So, the first step to design the neural network is to construct the database to use for the learning.

5. Design of neural network

In order to construct a reduced emulation model, the neural network design is the main task to perform. For this, a complete data set must be collected.

a. The data set

Neural model is a black box obtained with a supervised learning of a non linear relation between input and output data sets. For this, we need to collect the available input data of the process and to determine the desired output (Thomas and Thomas 2008).

First, each log gives information which is collected by a scanner in input of the canter line. This information is relating to the product dimension, as length (Lg) and three values for timber diameter (diaPB ; diaGB ; diaMOY). These variables are used to control the log to RQM4 or RQM5 queues which is additional information (RQM). In addition of this dimensional information, we have to characterize the process variables at the time of the log arrival. Particularly, the input stock of the trimmer (Q_trim), the utilization rate of the trimmer (U_trim) and the number of logs present in the different conveyors RQM4, RQM5 and RQM7 (Q_rqm4; Q_rqm5; Q_rqm7) must be taken. Moreover, the sum of these number is also used (Q_rqm = Q_rqm4+Q_rqm5+Q_rqm7). The last type of information is related to the cutting plan of the logs. In fact, each log will be cut into *n* main or secondary products. In our application, the cutting plan divides the log into 7 products:

- 2 secondary products resulting from the first step of cutting process on saw CSMK of the canter line,
- 2 secondary products resulting from the second step cutting process on saw CSMK of the canter line after staying in the RQM7 queue,
- 3 main products resulting from the third step of cutting process on saw MKV of the canter line.

These two saws (CSMK and MKV) belong to the canter line. These 7 products can be classified into three categories according to the location (CSMK or MKV) and the time during the cutting process (first or second cutting). This information is given by the variable (T_piece) which can take as values type1, type2 and type3. The last information is the thickness (in mm) of the product which is also the reference. In our case, we are taking into account only two references: main products 75; secondary products 25 (ref). However, preceding works (Thomas and Thomas 2008) have shown that this data has no impact on the

result and so it will no be taken into account. Consequently, the neural networks input variables are: Lg; diaGB; diaMoy; diaPB; T_piece; Q_trim; U_trim; Q_rqm; Q_rqm4; Q_rqm5; Q_rqm7; RQM. In our application 12775 products are simulated. Among these 12 inputs data, two different categories exist:

- Continuous one (quantitative) [Lg; diaGB; diaMoy; diaPB; Q_trim; U_trim; Q_rqm; Q_rqm4; Q_rqm5; Q_rqm7]. These data are continuous ones and so are well adapted to be used by learning procedure.
- Discrete one (qualitative) [T_piece; RQM]. These data are qualitative. So the study of their impact on the learning process is the core of this paper.

Our objective is to estimate the delay (ΔT) corresponding to the duration of the throughput time for the 12775 products. ΔT is measured between the process input time and the trimmer queue input time. In practice ΔT is the output of the neural network:

$$\Delta T = \sum_{i=1}^{n_i} w_i^2 . g\left(\sum_{h=1}^{12} w_{ih}^1 . x_h^0 + b_i^1 \right) + b^2 \qquad (5)$$

b. The structuring of the data set

Now, all the data which characterize the process are collected. However, it can be noticed that two different categories have been determined in this dataset, continuous ones and discrete ones. Neural networks are generally used in order to perform a mapping between continuous spaces. So, the difficulty, here, is to determine how the discrete data can be used.

In order to determine this, two different approaches may be proposed.

The first and simplest one is to consider all the discrete data like continuous ones, and to present them to the input of an unique neural network. In the present case, the structure of the emulation model is presented figure 7 where the considered neural network uses 12 inputs.

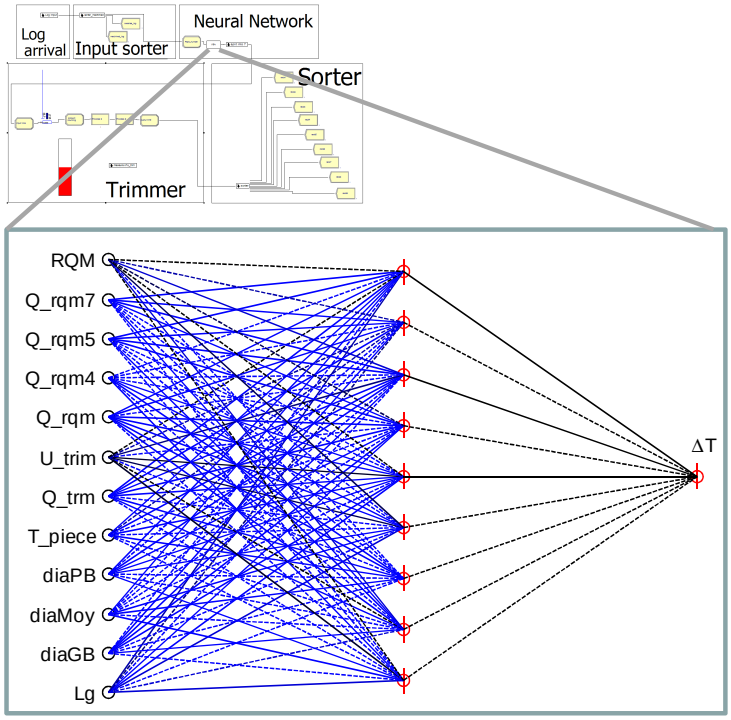

Figure 7. The emulation model – first approach

As said previously, preceding works have shown that the data RQM has a great influence on the behavior of the system. It is very different if RQM is 4 or if RQM is 5. So, an approach for dealing with this fact is to make two different models in order to model it in these two cases and to switch from one to another with the value of RQM. This approach can be related to the multiple-model approach (Delmotte *et al*. 1996).

With this approach, the emulation model includes two different neural networks which are used in function of the value of the data RQM.

So, two neural models have to be learned by using respectively the RQM=4 data and the RQM=5 data uniquely. These two neural networks have 11 inputs: Lg; diaGB; diaMoy; diaPB; T_piece; Q_trim; U_trim; Q_rqm; Q_rqm4; Q_rqm5; Q_rqm7. The structure of this emulation model is presented figure 8.

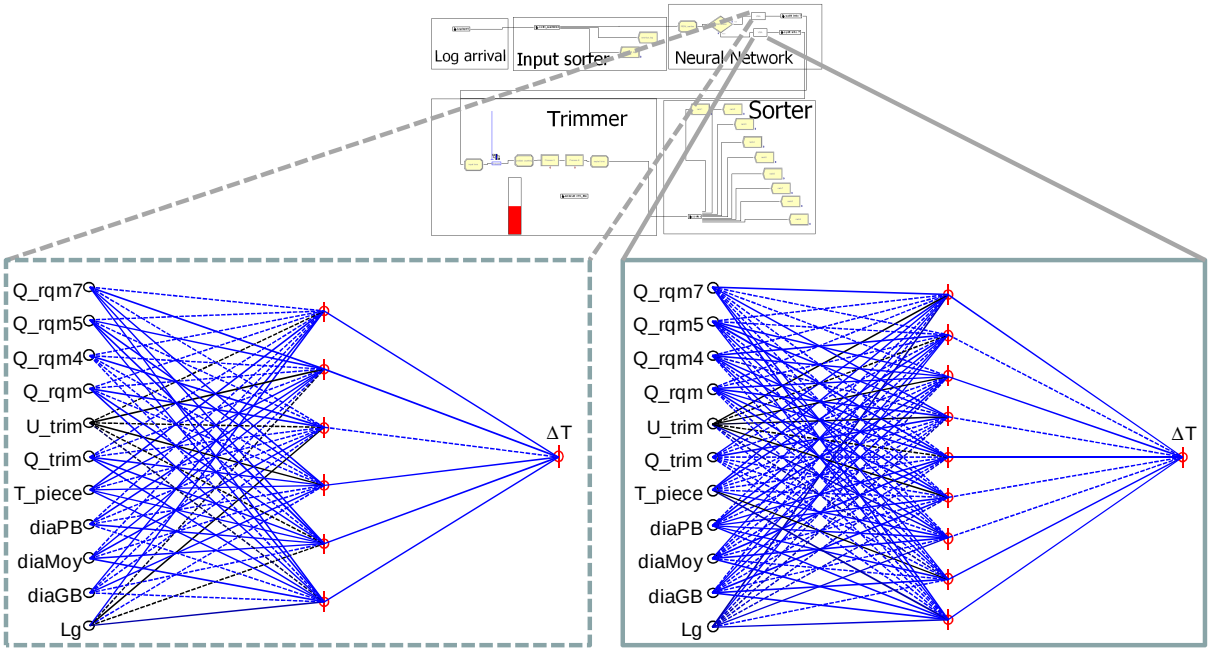

Figure 8. The emulation model – second approach

c. The learning

For the two approaches, the learning of the network is supervised. So, it is necessary to divide the database into two datasets, learning and validation ones. Only the number of hidden neurons is always unknown and should be determined. In order to determine it, the learning starts from an over parameterized structure and a weight elimination method is used to remove spurious parameters.

The learning approach corresponds to a local search of a minimum. So, in function of the initial weights, the results may be different. In order to evaluate the dispersion of the results, 30 different sets of initials weights are used.

6. Validation of the emulation models

In preceding works (Thomas and Charpentier 2005), a complete model of the sawmill has been constructed and validated with the real process. Here, this complete model is also used in order to compare the results obtained with the two reduced emulation models with it.

a. First approach

Table 1. Mean and standard deviation of the residuals – first approach

	Learning residual		Validation residual	
	Mean (s)	StD	Mean (s)	StD
Mean	78.61	586.09	74.33	582.06
StD	43.94	146.50	41.61	145.44
Min	17.11	408.45	12.35	413.93
Max	213.08	1168.80	206.75	1170.93

The 30 learnings on the different weight sets have been performed with the initial over parameterized structure composed by the 12 inputs and the 10 hidden neurons (5) which corresponds to 141 parameters. In the table 1, the mean and the standard deviation of the residuals obtained on the learning and the validation data sets are presented. The residuals represent the errors performed by the model for the estimation of throughput times ΔT comparatively to the desired ones.

It can be recalled that the objective of the learning is to obtain a white noise (normal distribution of mean null) as residual. These results show that the residuals obtained are always bad. In particular, the mean of the obtained residual may vary, in function of the initial weights from 17.11s to 213.08s on the learning data set. For the validation data set, the results are very similar, with a mean of residual varying from 12.35s to 206.75s. It can be noticed that the mean of the residuals is lower than 30s in only 10% of the cases in learning and 16.67% of the cases in validation. Concerning the standard deviation values, they are large and varying from 408.45 to 1168.8 for the learning data set and from 413.93 to 1170.93 for the validation data set. These two facts show that the learning is not efficient.

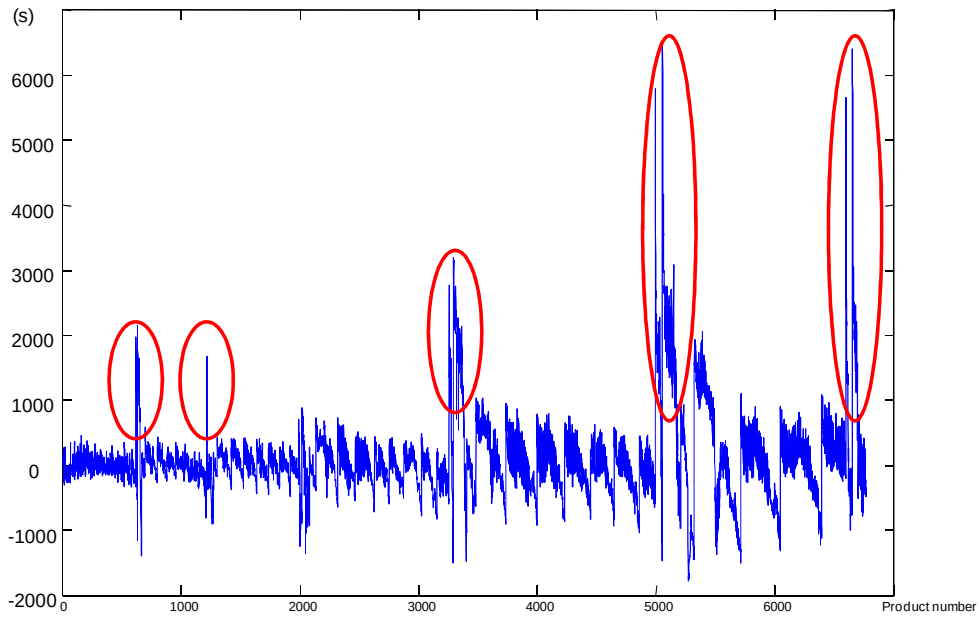

Figure 9. Residual obtained on the learning data set

Figure 9 shows an example of residual characteristic of those obtained on the validation and learning data sets for the 30 different initial weights. Except those surrounded by circle, they could be acceptable for validation. But those highlighted by the circles may be due to different causes:

i. the number of hidden neurons is not sufficient,
ii. the neural network does not succeed to learn some dynamics due to not taken into account root causes,
iii. some explicative variables (example, marginal products, exceptional breakdown…) could be not present in the input data.

In order to evaluate if the residuals surrounded by circle are due to cause i), other tests series have been implemented where the number of hidden neurons varied from up to 35 to less than 10. These tests have shown that 10 hidden neurons are sufficient. Moreover, the pruning algorithm prunes some of these ten hidden neurons into 56% of the cases.

For evaluating cause ii), so, in order to determine if some dynamics, due to not taken into account root causes, present in the data are not learned, the correlation between the different inputs and the residuals can be performed on the learning data set (table 2). The table 2 presents the mean, standard deviation, minimal and maximal values of the absolute value of

the correlation coefficients obtained between the 30 residuals and the 12 inputs on the learning data set.

Table 2. Coefficients correlation between residual and inputs – first approach

	Lg	diaGB	diaMoy	diaPB	T_piece	Q_trim	U_trim	Q_rqm	Q_rqm4	Q_rqm5	Q_rqm7	RQM
Mean	0.0354	0.0118	0.0393	0.1619	0.0350	0.0484	0.0298	0.0707	0.0628	0.0697	0.0525	0.2875
StD	0.0245	0.0096	0.0238	0.0692	0.0261	0.0324	0.0211	0.0467	0.0531	0.0456	0.0355	0.1310
Min	0.0002	0.0013	0.0014	0.064	0.0001	0.0002	0	0	0.0025	0	0	0.1124
Max	0.0882	0.0342	0.0843	0.3411	0.0959	0.1172	0.0813	0.1774	0.2280	0.1831	0.1314	0.6706

It can be noticed that Lg, diaGB, diaMoy, T_piece, U_trim present a correlation coefficient with residuals which is never significant (always smaller than 0.0959). U_trim, Q_rqm, Q_rqm5, Q_rqm7 present a minimal value of correlation to 0 because the pruning algorithm, in some case has pruned these inputs. Only two inputs have always a significant coefficient correlation with the residual: diaPB and RQM. So, on the two discrete inputs, T_piece and RQM, the correlation coefficients show that the dynamic of the first one is well taken into account by the network when the RQM not. Similar results can be obtained on the validation data set. These results are very similar with those obtained on the validation data set.

Moreover, these two data (RQM and T_piece) are discrete ones. So, the correlation test is not the most significant. Figure 10 presents an example of the residuals in function of RQM. It can be thus noticed that two different residuals exist depending of the value of RQM. These two residuals are biased. This fact implies that this model introduces a systematic error. So, in order to estimate the influence of RQM on the residual the best approach is to compare these two samples.

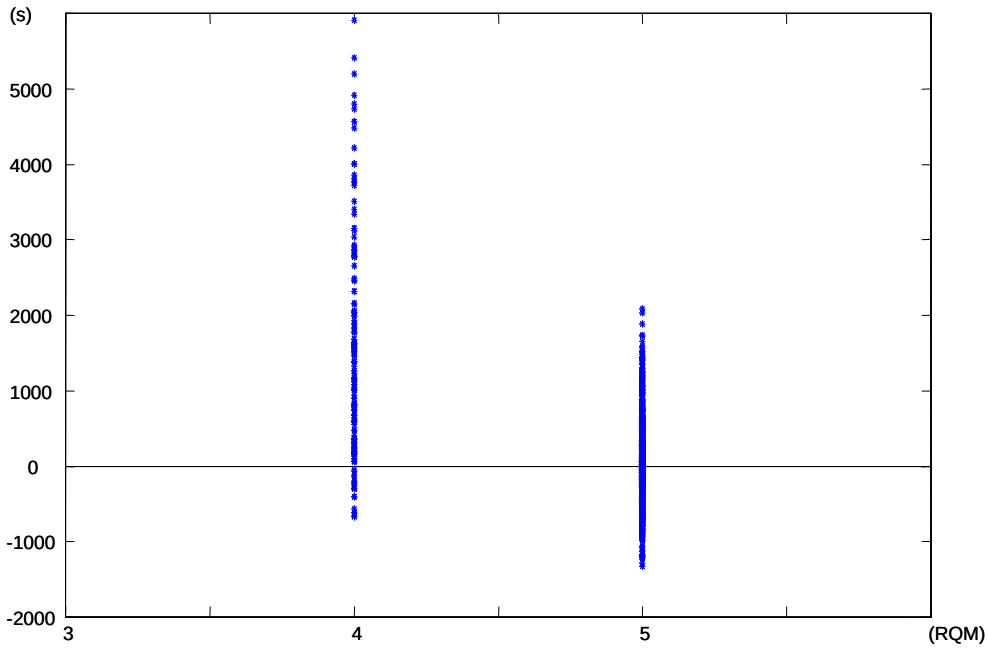

Figure 10. Residual function of RQM

For this, two tests can be performed. The first one is the T Student test which tests if the two samples of mean μ_1 and μ_2 have the same mean. The null hypothesis (H0) and its alternative (H1) are:

$$\begin{cases} H0: \ \mu_1 - \mu_2 = 0 \\ H1: \ \mu_1 - \mu_2 \neq 0 \end{cases} \quad (6)$$

The second test is the F Fisher test which is the ratio of the two variances σ^2_{max} and σ^2_{min} of the samples. The null hypothesis (H0) and its alternative (H1) are:

$$\begin{cases} H0: \ \sigma^2_{max} / \sigma^2_{min} = 1 \\ H1: \ \sigma^2_{max} / \sigma^2_{min} > 1 \end{cases} \quad (7)$$

The table 3 presents the results of these two tests with a confidence of 95% and 99% for the two variables RQM and T_piece for the 30 neural models constructed with the different initial sets of weights on the validation data set. The results on the learning data set are very similar. The data T_piece can take 3 values: type1; type2 and type3. So, the F test and the T test have to be performed two by two.

Table 3. Results of the Fisher and Student tests

	RQM		T_piece 1-2		T_piece 2-3		T_piece 1-3	
	F test	T test	F test	T test	F test	T test	F test	T test
Threshold 95%	1.092	1.961	1.070	1.961	1.077	1.961	1.070	1.961
Reject H0	100%	100%	96.67%	73.33%	43.33%	76.67%	96.67%	66.67%
Threshold 99%	1.130	2.583	1.101	2.583	1.127	2.583	1.101	2.583
Reject H0	100%	100%	93.33%	60%	10%	63.33%	90%	36.67%

These results show that RQM has an important influence on residual. Even with a confidence level of 99% no relation can be found between residuals obtained with RQM=4 and RQM=5. This is not the case with the T_piece data because the hypothesis of equality of mean (T test) is often not rejected and even the hypothesis of equality of variance (F test) is accepted to 90% between T_piece type2 and type3 with a confidence level of 99%. In conclusion, it seems that the residuals surrounded by circle figure 9 are due to cause ii): neural network does not succeed to learn some dynamics due to not taken into account root causes. In order to compensate this, a second approach is proposed.

b. Second approach

Here, two neural models have to be learned by using respectively the RQM=4 data only and the RQM=5 data only. These two neural networks have 11 inputs: Lg; diaGB; diaMoy; diaPB; T_piece; Q_trim; U_trim; Q_rqm; Q_rqm4; Q_rqm5; Q_rqm7. The learning begins with a structure using n_i=10 hidden neurons (5) which corresponds to 131 parameters. 30 different sets of initial weights are used. The table 4 presents the mean and the standard deviation of the residuals obtained on the learning and the validation data sets by using only RQM=4 data and RQM=5 data.

Table 4. Mean and standard deviation of the residuals – second approach

	RQM = 4				RQM = 5			
	Learning residual		Validation residual		Learning residual		Validation residual	
	Mean (s)	StD	Mean (s)	StD	Mean (s)	StD	Mean (s)	StD
Mean	12.36	478.00	8.40	528.33	7.22	332.80	7.75	335.54
StD	13.15	66.30	13.88	64.08	19.99	42.64	19.35	41.28
Min	-3.68	352.33	-19.55	376.15	-39.92	291.57	-37.28	291.83
(abs)	0.17		0.33		0.02		1.02	
Max	35.09	620.01	34.28	678.21	33.48	485.03	33.95	482.42

Table 5. Coefficients correlation between residual and inputs – second approach – RQM = 4

	Lg	diaGB	diaMoy	diaPB	T_piece	Q_trim	U_trim	Q_rqm	Q_rqm4	Q_rqm5	Q_rqm7
Mean	0.0225	0.0366	0.0371	0.0225	0.0257	0.0263	0.0189	0.0135	0.0157	0.0283	0.0168
StD	0.0313	0.0262	0.0262	0.0237	0.0227	0.0174	0.0208	0.0134	0.0132	0.0216	0.0106
Min	0.0005	0.0007	0.0020	0.0011	0.0000	0.0016	0.0014	0.0005	0.0003	0.0002	0.0011
Max	0.1305	0.0854	0.0870	0.1093	0.1123	0.0653	0.0791	0.0397	0.0451	0.0798	0.0398

Table 6. Coefficients correlation between residual and inputs – second approach – RQM = 5

	Lg	diaGB	diaMoy	DiaPB	T_piece	Q_trim	U_trim	Q_rqm	Q_rqm4	Q_rqm5	Q_rqm7
Mean	0.0135	0.0195	0.0227	0.0189	0.0405	0.0501	0.0275	0.0770	0.0722	0.0623	0.0530
StD	0.0257	0.0213	0.0272	0.0314	0.0453	0.0566	0.0267	0.0682	0.0695	0.0609	0.0445
Min	0.0009	0.0009	0.0002	0.0000	0.0000	0.0012	0.0016	0.0000	0.0000	0.0020	0.0005
Max	0.1058	0.0760	0.1053	0.1326	0.1330	0.2267	0.0920	0.2211	0.1847	0.2349	0.1577

The line (abs) presents the minimum of the mean in absolute value. It can be noticed that these values are very close to 0 to be compared with the results presented table 1 where the mean value is always greater than 12.35s. These results show that neural models present very similar residuals. In particular, the mean of the residuals is in the worst case, to 35.09s for the RQM=4 data and to 33.95s for the RQM=5 data. These results are to be compared with those presented table 1 where the mean of the residuals moves from 12.35s to 213.08s and where only 10% of the cases in learning and 16.67% of the case in validation give a mean lower than 30s. In order to determine if some dynamics present in the data are not taken into account by the learning of the two neural models, the correlation between the different inputs and the residuals can be performed on the learning data set for the RQM=4 data (table 5) and

for the RQM=5 data (table 6). Similar results can be obtained on the validation data set. The tables 5 and 6 present the mean, standard deviation, minimal and maximal values of the absolute value of the correlation coefficients obtained between the 30 residuals and the 11 inputs on the learning data set for the RQM=4 neural network and the RQM=5 neural network respectively. It can be noticed that, for the two neural models, no input is significantly correlated with the residual. In the worst case, the correlation coefficient obtained between Q_rqm5 input and the residual for the RQM=5 neural network is of 0.2349. However, for this input, in 76.67% of the cases, the correlation coefficient is lower than 0.01.

7. Conclusion and future work

The use of neural network in order to build a reduced model of emulation is investigated here. Within this framework, this paper focuses on the impact of discrete data on the learning results of the neural model.

The results have shown that some discrete data (T_piece) are perfectly taken into account without adaptation. This can be explained by the fact that, even if, these discrete data are useful for the comprehension of the system, they do not produce some very different behavior and a unique neural model can explain all its evolution. However, some discrete data (RQM) implies that some different behaviors of the process occur. These data imply that different models should be used in order to model all the system.

The perspectives of this work are to investigate how to use these discrete data in the best way. Besides, the proposed reduction emulation model approach must be applied to the modeling of one flexible manufacturing system in order to validate this approach for discrete events systems.

In addition, the system modeled may be changing. In this case, it may be interesting to use an on line learning rule in order to adapt the neural model to the evolution. Another perspective will be to investigate the advantages and disadvantages of this reduction model algorithm comparatively to a complete model. The computing times will be particularly studied.

Philippe Thomas received his Ph.D. from the University Henri Poincaré Nancy 1 in 1997. He spent five years with the Systèmes et Transports Laboratory at the Technical University of Belfort-Montbéliard, where he studied model-based diagnosis using neural networks. He is currently an Associate Professor and Researcher at the Centre de Recherche en Automatique de Nancy (CRAN-UMR 7039), which is affiliated with the CNRS and Nancy University. His research interests center on the use of neural networks to perform simulation models for scheduling.

André Thomas is Professor at ENSTIB (High School of Wood Sciences and Timber Engineering) and in charge of a technological research team at CRAN-CNRS – Nancy University. Certified in Value Management and CFPIM, he works with companies producing manufactured goods for research projects. His research topic is hybrid control of supply chains. Hybrid control refers to the use of centralized tactical planning and distributed short-term decision-making systems to control material flow on shop floors and into the supply chains. He is author of several papers published in scientific and technical reviews. He has developed several training tools in industrial engineering in the field of improvement and management of production systems.

Marie-Christine Suhner is Associate Professor at the ESSTIN (Nancy High School of Sciences and Technologies) - University Henri Poincaré Nancy 1 and researcher at the Centre de Recherche en Automatique de Nancy (CRAN-UMR 7039 CNRS Nancy University). She received her Master's Degree in Maintenance Engineering from ESSTIN in 1990 and her Ph.D. Degree in Automatic Control from the University Henri Poincaré Nancy 1 in 1994. Her research interests center on the use of Bayesian statistics applied to performance assessment and reliability of systems.

References

Barton, R.R., 1994. Metamodeling: A state of the art review. *Proc. of the 1994 Winter Simulation Conference*, 237–244.

Brooks, R.J., and Tobias, A.M., 2000. Simplification in the simulation of manufacturing systems. *Int. J. Prod. Res.*, 38(5): 1009-1027.

Chiu, M., and Lin., G., 2004. Collaborative supply chain planning using the artificial neural network approach. *J. of Manufacturing Technology Management*, 15(8), 787–796.

Chwif, L., Paul, R.J., and Pereira Barretto, M.R., 2006. Discret event simulation model reduction: A causal approach. *Simulation Modelling Practice and Theory*, 14: 930-944.

Cybenko, G., 1989. Approximation by superposition of a sigmoïdal function. *Math. Control Systems Signals*, 2(4): 303-314.

Delmotte, F., Dubois, L., and Borne, P., 1996. A general scheme for multi-model controller using trust. *Mathematics and Computers in Simulation*, 41:173-186.

Frantz, F.K., 1995. A taxonomy of model abstraction techniques. *Proc. of the 1995 Winter Simulation Conference*, Washington, USA.

Doumeing, M., 1989. Conception de systèmes flexibles de production. *Rapport interne du laboratoire GRAI*.

Funahashi, K., 1989. On the approximate realisation of continuous mapping by neural networks. *Neural Networks*, 2: 183-192.

Goldratt., E., and Cox, J., 1992. *The Goal: A process of ongoing improvement*, North River Press; 2nd Revised edition, Great Barrington, USA.

Ho, Y.C., 1987. Performance evaluation and perturbation analysis of discrete event dynamics systems. *IEEE Trans. on Automatic Control*, 32(7), 563–572.

Hung, Y.F., Leachman, R.C., 1999. Reduced simulation models of wafer fabrication facilities. *Int. J. Prod. Res.*, 37, 2685–2701.

Hwang, J.S., Hsieh, S., Chou, H.C., 1999. A Petri net based structure for AS/RS operation modeling. *Int. J. Prod. Res.*, 36, 3323–3346.

Innis, G.S., and Rexstad, E., 1983. Simulation model simplification techniques. *Simulation*, 41: 7-15.

Khouja, M., 1998. An aggregate production planning framework for the evaluation of volume flexibility. *Production Planning and Control*, 9(2), 127–137.

Kleijnen, J.P.C., and Sargent, R.G., 2000. A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research*, 120, 14–29.

Leachman, R.C., 1986. *Preliminary design and development of a corporate level production planning system for the semi conductor industry*, Eds Optimization in industry, Chichester, UK.

Li, J., Meerkov, S.M., and Zhang, L. 2009. Production systems engineering: Problems, solutions, and application, *13th IFAC Symp. on Information Control Problems in Manufacturing INCOM'09*, Moscow, Russia, June 3–5, 1–14.

Lopez, P., and Roubellat, F. 2001. *Ordonnancement de la production*, Hermès, Paris.

Mercé, C. 1987. *Cohérence des décisions en planification hiérarchisée*, Thèse de doctorat d'état, Université Paul Sabatier, Toulouse, France.

Page, E.H., Nicol, D.M., Balci, O., Fujimoto, R.M., Fishwick, P.A., L'Ecuyer, P., and Smith, R., 1999. An aggregate production planning framework for the evaluation of volume flexibility. *Winter Simulation Conf.*, 1509-1520.

Pierreval, H., 1996. A metamodel approach based on neural networks. *International Journal in Computer Simulation*, 6(3), 365–378.

Pritsker, A., and Snyder, K., 1994. *Simulation for planning and scheduling*, APICS, August.

Roder, P., 1994. *Visibility is the key to scheduling success*, APICS Planning and Scheduling, August

Setiono, R., and Leow, W.K., 2000. Pruned neural networks for regression. *6th Pacific RIM Int. Conf. on Artificial Intelligence PRICAI'00*, Melbourne, Australia, 500-509.

Shervais, S., Shannon, T.T., and Lendaris, G.G., 2003. Intelligent supply chain management using adaptive critic learning. *IEEE Trans. on Systems, Man and Cybernetics, Part A Systems and Humans*, 33(2), 235–244.

Suri, R., and Fu, B.R., 1994. On using continuous flow lines to model discrete production lines. *Discrete Event Dynamic Systems*, 4, 129-169.

Thierry, C., Thomas, A., and Bel, G., 2008. *Simulation for supply chain management*, John Wiley & Sons, London, UK.

Thomas, A., and Charpentier, P., 2001. De la pertinence de modèles réduits pour la prise de decision en réordonnancement. *Proc. of the 2nd International Conference on Integrated Design and Production CPI'01*, Fès, Morocco

Thomas, A., and Charpentier, P., 2005. Reducing simulation models for scheduling manufacturing facilities. *European Journal of Operational Research*, 161(1): 111-125.

Thomas, A., Genin, P., and Lamouri, S. 2008. Mathematical programming approaches for stable tactical and operational planning in supply chain and aps context. *Journal of Decision Systems*, 17: 425-455.

Thomas, P., and Bloch, G., 1996. From batch to recursive outlier-robust identification of non-linear dynamic systems with neural networks. *Proc. of the IEEE Int. Conf. on Neural Networks ICNN'96*, Washington D.C., USA, 1, 178–183.

Thomas, P., and Bloch, G., 1997. Initialization of one hidden layer feedforward neural networks for non linear system indetification. *Proc. of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics WC'97*, Berlin, Germany, 4, 195-300.

Thomas, P., Bloch, G., Sirou, F., and Eustache V.,1999. Neural modeling of an induction furnace using robust learning criteria. *J. of Integrated Computer Aided Engineering*, 6(1): 5-23.

Thomas, P., and Thomas, A., 2008. Sélection de la structure d'un perceptron multicouches pour la réduction d'un modèle de simulation d'une scierie. *CIFA'08*, Bucarest, Roumanie.

Tseng, T.Y., Ho, T.F., and Li, R.K., 1999. Mixing macro and micro flowtime estimation model: Wafer fabrication. *Int. J. of Production Research*, 37, 2447–2461

Vollmann, T.E., Berry, W.L., and Whybark D.C.,1992. *Manufacturing, Planning and Systems Control*. The Business One Irwin.

Ward, S.C., 1989. Argument for constructively simple models. *J. of the Op. Research Society*, 40(2): 141-153.

Zeigler, B.P., 1976. *Theory of modelling and simulation*. Wiley, New York.