

# A Novel Neuroglial Architecture for Modelling Singular Perturbation System

Samia Salah, M'hamed Hadj Sadok, Abderrezak Guessoum

Department of electronics, University Saad Dahlab Blida1, Algeria

---

## Article Info

### Article history:

Received Jan 23, 2018

Revised Jul 5, 2018

Accepted Jul 29, 2018

---

### Keyword:

Artificial neuro glial network  
Asynchronous machine  
Gerschgorin's circle theorem  
Glial network  
Singular perturbation method

---

## ABSTRACT

This work develops a new modular architecture that emulates a recently-discovered biological paradigm. It originates from the human brain where the information flows along two different pathways and is processed along two time scales: one is a fast neural network (NN) and the other is a slow network called the glial network (GN). It was found that the neural network is powered and controlled by the glial network. Based on our biological knowledge of glial cells and the powerful concept of modularity, a novel approach called artificial neuroglial Network (ANGN) was designed and an algorithm based on different concepts of modularity was also developed. The implementation is based on the notion of multi-time scale systems. Validation is performed through an asynchronous machine (ASM) modeled in the standard singularly perturbed form. We apply the geometrical approach, based on Gerschgorin's circle theorem (GCT), to separate the fast and slow variables, as well as the singular perturbation method (SPM) to determine the reduced models. This new architecture makes it possible to obtain smaller networks with less complexity and better performance.

*Copyright © 2018 Institute of Advanced Engineering and Science.  
All rights reserved.*

---

## Corresponding Author:

Samia Salah,  
Department of Electronics,  
University Saad Dahlab Blida1,  
Soumaa Road, BP 270, Blida, Algeria, (+213) 25 43 38 50  
Email : salah\_samia@yahoo.fr

---

## 1. INTRODUCTION

The last few years have witnessed a tremendous growth in the field of intelligent systems. Inspired by biological neural networks, one such success has been achieved in evolution of artificial neural networks (ANNs). ANNs are characterized by their distinctive capabilities of exhibiting massive parallelism, generalization ability and being good function approximators. This renders them useful for solving a variety of problems in pattern recognition, prediction, optimization and associative memory [1], [2]. Additionally, they are also being employed in system modeling and control [3], [4].

These ANNs, efficient in numerous applications, are not as well suited for approximating non-linear and high-dimensional functions with multiple time dynamics like the ones in singular perturbation systems (SPSs) which increases the difficulties in system modeling, analysis and controller design. An effective way to overcome this problem is to separate the original system states into subsystems that change rapidly and those that vary slowly on the chosen time scale, using singularly perturbation method (SPM).

Some recent research results using the (SPM) to analyze and control the SPSs are published in [5], [6]. However, accurate and faithful mathematical models for those systems are usually difficult to obtain due to the uncertainties and nonlinearities. In this case, adequate system identification becomes important and necessary, before a singular perturbation theory-based control scheme can be designed.

Recently, research using multi times scale neural networks have been proposed in literature to solve the system identification problem of the nonlinear SPSs. Among them there are multi-time-scale dynamic

neural network (DNN) proposed in [7]. Or recurrent neural network (RNN) proposed in [8]. In these papers, training methods are based on a gradient descent updating algorithm with fixed "learning gain", such as back propagation (BP) and RNN algorithms. The main drawback of these training methods is that the convergence speed is usually very slow. To accelerate the training process, researchers investigated the extended Kalman filter (EKF) based training methods for NN in [9]. The theoretical analysis of EKF based training algorithm requires the modeling uncertainty of the NN to be a Gaussian process, which may not be true in real applications. Some other researchers also studied optimal bounded ellipsoid (OBE) algorithm-based learning laws for NN [10]-[12]. All of these methods are complex and computationally intensive.

In this paper, we propose a new multi time-scale NN architecture called "artificial neuroglial network" (ANGN) based on the powerful concept of "modularity" to solve the problems of singular perturbation system training. The basic idea is to use the knowledge about the nervous system and the human brain, where the information flows along two different pathways and is processed along two-time scales: one is a fast-neural network (NN) and the other is a slow network called the glial network (GN). It was found that the neural network is powered and controlled by the glial network [13], [14].

In our experiment, for a given application, depending on the complexity and the physical characteristics of the problem, we divide our global model into two sub-models: slow and fast ones using the singular perturbation method (SPM). The first difficulty that arises when decoupling variables is the identification of both the slow and fast model variables. The solution is based on Gerschgorin's circle geometric theorem (GCT) [15]. This technique makes it possible to locate the eigenvalues in the complex plane within groups of circles. The grouping of the modes is immediate whenever circles are disjoint, and afterward, the number of slow and fast modes is determined.

Validation of the proposed approach is carried out on the ASM model, under the singularly perturbed standard form. Subsequently, an algorithm is adopted to test the effectiveness and performance of the proposed ANGN. This new architecture has made it possible to obtain networks of considerably smaller size with simple structures which have a strong nonlinear approximation capability and which enables it to model nonlinear singularly perturbed systems more accurately with less computation complexity, compared to the conventional neural network model.

## 2. SINGULAR PERTURBATION METHOD

This method is used for multi-time scale systems that can be reduced to the standard form of equation (3) by the determination of the parasitic term  $\varepsilon$ . Consider the state model of a linear system of dimension n:

$$\begin{cases} \dot{X} = AX + BU \\ Y = CX \end{cases} \quad (1)$$

Evolving according to two-time scales, it can be decoupled into two slow and fast subsystems. The state vector X contains all the state variables corresponding to the dynamic elements. If x is the set of state variables of slow elements, and z is the set of fast elements the model is written

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U, \quad (2)$$

$$y = [C_1 \quad C_2] \begin{bmatrix} x \\ z \end{bmatrix}$$

with :

$$-x(t_0) = x_0, \text{ and } z(t_0) = z_0$$

$A_{21}, A_{22}, B_2$  - are very large compared to  $A_{11}, A_{12}, B_1$ .

We introduce the parameter  $\varepsilon$  to normalize our model, we write:  $A_{21}^* = \varepsilon A_{21}$ ,  $A_{22}^* = \varepsilon A_{22}$ ,  $B_2^* = \varepsilon B_2$ .  $\varepsilon$  can

be given by:  $\varepsilon = \frac{1}{\|A_{22}^{-1}\|(\|A_0\| + \|A_{12}\| \cdot \|L_0\|)}$  with :  $L_0 = -A_{22}^{-1} \cdot A_{21}$  and  $A_0 = A_{11} + A_{12}^{-1} \cdot L_0$

By assuming that matrix  $A_{22}$  is invertible, the state equation in the standard singularly perturbed form with  $\varepsilon$  as the perturbation parameter is then written as:

$$\begin{bmatrix} \dot{x} \\ \varepsilon \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21}^* & A_{22}^* \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2^* \end{bmatrix} U, \quad (3)$$

### 2.1. Slow and fast reduced models

The slow reduced model is determined from eq. (3) by considering that  $\varepsilon = 0$ :

$$\begin{cases} \dot{x}_s = A_s x_s + B_s u_s \\ y_s = C_s x_s + D_s u_s \\ z_s = -A_{22}^{-1}(A_{21} x_s + B_2 u_s) \end{cases} \quad (4)$$

where  $x_s, z_s, u_s, y_s$  are the slow components of the variables  $x, z, u, y$  respectively, with:

$$\begin{aligned} A_s &= A_{11} - A_{12} A_{22}^{-1} A_{21} \\ B_s &= B_1 - A_{12} A_{22}^{-1} B_2 \\ C_s &= C_1 - C_2 A_{22}^{-1} A_{21} \\ D_s &= -C_2 A_{22}^{-1} B_2 \end{aligned} \quad (5)$$

with:  $x_s(t_0) = x_0$ . The initial value of the slow components  $z_s$  is:  $z_s(t_0) = -A_{22}^{-1} A_{21} x_s(t_0)$

which is generally different from  $z_0$ . The fast variables  $z$  cannot therefore be approximated by  $z_s$  in the time interval  $[0, T]$ . We introduce the corrective term  $z_f$ , defined by:  $z_f = z - z_s$  which represents rapid changes in  $z$ . Therefore, the boundary layer equation, expressed in the dilated time  $\tau$  follows:

$$\left( \varepsilon \frac{dz_f}{dt} = \frac{dz_f}{d\tau} \right); \quad (6)$$

The fast reduced model is then written:

$$\begin{cases} \frac{dz_f}{d\tau} = A_{22} z_f(\tau) + B_2 u_f(\tau) \\ y_f = C_2 z_f(\tau) \\ z_f(t_0) = z_0 + A_{22}^{-1} A_{21} x_0 \end{cases} \quad (7)$$

## 3. IDENTIFICATION OF THE GEOMETRIC DYNAMICS

Setting the previous standard form assumes: a) Knowledge of the eigenvalues to determine the size of the slow and fast eigenvectors. b) A suitable grouping of slow modes and fast modes.

Our attention is focused on geometrical methods including the circles of Gerschgorin. The localization of the eigenvalues on the complex plane makes it possible to put a system in the standard form without having to calculate these eigenvalues. In the case of GCT, the grouping of modes is immediate as soon as the eigenvalues are circumscribed in disjoint circles. The geometric method based on GCT for the selection and separation of different time scales is represented as follows:

### 3.1. Gerschgorin's circles theorem (GCT)

Noting  $(a_{ij}, i, j = 1, \dots, n)$  as the elements of the state matrix  $A$ ,  $p_i, Q_i$  are expressed by

$$p_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, 2, \dots, n \tag{8}$$

$$Q_i = \sum_{\substack{i=1 \\ j \neq i}}^n |a_{ij}|, \quad j = 1, 2, \dots, n \tag{9}$$

Geometrical separation of the different dynamic modes is based on the application of the following two theorems. These theorems obtained from Gerschgorin give a localization of the eigenvalues on the complex plane.

**3.1.1. Theorem 1**

All the eigenvalues of a matrix of arbitrary rank n, are contained in n circle bundles centered at  $a_{11}, a_{22}, \dots, a_{nn}$  and radii  $R_{l1}, R_{l2}, \dots, R_{ln}$  for the lines or  $R_{c1}, R_{c2}, \dots, R_{cn}$  for the columns, which are obtained by summing the modules of the off-diagonal terms appearing in the same line or column:

$$R_{li} = p_i \text{ and } R_{ci} = Q_i.$$

**3.1.2. Theorem 2**

When a group of k line-circles (or k column-circles) is completely disjoint from the other circles, it contains k eigenvalues [16]. When a group of k circles is completely disjoint from the other circles, it can be said that the system then has at least two-time scales. Whether this group of circles is to the right or to the left of the other circles, we can determine the k slow modes corresponding to these k circles or respectively the k fast modes. Each circle represents a state of the system. It is then possible to give an adequate partition of the model. Generally, this direct method does not make it possible to conclude immediately in all cases. Dauphin-Tanguy [17] then proposes the use of transformations:

$$S_k = \text{diag}(1, \dots, 1, \alpha_k, 1, \dots, 1), \quad k = 1, 2, \dots, n;$$

these parameters allow the variation of the circles size and their optimization lead to circles of minimum radii. However, this method does not separate all systems.

**3.1.3. Changing the radius size**

Let the matrix be:

$$S_k = \text{diag}(1, \dots, 1, \alpha_k, 1, \dots, 1), \quad k = 1, 2, \dots, n \tag{10}$$

The change of base  $\underline{X}' = S_k \underline{X}$  leads to a new state matrix. The radii  $R_{lk}$  and  $R_{ck}$  become  $R_{lk} \alpha_k$  and  $R_{ck} / \alpha_k$  respectively [18]. If the operation is repeated several times, the aggregated transformation is:

$$\begin{cases} \underline{X} = S \underline{X} \\ \underline{A} = S \underline{A} S^{-1} \end{cases} \text{ with: } S = \prod_k S_k \tag{11}$$

If there are two disjoint sets of circles, then the permutation matrix is:

$$\begin{cases} \underline{X} = P \underline{X} \\ \underline{A} = P \underline{A} P^{-1} \end{cases} \tag{12}$$

**3.1.4. Moving Circle Centers**

In order to improve the separation of dynamics, it is sometimes necessary to introduce a displacement of the circles which is characterized by the following transformation [18]:

$$T_l = I_n + B_l \cdot J_{ij} . \quad (13)$$

Only the elements of line  $i$  and column  $j$  change, the centers of the circles  $i$  and  $j$  are shifted from  $a_{ii}$  and  $a_{jj}$  to  $a_{ii} + B_l a_{ji}$  and  $a_{jj} + B_l a_{ji}$ , respectively. The choice of  $B_l$  can be made in such a way that  $X_{ij} = a_{ij} + B_l(a_{jj} - a_{ii}) - B_l^2 a_{ji} = 0$ . If several circles intersect, the terms  $B_l (l = 1, 2, \dots)$  are calculated in the same way, so the final transformation is:

$$\begin{cases} X = T X \\ A = T A T^{-1} \end{cases} \quad \text{with: } T = \prod T_l , \quad (14)$$

If two groups of circles are disjoint, the permutation matrix  $P$  is again:

$$\begin{cases} X = P X \\ A = P A P^{-1} \end{cases} \quad (15)$$

#### 4. CONCEPT OF MODULARITY

The application of concept of modularity to define the new architecture of small artificial networks involves the following four steps:

##### 4.1. The decomposition

The decomposition of a task into subtasks is the first step toward the application of modularity. It can be done on the input space (horizontal decomposition) or on the input variables (vertical decomposition) [19].

##### 4.2. Organization of the modular architecture.

The interconnection of the modules can be parallel or in series. In the parallel architecture, all modules process their information simultaneously. The global output involves some modules or all of them, depending on the application. The cooperation link between the modules which can be of type "and" or of type "or" [20].

##### 4.3. Nature of learning

The organization of NNs in a modular architecture makes learning more difficult. The modules of such an architecture can follow different learning processes.

##### 4.3.1. Independent learning

Training modules independently seems to be the simplest way. This suggests that the other modules of the architecture do not participate in learning. The interaction between the modules then occurs only during restitution phase [21].

##### 4.3.2. Cooperative learning

The proposed idea is to use a global method to train all modules at the same time. It is necessary then to have a fixed architecture, determined in advance. An example is given by ME [22].

##### 4.4. Communication between modules

The techniques for calculating the overall output of a multi-network architecture are diversely varied, among which is the technique of weighted votes [23]. A weight is then associated with each classifier representing a measure of performance. Another technique is to minimize the mean square error (MSE) of the global output.

#### 5. PROPOSED NEUROGLIAL NETWORK ARCHITECTURE

The architecture adopted for our ANG is partly based on the concepts of modularity and remains very close to the above-mentioned architecture "ME". In this architecture, a number of NNs (experts) are

supervised by a GN (Figure 1). The glial supervisor network determines the weighting coefficients of each expert's participation according to the input. The ANGN uses the "divide and conquer" strategy in which the responses from the expert NNs are combined into a single rapid response. The latter is aggregated to the slow response of the glial supervisor network, resulting in the overall response of our system. Algorithm developed in this article is based on softmax function. In this algorithm, the supervisor GN evaluates the performance of each expert NN according to the input and selects the best of them to be activated.

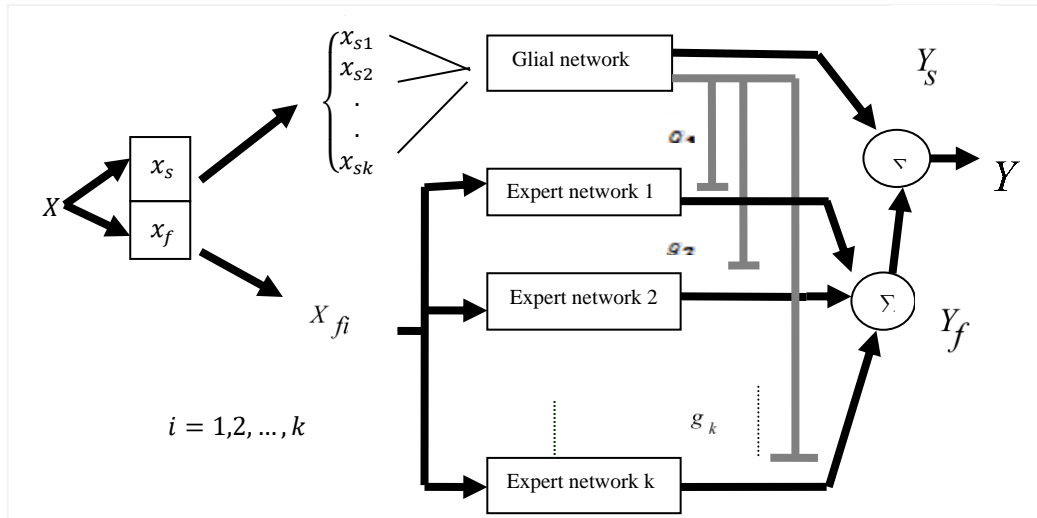


Figure 1. Neuroglial network architecture

As illustrated in Figure 1, our global ANGN is composed of  $K$  fast NNs and a slow supervisor GN. The vector of the inputs is divided into two vectors  $X_s$  and  $X_f$  representing, respectively, the slow inputs and the fast inputs of the network. The vector  $X_s$  is assigned to the supervisor network and the vector  $X_f$  is assigned to the various experts. The responses of the expert modules are combined to form the fast output. The supervisor GN has two outputs, the first one, which is used for control and supervision of the experts by selecting the most suitable network and deactivating the others for each input vector. The second output represents the slow response of the GN, this output is aggregated at the fast output to form the global response of the ANGN. The structure of the global ANGN is close to multi-model approach. Indeed, each expert network is specialized in a precise sub-problem, a vertical decomposition of the input variables into fast and slow inputs, as well as a horizontal decomposition of the fast input space  $X_f$  and slow  $X_s$  is used.

**5.1. Algorithm based on function softmax**

In the ANGN, the vector of the fast inputs  $X_f$  is sectioned both sequentially and in parallel into  $K$  vectors  $X_{f1}, X_{f2}, \dots, X_{fK}$ . These  $K$  vectors  $X_{fi}$  constitute the respective inputs of the  $K$  experts. The slow input vector  $X_s$  is also sectioned in the same way into  $K$  vectors  $X_{s1}, X_{s2}, \dots, X_{sK}$  which are applied consecutively to the supervisor GN.

In this algorithm, each vector of the fast inputs  $x_{i(f)}$ , ( $i = 1, 2, \dots, K$ ) is applied to all experts at the same time. These modules learn different examples from the learning base and specialize in specific groups of responses that are then weighted by the supervisor GN according to their absolute differences and the desired response. The expert whose response is the closest to the desired response will have the highest weight.

The supervisor GN, whose input is the vector of slow inputs  $x_{i(s)}$ , evaluates the performance of each expert according to the input and selects the best one to be activated. This algorithm shows many similarities to the one developed by Jacob [22] in the architecture "mixtures of experts". These similarities are related to the supervision and selection of experts, however there are three main differences:

- In the "mixture of experts" architecture, the supervisor and the experts have the same time scale. For our ANGN, the two types of modules work on two different time scales, slow and fast.
- The task of the supervisor in the ME approach is to supervise and control the competition of experts. In our approach, the GN supervises and controls the competition of experts as well as contributing to the overall response of the system by providing the slow response.

- For the selection, the experts' weighting is binary (0 or 1). In the ME approach weights are probabilities between 0 and 1.

Before presenting our algorithm, it is worth noting that the experts' learning are cooperative, which means experts learn simultaneously and divide the task during the learning process. The weights of the expert and those of the GN, for the selected vector, are updated at the same time by propagation. The learning of experts and the GN is carried out simultaneously by following these steps:

1. The separation of input vector  $X$  into both slow and fast vectors:  $x_s$  and  $x_f$ .
2. **Each** vector  $x_{i(f)}$ , ( $i = 1, 2, \dots, K$ ) is intended for all experts.
3. Each vector  $x_{i(s)}$ , ( $i = 1, 2, \dots, K$ ) is intended for the supervisor GN.
4. The learning of the GN to obtain the desired slow response corresponding to the input.
5. The selection of the expert according to the value of the probability  $p(i/x_s)$  and selection of the  $i$ th expert by evaluating the slow input  $x_{(s)}$ .
6. The output of expert  $i$  represents the conditional average of the desired response with respect the input and the expert network.

The learning algorithm of the ANGN architecture.

1. Initialization of the synaptic weights of experts and the GN.
2. for each slow input vector  $x_{i(s)}$  :

2.1 Calculate for  $i = 1, 2, \dots, k$

for  $m = 1, 2, \dots, q$

$$u_i(k) = X_{i(s)}^T a_i(k)$$

$$g_i(k) = \frac{\exp(u_i(k))}{\sum_{j=1}^k \exp(u_j(k))}$$

$$y_{i(f)}^m = X_{i(f)}^T W_{i(f)}^m(k)$$

$$y_{i(f)}(k) = [y_i^1, y_i^2, \dots, y_i^q]^T$$

$$h_i(k) = \frac{g_i(n) \exp\left(\frac{-1}{2} \|d_f - y_{i(f)}\|^2\right)}{\sum_{j=1}^k g_j \exp\left(\frac{-1}{2} \|d_f - y_{i(f)}\|^2\right)}$$

$$y_{i(s)}^m = X_{i(s)}^m W_{i(s)}^m(k)$$

$$e_{i(s)}^m(k) = d_s^m(k) - y_{i(s)}^m(k)$$

$$W_{i(s)}^m(k+1) = W_{i(s)}^m(k) + \eta e_{i(s)}^m(k) X_{i(s)}$$

$$e_{i(f)}^m(k) = d_{(f)}^m - y_{i(f)}^m(k)$$

$$W_{i(f)}^m(k+1) = W_{i(f)}^m(k) + \eta h_i(k) e_{i(f)}^m(k) X_{i(f)}$$

$$a_i(k+1) = a_i(k) + \eta (h_i(k) - g_i(k)) X_{i(s)}$$

end for

2.2. Repeat step 2.1 until the algorithm converges.

2.3. If  $\max(g_i)$  than  $g_i = 1$

else  $g_i = 0$

$$2.4. \quad y_{i(f)} = \sum_{i=1}^K g_i y_{i(f)}$$

$$Y_i = \sum_{i=1}^K (y_{i(s)} + y_{i(f)})$$

} Aggregation of the two slow and fast outputs

end for

## 6. APPLICATION

The ASM is a highly coupled nonlinear complex system and is a typical example of two time-scales system. The performance of the proposed ANGN architecture is assessed on both the reduced, slow and fast models of the machine. The state model of the induction machine in the stationary coordinate system  $(\alpha, \beta)$  can be written as:

$$\frac{d}{dt} \begin{bmatrix} \phi_{s\alpha\beta} \\ \phi_{r\alpha\beta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sigma T_s} I_2 & \frac{B_r}{\sigma T_s} R(-\theta) \\ \frac{1-\sigma}{\sigma T_r B_r} R(\theta) & -\frac{1}{\sigma T_r} I_2 \end{bmatrix} \begin{bmatrix} \phi_{s\alpha\beta} \\ \phi_{r\alpha\beta} \end{bmatrix} + \begin{bmatrix} v_{s\alpha\beta} \\ 0 \end{bmatrix} \quad (16)$$

$$T_{em} = P \frac{B_r}{\sigma L_s} \varphi_{s\alpha\beta}^T R(-\theta) J_2 \phi_{r\alpha\beta} \quad (17)$$

Or:

$$A = \begin{bmatrix} -\frac{1}{T_{sp}} I_2 & \frac{B_r}{T_{sp}} R(-\theta) \\ \frac{1-\sigma}{T_{rp} B_r} R(\theta) & -\frac{1}{T_{rp}} I_2 \end{bmatrix}$$

with :  $T_{sp} = \sigma T_s$ ,  $T_{rp} = \sigma T_r$ ,  $B_r = \frac{M}{L_r}$  and  $\sigma = (1 - M^2)/(L_s * L_r)$

Application of the GCT to the state matrix A results in two circles which intersect (Figure 2a). A change in the size of rays is carried out by transformation 1.

$$\phi_1 = P_1 \begin{bmatrix} \phi_{s\alpha\beta} \\ \phi_{r\alpha\beta} \end{bmatrix}, P_1 = \begin{bmatrix} I_2 & 0 \\ 0 & B_r R(-\theta) \end{bmatrix}$$

To get:

$$A_1 = \begin{bmatrix} -\frac{1}{T_{sp}} I_2 & \frac{1}{T_{sp}} I_2 \\ \frac{1-\sigma}{T_{rp}} I_2 & -\frac{1}{T_{rp}} I_2 - \omega J_2 \end{bmatrix} \quad (18)$$

The centers of the two circles (Figure 2a) are relocated by:

$$\phi_2 = P_2 \phi_1, P_2 = \begin{bmatrix} I_2 & 0 \\ -I_2 & I_2 \end{bmatrix}$$

And

$$A_2 = \begin{bmatrix} 0 & \frac{1}{T_{sp}} I_2 \\ -\frac{\alpha\sigma}{T_{sp}} I_2 + \omega J_2 & -\frac{\alpha+1}{T_{sp}} I_2 - \omega J_2 \end{bmatrix}$$

The new line circles are still double and disjoint (Figure 2b); the final transformation is:

$$\Phi = P_1 P_2 \begin{bmatrix} \phi_{s\alpha\beta} \\ \phi_{r\alpha\beta} \end{bmatrix}, P_1 P_2 = \begin{bmatrix} I_2 & 0 \\ -I_2 & B_r R(-\theta) \end{bmatrix} \quad (19)$$



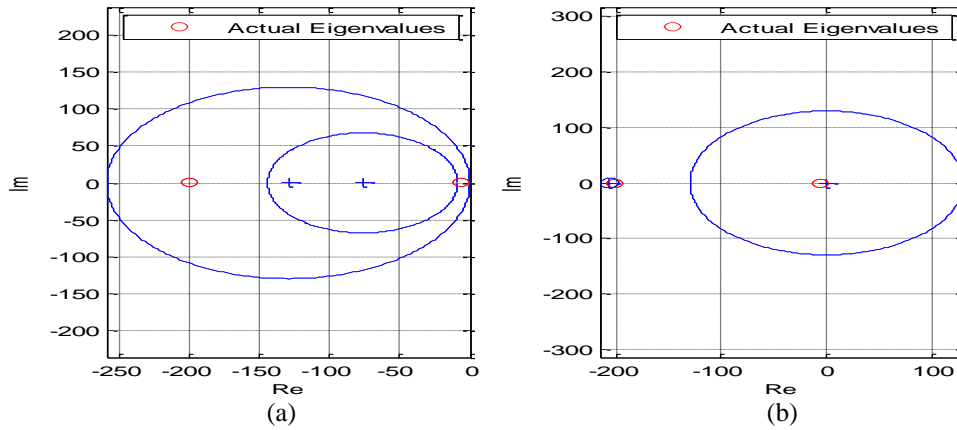


Figure 2. (a) Circles that intersect; (b) Disjointed Circles

In this case, the slow and fast components are easily identified. We can then apply SPM to develop the slow and fast submodels. By putting:

$$\phi = \begin{bmatrix} x \\ z \end{bmatrix}$$

We get:

$$\frac{d}{dt} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{T_{sp}} I_2 \\ -\frac{1}{T_{rp}} I_2 & -\frac{\alpha + 1}{T_{sp}} I_2 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} I_2 & 0 \\ -I_2 & B_r R(-\theta) \end{bmatrix} \begin{bmatrix} v_{s\alpha\beta} \\ 0 \end{bmatrix}, \tag{20}$$

$$T_{em} = \frac{P}{L_{sp}} x^T J_2 z$$

This form is standard, the flux  $x$  is slow and the flux  $z$  is fast. By decomposing the fluxes, the voltages and the torque, we obtain:

$$\begin{aligned} x &= x_s(t) \\ z &= z_s(t) + z_f(\tau) \\ v_{s\alpha\beta} &= v_{s\alpha\beta(s)}(t) + v_{s\alpha\beta(f)}(\tau) \\ T_{em} &= T_{em(s)} + T_{em(f)} \end{aligned} \tag{21}$$

The reduced slow model is then:

$$\frac{d}{dt} \begin{bmatrix} \phi_{s\alpha} \\ \phi_{s\beta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{(1+\alpha)T_r} & 0 \\ 0 & -\frac{1}{(1+\alpha)T_r} \end{bmatrix} \begin{bmatrix} \phi_{s\alpha} \\ \phi_{s\beta} \end{bmatrix} + \frac{\alpha}{1+\alpha} \begin{bmatrix} v_{s\alpha(s)} \\ v_{s\beta(s)} \end{bmatrix}$$

$$T_{em(s)} = \frac{pT_s}{(1+\alpha)L_s} (\phi_{s\alpha} v_{s\beta(s)} - \phi_{s\beta} v_{s\alpha(s)})$$

The reduced fast model is:

$$\frac{d}{dt} \begin{bmatrix} I_{s\alpha(f)} \\ I_{s\beta(f)} \end{bmatrix} = \begin{bmatrix} -\frac{1+\alpha}{T_s} & 0 \\ 0 & -\frac{1+\alpha}{T_s} \end{bmatrix} \cdot \begin{bmatrix} I_{s\alpha(f)} \\ I_{s\beta(f)} \end{bmatrix} - \frac{1}{L_s} \cdot \begin{bmatrix} v_{s\alpha(f)} \\ v_{s\beta(f)} \end{bmatrix}$$

$$I_{s\alpha\beta(s)} = \frac{1}{L_{sp}} \left( \frac{\alpha\sigma}{1+\alpha} \phi_{s\alpha\beta(s)} + \frac{T_{sp}}{1+\alpha} v_{s\alpha\beta(s)} \right),$$

$$T_{em(f)} = p(\phi_{s\alpha} I_{s\beta(f)} - \phi_{s\beta} I_{s\alpha(f)})$$

$$\begin{cases} \phi_{r\alpha\beta(s)} = \frac{1}{B_r} R(\theta) \left( \left( 1 - \frac{\alpha\sigma}{1+\alpha} \right) \phi_{s\alpha\beta(s)} - \frac{T_{sp}}{1+\alpha} v_{s\alpha\beta(s)} \right) \\ \phi_{r\alpha\beta(f)} = -\frac{L_s}{B_r} R(\theta) I_{s\alpha\beta(f)} \end{cases}$$

**6.1. Results and discussion.**

The ANGN\_MAS model is composed of two groups of inputs  $v_{sa}, v_{s\beta}$ , which are decomposed into two slow inputs  $v_{sa(s)}, v_{s\beta(s)}$  and two fast inputs  $v_{sa(f)}, v_{s\beta(f)}$ . The ANGN\_MAS consists of four experts and a glial supervisor network. The four experts and the glial network have similar architectures, consisting of an input layer of four neurons and an output layer of one neuron. The effectiveness of the ANGN algorithm proposed in this paper is illustrated by the performance index root mean square (RMS) value. The RMS of the states error is calculated as:

$$RMS = \sqrt{\frac{\sum_{i=1}^n e^2(i)}{n}}$$

where n is the number of simulation steps, and  $e(i)$  is the difference between the state variables of the model and the true system at the ith step. Figure 3 (a) and (b) show the good convergence of this learning algorithm. The minimum value, which is very close to zero occurs after only two iterations. The algorithm converged perfectly and quickly.

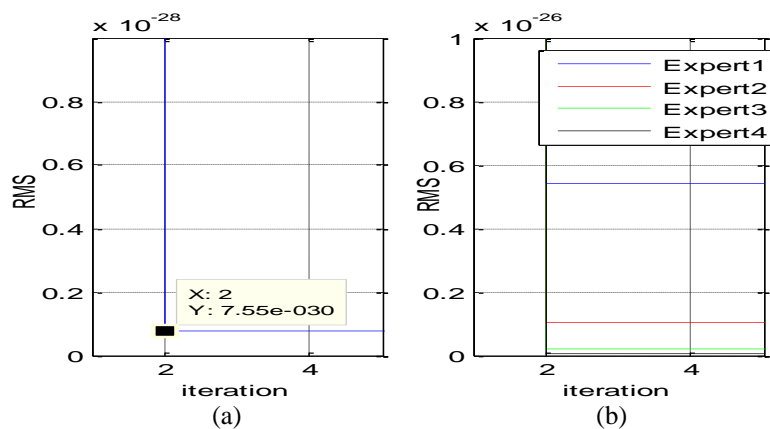


Figure 3. (a) Evolution of the glial network; (b) Evolution of experts

For a comparison with the algorithms in [7], we choose the same parameters of the induction motor. The simulation results are presented in Figure 4 and 5. The RMS values for state variables are presented in Table 1.

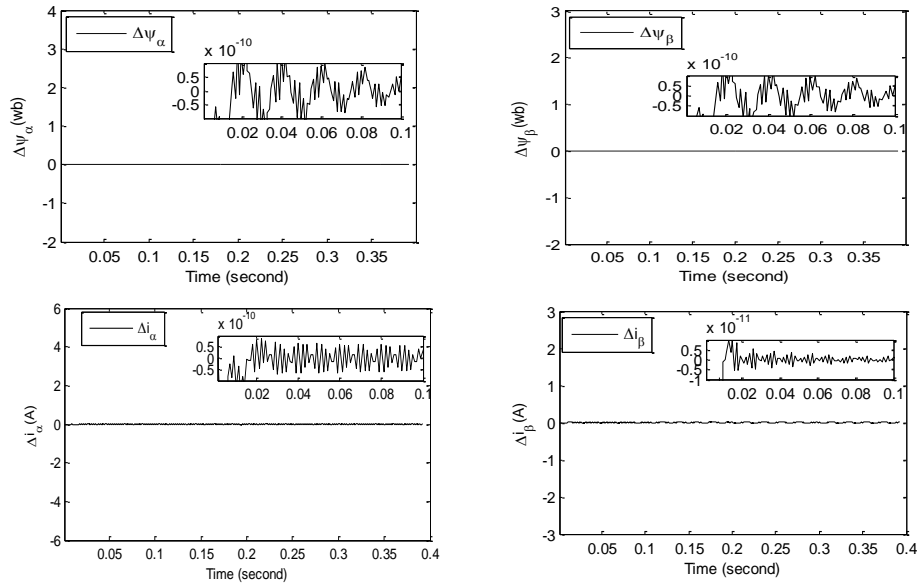


Figure 4. Error for states  $\varphi_{r\alpha}$ ,  $\varphi_{r\beta}$ ,  $i_{\alpha}$ ,  $i_{\beta}$ : of proposed ANGn

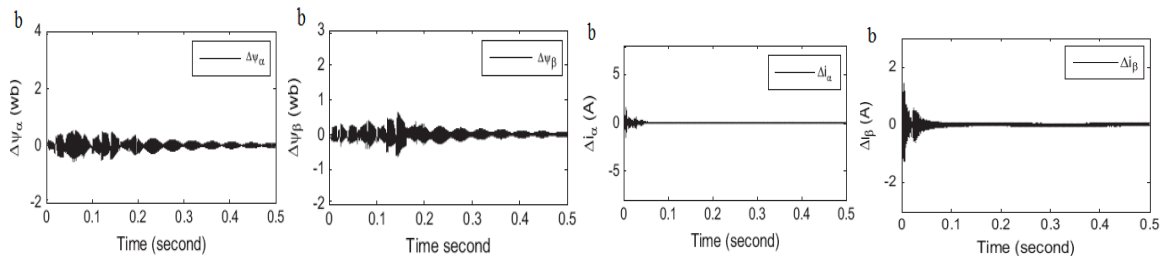


Figure 5. Error for states  $\varphi_{r\alpha}$ ,  $\varphi_{r\beta}$ ,  $i_{\alpha}$ ,  $i_{\beta}$ : [7]

Table 1. The RMS values for state variables

	$\varphi_{r\alpha}(\text{wb})$	$\varphi_{r\beta}(\text{wb})$	$i_{\alpha}(\text{A})$	$i_{\beta}(\text{A})$
RMS (ref [7])	0.0575	0.05334	0.0446	0.0452
RMS (ANGn)	$1.029 \times 10^{-23}$	$3.33 \times 10^{-24}$	$3.33 \times 10^{-23}$	$9.8 \times 10^{-23}$

The RMS values of all state variables in Figure 4 and 5 demonstrate that the performance has been improved compared to those [7], it is very clear from these figures that the identification errors  $\Delta\varphi_{\alpha}$ ,  $\Delta\varphi_{\beta}$ ,  $\Delta i_{\alpha}$ ,  $\Delta i_{\beta}$  of the proposed ANGn are greatly reduced.. We can confirm that the state variables of ANGn follow those of the nonlinear system more accurately and faster. This is due to the good separation of slow and fast modes which result that the complexity of the architecture sub-networks of ANGn are considerably reduced.

We conducted other comparisons of the proposed approach with both the modular approach in [24] and the mixture of experts (ME) approach. The RMS values of the  $T_{em}$  are given in Table 2. From Table 2, it is very clear that RMS values of ANGn are much smaller than both the values [24] and ME, which means that the proposed algorithm in this paper can achieve more accurate results. From Figure 3(a) and 3(b), we can see. Therefore the excellent performance of the proposed model in terms of convergence speed and reduction of the optimal RMS error than both the performance in Figure 6 and performance in Figure 7. It is largely due to the small size of the sub-networks and the very limited number of examples presented at the input of each expert. These advantages are the main properties of modularity.

Table 2. Comparison performance models

Models	Number of iterations	RMS
ref [24]	100	1.08932 E-27
ME	90	7.934 E-24
ANGN	2	17.455 E-30

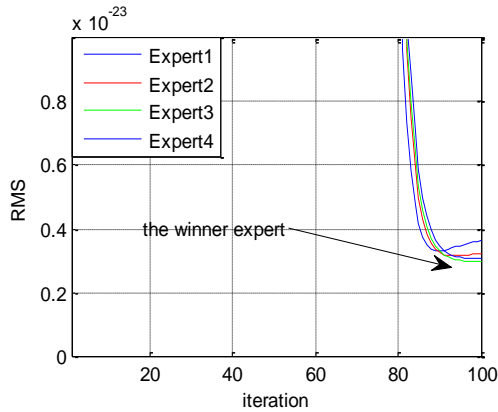


Figure 6. Evolution of RMS of the ME model

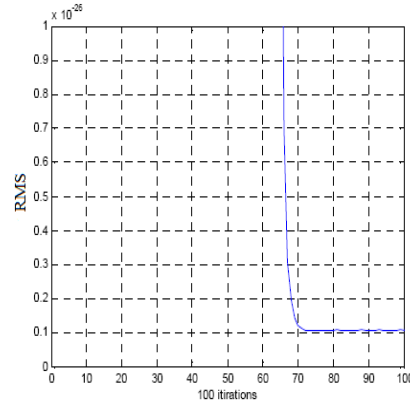


Figure 7. Evolution of RMS in ref [24]

## 7. CONCLUSION

In this paper, a novel architecture called artificial neuroglial network (ANGN), inspired by a recently discovered fact in biology, was developed along with a modular algorithm based on the softmax function. Because of its good characteristics, an ASM put in standard singularly perturbed form is chosen to validate the proposed ANGN architecture. The slow and fast reduced models are obtained in two steps:

- Application of a geometric approach based on GCT to decouple the slow and fast variables.
- Application of the SPM to develop the reduced models.

The simulation results related to the learning of the ANGN\_ASM model demonstrate the fast and accurate convergence properties of the proposed algorithm. ANGN can modeling ASM very well with less computational complexity. The next step is to use these reduced models to develop controllers for the ASM.

## REFERENCES

- [1] S. Prasad, *et al.*, "Comparison of Accuracy Measures for RS Image Classification using SVM and ANN Classifiers," *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue: 7(3), pp. 1180-1187, 2017.
- [2] S. R. Borra, *et al.*, "An Efficient Fingerprint Identification using Neural Network and BAT Algorithm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue: 8(2), pp. 1194-1213, 2018.
- [3] H. Mohammed and A. Meroufel, "Contribution to the Artificial Neural Network Speed Estimator in a Degraded Mode for Sensor-Less Fuzzy Direct Control of Torque Application Using Dual Stars Induction Machine," *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue: 5(4), pp. 729-741, 2015.
- [4] Z. Mekrini and S. Bri, "High-Performance using Neural Networks in Direct Torque Control for Asynchronous Machine," *International Journal of Electrical & Computer Engineering*, vol/issue: 8(2), 2018.
- [5] P. Kokotovic, *et al.*, "Singular perturbation methods in control: analysis and design," vol. 25, 1999.
- [6] D. Naidu, "Singular perturbations and time scales in control theory and applications: an overview," *Dynamics of Continuous Discrete and Impulsive Systems Series B*, vol. 9, pp. 233-278, 2002.
- [7] Z. J. Fu, *et al.*, "Robust on-line nonlinear systems identification using multilayer dynamic neural networks with two-time scales," *Neurocomputing*, vol. 113, pp. 16-26, 2013.
- [8] X. Li and W. Yu, "Dynamic system identification via recurrent multilayer perceptrons," *Information sciences*, vol/issue: 147(1-4), pp. 45-63, 2002.
- [9] C. S. Leung and L. W. Chan, "Dual extended Kalman filtering in recurrent neural networks," *Neural Networks*, vol/issue: 16(2), pp. 223-239, 2003.
- [10] D. D. Zheng, *et al.*, "Indirect adaptive control of nonlinear system via dynamic multilayer neural networks with multi-time scales," *International Journal of Adaptive Control and Signal Processing*, vol/issue: 29(4), pp. 505-523, 2015.
- [11] D. D. Zheng, *et al.*, "Identification and trajectory tracking control of nonlinear singularly perturbed systems," *IEEE Transactions on Industrial Electronics*, vol/issue: 64(5), pp. 3737-3747, 2017.

- [12] D. D. Zheng, *et al.*, “Robust identification for singularly perturbed nonlinear systems using multi-time-scale dynamic neural network,” *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, 2017.
- [13] Astrocytes, “Science & vie,” pp. 67, 2005.
- [14] A. B. Porto and A. Pazos, “Neuroglial behaviour in computer science,” *Artificial neural networks in real-life applications*, IGI Global, pp. 1-21, 2006.
- [15] Y. V. Hote and A. N. Jha, “New approach of Gerschgorin theorem in model order reduction,” *International Journal of Modelling and Simulation*, vol/issue: 35(3-4), pp. 143-149, 2015.
- [16] O. Touhami, *et al.*, “Dynamics separation of induction machine models using gerschgorin's circles and singular perturbations,” *Electrical and Electronics Engineering, 2004. (ICEEE). 1st International Conference on*, 2004.
- [17] G. D. Tanguy, *et al.*, “Singular perturbation method and reciprocal transformation on two-time scale systems,” *Multivariable Control*, Springer, pp. 327-342, 1984.
- [18] H. Guesbaoui and C. Iung, “Multi-time scale modelling in electrical machines,” *Systems, Man and Cybernetics, 1993. 'Systems Engineering in the Service of Humans', Conference Proceeding, International Conference on*. 1993.
- [19] E. Ronco, *et al.*, “Modular neural network and self-decomposition,” *Connection Science (special issue: Combining Neural Nets)*, 1996.
- [20] Y. Bennani, “A modular and hybrid connectionist system for speaker identification,” *Neural Computation*, vol/issue: 7(4), pp. 791-798, 1995.
- [21] I. Kirschning, *et al.*, “A parallel recurrent cascade-correlation neural network with natural connectionist glue,” *Neural Networks, 1995. Proceedings, IEEE International Conference on*, 1995.
- [22] R. A. Jacobs, *et al.*, “Adaptive mixtures of local experts,” *Neural computation*, vol/issue: 3(1), pp. 79-87, 1991.
- [23] H. Drucker, *et al.*, “Boosting and other ensemble methods,” *Neural Computation*, vol/issue: 6(6), pp. 1289-1301, 1994.
- [24] S. Salah, *et al.*, “Application New Approach with Two Networks Slow and Fast on the Asynchronous Machine World Academy of Science, Engineering and Technology,” *International Journal of Electrical and Computer Engineering*, vol/issue: 7(8), pp. 1087-1091, 2013.