**Advances in
Radio Science**

Macro-modelling via radial basis functionen nets

C. Wiegand[1,2], **C. Fischer**[1], **R. Kazemzadeh**[2,3], **C. Hedayat**[1,2], **W. John**[2], and **U. Hilleringmann**[1]

[1]University of Paderborn, Department of Electrical Engineering, Sensor Technology Group, Germany
[2]Fraunhofer Institute for Reliability and Microintegration (IZM) Advanced System Engineering (ASE), Paderborn, Germany
[3]Leibniz University of Hannover Institute of Electromagnetic Theory (TET), Hannover, Germany

Abstract. By the rising complexity and miniaturisation of the device's dimensions, the density of the conductors increases considerably. Referring to this, locally transient interactions between single physical values become apparent. Therefore, for the investigation and optimisation of integrated circuits it is essential to develop suitable models and simulation surroundings which allow for memory and time-efficient calculation of the behaviour. By means of the dynamic reconstruction theory and the radial basis functions nets the so-called black box models are provided. The description of black box models is derived from the input and output behaviour or so-called time series of a dynamic system. Concerning the time series, the black box model adapts its parameters via the extended Kalman filter. This paper provides a modelling approach that enables fast and efficient simulations.

1 Introduction

The increasing simulation time is a big limiting factor during the design and the development of HDI/HDP systems. Hence, it is more and more crucial to provide models which allow for quick and accurate simulations (Wiegand et al., 2007b,a; Stievano, 2002). The classical modelling approach, based on the transistor level circuits and the symbolic analysis, requires unreasonably long computation times. The black box modelling (BBM) via radial basis functions (RBF) nets for integrated circuits is an adequate solution that provides robust and fast simulations. In the following, a method is presented for developing mathematical models to reduce the very growing simulation times for transient analysis. This modelling approach is based on the dynamic reconstruction of dynamic systems. That is, a mathematical formulation needs to be found, that describes the dynamics of the system to be examined as accurately as possible without necessarily considering the physical characteristics of the circuit.

In Sect. 2 the modelling flow for developing black box models is discussed, although the radial basis functions nets are the main focus of all considerations. Section 3 describes the architecture of the RBF nets. The parameter adaptation, i.e. the extended Kalman filter, is explained in Sect. 4. In Sect. 5 a NOR element with two inputs and three outputs is examined. Then, an example of a diode is considered and the RBF model is parameterised in such a way so as to allow for the temperature to be regarded as a further input.

2 Modelling flow

It is necessary for the developing of black box models by means of radial basis functions nets to choose a suitable time series (measurements or simulations), so that the model is as close to the original system as possible. Thus, the training set must comprise a high information content with which a reliable model can be provided. The next step in the modelling chain (see Fig. 1) is the embedding (Sauer et al., 1991) of this time series in the RBF net, i.e. the choosing of centres. This can be done with the help of the so-called locally regularised orthogonal least square (Chen et al., 1996,?; Chena et al., 1990a) or the K-Means-Cluster-Algorithm (Sing et al., 2003). Within this work the centres are determined by the clustering procedure, because the initialisation of the remaining parameters of the net can be determined via least square methods. Afterwards the parameter adaptation (parameter estimation) is achieved by means of the extended Kalman filter (EKF) (Haykin, 2001; Ahmida and Charef, 2002). The final step is the Validation, i.e., assuming that the error between the model and the system is in the expected order of magnitude, a robust simulation model is provided. If this case does not occur, it has to be assessed, whether the training signal was chosen appropriately since the centres were possibly not picked wisely, or the parameter adaptation diverged or remained in a local minimum.

Fig. 1. Modelling flow.

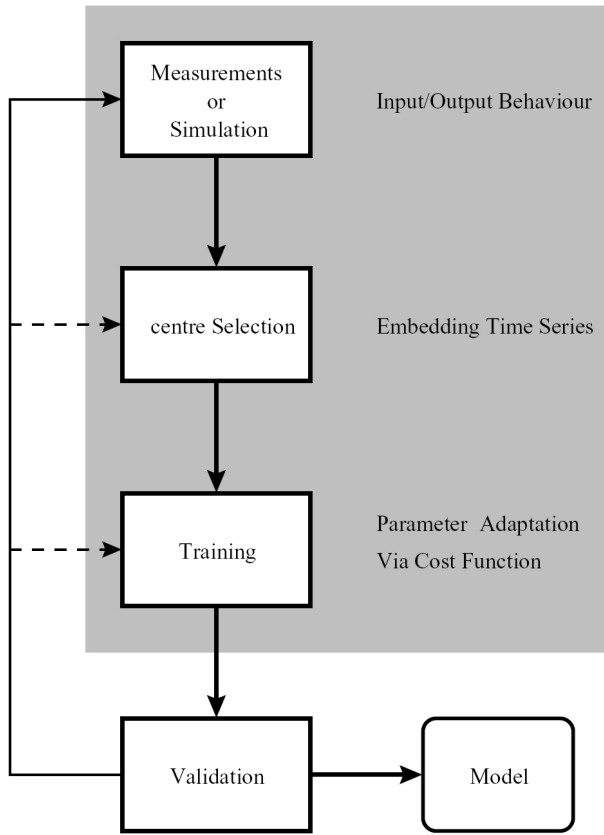**Fig. 2.** Schematic representation of a multiple input and multiple output radial basis functions net.

3 Radial basis functions nets

With using RBF networks, it is possible to represent any continuous nonlinear function. Because of this important feature, they are utilised to model nonlinear dynamic systems and integrated circuits. Radial basis functions nets are special feedforward neural networks consisting of three layers with a hidden layer.

Input layer: definition of the number of inputs, i.e. within the regressor vector the number of inputs is fixed.

Hidden layer: signal processing via p radial basis functions (Gaussian) or so-called neurons. Furthermore the embedding of the data occurs just in the hidden layer.

Output layer: definition of the number of outputs. Regarding weighted transitions the neurons outgoing signals are added up within every output.

Figure 2 shows a schematic representation of a typical RBF network with multiple inputs and multiple outputs (MIMO). The response of a MIMO radial basis functions net can be formulated as follows:
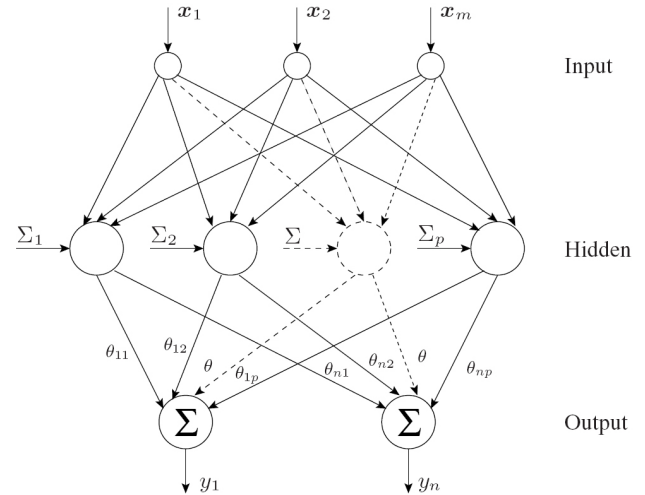
$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_m \end{bmatrix} = \begin{bmatrix} \theta_{11} & \cdots & \theta_{1p} \\ \vdots & \ddots & \vdots \\ \theta_{m1} & \cdots & \theta_{mp} \end{bmatrix} \begin{bmatrix} \phi_1(\zeta_1) \\ \vdots \\ \phi_p(\zeta_p) \end{bmatrix} \qquad (1)$$

where $\phi_i(\zeta_i)$ denotes the ith gaussian activation function in the hidden layer and can be written as

$$\phi_i(\zeta_i) = e^{-\zeta_i} \text{ mit } \zeta = (x - c_i)^T \Sigma_i^{-1} (x - c_i) \qquad (2)$$

and x is the regressor vector and is denoted by

$$x = \begin{bmatrix} x_1^T, \cdots, x_n^T \end{bmatrix}^T. \qquad (3)$$

Within the regressor vector the n inputs are defined; m and n describe the number of outputs and the number of basis functions; c_i is the ith centre and $\Sigma_i = \text{diag}(\sigma_i) = \text{diag}([\sigma_{1i}^2, \cdots, \sigma_{ri}^2])$ represents the shape of the ith basis function. Equation 1 can be described in the following way:

$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_m \end{bmatrix} = \begin{bmatrix} \theta_1^T \\ \vdots \\ \theta_m^T \end{bmatrix} \begin{bmatrix} \phi_1(\zeta_1) \\ \vdots \\ \phi_p(\zeta_p) \end{bmatrix}. \qquad (4)$$

$$\hat{y} = \Theta \phi$$

In a standard design, the RBF net provides a static input and output behaviour. For temporal processing, time delays have to be introduced to convert the static RBF network into a dynamic network. There are only very few possibilities to include memory into the architecture. For example discrete

time FIR[1]/IIR-Filters[2] can be used instead of the weights or global feedback paths (Howlett, 2001a,b) could be introduced. A simple method to create dynamics can be expressed by:

$$x_i(t) = [u_i(t-1), \cdots, u_i(t-d_1),$$
$$y_i(t-1), \cdots, y_i(t-d_2)]^T, \quad (5)$$

where d_1 and d_2 denote the input embedding dimension and the output embedding dimension. By making such changes, a state space model can be described (Wiegand et al., 2007a). The first order derivation can be employed to transform the discrete time representation into a continuous state space model.

4 Parameter estimation

Generally, the parameters of a RBF net can be determined by a huge number by algorithms, for example, by the locally regularised least square algorithm with centre selection (Chena et al., 1990a; Chen et al., 1990b, 1996). Nevertheless, non-linear optimisation methods must then be employed to adapt the shapes Σ_i of the basis functions. Within this work, the extended Kalman filter (EKF) is used for parameter identification of non-linear systems. In accordance to this, all parameters are determined iteratively by the EKF, including those which are in the argument of the gaussian function. For the initialisation and the improvement of the convergence, the K-Means cluster algorithms is used to determine the centres. In addition, this cluster algorithm delivers an facile possibility for the calculation of the shapes (widths) of the basis functions, to be able to determine the weights by means of least square methods.

4.1 K-Means cluster algorithm

The algorithm is an iterative procedure to divide a set of training data into groups (cluster). The only available information is the training data and the number of the clusters. This is also known as unsupervised learning. A random choice of the cluster centres $c(0)$ within the training set $\mathcal{T}$ serves as the starting point. Next, in every iteration step k, every set of trainig data is allocated to the nearest situated cluster centre and thus the subsets $\mathcal{M}_i$ are obtained. The identification of the new centres
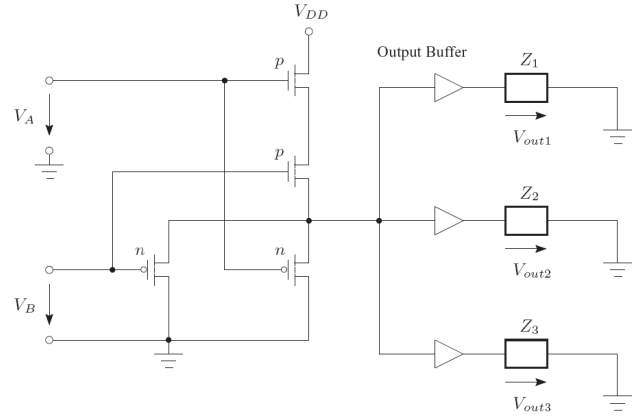
$$c_i(k+1) = \frac{1}{N_i} \sum_{x \in \mathcal{M}_i} x \quad (6)$$

is accomplished by means of average determination via the sets

$$\mathcal{M}_i = \{x \in \mathcal{T} : c_i(k) = \underset{c(k)}{\operatorname{argmin}} \|x - c(k)\|^2\}, \quad (7)$$

[1]FIR: Finite Impulse Response
[2]IIR: Infinite Impulse Response

Fig. 3. NOR-Circuit with two inputs (V_A und V_B) and three outputs (V_{out1}, V_{out2} und V_{out3}). The buffers of the respective outputs are four cascaded inverters. Also, the outputs are concluded with different complex impedances (Z_i, where $i \in [1, 2, 3]$).

whereas N_i denotes the number of elements which are in the set $\mathcal{M}_i$. An initialisation of the matrices Σ can take place after the determination of the centres, i.e. after the last iteration step:

$$\sigma_{i,j} = \sqrt{\frac{1}{N_i} \sum_{x \in \mathcal{M}_i} (x_j - c_{i,j}(K))^2} \quad (8)$$

$$\sigma_i = [\sigma_{i,1}, \cdots, \sigma_{i,r}]^T \quad (9)$$

$$\Sigma_i = (\operatorname{diag}(\sigma_i))^2 \quad (10)$$

With the calculated initial centres and the widths of the basis functions, it is possible to determine the weights, i.e., by utilising the least square methods.

4.2 Extended Kalman Filter

Iteratively, the EKF estimates the parameter vector

$$w_k = \left[\theta_1^T, \cdots, \theta_p^T, \sigma_1^T, \cdots, \sigma_p^T, c_1^T, \cdots, c_p^T\right]^T. \quad (11)$$

using

$$w_k = w_{k-1} + K_k e \quad (12)$$

where K_k is the Kalman gain.

$$K_k = P_{k-1} H_k \left[R_k + H_k^T P_{k-1} H_k\right]^{-1} \quad (13)$$

$e = y - \hat{y}$ is the error between the observed outputs y (target vector) and the estimated outputs $\hat{y}$ of the RBF network; $H_k = \nabla_w f(x, w)$ is the matrix of the partial derivative of $f(x, w)$. For this reason, when considering the jth output, the derivations of $f_j(x, w)$ are related respectively to the parameters c_i, θ_{ji} and Σ_i:
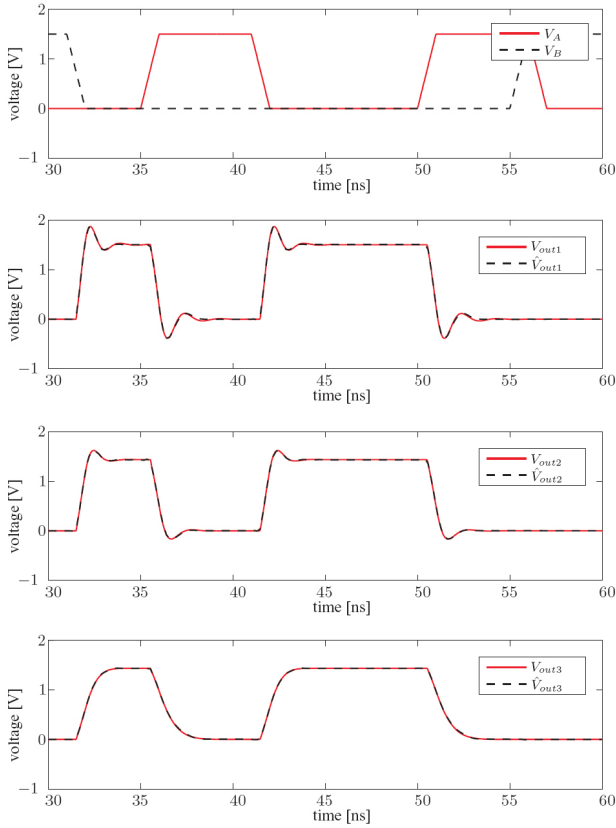
Fig. 4. Input and output signals of the simulated system and model. The output voltages marked with a hat denote the outputs of the RBF model.

$$\frac{\partial f_j(\boldsymbol{x}, \boldsymbol{w})}{\partial \boldsymbol{c}_i} = 2\theta_{ji} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{c}_i) e^{-\zeta} \qquad (14)$$

$$\frac{\partial f_j(\boldsymbol{x}, \boldsymbol{w})}{\partial \theta_{ji}} = e^{-\zeta} \qquad (15)$$

$$\frac{\partial f_j(\boldsymbol{x}, \boldsymbol{w})}{\partial \boldsymbol{\sigma}_i} = 2\theta_{ji} \widetilde{\boldsymbol{\Sigma}}_i^{-1} \mathrm{sq}(\boldsymbol{x} - \boldsymbol{c}_i) e^{-\zeta} \qquad (16)$$

where $\widetilde{\boldsymbol{\Sigma}}_i$ is defined as

$$\widetilde{\boldsymbol{\Sigma}}_i = \mathrm{diag}\left(\widetilde{\boldsymbol{\sigma}}_i\right) = \mathrm{diag}\left([\sigma_{1i}^3, \cdots, \sigma_{ri}^3]\right) \qquad (17)$$

with

$$\mathrm{sq}(\boldsymbol{x} - \boldsymbol{c}_i) = [(x_1 - c_{1i})^2, \cdots, (x_r - c_{ri})^2]^T. \qquad (18)$$

$\boldsymbol{R}_k$ and $\boldsymbol{P}_k$ denote the variance of the measurement noise and the error covariance matrix. The error covariance matrix is a positive definite symmetric $\nu \times \nu$ matrix, where ν is the number of parameters being updated and can be determined by:

$$\boldsymbol{P}_k = \left[\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k^T\right] \boldsymbol{P}_{k-1} + q_0 \boldsymbol{I}. \qquad (19)$$
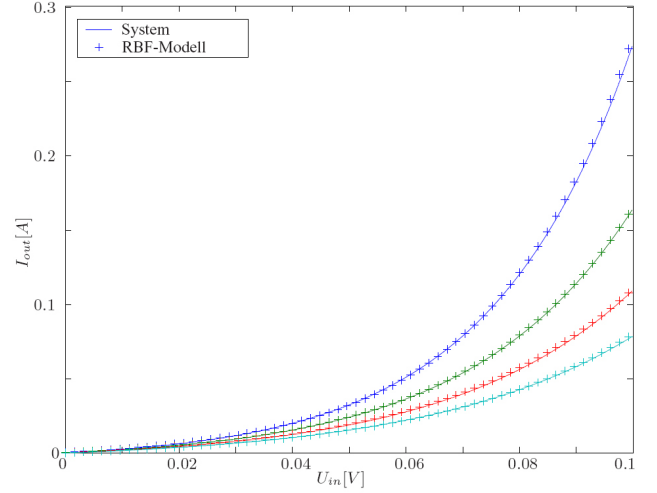
Fig. 5. Identity lines of the diode and the model for different temperatures.

According to Ahmida and Charef (2002), q_0 regulates the allowed random step towards the gradient. For fast parameter estimation, the training data is subdivided into smaller sets. These sets are used to adjust the parameters of the RBF net. Afterwards, a test is performed as well as an update of the regressor vector.

5   Simulations

In the following section, two different nonlinear systems are considered.

The first example corresponds to a two input and three output NOR-Circuit (see Fig. 3). In the second example a diode is envisaged, in which the ability of parameterisation of the RBF nets is illustrated.

5.1   Example 1: NOR Circuit

At first a NOR circuit is considered, which was built-up and simulated by using the Agilent Advanced Design System (ADS) environment. This circuit was implemented with two inputs and three outputs with different complex loads, from which a MIMO system arises. The circuit is implemented within 130 nm technology derived from an ADS-implemented Philips library. The radial basis functions net has been trained using a set of 1000 data samples of the trapezoidal input signals and the corresponding output signals. The RBF net is developed with 15 centres (selected via the presented cluster algorithm) and the regressor vector is defined as follows:

$$\boldsymbol{x}(k) = \left[\boldsymbol{u}^T(k), \boldsymbol{u}^T(k-1), \boldsymbol{y}^T(k-1)\right]^T \tag{20}$$

$$\boldsymbol{u}(k) = [u_1(k), u_2(k)]^T \tag{21}$$

$$\boldsymbol{y}(k) = [y_1(k), y_2(k), y_3(k)]^T \tag{22}$$

In Fig. 4 the input signals (V_A und V_B) as well as the response of the ADS-system (V_{out1}, V_{out2} und V_{out3}) and the RBF model ($\hat{V}_{out1}$, $\hat{V}_{out2}$ und $\hat{V}_{out3}$) are shown. The model is able to represent the dynamic as well as the logical behaviour of the system. The mean square error (MSE) is about $4\cdot10^{-5}$.

5.2 Example 2: diode

The system under consideration is a diode (SISO), but the RBF net was implemented with a second input. This second input is virtual and corresponds to the parameterisation of the model taking the temperature drift into account. This means that the temperature is considered at the modelling level as if it were an other input. This is therefore a simple example that shows the ability to parameterise the RBF nets. The most simple identity equation is

$$y(k) = I_s \left(e^{\frac{qU_{in}}{kT}} - 1\right), \tag{23}$$

where U_{in} denotes the input voltage, k and q stand for the Boltzmann constant and the elementary charge. I_s is the saturation current and for simulation purposes I_s was fixed to $5\cdot10^{-3}$ A. During the training, the temperature was varied between 18°C to 33°C. The training signal, consisting of 1000 values, had a sine-shaped course and formed, in combination with the temperature, the two-dimensional regressor. For conclusive testing, four characteristic curves were simulated for 21°C, 24°C, 27°C und 30°C (Fig. 5). The mean square error is in the order of magnitude 10^{-6}.

6 Conclusions

In this paper it was explained how a MIMO RBF net can be generated by means of the K-Means cluster algorithm and the extended Kalman filter. It became apparent that it is possible to model physical parameters in the same fashion. To achieve a noticeable gain in speed, the systems to be modelled must be substantially more complex. Nevertheless, it was shown that the RBF nets are able to carry out precise and fast system-level simulations. The future efforts will concentrate on the application of the findings in a industrial environment.

References

Chen, S., Billings, S. A., Cowan, C. F. N., and Grant, P. M.: Nonlinear system identification using radial basis functions, Int. J. Control, 21, 2513-2539, 1990a.

Chen, S., Billings, S. A., Cowan, C. F. N., and Grant, P. M.: Practical identification of NARMAX models using radial basis functions, Int. J. Control, 52, 1327-1350, 1990b.

Chen, S., Chng, E. S., and Alkadhimi, K.: Regularized orthogonal least squares algorithm for constructing radial basis function networks, Int. J. Control, 64, 829-837, 1996.

Haykin, S.: Kalman Filtering and Neural Networks, John Wiley & Sons, Inc., New York, Chichester, Weinheim, Brisbane, Singapore, Toronto, 2001.

Howlett, R. J. and Jain, L. C.: Radial Basis Funcrion Networks 1 – Recent Developments in Theory and Application, Physica-Verlag, 2001a.

Howlett, R. J. and Jain, L. C.: Radial Basis Funcrion Networks 2 – New Advances in Design, Physica-Verlag, 2001b.

Sauer, T., Yorke, J. A., and Casdagli, M.: Embedology, J. Stat. Phys., 65, 579-616, 1991.

Sing, J. K., Basu, D. K., Nasipuri, M., and Kundu, M.: Improved K-means Algorithm in the Design of RBF Neural Networks, TENCON 2003, Conference on Convergent Technologies for Asia-Pacific Region, 2, 841-845, 2003.

Stievano, I. S., Maio, I. A., and Canavero, F. G.: Parametric Macromodels of Digital I/O Ports, IEEE T. Adv. Packaging, Special Selection on EPEP01, 25(2), 255–264, 2002.

Wiegand, C., Hedayat, C., John, W., Radic-Weissenfeld, L., and Hilleringmann, U.: Nonlinear Identification of Complex Systems using Radial Basis Function Networks and Model Order Reduction, IEEE International Symposium on EMC, Honolulu Hawaii, USA, 2007a.

Wiegand, C., Radic-Weissenfeld, L., Hedayat, C., John, W.: Black Box Model and Singular Value Based Model Order Reduction, 18th International Zurich Symposium on EMC, 2007b.

Ahmida, Z. and Charef, A.: Nonlinear Systems Modelling Using RBF Neural Networks: A Ransdom Learning Approach to the Resource Allocating Network Algorithm, Proceedings if the 10th Mediterranean Conference on Control and Automation, 2002.