iVAMS: Intelligent Metamodel-Integrated Verilog-AMS for Circuit-Accurate System-Level Mixed-Signal Design Exploration

Geng Zheng[*], Saraju P. Mohanty[†], Elias Kougianos[‡], and Oghenekarho Okobiah[§]
NanoSystem Design Laboratory, University of North Texas, Denton, TX 76203, USA.
Email: gengzheng@my.unt.edu[*], saraju.mohanty@unt.edu[†], eliask@unt.edu[‡], and oo0032@unt.edu[§]

***Abstract*—The gap between abstraction levels in analog design is a major obstacle for advancing analog and mixed-signal design automation. Intelligent surrogate models for low-level analog building blocks are needed to bridge behavioral and transistor-level simulations. With this objective, artificial neural network (ANN) metamodels are incorporated in Verilog-AMS to capture the highly nonlinear response of the analog block. Parameterized ANN Verilog-AMS behavioral metamodels are constructed for efficient system-level design exploration. The application of these intelligent metamodels to multi-objective analog block optimization is demonstrated. To the best of the authors' knowledge *this is the first paper to integrate artificial neural network models in Verilog-AMS*. To demonstrate the application of iVAMS, a biologically-inspired "firefly optimization algorithm" is applied to an OP-AMP design. The optimization process is sped up by 5580× due to the use of iVAMS with negligible loss in accuracy.**

I. Introduction

Design automation tools for digital circuits have easily followed technology trends as they have high regularity. Thus their performance can be well controlled at different abstraction levels from top to bottom. In contrast, a hierarchical way to tightly control and predict the performance of analog and mixed-signal (AMS) blocks at every design phase has not yet been established in industrial practice [1].

In a hierarchical approach, design information should propagate seamlessly between abstraction levels. A key enabler of this approach is efficient surrogate models for low-level AMS building blocks. When a design variable, such as the size of a transistor, in the building block changes, this model could capture the resultant changes of the block characteristics immediately and pass them to higher levels. This model should also allow a system-level algorithm to fine tune its local design variable value to optimize the overall system performance.

We propose **the following requirements** for such AMS metamodels: (*a*) The model should be capable of modeling the building block performance metrics for fast design optimization; (*b*) The model should be able to be used in high-level AMS behavioral simulations; (*c*) The model should be parameterized so that it can capture the entire response surface of the building block with reasonable accuracy in a large design space; (*d*) The construction of such models should only cost a small portion of an analog designer's time and the CPU time required for this process should be moderate.

The last requirement reflects the fact that the designer's time is more valuable than CPU time [2]. While our aim is to minimize CPU time, minimizing the burden imposed on the designer has higher priority. With the aforementioned considerations, we propose an intelligent metamodel integrated Verilog-AMS (iVAMS) module for analog blocks. iVAMS aims at closing the gap between abstraction levels in analog design which is currently regarded as the "number one" requirement for advancing AMS design automation [1]. The **overall contribution of this paper, iVAMS**, enables system-level or behavioral modeling with circuit-level intelligent artificial neural network metamodels such that the gap between system-level speed and circuit-level accuracy is bridged.

Metamodeling nano-CMOS AMS circuits is becoming popular [3], [12]. In [4], an OP-AMP design space was partitioned and the sub-regions were approximated with local low-order polynomials to capture nonlinearity in design space. ANN metamodels of OP-AMP parameterized macromodels (meta-macromodel) of our paper avoid such design space partitioning. Using ANN for behavioral modeling has been explored in the literature [5]. From the prior research, the closest one to the current paper is [6]. However, it implemented an ANN with built-in training algorithm but not a behavioral model for any common analog circuit.

The rest of this paper is organized as follows: Section II presents the concept of iVAMS. Section III uses a 90 nm OP-AMP as a case study to demonstrate the generation and use of iVAMS for an analog block.

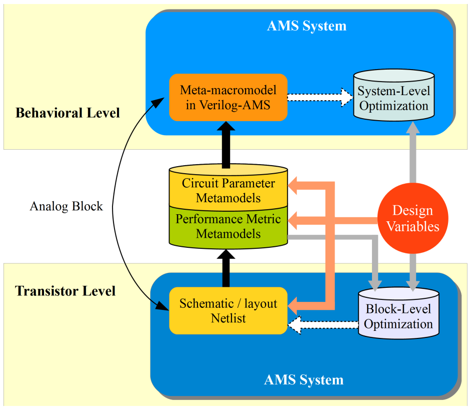
AMS System
Behavioral Level
Meta-macromodel in Verilog-AMS
System-Level Optimization
Analog Block
Circuit Parameter Metamodels
Performance Metric Metamodels
Design Variables
Transistor Level
Schematic / layout Netlist
Block-Level Optimization
AMS System

Fig. 1. The concept of the proposed iVAMS.

II. Proposed Intelligent Verilog-AMS (iVAMS)

Simulating an entire AMS system with transistor-level netlists, though accurate, is formidable. A common solution is to simulate the AMS system at the behavioral level. Parameterized behavioral models share the same design variables as the transistor-level design. When the values of the design variables change, the values of the circuit parameters for the behavioral model change accordingly. Thus the impact of the transistor-level changes on the behavioral level are captured. iVAMS provides such Verilog-AMS modules, as shown in Fig. 1.

The iVAMS generation flow is shown in Fig. 2. iVAMS is based on a set of ANN metamodels [11]. The metamodels are sampling-based and fall into two categories: the performance metric metamodel (PMM) and the circuit parameter metamodel (CPM). Given an analog block, PMMs estimate its performance for any point in the design space. No actual circuit simulation is required in this estimation process. Thus, PMMs provide an ultra-fast way to explore the design space of the analog block. CPMs, on the other hand, estimate the circuit parameters required to construct a macromodel for the analog block. By integrating the CPMs into a macromodel and describing them in Verilog-AMS, a parameterized behavioral model, called **meta-macromodel**, can be obtained. A macromodel is usually a white-box or grey-box model [7] that retains certain amount of physical information of the analog block [8], [4], [9]. Constructing a macromodel for an analog block requires physical insight and consumes a great portion of the designer's time. Constructing a metamodel requires sampling the design space, which consumes CPU time. Meta-macromodeling selects a suitable macromodel and estimates the required parameters using metamodels. Given an analog block, the goal of iVAMS generation is to obtain the neural network metamodels (PMMs and CPMs) and the meta-macromodel integrated Verilog-AMS module.

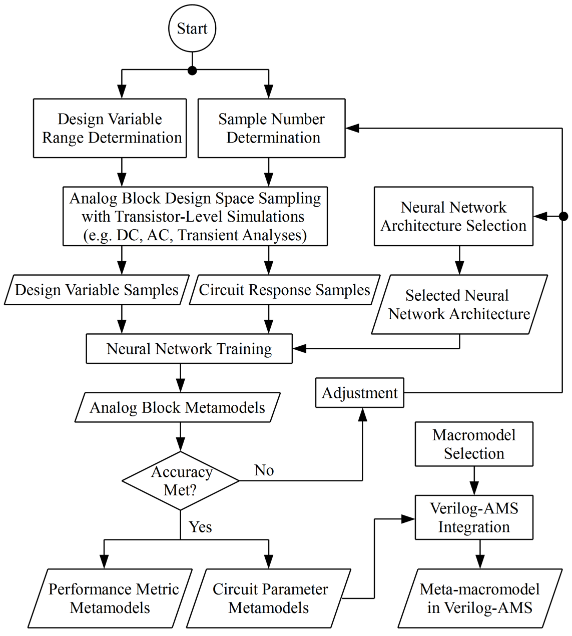
Start
Design Variable Range Determination
Sample Number Determination
Analog Block Design Space Sampling with Transistor-Level Simulations (e.g. DC, AC, Transient Analyses)
Neural Network Architecture Selection
Design Variable Samples
Circuit Response Samples
Selected Neural Network Architecture
Neural Network Training
Adjustment
Analog Block Metamodels
Accuracy Met?
No
Yes
Macromodel Selection
Verilog-AMS Integration
Performance Metric Metamodels
Circuit Parameter Metamodels
Meta-macromodel in Verilog-AMS

Fig. 2. iVAMS generation flow.

III. iVAMS Case Study Circuit: A 90nm OP-AMP

We apply iVAMS to the 90nm fully differential OP-AMP shown in Fig. 3 with a 1 V supply. It consists of an operational transconductance amplifier (OTA), a common-source amplifier, and a common-mode feedback (CMFB) circuit.

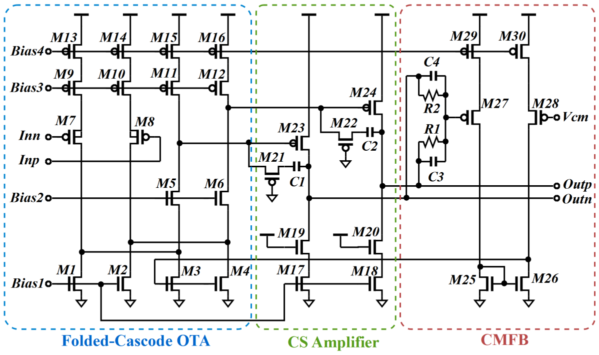

Fig. 3. OP-AMP schematic.

A. OP-AMP iVAMS Metamodel Generation

The performance metrics of interest for this OP-AMP are the open-loop DC gain (A_0), bandwidth (BW), phase margin (PM), slew rate (SR), and power dissipation (PD) whose metamodels are generated. In order to construct a meta-macromodel for system-level design exploration, metamodels for the circuit parameters used in the macromodel are also required. These circuit parameters include the transconductance g_m, and the positive and negative maximum available currents I_p and I_n of the op-amp input stage. With sufficient neurons in the hidden layer, an ANN can approximate any function [10].

The input layer, $\mathbf{x}$, is a vector of the OP-AMP design variables which include the bias current and the transistor widths and lengths. There are thirty transistors in the OP-AMP design. By properly grouping the transistors, we have sixteen design variables ($N = 16$). In this paper, the ANN hidden layer consists of four neurons ($M = 4$) with hyperbolic tangent function as the activation function f_1. The output layer is a single neuron employing a linear activation function f_2. The model output $\hat{y}$ is one of the performance metrics or circuit parameters. W_1 is a matrix composed of the weights of the connections from the design variables in the input layer to the neurons in the hidden layer. Similarly, W_2 is formed by the weights of the connections from the hidden layer to the output layer. Additional control on each neuron is through bias b_{ij} ($i = 1, 2$ and $j = 1, 2, ..., M$). **The ANNs were trained using 500 samples with Bayesian Regulation training**.

In order to evaluate the accuracy of the iVAMS metamodels, **a verification set consisting of 2000 samples** of design variable and circuit response pairs was generated. This sampling is for verification purposes only and need not be used when iVAMS is used in a design flow. The output $\hat{y}$ computed by the metamodel is compared with the "true" output y obtained from transistor-level simulations. To comparatively assess the proposed iVAMS ANN metamodels, we generated second-order polynomial (PO) metamodels and compared their accuracy. The results are listed in Table I.

TABLE I
ACCURACY OF THE OP-AMP iVAMS METAMODELS.

Metamodel		Accuracy Metric			
Output	**Type**	R^2	**RMAE**	**RRSE**	**RMSE**
A_0	NN	0.959	1.324	0.202	41.93 V/V
	PO	0.973	1.044	0.163	33.78 V/V
BW	NN	0.987	0.894	0.116	2.12 kHz
	PO	0.986	0.965	0.117	2.14 kHz
PM	NN	0.901	2.161	0.317	4.99^o
	PO	0.348	4.466	0.807	12.70^o
SR	NN	0.989	0.483	0.105	0.292 mV/ns
	PO	0.985	0.662	0.119	0.332 mV/ns
P_D	NN	0.996	0.523	0.062	8.306 μW
	PO	0.980	1.314	0.141	18.817 μW
g_m	NN	0.999	0.106	0.018	1.769 μA/V
	PO	0.999	0.101	0.021	1.973 μA/V
I_p	NN	0.991	0.675	0.095	0.311 μA
	PO	0.729	3.407	0.521	2.506 μA
I_n	NN	0.994	0.494	0.080	0.261 μA
	PO	0.749	3.727	0.501	2.412 μA

The ANN metamodels achieve higher accuracy overall except for A_0. Another advantage of the ANN metamodels over the PO metamodels is that their accuracy can be further improved by adding more neurons to the hidden layer. All ANN metamodels in this work employ a 4-neuron hidden layer for simplicity. In practice, adaptively adjusting the hidden layer size is recommended to find the optimal model.

B. iVAMS Meta-Macromodel Construction

An OP-AMP meta-macromodel can facilitate fast system-level design exploration. Meta-macromodel construction starts with macromodel selection. Some OP-AMP macromodels that can be used include the structural model in [8], the linear time-invariant (LTI) model in [4], and the symbolic model in [9]. We adopted a symbolic model similar to the one in [9] since it not only models the op-amp small-signal behavior but also the large-signal behavior such as slew-rate limitation. This model, combined with the iVAMS CPMs to form the op-amp meta-macromodel, is shown in Fig. 4.

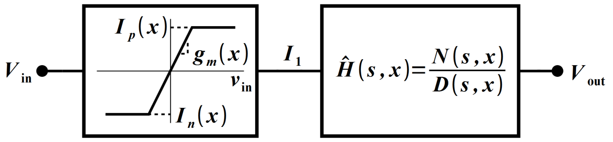

Fig. 4. The iVAMS op-amp meta-macromodel.

The two-stage model in Fig. 4 takes into account the slew-rate effect due to the limited maximum available positive and negative currents I_p and I_n. These circuit parameters, together with the transconductance of the first stage g_m and the OP-AMP small-signal function $\hat{H}(s)$, are functions of the design variables. They are estimated using the iVAMS CPMs. With f_1 being a hyperbolic tangent function and f_2 being a pure linear function, this architecture can be expressed mathematically as: $\hat{y} = b_{21} + \sum_{j=1}^{M} w_{2,j} \cdot \tanh\left(b_{1j} + \sum_{i=1}^{N} w_{1,ij} \cdot x_i\right)$, where $w_{1,ij} \in W_1$ and $w_{2,j} \in W_2$. Implementing this equation in Verilog-AMS is an essential step of building the iVAMS meta-macromodel. An example is shown in Algorithm 1. After training the neural networks for the CPMs, the neural network weights and biases are stored in text files. In the `initial` block of the OP-AMP Verilog-AMS module, these weights and biases are read from the files, and the function `nn_metamodel` computes the circuit parameter values for the meta-macromodel. The computed circuit parameter values are used in an `analog` process in the Verilog-AMS module to realize the model in Fig. 4. The OP-AMP small-signal function can be described using a function such as `laplace_nd`.

Algorithm 1 iVAMS code for the OP-AMP.

```
function real nn_metamodel;
integer w1, w2, b1, b2, i, j, readfile,...; real w, b, v, u;
// Read metamodel weights and bias from
// text files w1, w2, b1, and b2.
begin
w1 = $fopen("w1.txt", "r"); w2 = $fopen("w2.txt", "r");
b1 = $fopen("b1.txt", "r"); b2 = $fopen("b2.txt", "r");
v = 0.0;
for (j = 0; j < nl; j = j + 1)
begin
    u = 0.0;
    for (i = 0; i < size_x; i = i + 1)
    begin
        readfile = $fscanf(w1, "%e", w); u = u + w * x[i];
    end
    readfile = $fscanf(w2, "%e", w);
    readfile = $fscanf(b1, "%e", b); v = v + w * tanh(u + b);
end
readfile = $fscanf(b2, "%e", b); nn_metamodel = v + b;
$fclose(w1); $fclose(w2); $fclose(b1); $fclose(b2);
end
endfunction
```

The quality of the meta-macromodel relies on the accuracy of the CPMs and the suitability of the selected macromodel. In order to validate the iVAMS meta-macromodel, simulations using the constructed Verilog-AMS module were compared with those using the SPICE model. The results of the AC analysis are shown in Fig. 5. The difference seen in the frequency responses can be reduced by adaptively adjusting the ANN hidden layer sizes instead of using a fixed value.

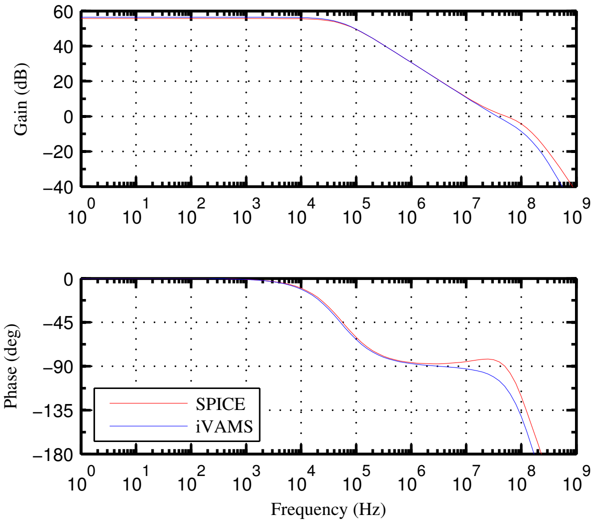

Fig. 5. AC analyses of the OP-AMP.

C. Block-Level Multiobjective Optimization Using iVAMS

We now demonstrate block-level optimization for the OP-AMP design using the iVAMS PMMs. The optimization **problem is to maximize SR and to minimize PD, constrained by the requirements for $\mathbf{A_0}$, BW, and PM**. An effective approach is to find the Pareto front (PF) that consists of a set of non-dominated solutions for the optimization problem. The designer can then select one design from this solution set to implement. In this paper, a metaheuristic **multiobjective firefly algorithm (MOFA)** is used. It mimics the behavior of tropic firefly swarms that are attracted toward flies with higher flash intensity. Preliminary studies show that its performance could surpass well established multi-objective algorithms [13].

The goal of the MOFA is to find K Pareto points that constitute the PF through a predetermined number of iterations, t_{max}. The algorithm starts with K randomly generated designs [13]. In each iteration, the performance of the K OP-AMP designs are estimated using the iVAMS PMMs. The best design determined by the combined weighted sum of the objectives, $\psi(\mathbf{x})$, is computed for each current design. $\widehat{SR}(\mathbf{x})$ and $\widehat{PD}(\mathbf{x})$ are the slew rate and power dissipation for a design $\mathbf{x}$ estimated using the iVMAS PMMs, and are normalized with respect to their own sample mean and standard variation. The current designs will be moved to new locations within the design space using the computed move vectors. The PMMs are then used to check whether the new designs satisfy the constraints. The optimization specifications and an arbitrarily selected optimal design from the PF of a MOFA optimization are shown in Table II.

TABLE II
OP-AMP Design Optimization.

Performance	Constraint	Selected Optimal Design	
		Predicted	True
A_0 (dB)	> 43	56.4	55.7
BW (kHz)	> 50	56.8	56.7
PM (degree)	> 70	81.9	88.5
	Objective		
SR (mV/ns)	Maximized	5.54	5.49
P_D (μW)	Minimized	85.11	85.77

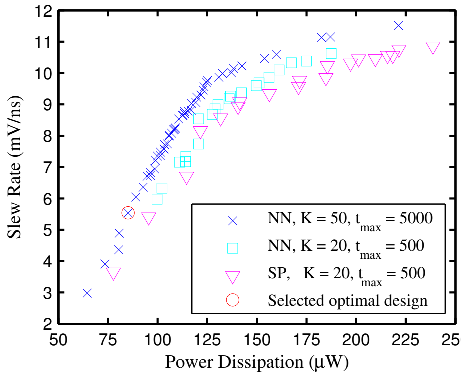

Fig. 6. Pareto fronts from MOFA OP-AMP optimization.

Three optimization runs have been performed for MOFA and presented in Fig. 6. The first run is aimed at finding the true PF. In order to approximate the true PF, K and t_{max} have to be sufficiently large. In this run, $K = 50$ and $t_{max} = 5000$ were set experimentally. Although these numbers are large, the runtime for this optimization is small due to ANN iVAMS PMMs. To compare the iVAMS-assisted MOFA (iVAMS-MOFA) with the same algorithm using SPICE models (SPICE-MOFA) to evaluate the OP-AMP performance, we may run a similar optimization. However, running MOFA using SPICE (SP) models with such large K and t_{max} would be extremely time consuming. Therefore, we alternatively decreased there numbers to $K = 20$ and $t_{max} = 500$ and performed two optimization runs to compared the speed of iVAMS-MOFA and SPICE-MOFA. It can be seen that **the iVAMS-MOFA is 5580 times faster than the SPICE-MOFA**.

IV. Conclusions

The circuit-level accurate behavioral modeling framework called iVAMS has been presented. The creation of an iVAMS module for an op-amp block has been discussed. The use of the iVAMS for block-level optimization has been demonstrated using a novel multi-objective firefly algorithm. Construction of parameterized behavioral models using iVAMS is also exemplified through an op-amp case study. Future research includes enhancing iVAMS with yield-estimation capability to address variability of nanometer circuits.

References

[1] H. Graeb, "ITRS 2011 analog EDA challenges and approaches," in *Proc. Design, Automation & Test in Europe Conf.*, 2012, pp. 1150–1155.

[2] M. J. Krasnicki, R. Phelps, J. R. Hellums, M. McClung, R. A. Rutenbar, and L. R. Carley, "ASF: a practical simulation-based methodology for the synthesis of custom analog circuits," in *Proc. IEEE/ACM Int. Conf. Computer Aided Design*, 2001, pp. 350–357.

[3] O. Garitselov, S. P. Mohanty, and E. Kougianos, "A Comparative Study of Metamodels for Fast and Accurate Simulation of Nano-CMOS Circuits," *IEEE Trans. Semicond. Manuf.*, vol.25, no.1, pp. 26–36, 2012.

[4] J. Wang, X. Li, and L. T. Pileggi, "Parameterized macromodeling for analog system-level design exploration," in *Proc. 44th ACM/IEEE Design Automation Conf.*, 2007, pp. 940–943.

[5] D. De Jonghe and G. Gielen, "Efficient analytical macromodeling of large analog circuits by transfer function trajectories," in *Proc. IEEE/ACM Int Computer-Aided Design (ICCAD) Conf*, 2011, pp. 91–94.

[6] K. Suzuki, A. Nishio, A. Kamo, T. Watanabe, and H. Asai, "An application of Verilog-A to modeling of back propagation algorithm in neural networks," in *Proc. 43rd IEEE Midwest Symp. Circuits and Systems*, vol. 3, 2000, pp. 1336–1339.

[7] R. Romijn, L. zkan, S. Weiland, J. Ludlage, and W. Marquardt, "A grey-box modeling approach for the reduction of nonlinear systems," *Journal of Process Control*, vol. 18, no. 9, pp. 906–914, 2008.

[8] G. J. Gomez, S. H. K. Embabi, E. Sanchez-Sinencio, and M. Lefebvre, "A nonlinear macromodel for CMOS OTAs," in *Proc. IEEE Int Circuits and Systems ISCAS '95. Symp*, vol. 2, 1995, pp. 920–923.

[9] H. Zhang and G. Shi, "Symbolic behavioral modeling for slew and settling analysis of operational amplifiers," in *Proc. IEEE 54th Int Midwest Symp. Circuits and Systems*, 2011, pp. 1–4.

[10] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 303–314, 1989.

[11] O. Garitselov, S. P. Mohanty, and E. Kougianos, "Fast-Accurate Non-Polynomial Metamodeling for Nano-CMOS PLL Design Optimization," in *Proc. 25th International Conf. VLSI Design*, 2012, pp. 316–321.

[12] M. B. Yelten, T. Zhu, S. Koziel, P. D. Franzon, and M. B. Steer, "Demystifying surrogate modeling for circuits and systems," *IEEE Circuits Syst Mag.*, vol. 12, no. 1, pp. 45–63, 2012.

[13] X.-S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, pp. 1–10, 2012.