Principal Component Analysis for Greenhouse Modelling

NATHALIE PESSEL and JEAN-FRANCOIS BALMAT
Laboratoire Systèmes Information Signal
Equipe COSI
Université du Sud-Toulon-Var
B.P 20132 - 83957 La Garde Cedex
FRANCE
nathalie.pessel@univ-tln.fr – balmat@univ-tln.fr

Abstract: - This paper presents a Principal Component Analysis (PCA) study for neuronal modelling of a complex system. The PCA transforms a set of correlated variables into a smaller set of uncorrelated variables without lose the original information. Thanks to this first stage it is possible to design a simplified structure of the model. The right choice of the architecture is crucial for the application of neural nets in process identification.

The proposed study allows to validate the association of the PCA with neuronal model for a real multivariable process modelling: an experimental greenhouse. The object is to estimate the internal climate (temperature and hygrometry) by reducing the number of the input variables. Thus, we compare two different structures of neural networks. Several tests and results are presented and discussed.

Key-Words: - Principal component analysis, neural networks, multivariable system modelling, model reduction.

1 Introduction

The identification of multivariable systems is uneasy assignment, especially when the system is non-linear, non-stationary and strongly disturbed. In this case, it is often difficult to define a knowledge model (White Box model) which is the perfect representation of the system. The design of the models includes an important stage of selection and analyze of the set of the variables. Indeed, determining the relevance of the inputs is of great importance in practical modelling problems [5]. In these conditions, the reduction of the number of variables (size of the input vector) and the synthesized data representation turn out to be important elements in the modelling of these systems.

This paper presents a Principal Component Analysis (PCA) study associated to a neuronal approach (Black Box model) for the modelling of a multivariable system. In the literature, the PCA is used to create a set of new uncorrelated variables (principal components) [7], [11]. In this paper, contrary to the classical uses, the PCA is used to select the relevant input variables (physical sensors). To validate this approach, we achieved an application on a real system (an experimental greenhouse). Successive studies about this application area allowed us to get a proficient knowledge of this system. The reactions of this system are now known and some data evolutions can intuitively be foreseen. The PCA allows to give mathematical explanations of intuitive facts [10].

The following paragraphs detail all components required for our study. At first, we introduce the PCA, for the variables selection. The second part describes the principle of the neuronal modelling and presents the type of used neural networks. Section 4, we detail our experimental process: a greenhouse. In the last section, we present some results.

2 Principal Component Analysis

The PCA is a statistical method, which is included in the more general context of the factorial analysis. The PCA allows to reduce a complex correlation system into a smaller number of dimensions [6], [7]. When there are correlations between the m descriptive variables of a data distribution, the m dimensions of the data space exceeds the n number of characteristic variables necessary to describe these data. The higher the correlations between data descriptive variables, the smaller the number of useful characteristic variables for their representation.

In this paper, the PCA is used to choose into the set of correlated variables given by the greenhouse sensors, a reduced set of uncorrelated variables (sensors). The set of the initial data is represented by the matrix *X*. The size of this matrix is $p \times r$ (*p* variables and *r* samples).

The variables of our system have different scales and units. However, we wish that each variable to have the same weight in the system analysis. The data of each variable are centred and reduced. The PCA is said normed. A new matrix of data X_{norm} is defined by:

$$X_{norm}(i) = \frac{X_i - \overline{X}_i}{S_i} \qquad (1)$$

in which X_i is the i^{th} row vector of the matrix *X*, $\overline{X}_i$ is the mean of this vector with $\forall i \in [1, p]$ and S_i is the standard deviation of the considered variable *i*.

The identification of the PCA model parameters (i.e. principal components) is achieved by the estimation of the eigenvalues $\lambda_1,...,\lambda_p$ and eigenvectors $u_1,...,u_p$ of the correlation matrix R_x. The principal components Y_i are a new set of data estimated by:

$$Y_i = X_{norm}\, u_i \qquad (2)$$

in which u_i is the i$^{\text{th}}$ eigenvectors of the correlation matrix R_x.

The eigenvalues correspond to the variance part of each component. We can explain the percentage of variability of each Principal Component (PC) by:

$$W_{PC}(i) = \frac{\lambda_i}{\sum \lambda} \times 100 \qquad (3)$$

These percentages allow to select the number of representative PC of the system.

The study of the correlation of the initial variables is based on the correlations of PC with each initial variable. The coefficients of correlation C_i are obtained by the multiplication between each eigenvector and the square root of the eigenvalue associated (they vary between -1 and 1):

$$C_i = u_i \sqrt{\lambda_i} \qquad (4)$$

The PCA allows to achieve a graphic representation of the information by using the coefficients of correlation. The new space of visualization is a circle axes of which are two of selected PC. This circle is called the “factorial space” or “correlation circle”.

Two correlated variables can be identified by studying the projections of their coefficients of correlation. Indeed, two variables are correlated if the projections of their vector are close both to the circle and to themselves. The angle between two variables projected on the correlation circle is equal to the coefficient of correlation between these variables. This angle (α) is measured by its cosine:

$$cos\,\alpha = angle(C_i, C_j) \qquad (5)$$

with *i* and $j \in [1, p]$. Thus, if the projection of two variables are both closed to themselves (α little different of $2k\pi$), so the variables X_i and X_j are correlated. By opposition, if α is equal to 90°, the variables X_i and X_j are not correlated.

Therefore, this analysis enables to define a simplified and efficient model of our system. This new reduced system is used to modelling our system with a multilayer neural network.

3 Modelling by neural networks

Neural networks are considered to be useful for this purpose due to their ability to approximate a wide class of continuous functions. In this case, the identification is based exclusively on measured data. The identification process is called “black box” modelling. The number of input and output nodes is determined according to the nature of the modelling problem being tackled, the input data representation and the form of the network output required.

For these reasons, to define a neuronal model [5], it is very important to choose the input variables and the set of data judiciously. Generally, it is necessary to do a pre-processing and an analysis of data files. The number of samples (measures) and the number of the input variables (sensors) are the parameters which have a strong influence on the structure of neural networks.

In this study, we are interested us to the reducing of the dimension of the inputs in order to improve their relevance in comparison to the system to be modelised. It is necessary to remember that the choice of the neuronal structure must take into account the bias/variance dilemma (apprenticeship capacity / generalisation capacity).

Thus, it can be interesting to define a neural network with a less complex structure but with a sufficient number of parameters. In this way, we use the PCA method, and we model an experimental process with a neural network [1], [2].

In this study, we use basic multilayer feedforward neural networks (Fig. 1) with one hidden layer. The neurons number of the hidden layer is related to the complexity of the system being modelised. The adjustment of the network's weights is accomplished by using the Levenberg-Marquardt algorithm, such as:

$$\Delta W = (J^T J + \eta I)^{-1} J^T \varepsilon \qquad (6)$$

where *W* is the weights matrix, *J* is the Jacobian matrix of derivatives of each error to each weight, ε is the error vector and η is a learning parameter.

For $\eta = 0$, the algorithm becomes the Gauss-Newton method. When η is very large (changes to ∞) it becomes the steepest descent or the Error Backpropagation algorithm.
The parameter is automatically adjusted at each iteration in order to secure convergence [12]. The parameter η is initialized and it is increased when the error increases. The second-order convergence properties of the Levenberg-Marquardt method enabled faster training of the models.

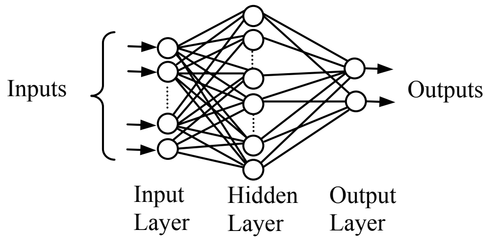

Fig. 1. Multilayer feedforward neural network

After the learning stage, the weights are fixed and used in the validation stage.

4 Greenhouse description

Climatic management aims at simultaneously maintaining the sets of the climatic factors (temperature, hygrometry, rate of CO_2) according to their respective references while respecting certain rules (absolute or conditional prohibitions, priorities, times of temporization) imposed by the user.
In our laboratory, an experimental greenhouse is used to search and develop new commands [3]. Our objective is to develop a regulation, which takes into account the state of the plants.
Our greenhouse is equipped with many sensors (sensors for internal and external temperature *Ti* and *Te*, data expressed in °C ; for internal and external hygrometry *Hi* and *He,* data expressed in % ; for the wind velocity *Vv*, data expressed in $m.s^{-1}$ and for the global radiation *Rg*, data expressed in $W.m^{-2}$) and with various commands (of the heating *Ch*, binary command ; the roofing *Ov*, command expressed in degrees ; the moistening *Br*, binary command and the shutter *Rd*, command expressed in centimetres) (Fig. 2).
For this system, the definition of a knowledge model is difficult. The dynamic behaviour of the greenhouse climate [9] depends of:

- an important number of variables (references, perturbations, commands, sensors)
- the complexity of phenomena due to the process (biologic, weather, evolution of plants…).

Thus, the process is multivariable, non-linear, non-stationary and strongly disturbed.

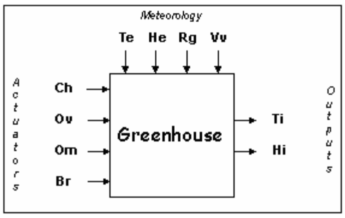

Fig. 2. The greenhouse model

Moreover, the perturbations (the wind velocity and the global radiation, for instance) can sometimes be more powerful than the command (the heating, for instance). The modelling of the system must take into account these difficulties.

5 Experimental results

Results presented below are organized in three parts. The first part presents data used to validate our work. The second part illustrates data associations issued from a PCA applied on experimental greenhouse data. The last part concerns the neuronal modelling of our system with and without the reduction of the input variables number.

5.1 Data presentation

The experimentations relating to the real data were carried out using experimental greenhouse data of days in March. The scale of time is very short, about one minute. The size of the file for a day is 10 × 1440 (10 variables and 1440 samples). The 10 variables are 6 sensors and 4 actuators. The actuators are “actionable” independently in function of the part of daytime.
A first data group used to the training stage has been formed with three days in March. The size of the apprenticeship file for three day is 10 × 4320.

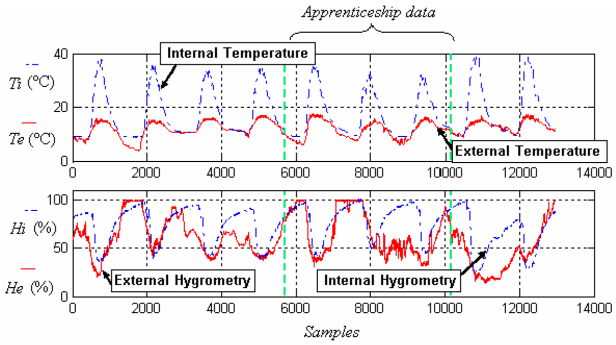

Fig. 3. Temperature and hygrometry of validation stage

The validation stage is carried out by using nine days on March (Fig. 3 and Fig. 4). The size of the validation file for ten day is 10 × 12960.

The days used to the training stage is included in the validation stage. The Fig. 3 depicts temperature and hygrometry of validation stage. Fig. 4 presents the disturbances and actuators of the validation stage and the apprenticeship data borders [8].

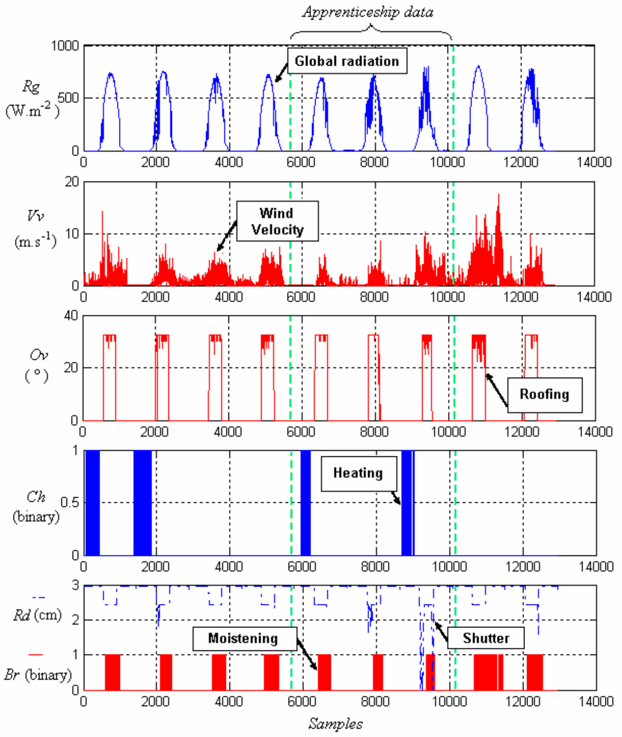

Fig. 4. Disturbances and actuators of validation stage

5.2 Reduction of the variables number

We have applied a PCA on the 10 variables. The first result given by the PCA allows us to determine the number of PC necessary to represent the system. This choice depends on the percentage of variability of each PC (1). Fig. 5 is a plot of the eigenvalues versus the PC number (including the percentage of variability of each PC).

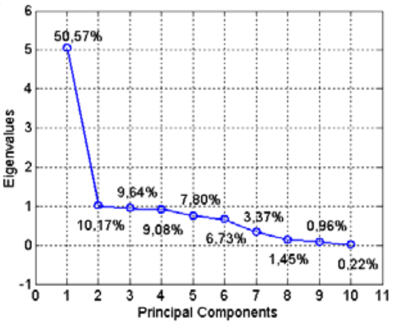

Fig. 5. Percentage of variability for each PC

We observe that the first four PC allow us to explain 80% of the variability and the first two 60%. The first PC selected create a new space of visualization which allows to evaluate correlations between variables.

Fig. 6 shows the projections of the variables in the correlation circle on plane PC1-PC2.

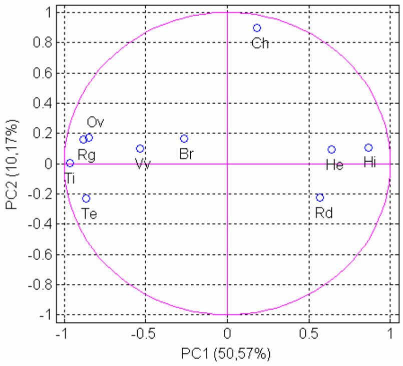

Fig. 6. Correlation circle on plane PC1-PC2

Three groups of two variables can be identified. By definition, two variables are correlated if they are close both to themselves and to the circle. We observe that *Rg* and *Ov* are two correlated variables; *Te* and *Ti* are correlated in the same way, just as *He* and *Hi* are. Supplementary information is contained in Fig. 6, the two groups {Rg, Ov} and {Te, Ti} are located on the same side of the circle and close to the circle. We conclude these four variables are correlated. These results are found in the Table 1 which presents angles (in degrees) between two variables projected on the plane PC1-PC2.

	Te	*Ti*	*He*	*Hi*	*Vv*	*Rg*	*Ch*	*Br*	*Ov*	*Rd*
Te	-	**15.0**	173.2	172.1	-	25.1	116.2	-	26.3	143.5
Ti		-	171.8	172.9	-	**10.1**	101.2	-	**11.3**	158.5
He			-	**1.1**	-	161.7	70.6	-	160.58	29.6
Hi				-	-	162.8	71.7	-	161.6	28.6
Vv	-	-	-	-	-	-	-	-	-	-
Rg					-	-	91.1	-	**1.2**	168.6
Ch					-		-	-	89.9	100.3
Br	-	-	-	-	-	-	-	-	-	-
Ov					-			-	-	169.8
Rd					-			-		-

Table 1. Angle (in degrees) between two variables projected on the plane PC1-PC2

The angle allows to quantify the notion of "close both to themselves". Indeed, two variables are closed both to themselves if the angle which they formed is little different than $2k\pi$ (< 18°, angles written in thick). Moreover, we can observe that the two variables *Vv* and *Br* are missing because they are not close to the circle. We observe the same correlations as with the correlation circle.

Fig. 7 presents the two correlation circles on planes PC1-PC3 and PC2-PC3.

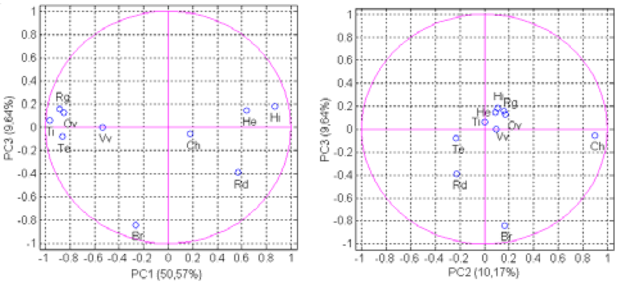

Fig. 7. Correlation circles on plane PC1-PC3 (a) and on plane PC2-PC3 (b)

The previous observations are verified by the projections of the variables on these other planes.

Fig. 8 presents the projection of the samples on plane PC1-PC2. The samples are separated in four groups according to the global radiation:

- the day samples for *Rg* > to 15 %
- the night samples for 0 < *Rg* < 5 %,
- the daybreak samples for 5 < *Rg* < 15 %,
- the dusk samples for 15 > *Rg* > 5 %.

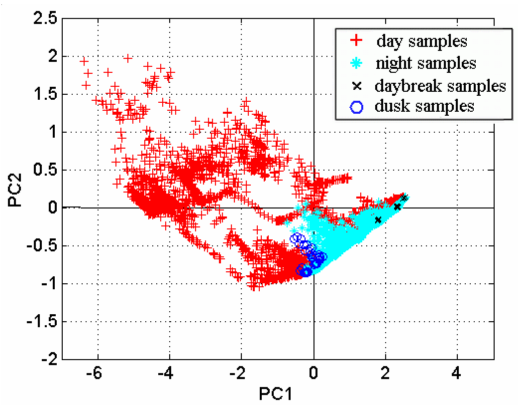

Fig. 8. Samples representation on the plane PC1-PC2

We observe that samples corresponding to night, daybreak and dusk are brought together and focused on the bottom right hand side of the plot. Moreover, the day samples are located on the left part and the greater part of them at the top of the plot. This samples distribution illustrates the disposition of the variable groups defined in Fig. 6: day area which is characterised by the global radiation and the temperature, is located in the left hand side on plane PC1-PC2. By contrast, the daybreak area is located on the right hand side on plane PC1-PC2 like group {*He*, *Hi*} on Fig. 6.

The PCA applied to experimental greenhouse data allows to create two correlated variable groups {*Te*, *Ti*, *Rg*, *Ov*} and {*He*, *Hi*}. The neural network modelling of the system is carried out with six variables.

5.3 Greenhouse modelling with Neural Networks

The goal of this part is to construct different neural networks (with and without the reduction of the number of the input variables) and to compare the models quality obtained. The reduction of the dimension of the state inputs allows to transform and simplify the model structure. For each structure model, we keep the same number of neurons in hidden layer.

The transfer function for each output neuron is linear with bias (Fig. 9.a) when for each hidden neuron is sigmoid with bias (Fig. 9.b).

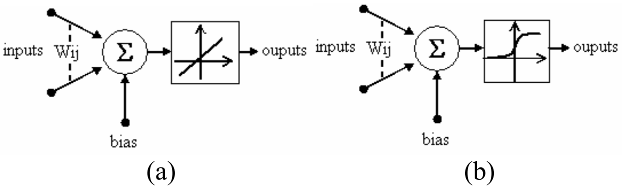

Fig. 9. Transfer function of the neurons
(a: Linear transfer function, b: Sigmoid transfer function)

As explained in the previous sections, we search for eliminate the input variables to reduce the parameters number of the neuronal model. The PCA method allows us to group the following input variables: {*Te, Ti, Rg, Ov*} and {*He, Hi*}. For each group, we must choose one variable. This choice depends of the system knowledge (physical and experimental). In this way, for our system, we have selected *Te* for the first group and *He* for the second. So, we obtain different networks (statics or dynamics) in accordance with the choice of the inputs. The static networks realize a non-linear algebraic function of the inputs while the dynamic networks are governed by a recurrent equation. The model structures are depicted in Fig. 10. Like this, we present the simplified structure (Fig. 10.a) and complete structure (Fig. 10.b) which correspond respectively at a static and dynamic neural network.

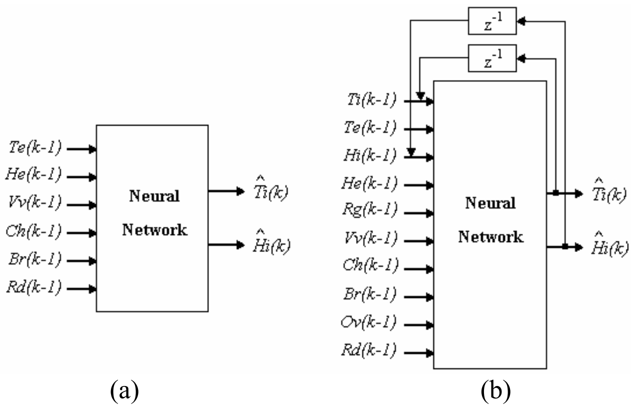

Fig. 10. Neuronal models with (a) and without (b) the reduction

For the training stage, we use three days of March (4320 samples) and for the validation stage, we use nine days of March (12960 samples). The training parameters are:

- number of iterations = 200
- initial apprenticeship coefficient = 0.001
- number of inputs units = 10 or 6
- number of hidden units = 8

In the following figures, the internal temperature curves (real Ti and estimated $\hat{Ti}$) and the internal hygrometry curves (real Hi and estimated $\hat{Hi}$) are illustrated.

We plot $\hat{Ti}$ and $\hat{Hi}$ by using the complete model with all the input variables (Fig. 11 and Fig. 13) and by using the simplified model with six input variables (Fig. 12 and Fig. 14).

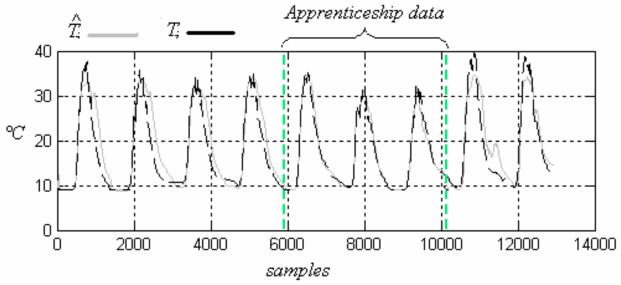

Fig. 11. *Ti* (complete model, validation data)

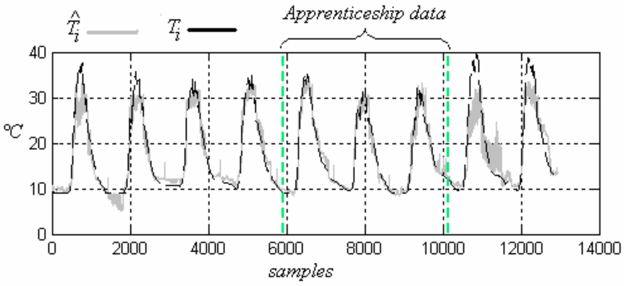

Fig. 12. *Ti* (simplified model, validation data)

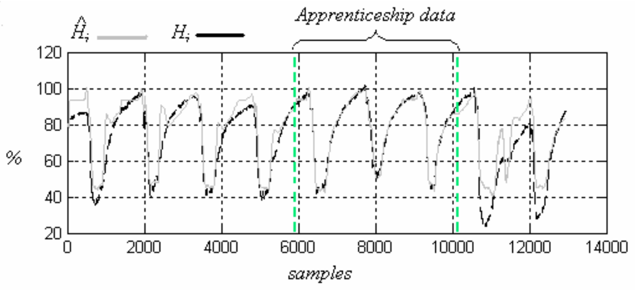

Fig. 13. *Hi* (complete model, validation data)

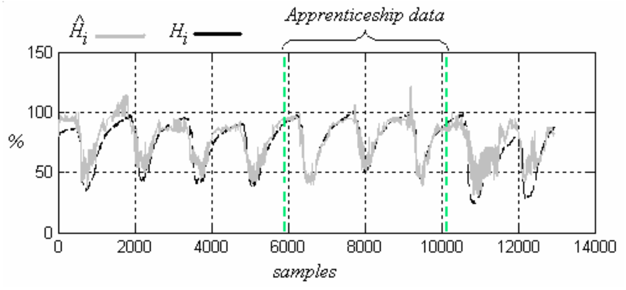

Fig. 14. *Hi* (simplified model, validation data)

To compare these models, we use several criteria which are the Mean Error (ME), the Variance Error (VE) (Table 2) and the Akaike's Information Criterion (AIC) [4]:

$$AIC = N\,ln\left(\frac{1}{N}\sum_{k=1}^{N}\left(\hat{Ti}(k)-Ti(k)\right)^2\right)+2q \qquad (7)$$
$$q=\left(N_i+1\right)N_h+\left(N_h+1\right)N_0$$

where N is the total number of data, N_i is the number of inputs, N_h is the number of hidden units, and N_0 is the number of outputs. q is the number of parameters used (weights and bias).

	Criteria	Complete structure	Simplified structure
Ti	ME	0.64 °C	0.99 °C
	VE	1.0	1.16
Hi	ME	1.32 %	3.17 %
	VE	2.01	4.45

Table 2. ME and VE for each model (apprenticeship data)

The AIC criterion takes into account the model complexity and the mean square error (compromise between goodness of fit and parsimony). The best model is the one that has the smallest AIC.

For the internal temperature estimation, we remark similar performances (see Table 3). The internal hygrometry modelling is more difficult because *Hi* is more sensible at the quick variations of the external hygrometry. For the simplified model, the results are less efficient. We can explain it because we take into account only of the external hygrometry (Fig. 10).

	Criteria	Complete structure	Simplified structure
Ti	ME	1.83 °C	1.72 °C
	VE	4.59	4.87
	AIC	14.9581	14.9560
Hi	ME	4.98 %	7.44 %
	VE	28.18	50.71
	AIC	17.4380	18.0310

Table 3. ME, VE and AIC for each model (validation data)

In conclusion, the obtained results allow to show a correct modelling quality for the two structures. The simplified model is able to give a good estimation of the internal climate. In the apprenticeship stage, we can see that the Execution Time (ET) is longer when we use all the input variables (Table 4).

	Complete structure	Simplified structure
ET	175.5	123.2

Table 4. Execution Time (ET) (in elapsed CPU seconds) for each model (apprenticeship stage)

Therefore, it can be concluded that the proposed structure model represent a viable alternative to the experimental greenhouse modelling.

6 Conclusion

This paper presents a PCA applied to the complex system modelling. The aim is to simplify the model by keeping an efficient model. Thanks to the PCA and the knowledge of the system, we can define a set of uncorrelated and relevant variables (input sensors). Thus, we show that it is possible to obtain a simplify model. This first stage allows us to design the structure of a neuronal model.

We evaluated this approach on a real process: an experimental greenhouse. Some experiments were carried out with different sets of data.

The first remark is that the correlated variables emphasized by the PCA method reflect those sensed by the knowledge of the system. In addition, the PCA allows to explain mathematically and automatically the redundancy between variables.

Secondly, the neural networks using allows to obtain an efficient model when we reduce the number of input variables. In this case, we note that the variables associated of the internal climate (*Ti* and *Hi*) are not necessary. So, we can obtain the evolution of the internal climate in the experimental greenhouse by only exploring meteorology sensors. Therefore, the number of sensors can be reduced by preserving a good quality of modelling.

The paper presents an original approach which associates the expert knowledge of a system with the PCA. The results obtained on a real system (experimental greenhouse) are efficient.

In a future work, this simulation model will be used to compare several types of control laws to regulate the greenhouse micro-climate.

References:

[1] J.F. Balmat, F. Lafont, "Multi-model architecture supervised by Kohonen map", *in Sciences of Electronic, Technology of Information and Telecommunication,* 2003, pp. 98-104.

[2] M. A. Bussab, J. I. Bernardo, A. R. Hirakawa, "Greenhouse Modeling Using Neural Networks", *in Proc. of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, 2007, pp. 131-135.

[3] F. Lafont, J.F. Balmat, "Fuzzy logic to the identification and the command of the multidimensional systems", Invited Paper, *International Journal of Computational Cognition*, Vol. 2, n° 3, 2004, pp. 21-47.

[4] B. Minasny, A.B. McBratney, "Neural networks package for fitting – Pedotransfer functions", *Technical note, Australian Centre for Precision Agriculture*, 2002.

[5] M. Norgaad, O. Ravn, N.K. Poulson, L.K. Hansen, "Neural Networks for Modelling and Control of Dynamic Systems", Advanced Text books in *Control and Signal Processing*, Springer, 2000.

[6] B. Parinet, A. Lhote, B. Legube, "Principal component analysis: an appropriate tool for water quality evaluation and management – application to a tropical lake system", *In Ecological modelling*, Vol. 178, 2004, pp. 295-311.

[7] R. Penha, J. W. Hines, "Using Principal Component Analysis Modeling to Monitor Temperature Sensors in a Nuclear Research Reactor", *In Proc. of* the *Maintenance and Reliability Conference*, 2001.

[8] N. Pessel, J. Duplaix, J.-F. Balmat, F. Lafont, "Data Analysis for Neuro-Fuzzy Model Approach", *in IEEE Int. Workshop on Soft Computing Applications,* 2005, pp. 44-50.

[9] P. Salgado, J. Boaventura Cunha, "Greenhouse climate hierarchical fuzzy modelling", *Control Engineering Practice*, Vol. 13, Elsevier, 2005, pp. 613-628.

[10] S. Saitta, B. Raphael, I. F. C. Smith, "Data Mining for Decision Support in Multiple-Model System Identification*", In Proc. of the 6th WSEAS Int. Conf. on Simulation, Modelling and Optimization*, 2006, pp. 161-166.

[11] N. Székely, "Simplifying the Model of a Complex Industrial Process Using Input Variable Selection"*, Periodica Polytechnica*, Vol. 47, 2003, pp. 141-147.

[12] B.M. Wilamowski, S. Iplikçi, M.O. Efe, O. Kaynak, "An Algorithm for Fast Convergence in Training Neural Networks", *Int. Joint Conf. On Neural Networks,* 2001, pp. 1778-1782.