Proc. of the 5th WSEAS/IASME Int. Conf. on Electric Power Systems, High Voltages, Electric Machines, Tenerife, Spain, December 16-18, 2005 (pp104-109)

On Line Neurofuzzy Modeling for Permanent Magnet Synchronous Machines

EDMARY ALTAMIRANDA, ESSAM HAMDI
Department of Energy and Environment
Division of Electric Power Engineering
Electrical Machines and Drive Systems Group
Chalmers University of Technology
Gothenburg-Sweden

Abstract:-This paper presents a neurofuzzy structure for on line modeling of Permanent Magnet Synchronous Machines (PMSM). The model structure is based on recurrent fuzzy neurons (RFN) presented in [1,2], which are used to synthesize a single layer RFN network for nonlinear discrete state space representation of the PMSM dynamics in the d,q rotating reference frame. The proposed scheme allows obtaining on line, a time varying nonlinear model with an appropriate structure for linearizing control laws design. The efficiency of the neurofuzzy structure for PMSM modeling is illustrated by computer simulations.

Key-Words: Fuzzy Systems, Neural Dynamics, Nonlinear Models, Permanent Magnet Synchronous Machines.

1 Introduction

Permanent Magnet Synchronous Machines are often used in variable speed and low power applications such as servomotors and special-purpose alternators. Their main advantages over conventional synchronous machines is the absence of excitation windings [3,4]. PMSM dynamics involve highly nonlinear behaviors. One main source of nonlinearities is the windings inductances. What complicates matters further is that the saturation levels (and nonlinearities) are function of the mechanical load. These nonlinearities and uncertain variations in inductances make it difficult to synthesize an appropriate nonlinear control task. Therefore, the control laws for such a kind of machine are usually designed in terms of simplified models that do not take into account the complete system behavior. Some contributions on identification and analytical modeling of PMSM inductances may be found in [4,5]. In [4] it is also presented a torque control technique which considers the inductance harmonics identification to compensate torque pulsations in closed loop.

Neural Networks and Fuzzy Logic have been widely used for nonlinear modeling and control purposes. Conventional feed-forward multilayered neural networks facilitate nonlinear mapping from an input space to an output space. These network systems involve a training phase and their ability to synthesize complex nonlinear maps may vary with the number of layers and neural elements in each layer. Most of the times, it is difficult to know in advance the exact number of neural elements necessary and sufficient to achieve an adequate mapping. Since all the initial weights are randomly assigned and the error weight space might have local minima, there may be a significant learning error even after a long learning period. Referring to modeling of dynamic systems, a neural structure with recurrent terms may prove to be a better alternative than a feed-forward neural structure. Even more, for some problems a small feedback system is equivalent to a big, and possibly infinitely large, feed forward system [6]. Some other results that refer to modeling and control of dynamic systems using recurrent neural networks are reported in [7,8]. In the present work, a single layer network of recurrent fuzzy neurons [1,2] has been used to achieve on line modeling of permanent magnet synchronous machines (PMSM) in the d,q rotating reference frame. The RFN structure possesses fuzzy synapses and recurrent connections that make it a good candidate for on line modeling of complex nonlinear systems. The paper is organized as follows. In Section 2 the RFN based modeling mathematical formulations are presented, in Section 3 it is exposed the application of the RFN technique for PMSM on line modeling, Section 4 presents simulation results and Section 5 gives some concluding remarks.

2 The Recurrent Fuzzy Neuron

The Recurrent Fuzzy Neuron (RFN) is a structure with fuzzy synapses and recurrent connections which provide dynamic characteristics to the fuzzy neuron, making it a good candidate for on line modeling of nonlinear systems. The structure of the RFN is shown in Figure 1 [1,2].

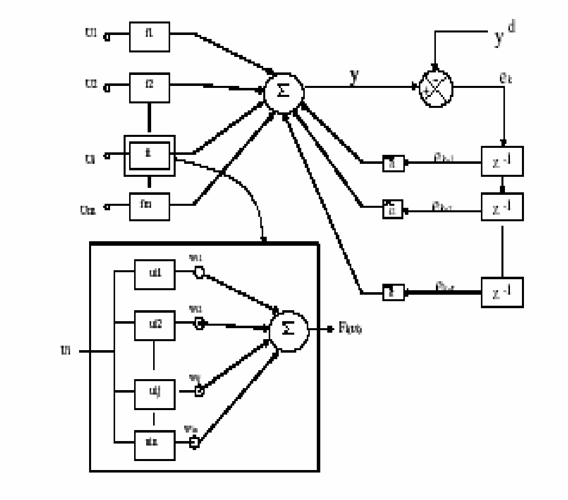

Fig.1 The Recurrent Fuzzy Neuron

2.1 RFN Neural Networks for On Line Modeling of Nonlinear Systems.

The RFN presented in [1,2] may be used to model a variety of nonlinear SISO and MISO systems using only one neuron for input/output representation. The feed-forward and recurrent connections f_i and $\tilde{f}_r$, respectively, possess, nonlinear synaptic weights which are determined by fuzzy IF-THEN rules. The synapse output is obtained by fuzzy inference with defuzzification and therefore, the output of this RFN may be represented by

$$y(k) = \sum_{i=1}^{m} f_i(u_i(k)) + \sum_{r=1}^{l} \tilde{f}_r(e(k-r)) \qquad (1)$$

where u_i, $i=1..m$ corresponds to the *i-th* feed-forward neural input at the time *k*, *e(k-r) r=1..l*, is a recurrent term associated with the neuron output error at time *k-r* and *l* must be, at most, equal to the estimated system order. The nonlinear synapses are defined as follows:

$$f_i(u_i(k)) = \frac{\sum_{j=1}^{m_i} \mu_{ij}(u_i(k)) w_{ij}(k)}{\sum_{j=1}^{m_i} \mu_{ij}(u_i(k))}$$

$$\tilde{f}_r = \frac{\sum_{j=1}^{l_r} \tilde{\mu}_{rj}(e(k-r)) \tilde{w}_{rj}(k)}{\sum_{j=1}^{l_r} \tilde{\mu}_{rj}(e(k-r))} \qquad (2)$$

where μ_{ij} and $\tilde{\mu}_{rj}$ correspond to the fuzzy sets that characterize u_i and $e(k-r)$ respectively , $w_{ij}(k)$ and $\tilde{w}_{rj}(k)$ are the consequents singleton weights associated to the nonlinear synapses IF THEN rules. In [1,2] it is proved that only one RFN is enough to model complex nonlinear systems with only one output, therefore a single layer RFN neural network will be able to synthesize nonlinear models for multiple output complex systems.

2.1.1 RFN based State Vector Model

In order to achieve on line RFN based modeling using a discrete state space representation, it is necessary to synthesize single layer RFN neural networks as follow:

$$\tilde{x}(k+1) = N_1(x(k), u(k), e(k))$$
$$\tilde{y}(k) = N_2(x(k), u(k), e^y(k)) \qquad (3)$$

Where *u(k), x(k), y(k), e(k)* and $e^y(k)$ are discrete time sequences, $x(k) = [x_1(k), x_2(k), \ldots, x_n(k)]^T$ is the state vector, $u(k) = [u_1(k), u_2(k), \ldots, u_m(k)]^T$ the input vector, $y(k) = [y_1(k), y_2(k), \ldots, y_s(k)]^T$ the output vector, $e(k)=[e_1(k), e_2(k), \ldots, e_n(k)]^T$ the state error vector and $e^y(k)=[e^y_1(k), e^y_2(k), \ldots, e^y_s(k)]^T$ the output error vector. This definition is valid if and only if *x(k)* is completely accessible. Hence equations (3) represent a *m*-input *s*-output nonlinear system of order *n* and it may be synthesized using single layer RFN neural networks for N_1 *and* N_2.
For N_1 synthesis:

$$\tilde{x}_i(k+1) = \sum_{l=1}^{n} f_l(x_l(k)) + \sum_{j=1}^{m} \hat{f}_j(u_j(k)) + \tilde{f}_i(e_i(k)) \qquad (4)$$

Where f_l and $\hat{f}_j$ correspond to the nonlinear feed-forward connections and $\tilde{f}_i$ correspond to the nonlinear recurrent connections. N_1 will be a single layer neural network with i=*1...n* neurons.
For N_2 synthesis:

$$\tilde{y}_v(k+1) = \sum_{l=1}^{n} f_l^y(x_l(k)) + \sum_{j=1}^{m} \hat{f}_j^y(u_j(k)) + \tilde{f}_v^y(e_v^y(k)) \qquad (5)$$

Where f_l^y and $\hat{f}_j^y$ correspond to feed-forward connections and $\tilde{f}_v^y$ correspond to recurrent connections respectively. N_2 will be a single layer neural network with *v =1...s* neurons.

2.2 RFN Neural Network Learning Algorithm

The general learning algorithm is an extension of results presented in [1,2]. This is in order to make it applicable for multi-output systems. The training is defined in terms of a steepest descent method, where the weight changes are achieved for a set of input patterns *P*. The error index is given by the average squared error for *P* patterns in the following way:

$$E(k)=\frac{1}{2P}\sum_{v=1}^{S}\sum_{q=1}^{P}e_{vq}^{2}(k)= \frac{1}{2P}\sum_{v=1}^{S}\sum_{q=1}^{P}(y_{vq}(k)-y_{vq}^{d})^{2} \qquad (6)$$

Where $y_{vq}(k)$ is the *v-th* RFN network output, $y_{vq}^{d}(k)$ is the *v-th* desired output, corresponding to pattern *q* at time *k* and $e_{vq}(k)$ is a learning error between the *v-th* RFN network output and the *v-th* desired output at time *k*, *S* is the number of RFN network outputs.

During the learning sessions, the updating rule, for feed-forward connections, is given by:

$$w_{i,j}(k)=w_{i,j}(k+1)+\Delta w_{i,j}(k) \qquad (7)$$

$$\Delta w_{i,j}(k)=-\alpha\sum_{q=1}^{P}e_{q}(k)\mu_{i,j}(u_{q,i}(k)) \qquad (8)$$

For recurrent connections, the updating rule is given as:

$$\Delta\tilde{w}_{r,j}(k)=\Delta_1\tilde{w}_{r,j}+\Delta_2\tilde{w}_{r,j}$$
$$\Delta_1\tilde{w}_{r,j}(k)=-\alpha\sum_{q=1}^{P}e_q(k)(\tilde{\mu}_{r,j}(e_q(k-r))$$
$$\Delta_2\tilde{w}_{r,j}(k)=-\alpha\sum_{q=1}^{P}e_q(k)\left[\frac{\partial y_q(k)}{\partial e_q(k-r)}\right]z_q^r(k) \qquad (9)$$

where the μ_{ij} and $\tilde{\mu}_{rj}$ terms correspond to membership functions associated to weights $w_{ij}(k)$ and $\tilde{w}_{rj}(k)$, respectively.

$$\frac{\partial y_q(k)}{\partial e_q(k-r)}=\tilde{w}_{r,j}(k)\dot{\tilde{\mu}}_{r,j}(e_q(k-r))+ \tilde{w}_{r,j+1}(k)\dot{\tilde{\mu}}_{r,j+1}(e_q(k-r)) \qquad (10)$$

Let $z_q^r(k)$ be defined as:

$$z_q^r=\frac{\partial e_q(k-r)}{\partial\tilde{w}_{r,j}(k)} \qquad (11)$$

In order to provide dynamic characteristics to the updating rule for recurrent connections, equation (11) is determined through the successive application of the chain rule. Without loss of generality, only first order terms are considered and therefore $z_q^r(k)$ may be represented by the following first order time-varying linear system:

$$z_q^r(k)=\frac{\partial e_q(k-r)}{\partial e_q(k-r-1)}z_q^r(k-1)+\frac{\partial e_q(k-r)}{\partial\tilde{w}_{r,j}(k)}$$
$$z_q^r(k)=\left[\tilde{w}_{r,j}(k-1)\dot{\tilde{\mu}}_{r,j}(e_q(k-r-1))+ \tilde{w}_{r,j+1}(k-1)\dot{\tilde{\mu}}_{r,j+1}(e_q(k-r-1))\right]z_q^r(k-1)+ \tilde{\mu}_{r,j}(e_q(k-r)) \qquad (12)$$

$$\Delta\tilde{w}_{r,j}=-\alpha\sum_{q=1}^{p}e_q\left[\mu_{r,j}(e_q(k-r))+\frac{\partial y_q(k)}{\partial e_q(k-r)}z_q^r(k)\right] \qquad (13)$$

3 On Line Neurofuzzy Modeling of PMSM

In Permanent Magnet Synchronous Machines (PMSM) the exciting coil is replaced with permanent magnets. The continuous excitation causes the motor to act as a sub-excited synchronous engine. Due to this characteristic, the permanently excited motor can be easily integrated into already existing systems without limitations, this make this kind of machine attractive for variable speed and low power applications. PMSM may operate in motoring or generating mode. The mode of operation is determined by the sign of the mechanical torque (positive for motoring, negative for generating). The electrical and mechanical parts may be represented by a second order state space model. The model assumes that the flux established by the permanent magnets in the stator is sinusoidal, which implies that the electromotive forces are also sinusoidal [3]. The Electrical System in the d,q rotor reference frame is described by [3]:

$$\frac{d}{dt}i_d=\frac{1}{L_d}v_d-\frac{R}{L_d}i_d+\frac{L_q}{L_d}np.w_r.i_q$$
$$\frac{d}{dt}i_q=\frac{1}{L_q}v_q-\frac{R}{L_q}i_q-\frac{L_d}{L_q}np.w_r.i_d-\frac{\lambda.np.w_r}{L_q} \qquad (14)$$
$$T_e=1.5.np.\left[\lambda.i_q+(L_d-L_q).i_d.i_q\right]$$

Proc. of the 5th WSEAS/IASME Int. Conf. on Electric Power Systems, High Voltages, Electric Machines, Tenerife, Spain, December 16-18, 2005 (pp104-109)

where

L_q, L_d: q and d axis inductances

R: resistance of the stator windings

I_d, I_q : q and d axis currents

v_d, v_q: q and d axis voltages

w_r: rotor angular velocity

λ: amplitude of the flux induced by the permanent magnets of the rotor in the stator phases

np: number of poles pairs

T_e: electromagnetic torque

The mechanical system is described by:

$$\frac{dw_r}{dt} = \frac{1}{J}.(T_e - F.w_r - T_m)$$
$$\frac{d\theta}{dt} = w_r \tag{15}$$

where

J: combined inertia of rotor and load

F: combined viscous friction of rotor and load

θ: rotor angular position

T_m: Mechanical Torque

In order to illustrate the efficiency of the proposed neurofuzzy algorithm for on line modeling of PMSM, the dynamical nonlinear system described above is considered for computer simulations.

The nonlinearities associated to PMSM make it difficult to synthesize appropriate nonlinear control tasks. This is because windings' inductances and saturation levels are function of the mechanical load. Therefore, the control laws for such a kind of machine are usually designed in terms of simplified models that do not take into account the complete system behavior. In order to deal with such difficulties it is proposed a time varying discrete state space model based on RFN neural networks which enables achieving on line modeling of the PMSM using current, voltage and speed measurements.

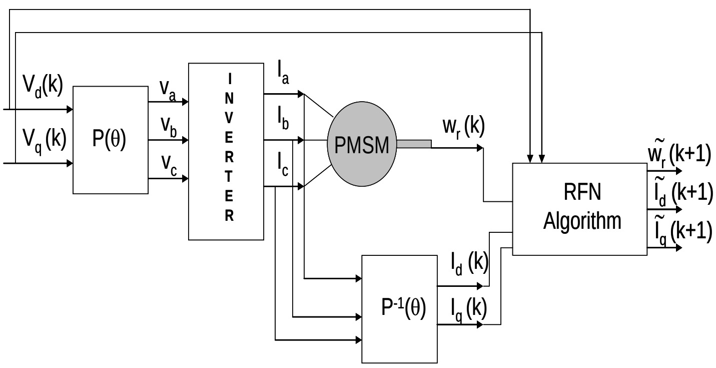

Fig. 2 Open Loop On Line RFN Modeling Scheme

Fig. 2 illustrates the proposed scheme where $P(\theta)$ is the Park transformation [4]:

$$P(\theta) = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{16}$$

The presented open-loop modeling scheme (Fig. 2) employs a RFN algorithm. This neurofuzzy algorithm, however, may also be used in a closed loop scheme for nonlinear control purposes. Here, it has been only considered the estimation of currents and speed because position can be obtained by the speed integration.

According to equations (3) and the previous knowledge about the system, the discrete RFN nonlinear model for PMSM may be defined as follow:

$$\tilde{x}(k+1) = N_1(x(k), u(k), e(k))$$
$$\tilde{y}(k) = [x_1(k)\; x_2(k)\; x_3(k)]^T \tag{17}$$

where $x(k) = [x_1(k),x_2(k),x_3(k)]^T=[I_d(k),I_q(k),w_r(k)]^T$ is the state vector, $u(k) = [u_1(k),u_2(k)]=[v_d(k),v_q(k)]^T$ the input vector, $y(k) = [y_1(k),y_2(k),y_3(k)]^T= [x_1(k),x_2(k),x_3(k)]^T$ the output vector, $e(k)=[e_1(k), e_2(k),e_3(k)]^T$ the state error vector. In this case it is assumed that it is possible to measure, the currents and speed therefore the output vector has been linearly defined.

In order to obtain a nonlinear model that may be attractive for linearizing control laws synthesis and taking into account the information of the continuous model equations (14) and (15), equation (17) may be redefined in the following way:

$$\begin{pmatrix} \tilde{x}_1(k+1) \\ \tilde{x}_2(k+1) \\ \tilde{x}_3(k+1) \end{pmatrix} = N_1(x(k),e(k)) + \begin{pmatrix} \frac{T}{L_{d0}} & 0 \\ 0 & \frac{T}{L_{q0}} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_1(k) \\ u_2(k) \end{pmatrix} \tag{18}$$

$$\tilde{y}(k+1) = [x_1(k)\; x_2(k)\; x_3(k)]^T$$

Where T is the sample time and L_{d0} and L_{q0} are the q and d axis inductances fundamental terms.

3 Simulation Results

The open loop on line modeling scheme was simulated using Matlab and Simulink. The continuous system was simulated with a fixed step size *T=0.0001 sec*. The RFN algorithm takes samples from the continuous system each step to achieve the modeling task. In order to provide a more realistic behavior to the continuous system and also to verify the efficiency of the RFN algorithm, *q* and *d* axis harmonic inductances of six order were included [4]. The machine parameters used for this simulation are obtained from [4] as follows:

$R = 1\,\Omega;\ \lambda = 0.075\ Wb;\ np = 3;\ J = 0.0343\ Kg.m^2;$

$F = 0.4777\ N.s;\ Tm = 1.5\ N.m$

$L_d = L_{d0} + L_{d6}\cos(6\theta);\ L_q = L_{q0} + L_{q6}\cos(6\theta)$

$L_{d0} = 0.00393\ H;\ L_{d6} = -0.000272\ H$

$L_{q0} = 0.00686\ H;\ L_{q6} = 0.000492\ H$

The RFN algorithm requires the definition of fuzzy sets for each variable according to the possible machine operational ranges, these fuzzy sets must be complementary triangular functions [1,2] and are shown in Fig. 3 and Fig. 4.

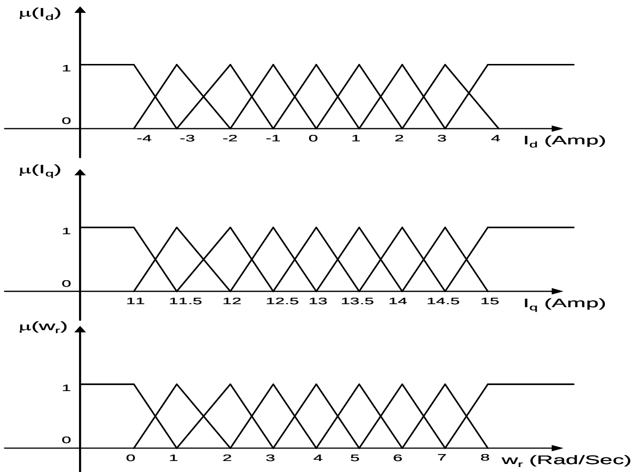

Fig. 3 Fuzzy Sets for State Space Variables

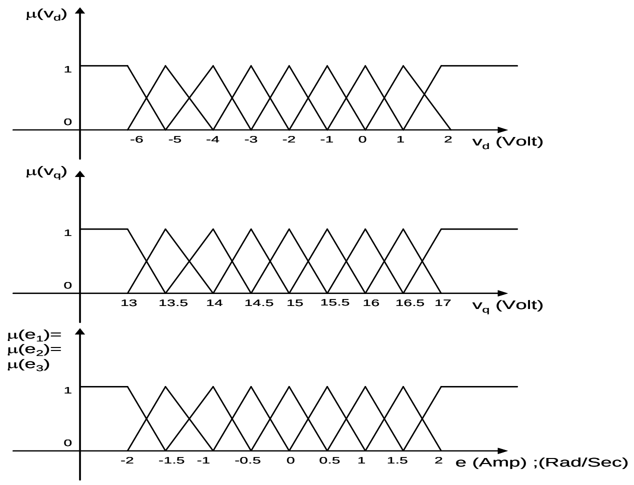

Fig. 4 Fuzzy Sets for Input Variables and Errors

The open loop modeling was achieved using both sinusoidal and step input voltages. The RFN structure has 3 recurrent neurons, the learning rate that provided the best results was α=0.3 and the number of patterns per time instant *P=1*.

3.1 Simulation Case 1: Sinusoidal Input Voltages

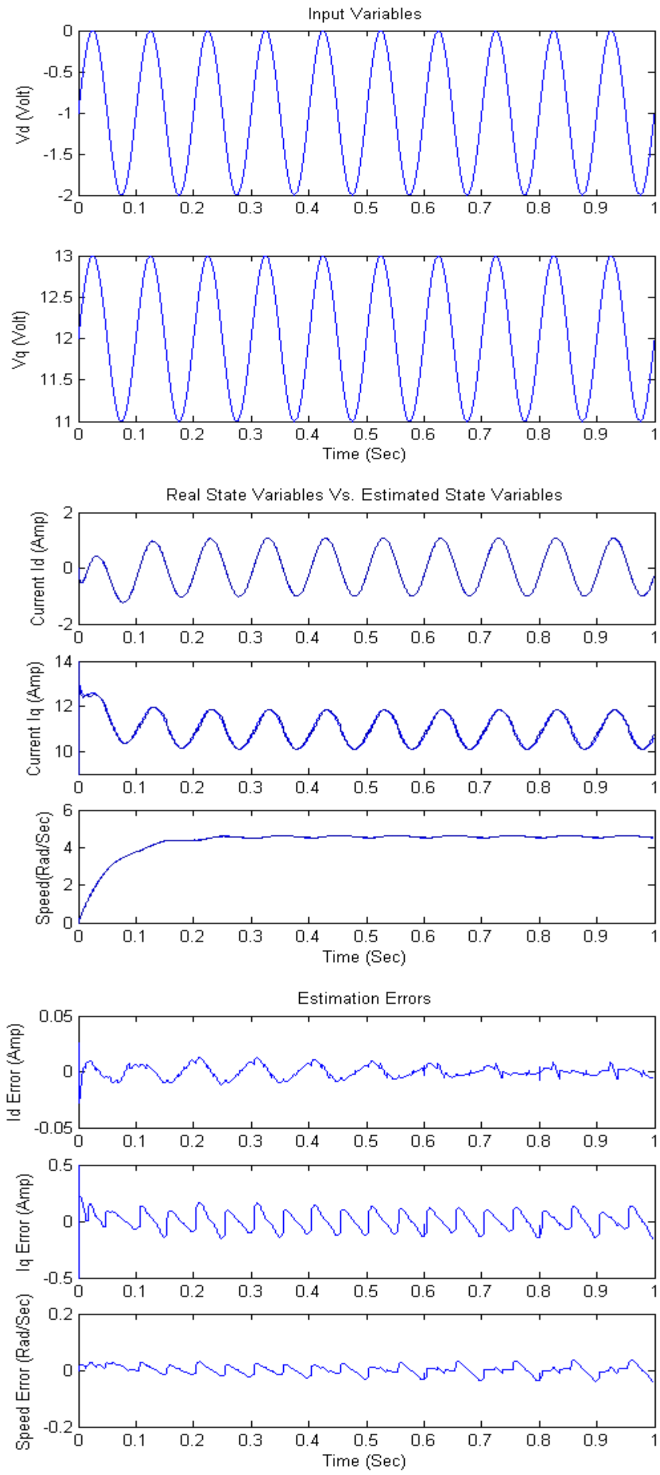

Fig.5 On Line Modeling Simulation Results for Sinusoidal Input Voltages

3.2 Simulation Case 2: Step Change Input Voltages

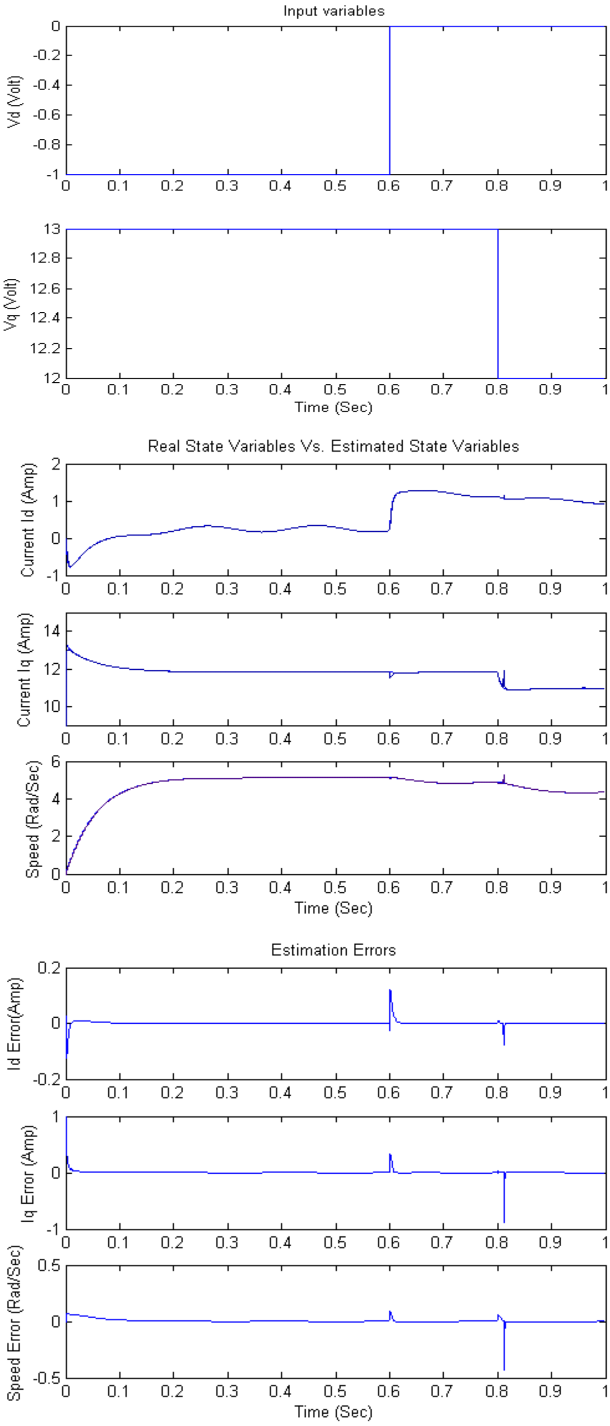

Fig.6 On Line Modeling Simulation Results for Step Change Input Voltages

For this simulation case it is observed that estimation errors get bigger during the step changes due to the high rate of such changes. However, the RFN algorithm corrects the deviations, successfully achieving the on line modeling task.

4 Conclusions

A neurofuzzy scheme, based on recurrent fuzzy neurons [1,2], for on line modeling of Permanent Magnet Synchronous Machines (PMSM) has been presented. The proposed scheme allows obtaining on line, a time varying nonlinear model with a discrete state space representation in the d,q rotating reference frame, with an appropriate structure for linearizing control laws design. The efficiency of the neurofuzzy scheme for PMSM modeling was illustrated by computer simulations.

5 Acknowledgments

The authors wish to thank Swedish Institute (SI.) for supporting this research by providing the visiting fellowship of Dr. Altamiranda.

References

[1] E. Altamiranda and E. Colina, A Recurrent Fuzzy Neuron for On Line Modeling of Non-linear Systems, I5th Triennial IFAC World Conference Barcelona, Spain, July, 2002.

[2] E. Altamiranda and E. Colina, A Recurrent Fuzzy Neuron with Evolutive Learning Rate for On Line Modeling of Nonlinear Systems WSEAS Transactions on Systems, Issue 8, Vol. 3, October 2004, pp. 2611-2617.

[3] M. Krcum, A. Gudelj and Z. Juric, Dynamic Simulation of Permanent Magnet SynchronousMachine, IEEE MELECON, Dubrovnik,Croatia, May, 2004, pp. 1117-1120.

[4] A. Madani, J.P Barbot, F. Colartino and C.Marchand, Reduction of torque pulsations by Harmonics identification of a Permanent Magnet Synchronous Machine, In Proc. of the 4th IEEE Conference on Control Applications, September, 1995, pp.787-792.

[5] A. Bogdan, A. Keyhani, A. EL-Antably, W. Lu and M. Dai, Analytical Model for Permanent Magnets Motors with Surface Mounted Magnets, IEEE Transactions on Energy Conversion, Vol. 18, No. 3, September, 2003.

[6] R. Williams and D. Zisper, A learning algorithm for continually running fully recurrent neural networks, Neural computation, Vol. 2, 1989, pp. 4-27.

[7] K. S. Narendra and K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Trans. Neural Networks, Vol. 2, No. 2, 1990, pp. 4-27.

[8] A. Parlos, A. Atiya and K. Chang, Recurrent multilayer perceptron for nonlinear systems identification, Proc. IJCNN II, 1991, pp.537-540.