IDENTIFICATION AND VALIDATION OF A NEURAL NETWORK ESTIMATING THE FLUXES OF AN INDUCTION MACHINE.

L. Constant *, R. Ruelland *, B. Dagues *, I. Rivals **, L. Personnaz **

email :dagues@leei.enseeiht.fr

Laboratoire d'Electrotechnique et d'Electronique Industrielle (*)
Unité Mixte de Recherche INPT-ENSEEIHT / CNRS n°5828
BP 7122-2 rue Camichel-31071 TOULOUSE Cedex 7-France
☎ (33) 5 61588339 - Fax : (33) 5 61638875

Laboratoire d'Electronique (**)
de l'Ecole Superieure de Physique et de Chimie Industrielles de Paris (ESPCI)
UPR CNRS 9081 NETS - PARIS
10 rue Vauquelin
75231 PARIS Cedex 05 - FRANCE

Abstract : In the frame of a study on real-time emulators of electromechanical systems, we have built a neural model of an induction machine. An original methodology is used to design the neural network architecture, as well as training sequences allowing a proper identification of the induction machine behavior in its whole operating range. In the same spirit, exhaustive test sequences are built in order to obtain an accurate estimate of the neural model performance.

1.- INTRODUCTION

The present paper deals with the modeling of electromechanical systems using neural networks. In the frame of a study on emulators of {static converters / electrical machines / sensors} associations [2], our aim is to simulate complex electromechanical systems in real-time. The ability to parsimoniously approximate nonlinear mappings [3], as well as the possibility of parallel computing, make neural networks efficient in terms both of accuracy and computation time. The use of feedback (dynamic) neural networks [4] [5] thus opens up new horizons to the nonlinear modeling of complex dynamic nonlinear systems. The design of a first neural model of the induction machine has been presented in the previous papers [6] [7]. This neural model consists of two parts. The first part is a feeback neural network whose state outputs are the stator and rotor fluxes, and whose inputs are the stator voltages and the mechanical rotor speed. The second part is a feedforward (static) network, which computes the stator currents and the electromagnetic torque from the fluxes. This neural model estimates the output variables of interest with a global error of 3% on a machine start-up test sequence. This relatively large error is mainly due to the inaccuracy of the feedback network. This paper thus presents a new methodology for the design of training sequences allowing a better identification of the feedback network in the whole operating range of the fluxes [1]. In order to take the fast dynamics of the converter into account, a very small sampling period must be used (at most T=10 µs). A major difficulty encountered for this design was hence the necessity to avoid too large sequences for the exploration of the whole state space.

2.- NEURAL MODEL ARCHITECTURE

The neural model architecture is shown in Fig. 1. This architecture is directly derived from the analytic relations (1) of a two-phase model in the reference frame (α, β) fixed on the stator. All the training and test sequences are generated using the two-phase model, which will be termed "reference model".

$$\begin{cases} \dfrac{d\phi_{s\alpha}}{dt} = -R_s \dfrac{1}{\sigma.L_s}\phi_{s\alpha} + R_s \dfrac{1-\sigma}{\sigma.M}\phi_{r\alpha} + V_{s\alpha} \\ \dfrac{d\phi_{s\beta}}{dt} = -R_s \dfrac{1}{\sigma.L_s}\phi_{s\beta} + R_s \dfrac{1-\sigma}{\sigma.M}\phi_{r\beta} + V_{s\beta} \\ \dfrac{d\phi_{r\alpha}}{dt} = +R_r \dfrac{1-\sigma}{\sigma.M}\phi_{s\alpha} - R_r \dfrac{1}{\sigma.L_r}\phi_{r\alpha} - \Omega.\phi_{r\beta} \\ \dfrac{d\phi_{r\beta}}{dt} = +R_r \dfrac{1-\sigma}{\sigma.M}\phi_{s\beta} - R_r \dfrac{1}{\sigma.L_r}\phi_{r\beta} + \Omega.\phi_{r\alpha} \end{cases} \quad (1)$$

In the architecture of the neural network, a hidden neuron is linked with only one output neuron, in this way each flux is computed with an autonomous subnetwork.

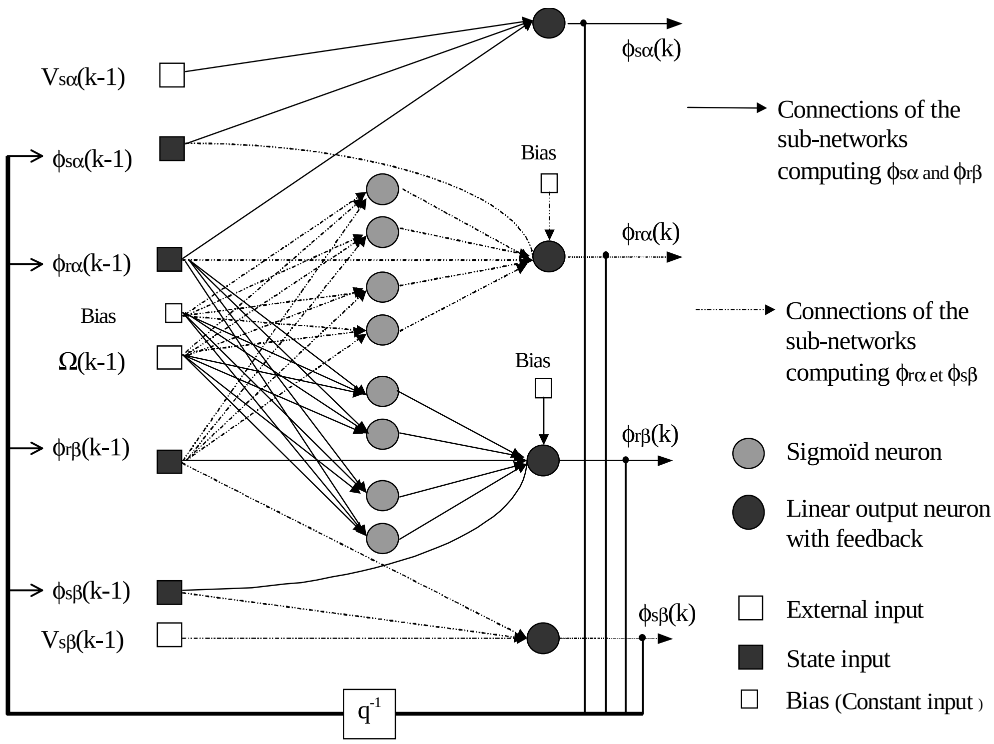

Fig. 1: The feedback neural network for the estimation of the fluxes.

According to the reference model, the two stator fluxes are estimated by two linear functions (linear neuron); the two rotor fluxes are estimated by two non linear subnetworks. These subnetworks have only the inputs variables involved in the corresponding differential equations, one layer of 4 hidden neurons, and a linear output neuron (Fig. 1). The inputs intervening in the non-linearity are linked to the hidden neuron (the non-linearity is the multiplication between the rotor flux components and the speed (1)).

3.- DESIGN OF THE TRAINING SEQUENCES

The estimation of the parameters of the linear subnetworks being straightforward, we have focused on the design of excitating training sequences for the nonlinear subnetworks, i. e. those estimating the rotor fluxes. To facilitate the training, all the variables (voltage, flux, speed) are set to the same scale (Fig. 2, Fig. 4, Fig. 6, Fig. 7). Our goal was to explore the operating space within the shortest possible time, in order to facilitate the training, the latter is achieved using a quasi-newtonian algorithm, the gradient of the cost-function being computed with the backpropagation algorithm [4] [5]. We constrained the training sequence to have a duration of at most 1s, i.e. 100 000 time periods (T=10 µs).

3.1.- Steady-state training and testing at nominal flux

The mechanical dynamics being extremely slow as compared to the electromagnetical ones, it seems justified to train the neural network on steady-state regimes only, a steady-state corresponding to constant values of the rotor speed and of the electromagnetic torque. We first tested this assumption at nominal flux, the motor working off load and the nominal flux module being maintained by a vector control. The problem was then to determine how many different rotor speed values were necessary to describe the operating range. The nominal rotor speed is denoted by Ω_nom. We built three different training sequences:

- SEQUENCE 1 consists of 2 steady-states at 0.2Ω_nom and 1.8Ω_nom;
- SEQUENCE 2 consists of 3 steady-states at 0.2Ω_nom, Ω_nom, and 1.8Ω_nom;
- SEQUENCE 3 consists of 5 steady-states at 0.2Ωn_nom, 0.6Ω_nom, Ω_nom, 1.4Ω_nom and 1.8Ω_nom.

The sequence 2 is shown on Fig. 2.

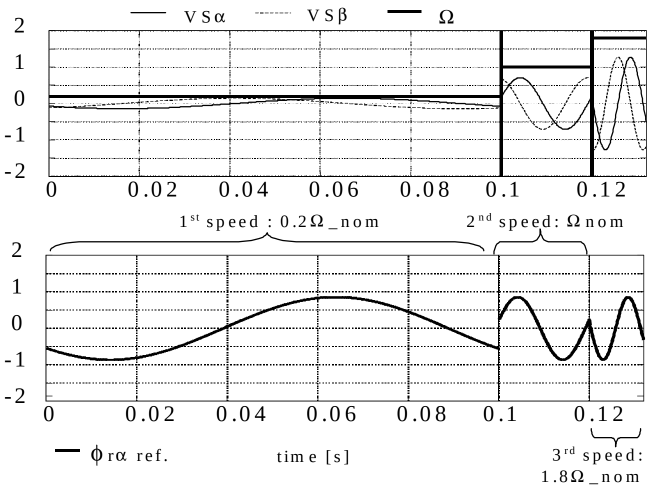

Fig. 2: Example of training sequence: SEQUENCE 2

We performed 3 trainings on the 3 different training sequences, thus obtaining 3 different neural networks, whose performances where evaluated on the whole range of the rotor speed (0Ω_nom-2Ω_nom), i.e. on steady-states which were not learnt. As a matter of fact, the performances are very similar, and entirely satisfactory.

3.2.- Test on steady states at half nominal flux

We then tested the networks on sequences corresponding also to steady-states of the speed, but at half nominal flux (not learnt), in the range (0Ω_nom-2Ω_nom) of the rotor speed. The flux module is maintained at half its nominal value by vector control. The 3 networks still performed similarly, but their error increased quasi linearly with the rotor speed (Fig. 3). A closer observation of the results led to the conclusion that the inaccuracy was mainly due to an error on the flux phase, the error on the flux module being negligible (only of 2 per thousand).

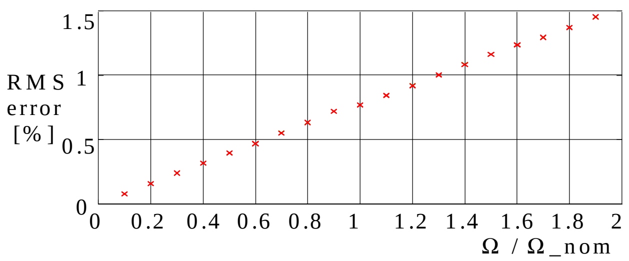

Fig. 3: RMS error on the rotor flux φrα for the test on steady states at half-nominal flux

This error on the flux phase can be explained by the electromechanical equations in steady-state:

$$Cem = K.\phi_s.\phi_r.\sin(\gamma) \cong K.\phi_s.\phi_r.\gamma = f.\Omega \qquad (2)$$

K: constant
f : viscous friction coefficient
γ :flux phase (i.e. phase difference between φs and φr)

Equation. (2) shows that the flux phase γ varies quasi linearly with the rotor speed Ω. On the other hand, the flux phase at half nominal flux is always larger than that a nominal flux. Since the neural model was trained at nominal flux only, it is unable to reproduce a larger flux phase. Due to the linear dependency between flux phase and rotor speed, the larger the speed, the larger the error made by the neural model on the flux phase. We draw the conclusion that the flux phase should vary in the training sequences.

3.3.- Final training sequence

We have seen that: (i) it is sufficient to train the network on a few steady-states of the rotor speed (section 3.1); (ii) the accuracy of the neural model prediction of the flux module at half nominal flux is excellent, even if the network is trained only at nominal flux module (section 3.2); (iii) as shown by Equation (2), the flux phase varies when the electromagnetic torque varies. The final training sequence is thus chosen to consist of 6 subsequences (with a different rotor speed for each : 0.1Ω_nom, 0.5Ω_nom, Ω_nom and their opposite), with for each of them:
The rotor speed is constant
The rotor flux module is constant and nominal
The electromagnetic torque varies linearly with the time between –1.2Tem_nom and +1.2Tem_nom.

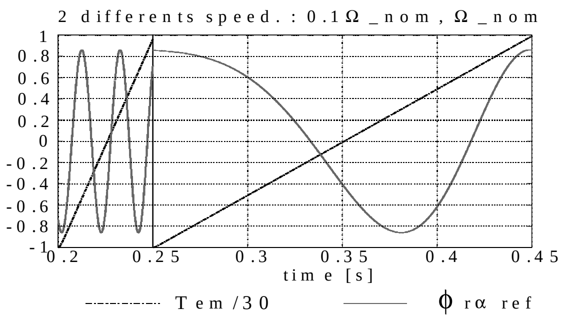

Fig. 4: A part of the final training sequence

A part of the training sequence is shown on Fig. 4. After training the neural model achieves a root mean square (RMS) of 0.03% on this training sequence.

4.- PERFORMANCE OF THE NEURAL MODEL

4.1.- Test in the torque/speed space

The following tests are performed on all the 4 outputs of the neural model obtained after training on the training sequence presented in section 3.3. In order to validate this model, a specific test sequence exploring the whole operating range of the induction machine was built. This sequence consists of a series of subsequences, each one corresponding to specific steady-state values of the rotor speed and of the electromagnetical torque, as shown on Fig. 5. The grid spans the whole range of interest [-2Ω_nom, 2Ω_nom] for the speed, and [-32Tem_nom, 32Tem_nom] for the torque. The extreme values of the torque are unrealistic, but have been tested in order to highlight the good precision of the model on a large operating range.

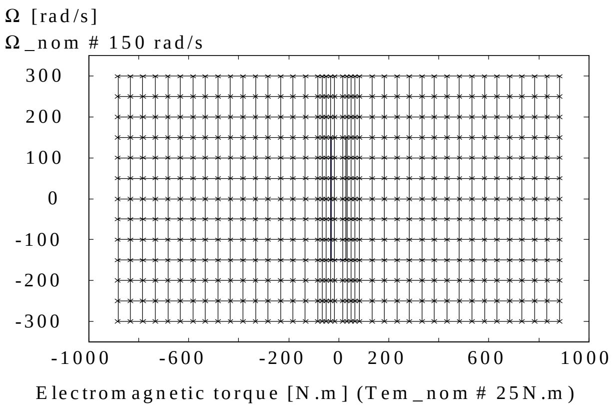

Fig. 5 : Test grid in the torque/speed space

This sequence has been constructed to test a maximum of operating point in a minimum time of simulation. In order to provide a good estimate of the performance, the test sequence had to include not only steady-state regimes, but also transient regimes. We thus followed a specific strategy to describe the above grid, strategy which is illustrated in Fig. 6. This figure represents the evolution of the setpoint torque and speed during a part of the test sequence.

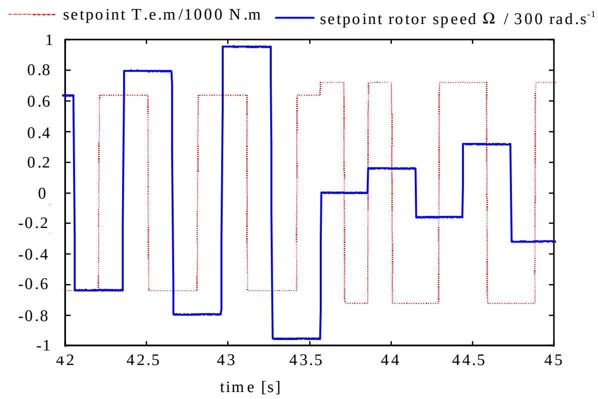

Fig. 6 : Evolution of the setpoint variables on a part of the test sequence.

For one absolute value of the setpoint torque, we test all the values of the speed in the range [-2Ω_nom, 2Ω_nom]. The torque is then increased, etc. This strategy allows to visit each point of the grid (and thus each steady-state), and also to explore many different transients. In order to follow these setpoints, the control of the induction machine is performed in the following way. In order to vary the rotor speed, a load torque is applied according to: Tl = - k * (Ω_setpoint - Ω). The value of the gain k is chosen large enough so that steps of the speed can be obtained. Once the speed setpoint value is reached, we maintain the speed value by fixing an ideal load torque given by: Tl = -Tem + fΩ. We vary the electromagnetic torque using a vector control. Setpoint torque and speed are then maintained constant as long as necessary for the machine to reach the steady-state. The control of the load torque is of course physically unrealistic, but it is used here in order to shorten the test sequence, and to include very "hard" transient regimes.

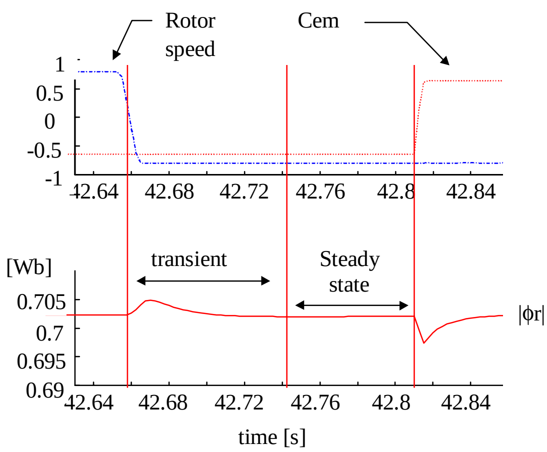

Fig. 7 : Evolution of the module of the rotor flux during a transient and the corresponding steady-state.

Fig. 7 shows the evolution of the module of the rotor flux during a transient regime and the following steady-state, for a given step of speed and electromagnetic torque. For each couple of (torque, speed) values, we compute (i) the RMS errors between the outputs of the neural model and the reference model fluxes on the transient regime (i.e. the error during the step of a setpoint speed or torque), and (ii) the corresponding RMS steady-state errors. These RMS errors are shown on Fig. 8 (steady-state) and Fig. 9 (transients). In the domain spanned by the training sequence, i.e. [-Ω_nom, Ω_nom] x [1.2Tem_nom, 1.2Tem_nom], the RMS error stays below 0.3% on both steady-states and transients; this RMS error is limited to 1.5% in the whole test domain.

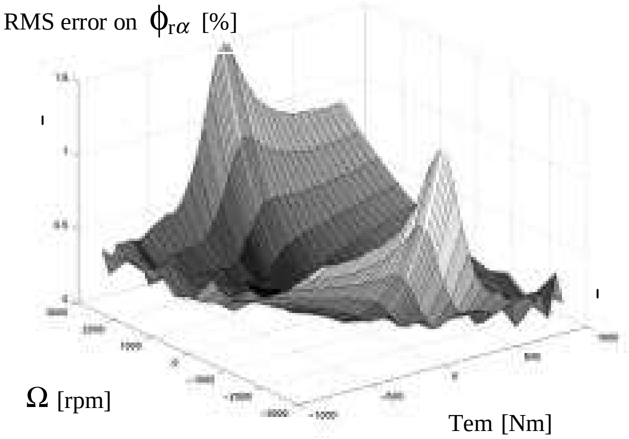

Fig. 8: Steady-state RMS error on the rotor flux in the alpha axis.

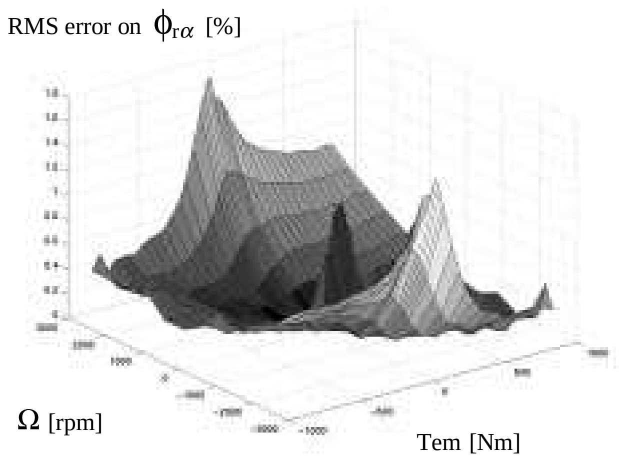

Fig. 9 : Transient RMS error on the rotor flux in the alpha axis.

4.2.- Test on a usual input sequence for the induction machine

We designed a second test sequence, which corresponds to some of the most frequent events happening during the speed control of an induction machine: a machine start-up, different steps of load torque, and a setpoint speed inversion, as shown on Fig. 10.

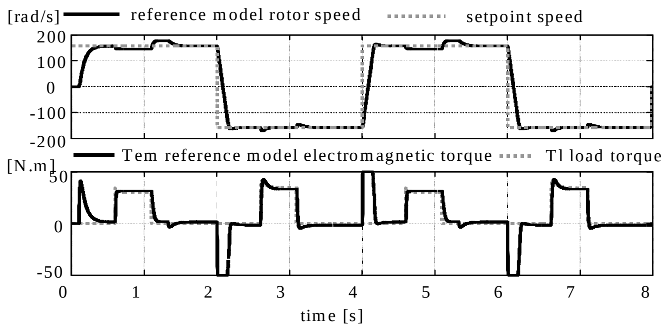

Fig. 10 : Evolution of the setpoints (speed, torque) and of the main variables on the test sequence.

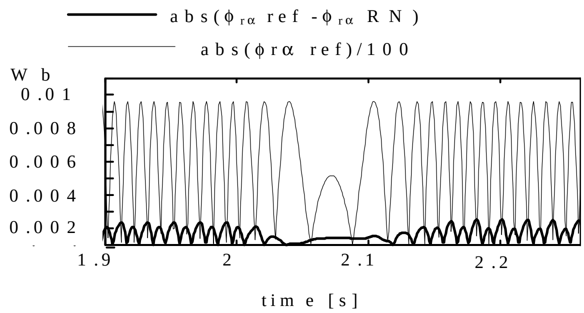

Fig. 11 : Error of the neural model on the rotor flux in the alpha axis during a speed inversion.

As shown on Fig. 11, the errors are below 0.3% of the reference model variables, and the average is around 0.1%.

5.- CONCLUSION

The results on the two test sequences give good estimates of the high accuracy of our neural model of the electromagnetic part of the induction machine. We could obtain this accuracy thanks to the thorough exploration of the operating domain during the model training. We have thus demonstrated the feasibility of the accurate modeling of an induction machine using a feedback neural network. Future work will concern the modeling of an induction machine with magnetic saturation, i.e. including additionnal non-linearities: this will lead us to fully explore the potentials of neural networks for highly nonlinear modeling.

References

[1] R. Ruelland, L. Constant, B. Dagues, 1998, *Modélisation par réseaux de neurones de dispositifs électriques*, Rapport du Diplôme d'Etudes Approfondies en Génie électrique, Institut National Polytechnique de Toulouse, France.

[2] S. Ben Saoud, B.Dagues, H. Schneider, M. Metz, JC. Hapiot, 1996, *Real time emulator of static converters / electrical machines- Application to the test of control unit*, ISIE' 96, 17-20 June, 1996, WARSAW, Poland

[3] K. Hornik, M. Stinchcombe, H.White, 1989, *Multilayer feedforward networks are universal approximators*, Neural Networks **2**, pp. 359-366.

[4] I. Rivals, L. Personnaz., 1996, *Black-box modeling with state-space neural networks*, in: Neural Adaptive Control Technology , R. Zbikowski and K.J. Hunt (eds.), (World Scientific, 1996).

[5] O. Nerrand, P. Roussel-Ragot, L. Personnaz, G. Dreyfus, 1993, *Neural networks and nonlinear adaptive filtering: unifying concepts and new algorithms*, Neural Computation **5**, pp. 165-199.

[6] L. Constant, P. Lagarrigues, B. Dagues, I. Rivals, L. Personnaz, 1998, *Neural modeling of an induction machine*, in Proceedings of System Modeling Control, Zakopan, Poland

[7] L. Constant, P. Lagarrigues, B. Dagues, I. Rivals, L. Personnaz, 1999, *Modeling of Electromechanical System using Feedback Neural Network*, in Computational Intelligence and Applications, P.S. Szczepaniak eds, XIV, 369 pp (Studies in Fuzziness and Soft Computing, Vol. 23) ISBN 3-7908-1161-0 Physica-Verlag, c/o Springer-Verlag