

MODELING TECHNIQUES FOR CAPACITIVE HARVESTERS OF VIBRATION ENERGY

Dimitri Galayko¹, Ken Caluwaerts¹, Philippe Basset²

¹LIP6, University of Paris-VI, France, ²University Paris-Est, ESYCOM/ESIEE, France

Abstract This paper reports on a comparative study of the modeling techniques for a capacitive resonant vibration energy harvester composed of a resonator, a capacitive transducer and a conditioning circuit. Three accurate models were implemented in three different environments: VHDL-AMS language, SystemC-AMS plate-form and Simulink (Matlab). However, to reduce the modeling time, we also created a simplified Scicos model in which the mechanical part is modeled using SDF blocs such as integrators and adders, and the conditioning circuit was modeled as a Matlab-like script. This simplified model was the most efficient regarding the simulation time, while keeping very good accuracy of the results.

Keywords: vibration energy harvesting, AMS modeling, Scilab, Scicos, VHDL-AMS, SystemC-AMS

1 INTRODUCTION

This paper presents a comparative study of different modeling approaches for electromechanical energy harvesters employing a capacitive electromechanical transducer. Theoretical study of the modeled system has been presented in [1]. The modeled system includes a mechanical block (resonator), a capacitive transducer and a conditioning electronic circuit (fig. 1) [2], [3]. The conditioning circuit is composed of a charge pump responsible for the electromechanical energy conversion, a flyback circuit responsible for the storage of the harvested energy and a switch activating the flyback at appropriate timing [3]. In our models, switching happens when the voltage V_{store} crosses one of two thresholds, V_1 and V_2 : the switch operates like a finite-state automaton (fig. 2). This study presents four modeling approaches, allowing both precise and "macroscopic" modeling of the system.

The first models of the system was based on a mixed VHDL-AMS/ELDO modeling technique. This model, implemented in the Mentor Graphics Ad-

vanceMS environment, provided very accurate results with reasonable modeling time (tens of minutes), for modeling few seconds of the system behaviour, with a vibration frequency of 300 Hz. VHDL-AMS model is precise, and allows an insight into the details of the circuit operation. This model was very useful for the study of the harvesting mechanism.

Our previous studies highlighted that to get an optimal power yield, one needs an intelligent energy management and an adaptation of the harvester operation to the environmental conditions. To carry out the study of the system-level power management, a long-term simulation was needed. For this, the VHDL-AMS precise model was not a good candidate. For this reason, we developed a model in Scilab/Scicos (a freeware Matlab/Simulink counterpart developed by INRIA [4]). This model is behavioural and functional, its goal is to highlight a global behaviour without getting an insight into the physics of the block operation. This simplified model is platform-independent, and can be implemented in the most behavioural modeling plate-forms (Matlab, VHDL-AMS, SystemC-AMS, etc...).

Implementation of intelligent power management requires the use of digital/software blocks, which are often modeled in SystemC [5], and which need to be interfaced with the low-level electromechan-

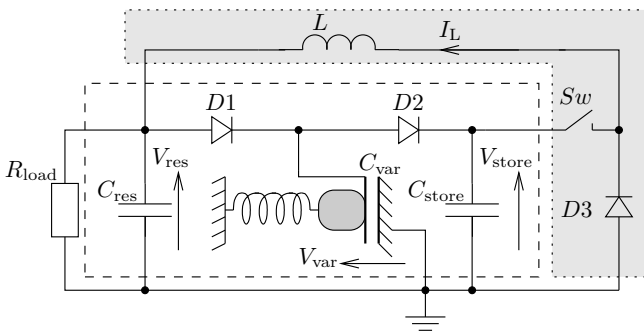


Figure 1: The studied system diagram [2].

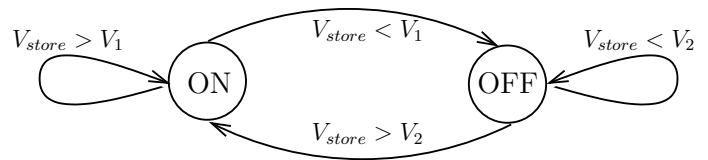


Figure 2: Switch behavioural model.

ical model. For this reasons, we also developed a SystemC-AMS model of the harvester. Such a model is naturally compatible with a SystemC model of the digital part. The main challenge of SystemC-AMS modeling was the representation of non-linear and switching elements (diodes, switches).

To complete our comparative modeling study, we also implemented a precise harvester model in the Simulink environment.

The paper is organized in the following way. In sections 2 and 3 we briefly present the VHDL-AMS and SystemC-AMS models. In section 4 the simplified scilab model will be presented, section 5 focused on the presentation of the Simulink model.

2 VHDL-AMS model

For VHDL-AMS modeling, the system has been subdivided into four blocks: the resonator+transducer, the charge pump, the flyback and the switch. The resonator/transducer and the switch blocks were modeled in VHDL-AMS language. For the resonator/transducer block, the corresponding physical differential equations were encoded in the VHDL-language. For the switch, the model of fig. 2 was directly implemented. The switch was modeled as a controlled resistor with different resistance values at the ON and OFF states.

The electrical blocks were modeled as Eldo models. The diodes were represented by their exponential models, with zero threshold voltage.

The modeling results fit well with the analytical prediction. More information about the model and the results can be found in [3].

3 SystemC-AMS model

There were two motivations to develop a SystemC-AMS model. Firstly, at our knowledge, this is the only modeling platform which allows mixing software blocks with low-level physical subsystems in the same model. Secondly, contrary to the VHDL-AMS modeling platforms, SystemC-AMS is FOSS (Free and Open Source Software), which makes it attractive for use in academic environment.

SystemC-AMS (version 0.15RC5) is a library of classes written in C++ language, and programming in SystemC-AMS corresponds, roughly, to writing a simple C++ program [5]. Here we present shortly the modeling facilities of SystemC-AMS and summarize their use for the harvester model building.

A SystemC-AMS offers two modeling tools: a Timed Data Flow solver (TDF), and Linear Electrical

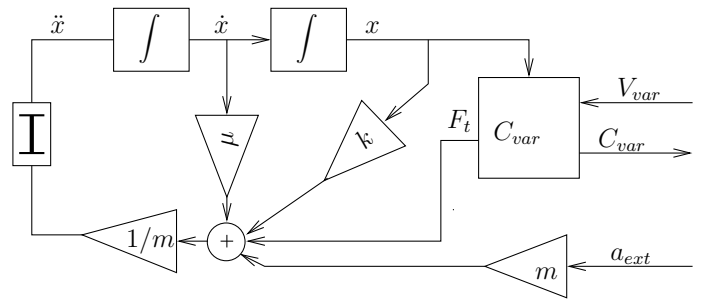


Figure 3: Timed data flow model of resonator/transducer block.

Network solver (LinElec).

The TDF solver allows to model systems composed of unidirectional mathematical blocks, like Simulink or Scicos solvers. However, there are two differences. Firstly, SystemC-AMS imposes a fixed time step for all blocks. The time step of individual blocks can be a multiple of the "basic" minimal time step of the system, but it cannot change runtime, unlike most modeling platform based on equation solving. An another difference is that SystemC-AMS does not provide a non-linear solver, which imposes some limitations on modeling systems with strong non-linearities (like exponential functions of diodes) [6].

The Linear Electrical Network solver (LinElec) is offered for modeling of linear electrical networks. Prior to the modeling, the SystemC-AMS core solves the matrix equation corresponding to the linear network, and at runtime, closed-form algebraic expressions are used for computing electrical quantities. The LinElec and TDF models can be interconnected. Firstly, an electrical quantity can be read by a TDF module; secondly, TDF signals can change the parameter values of linear components. Thus, it is possible to model a linear electrical network with varying parameters. This property allows modeling non-linear and switching behaviour, as well as variable capacitors. For example, a diode can be modeled as a resistor which is controlled by voltage on its electrodes.

For the studied system, the mechanical part, being "slow" and weakly non-linear (the only non-linear element was the electromechanical transducer), was modeled in the TDF domain. The diagram of the corresponding second-order model is given in fig. 3.

The conditioning circuit was modeled in LinElec solver; the models were interconnected through the inputs/outputs of the blocks C_{var} (fig. 3). The numeric results were very similar to those of the VHDL-AMS model (under 3% of relative difference).

4 Simplified high-level Simulink-like model

In both VHDL-AMS and SystemC-AMS models, although the results were very precise, the modeling simulation time was excessively large for long-term system-level modeling. This was because of the non-linear elements (diodes) of the conditioning circuit.

For this reason, we proposed to model mechanical system as a Signal Flow diagram in Scicos (an INRIA freeware, similar to Simulink), in which the transducer was represented by a single block encapsulating both the transducer and the conditioning circuit. This block was defined by a Scilab script given in listing 1.

Listing 1: Scilab script modeling the conditioning circuit.

```
// function calculating the force
// input value: the current mobile mass position x
// output value: the generated electrical force
// the electrical state variable are global
global V0, Vstore, Qvar, C, Vres, V1, V2
// the transducer capacity value
C=capa(x);
dcdx=dc_dx(x); // the capacity spatial gradient
// a guess value of Vvar voltage is calculated
Vvar=Qvar/C;
// the diode threshold : supposed to be zero here
eps=1e-10;
// modeling of D1
if Vres-Vvar>eps then
// if true, D1 diode is maybe ON, and
// a new guess is calculated for Vvar
Vvar_forseen=(Qvar+Vres*Cres)/(C+Cres);
if Vvar_forseen < Vres then
// if true, the diode D1 is really on, and
// Cvar is connected with Cres
Vvar=Vvar_forseen;
Vres=Vvar; Qvar=Vvar*C;
end;
//else, Vvar equal to the firstly calculated guess
end
// modeling of D2 is identical...
// .....
// modeling of the switch and
// of the conditioning circuit described in [3]
// The charge pump is returned to an old state
// corresponding to Vstore=V1, and the harvested
// energy is put on Cres
if Vstore>V2 then
Vstore_old=Vstore;
Vres_old=Vres; Vstore=V1;
// back in the time : the value of Vres when
// Vstore was equal to V1
Vres=((Vres_old*Cres+Vstore_old*Cstore)-
```

```
Vstore*Cstore)/Cres;
// calcul. of the harvested energy from V1 to V2
W_harv=Cres*Cstore/(Cres+Cstore)*
( (Vstore_old-Vres_old)^2-
(Vstore-Vres)^2 )/2;
// updating Vres, accounting for
// the harvested energy
Vres=sqrt(2*(Cres*Vres^2/2+W_harv)/Cres);
end;
y1=Vvar^2/2*dcdx; //the block output
```

In this script, the C_{store} and C_{res} capacitors are represented by the global variables V_{store} and V_{res} storing their voltage values. The charge of the capacitor C_{var} is stored in the variable Q_{var} .

At the input, the script receives a new mobile mass position. The transducer capacitance is evaluated, and the new transducer voltage (V_{var}) is calculated. Then this voltage is compared with V_{store} and V_{res} voltages. The blocks if...end model the operation of the diodes D1 and D2: if $V_{res} > V_{var}$ or $V_{store} < V_{var}$, one of the diodes is on, and in the corresponding if...else block the charge values of C_{var} and the corresponding capacitors are updated.

The flyback circuit is modeled by an independent block if...else controlling the crossing of the V_2 threshold. If this is the case, this block if...else resets V_{store} to V_1 value, calculates the harvested energy and updates the voltage of C_{res} capacitor.

The only introduced error concerns the timing: our model assumes the flyback to be instantaneous, which leads to an advance on the real voltage evolution (cf. the section 6). It is possible to account for it by introducing an appropriate delay. However, the error is small since the flyback process takes a negligible time in the overall circuit operation [3].

5 Precise Simulink model

To complete our study, we built a Simulink model of the harvester. This model was based on the complete system of differential equations describing the operation in both the electrical and mechanical domains. These equations were represented with Signal Flow models using standard Simulink blocks.

The main difficulty was related with exponential diode models: the most of approximation methods manage badly exponential functions. This especially concerned the diode D3 playing a key role in the operating of the flyback circuit. In fact, when the switch turns off, there is a very abrupt discontinuity on the diode current and voltage, related with the cutting of the current in inductive circuit (this is a typical behaviour of a BUCK DC-DC converter on which the

Table 1: Numerical values of modeling test case

k , nm ⁻¹	m kg	μ , Nsm ⁻¹	ω , rad.s ⁻¹	a_{ext} , ms ⁻²
152.6	46e-6	2.19e-3	$2\pi \cdot 298$	$10 \cdot \sin(\omega t)$
C_{res} , F	C_{store} , F	Cp , F	L , H	R_L , Ω
10^{-6}	$3.3 \cdot 10^{-9}$	10^{-13}	$2.5 \cdot 10^{-3}$	10^{-1}
Δt , s	R_{ONDI} , Ω	R_{OFFDI} , Ω	R_{ONSW} , Ω	R_{OFFSW} , Ω
$4 \cdot 10^{-9}$	10^{-9}	10^{10}	10^{-9}	10^{15}
Switch low threshold, V_1 , V		Switch high threshold V_2 , V		
6		13		

Table 2: Simulation time for 1 s of system operation

SystemC-AMS	Simulink
Intel Core 2 Duo 2 GHz, 2 cores, 2 Gb	Dual Intel Xeon 4 cores, 3 GHz, 6 Gb
110 min	3.5 min
VHDL-AMS	Scilab
Sun Ultra-80, 4 proc. Ultrasparc II, 450 MHz, 4 GO	PowerPC G4 1.5 GHz, single core, 512 Mb
5.75 min	3 min 50 sec

flyback circuit operation is based). For this reason, we used a simplified quadratic model for the diode.

6 Modeling results

Here we present the result of this comparative study.

We used all four presented models to simulate a one second behaviour of the same harvester with the parameters presented in table 1.

The three precise models (VHDL-AMS, SystemC and Simulink) gave very similar numeric results, with relative difference less than 3%. To prove it, we simulated 1 second of the the system behaviour, and compared point by point the resulting voltages V_{store} and V_{res} saved with time interval of 0.01 s (table 2). (The V_{res} is the most important quantity, since it translates the accumulation of the harvested energy.)

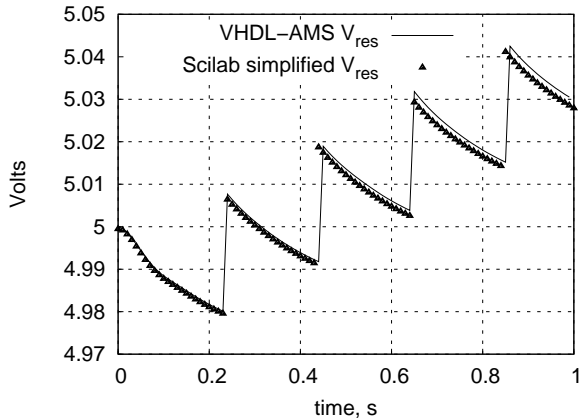


Figure 4: Evolution of V_{res} voltage from VHDL-AMS and the simplified Scilab model.

The simplified Scicos/Scilab model produced very satisfactory results as well. The plot of V_{res} obtained with Scicos is given in fig. 4. One can see that the curves are very similar, but the V_{res} issued from the Scicos model "advances" the similar plot issued from VHDL-AMS. This error can be reduced if the script given in listing 1 takes into account the delay introduced by the flyback (cf. section 6), but for system-level study this error is not significant.

7 Conclusions

The study compared four modeling techniques for a multi-physics mixed-signal energy harvesting system. The main bottleneck of the modeling is the presence of exponential diodes which made the convergence difficult, specially in the context of the Buck DC-DC converter. Electrical simulators like like Eldo or AdvanceMS are specially designed to manage such situations. For this reason, we conclude that if precise modeling is required, with a deep insight in the internal block operation, VHDL-AMS/ELDO mixed modeling is the best approach.

If a system-level or long-term simulation is required, the proposed simplified Scilab model seems to be the most appropriate. Giving very accurate results comparing with the precise models, it is easily implementable in any Scilab, Simulink, SystemC-AMS or any similar Signal Data Flow solver. It would be particularly interesting to implement this model in SystemC-AMS: such model will not suffer from hard non-linearities of the electrical network, and a time-efficient high-level mixed analog/digital/software modeling will be possible.

8 Acknowledgments

This work is partially funded by the French National Research Agency through the contract N°JC05-54551 and is supported by the French competitiveness cluster *AdvanCity*.

References

- [1] D. Galayko *et al.* *DTIP2008*, april 2008.
- [2] B. C. Yen *et al.* *IEEE transactions on Circuits and Systems*, vol. 53, february 2006.
- [3] D. Galayko *et al.* *BMAS'2007*, 2007.
- [4] www.scilab.org, "The official scilab web site,"
- [5] A. Vachoux *et al.* *ISCAS2005 proceedings*, 2005.
- [6] K. Caluwaerts *et al.* *FDL'2008*, 2007.