

# Modelling Multistability and Hysteresis in ESD Clamps, Memristors and other Devices

Tianshi Wang

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA  
email: tianshi@berkeley.edu

**Abstract**—Multistability and hysteresis are widely occurring phenomena in devices, leading to many misconceptions among compact model developers. In this paper, we show how hysteretic devices can be modelled in general using only continuous/smooth primitives in a differential equation format suitable for simulation, and how the models can be implemented properly in languages like Verilog-A and ModSpec (MATLAB®). Apart from the generic formalism, several concrete device examples are described and analyzed, including a new compact model for ESD protection devices, new memristor models, a simplified electro-thermal model for HBTs, and a modified Stoner-Wohlfarth model for ferromagnets. Common features of these models are studied to further illustrate the modelling methodology for multistability and hysteresis in devices.

## I. Introduction

Many devices feature multiple stable equilibrium points. For example, in ESD protection devices, a phenomenon known as “snapback” in the  $I$ - $V$  characteristic curves allows the devices to draw different amounts of current at the same input voltage, depending on the occurrence of impact ionization [1]. Similarly, recently developed memristive devices can be in either high- or low-resistance state when powered off, depending on the voltages that have been applied to them before [2, 3]. It has also been observed that HBT models can have multiple possible voltage biases generating the same current when considering thermal effects [4]. Another example is from magnetics — bi-stable magnetization under a small external magnetic field is a crucial property in ferromagnetic materials and devices [5]. In such devices with multistability, sweeping their inputs up and down often generates hysteresis, *i.e.*, a looping behaviour in the input-output ( $I$ - $O$ ) graph.

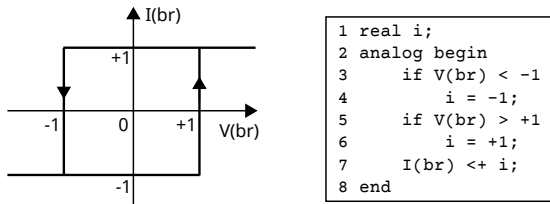


Fig. 1: Example of problematic Verilog-A code for modelling  $I$ - $V$  hysteresis.

These device properties are often difficult to model properly, or understand intuitively. In fact, compact model developers are often befuddled by them, resulting in a plethora of problematic models. Figure 1 shows a very simple example; similar code with *if-else* statements and memory states<sup>1</sup> for modelling hysteresis can be found as part of several published compact models for ESD clamps [8] and memristors [9, 10]. Therefore, in this paper, we would like to first clarify several common

confusions about the multistability and hysteresis observed in devices.

Firstly, although multistability normally implies that there will be a sudden jump in a device’s response when sweeping its input, it does not mean there has to be discontinuity in the model equations. It doesn’t justify the use of *if-else* statements either. In fact, continuous and smooth model equations can also create abrupt changes in device responses; designing such smooth equations is the key in modelling multistability and hysteresis properly.

Moreover, although hysteresis implies time dependence between inputs and outputs, it does not mean the device has to know the absolute time. Neither does it need to access the input at which it was evaluated at the last time point. In fact, a properly written compact model should not be specific to time-dependent simulation algorithms; it should run in other analyses, such as DC, small signal AC, Harmonic Balance, *etc.*, as well. Several existing compact models [9, 11] incorporate  $I$ - $V$  hysteresis by accessing `$abstime` and implementing time integration inside; their use is limited to only transient simulation. There are also Verilog-A models that use memory states for storing and accessing the input value in the previous device evaluation, limiting their robustness in PSS simulations [6].

Another misconception is that a model needs to be an analog behaviour model [12] to have hysteresis. This leads to the use of many simulator directives, *e.g.*, `@initial_step`, `analysis()`, `@cross()`, `$bound_step()`, *etc.*, whereas these constructs are in fact unnecessary for modelling hysteresis, and should be avoided [7, 13].

Multistability does not mean ill-posedness [14] — it does not mean there has to be a region in the state space with zero derivatives to keep the device output from moving. While some models [11, 15] use such “flat” regions for modelling multistability, this approach results in singular circuit Jacobian matrices and undefined model behaviours in these regions.

In fact, the basic requirements on a device model with multistability and hysteresis are no different from those on general compact models — the model should still be formulated in the Differential Algebraic Equation (DAE) format [16]; it should use continuous/smooth functions and should be well-posed [14, 17, 18]. In this paper, we explain the correct generic way of modelling and analyzing devices with multistability and hysteresis. In Sec. II, we start by considering a simple two-terminal device. We show that by including an internal state variable and designing its dynamics properly,  $I$ - $V$  hysteresis can be included in the model. Specifically, we show how

<sup>1</sup>A memory state, or hidden state [6], in Verilog-A is a variable used without assigning a value. They should be avoided in compact models [7].

a key property of the model — a single continuous/smooth curve in the state space that contains all the steady state solutions, is connected to the well-posedness of the model, and how a negative-sloped fold in the curve results in the abrupt transitions observed in device responses. In the meanwhile, we also demonstrate the usefulness of the homotopy analysis [19], which was originally developed mainly to aid DC convergence, in characterizing and analyzing hysteretic devices. Then we write this example model both in the ModSpec format [20], which is the DAE format for specifying devices in the Berkeley MAPP [21, 22], and in the Verilog-A language. For the Verilog-A implementation, we use the most consistently supported features of the language, such that the model will run in all main-stream simulators, including Spectre<sup>®</sup>, HSPICE and Xyce.

Then from Sec. III to Sec. VI, we apply the insights gained from studying the generic hysteretic model in Sec. II to more concrete device examples. In Sec. III, we study the snapback phenomenon in ESD clamps. By introducing an internal state representing the occurrence of impact ionization and designing a fold in its DC solution curve, we develop a compact model for ESD clamps that not only captures the snapback and multistability phenomena in the devices, but also includes the time dependence of ionization. It also works robustly in circuit simulations.

In Sec. IV, we consider memristor models as another example. Existing models for memristive devices all suffer from issues related to mathematical ill-posedness. In particular, as noted in [18], they don't generate correct DC responses. We show that the reason for the DC failure is indeed the lack of a single DC solution curve in steady state. Guided by the model template, we propose well-posed models for memristors. They preserve the accuracy and physics in existing models, while fixing their model problems. The result is a collection of models for various types of memristors, all working in all the common circuit analyses in major simulators.

The insight of modelling hysteresis via constructing a continuous DC curve with a fold is useful not just for designing new models, but also for analyzing the problems with existing ones. It is known that HBT models with thermal effects can have uniqueness problems [4]. In Sec. V, we use the same simulation techniques to reproduce a multistable voltage problem in a simplified electro-thermal HBT model, and show that the multistability originates from the same mechanism. Furthermore, this insight also guides us in fixing the uniqueness problem when multistability is undesired in the model.

For ferromagnetic devices, in Sec. VI, we study the Stoner-Wohlfarth model [23] that captures the response of a magnet's internal magnetization to an external magnetic field. It is known that the SW model suffers from problems related to unbounded unknowns; we show that it corresponds to having infinite number of DC solution curves in the state space. Furthermore, once we redefine the variable of simulation and fix the unboundedness problem, a single DC curve shows up, bringing this magnetic model into our modelling framework as well.

We would like to note that the models developed and studied in this paper all consist of relatively simple equations; the purpose is to illustrate the modelling methodology for devices with

multistability and hysteresis. More complex versions of them, with more physical effects taken into consideration, are part of the future work.

## II. How to Model Hysteresis Properly

In this section, we first study how to model  $I$ - $V$  hysteresis in two-terminal devices properly. The equation of a general two-terminal device without memory can be written as

$$I(t) = f(V(t)), \quad (1)$$

where  $V(t)$  is the voltage across the device,  $I(t)$  the current through it. For example,  $f(V(t)) = \frac{V(t)}{R}$  describes a simple linear resistor.

For devices with  $I$ - $V$  hysteresis,  $I(t)$  and  $V(t)$  cannot have a simple algebraic mapping like (1). Instead, we introduce a state variable  $s(t)$  into (1) and rewrite the  $I$ - $V$  relationship as

$$I(t) = f_1(V(t), s(t)). \quad (2)$$

The dynamics of the internal state variable  $s(t)$  is governed by a differential equation:

$$\frac{d}{dt}s(t) = f_2(V(t), s(t)). \quad (3)$$

In this formulation, we cannot directly calculate the current based on the voltage applied to the device at a single time  $t$ ;  $I(t)$  also depends on the value of  $s(t)$ . On the other hand, at time  $t$ , the value of  $s(t)$  is determined by the history of  $V(t)$  according to (3). Therefore, we can think of the device as having internal “memory” of the history of its input. If we choose the formula for  $f_1$  and  $f_2$  in (2) and (3) properly, as we sweep the input voltage, hysteresis in the current becomes possible.

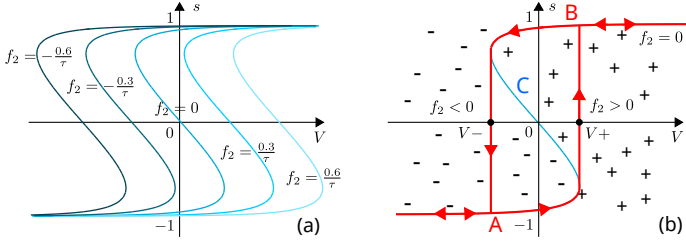
In the rest of this paper, (2) and (3) serve as a model template for devices with  $I$ - $V$  hysteresis. To illustrate its use, we design a device example, namely “hys\_example”, with functions  $f_1$  and  $f_2$  defined as follows.

$$f_1(V(t), s(t)) = \frac{V(t)}{R} \cdot 0.5 \cdot (s(t) + 1). \quad (4)$$

$$f_2(V(t), s(t)) = \frac{1}{\tau} (\tanh(K \cdot (V(t) + 2 \cdot s(t))) - s(t)). \quad (5)$$

The choice of  $f_1$  is easy to understand. If we assume  $s(t)$  is within  $(-1, 1)$ , incorporating  $0.5 \cdot (s(t) + 1)$  as a factor modulates the conductance of the device between 0 and  $1/R$ . The choice of  $f_2$  determines the dynamics of  $s(t)$ . And when  $f_2 = 0$ , the corresponding  $(V, s)$  pairs will show up as part of the DC solutions of circuits containing this device. Therefore, if we visualize the values of  $f_2$  in a contour plot, such as in Figure 2 (a), the curve representing  $f_2 = 0$  is especially important. In (5), through the use of the tanh function plus a linear term in  $s$ , we design the  $f_2 = 0$  curve to fold back in the middle, crossing the  $V = 0$  axis three times. In this way, when  $V$  is around 0, there are three possible values  $s$  can settle on, all satisfying  $\frac{d}{dt}s(t) = f_2 = 0$ . This multiple stability in state variable  $s$  is the foundation of hysteresis found in the DC sweeps on the device.

Figure 2 (b) illustrates how hysteresis takes place in DC sweeps. In Figure 2 (b), we divide the  $f_2 = 0$  curve into three parts: curve A and B have positive slopes while C has a negative one. When we sweep  $V$  towards the right at a very slow speed to approximate DC conditions, starting from a negative value left of  $V = 0$ , at the beginning, there is only one possible DC solution



**Fig. 2:** Contour plot of  $f_2$  function in (5) and predicted  $s$ - $V$  hysteresis curve based on the sign of  $f_2$ .

of  $s$ . As we increase  $V$ , the  $(V, s)$  pair will move along curve A, until A ends when  $V$  reaches  $V_+$ . If  $V$  increases slightly beyond  $V_+$ , multiple stability in  $s$  disappears.  $(V, s)$  reaches the  $f_2 > 0$  region and  $s$  will grow until it reaches the B part of the  $f_2 = 0$  curve. This shows up in the DC solutions as a sudden jump of  $s$  towards curve B. Similarly, when we sweep  $V$  in the other direction starting from the right of  $V_+$ , the  $(V, s)$  pair will follow curve B, then have a sudden shift to A at  $V_-$ . Because  $V_+ > V_-$ , hysteresis occurs in  $s$  when sweeping  $V$ , as illustrated in Figure 2 (b). Since  $s$  modulates the device's conductance, there will also be hysteresis in the  $I$ - $V$  relationship.

Note that the analysis of the origin of hysteresis does not involve absolute time. It is the fold in the DC solution curve defined by  $f_2(v, s) = 0$  that generates multiple stable equilibria in the device, which then result in abrupt changes and hysteresis when sweeping the device's input. As mentioned earlier,  $s$  can be thought of as encoding the memory of  $V$  from the past. Its multiple equilibria reflect the different possible sets of history of  $V$ . And the separation between  $V_+$  and  $V_-$  in the DC curves ensures that no matter at what speed we sweep  $V$ , there will always be hysteresis in the  $s$ - $V$  relationship.

Model equations for `hys_example` defined in (4) and (5) can be written as a compact model into the Berkeley MAPP using the ModSpec format. The ModSpec format specifies a device model in the following formulation [20, 21].

$$\dot{\vec{z}} = \frac{d}{dt} \vec{q}_e(\vec{x}, \vec{y}) + \vec{f}_e(\vec{x}, \vec{y}, \vec{u}), \quad (6)$$

$$0 = \frac{d}{dt} \vec{q}_i(\vec{x}, \vec{y}) + \vec{f}_i(\vec{x}, \vec{y}, \vec{u}). \quad (7)$$

Vectors  $\vec{x}$  and  $\vec{z}$  contain the device's I/Os – inputs/outputs related to the circuit's connectivity:  $\vec{z}$  comprises those I/Os that can be expressed explicitly (for `hys_example`, it contains only  $I$ ), while  $\vec{x}$  comprises those that cannot (for `hys_example`, it is  $V$ ).  $\vec{y}$  contains the model's internal unknowns (for `hys_example`, it is  $s$ ), while  $\vec{u}$  provides a mechanism for specifying time-varying inputs within the device (e.g., as in independent voltage or current sources). The functions  $\vec{q}_e$ ,  $\vec{f}_e$ ,  $\vec{q}_i$  and  $\vec{f}_i$  define the differential and algebraic parts of the model's explicit and implicit equations.

For `hys_example`, we can write its model equations in the ModSpec format as follows.

$$\vec{f}_e(\vec{x}, \vec{y}, \vec{u}) = \frac{\vec{x}}{R} \cdot (\tanh(\vec{y}) + 1), \quad \vec{q}_e(\vec{x}, \vec{y}) = 0, \quad (8)$$

$$\vec{f}_i(\vec{x}, \vec{y}, \vec{u}) = \tanh(K \cdot (\vec{x} + 2 \cdot \vec{y})) - \vec{y}, \quad \vec{q}_i(\vec{x}, \vec{y}) = -\tau \cdot \vec{y},$$

with  $\vec{x} = [V]$ ,  $\vec{y} = [s]$ ,  $\vec{z} = [I]$ ,  $\vec{u} = []$ .

```
1 function MOD = hys_ModSpec()
2   MOD = ee_model();
3   MOD = add_to_ee_model(MOD, 'name', 'hys');
```

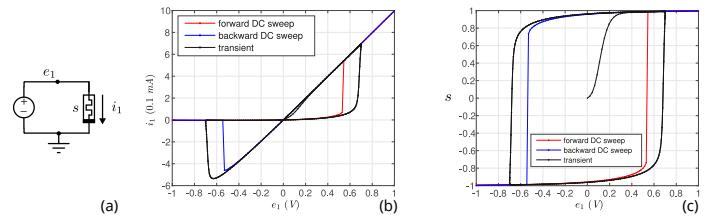
```
4   MOD = add_to_ee_model(MOD, 'terminals', {'p', 'n'});
5   MOD = add_to_ee_model(MOD, 'explicit_outs', {'ipn'});
6   MOD = add_to_ee_model(MOD, 'internal_unks', {'s'});
7   MOD = add_to_ee_model(MOD, 'implicit_eqn_names', {'ds'});
8   MOD = add_to_ee_model(MOD, 'parms', {'R', 1e3, ...
9     'K', 1, 'tau', 1e-3});
10  MOD = add_to_ee_model(MOD, 'fgei', {'@fe', '@qe', '@fi', '@qi'});
11  MOD = finish_ee_model(MOD);
12 end % hys_ModSpec

13
14 function out = fe(S)
15   v2struct(S); % populates workspace with vpn/R/K/tau
16   out = vpn/R * 0.5 * (1+s); % ipn
17 end % fe
18
19 function out = qe(S)
20   out = 0; % ipn
21 end % qe
22
23 function out = fi(S)
24   v2struct(S);
25   out = tanh(K*(vpn + 2*s)) - s;
26 end % fi
27
28 function out = qi(S)
29   v2struct(S);
30   out = -tau * s;
31 end % qi
```

**Listing 1:** `hys_example_ModSpec.m`

We can enter the model information in (8) into MAPP by constructing a ModSpec object MOD. The code in Listing 1 shows how to create this device model for `hys_example` entirely in the MATLAB® language. For more detailed description of the ModSpec format, users can issue the command “`help ModSpec_concepts`” in MAPP.

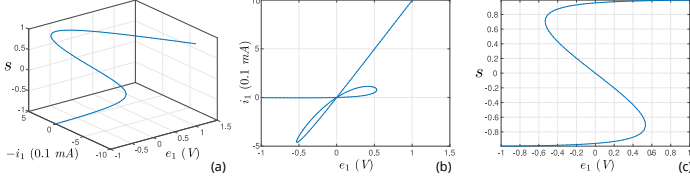
We can then simulate the model specified with Listing 1 using MAPP in various circuit analyses. Figure 3 shows the results from DC sweep and transient simulation with input voltage sweeping up and down. It confirms that hysteresis takes place in both  $I$ - $V$  and  $s$ - $V$  relationships of the device.



**Fig. 3:** Results from DC sweep and transient simulation in MAPP, showing hysteresis in both  $s$  and  $i_1$  when sweeping the input voltage, in either type of the analyses.

When we sweep  $V$  back and forth, curve C, the one with a negative slope in Figure 2 (b) never shows up in solutions. The reason is that, although it also consists of solutions of  $f_2 = 0$ , these solutions are not stable. With a little perturbation, whether from physical noise or numerical error, a  $(V, s)$  point on curve C will move to either A or B. These unstable solutions can be captured using the homotopy analysis [19]. Homotopy analysis can track the DC solution curve in the state space. Results from homotopy analysis are shown in Figure 4. We note that all the circuit's DC solutions indeed form a smooth curve in the state space. The side view of the 3-D plot displays curve C we have designed in our model equation (5). The corresponding curve in the top view connects the two discontinuous DC sweep curves in Figure 3; it consists of all the unstable solutions in the  $I$ - $V$  relationship. These results from homotopy analysis provide us with important insights into the model. They reveal that there

is a single smooth and continuous DC solution curve in the state space, which is an indicator of the well-posedness of the model. They also illustrate that it is the fold in the smooth DC solution curve that has created the discontinuities in DC sweep results. These insights are important for the proper modelling of hysteresis.



**Fig. 4:** Results from homotopy analysis in MAPP: (a) 3-D view of all the DC solutions; (b) top view of the DC solutions shows the folding in the  $I$ - $V$  characteristic curve, explaining the  $I$ - $V$  hysteresis from DC and transient voltage sweeps in Figure 3; (c) side view of the DC solutions.

Moreover, the top view explains the use of internal state  $s$  for modelling hysteresis from another angle. Without the internal state, it would be difficult if not impossible to write a single equation describing the  $I$ - $V$  relationship shown in Figure 4 (b). With the help of  $s$ , we can easily choose two simple model equations as (4) and (5), and the complex  $I$ - $V$  relationship forms naturally.

```

1  `include "disciplines.vams"
2  module hys(p, n);
3      inout p, n;
4      electrical p, n, ns;
5      parameter real R = 1e3 from (0:inf);
6      parameter real K = 1 from (0:inf);
7      parameter real tau = 1e-3 from (0:inf);
8      real s;
9
10     analog begin
11         s = V(ns, n);
12         I(p, n) <= V(p, n)/R * 0.5 * (1+s);
13         I(ns, n) <= tanh(K*(V(p, n) + 2*s)) - s;
14         I(ns, n) <= ddt(-tau*s);
15     end
16 endmodule

```

**Listing 2:** hys\_example.va

`hys_example` can also be implemented in the Verilog-A language. Apart from the differences in syntax, Verilog-A differs from ModSpec in one key aspect — the way of handling internal unknowns and implicit equations. Verilog-A models a device with an internal circuit topology, *i.e.*, with internal nodes and branches defined just like in a subcircuit. The variables in a Verilog-A model, the “sources” and “probes”, are potentials and flows specified based on this topology. Coming from this subcircuit perspective, the language doesn’t provide a straightforward way of dealing with general internal unknowns and implicit equations inside the model, *e.g.*, the state variable  $s$  and the equation (3) in `hys_example`.

As a result, an internal unknown is often declared as a general variable using the `real` statement. `idt()`, `$abstime` and hard-coded time integration methods are often used for describing implicit differential equations. These approaches should be avoided in modelling [7, 18]. Instead, in this paper, we show how to properly model both the state variable  $s$  by considering it as a voltage, and the implicit equation by treating it as the KCL at an internal node. As in Listing 2, we declare an internal branch, whose voltage represents  $s$ . One end of the branch is an internal node that doesn’t connect to any other branches. In this

way, by contributing  $\tanh(K \cdot (V + 2 \cdot s)) - s$  and  $\text{ddt}(-\tau \cdot s)$  both to this same branch, the KCL at the internal node will enforce the implicit differential equation in (5).

Declaring  $s$  as a voltage is not the only way to model `hys_example` in Verilog-A. Depending on the physical nature of  $s$ , one can also use Verilog-A’s multiphysics support and model it as a potential in other disciplines. One can also switch potential and flow by defining  $s$  as a flow instead. The essence of our approach is to recognize that state variable  $s$  is a circuit unknown, and thus should be modelled as a potential or flow in Verilog-A, for the consistent support from different simulators in various circuit analyses.

### III. Modelling ESD Snapback

ESD protection devices feature a phenomenon known as snapback — the current through a device does not monotonically grow with the input voltage, but folds back within a certain voltage range. This fold in the  $I$ - $V$  graph can be observed in Transmission Line Pulse (TLP) measurements. It physically means that, when the device is put in a circuit, as its input voltage increases beyond a certain trigger point, namely  $V_{t1}$ , impact ionization begins to happen and the amount of current through the device suddenly jumps. And the high current can sustain itself when the voltage is swept back to  $V_{t1}$ ; the device will turn “off” only at a lower voltage when the current drops below a hold current  $I_H$ , corresponding to a voltage  $V_{H}$  normally smaller than  $V_{t1}$ . In between  $V_H$  and  $V_{t1}$ , the device can have different currents depending on whether it is in the “on” or “off” state.

To incorporate such devices in circuit simulation, some specialized algorithms have been developed [24, 25]. As for compact models, some physics-based ones leverage existing models for BJTs and MOS devices and design subcircuits around them for approximating the device structure and characteristics [26–28]. In comparison, behavioural models for ESD clamps [1, 29, 30] have much lower model complexity, which simplifies parameter extraction significantly and provides more intuitions into the operation of the devices. Among the available behavioural models, [1] is the first to be able to capture the time dependence in the on/off transition in ESD protection devices. The model discussed in this section is based on it.

From the discussion in Sec. II, we note the similarity between the ESD snapback behaviour and the model template we have developed for a general hysteretic device. This indicates that the multistability observed in ESD protection devices can also be modelled by introducing a state variable  $s$ , and designing a fold in the steady state curve of its dynamics. Adapted from [1], we model the  $I$ - $V$  relationship as

$$I = I_{off} + s \cdot I_{on}, \quad (9)$$

where  $I_{on}$  and  $I_{off}$  are empirical equations for on-state and off-state currents:

$$I_{on} = G_{on} \cdot (V - V_H), \quad (10)$$

$$I_{off} = I_S \cdot e^{-V/V_T} \cdot \sqrt{1 + \frac{\max(V, 0)}{V_D}}. \quad (11)$$

Here,  $s$  is a state variable between 0 and 1; it is an indicator of whether impact ionization is present. It should grow to 1

when  $V > V_{I1}$ , and fall back to 0 when  $V < V_{IH}$ ; in between, it can have multiple steady state values.

The dynamics of this state variable can be modelled in similar ways as discussed in Sec. II. In the formulation of the growth of  $s$  in (5), the steady state of  $s$  is naturally limited to  $(-1, 1)$ ; we convert it to  $(0, 1)$  by using  $s^* = 2 \cdot (s - 0.5)$  in (5) instead. Similarly, in (5), when the voltage across the device is swept up and down, the transition voltage thresholds are around  $\pm 1$ ; we bring these thresholds to  $V_{I1}$  and  $V_{IH}$  by first convert  $V$  to  $V^*$  before putting it in (5).

$$V^* = \frac{2}{V_{I1} - V_{IH}} \cdot (V - 0.5 \cdot V_{I1} - 0.5 \cdot V_{IH}). \quad (12)$$

Then the dynamic of the internal ionization indicator state is modelled as follows.

$$\tau \cdot \frac{d}{dt} s = \tanh(K \cdot (V^* + 2 \cdot s^*)) - s^*. \quad (13)$$

This is essentially the same  $f_2$  function in the model template from Sec. II, with its steady state solutions forming a similar curve as in Figure 2 with a similar negative-sloped fold in the middle. The fold explains the multiple stable currents within  $(V_{IH}, V_{I1})$ , as well as the  $I$ - $V$  hysteresis in the device. Equations (9) and (13) then constitute a behavioural model for ESD protection devices.

```

1 function MOD = ESD_snapback_ModSpec()
2   MOD = ee_model();
3   MOD = add_to_ee_model(MOD, 'name', 'ESD snapback');
4   MOD = add_to_ee_model(MOD, 'terminals', {'p', 'n'});
5   MOD = add_to_ee_model(MOD, 'explicit_outs', {'ipn'});
6   MOD = add_to_ee_model(MOD, 'internal_unks', {'s'});
7   MOD = add_to_ee_model(MOD, 'implicit_eqn_names', {'ds'});
8   MOD = add_to_ee_model(MOD, 'parms', {'Gon', 0.01, ...
9     'VH', 16, 'VT1', 48, 'VIH', 26, 'Is', 1e-12, ...
10    'VT', 0.026, 'VD', 0.7, 'K', 1, 'tau', 1e-9, ...
11    'C', 1e-13, 'maxslope', 1e15, 'smoothing', 1e-10});
12   MOD = add_to_ee_model(MOD, 'fqi', {'@fe', '@qe', '@fi', '@qi'});
13   MOD = finish_ee_model(MOD);
14 end
15
16 function out = fe(S)
17   v2struct(S);
18   Ion = smoothclip(Gon*(vpn - VH), smoothing)...
19     - smoothclip(-Gon*VH, smoothing);
20   Ioff = Is * (1 - limeexp(-vpn/VT, maxslope))...
21     * sqrt(1 + max(vpn, 0)/VD);
22   out = Ioff + s * Ion; % ipn
23 end
24
25 function out = qe(S)
26   v2struct(S);
27   out = C * vpn;
28 end
29
30 function out = fi(S)
31   v2struct(S);
32   Vstar = 2*(vpn-0.5*VT1-0.5*VIH)/(VT1-VIH);
33   sstar = 2*(s-0.5);
34   out = tanh(K*(Vstar + 2*sstar)) - sstar;
35 end
36
37 function out = qi(S)
38   v2struct(S);
39   out = -tau*s;
40 end

```

Listing 3: ESD\_snapback\_ModSpec.m

```

1 'include "disciplines.vams"
2 module ESDsnapback(p, n);
3   inout p, n;
4   electrical p, n, ns;
5
6   parameter real Gon = 0.1 from (0:inf);
7   parameter real VH = 16 from (0:inf);

```

```

8   parameter real VT1 = 48 from (0:inf);
9   parameter real VIH = 26 from (0:inf);
10  parameter real Is = 1e-12 from (0:inf);
11  parameter real VT = 0.026 from (0:inf);
12  parameter real VD = 0.7 from (0:inf);
13  parameter real K = 1 from (0:inf);
14  parameter real C = 1e-13 from [0:inf];
15  parameter real tau = 1e-9 from (0:inf);
16  parameter real maxslope = 1e15 from (0:inf);
17  parameter real smoothing = 1e-10 from (0:inf);
18  real s, Ion, Ioff, Vstar, sstar;
19
20  analog function real smoothclip;
21    input x, smoothing;
22    real x, smoothing;
23    begin
24      smoothclip = 0.5*(sqrt(x*x + smoothing) + x);
25    end
26  endfunction // smoothclip
27
28  analog begin
29    s = V(ns, n);
30    Ion = smoothclip(Gon*(V(p, n)-VH), smoothing)
31      - smoothclip(-Gon*VH, smoothing);
32    Ioff = Is * (1 - limeexp(-V(p, n)/VT))
33      * sqrt(1 + max(V(p, n), 0)/VD);
34    Vstar = 2*(V(p, n)-0.5*VT1-0.5*VIH)/(VT1-VIH);
35    sstar = 2*(s-0.5);
36    I(p, n) <+ Ioff + s * Ion;
37    I(p, n) <+ ddt(C * V(p, n));
38    I(ns, n) <+ tanh(K*(Vstar + 2*sstar)) - sstar;
39    I(ns, n) <+ ddt(-tau*s);
40  end
41 endmodule

```

Listing 4: ESD\_snapback.va

The code implementation in ModSpec and Verilog-A are shown in Listing 3 and Listing 4 respectively. Figure 5 shows simulation results from MAPP. Results from DC and transient voltage sweeps are overlaid, demonstrating the hysteresis in the  $I$ - $V$  graph; homotopy results are plotted in Figure 5 (b), confirming the fold we have designed in the model's steady state curve. Transient results in Figure 5 (a) also show that ionization does not happen instantaneously; same as [1], our model captures the time dependence of impact ionization. Moreover, on top of [1], our model also captures the  $I$ - $V$  hysteresis in DC sweeps. And it does so without sacrificing the model's smoothness or its robustness in simulation. The model works well in various circuits. As an illustration, we simulate an ESD clamp with the HBM configuration in Figure 6. Transient results confirm that it implements a clamp at  $V_{IH} \approx 30V$ .

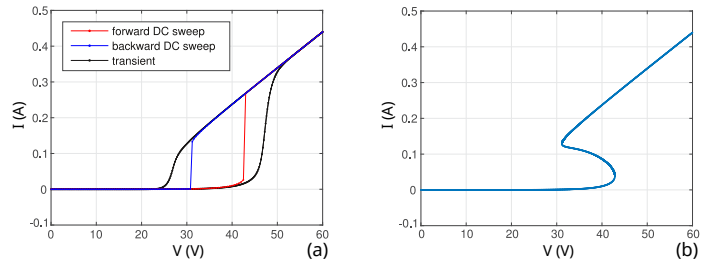


Fig. 5: Forward/backward DC, transient voltage sweep responses, and homotopy analysis results from the ESD clamp model in Listing 3.

Note that there is certain arbitrariness in the choice of the equation for the dynamics of the internal state; we are simply reusing the equation in (5) to illustrate the idea of modelling ESD snapback with a fold in the model's steady state solution curve. The transition points in (5) are not exactly  $\pm 1$ , and the exact value of  $d/dt s$  is mainly controlled by the order of the time constant parameter  $\tau$ . This choice is partly due to the fact that



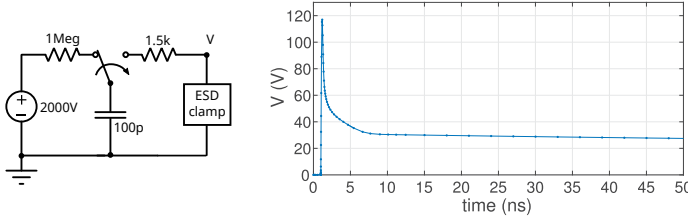


Fig. 6: HBM test bench for an ESD clamp and transient simulation results for the voltage across the clamp.

there are currently no well-established formulae for the growth rate of impact ionization. Making the ionization dynamics more physical is part of the future research in the development of simulation-ready ESD clamp models.

#### IV. RRAM and Memristor Models

An RRAM is a memristive device consisting of two metal electrodes and a thin oxide film separating them. Depending on whether a conductive filament in the film connects the electrodes, the device can be in either low- or high-resistance state. Therefore, the internal state variable for RRAM models can be defined as the  $gap$  between the tip of the filament and the opposing electrode. By filling in the model template in Sec. II and designing the  $f_1$  equation for current calculation and  $f_2$  for  $gap$  dynamics, we will have compact models for RRAM devices.

Among the existing models for RRAMs and other memristive devices, the formula for  $f_1$  are mostly consistent [9, 10, 15, 31]. In this section, we choose to use the  $f_1$  function in [9, 10]:

$$f_1(V, gap) = I_0 \cdot \exp\left(-\frac{gap}{g_0}\right) \cdot \sinh\left(\frac{V}{V_0}\right), \quad (14)$$

where  $I_0$ ,  $g_0$ ,  $V_0$  are fitting parameters.

For  $f_2$ , we can adapt the  $gap$  growth formulation in [9, 10] and write it as

$$f_2(V, gap) = -v_0 \cdot \exp\left(-\frac{E_a}{V_T}\right) \cdot \sinh\left(\frac{V \cdot \gamma \cdot a_0}{t_{ox} \cdot V_T}\right), \quad (15)$$

where  $v_0$ ,  $E_a$ ,  $a_0$  are fitting parameters,  $t_{ox}$  is the thickness of the oxide film,  $V_T = k \cdot T / q$  is the thermal voltage, and  $\gamma$  is the local field enhancement factor [32].

While there are small differences among the  $f_2$  functions in models developed by various groups [9, 10, 15, 31], they differ mainly in the definitions of fitting parameters. A property they all share is that the sign of  $f_2$  is the same as that of  $(-\sinh(vtb))$ . Put in other words,  $gap$  begins to decrease whenever  $vtb$  is positive, and vice versa, as illustrated in Figure 7 (a). While there is some physical truth to this statement, considering that an RRAM device will eventually be destroyed if applied a constant voltage for an indefinite amount of time, for the model to work in numerical simulation, the state variable  $gap$  has to be bounded.

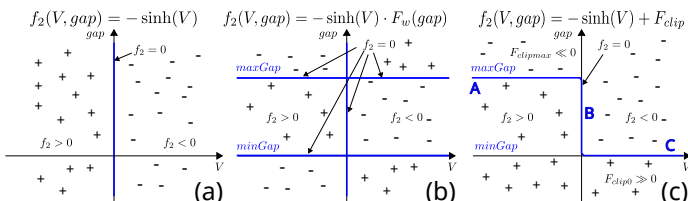


Fig. 7: Illustration of several choices of  $f_2$  in RRAM model.

Ensuring that the upper and lower bounds for  $gap$  are always respected in simulation is one major challenge for the compact modelling of RRAM devices. To address this challenge, several approaches have been attempted in the existing RRAM compact models. Some models [9, 10] directly use if-then-else statements on  $gap$ . They declare  $gap$  as a real variable in Verilog-A, then directly enforce “if ( $gap < 0$ )  $gap = 0$ ;”. This practice excludes the model from the differential equation framework; they are not suitable for simulation analyses. Another category of models multiply the  $f_2$  in (15) with a window function [11, 15, 31] that sets  $\frac{d}{dt}gap = f_2 = 0$  when  $gap = maxGap$  and  $gap = minGap$ . Such a window function is constructed either by directly using  $step()$  functions [31], or by adapting from some smooth windows, such as Joglekar [33], Biolek and Prodromakis windows [34]. However, there are subtle and deeper problems with this approach. The problems can be illustrated by analyzing the sign and zero-crossings of function  $f_2$  shown in Figure 7 (b). The  $f_2 = 0$  curves consist of three lines: the  $maxGap$  and  $minGap$  lines, and the  $V = 0$  line, with two intersections. Some parts of these lines, such as the ones for ( $gap > maxGap$ ), ( $V > 0$ ,  $gap \approx maxGap$ ) and ( $V < 0$ ,  $gap \approx minGap$ ), are model artifacts rather than physical steady state solutions. The existence of multiple DC curves can result in unphysical results in DC and transient simulations, and also cause convergence issues for homotopy analysis [18].

In this paper, we try to bound variable  $gap$  while keeping the DC solutions in a single continuous curve, illustrated as the  $f_2 = 0$  curve in Figure 7 (c). This is inspired by studying the model template `hys_example` in Sec. II. The sign and zero-crossing of  $f_2$  for our RRAM model are closely related to those of the  $f_2$  function for `hys_example` (shown in (5)). To construct the desired  $f_2 = 0$  solution curve, we modify the original  $f_2$  in (15) by adding clipping terms to it that are smooth and continuous. The clipping terms can also leave the values from the original  $f_2$  function in (15) almost intact when  $minGap < gap < maxGap$ . The code implementations in ModSpec and Verilog-A are shown in Listing 5 and Listing 6 respectively. Simulations in various simulators confirm that the models work robustly; some transient simulation results from MAPP are provided in Figure 8.

```

1 function MOD = RRAM_ModSpec()
2   MOD = ee_model();
3   MOD = add_to_ee_model(MOD, 'name', 'RRAM');
4   MOD = add_to_ee_model(MOD, 'terminals', {'t', 'b'});
5   MOD = add_to_ee_model(MOD, 'explicit_outs', {'itb'});
6   MOD = add_to_ee_model(MOD, 'internal_unks', {'Gap'});
7   MOD = add_to_ee_model(MOD, 'implicit_eqn_names', ...
8     {'dGap'});
9   MOD = add_to_ee_model(MOD, 'parms', {'g0', 0.25, ...
10     'V0', 0.25, 'I0', 1e-3, 'Vel0', 10, ...
11     'Beta', 0.8, 'gamma0', 16, 'Ea', 0.6, ...
12     'a0', 0.25, 'tox', 12});
13   MOD = add_to_ee_model(MOD, 'parms', {'maxGap', 1.7, ...
14     'minGap', 0, 'maxslope', 1e15, ...
15     'smoothing', 1e-8, 'Kclip', 50, 'GMIN', 1e-12});
16   MOD = add_to_ee_model(MOD, 'fqi', {'fe', 'qe', 'fi', 'qi'});
17   MOD = finish_ee_model(MOD);
18 end
19
20 function out = fe(S)
21   v2struct(S);
22   out = I0*safeexp(-Gap/g0, maxslope) ...
23     * safesinh(vtb/V0, maxslope) + GMIN*vtb;
24 end
25
26 function out = qe(S)
27   out = 0; % itb
28 end
29

```

```

30 function out = fi(S)
31     v2struct(S);
32     T = 300;
33     k = 1.3806226e-23; % Boltzmann's Constant
34     q = 1.6021918e-19; % Electron Charge
35     Gamma = gamma0 - Beta * Gap^3;
36     ddt_Gap = - Vel0 * exp(- q*Ea/k/T) ...
37         * safesinh(vtb*Gamma*a0/tox*q/k/T, maxslope);
38     Fw1 = smoothstep(minGap-Gap, smoothing);
39     Fw2 = smoothstep(Gap-maxGap, smoothing);
40     clip_minGap = (safeexp(Kclip*(minGap-Gap), maxslope) ...
41         - ddt_Gap) * Fw1;
42     clip_maxGap = (-safeexp(Kclip*(Gap-maxGap), maxslope) ...
43         - ddt_Gap) * Fw2;
44     out = ddt_Gap + clip_minGap + clip_maxGap;
45 end
46
47 function out = qi(S)
48     v2struct(S);
49     out = - 1e-9 * Gap;
50 end

```

Listing 5: RRAM\_ModSpec.m

```

1  'include "disciplines.vams"
2  'include "constants.vams"
3  module RRAM(t, b);
4      inout t, b;
5      electrical t, b, nGap;
6      parameter real g0 = 0.25 from (0:inf);
7      parameter real V0 = 0.25 from (0:inf);
8      parameter real Vel0 = 10 from (0:inf);
9      parameter real I0 = 1e-3 from (0:inf);
10     parameter real Beta = 0.8 from (0:inf);
11     parameter real gamma0 = 16 from (0:inf);
12     parameter real Ea = 0.6 from (0:inf);
13     parameter real a0 = 0.25 from (0:inf);
14     parameter real tox = 12 from (0:inf);
15
16     parameter real maxGap = 1.7 from (0:inf);
17     parameter real minGap = 0.0 from (0:inf);
18
19     parameter real smoothing = 1e-8 from (0:inf);
20     parameter real GMIN = 1e-12 from (0:inf);
21     parameter real Kclip = 50 from (0:inf);
22
23     real Gap, ddt_gap, Gamma, Fw1, Fw2;
24     real clip_minGap, clip_maxGap;
25
26     analog function real smoothstep;
27         input x, smoothing;
28         real x, smoothing;
29         begin
30             smoothstep = 0.5*(x/sqrt(x*x + smoothing)+1);
31         end
32     endfunction // smoothstep
33
34     analog begin
35         Gap = V(nGap, b);
36         I(t, b) <+ I0 * limexp(-Gap/g0) * sinh(V(t, b)/V0)
37             + GMIN*V(t, b);
38
39         Gamma = gamma0 - Beta * pow(Gap, 3);
40         ddt_gap = -Vel0 * exp(-Ea/$vt) * sinh(V(t, b)
41             * Gamma*a0/tox/$vt);
42
43         Fw1 = smoothstep(minGap-Gap, smoothing);
44         Fw2 = smoothstep(Gap-maxGap, smoothing);
45         clip_minGap = (limexp(Kclip*(minGap-Gap))
46             - ddt_gap) * Fw1;
47         clip_maxGap = (-limexp(Kclip*(Gap-maxGap))
48             - ddt_gap) * Fw2;
49
50         I(nGap, b) <+ ddt_gap + clip_minGap + clip_maxGap;
51         I(nGap, b) <+ ddt(-1e-9*Gap);
52     end
53 endmodule

```

Listing 6: RRAM.va

While the intention of adding the clipping terms is to set up bounds for variable  $gap$  and to construct DC solution curve in Figure 7 (c), there is also some physical justification to our approach. As a physical quantity,  $gap$  is indeed bounded by

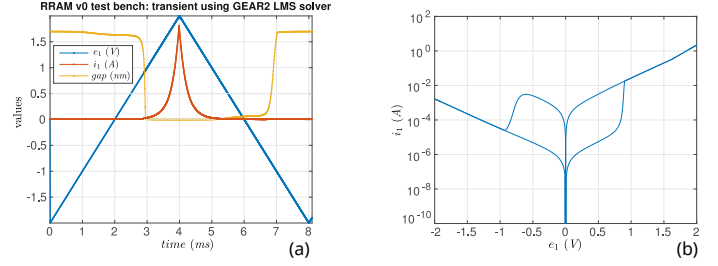


Fig. 8: Transient results on the circuit (same as in Figure 3) with a voltage source connected to an RRAM device.

definition. Therefore,  $\frac{d}{dt}gap = f_2$  cannot look like Figure 7 (a) or (b) in reality. The  $f_2 = 0$  curves must have the A and B parts labelled in Figure 7 (c). One can think of the clipping terms as infinite amount of resisting “force” to keep  $gap$  from decreasing below  $minGap$ , or increasing beyond  $maxGap$ . The analogy is the modelling of MEMS switches, where the switching beam’s position is often used as an internal state variable. This variable reaches its bound when the switching beam hits the opposing electrode (often the substrate). The position does not move further. The beam cannot move into the electrode because of the huge force resisting it from causing any shape change in the structures. Similarly, in RRAM modelling, if the variable  $gap$  represents its physical meaning accurately, one can expect such “forces” to exist to make it a bounded quantity. This physics intuition matches well with our proposed numerical technique of using fast growing exponential components to enforce the bounds.

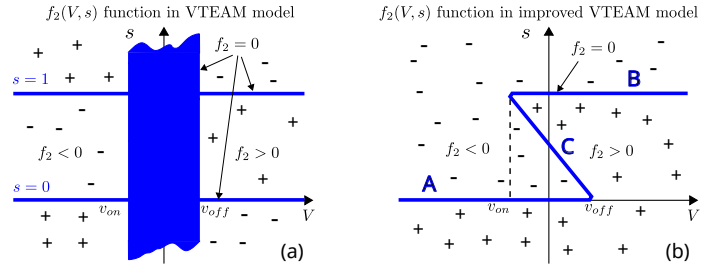


Fig. 9:  $f_2$  function in VTEAM memristor model contains a flat region around  $V = 0$  for the modelling of DC hysteresis. The proper way is to design a single solution curve of  $f_2 = 0$  that folds back around  $V = 0$ .

Note that the DC curve for the RRAM model in Figure 7 (c) does not bend with an opposite slope in the middle; the model is at the “cusp” of hysteresis and multistability. This matches physical reality, since RRAMs are considered non-volatile memories only within a certain data retention time. Apart from RRAMs, several compact models are designed for other memristive devices, including those with true DC hysteresis. As mentioned in Sec. I, they [11, 15] use regions with flat  $f_2$  functions for modelling the multistability, resulting in the zero-crossings of  $f_2$  forming an area shown in Figure 9 (a). From the discussion in this paper, to model the same effect in these memristive devices while respecting the well-posedness of the model, the steady state curve should resemble the one shown in Figure 9 (b) instead. Once we have this understanding, it becomes easy to design the  $f_2$  functions in more memristor models [18] to bring about the desired DC solution curve as in Figure 9 (b).

## V. HBT Model with Thermal Effects

It has been known that HBT models with thermal effects can have issues with multiple stable DC operating points [4]. This phenomenon can create convergence issues in simulation, and is also in general confusing to both model developers and circuit designers. Its origin is discussed in [4]. In this section, we show that the reason multistability and hysteresis arise from a perfectly smooth model can again be explained by a fold in the DC solution curve. Furthermore, we also show how this insight can be used to modify this model behaviour.

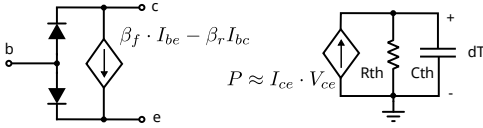


Fig. 10: Simplified electro-thermal HBT model.

The core of a HBT DC model with thermal effects is sketched in Figure 10. It consists of two back-to-back diodes for modelling the junctions in a HBT device; the bipolar amplifying effect is modelled via a voltage-controlled current source (VCCS), multiplying the forward and reverse diode currents by  $\beta_F$  and  $\beta_R$  respectively. The thermal effect is modelled by modulating  $\beta_F$  with temperature:

$$\beta_F = \beta_{F0} - d\beta_F \cdot dT, \quad (16)$$

where  $dT$  is the temperature relative to room temperature. The higher  $dT$ , the lower  $\beta_F$ , the smaller the amplifying effects. The dynamics of temperature is governed by the power dissipation (mostly  $V_{ce} \cdot I_{ce}$ ), and the thermal resistance/capacitance associated the dissipation.

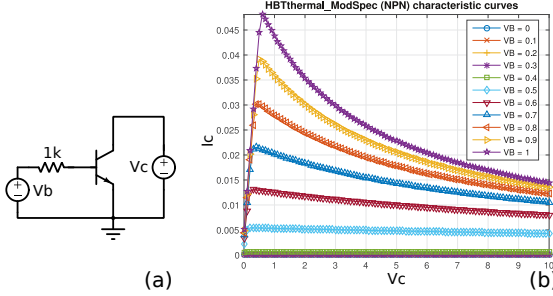


Fig. 11: Test bench and characteristic curves of the electro-thermal HBT model.

Without thermal effects, when the HBT is biased as in Figure 11, the model generates characteristic curves with a saturated  $I_{ce}$  after  $V_{ce}$  reaches certain thresholds. With thermal effects, as  $V_{ce}$  increases, power dissipation grows, leading to a rising temperature, which in return lowers  $I_{ce}$ . As a result, the characteristic curves bend down, as shown in Figure 11 (b). With certain choices of parameters, the curves can begin crossing each other [4] — the same  $I_{ce}$  can result from different  $V_{be}$  biases.

The existence of multistability indicates that when the current drawn from a HBT device is varied smoothly, the number of solutions for the voltage biases can change, resulting in sudden changes in voltages. As an illustration, in Figure 12, we connect a HBT device in diode mode, with a 1k resistor between its collector and base nodes. When different currents are drawn from the device, multistability in voltages can be observed as discontinuous solutions in DC sweeps. Again, the reason discontinuity arises from smooth model equations is that there is a fold in the steady state curve. Homotopy can be applied to

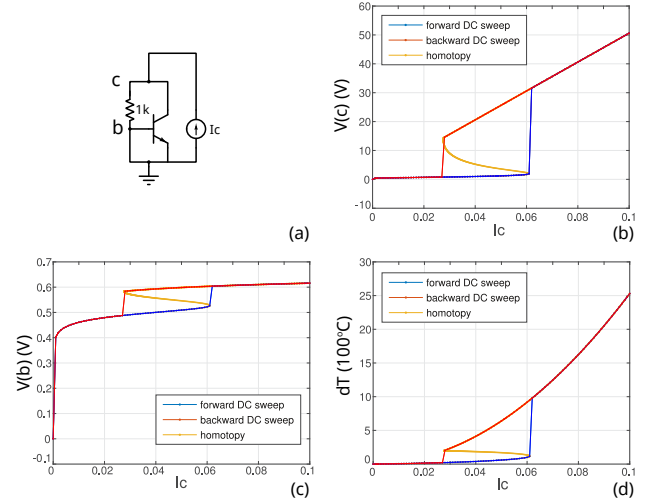


Fig. 12: Results from DC sweeps and homotopy for a test circuit of a HBT device.

track this curve and also calculate the unstable solutions missing from DC sweeps. In each slice of the state space, *i.e.*, in the  $V(c)$ - $I_c$ ,  $V(b)$ - $I_c$ ,  $dT$ - $I_c$  plots in Figure 12, a fold that completes the disconnected DC sweep curves can be seen.

There are several implications from these results. Firstly, it confirms that the multistability and hysteresis observed in HBT models indeed also arise from a fold in the DC curve; it agrees with our generic formalism in this paper nicely. The results also show the usefulness of homotopy analysis in the analysis and understanding of device models.

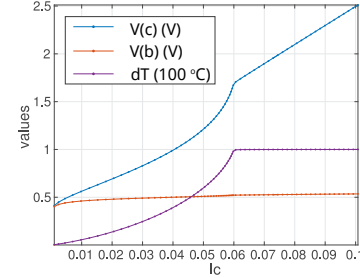


Fig. 13: DC solutions of the test circuit in Figure 12 with the modified HBT model.

Furthermore, the understanding gained from the results and discussions in this paper can help remove the multistability in HBT models when it is undesired in some applications. From Figure 12, multiple stable temperatures constitute part of the fold. If we set up an upper bound for temperature, *i.e.*, we prevent it from growing high enough to enter the folding back region, the entire DC solution curve will change shape, *i.e.*, multistability in not only the temperature, but also  $V(b)/V(c)$  voltages will all disappear. Setting up a  $dT_{max}$  has been mentioned in [4], but there is no description of how it can be achieved in a physical way without turning the HBT model into an analog behavioural model. From our discussion on the memristor models, similar clipping effect can be designed by changing the thermal resistance  $R_{th}$  to be nonlinear:

$$R_{th} = \text{smoothstep}(dT_{max} - dT) \cdot R_{th0}, \quad (17)$$

such that its value decreases significantly as temperature gets close to the upper limit, thus dissipating more energy to prevent the temperature from building up further. Similar to the memristor examples, the clipping effect of  $R_{th}$  also has physical justifications, as in real devices, the power dissipation



of materials is often nonlinear so that the temperature is unlikely to grow too high.

In conclusion of the discussion on the HBT model, the methodology we study in this paper not only helps design models with the desired hysteric behaviours, but also helps explain the origin of the multistability found in existing device models, and guide us in modifying the model characteristics.

## VI. Stoner-Wohlfarth Model for Ferromagnets

Another place where hysteresis is often observed is in magnetic materials and devices. Modelling magnetic hysteresis is not only important for simulating circuits with magnetic components, such as iron cores in transformers, but also useful in analyzing and understanding the underlying physics. The Stoner-Wohlfarth (SW) model [23] is a classic, widely-used model for the magnetization in single-domain ferromagnets. The core of SW model is the formulation of an energy with respect to the angle of magnetization:

$$E(h, \varphi) = \frac{1}{4} - \frac{1}{4} \cos(2(\varphi - \theta)) - h \cos(\varphi), \quad (18)$$

where  $\varphi$  describes the direction of the internal magnetization  $M$ , i.e.,  $M = \cos(\varphi)$ ;  $h$  is the strength of the external magnetic field;  $\theta$  is a fixed angle between the easy axis of the magnet and the axis of  $h$ . The energy landscapes  $E(h, \varphi)$  with  $\theta = \pi/4$  and different  $h$  values are plotted in Figure 14.<sup>2</sup> When there is no external magnetization field, i.e.,  $h = 0$ , the energy landscape corresponds to the sinusoidal curve in the middle of the plot. According to this curve, there is a steady-state solution in  $\varphi$  that locally minimizes the energy function after every  $\pi$  distance. These solutions correspond to two stable values for  $M$ . When  $h$  increases or decreases beyond certain thresholds, one of the two steady states in  $M$  vanishes. As visualized in Figure 14, when  $h$  is swept back and forth,  $\varphi$  jumps from valley to valley in the energy landscape, generating hysteresis in  $M$ .

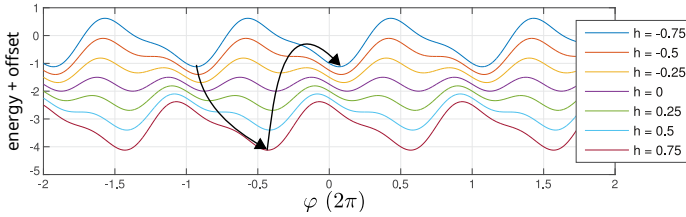


Fig. 14: Energy landscapes of SW model with different external magnetic field  $h$ .

The SW model is known to have problems in simulation [35]. Since the model is based on the magnetization angle, any solution shifted by  $2k\pi$  is also a solution. The problem is exacerbated by the fact that  $\varphi$  grows monotonically when  $h$  is swept, as illustrated in Figure 14, resulting in unbounded  $\varphi$  in simulation.

The magnetic hysteresis from the SW model can also be analyzed through calculating the steady state solutions using the homotopy algorithm. The homotopy results, plotted in Figure 15, are different from other examples we have studied in this paper. There is no single DC curve. Instead, there are infinite number of separated ones, and as  $h$  is swept, the DC solution in  $\varphi$  jumps up from one to another. Evidently, the lack of a single DC solution curve is connected with the unbounded  $\varphi$  problem in the SW model.

<sup>2</sup>Offsets are added to the plots to separate them from each other vertically.

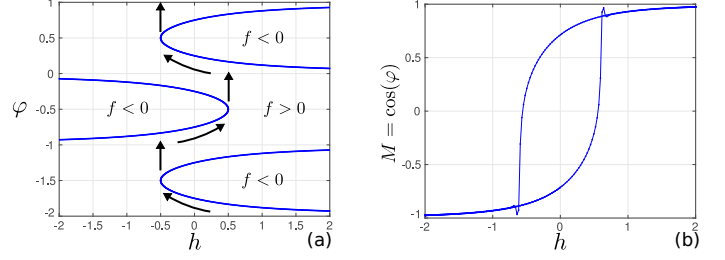


Fig. 15: The steady state solutions of in the SW model from homotopy analysis and magnetic hysteresis from transient  $h$  sweeping.

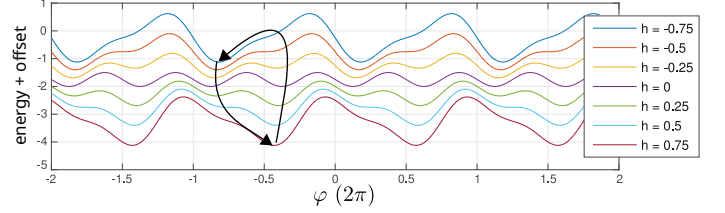


Fig. 16: Energy landscapes of the modified SW model.

From our observation, the reason  $\varphi$  becomes unbounded is that under the perturbation of  $h$ , it is always moving to the right when  $\theta > 0$ , while in fact, it is equivalent if we flip both input  $h$  and state  $\varphi$  around  $\theta$  to make  $\varphi$  jump to the left instead. If the flipping is done for only  $h < 0$ ,  $\varphi$ 's movement should be contained,<sup>3</sup> as illustrated in Figure 17. The partially flipped version of the energy landscape in this modified model is as follows.

$$\hat{E}(h, \hat{\varphi}) = \frac{1}{4} - \frac{1}{4} \cos(2(\hat{\varphi} - \theta)) - h \cos(\hat{\varphi} - 2\theta \cdot \text{smoothstep}(-h)), \quad (19)$$

where  $\hat{\varphi}$  is the new variable for simulation, and is related to the original  $\varphi$  through

$$\varphi = \hat{\varphi} - 2\theta \cdot \text{smoothstep}(-h). \quad (20)$$

The modified SW model is implemented as a DAE object in MAPP [21] in Listing 7. It can also be written with MAPP's multiphysics modules [36], or with Verilog-A's magnetic discipline [5]. The modified model still has an energy landscape smoothly varying with input  $h$  but is better numerically, in the sense that  $\hat{\varphi}$  does not grow without bound in simulation. This improvement corresponds to a reshaped DC curve for the model. In Figure 17, we plot all the steady state solutions within  $-1 \leq h \leq 1$ . The DC solutions in the improved model indeed form a single curve with a similar shape as the other examples

<sup>3</sup>In this case, any other solution shifted by  $2k\pi$  is still a solution, but this can be easily fixed by adding clipping terms beyond one cycle to eliminate other solutions.

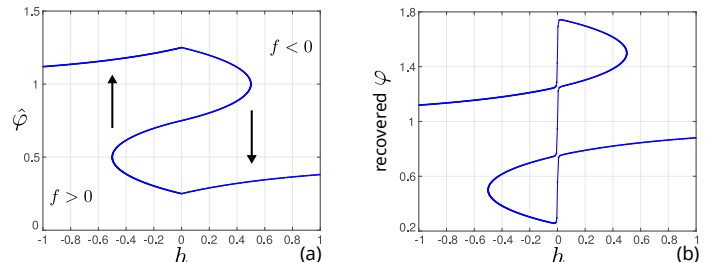


Fig. 17: Homotopy results of the steady state solutions in the modified SW model.

in the previous sections; the multistability indeed comes from the fold in the continuous DC curve in the state space. The reconstructed  $\phi$  corresponding to this curve is plotted in Figure 17 (b). The solutions in  $\phi$  still form a single continuous curve, due to the smooth function used in (20). The cos values of  $\phi$ , corresponding to magnetization  $M$ , form the same  $M$ - $h$  hysteresis loop as shown in Figure 15 (b).

```

1 function DAE = SWmagnet_DAE()
2   DAE = init_DAE();
3   DAE = add_to_DAE(DAE, 'name', ...
4     'Stoner-Wohlfarth model for ferromagnets');
5   DAE = add_to_DAE(DAE, 'uname(s)', {'phi_hat'});
6   DAE = add_to_DAE(DAE, 'eqname(s)', {'phi_hat_dot'});
7   DAE = add_to_DAE(DAE, 'inputname(s)', {'h'});
8   DAE = add_to_DAE(DAE, 'parm(s)', {'theta', pi/4, ...
9     'tau', 1e-6, ...
10    'maxslope', 1e15, ...
11    'Kclip', 10, ...
12    'smoothing', 1e-5});
13   DAE = add_to_DAE(DAE, 'f', @f);
14   DAE = add_to_DAE(DAE, 'q', @q);
15   DAE = finish_DAE(DAE);
16 end
17
18 function fout = f(S)
19   v2struct(S);
20   phi_hat_dot = 0.5 * sin(2 * (phi_hat - theta)) + ...
21     h * sin(phi_hat - smoothstep(h, smoothing)*2*theta);
22   clip_0 = safeexp(Kclip*(-phi_hat), maxslope) ...
23     * smoothstep(-phi_hat, smoothing);
24   clip_3pi = -safeexp(Kclip*(phi_hat-3*pi), maxslope) ...
25     * smoothstep(phi_hat-3*pi, smoothing);
26   fout = phi_hat_dot + clip_0 + clip_3pi;
27 end
28
29 function gout = q(S)
30   gout = [S.tau * S.phi_hat];
31 end

```

Listing 7: SWmagnet\_DAE.m

We would like to note that the modified SW model is only equivalent to the original one in DC responses; during transient simulations,  $\phi$  from (19) and  $\phi$  from (18) won't correspond to each other exactly. But since the original SW model is itself a steady state model for magnetization, the modified one with the proposed flipping mechanism is equally useful in the analysis of magnetic hysteresis.

## VII. Summary

In this paper, we have developed a generic formalism for hysteretic devices. We have shown how abrupt changes in device characteristics can result from entirely smooth model equations, and how to model these properties correctly by designing a fold in the model's steady state solution curve. With this understanding, we have been able to design, analyze, and improve upon several models for various devices, including ESD clamps, RRAMs/memristors, HBTs, and the SW model for ferromagnets. The result is not only a class of compact models that work robustly in simulation, but also a further understanding of the multistability and hysteresis in devices.

## Acknowledgments

The author would like to thank Colin McAndrew for suggesting the study on ESD snapback modelling, and for many useful comments on an initial draft of the paper. The author would also like to thank Eric Keiter for bringing the uniqueness problem in HBT models to our attention. This work was supported through the NCN-NEEDS program, which is funded by the NSF (contract

1227020-EEC) and by the SRC.

## References

- [1] R. Ida and C.C. McAndrew. A physically-based behavioral snapback model. In *Electrical Overstress/Electrostatic Discharge Symposium (EOS/ESD)*, pages 1–5. IEEE, 2012.
- [2] L.O. Chua. Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, 18(5):507–519, 1971.
- [3] S.R. Williams. How We Found The Missing Memristor. *IEEE Spectrum*, 45(12):28–35, December 2008.
- [4] M. Rudolph. Uniqueness problems in compact HBT models caused by thermal effects. *IEEE Transactions on Microwave Theory and Techniques*, 52(5):1399–1403, 2004.
- [5] M.C. Williams, R.S. Vogelsong, and K.S. Kundert. Simulation and modeling of nonlinear magnetics. In *IEEE International Symposium on Circuits and Systems*, volume 1, pages 736–739. IEEE, 1995.
- [6] K. Kundert. Hidden State in SpectreRF. *The Designer's Guide*, 2013.
- [7] C.C. McAndrew, G.J. Coram, K.K. Gullapalli, J.R. Jones, L.W. Nagel, A.S. Roy, J. Roychowdhury, A.J. Scholten, Geert D.J. Smit, X. Wang and S. Yoshitomi. Best Practices for Compact Modeling in Verilog-A. *IEEE J. Electron Dev. Soc.*, 3(5):383–396, September 2015.
- [8] L. Wei, C.E. Gill, W. Li, R. Wang and M. Zunino. A convergence robust method to model snapback for ESD simulation. In *CAS 2011 Proceedings (2011 International Semiconductor Conference)*, 2011.
- [9] Z. Jiang, S. Yu, Y. Wu, J.H. Engel, X. Guan and H.-S. P. Wong. Verilog-A compact model for oxide-based resistive random access memory (RRAM). In *International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, pages 41–44. IEEE, 2014.
- [10] P.-Y. Chen and S. Yu. Compact Modeling of RRAM Devices and Its Applications in 1T1R and 1S1R Array Design. *IEEE Transactions on Electron Devices*, 62(12):4022–4028, 2015.
- [11] S. Kvatinsky, E.G. Friedman, A. Kolodny and U.C. Weiser. TEAM: threshold adaptive memristor model. *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, 60(1):211–221, 2013.
- [12] D. FitzPatrick and I. Miller. *Analog behavioral modeling with the Verilog-A language*. Springer Science & Business Media, 2007.
- [13] T. Wang and J. Roychowdhury. Guidelines for Writing NEEDS-compatible Verilog-A Compact Models. <https://nanohub.org/resources/18621>, Jun 2013.
- [14] Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, 13(49-52):28, 1902.
- [15] C. Yakopcic, T.M. Taha, G. Subramanyam and R.E. Pino. Generalized memristive device SPICE model and its application in circuit design. *IEEE Trans. CAD*, 32(8):1201–1214, 2013.
- [16] J. Roychowdhury. Numerical simulation and modelling of electronic and biochemical systems. *Foundations and Trends in Electronic Design Automation*, 3(2-3):97–303, December 2009.
- [17] Sybil P Parker. *McGraw-Hill Dictionary of Scientific and Technical Terms*. McGraw-Hill Book Co., 1989.
- [18] T. Wang and J. Roychowdhury. Well-Posed Models of Memristive Devices. *arXiv preprint arXiv:1605.04897*, 2016.
- [19] J. Roychowdhury and R. Melville. Delivering Global DC Convergence for Large Mixed-Signal Circuits via Homotopy/Continuation Methods. *IEEE Trans. CAD*, 25:66–78, Jan 2006.
- [20] D. Amsallem and J. Roychowdhury. ModSpec: An open, flexible specification framework for multi-domain device modelling. In *Computer-Aided Design (ICCAD), 2011 IEEE/ACM International Conference on*, pages 367–374. IEEE, 2011.
- [21] T. Wang, K. Aadithya, B. Wu, J. Yao, and J. Roychowdhury. MAPP: The Berkeley Model and Algorithm Prototyping Platform. In *Proc. IEEE CICC*, pages 461–464, September 2015. DOI link.
- [22] T. Wang, A.V. Karthik, B. Wu and J. Roychowdhury. MAPP: A platform for prototyping algorithms and models quickly and easily. In *IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, pages 1–3. IEEE, 2015.
- [23] E.C. Stoner and E.P. Wohlfarth. A mechanism of magnetic hysteresis in heterogeneous alloys. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 240(826):599–642, 1948.
- [24] C.H. Diaz and S. Kang. New algorithms for circuit simulation of device breakdown. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11:1344–1354, 1992.
- [25] C. Jiao and Z. Yu. A robust novel technique for SPICE simulation of ESD snapback characteristic. In *8th International Conference on Solid-State and Integrated Circuit Technology Proceedings*, pages 1367–1369. IEEE, 2006.
- [26] Y. Zhou, D. Connerney, R. Carroll, and T. Luk. Modeling MOS snapback for circuit-level ESD simulation using BSIM3 and VBIC models. In *Sixth international symposium on quality electronic design*, pages 476–481. IEEE, 2005.
- [27] J. Li, S. Joshi, R. Barnes, E. Rosenbaum. Compact modeling of on-chip ESD protection devices using Verilog-A. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1047–1063, 2006.
- [28] Y. Zhou, J. Hajjar, A.W. Righter, and K.P. Lisiak. Modeling Snapback of LVTSCR Devices for ESD Circuit Simulation Using Advanced BJT and MOS Models. In *Electrical Overstress/Electrostatic Discharge Symposium (EOS/ESD)*, volume 29, page 175, 2007.
- [29] R. Mertens, E. Rosenbaum. A physics-based compact model for SCR devices used in ESD protection circuits. In *Reliability Physics Symposium (IRPS), 2013 IEEE International*, pages 2B–2. IEEE, 2013.
- [30] P.A. Juliano, E. Rosenbaum. A novel SCR macromodel for ESD circuit simulation. In *Electron Devices Meeting, 2001. IEDM '01. Technical Digest. International*, pages 14–3. IEEE, 2001.
- [31] P. Sheridan, K.-H. Kim, S. Gaba, T. Chang, L. Chen and W. Lu. Device and SPICE modeling of RRAM devices. *Nanoscale*, 3(9):3833–3840, 2011.
- [32] J. McPherson, J. Kim-Y., A. Shanware and H. Mogul. Thermochemical description of dielectric breakdown in high dielectric constant materials. *Applied Physics Letters*, 82:2121, 2003.
- [33] C. Yakopcic. *Memristor Device Modeling and Circuit Design for Read Out Integrated Circuits, Memory Architectures, and Neuromorphic Systems*. PhD thesis, University of Dayton, 2014.
- [34] S. Kvatinsky, K. Talisveyberg, D. Fliter, E.G. Friedman, A. Kolodny and U.C. Weiser. Verilog-A for memristors models. In *CCIT Tech. Rep. 801*, 2011.
- [35] C. Tannous and J. Gieraltowski. The Stoner-Wohlfarth model of ferromagnetism. *European journal of physics*, 29(3):475, 2008.
- [36] T. Wang and J. Roychowdhury. Multiphysics Modelling and Simulation in Berkeley MAPP. In *Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO), 2016 IEEE MTT-S International Conference on*, pages 1–3. IEEE, 2016.