Use of Neural Nets to Simulate the Spinning Process

F. Pynckels[+], P. Kiekens[*], S. Sette[*], L. Van Langenhove[*] and K. Impe[**]

[+] *Ministry of Finance, Treasury Department, Kunstlaan 30, 1040 Brussels, Belgium*
[*] *Department of Textiles, University of Ghent*
[**] *Ministry of The Flemish Community*

In a previous paper, a description was given how the spinnability of a given fibre quality on a rotor and ring spinning machine can be predicted with a reliability of 95% by means of a neural network. This paper goes further. It describes how yarn properties can be deducted from fibre properties and spinning machine settings. With other words, a description is given how to construct, train and use a neural network to simulate the spinning process (predict yarn properties) on rotor and ring spinning machines with an accuracy of over 95%.

1. INTRODUCTION

The aim of the present study (which falls within the framework of a BRITE/EURAM project BREU0052) consists in predicting the yarn properties, given the relevant fibre properties and production conditions. These conditions are determined by the type of machines and their settings. By means of a neural network, as described in *Pynckels (1995)*, it is possible to determine the spinnability of the fibres mentioned above, under the given production conditions. Once the spinnability is assured, one can predict the yarn properties by means of multiple interconnected neural networks. As mentioned in *Pynckels (1995)*, statistical methods as for instance multiple linear regression are not suited to resolve this problem since they assume in advance that the factors are independent and linear. Both assumptions are false in the case of a spinning process. With other words, a method is needed that does not oversimplify the problem in order to reduce the mathematical complexity of the solution. It is especially for this kind of problems that neural networks have proven to be a valid alternative as described in *Ramesh (1995)*.

2. TEST SET-UP

The test set-up is comparable to the one described in *Pynckels (1995)*. Twenty cotton types were selected based on their divergent properties. From the seventy-three important properties, a selection of 35 representative parameters was made using statistical methods. Afterwards, another 21 parameters were eliminated by means of neural network methods. The 14 fibre properties and the 5 machine parameters studied, together with the 9 yarn properties to predict are to be found in table I, table II and table III.
Yarn of 25, 30 and 50 tex was spun with twist factor α_{tex} of 3500, 4000 and 4500. Ring spinning was done on an SKF lab ring spinning machine. Rotor spinning was carried out on a Platt rotor spinner. For a total of 1382 spinnable cases, the resulting yarn properties were measured, and were kept, together with the corresponding fibre properties and machine parameters. These values are used later on to assess whether the used method is capable of simulating the spinning process, and whether the methods accuracy level is acceptable.

3. METHOD

3.1 Introduction

The aim of the designed method is to predict the yarn properties, given the fibre properties and the machine parameters. Opposite to *Cheng (1995)* and *Ramesh (1995)*, where one single net was used, coupled self-training systems were opted for in the research presented here. More precisely, several neural networks were trained, and put in cascade. It's to say, the output of the first neural network was fed into several other neural networks. These in turn generated output values that were used as input for yet another neural network. A detailed description why such a system was chosen, and how it works is given below. An introduction on neural networks (types and construction, training and testing, learning cycles, generalization versus specification properties, formulae for implementation) is to be found in *Klimasauskas (I)*, *Klimasauskas (II)*, *Lippman (1987)*, *Parker (TR-47)* and *Pynckels (1995)*.

3.2 Neural network construction

Although one large neural network, with a number of inputs equal to the sum of the number of fibre properties and the number of machine parameters, and a number of outputs equal to the number of yarn properties that must be predicted, can be opted for, an other neural network configuration was chosen. The main reason for choosing an other network topology is (as experience has shown) the inaccuracy of a single neural network to resolve a complex problem as the one stated above. More precisely, the major disadvantage of one large neural network is:

- The time to train the neural network is, in the case of a fully connected network, exponentially dependant on the number of input elements, hidden layer elements and output elements. The larger the network, the longer network training will take to accomplish, and the more complex it will be to train the network the way it should behave.

Taken the previous into account, one could opt for multiple neural networks, each of which calculate one yarn property, given all the fibre properties and machine parameters. The major disadvantages of this approach are:

- Since the number of inputs for each neural network still equals the sum of the number of fibre properties and machine parameters, since the number of neurons in each hidden layer largely depends upon the number of inputs and on the problem, and consequently is the same as for one large neural network, only the number of outputs decreases per neural network. On the other hand, the total duration for training the neural networks will increase since several neural networks must be trained.

- Since the yarn properties are interrelated, and the outputs generated by the different neural networks are not, an inherent and very useful characteristic of the yarn properties is neglected.

The previous reasoning points towards a certain direction to follow while constructing one or more neural networks to solve the stated problem:

- The number of neural networks to train is allowed to increase if the number of input neurons per network, and hence the number of neurons in the hidden layers decreases.

- The number of neural networks to train is allowed to increase if the number of hidden layers per network is kept as small as possible.

- The outputs of the different independent neural networks should be interrelated one way or the other.

So, a method is to be used that decreases the number of inputs by means of pre-filtering, and that interrelates the outputs by means of post filtering. The filtering may be done using mathematical methods, or also by using neural networks as filters. The connection scheme in figure 1 shows a construction of multiple connected neural networks, including a pre-filter neural network and a post filter neural network.

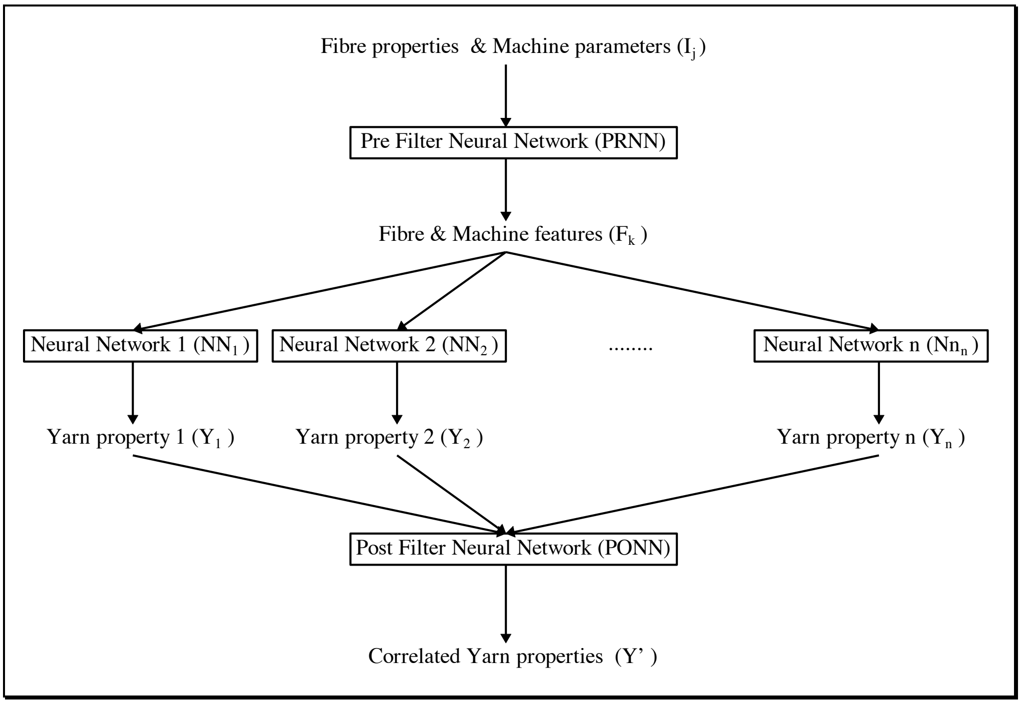

Figure 1 : Multiple connected neural nets

The network in figure 1 processes information as follows. Inputs I_j (fibre properties and machine parameters) are reduced by the pre filter neural network (PRNN) to fibre and machine features F_k . The features are fed into the neural nets NN_1 , NN_2 , ... , NN_n . Each of these neural nets predicts one yarn property (Y_1 , Y_2 , ... , Y_n). These yarn properties are then correlated by a post filter neural network (PONN) that predicts a yarn property vector (Y'). From here on the following notations will be used: N_I is the number of inputs, N_F is the number of features and N_Y is the number of yarn properties.

An extra advantage of this network construction is that eliminating a yarn property only requires the post filter network to be retrained. Likewise, adding a yarn property to predict only requires one extra network to be trained and the post filter network to be retrained. With other words, minor changes in desired output predictions only result in minor retraining. This means that adapting the described network construction to the wishes of some individual can be done fast and with great ease.

3.3 Construction of a pre-filter neural network

The main purpose of the pre-filter is to reduce the number of inputs for the neural networks NN_1 to NN_n each of which predict one yarn property, given features of fibre properties and machine parameters. Such a reduction results in a reduced number of hidden neurons, and hence in a reduction of the time to train those neural networks (NN_1 to NN_n). As stated before, the total number of inputs (fibre properties and machine parameters) is called N_I . The pre filter should reduce the number of input values (by combining them by means of a non-linear transformation) to a number of features N_F . The only condition for this reduction is that the reduced number of features must be 'reverse engineerable' to the correct original input value vector I. A fully connected cumulative backpropagation network can be constructed, as the one in figure 2, where there is only one hidden layer, and where the number of hidden neurons is kept smaller than the number of inputs. Such a network can be trained to reproduce the inputs. It's to say, the input values are also the desired output values, and consequently $N_i = N_o = N_I$. Decreasing the number N_h results in an increased error in the reproduction of the input values. This error however will be acceptable until N_h becomes less than a certain threshold value N_F . Consider the subnetwork consisting of the input neurons and the hidden neurons (the hidden neurons now serve as output neurons for this subnetwork, and there are no hidden layers in the subnetwork). This subnetwork can be seen as a feature extractor network, if the number N_h is chosen to be equal to N_F . The feature extractor subnetwork is shown in figure 2 within the boundaries of the dashed rectangle.

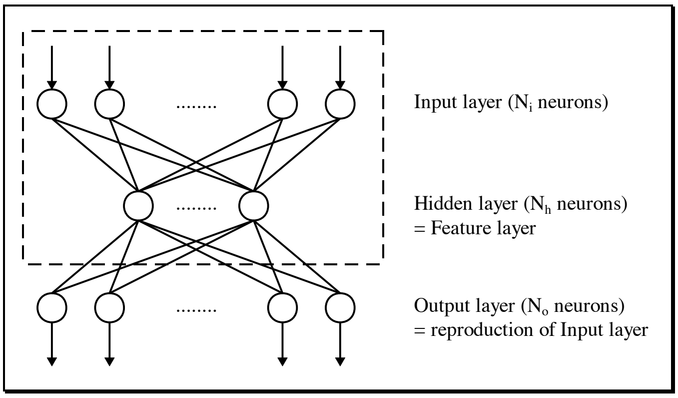

Figure 2 : Feature extractor neural subnetwork

3.4 Construction of a post filter neural network

As the neural network in figure 2 is studied in detail, one can see that the method used to extract the features is a reduction of the dimension of the input domain. Afterward, the reproduction of the input values embeds the feature domain into the domain of the input values by means of a non-linear transformation. The same type of network can be used to reduce and embed the yarn properties to yarn features and back to yarn properties. Since the yarn features are a sublimation of all the yarn properties in a subspace of the yarn properties domain, the degree of freedom is reduced. Afterward, the yarn features are used to regenerate a value vector that approximates the original yarn properties vector, but that is in fact a vector in the output space of the neural network onto which all yarn properties vectors will be mapped that are similar but distorted by noise or invalid interrelation. One can conclude that the same type of network that, if partially used, serves as feature extractor, also can serve as a noise filter and interrelation enhancer, if used in its three-layer form.

4. RESULTS

4.1 Yarn properties

The yarn properties of 1382 cases were measured, and were kept, together with the corresponding fibre properties and machine parameters used during the spinning process. From those 1382 data records, 1200 measurements were chosen at random to serve as learn values for all the neural subnets, and thus for the entire neural network. The resulting 182 data records were used as test values, which implicates that they were never used during the learning cycles of the neural subnets. In this way, it was possible to verify the specification behaviour (by means of the learn values) and the generalisation behaviour (by means of the test values) of the constructed neural network, as described in *Pynckels (1995)*.

4.2 Pre-filter neural network

Conform to the theory in 3.3, several neural nets were tested. For each net, the number of input neurons N_i and thus also the number of output neurons N_o was 19, being the sum of the number of fibre properties and the number of machine parameters used during this research. The number of hidden neurons (N_h) was varied from 19 down to 1. Figure 3 shows the total reproduction errors from each network for respectively the learn values and the test values. Taking this figure into account, one can decide on the number of features N_F by deciding upon the maximum tolerable reproduction error (ξ) and by choosing N_F to be equal to the minimal N_h for which the error is smaller than ξ. If ξ is chosen to be 0.02 (5% of the maximal reconstruction error), then it can be derived from figure 3 that the best and only choice for N_F is 16. This means that the error level of the described feature extractor network (extracting 16 features out of a total of 19 parameters) is below 0.02. The error level must be chosen low, since errors generated by this pre filter subnetwork are propagated further on through the entire neural network, resulting in much larger errors at the network output neurons. Since it is possible to use the output values from the hidden nodes of the pre filter neural network as input for the neural nets NN_1 , NN_2 , ... , NN_n and since these values can be generated, starting from the real input values of the neural network, the pre filter neural network can be called a *Feature Extractor Network* as we described it before.

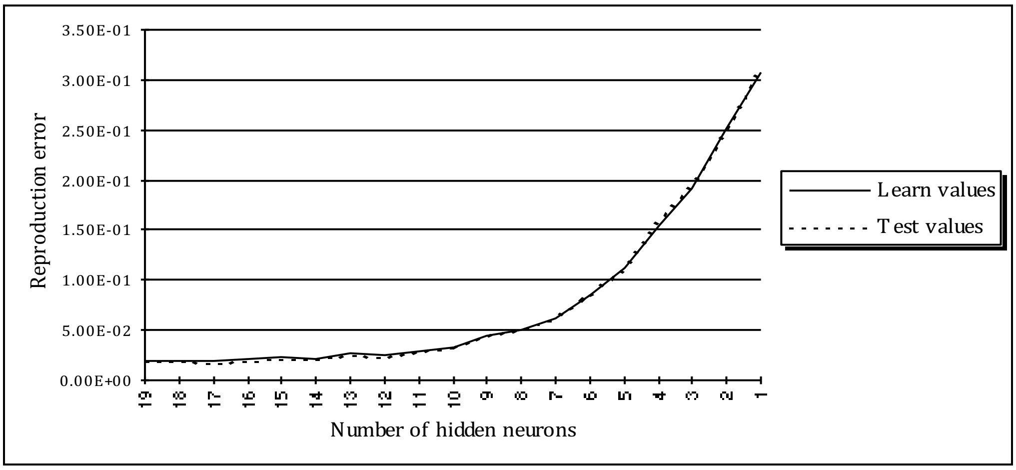

Figure 3 : Reproduction errors of the pre-filter neural network

4.3 Post filter neural network

As stated in 3.4 a network like the one in figure 2 was used as a post filter neural network. The same method used in 4.2 led to the error graphs in figure 4. Since the research focuses on 9 yarn properties to predict, a maximum of 9 hidden neurons can be used in the hidden layer of the post filter neural network. For a maximum reconstruction error of $\xi = 0.14$ (40% of the maximal reconstruction error), the number of hidden neurons N_h must be 7. In this case, the error margin can be chosen much larger, since the errors generated by this subnetwork are not propagated through other subnetworks, and consequently do not generate larger errors.

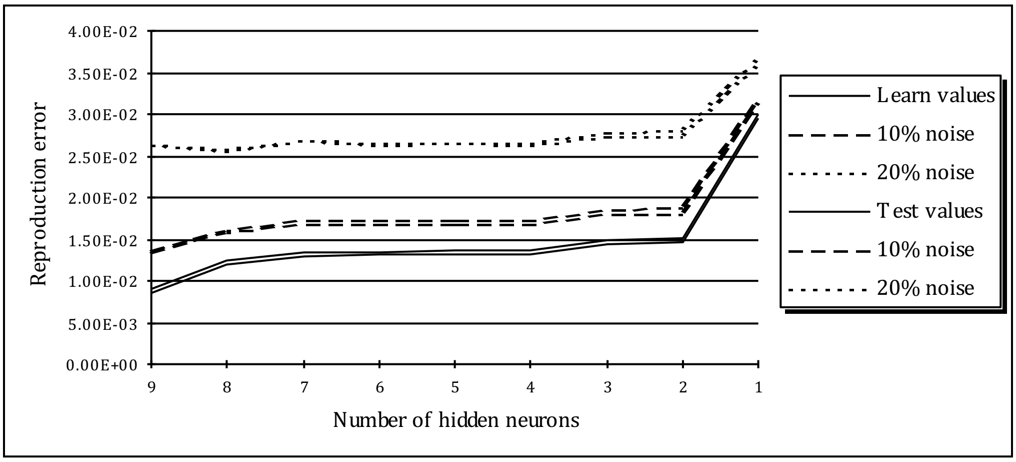

Figure 4 : Reproduction errors of the post filter neural network

Once the number of hidden neurons was decided upon, the post filter neural network was tested to confirm its noise filtering capabilities. This was done by generating a uniform noise at all input nodes with a maximal distribution of respectively 10% and 20% of the spread of the domain of the input neurons. Figure 4 shows clearly that the post filter neural network behaves as was expected for noise percentages of 20% and below. Research showed that, for noise percentages above 30% (an average uniform error of 30% on the input of every input neuron), the noise filter does not seem to generate correct results, but then again, what noise filter does?

Since generating noise on the inputs of the post filter neural network results in a correlation distortion of the values of the input vector, figure 4 also proves that the post filter neural network enforces correlation at its outputs (which are nothing else than the correlation enhanced version of the inputs). Indeed, because the calculated reconstruction error is the Euclidian distance between the generated output vector and the original input vector (without noise), and because this distance is smaller than the distance between the original input vector and the noise distorted input vector, it is obvious that the post filtering neural network enhances the correlation between the values of the input vector. With other words, the post filter neural network behaves as a *Correlation Enhancing Noise Filter*, the way we described it in 3.2 .

4.4 Prediction of yarn properties

Given the constructed feature extractor, it is possible to calculate the features for all the learn values. This can be done by feeding the yarn properties and the machine parameters as input vectors to the feature extractor, and by calculating the output of the hidden layer (feature layer) for each input vector. The feature vectors, together with the corresponding yarn properties can now be used as learn values to train the respective neural nets NN_1 , NN_2 , ... , NN_n . These nets can be constructed and trained as described in *Pynckels (1995)*.

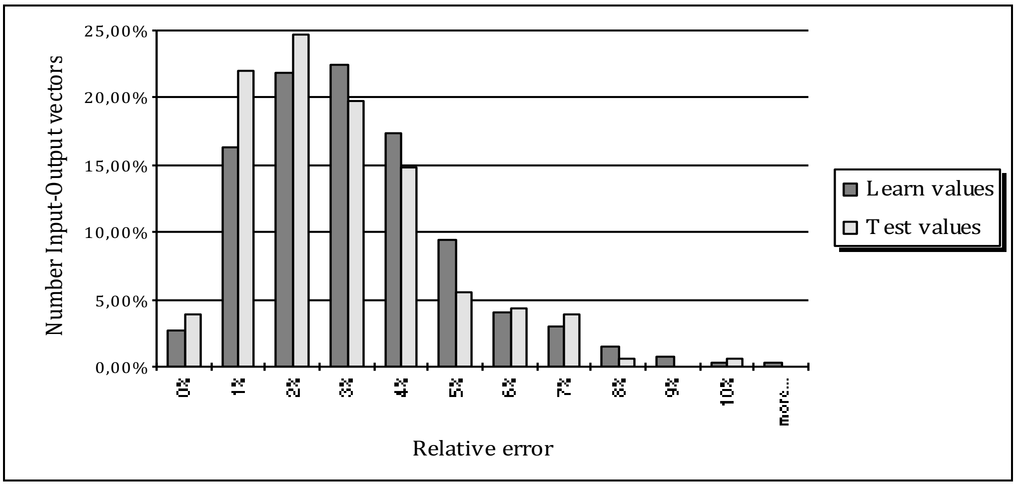

Figure 5 : Prediction error distribution

After training the nets NN_1 , NN_2 , ... , NN_n they can be combined with the feature extractor (extracting 16 features from 19 parameters) and with the correlation enhancing noise filter to get the desired neural network. Figure 5 shows the results for the learn values and for the test values as they are described in 4.1 . The error calculation is done by taking the Euclidian distance between the output vector (yarn property values) as it is generated by the neural network, and the real yarn property vectors as they were measured during the production of yarns. This distance is divided by the vector length of the real property vectors to retrieve a relative (percentual) error value.

The calculated relative error values result in error distribution graphs as the ones shown in figure 5. This figure shows for how many learn vectors and test vectors the interconnected neural network generates a relative error value of respectively 0%, 1%, 2%, etc... For values the network never saw before (test values), it generates an average error of 3.31%, with an average deviation of 1.37%. One can conclude that the neural network generates an error that is less than or equal to 5% (an accuracy of 95%) in 90.66% of the cases, and almost always an error that is less than or equal to 7% (an accuracy of 93%).

5. CONCLUSION

It can be stated that a neural network is a very useful and versatile tool not only to predict the spinnability of a yarn, but also to simulate the spinning process so a prediction can be made what the yarn properties will be, given fibre properties and machine parameters. The results have shown that a reliability of 95% and more can be reached. It goes without saying that these figures must be interpreted with the necessary caution, since although many tests were carried out, they were done on a very small scale. The basic structure of the interconnected neural networks as constructed in this paper can be applied to the prediction of the results of a spinning process using rotor or ring spinning machines. Therefore, the network must be trained with a series of data on fibre properties, machine settings and yarn specifications.
Further research is conducted in the field of coupling neural networks and fuzzy expert systems or genetic algorithms. The final goal is to be able to decide in which way a certain type of spinning machine must be set to spin a fibre into a yarn with properties determined beforehand. The used fibres and machine settings are depending on the availability of certain fibres, on the domain of the possible spinning machine settings, on the availability of the spinning machine for the production process and on a ruler base that drives the expert system.

REFERENCES

Cheng (1995) L. Cheng and D.L. Adams. 'Yarn Strength Prediction using Neural Networks, Part I : Fibre Properties and Yarn Strength Relationship', *Textile Research Journal*, Volume 65, Number 9, 1995, pp. 495-500

Klimasauskas (I) C. Klimasauskas and P. Guiver. *Neural Networks-I (revision 2.00)*, NeuralWare Inc.

Klimasauskas (II) C. Klimasauskas and P. Guiver. *Neural Networks-II (revision 2.00)*, NeuralWare Inc.

Lippman (1987) R.P. Lippmann. 'An Introduction to Computing with Neural Networks', *IEEE ASSP Magazine*, 1987 April, pp. 4-22

Parker (TR-47) D.B. Parker. 'Learning Logic', *Report TR-47*, Cambridge, Massachusetts Institute of Technology, Center for Computational Research in Economics and Management Science.

Pynckels (1995) F. Pynckels, P. Kiekens, S. Sette, L. Van Langenhove and K. Impe 'Use of Neural Nets for Determining the Spinnability of Fibres', *Journal of the Textile Institute*, Volume 86, Number 3, 1995, pp. 425-437

Ramesh (1995) M.C. Ramesh, R. Rajamanicham and S. Jayaraman. 'The prediction of Yarn Tensile Properties by using Artificial Neural Networks', *Journal of the Textile Institute*, Volume 86, Number 3, 1995, pp. 459-469

ACKNOWLEDGEMENTS

The authors would like to thank the Commission of the European Community for its financial support in the research, which falls within the framework of a BRITE/EURAM project on the setting-up of an expert system for the cotton spinning mill. We would also like to thank the laboratories of the Institut Textile de France-Section Nord, Trützschler, Harlander Coats and Zellweger for their help in the analysis of the fibres.

APPENDIX A FIBRE PROPERTIES, MACHINE PARAMETERS AND YARN PROPERTIES

The tables I, II and III show the fibre properties studied, the machine parameters taken into account, and the yarn properties predicted by the neural network. Due to available fibre qualities, some of the properties are overlapping each other.

Table I : Fibre properties

Parameters measured at the Textile Department of the University of Ghent

Symbol	Instrument	Fibre Property
hEL	HVI	elongation at break
hLEN	HVI	mean length
hMIC	HVI	micronaire
hSTR	HVI	strength
hUN	HVI	length uniformity

Parameters measured at the laboratory of the Institut Textile de France - Section Nord (Lille, France)

Symbol	Instrument	Fibre Property
hAREA	HVI	amount of trash (total surface taken by trash)
hB	HVI	yellowness
hRD	HVI	brightness
fFINdtex	FMT	fineness, expressed in dtex
fMIC	FMT	micronaire
fPM	FMT	% mature fibres

Other parameters

Symbol	Instrument	Fibre property
hCNT	HVI	number of trash particles
fMAT	FMT	maturity of fibres
fSTAND	FMT	standard fibre fineness for a maturity of 1

Table II : Machine parameters

Symbol	Machine parameter
BS	breaker speed
RS	rotor speed
ST	spin tube (number of carves)
TW	twist
YC	yarn count

Table III : Yarn properties

Symbol	Yarn property
CV_UST	linear regularity
ELO	elongation
TENA	tenacity
NUMBER	other irregularities per 1000 meters (yarncount)
TW	twist
THIN	thin places per 1000 meters
THICK	thick places per 1000 meters
NEPS	neps per 1000 meters
HRS	number of hairs per 250 meters