

Probabilistic Model Checking of Resistive Electrical Circuits

Bogdan Vukobratović, and Staniša Dautović, *Student Member, IEEE*

Abstract — This paper presents a novel approach to the formal verification of resistive electrical circuits using the probabilistic model checking.

Keywords — formal verification, probabilistic model checking, resistive electrical circuits, sensitivity analysis.

I. INTRODUCTION

The formal verification of a hardware system is a formal mathematical procedure of proving or disproving the correctness of a designed system with respect to a certain formal specification or property. The formal mathematical proof (also called static verification) should be distinguished from the functional (dynamic) verification, validation or simulation of the behavior of the system under test.

Based on a successful application of different formal verification techniques for digital circuits (like Binary Decision Diagrams (BDD) based equivalence checking, Linear Time Logic (LTL) and Computation Tree Logic (CTL) based model checking etc.) [1], formal verification is applied to analog circuits as well. In contrast to digital logic (basically, computing over finite fields \mathbb{GF} and discrete time \mathbb{N}), analog circuits are continuous in signal values (computing over field \mathbb{R}) and time (again \mathbb{R}), resulting in much harder verification tasks. One has to deal with infinite state space and dense metric time.

The keyword/key phrase “sensitivity analysis” of electrical circuits/networks [2] is coined before the wider introduction of the formal verification methods into the analog world. It may be the case that the contents of two research fields have large overlapping and same goals. When one talks about the “tolerance analysis”, “sensitivity analysis”, “noise analysis” or “worst-case simulation” in this context, it usually refers to changes in input-output behavior under (small, allowed/guaranteed by industrial manufacturer) changes/variations in the nominal values of circuit parameters, like resistors R , capacitors C or inductors L . In reality, there is no such thing as a “perfect” component, and even the most precise electrical

components available in the market have their own tolerances. For example, if we consider a simple 100-ohm resistor, it will always have a tolerance value of, for example, five percent (5%). Thus, the minimum and maximum values of this component in mass production are 95 and 105 ohms. It is also common to talk about “robustness of perturbed systems”, addressing behavior under extreme min/max values (95 and 105 ohms).

Specified quantities of interest can be also stability of output voltage and/or current, guaranteed attenuation and/or amplification, stability of poles and zeros of the system, maximum deviation from specified frequency characteristics (amplitude and phase characteristics), limited power dissipation, tracing of desired trajectory in state-space, stability of operating points etc. All those properties are also of interest in formal verification of analog circuits.

The probabilistic interpretation of voltages and currents in the circuit theory [3] is not a mainstream of engineering practice. Modeling of electrical circuits based on Markov chains is well known in mathematics, where it represents an interesting illustration of the theory of Markov chains. This situation exists probably due to the lack of further practical applications. Applying Probabilistic Symbolic Model checker (PRISM) [4] in the verification of an actual input-output behavior of the circuit (with respect to the specified properties), or in estimation of tolerance and robustness of the circuit behavior (with respect to the changes or perturbances in circuit parameters), could represent such a well motivated practical application.

There are numerous proposed approaches to formal verification of analog circuits: a) influential German school of Lars Hedrich and Erich Barke has several results (verification of linear and nonlinear circuits, both in time and frequency domain, based on sampling of the state space, or interval arithmetic, or model checking in CTL-AT logic (logic CTL extended with real time) etc.); b) work of C.-J. Richard Shi (based on symbolic determinants and Determinant Decision Diagrams, DDDs); c) work of Chris J. Myers et al. (based on Timed Hybrid Petri Nets); d) work of Oded Maler (based on Signal Temporal Logic (STL) and variants of different Real-time Temporal Logics, such as MITL[a,b]); e) work of Seshadri and Abraham (based on global optimization techniques); f) work of Ghosh and Vemuri (using higher-order logic proof checker PVS), to name just a few. We omit here the extensive list of references due to the limitation on paper’s length.

The proposed approach is a novel one, in the context of the formal verification of analog electrical circuits. We are

Bogdan Vukobratović is a M.Sc. student at Department for Power, Electronics and Communications Engineering, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia. (E-mail: bogdan.vukobratovic@gmail.com).

Staniša Dautović is with the Department for Power, Electronics and Communications Engineering, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia. (E-mail: dautovic@uns.ns.ac.yu).

aware of only one paper (not involving PRISM), whose authors are tracing similar path [5], but they use so called Markov Network, and not the probabilistic interpretation of voltages and currents in Discrete-time Markov chain (DTMC) and Continuous-time Markov chain (CTMC) models of electrical circuits.

II. PROBABILISTIC INTERPRETATION OF VOLTAGES AND CURRENTS AND MODELING OF RESISTIVE ELECTRICAL CIRCUITS AS DTMCs

The probabilistic modeling of electrical circuits is introduced in [3]. We present here a short introduction following terminology and an original example taken from the reference book [3].

Random walk in one dimension will be explained with an example: a man walks along a 5-block stretch of Madison Avenue. He starts at corner x and, with probability p , walks one block to the right and, with probability $q = 1-p$, walks one block to the left. When he comes to the next corner he again randomly chooses his direction along Madison Avenue. He continues until he reaches corner 5, which is home, or corner 0, which is a bar. If he reaches either home or the bar, he stays there.

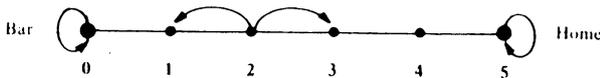


Fig. 1. Random walk example; taken from [3].

When modeling a resistive electrical circuit, which consists of resistive net and voltage sources applied on certain nodes, two dimensional walk is considered, where a walker might have more than two choices at any node. In this model, node 0 (bar) models ground, node 5 (home) voltage source and probabilities p and q model resistor values joined to the edges of the underlying graph.

When modeling a general affine resistive electrical circuit, probability that the walker will choose to go to node Y when he is in node X , is equal to the electrical conductance G (reciprocal value of electrical resistance R) between nodes Y and X , divided by the sum of the conductances from node X to all other nodes.

Here we give the probabilistic interpretation of voltages and currents in resistive circuit/networks as defined in [3].

Claim 1: (Probabilistic interpretation of voltages) Let h_X be the hitting probability, that random walker, starting at node X , will reach node A before B . If a unit voltage V_A , is applied between nodes A and B of the circuit, then the voltage V_x at any node X is equal to h_X .

If an arbitrary voltage V_A between nodes A and B is assumed instead of the unit voltage, then the hitting probability h_X would be replaced by an expected value in a game where the player starts at X and is rewarded V_A if A is reached before B and 0 otherwise. \square

Claim 2: (Probabilistic interpretation of currents) When

a unit current flows into node A and out of node B , the current i_{XY} flowing through the branch connecting X to Y is equal to the expected net number of times that a walker, starting at A and walking until he reaches B , will move along the branch from X to Y . These currents are proportional to the currents that arise when a unit voltage is applied between A and B , the constant of proportionality being the effective resistance of the network. \square

The problem of calculating voltages and currents can be solved by using Monte Carlo method, method of relaxation, solving linear equations or by calculating steady state probabilities of underlying DTMC. The latest approach can be implemented in the symbolic model checker PRISM. The Markov chain is presented by its transition matrix \mathbf{P} . If p_{XY} is probability that the walker will go from X to Y , then the transition matrix for the random walk example (Fig. 1.) is given on the Fig. 2.

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ p_{10} & 0 & p_{12} & 0 & 0 & 0 \\ 0 & p_{21} & 0 & p_{23} & 0 & 0 \\ 0 & 0 & p_{32} & 0 & p_{34} & 0 \\ 0 & 0 & 0 & p_{43} & 0 & p_{45} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 2. Transition matrix \mathbf{P} of DTMC for the random walk example (Fig. 1.)

For more information on these topics, please see [3].

III. PROBABILISTIC SYMBOLIC MODEL CHECKER PRISM

PRISM is a probabilistic symbolic model checker, a tool for formal modeling and analysis of systems, which exhibit random or probabilistic behavior. It supports three types of probabilistic models: DTMCs, CTMCs and Markov decision processes (MDPs), plus extensions of these models with costs and rewards.

The fundamental components of the PRISM language are *modules* and *variables*. A module contains a number of local *variables*. The values of these variables at any given time constitute the state of the module. The behavior of each module is described by a set of *commands*,

```
[ ] guard->prob_1:update_1+...+prob_n:update_n (1)
```

The *guard* is a predicate over all the variables in the model (including those belonging to other modules). Each *update* describes a transition which the module can make if the *guard* is true. A transition is specified by giving the new values of the variables in the module, possibly as a function of other variables. Each *update* is also assigned a probability (or in some cases a rate) which will be assigned to the corresponding transition.

The random walk example transition matrix (given on Fig.2.) described in the PRISM language is shown in Fig.3.

```

dtmc
module M1
  x : [0..5] init 2;

  [ ] x=0 -> (x'=0);
  [ ] x=1 -> P10:(x'=0) + P12:(x'=2);
  [ ] x=2 -> P21:(x'=1) + P23:(x'=3);
  [ ] x=3 -> P32:(x'=2) + P34:(x'=4);
  [ ] x=4 -> P43:(x'=3) + P45:(x'=5);
  [ ] x=5 -> (x'=5);
endmodule

```

Fig.3. PRISM model of the random walk example (Fig. 1.)

If the voltage source other than unit voltage is to be modeled, `costs` and `rewards` feature is used like in Fig. 4.

```

rewards x=5: Va; endrewards

```

Fig. 4. Usage of costs and rewards in PRISM

In order to analyze a probabilistic model which has been specified and constructed in PRISM, it is necessary to identify one or more *properties* of the model which can be evaluated by the tool. Properties are expressed in a language based on the logics PCTL (for DTMCs and MDPs) and Continuous Stochastic Logic (CSL) for CTMCs, probabilistic extensions of the CTL.

The full explanation of property expression available in PRISM, exceeds the scope of this article, so please refer to [4] and PRISM web-site for more information.

Although resistive electrical circuits are modeled as DTMCs, CTMC PRISM engine will be used, because the current release of PRISM does not support the S (“steady state”) operator with DTMC models. The reference [6] describes the relationship in existence and form of the stationary distribution of the irreducible CTMC X with those of its embedded DTMC \bar{X} . This enables us to use CTMC model of a resistive circuit in PRISM, providing us with the equivalent results for steady state probabilities, which represent potentials of nodes in our model. For more details, please refer to [6].

IV. EXAMPLES AND RESULTS

A. Proof of concept: a simple example

In this example, two resistors $R_1=1.5k\Omega$ and $R_2=1k\Omega$ are connected in series to the 5V constant voltage source (voltage divider). Furthermore, let the tolerance for these resistors be $\pm 1\%$. The model in PRISM is given on Fig. 5.

```

ctmc

const R1;
const R2;

module M1
  x : [1..3] init 2;

  [ ] x=1 -> (x'=1);
  [ ] x=2 -> 1/R1:(x'=1) + 1/R2:(x'=3);
  [ ] x=3 -> (x'=3);
endmodule

rewards x=1 : 5; endrewards

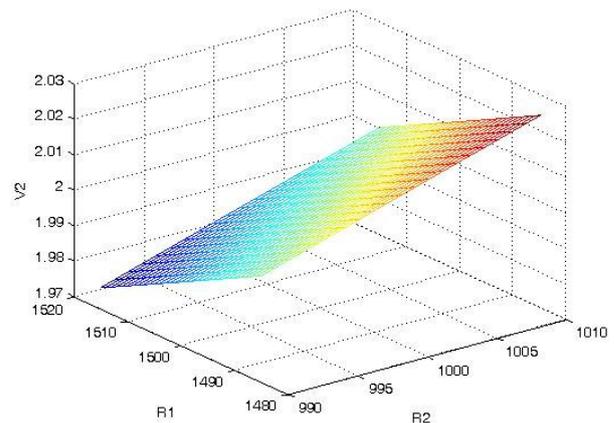
```

Fig. 5. CTMC PRISM model of the voltage divider

The `const` statement defines constants to be used in the model, and they can be initialized later, during experiments in PRISM. The initial state for the model represents the starting point for the random walker. For an example, if we want to calculate the potential V_2 of node 2, the PRISM construct `x: [1..3] init 2` will set the initial state to 2. Calculating the V_2 is done by calculating the following property:

$$R=? [S] \quad (3)$$

The meaning of the property (3) is: “What is the expected reward (i.e., potential in our case) in the steady state?”. Within the experiment, we calculate in PRISM steady state probabilities for all combinations of resistor values $R_1 \in [1485, 1515]$ and $R_2 \in [990, 1010]$, with step of 1Ω . As it can be seen on Fig. 6., values for potential V_2 belong to the interval $V_2 \in [1.976, 2.024]$, i.e. it varies 1.2% from the nominal value $V_2=2V$. The plot on the Fig. 6. is monotone, but should not be misinterpreted as linearly dependant on resistor values. The plotted result focuses on a very narrow interval of 2-dimensional function domain.

Fig. 6. Potential V_2 as a function of resistor values R_1, R_2

B. Example 2: Robustness of perturbed system

Here we describe a more advanced example of the modeling and model checking of resistive circuits in PRISM. In Example 2, as a difference to the simple model from Example 1, we have changes in the referent values 5V of voltage sources V_{IN1} and V_{IN2} , as well as variable circuit topology, modeled with switches S_1 and S_2 . We will analyze circuit behavior under extreme min/max values of seven perturbed circuit parameters $R_1, R_2, R_3, R_{S1}, R_{S2}, V_{IN1}$ and V_{IN2} .

On the resistive circuit shown on Fig. 7. there are 3 resistive loads: $R_1=6k\Omega$, $R_2=4k\Omega$ and $R_3=5k\Omega$, all with 10% tolerances. Resistors $R_4=R_5=R_6=R_7=R_8=R_9=10\Omega$ are modeling non-ideal wire resistances. Voltage sources V_{IN1} and V_{IN2} give 4.85V to 5.15V and have $R_{IN1}=R_{IN2}=100\Omega$ internal resistance. Loads R_2 and R_3 can be disconnected from the net by two switches having resistance of 1Ω when on, and $1M\Omega$ when off. The PRISM model is given on Fig. 8. The results are shown on Fig. 9., where the potential values V_2 of node 2 are plotted, for all possible $2^7=128$ combinations of the seven changing parameters. The

changes in these parameters are modeled through variables $kR_1, kR_2, kR_3, kV_{IN1}, kV_{IN2}, kR_{S1}$ and kR_{S2} , all taking values 0 or 1, in order to provide minimum and maximum parameter values.

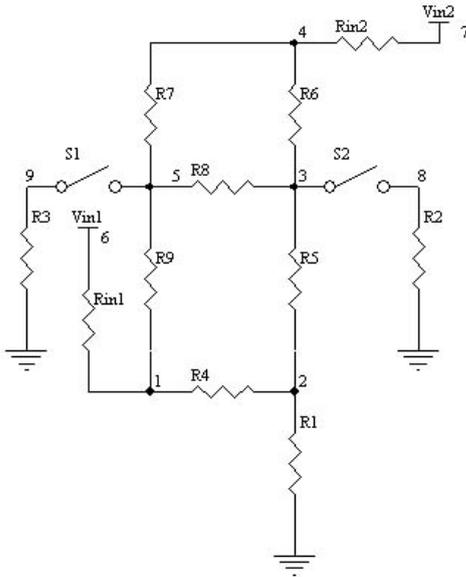


Fig. 7. Resistive circuit from Example 2.

```
ctmc

const double GIN1=1/100;
const double GIN2=1/100;
const double G4=1/10;
const double G5=1/10;
const double G6=1/10;
const double G7=1/10;
const double G8=1/10;
const double G9=1/10;

formula G1=1/(5400+ kR1*1200);
formula G2=1/(3600 + kR2*800);
formula G3=1/(4500 + kR3*1000);
formula GS1 = 1/(1 + kRS1*1000000);
formula GS2 = 1/(1 + kRS2*1000000);

module M1
  x : [0..9];
  kR1: [0..1]; kR2: [0..1];
  kR3: [0..1]; kVIN1: [0..1];
  kVIN2: [0..1]; kRS1: [0..1];
  kRS2: [0..1];

  [] x=0 -> x'=0;
  [] x=1 -> G4:(x'=2) + G9:(x'=5) + GIN1:(x'=6);
  [] x=2 -> G1:(x'=0) + G4:(x'=1) + G5:(x'=3);
  [] x=3 -> GS2:(x'=8) + G5:(x'=2) + G6:(x'=4) +
G8:(x'=5);
  [] x=4 -> G6:(x'=3) + G7:(x'=5) + GIN2:(x'=7);
  [] x=5 -> GS1:(x'=9) + G9:(x'=1) + G8:(x'=3) +
G7:(x'=4);
  [] x=6 -> x'= 6;
  [] x=7 -> x'= 7;
  [] x=8 -> G2:(x'=0) + GS2:(x'= 3);
  [] x=9 -> G3:(x'=0) + GS1:(x'= 5);

endmodule

rewards
  x=6: (4.85 + kVIN1*0.3);
  x=7: (4.85 + kVIN2*0.3);
endrewards

init true endinit
```

Fig. 8. PRISM model for circuit on Fig. 7.

The initial state for the model represents the starting point for the random walker. We used `init true endinit` construct, which sets all the states as initial, but we will

specify the true initial state for x later with properties. If variables are not initialized, as it is the case with $kR_1, kR_2, kR_3, kV_{IN1}, kV_{IN2}, kR_{S1}$ and kR_{S2} , PRISM will cycle through all their possible states. Except for the x -variable, no transitions are specified for other 7 variables, so their values will remain the same as initial during the random walk.

For an example, properties of interest to be checked can be

$$R=? [S \{x=2\}\{min\}] \tag{4}$$

Equation (4) means: “In a steady state, what is the minimal expected reward the walker will have, starting from node 2, every time with different $kR_1, kR_2, kR_3, kV_{IN1}, kV_{IN2}, kR_{S1}$ and kR_{S2} values?”. Using Claim 1, this value equals the minimal potential of the node 2 in the circuit from Fig. 7., under all 128 parameter combinations.

Another property to be verified can be

$$(x=2) \Rightarrow R>4.67 [S] \ \& \ R<5.12 [S] \tag{5}$$

Equation (5) means: “In a steady state, is it true that the expected reward the walker will have, starting from node 2, will be between 4.67V and 5.12V, for all possible $kR_1, kR_2, kR_3, kV_{IN1}, kV_{IN2}, kR_{S1}$ and kR_{S2} combinations?”. If satisfied, property (5) guarantees that the potential of node 2 will stay within the specified boundaries [4.67, 5.12], under all 128 experimental setups. Fig. 9. justifies the correctness of the results obtained from the verification of the statement (5).

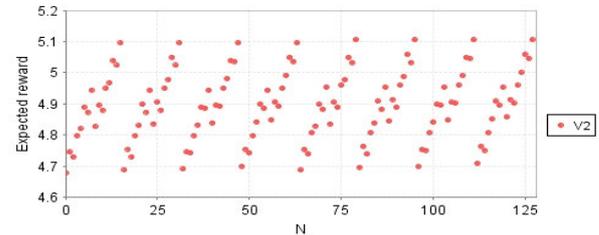


Fig. 9. Values of V2 for all possible parameter combinations

REFERENCES

[1] T. Kropf, *Introduction to Formal Hardware Verification*, Springer; 1999.
 [2] L. Kolev, “Worst-case tolerance analysis of linear DC and AC electric circuits”, *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, Volume 49, Issue 12, Dec 2002 Page(s): 1693 – 1701.
 [3] P.G. Doyle and J.L. Snell, *Random Walks and Electrical Networks*, Mathematical Assn of America, 1984. Available: <http://math.dartmouth.edu/~doyle/docs/walks/walks.pdf>
 [4] M. Kwiatkowska, G. Norman and D. Parker, “PRISM: Probabilistic Symbolic Model Checker”, In *Proc. PAPM/PROBMIV’01 Tools Session*, pages 7-12. Available as Technical Report 760/2001, University of Dortmund. September 2001. Available: <http://www.prismmodelchecker.org>
 [5] C. Borgelt and R. Kruse, “Probabilistic Graphical Models for the Diagnosis of Analog Electrical Circuits”, in L. Godo (Ed.): *ECSQARU 2005, LNAI 3571*, pp. 100–110, 2005. Springer-Verlag Berlin Heidelberg 2005. Available: <http://www.springerlink.com/content/83wymce7erquwkdg/fulltext.pdf>
 [6] A. Pacheco, *Stochastic Manufacturing and Service Systems*, Class Notes, Spring 2002. Available: <http://www.math.ist.utl.pt/~mjmorais/CN-2002.pdf>