Abstract—This paper presents a multi-domain design approach combining High Frequency (HF) and digital design. Simulated or measured S-parameters of a designed HF network are converted to a VHDL-AMS model, which can be combined with the digital part of the system in a RF-digital co-design environment. A chipless RFID system is studied as a case study. Comparing simulation results produced by ADS and AMS models validate capability of the proposed method. This method can be used to combine a variety of disciplines with HF design environments.

Index Terms—Chipless RFID, high frequency design, mixed-mode design, scattering parameters, VHDL-AMS modeling.

I. INTRODUCTION

Advances in electronic design technology and market demands are pushing the industry and consequently the research field towards more inclusive and efficient multidisciplinary system level design tools and methods. System level design approaches normally require design combinations between different disciplines over a diverse range of specifications. Although system level design methods have been able to fill the gaps between some design fields, engineers often need to be involved with specification transfers between different (Computer Aided Design) CAD tools, which might need deep understanding of the different fields.

High Frequency (HF) systems in combination with low-power/high-speed electronic circuits design have found a wide range of applications. Although, both of which are well developed in terms of theory, tools and applications; it would not be a trivial task to co-design, simulate or optimize the system considering both fields at the same time, due to their different natures. Therefore, we either need to focus on new tools for modeling of such mixed mode circuits ignoring very well developed older CAD tools; or utilize current tools and try to fill the gap using mixed-mode languages. Accordingly, most of attempts present methods to automatically generate physical plausible (Analog Mixed Signal) AMS description of all or a part of the mixed-mode system.

There are similar works in which solutions are presented to combine different design fields. Frank et al in [1] present a solution which tries to link between SPICE and VHDL-AMS simulators via an interface. Synchronization and data exchange between the simulators is the main challenge in this method that can decrease speed of the simulation and also generality of the method. Another approach, which tries to convert VHDL-AMS models to models that can be defined in traditional CAD tools, is presented in [2] but since traditional tools are not designed to simulate mixed-mode systems efficiently, this method cannot be very effective when the system complexity increases. Similarly, automatically generation of AMS models for power electronics systems [3], [4], Analog to Digital Converters (ADC’s) [5] and MATLAB/SIMULINK ADC models [6] are examples of this general approach. There are other works trying to present more general approaches for automated system and fault modeling [7], [8]. Each of the mentioned works have proposed different CAD tools in their special fields but to best of our knowledge there is no work on RF systems AMS modeling using Scattering parameters (S-parameters).

The Advanced Design System (ADS) of Agilent Technologies is a popular and powerful tool for HF designers with the ability to be used for a variety of designs. This paper proposes a design method that can help digital and HF designers to co-design a mixed-mode system. The final system is described in VHDL-AMS, where the digital part can be easily extended to MEMS, power electrical, mechanical and any other system that can be describable by VHDL-AMS, which is a multi-domain description language and does not have the limitations of the tools such as ADS. This approach can lead to a productivity gain by letting engineers design both digital and HF components together as these components would exist on the final system. This higher level of abstraction gives the design team a fundamental understanding early in the design process of the intricacies and interactions of the entire system and enables better system tradeoffs, better and earlier verification, and overall productivity gains through reuse of early system models as executable specifications.

In the proposed method, at first, the HF circuit is designed on ADS to satisfy design constraints. Then the simulated S-parameters of the circuit are used to generate VHDL-AMS model. The generated model is then delivered to a digital designer, to use the model as a black box in the final designs and completes the system model. This method is implemented as an
open access tool (called HF/Digital converter) to the community for further investigations, research and improvements.

Paper is organized as follows: Section II introduces the proposed method, its usage, considerations and limits. In Section III, design procedure and operating fundamentals of a chipless RFID system are discussed, where different parts of the modeled system are described and simulated to show the model validity and performance of the method. Paper is concluded in Section IV.

II. PROPOSED METHOD

The proposed method starts with design constrains of a mixed-mode system consists of digital and HF parts. Initially, HF and digital design are examined for their limitations and requirements. The way HF and digital parts connect and communicate to each other is also decided in this step. Then the HF and digital circuit designers start to work independently. The design flow of the proposed method is presented in Fig. 1.

As it can be seen in the Fig. 1, HF designer may design a new circuit or use an existing design, but in both of these cases the HF circuit should be described by its scattering parameters in a text file that can be generated by a HF simulator such as ADS or by any other tool. The VHDL-AMS model of the HF circuit will be created from this file. Block diagram of Fig. 2 shows behavioral specification of the generated model by this tool. The proposed model is based on the S-parameters ($S_{11}$, $S_{12}$, $S_{21}$, and $S_{22}$) of the HF design. These parameters are transformed into a matrix representation of the magnitude and phase for different frequency responses. This matrix is used to generate the VHDL-AMS model of the HF network. The model of each HF network is a behavioral, frequency domain model with three input and two output ports. Input ports get frequency, magnitude and phase of the input signal and the output ports produce the magnitude and phase of the output signal, all as real numbers. This model in combination with the model of other parts (digital, MEMS, energy harvesters etc.) is used to verify the design process.

As it is seen in Fig. 2, this method produces VHDL-AMS codes from original HF description, i.e. S-parameters, given from ADS or other tools in the form of lookup tables and utilizes them directly to produce the behavioral model.

Another way, which might be even shorter and optimum, is: the input data charts of HF description are converted to analytical functions describing the HF behavior to be used to produce the code to model the system. In this case, there is a function for the magnitude and a function for the phase of each S-parameter.

Analytical functions are extracted by a curve fitting algorithm. Block diagram of Fig. 3 shows behavioral specification of the generated model by in this approach using fitted function of S-parameters.
General form of the proposed functions for S-parameters is presented in (1) in which \( F(x) \) is a function of frequency. Therefore, each S-parameter description of the HF system is written in the form of generalized functions with different coefficients in compact matrixes forms as:

\[
F(x) = \sum_{j=1}^{k} a_{ij} x^{j-1} - \sum_{l=1}^{m} b_{ij} x^{l-1} = \sum_{i=1}^{k} f_i(x) = \sum_{i=1}^{m} g_i(x) \tag{1}
\]

Where \( x \) represents frequency and coefficients of \( F(x) \) are expressed in the matrix form presented in (2a), (2b) as:

\[
A_i = \begin{bmatrix} a_{i1} & \cdots & a_{il} \\ \vdots & \ddots & \vdots \\ a_{ki} & \cdots & a_{km} \end{bmatrix}, \tag{2a}
\]

\[
B_i = \begin{bmatrix} b_{i1} & \cdots & b_{il} \\ \vdots & \ddots & \vdots \\ b_{ki} & \cdots & b_{km} \end{bmatrix}. \tag{2b}
\]

\( A_{ij} \) are the nominator coefficients and \( B_{ij} \) are the denominator coefficients in (1). And in (1) can be calculated as (3a-3d):

If \( X_n = \begin{bmatrix} x^0 \\ x^1 \\ \vdots \\ x^n \end{bmatrix}, X_m = \begin{bmatrix} x^0 \\ x^1 \\ \vdots \\ x^m \end{bmatrix} \),

And

\[
V_f = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{bmatrix}, V_g = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_k(x) \end{bmatrix} \tag{3b}
\]

then:

\[
V_f = A_j \times X_n = \begin{bmatrix} a_{i1} & \cdots & a_{il} \\ \vdots & \ddots & \vdots \\ a_{ki} & \cdots & a_{km} \end{bmatrix} \begin{bmatrix} x^0 \\ x^1 \\ \vdots \\ x^n \end{bmatrix} = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{bmatrix}. \tag{3c}
\]

\[
V_g = B_j \times X_m = \begin{bmatrix} b_{i1} & \cdots & b_{il} \\ \vdots & \ddots & \vdots \\ b_{ki} & \cdots & b_{km} \end{bmatrix} \begin{bmatrix} x^0 \\ x^1 \\ \vdots \\ x^m \end{bmatrix} = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_k(x) \end{bmatrix}. \tag{3d}
\]

The following section provides more details of the method using a case study.

III. THE CASE STUDY: A CHIPLESS RFID

Radio Frequency Identification (RFID) is a technology that utilizes radio frequency signals to identify objects. In this technology every object is marked by a tag; every tag has a unique response to a particular request signal. By sending the signal for a tag and analyzing its response, objects can be identified. The unit which generates the spectrum, receives the responses and analyzes them is called "Reader". Block diagram of a typical RFID system is shown in Fig. 4.
Depending to the type of the tags, RFID systems can be categorized in three groups. The traditional and the first one is called Active-RFID [9]. In this type of the systems there is a power supply and a chip in every tag. This chip is responsible to generate a digital code as the ID of the tag. This digital code is then modulated and sent via an antenna. The second type of the RFID systems is passive-RFID systems [10]. These systems are like active ones; but there is no power supply in the tags and the chip of each tag is supplied by the power of rectified RF signal received by the input antenna.

The third type of RFID systems are called chipless-RFID [11]–[14]; in which there is no chip or power supply in the tags and the unique ID of each tag is encoded in the magnitude or phase [15] of the reflected wave from the tag or even both of them. Based on this mechanism, different responses to a unique input spectrum are produced for every individual chipless-tag. Chipless tags are cheap and simple which can even be printed on different objects by conducting inks [16]–[18]. Therefore, this technology can be widely used for low cost and mass production applications.

One of the features of every RFID system is that there is a mixed-mode circuit in its reader to decode the backscattered wave. This is the reason that encourages us to describe an RFID system using a high level specification language such as VHDL-AMS. To do so we need more understanding of the frequency specification of the system.

A. The Chipless RFID system to be modeled

The chipless system of [11] is our case to be studied. In this system a chirp of frequencies is generated by the "Reader" and is sent by an antenna. The signals of this chirp have constant magnitude and phase. After propagation in space this spectrum is received by the tag and passes through a transmission line where a multi-resonator is placed nearby. This multi-resonator determines the ID of each tag. Each resonator generates a minimum in the magnitude and a ripple in phase in its corresponding resonance frequency. Therefore, to encode six data bits, six frequencies are required. Because each frequency can carry two states, placing or termination of the resonator of each frequency determines its corresponding logic value. Fig. 5 shows the layout and simulated "S21" of two chipless tags and their corresponding encoded spectrum and IDs [19].

The received spectrum is changed according to the tag ID then backscattered to the reader base, where its modified spectrum will be decoded. The ID can be derived from both magnitude and phase. The RX (Receive) and TX (Transmit) antennas of both tag and reader are cross polarized to prevent interference. So the signal that is sent by the TX antenna of one side can only be received by the Rx antenna of the other side.

Nevertheless, all kind of nonlinearities can be modeled in AMS, in this study for the sake of generality, antennas and propagation medium are assumed ideal so they make no change or attenuation in the identification spectrum. Fig. 5. (a), (b), (c) respectively show the layout, magnitude and phase of "S21" for a 6-bit chipless tag with "111111" ID and Fig. 5. (d), (e), (f) show the similar results for a similar tag with "010101" ID. The simulated S-parameters of these two tags are used to automatically generate their VHDL-AMS models by HF/Digital converter to be used in combination with the digital design process.

Fig. 5. (a) Layout of a tag with ID=111111 (b) Simulated magnitude of S21 for the first tag (c) Simulated phase of S21 for the first tag (d) Layout of a tag with ID=010101 (e) Simulated magnitude of S21 for the second tag (f) Simulated phase of S21 for the second tag.
For the other structure, Curve fitting results and obtained functions of these two tags are presented below. First for tag. 1 with "010101" ID. Equation (4a), shows the fitted function of magnitude spectrum and equations (4b), (4c) show coefficients matrix of tag.1 magnitude spectrum given at the bottom of the page.

\[
F_1(x) = \sum_{x=1}^{5} \frac{\sum_{j=1}^{5} a_{xj} x^{j-1}}{ \sum_{j=1}^{5} b_{xj} x^{j-1}}.
\]  

\[
B_y = \begin{bmatrix}
0.01698 & -0.1339 & 0.006565 & 1.032 & 1 \\
1.772 & -2.662 & 1 & 0 & 0 
\end{bmatrix}.
\]  

Equations (5a), (5b), (5c) show the fitted function and coefficients matrix of tag.1 with "010101" ID phase spectrum respectively. Equation (5b) is given at the bottom of the page.

\[
F_2(x) = \sum_{x=1}^{6} \frac{\sum_{j=1}^{5} a_{xj} x^{j-1}}{ \sum_{j=1}^{5} b_{xj} x^{j-1}}.
\]  

\[
A_y = \begin{bmatrix}
0.011052 & -0.08776 & 0.004164 & 0.676 & 0.6552 \\
0.6044 & -0.9084 & 0.342 & -0.0004836 & 0.0001576 
\end{bmatrix}.
\]  

\[
B_y = \begin{bmatrix}
0.0168 & -0.1333 & 0.007268 & 1.031 & 1 \\
1.938 & -1.14 & 1.568 & 1 & 0 
\end{bmatrix}.
\]  

For tag.2 with "111111" ID. Equation (6a), (6b), (6c) show the fitted function and coefficients matrix of tag.2 magnitude spectrum respectively. Equations (6b), (6c) are given at the bottom of the page.

\[
F_3(x) = \sum_{x=1}^{5} \frac{\sum_{j=1}^{5} a_{xj} x^{j-1}}{ \sum_{j=1}^{5} b_{xj} x^{j-1}}.
\]  

\[
A_y = \begin{bmatrix}
0.2404 & -2.138 & 0.537 & 18.07 & 19.96 & 3.54 \\
158.2 & -162.5 & -87.57 & 137.7 & -35.57 & 0.00414 \\
0.2991 & 1.051 & 1.174 & 0.4235 & 0 \\
3573 & 2.317e+004 & 3.206e004 & -1.867e+004 & 0 \\
-0.005686 & 0.007592 & 0.004529 & 0.002537 & -0.01211 \\
-0.004342 & -0.419 & 1.536 & -1.824 & 0.7085 \\
-0.6 & -0.6204 & -0.002245 & 0.08179 & -0.01059 \\
1.185 & 2.177 & 1 & 0 & 0 \\
3763 & 2.427e+004 & 3.342e+004 & -1.966e+004 & 0 \\
0.3635 & -0.3011 & -0.7537 & -0.06529 & 1 \\
1.744 & -2.641 & 1 & 0 & 0 \\
0.01752 & -0.1363 & 0.003262 & 1.033 & 1
\end{bmatrix}.
\]  

Equation (7a), (7b), (7c) show the fitted function and coefficients matrix of tag.2 with "111111" ID phase spectrum respectively. Equations (7b), (7c) are given at the bottom of the page.

\[
F_4(x) = \sum_{x=1}^{4} \frac{\sum_{j=1}^{5} a_{xj} x^{j-1}}{ \sum_{j=1}^{5} b_{xj} x^{j-1}}.
\]  

\[
B_y = \begin{bmatrix}
-2.898e+006 & 3.953e+006 & -5.701e+005 & -8.54e+005 & 1.789e+005 & 2.353e+004 \\
-0.1267 & 0.8403 & -0.2978 & -4.14 & 3.87 & 8.854 \\
-4.353 & -29.61 & -44.28 & 22.51 & 3.436 & -0.05575 \\
0.6126 & -0.5807 & -1.427 & 0.7421 & 1 \\
-2.894e+004 & 3.175e+004 & 1731 & -693 & 1 \\
0.01744 & -0.1359 & 0.002692 & 1.032 & 1 \\
0.08114 & 0.5693 & 1 & 0 & 0
\end{bmatrix}.
\]
B. Simulation results

To complete the procedure of modeling, "Reader" should be modeled as well. This model can also be used as a test bench for the automatically generated models. The "Reader" is considered to have a structure as shown in Fig. 6.

Fig. 6. A behavioral block diagram of the "Reader" unit of a Chipless-RFID system.

In this structure the "spectrum generator" block generates the spectrum that is needed to detect tags. In this generated spectrum, magnitude and phase of all signals are considered to be "1.0" and "0.0" respectively. The "Reader decoder" block extracts the digital code that is hidden in the backscattered spectrum and converts it to a quantized value. Each quantized value represents a logic state in a specific frequency. Finally, every quantized value is translated to a digital value in one of the "Output Code" pins; these pins are latched and keep their values until the whole spectrum is decoded. Therefore, there is a transient time for the output code in which the code is not valid. Simulation results of the generated model are shown in Fig. 7.

Please note that here, "time" is used to represent the independent variable (frequency) in the traces to summarize the code but the equivalent frequency of each time point that is used in the model is presented in Fig. 7. (e) and the value of every dot in the output traces should be considered with its corresponding frequency, not with time. Fig. 7. (a), (b) respectively show the magnitude and phase of the output signal of RX antenna in the "Reader" block for the tag of Fig. 5. (a) with "111111" ID and Fig. 7. (c), (d) show the same parameters for the other tag. These traces are resulted from the simulation of the VHDL-AMS model and are in good agreement with the simulated results of ADS as depicted in Fig. 5. Small differences between the simulated results of the generated model and ADS results are due to the different interpolation methods that each tool uses to approximate the values of S-parameters for frequencies in which the S-parameters are not exactly defined. These differences decrease as the simulated points that are used to generate the model increase. The IDs of the studied tags that are correctly decoded by the "reader" model are shown in Fig. 7. (f), (g).

The simulated traces and decoded IDs, verify the validity of the system model and confirms that the proposed modeling procedure is successful for this type of systems. The following code mentioned in appendix is the shortened model of the under-study case, generated by the proposed tool.

IV. Conclusion and Future Works

The proposed method to generate, VHDL-AMS model of mixed-mode systems with HF networks was used to model a Chipless RFID system and the simulation results confirm its validity. This work shows that the proposed CAD tool, can effectively model a matched two-port HF network in VHDLAMS. Using the proposed design method, digital and HF can be co-designed in the form of a mixed-mode system with minimum required knowledge about each field. The proposed method can be easily extended to the design of mixed-mode systems with more sections such as MEMS, power electrical and mechanical systems because of the flexibility of VHDL-AMS. Further, this is the first attempt to model a chipless RFID system. In modeling of this system, the antennas and propagation medium are assumed ideal and the HF networks which are modeled by the proposed tool are assumed two port networks with matched I/O ports; consideration of the multi-port HF networks that are not matched in their ports and also non-ideal modeling of the propagation medium and antennas are subject of our future works.

APPENDIX

This appendix presents a shortened VHDL-AMS model of the under-study case, generated by the proposed tool. Also there are models of some other required blocks like Reader source, Reader decoder and Digitizer to complete the simulation.
entity two_port is
   port ( quantity inspecmag , inspecphase , freq : in real ;
         quantity outspecmag , outspecphase : out real ) ;
end entity two_port ;
architecture curve_fit of two_port is
   quantity s21m , s21p : real ;
   constant p1 : real := 1.638 ;
   constant p2 : real := 1.69 ;
   constant p3 : real := 0.01041 ;
   constant p4 : real := -0.2194 ;
   constant p5 : real := 0.02763 ;
   begin
      freq0 := 1500000000.000 ;
      step:=4000000.000;
      count := (freq - freq0) / step ;
      IF (count >=1.0 AND count < 2.0) USE
         s21m ==000.997;     s21p==152.265;
         s11m ==000.007;     s11p==046.211;
         s12m ==000.997;     s12p==152.265;
         s22m ==000.007;     s22p==078.346;
      ELSIF (count >= 2.0 AND count <  3.0) USE
         s21m ==000.997;     s21p==151.699;
         s11m ==000.007;     s11p==044.767;
         s12m ==000.997;     s12p==151.699;
         s22m ==000.007;     s22p==078.611;
      ELSIF (count >= 3.0 AND count <  4.0) USE
         s21m ==000.007;     s21p==151.000;
         s11m ==000.007;     s11p==043.000;
         s12m ==000.007;     s12p==151.000;
         s22m ==000.007;     s22p==078.611;
      ELSIF (count >= 4.0 AND count <  5.0) USE
         s21m ==000.007;     s21p==149.500;
         s11m ==000.007;     s11p==041.500;
         s12m ==000.007;     s12p==149.500;
         s22m ==000.007;     s22p==078.611;
      ELSIF (count >= 5.0 AND count <  6.0) USE
         s21m ==000.007;     s21p==148.000;
         s11m ==000.007;     s11p==040.000;
         s12m ==000.007;     s12p==148.000;
         s22m ==000.007;     s22p==078.611;
      ELSIF (count >= 6.0 AND count <  7.0) USE
         s21m ==000.007;     s21p==146.500;
         s11m ==000.007;     s11p==038.500;
         s12m ==000.007;     s12p==146.500;
         s22m ==000.007;     s22p==078.611;
      ELSIF (count >= 7.0 AND count <  8.0) USE
         s21m ==000.007;     s21p==145.000;
         s11m ==000.007;     s11p==037.000;
         s12m ==000.007;     s12p==145.000;
         s22m ==000.007;     s22p==078.611;
      ELSIF (count >= 8.0 AND count <  9.0) USE
         s21m ==000.007;     s21p==143.500;
         s11m ==000.007;     s11p==035.500;
         s12m ==000.007;     s12p==143.500;
         s22m ==000.007;     s22p==078.611;
      ELSIF (count >= 9.0 AND count < 10.0) USE
         s21m ==000.007;     s21p==142.000;
         s11m ==000.007;     s11p==034.000;
         s12m ==000.007;     s12p==142.000;
         s22m ==000.007;     s22p==078.611;
      END USE;
   end architecture curve_fit ;
architecture behav of reader_source is
   begin
      outmag == 1.0 ;
      outphase == 0.0 ;
      freq := 1500000000.000 + 160000000000.000 * now ;
end architecture behav ;
architecture behav of reader_decoder is
   begin
      IF       (freq > 1620000000.0 and freq < 1670000000.0 and
               inmag < 0.8) USE
         code==32.0;
      ELSIF  (freq > 1710000000.0 and freq < 1760000000.0  and
               inmag < 0.8) USE
         code == 16.0  ;
      ELSIF  (freq > 1830000000.0 and freq < 1840000000.0 and
               inmag < 0.8) USE
         code == 8.0  ;
      ELSIF  (freq > 1940000000.0 and freq < 1950000000.0 and
               inmag < 0.8) USE
         code == 4.0  ;
      ELSIF  (freq > 2060000000.0 and freq < 2080000000.0 and
               inmag < 0.8) USE
         code == 2.0  ;
      ELSIF  (freq > 2180000000.0 and freq < 2220000000.0 and
               inmag < 0.8) USE
         code == 1.0  ;
      ELSE
         code == 0.0  ;
      END USE ;
   end architecture behav ;
architecture behav of dig is
   signal code_sig : bit_vector(5 downto 0) := "000000";
   constant period : time := 10 us ;
   begin
      p0 : process
         variable ana_var : real;
         begin
            ana_var := anain    ;
            if ana_var = 32.0 then
               code_sig(5) <= '1';
            elsif ana_var= 16.0 then
               code_sig(4) <= '1';
            elsif ana_var= 8.0 then
               code_sig(3) <= '1';
            elsif ana_var= 4.0 then
               code_sig(2) <= '1';
            elsif ana_var= 2.0 then
               code_sig(1) <= '1';
            elsif ana_var= 1.0 then
               code_sig(0) <= '1';
            else
               code_sig <= code_sig ;
            end if;
            wait for period;
            end process p0;
            code <= code_sig ;
   end architecture behav ;
INTERNATIONAL JOURNAL OF DESIGN, ANALYSIS AND TOOLS FOR INTERGRATED CIRCUITS AND SYSTEMS, VOL. 4, NO. 1, DECEMBER 2013

REFERENCES


Arash Jamali received the B.S. degree in electrical engineering from Razi University, Kermanshah, Iran, in 2010, and is currently working towards the M.S. degree in the Department of Electrical Engineering, Razi University. His research interests include mixed mode design, high frequency design, Energy Harvesting, and VLSI design.

Arash Ahmadi received the BSc and the MSc degrees in Electronics Engineering from Sharif University of Technology and Tarbiat Modares University, Tehran, Iran, in 1993 and 1997 respectively. He joined soon after to the Electrical Engineering Department of Razi University, Kermanshah, Iran as a faculty member. He received his PhD degree in Electronics from University of Southampton, United Kingdom in 2008. From 2008 to 2010, he was a Fellow Researcher with University of Southampth. He is currently an Assistant Professor in the Electrical Engineering Department, Razi University. His current research interests include hardware implementation of signal processing systems, high-level synthesis and bio-inspired computing.

Omid Sadeghi Fathabadi received his BSc and MSc in electrical engineering from Razi University, Kermanshah, Iran, in 2009 and 2011. He is currently a PhD student at the University of Tasmania, Hobart, Australia. His current research interests are automated modeling of high frequency circuits, biomedical signal processing and modeling, automatic control, and system identification.