

## A comparison of piecewise linear model descriptions

***Citation for published version (APA):***

Kevenaar, T. A. M., & Leenaerts, D. M. W. (1992). A comparison of piecewise linear model descriptions. *IEEE Transactions on Circuits and Systems*, 39(12), 996-1004. <https://doi.org/10.1109/81.207720>

***DOI:***

[10.1109/81.207720](https://doi.org/10.1109/81.207720)

***Document status and date:***

Published: 01/01/1992

***Document Version:***

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

***Please check the document version of this publication:***

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

***General rights***

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

***Take down policy***

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# A Comparison of Piecewise-Linear Model Descriptions

Tom A. M. Kevenaar and Domine M. W. Leenaerts

**Abstract**—In this paper current methods to store piecewise-linear mappings will be discussed. These methods are well-known in literature but have never been thoroughly compared. To be able to do so, the model descriptions will all be transformed into a most general form. Among the aspects of comparison will be the ease of modeling, the class of functions that can be modeled using a certain model description and the suitability to use the models in simulators. No best model description will be found but it will become apparent that some models are better suited for certain applications than others.

## I. INTRODUCTION

**M**OST SYSTEMS used in present day electronics are nonlinear. For small signal responses, these systems may be linearized in a chosen bias point and afterwards the system equations can be solved as if the system was linear. If, however, large signals are applied, the nonlinear behavior must be taken into account. Getting insight in nonlinear systems is not a trivial task and even for very small systems it is often not possible to solve the system equations analytically so that one has to resort to computers.

Electronic systems or networks are nonlinear because they are built from nonlinear (and linear) components. To be able to simulate these networks with either a circuit- or a network simulator, the nonlinear behavior of the components must be modeled first. In this modeling phase, properties of the component that are not considered important for the behavior of the system may be omitted or simplified. For example, in the MOSFET model used in SPICE during transient analysis, the noise of the MOSFET is not modeled although in the real transistor it is always present. Another example is found in logic simulators or switch-level simulators: although the models of the logic gates are rather simplified as compared to the real physical behavior, the results of the simulations are usually satisfactory.

If the model is derived, it often may still contain nonlinear effects which in general can not exactly be represented by familiar analytical nonlinear functions which can only be used inside a simulator. The next phase then is to approximate the nonlinear effect by a known nonlinear mathematical function.

Manuscript received April 30, 1992; revised September 3, 1992. This paper was recommended by Associate Editor M. Hasler.

The authors are with the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands.

IEEE Log Number 9205009.

The most obvious method perhaps is to use polynomial functions. In the past, a lot of research has been done in the area of approximating nonlinear functions using polynomials [1] as well as in the field of calculating transcendental functions. Applying such methods leads to simulators like SPICE [2]. A MOSFET is modeled as a connection of nonlinear branches that are each characterized by some polynomial or transcendental function.

Another way to store a nonlinear function is by means of table-look-up. Several sample points of a nonlinear function are stored in a table. If a function value is needed, the entry in the table is used or interpolated. The main advantage of this method is the speed of function evaluation that is very fast. In the SPICE-like simulators, a large part of the simulation time is consumed by evaluation of the nonlinear functions [3]. In contrast, for table-look-up models the memory needed to store a model is much larger. If technology changes, all the tables have to be generated again.

A third kind of nonlinear approximation is by using piecewise-linear (PL) functions: in the modeling phase, the nonlinear function is repeatedly approximated locally by a linear mapping. This results in a large collection of linear mappings that has to be stored. However, it appears to be possible to store the PL-mapping in a compact closed form model [4], [7]–[11]. These models can then be used in so called PL-simulators [5], [6], [18].

An additional advantage of modelling a nonlinear system with PL-functions is that it is easier to get insight in the behavior of these systems. As a drawback it can be mentioned that for non-scalar functions there is not always a direct way to find the parameters of the PL-mapping algorithmically as is the case with for example polynomial functions. In practice this problem appears to be not very severe.

In this paper several methods for storing PL-mappings in a closed form will be compared. This idea is prompted by the fact that recently two new models have been introduced in literature [7], [8].

One of the first models was given by Chua *cum suis* [9], [10] and will be referred to as Chua1. This description was followed a few years later by the model presented in [11] of which a modified form was given in [12]. These models will be referred to as Bokh1 and Bokh2, respectively.

An extension to Chua1 was recently given in [7] and will be called Chua2. The last model that will be treated here [8] is also an extension of Chua1 but differs from Chua2 and will be referred to as Güz1.

To be complete, the description given in [13] must also be mentioned. This one will not be treated here because it has several serious drawbacks as mentioned in [8].

The remaining of the paper is organized as follows. Section II starts with a short review of the fundamentals of PL-models and some definitions. This is followed by a description of the five models and some discussion. In Section III all the models will be transformed into the most general model description to be able to compare the models easily. In Section IV the comparison of the models will take place with respect to the ease of modelling and the suitability of the models to build simulators. In the last section, some conclusions about the models will be given.

## II. PIECEWISE-LINEAR MODELS

In this section, a survey will be given of existing piecewise linear models. The models of Chua [9], Güzelis [8], Kahlert [7], and van Bokhoven [11], [12] will be considered. To obtain more insight in the mathematical formulas, a review of some fundamentals will be given first.

### 2.1 Review of Some Fundamentals

All the PL functions to be considered are continuous functions, describing the mapping:

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m; \mathbf{x} \rightarrow f(\mathbf{x}) \quad (1)$$

unless otherwise is prescribed. Further, we will use CAPITAL notation for matrices and **bold** notation for vectors.

PL functions approximate the nonlinear behavior of a function using a collection of linear affine mappings. Each mapping is only valid within a certain subspace, called a polytope, which is a convex polyhedron in  $\mathbb{R}^n$ .

Such a polytope is bounded by a set of linear manifolds, called hyperplanes, with each hyperplane defined as

$$\mathbf{a}^T \mathbf{x} = 0 \quad (2)$$

where without loss of generality we assume that the hyperplane contains the origin. An important aspect in PL-modeling is the *consistent variation* property. To explain this, first the change of the mapping when crossing a hyperplane is treated.

Let us assume that the function  $f$  is defined in four regions  $R_1$ ,  $R_2$ ,  $R_3$ , and  $R_4$  with Jacobians  $J_1$ ,  $J_2$ ,  $J_3$ , and  $J_4$ , respectively (see Fig. 1). In the figure also the two hyperplanes  $H_1$  and  $H_2$  are depicted. Since  $f$  is continuous, the function values have to match at the boundaries  $H_1$  and  $H_2$ . For example, going from  $R_1$  to  $R_2$  crossing  $H_1$  means

$$\{\mathbf{x} \in H_1 | J_1 \mathbf{x} = J_2 \mathbf{x}\}. \quad (3)$$

The normal vector of  $H_1$ , denoted by  $\mathbf{a}_1$ , fully defines the boundary.

Since (3) holds for every  $\mathbf{x} \in H_1$  it can be proven [7], [14] that the following constraint is valid

$$J_1 - J_2 = \mathbf{c} \mathbf{a}_1^T \quad (4)$$

where  $\mathbf{c}$  is an arbitrary vector.

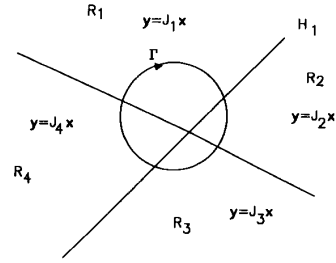


Fig. 1. A closed path  $\Gamma$  through the domain space.

Relation (4) implies that  $J_1$  changes with a dyadic vector product in  $J_2$  and that it is a valid description throughout the whole boundary. Relation (4) is in fact exactly the *consistent variation* property [7]. As a consequence it is clear that along a closed path  $\Gamma$  (see Fig. 1) the summation of the jumps, i.e., the changes in the Jacobians, over  $H_1$  and  $H_2$  must add up to zero.

Further, we will often use the term “degenerate” partitions. The definition of degeneration can be given in the following definition.

*Definition 1:* An intersection of  $(n - 1)$ -dimensional boundaries in  $\mathbb{R}^n$  is degenerate if three or more of them meet in an  $(n - 2)$ -dimensional manifold.

PL-models can be divided into two classes. The first class contains *explicit* models. For this class of models the output vector can be obtained at once (for a given input vector) by just substituting the input vector into the model.

The second class contains PL-models which are *implicit*. In such models the output vector can not be obtained immediately. First an algorithm has to be performed by which the output vector is computed.

The main goal of this paper is to compare the several PL-models. To be able to do this we will reformulate the explicit PL-models into the most general implicit PL-model, i.e., the model proposed by van Bokhoven [11]. To this purpose we define the modulus transformation,

*Definition 2:* The transformation  $z \rightarrow u, j$  defined by the function  $h(\cdot)$ :

$$u = h(|z| + z), \quad j = h(|z| - z)$$

with a strictly increasing  $h: \mathbb{R}_+ \rightarrow \mathbb{R}_+$  and  $h(0) = 0$ , is called the modulus transformation.

It is important to notice that application of the modulus transformation to a function  $j = j(u)$  automatically guarantees that  $u \geq 0$ ,  $j \geq 0$ , and  $u \cdot j = 0$ . To our purpose the function  $h(\cdot)$  will be defined as  $h(t) = t$ . For this situation is  $u = (|z| + z)$  and  $j = (|z| - z)$  and the sign of  $z$  determines either  $u = 0$  ( $j > 0$ ) or  $u > 0$  ( $j = 0$ ). This can be stated as follows.

*Corollary 1:* The modulus transform for  $h(t) = t$  can also be seen as a mapping  $u, j \rightarrow z$  satisfying  $|z| = (u + j)/2$  and  $z = (u - j)/2$ , with  $z \in \mathbb{R}$  and  $u, j \in \mathbb{R}_+$ .

Finally, the definition of the modulus of a vector is given as

*Definition 3:* Let  $\mathbf{a} \in \mathbb{R}^n$ , then  $|\mathbf{a}|$  is defined as

$$|\mathbf{a}| = (|a_1|, |a_2|, \dots, |a_n|)^T.$$

## 2.2 Explicit PL Models

In this section the most important explicit PL models will be considered. They all have a compact global analytical form.

**2.2.1. Model Chua1:** The formal definition of the canonical piecewise linear function  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ , proposed by Chua and Kang [9], [10] is expressed by

$$f(\mathbf{x}) = \mathbf{a} + B\mathbf{x} + \sum_{i=1}^{\sigma} c_i |\langle \alpha_i, \mathbf{x} \rangle - \beta_i| \quad (5)$$

where  $B \in \mathbb{R}^{m \times n}$ ,  $\mathbf{a}, c_i \in \mathbb{R}^m$ ,  $\alpha_i \in \mathbb{R}^n$ , and  $\beta_i \in \mathbb{R}^1$  for  $i \in \{1, \dots, \sigma\}$ . This model will be named Chua1.

From (5) several aspects can be observed.

First, the domain space is divided into a finite number of polyhedral regions by  $\sigma$  hyperplanes  $H$  of dimension  $n - 1$ . Hyperplane  $H_i$  is expressed by

$$\langle \alpha_i, \mathbf{x} \rangle - \beta_i = \alpha_i^T \mathbf{x} - \beta_i = 0 \quad (6)$$

which is also depicted in Fig. 2.

Crossing this hyperplane from region  $R_1$  into  $R_2$ , the Jacobian matrix changes with the amount  $-2c_i \alpha_i^T$ . This variation is independent of where  $H_i$  is crossed.

One can prove [10] that a PL function has a canonical representation like (5) if and only if it has linear partitions and satisfies the consistent variation property. For such functions the representation is unique for fixed-boundary parameters.

Further, it is important to notice that for each input vector there exists a unique output vector and that this vector always can be obtained. This implies that only functions in the strict sense can be modeled.

For one-dimensional functions ( $\mathbf{x} \in \mathbb{R}^1$ ) the hyperplanes reduce to points and hence each hyperplane only separates two regions in the domain space. Therefore, the consistent variation property is always satisfied and thus every one-dimensional PL-function can be represented by (5).

For multidimensional functions, some geometrical constraints always exist when  $\sigma > 1$ . The consistent variation property then results in a limited class of functions which can be described by (5).

**2.2.2. Model Güzl:** To be able to deal with a larger class of circuits than Chua1, Güzelis [8] developed a model in which some hyperplanes may be PL themselves.

This model, called Güzl, is defined as a mapping  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  with

$$f(\mathbf{x}) = \mathbf{s} + J\mathbf{x} + \sum_{i=1}^{\sigma} \mathbf{b}_i |\alpha_i^T \mathbf{x} + \beta_i| + \sum_{j=1}^{\tau} c_j |\delta_j + \gamma_j^T \mathbf{x} + \sum_{i=1}^{\sigma} d_{ji} |\alpha_i^T \mathbf{x} + \beta_i| \quad (7)$$

where  $\mathbf{s} \in \mathbb{R}^m$ ,  $J \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b}_i, c_j \in \mathbb{R}^m$ ,  $d_{ji} \in \mathbb{R}^1$ ,  $\alpha_i, \gamma_j \in \mathbb{R}^n$  and  $\beta_i, \delta_j \in \mathbb{R}^1$ .

From (7) one can observe that two kinds of boundaries are used. First there is a set of  $\sigma$  hyperplanes with

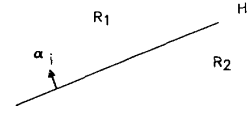


Fig. 2. Polytope boundary.

normal vectors  $\alpha_i$ ,  $i \in \{1, \dots, \sigma\}$ .

Secondly, there is a set of  $\tau$  piecewise affine (PWA) hyperplanes. These PWA hyperplanes are PL mappings themselves, i.e., PWA hyperplane  $j$  is expressed as

$$\delta_j + \gamma_j^T \mathbf{x} + \sum_{i=1}^{\sigma} d_{ji} |\alpha_i^T \mathbf{x} + \beta_i| = 0. \quad (8)$$

This canonical PL mapping (8) uses the first set of boundaries to define the PL-behavior of the PWA hyperplanes. This is visualized in Fig. 3 where  $H_1$  and  $H_2$  denote the first set of hyperplanes and  $C_1$  and  $C_2$  the PWA hyperplanes. In this way we have constructed nine regions to define a mapping.

It is proved in [8] that to have a description as (7) a PL mapping must satisfy the consistent variation condition, i.e., for each hyperplane or PWA hyperplane there exists a unique continuous vector.

From (7) it can be observed that when no PWA hyperplanes are used to compose the model, Güzl will be the same as Chua1. Using the PWA hyperplanes the model is more general than the one proposed by Chua and Kang.

**2.2.3. Model Chua2:** To overcome the shortcomings of Chua1, Kahlert and Chua [7] presented an extension to this model, which in this paper will be referred to as Chua2.

The complete canonical representation for PL-functions  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  with arbitrary region boundaries in two-dimensions can be given as

$$f(\mathbf{x}) = \mathbf{b} + B\mathbf{x} + \sum_{i=1}^{\sigma} c^i |\langle \alpha^i, \mathbf{x} \rangle - \beta^i| + \phi(\mathbf{x}) \quad (9)$$

with

$$\begin{aligned} \phi(\mathbf{x}) = & \sum_{j=1}^{\rho} \sum_{k=3}^{\delta^j} \tilde{c}_k^{j,i} \{ \|a_{k,j_1}^i (\langle \alpha^{j_1}, \mathbf{x} \rangle - \beta^{j_1})\| \\ & + a_{k,j_2}^i (\langle \alpha^{j_2}, \mathbf{x} \rangle - \beta^{j_2})\| \\ & - |a_{k,j_1}^i (\langle \alpha^{j_1}, \mathbf{x} \rangle - \beta^{j_1}) + |a_{k,j_2}^i (\langle \alpha^{j_2}, \mathbf{x} \rangle - \beta^{j_2})|\| \} \end{aligned} \quad (10)$$

where  $\mathbf{b}, c^i, \tilde{c}^{ijk} \in \mathbb{R}^m$ ,  $\alpha^i, \alpha^{j_1}, \alpha^{j_2} \in \mathbb{R}^n$ ,  $\beta^i, \beta^{j_1}, \beta^{j_2}, a_{k,j_1}^i, a_{k,j_2}^i \in \mathbb{R}^1$ , and  $B \in \mathbb{R}^{m \times n}$ . Recently in [25], a form with hyperplanes in higher dimensions is given that however in essence is the same as (9) and (10).

In order to get some insight in how this model is built, first assume that  $\phi(\mathbf{x}) \equiv \mathbf{0}$ . The model (9) then becomes equal to Chua1 which possesses the consistent variation property.

It was proven in [14] that if the partitioning of the domain space is nondegenerate (see Section 2.1), every PL-function in the strict sense can be given by this model.

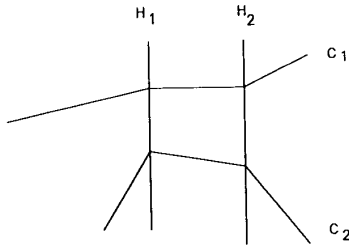


Fig. 3. Hyperplanes and PWA planes.

From this it is clear that trouble can be expected if there is a degenerate intersection that does not possess the consistent variation property.

An example, taken from [7], is given in Fig. 4(a). Clearly the intersection is degenerate and the consistent variation property is not satisfied so the function  $\phi(x) \neq 0$  is needed.

The first summation of  $\phi(x)$  ranges over all degenerate intersections of the partitioning and in this example can be omitted because there is only one intersection. The inner sum ranges, for every degenerate intersection, over all the planes causing the degeneracy. In the example, this sum contains one element because two planes do not cause degeneracy of that intersection. Thus the inner sum can be omitted too and the function  $\phi(x)$  gets a simple form.

Because the intersection is degenerate, it is possible to choose a subset of independent normal vectors (in this case two). The other normal vector can be formed by a linear combination. In the example  $\alpha^1$  and  $\alpha^2$  are chosen to form the independent set and  $\alpha^3 = a_1\alpha^1 + a_2\alpha^2$  and  $a_1 = a_2 = 1$ . The combination  $a_1\alpha^1 - a_2\alpha^2$  represents the plane  $x_2 = 0$ . By substituting a point from every region in  $\phi(x)$  it is possible to determine the value of  $\phi(x)$  which results in (Fig. 4(b))

$$\begin{aligned} R_1: 2\tilde{c}a_2\alpha^{2T}x \hat{=} J_\phi^1 \quad R_4: 2\tilde{c}a_2\alpha^{2T}x \\ R_2: -2\tilde{c}a_1\alpha^{1T}x \hat{=} J_\phi^2 \quad R_5: -2\tilde{c}a_1\alpha^{1T}x \\ R_3: 0 \quad R_6: 0. \end{aligned}$$

These terms are added to the consistent variation part of the mapping and give just enough freedom to remove this consistent variation constraint.

If the degeneracy is of a higher degree, several of these contributions have to be added, which accounts for the inner sum. The influences of all the degenerated intersections must also be added which is represented by the first sum. With this model every two dimensional PL-function in the strict sense can be modeled.

### 2.3. Implicit PL models

In this section two implicit PL-models will be treated. These PL-models use state variables to determine for which part of the domain space the mapping equations are valid.

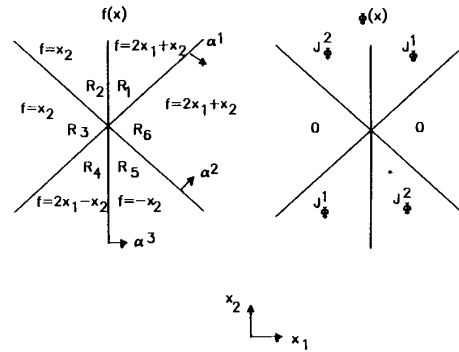


Fig. 4. An example for model Chua2.

2.3.1. Model Bokh1: In [11] van Bokhoven presented a PL model using state variables

$$y = Ax + Bu + f \tag{11}$$

$$j = Cx + Du + g \tag{12}$$

$$u \geq 0, j \geq 0, u^T j = 0 \tag{13}$$

to represent the PL mapping  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ , where  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{m \times k}$ ,  $f \in \mathbb{R}^m$ ,  $C \in \mathbb{R}^{k \times n}$ ,  $D \in \mathbb{R}^{k \times k}$ , and  $g \in \mathbb{R}^k$ . We will refer to this model as Bokh1.

In this model the vectors  $u$  and  $j$  are the state vectors. Relation (13) formulates the linear complementary condition of the state variables  $u$  and  $j$ . The state equation (12) describes the boundaries of the polytope and the linear mappings are collected in the system equation (11).

Suppose  $u = 0$  then relation (11) defines the linear mapping  $y = Ax + f$  valid for the polytope  $Cx + g \geq 0$ .

To obtain the output vector for a certain  $x_e$ , not lying in this polytope, some operations have to be performed.

Substituting  $x_e$  into (12) leads to

$$j = Du + q, u \geq 0, j \geq 0, u^T j = 0, q = Cx_e + g \tag{14}$$

which is known in the literature as the linear complementary problem (LCP).

Solving (14) means obtaining a set of  $u$  and  $j$  with some entries zero and other entries nonzero and satisfying property (13).

In the literature, several methods are discussed to solve the LCP problem [11], [15]–[17].

The vector  $u$  is then substituted into (11) to obtain the linear mapping  $y = \bar{A}x + \bar{f}$ , which is valid for  $x_e$ .

In Fig. 5, a hysteresis function is depicted for which the PL-model can be described as

$$y = (-1)x + (-1, 1)u + (1) \tag{15}$$

$$\begin{pmatrix} j_1 \\ j_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}x + \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}u + \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{16}$$

2.3.2. Model Bokh2: An other implicit PL-model was proposed by van Bokhoven in [12], where the mapping  $f$ :

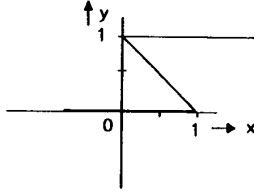


Fig. 5. Hysteresis function.

$\mathbb{R}^n \rightarrow \mathbb{R}^m$  is defined as

$$\mathbf{0} = I\mathbf{y} + A\mathbf{x} + B\mathbf{u} + \mathbf{f} \quad (17)$$

$$\mathbf{j} = D\mathbf{y} + C\mathbf{x} + I\mathbf{u} + \mathbf{g} \quad (18)$$

$$\mathbf{u} \geq \mathbf{0}, \mathbf{j} \geq \mathbf{0}, \mathbf{u}^T \mathbf{j} = 0 \quad (19)$$

where  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{m \times k}$ ,  $\mathbf{f} \in \mathbb{R}^m$ ,  $C \in \mathbb{R}^{k \times n}$ ,  $D \in \mathbb{R}^{k \times m}$  and  $\mathbf{g} \in \mathbb{R}^k$ .

In this model, further referred to as Bokh2, relation (17) contains the mapping equations and relation (18) the state equations.

The special property of this model is the fact that the hyperplanes are not only situated in the domain space but also in the co-domain space. However, because  $\mathbf{u}$  is multiplied by  $I$  only, the consistent variation property is always satisfied (in the (co)-domain space).

This unit matrix leads to advantages when simulating large systems. Using this model, the function of Fig. 5 can be given as

$$\mathbf{0} = (1)\mathbf{y} + (0)\mathbf{x} + (-2, 2)\mathbf{u} + (0) \quad (20)$$

$$\mathbf{j} = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \mathbf{y} + \begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \mathbf{x} + I\mathbf{u} + \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix}. \quad (21)$$

### III. RELATIONS BETWEEN THE MODELS

In this section all the above-mentioned models will be considered in relation to the implicit Bokh1. To this purpose we will use the modulus transform. Furthermore, it will be sketched that Güz1 and Chua2 can not always be transformed into Bokh2 and vice versa.

#### 3.1 Chua1

Let us start with Chua1 in (5). Reformulating this model using matrices leads to

$$f(\mathbf{x}) = \mathbf{y} = \mathbf{a}_1 + B_1\mathbf{x} + C_1|A_1\mathbf{x} - \mathbf{b}_1| \quad (22)$$

where the vectors  $\mathbf{c}_i$ ,  $\mathbf{a}_i$ , and elements  $\beta_i$  in (5) are placed as rows  $i$  in  $C_1$ ,  $A_1$  and  $\mathbf{b}_1$ , respectively.

Define  $\mathbf{z} = A_1\mathbf{x} - \mathbf{b}_1$ , then the modulus transformation (see Corollary 1, Section 2.1) yields,

$$|A_1\mathbf{x} - \mathbf{b}_1| = \frac{1}{2}(\mathbf{u} + \mathbf{j}) \quad (23)$$

$$A_1\mathbf{x} - \mathbf{b}_1 = \frac{1}{2}(\mathbf{u} - \mathbf{j}). \quad (24)$$

Relation (22) can be reformulated by substituting (23) and (24) into (22) and rewriting (24) to

$$\mathbf{y} = (B_1 - C_1A_1)\mathbf{x} + C_1\mathbf{u} + (\mathbf{a}_1 + C_1\mathbf{b}_1) \quad (25)$$

$$\mathbf{j} = (-2A_1)\mathbf{x} + I\mathbf{u} + (2\mathbf{b}_1) \quad (26)$$

which has the same form as Bokh1 with  $A = B_1 - C_1A_1$ ,  $B = C_1$ ,  $\mathbf{f} = (\mathbf{a}_1 + C_1\mathbf{b}_1)$ ,  $C = (-2A_1)$ ,  $D = I$  and  $\mathbf{g} = (2\mathbf{b}_1)$ .

From (25) and (26) one can observe that indeed Chua1 belongs to a subclass of the class Bokh1, because  $D$  has the special form  $D = I$ . Due to this unit matrix in front of  $\mathbf{u}$ , the consistent variation property is always satisfied.

Obviously, every implicit model in the form of Bokh1 with a unity matrix in front of vector  $\mathbf{u}$  in the state equation can be transformed into an explicit form.

#### 3.2 Güz1

Model Güz1, defined in (7), can be rewritten into

$$\mathbf{y} = \mathbf{s}_2 + J_2\mathbf{x} + B_2|A_2\mathbf{x} + \mathbf{b}_2| + C_2|\mathbf{d}_2 + G_2\mathbf{x} + D_2|A_2\mathbf{x} + \mathbf{b}_2| \quad (27)$$

using the same technique as in reformulating (5) into (22).

As before, define for this model

$$|A_2\mathbf{x} + \mathbf{b}_2| = \frac{1}{2}(\mathbf{u}_1 + \mathbf{j}_1) \quad (28)$$

$$A_2\mathbf{x} + \mathbf{b}_2 = \frac{1}{2}(\mathbf{u}_1 - \mathbf{j}_1) \quad (29)$$

$$|\mathbf{d}_2 + G_2\mathbf{x} + D_2|A_2\mathbf{x} + \mathbf{b}_2| = \frac{1}{2}(\mathbf{u}_2 + \mathbf{j}_2) \quad (30)$$

$$\mathbf{d}_2 + G_2\mathbf{x} + D_2|A_2\mathbf{x} + \mathbf{b}_2| = \frac{1}{2}(\mathbf{u}_2 - \mathbf{j}_2) \quad (31)$$

then, by substituting (28) and (30) into (27) and solving  $\mathbf{j}$  from (29) and (31), (27) can be transformed into

$$\begin{aligned} \mathbf{y} &= (J_2 - B_2A_2 - C_2G_2 + C_2D_2A_2)\mathbf{x} \\ &\quad + (B_2 - C_2D_2|C_2) \cdot \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} + (\mathbf{s}_2 - B_2\mathbf{b}_2 - C_2\mathbf{b}_2 + C_2D_2\mathbf{b}_2) \quad (32) \\ \begin{pmatrix} \mathbf{j}_1 \\ \mathbf{j}_2 \end{pmatrix} &= \begin{pmatrix} -2A_2 \\ -2G_2 + 2D_2A_2 \end{pmatrix} \mathbf{x} \\ &\quad + \begin{pmatrix} I & \mathbf{0} \\ -2D_2 & I \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} + \begin{pmatrix} -2\mathbf{b}_2 \\ -2\mathbf{d} + 2D_2\mathbf{b}_2 \end{pmatrix} \quad (33) \end{aligned}$$

which has the same form as Bokh1. Here  $\mathbf{0}$  is a matrix with entries equal zero. This means that there is a one-to-one relation between the coefficients of Güz1 and Bokh1. This means that the statement in [8] that there exists no systematic method to obtain the coefficients of Bokh1 is not true for the class of functions that can be modeled by Güz1.

Note that the total matrix  $D$  has a special form and can not be chosen freely. The first set of equations (i.e., with  $\mathbf{u}_1$  and  $\mathbf{j}_1$ ) denotes a set of hyperplanes in the strict sense. The second set consists of the PWA planes which indeed by matrix  $D_2$  depends on the first set of planes.

Obviously, Güz1 belongs to a superclass of Chua1 and belongs to the same class if the PWA hyperplanes are excluded, i.e.,  $D_2$  equals the zero matrix.

Model Güz1 is of a smaller class than Bokh1 and it is not possible to reformulate each Bokh1 into the explicit Güz1. Consider for instance the hysteresis function (15)–(16), which has a full matrix  $D$ .

### 3.3 Chua2

Also, Chua2 can be reformulated into Bokh1. To this purpose redefine Chua2 as

$$\begin{aligned} \mathbf{f}(\mathbf{x}) = \mathbf{y} = & \mathbf{b}_3 + B_3\mathbf{x} + C_3^a|A_3^a\mathbf{x} - \mathbf{d}_3^a| + C_3^1|A_3^1\mathbf{x} - \mathbf{d}_3^1| \\ & + C_3^2|A_3^2\mathbf{x} - \mathbf{d}_3^2| \\ & + \tilde{C}_3\{\|E_3(A_3^1\mathbf{x} - \mathbf{d}_3^1)| + F_3(A_3^2\mathbf{x} - \mathbf{d}_3^2)\| \\ & - \|E_3(A_3^1\mathbf{x} - \mathbf{d}_3^1) + F_3(A_3^2\mathbf{x} - \mathbf{d}_3^2)\|\} \quad (34) \end{aligned}$$

where the vectors  $c^i$ ,  $\alpha^i$ , and elements  $\beta^i$  in (9) are placed as rows  $i$  in  $C_3^a$ ,  $A_3^a$  and  $\mathbf{d}_3^a$ , respectively. The superscript 1 (2) denotes the information belonging to the first (second) independent normal vectors  $\alpha^{j1}$  ( $\alpha^{j2}$ ) as mentioned in Section 2.2.3..

Using the same technique as for reformulating Güz1 into Bokh1, this equation can be transformed into

$$\begin{aligned} \mathbf{y} = & (B_3 - C_3^a A_3^a - C_3^1 A_3^1 - C_3^2 A_3^2 + 2\tilde{C}_3 G)\mathbf{x} \\ & + (C_3^a, C_3^1, C_3^2, -\tilde{C}_3, \tilde{C}_3, \tilde{C}_3, -\tilde{C}_3)\mathbf{u} \\ & + (\mathbf{b}_3 + C_3^a \mathbf{d}_3^a + C_3^1 \mathbf{d}_3^1 - 2\tilde{C}_3 \mathbf{g}) \quad (35) \end{aligned}$$

$$\begin{aligned} \mathbf{j} = & \begin{bmatrix} -2A_3^a \\ -2A_3^1 \\ -2A_3^2 \\ -2E_3 A_3^1 \\ -2F_3 A_3^2 \\ 2G \\ -2G \end{bmatrix} \mathbf{x} + \begin{bmatrix} I & & & & & & \\ & I & & & & & \\ & & I & & & & \\ & & & I & & & \\ & & & & -2I & & \\ & & & & & -2I & \\ & & & & & & I \end{bmatrix} \mathbf{u} \\ & + \begin{bmatrix} 2\mathbf{d}_3^a \\ 2\mathbf{d}_3^1 \\ 2\mathbf{d}_3^2 \\ 2E_3 \mathbf{d}_3^1 \\ 2F_3 \mathbf{d}_3^2 \\ -2\mathbf{g} \\ 2\mathbf{g} \end{bmatrix} \quad (36) \end{aligned}$$

with  $G = (E_3 A_3^1 - F_3 A_3^2)$ ,  $\mathbf{g} = (E_3 \mathbf{d}_3^1 - F_3 \mathbf{d}_3^2)$ .

In (36) two types of hyperplanes are described. The first set contains hyperplanes in the strict sense. The last two equations in (36) describe hyperplanes whose normal vectors are dependent of equalities 4 and 5 in (36). This is to conform to the theory of the model that for each degenerated section two independent normal vector are chosen

(stored in 4 and 5 and the other normal vectors for the equality section can be formed as a linear combination of these vectors (and stored in the last two equations).

It is obvious from (36) that Chua2 is of a larger class than Chua1. Chua2 is also larger than Güz1 due to the fact that in Güz1 the PWA hyperplanes can not be formed without restrictions as seen in (8). In Chua2 there is no restriction in forming the hyperplanes.

Also, this model is a subclass of Bokh1. The matrix in front of the state vector  $\mathbf{u}$  is not a full matrix, and for instance, again the hysteresis function (15)–(16) can not be modeled using this model.

### 3.3. Bokh2

Finally the last presented model, i.e., Bokh2, will be transformed into Bokh1. To this purpose, only the mapping equation has to be substituted into the state equation leading to

$$\mathbf{y} = -A_4\mathbf{x} - B_4\mathbf{u} - \mathbf{f}_4 \quad (37)$$

$$\mathbf{j} = (C_4 - D_4 A_4)\mathbf{x} + (I - D_4 B_4)\mathbf{u} + (\mathbf{g}_4 - D_4 \mathbf{f}_4) \quad (38)$$

which has the same structure as (11) and (12) with  $C = (C_4 - D_4 A_4)$ ,  $D = (I - D_4 B_4)$  and  $\mathbf{g} = (\mathbf{g}_4 - D_4 \mathbf{f}_4)$ .

With  $D_4$  equal to zero, the state equation will have the same form as Chua1. Therefore, if hyperplanes are taken from the domain space only, Bokh2 is of the same class as Chua1.

However, if  $D_4$  is not equal to zero matrix, this model is more powerful than that of Chua1. Consider for example again, the hysteresis function (20)–(21) which has matrix  $D \neq 0$  and can not be modeled by Chua1. In contrast it can be seen that Bokh1 can not always be transformed into Bokh2: if  $A_4$ ,  $B_4$ , and  $\mathbf{f}_4$  are chosen,  $D_4$  must be chosen such that  $I - D_4 B_4 \equiv D$ . The degrees of freedom in  $D_4$  are not always sufficient to do this.

Compared to Güz1 this model is of a different class meaning that there are functions that be represented by Güz1 and not by Bokh2 and vice versa. To show this let for Bokh2 be given that

$$D_4 = \begin{pmatrix} D_{41} \\ D_{42} \end{pmatrix}, C_4 = \begin{pmatrix} C_{41} \\ C_{42} \end{pmatrix}, \mathbf{g}_4 = \begin{pmatrix} \mathbf{g}_{41} \\ \mathbf{g}_{42} \end{pmatrix}, B_4 = (B_{41}|B_{42})$$

Then (37) and (38) can be written in the form Bokh1

$$\mathbf{y} = -A_4\mathbf{x} - (B_{41}|B_{42})\mathbf{u} - \mathbf{f}_4$$

$$\begin{aligned} \begin{pmatrix} \mathbf{j}_1 \\ \mathbf{j}_2 \end{pmatrix} = & \begin{pmatrix} C_{41} - D_{41} A_4 \\ C_{42} - D_{42} A_4 \end{pmatrix} \mathbf{x} \\ & + \begin{pmatrix} I - D_{41} B_{41} & -D_{41} B_{42} \\ -D_{42} B_{41} & I - D_{42} B_{42} \end{pmatrix} \mathbf{u} + \begin{pmatrix} \mathbf{g}_{41} - D_{41} \mathbf{f} \\ \mathbf{g}_{42} - D_{42} \mathbf{f} \end{pmatrix} \end{aligned}$$

which under the conditions  $D_{42} B_{41} = D_2$ ,  $D_{41} B_{41} = \mathbf{0}$ ,  $D_{41} B_{42} = \mathbf{0}$  and  $D_{42} B_{42} = \mathbf{0}$  (and  $\mathbf{0}$  the matrix with entries equal zero) can be made identical to (32) and (33) which is a representation of Güz1. It can be shown that the com-

bined matrix  $(D_{41}|D_{42})^T$  does not necessarily contain enough degrees of freedom to fulfill the conditions just given. This means that it is not always possible to transform  $G_{\text{ü}1}$  into  $B_{\text{ok}2}$ . Because  $G_{\text{ü}1}$  can not describe nonhomeomorphic functions, the opposite is also true.

Because  $Chua2$  is of a larger class than  $G_{\text{ü}1}$  but also can only model homeomorphic functions, the same statements hold for the transformation of  $Chua2$  into  $B_{\text{ok}2}$  or the other way around.

### 3.4. Remarks

Let us briefly summarize the obtained results. It appears that all mentioned PL-models can be transformed into the most general model, i.e.  $B_{\text{ok}1}$  using a modulus transformation. Further, it is now possible to order the models in context to classes in which they belong.

Model  $Chua1$  belongs to the smallest class, only functions with hyperplanes in the strict sense can be modeled. Besides this the function must have the consistent variation property. A larger class of functions can be modeled with  $G_{\text{ü}1}$ . In this model the class is extended by means of PWA hyperplanes. A further extension in  $Chua2$ , able to model functions which do not possess the consistent variation property.  $B_{\text{ok}2}$  differs from the other models by the fact that the hyperplanes are constructed in the (co)-domain space.

It was shown that  $Chua1 \subset G_{\text{ü}1} \subset Chua2 \subset B_{\text{ok}1}$  and  $Chua1 \subset B_{\text{ok}2} \subset B_{\text{ok}1}$ . Furthermore, it was shown that  $B_{\text{ok}2} \not\supset G_{\text{ü}1}$ ,  $Chua2$  and  $B_{\text{ok}2} \not\subset G_{\text{ü}1}$ ,  $Chua2$ . The main difference between the explicit models  $Chua1$ ,  $Chua2$ ,  $G_{\text{ü}1}$  on one hand and  $B_{\text{ok}1}$ ,  $B_{\text{ok}2}$  on the other hand is that with the latter two it is possible to describe nonhomeomorphic functions. It can also be seen that any model  $B_{\text{ok}1}$  with a triangular  $D$  matrix can be written as an explicit relation that looks like a stacked or recursive  $Chua1$  model.

From this it is clear that of the five descriptions just given,  $B_{\text{ok}1}$  is the most general form. Therefore, application of this model will be of advantage in deriving theoretical results on the existence and the number of solutions of PL-mappings and for stating general properties of PL-mappings in terms of coefficients.

An important property of the  $B_{\text{ok}1}$  description is that it can explicitly produce a  $B_{\text{ok}1}$  model of the inverse mapping which in general is not possible for any of the other models.

## IV. THE MODEL COMPARISON

In the previous chapter, all the model descriptions were transformed into the most general form, i.e.,  $B_{\text{ok}1}$ . This is convenient to compare the models with respect to the class of functions that can be modeled using a certain description. There are, however, other aspects by which models may be compared. These will be treated in subsequent paragraphs.

### 4.1. Finding the Model Parameters

The first step in finding one of the model descriptions is to find a PL-mapping that approximates a nonlinear func-

tion for some given error criterion. Unfortunately, very few practical methods are known from the literature. For scalar functions a Tschebychev approximations can be used [23] that can be mapped directly on a PL-function. For multidimensional functions, no algorithm is known to find the PL-mapping in an algorithmic way. A theorem given by Kolmogorov [24] states that it is possible to realise a multi-dimensional function using a two level nesting of scalar functions and addition but the proof is not constructive and it is not known how these scalar functions must be obtained.

Strangely enough, this appears not to be a serious problem in practice. If 2-terminal components are used for circuit level simulation, the Tschebychev approximation can be used. Several of these methods can then be used to find the more complex model of, e.g., a MOSFET or bipolar transistor. For macromodels, the nonlinearities are usually not given as polynomials but as some kind of behavioral description that tends to become "more PL" as the size of the system increases. Moreover, in a lot of applications it is sufficient to use linear macromodels.

The problems just mentioned and outside the scope of this paper. Here we will assume that the PL-mapping is given.

For the models  $Chua1$ ,  $Chua2$ , and  $G_{\text{ü}1}$  an algorithm is given in [10], [7], and [8], respectively, to find the model parameters for a given PL-mapping. In chapter 3 it was shown that these models can be transformed into  $B_{\text{ok}1}$ . This means that it is possible for the class of functions that can be covered by the first three models, it is also possible to find the parameters for  $B_{\text{ok}1}$  in an algorithmic way. For the remaining functions that can be modeled by  $B_{\text{ok}1}$  no algorithm is known yet. These models are usually found by connecting several models and eliminating the internal variables. For example, the hysteresis function from Fig. 5 and the associated model (15)–(16) can be found by applying positive feedback on an op-amp as depicted in Fig. 6(a) where of course the saturating op-amp is modeled PL. For  $B_{\text{ok}2}$  it is straightforward to derive the model if the mappings and the partition in the (co)-domain are given. In Fig. 6(b), the hyperplanes and the mapping are given from which it is easy to find (20), (21).

### 4.2. Building PL-Simulators

Network- or circuit simulators that use PL-functions to describe the nonlinearities of the components are called PL-simulators. With all of the five models it is possible to build PL-simulators. Of the three oldest models  $Chua1$ ,  $B_{\text{ok}1}$ , and  $B_{\text{ok}2}$  simulators have been built that were reported in literature [18], [5], [22], and [6], respectively. The remaining models  $G_{\text{ü}1}$  and  $Chua2$  are quite recent and no results of building simulators using these models were reported yet.

*4.2.1. A Simulator Using Chua1:* Chua and Ying report in [18] that a simulator was built using  $Chua1$ . The restriction of this simulator is that it can only handle networks built from (non)linear controlled sources and 2-terminal



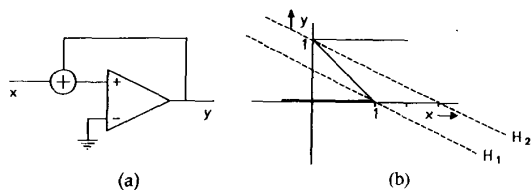


Fig. 6. Construction of the hysteresis function and the hyperplanes in the (co)-domain space.

resistors. In practice, this appears to be not a very serious limitation if networks at the circuit level are to be simulated. However, for larger systems using high-level macromodels this imposes a severe restriction. Furthermore, Chua1 may describe multidimensional functions. This means that for 2-terminal devices only a limited part of the potential of Chua1 is used.

Using one of the conventional circuit analysis methods (cut set, mesh, modified nodal, etc.) the circuit equations are transformed into the form of Chua1 where the  $x$  contains a set of network variables. The size of  $x$  depends on the chosen circuit analysis method. As with normal network analysis, the PL-equations  $f(x) = 0$  are then solved using a method introduced by Katzenelson. The global convergence properties of this method are better than of Newton-Raphson's method. With extensions to Katzenelson given in [19], this method will always find a solution for  $x$ . This is also mainly due to the fact that Chua1 can only model functions in the strict sense. If the hyperplanes of the compound model have a special structure (i.e., lattice) a more efficient method called breakpoint-hopping can be used [20], [21].

**4.2.2. Simulators using Bokh1:** With Bokh1, two simulators were built reported in [5] and [22]. The implicit Bokh1 is more powerful than Chua1 and its implicitness implies that it needs a powerful algorithm to find a solution. Although Katzenelson could be used, stronger methods are available such as Lemke [16] and Van de Panne [17] which are functionally equal. The models that can be used in these simulators are not restricted to two terminal devices and this is important if macromodels must be used.

Bokh1 allows to transform a number of connected models into a model of the same form without any of the internal variables of the network being present. This opens the possibility to derive compact models of complicated functions which is not possible for Chua1. An other possibility is to simply concatenate several models to form again a model of the form Bokh1 and add topological equations. This is the method used in PLATO [22]: the whole network is transformed into one large PL-model that is stored using sparse matrix techniques. A solution is then obtained using the Van de Panne algorithm. A different approach is followed in [5]: the hierarchy in the network (if present) is retained by storing a Jacobian for every node in the hierarchy tree.

Both methods suffer from the fact that implicitly, the tableau method is used to solve the network equations.

This means that all the voltages and currents in the network are calculated which is noticeable in the CPU times.

**4.2.3. A Simulator using Bokh2:** Recently a new PL-simulator called PLANET was developed [6], which uses Bokh2. As mentioned in chapter 3, this model is slightly less general than Bokh1 but stronger than Chua1, and comparable to Chua2 and Güz1. Bokh2 as used in PLANET is not restricted to 2-terminal devices so macro models can be used. In general, it is not possible to transform several connected models into a Bokh2 without any of the internal variables being present. In PLANET the hierarchy of the network to be simulated is retained and the models are kept separated from the topological equations. The main advantages compared to the simulators of the previous paragraph is that modified nodal analysis is used which causes a reduction in the number of variables that has to be calculated. The node voltages of the network are solved using a hierarchical implementation of the extended Katzenelson algorithm.

Furthermore, due to the hierarchy and the separation of topological equations and models, it is easy to modify the network which is useful in an automatic synthesis environment.

## V. CONCLUSIONS

In the previous chapters we have compared several PL-model descriptions. In chapter 3 an ordering was given in the context of the class of functions that can be modeled. It appeared that the most essential difference was that Bokh1 and Bokh2 are able to model nonhomeomorphic functions which is advantageous if one wants to use macromodels. As a consequence, it is not possible to rewrite these models in an explicit form. It is, however, possible by applying the modulus transform to rewrite Chua1, Chua2 and Güz1 into Bokh1 where the latter is in general not canonical.

In chapter 4 the models were compared with respect to finding the model parameters. It turned out that for functions in the strict sense it is possible to derive the models by means of an algorithm.

With the models several simulators were reported in literature each one with its own advantages and disadvantages. One of the advantages of PL-modeling is the ease of performing mixed-level simulations because simplified macro models can be used. On the circuit level it is not likely that PL-simulators can ever compete with simulators like SPICE. To keep PL-simulators as fast as possible, it is best to compute as little network variables as necessary.

It is clear that no best model can be given. It depends heavily on the application in which the model is to be used. It is clear, however, that Bokh1 is the most general form and is therefore best suited to derive theoretical results about PL-mappings.

The main contribution of this paper is that we have tried to compare several PL-models as objective as possible.

## ACKNOWLEDGMENT

The authors wish to thank W. M. G. van Bokhoven for valuable discussions.

## REFERENCES

- [1] R. W. Hamming, *Numerical Methods for Scientists and Engineers*. New York: McGraw-Hill, 1962.
- [2] A. Vladimirescu and S. Liu, "The simulation of MOS integrated circuits using SPICE 2," *Memorandum UCB / ERL M80 / 7*, Electronics Research Laboratory, College of Eng., Univ. Berkeley, Feb. 1980.
- [3] A. R. Newton and A. L. Sangiovanni-Vincentelli, "Relaxation based electrical simulation," *IEEE Trans. Computer-Aided Design*, vol. 3, pp. 308–331, 1984.
- [4] L. O. Chua and A. Deng, "Canonical piecewise linear modeling," *IEEE Trans. Circuits Syst.*, vol. 33, May, pp. 511–525, 1986.
- [5] J. T. J. van Eijndhoven, "A piecewise linear simulator for large scale integrated circuits," Ph.D. dissertation, Eindhoven Univ. Technology, Eindhoven, The Netherlands, 1984.
- [6] T. A. M. Kevenaer and D. M. W. Leenaerts, "A flexible hierarchical piecewise linear simulator," *Integration the VLSI Journal*, vol. 12, pp. 211–235, 1991.
- [7] C. Kahlert and L. O. Chua, "A generalized canonical piecewise linear representation," *IEEE Trans. Circuits Syst.*, vol. 37, Mar., pp. 373–382, 1990.
- [8] G. Güzelis and I. Gökner, "A canonical representation for piecewise affine maps and its applications to circuit analysis," *IEEE Trans. Circuits Syst.*, vol. 38, Nov., pp. 1342–1354, 1991.
- [9] L. O. Chua and S. M. Kang, "Section-wise piecewise linear functions: Canonical representation, properties and applications," *Proc. IEEE*, vol. 65, pp. 915–929, June 1977.
- [10] S. M. Kang and L. O. Chua, "A global representation of multidimensional piecewise linear functions with linear partitions," *IEEE Trans. Circuits Syst.*, vol. 25, Nov., pp. 938–940, 1978.
- [11] W. van Bokhoven, *Piecewise Linear Modelling and Analysis*. Delft, The Netherlands: Kluwer Technische Boeken, 1981.
- [12] W. M. G. van Bokhoven, "Piecewise linear analysis and simulation," in *Circuit Analysis, Simulation and Design*, A. E. Ruehli (Ed.). Amsterdam: North-Holland, 1986, Ch. 9.
- [13] J. L. Huertas, A. Rueda, and A. Rodriguez-Vaguez, "Multi-dimensional piecewise linear partitions: global canonical representation," in *Proc. ISCAS*, pp. 1106–1109, 1984.
- [14] L. O. Chua and A. C. Deng, "Canonical piecewise linear representation," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 101–111, Jan. 1988.
- [15] J. Katzenelson, "An algorithm for solving nonlinear resistive networks," *Bell Syst. Tech. J.*, vol. 44, pp. 1605–1620, Oct. 1965.
- [16] C. E. Lemke, "On complementary pivot theory," in *Nonlinear Programming (Proc. Symp.)*, J. B. Rosen, O. L. Mangasarian, K. Ritten, eds. New York: Academic Press, pp. 349–384, 1968.
- [17] C. van de Panne, "A complementary variant of Lemke's method for the linear complementary problem," *Math. Program.*, vol. 7, pp. 283–310, 1974.
- [18] L. O. Chua and R. P. Ying, "Canonical piecewise linear analysis," *IEEE Trans. Circuits Syst.*, vol. 30, pp. 125–140, Mar. 1983.
- [19] M. J. Chien and E. S. Kuh, "Solving piecewise linear equations for resistive networks," *Int. J. Circuit Theory and Applications*, vol. 4, pp. 3–24, 1976.
- [20] L. O. Chua and A. Deng, "Canonical piecewise linear analysis: Part II—Tracing driving point and transfer characteristics," *IEEE Trans. Circuits Syst.*, vol. 32, pp. 417–443, 1985.
- [21] —, "Canonical piecewise linear analysis: Generalized breakpoint hopping algorithm," *Int. J. Circuit Theory and Applications*, vol. 14, pp. 35–52, 1986.
- [22] M. T. van Stiphout, "PLATO: A piecewise linear analysis tool for mixed level circuit simulation," Ph.D. dissertation, Tech. Univ. Eindhoven, The Netherlands, 1990.
- [23] E. W. Cheney, *Introduction to Approximation Theory*. New York: McGraw Hill, 1966.
- [24] A. N. Kolmogorov, "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition," *Dokl. Akad. Nauk. SSSR 114*, pp. 369–373, 1957.
- [25] C. Kahlert and L. O. Chua, "The complete canonical piecewise-linear representation—Part I: The geometry of the domain space," *IEEE Trans. Circuits Syst.—I*, vol. 39, pp. 222–236, Mar. 1992.



**Tom Kevenaer** received the Ir. degree in electrical engineering in 1986 from the Eindhoven University of Technology, Eindhoven.

After his military service he joined Philips Labs for a project on a compact switchmode power supply. In 1989 he joined the electronic design group of the Eindhoven University of Technology as a Ph.D. student. His interests are in the field of circuit simulation, timing verification, and automation of analog design.



**Domine Leenaerts** received the Ir. and a Ph.D. degrees in electrical engineering in 1987 and 1992, respectively, both in the Department of Electrical Engineering at the Eindhoven University of Technology.

Since May 1992, he has been with this university as a research member. His research interests are in design of analog integrated circuits, nonlinear circuit theory, and in analog design automation.