SystemC-AMS Modeling of an Electromechanical Harvester of Vibration Energy

Ken Caluwaerts (ken.caluwaerts@ugent.be), Dimitri Galayko (dimitri.galayko@lip6.fr)
LIP6, University of Paris-VI, France

Abstract *This paper presents the results of modeling of a mixed non-linear, strongly coupled and multidomain electromechanical system designed to scavenge the energy of ambient vibrations and to generate an electrical supply for an embedded microsystem. The system is operating in three domains: purely mechanical (the resonator), coupled electromechanical (electrostatic transducer associated with the moving mass) and electrical circuit, including switches, diodes and linear electrical components. Although only linear networks can be properly modeled in SystemC-AMS, we propose a technique allowing modeling of electrical networks including non-linear components (diodes) as well as time-varying capacitors. The whole system was modeled using two solvers of SystemC-AMS simultaneously: the one allowing TDF (Timed Data Flow) modeling and the one allowing LIN ELEC (linear electrical) circuit analysis. The modeling results are compared with VHDL-AMS and Matlab Simulink models.*

1 INTRODUCTION

Capacitive harvesters require complex conditioning circuits which have a great impact on their energetic performances, and their optimization has been the subject of numerous studies [1][2]. A typical capacitive harvester includes:

- a mechanical resonator allowing an accumulation of the mechanical energy,
- an electrostatic (capacitive) transducer, with one electrode attached to a mobile mass, and the other fixed to the system which is submitted to external vibrations,
- a conditioning electrical circuit managing the flow of electrical charges on the transducer electrodes.

This paper presents the results of modeling of a vibrational energy harvester system whose conditioning circuit architecture was proposed in [3] (Fig. 1). Our original theoretical study and modeling of this system highlighted that to achieve a maximal energy yield, a complex and intelligent feedback control is needed, which requires the use of an embedded microcontroller. Since the system is non-linear and time-varying (there are switches and diodes), it can exhibit a complex behaviour, and an accurate and system-level modeling is needed for a proper design. SystemC-AMS is one of the only modeling platforms allowing mixing physical, electrical analog, digital and software blocks in the same model, thus naturally it is an excellent candidate for the implementation of an accurate global model [4][5].

However, one of the main limitations of SystemC-AMS is the difficulty to model complex electrical circuits in the charge-conservative domain. The existing beta-version of the SystemC-AMS library only provides the tools to build TDF models (Timed Data Flow, i.e., models composed from unidirectional blocks which can be non-linear) and linear electrical system models, including only linear electrical components. The use of diodes is not allowed in the charge-conservative models of electrical circuits.

In this work we highlight that this limitation can be circumvented using the SystemC-AMS mechanism allowing the connection between the linear conservative system domain and the TDF domain. This connection is possible thanks to the mechanism allowing the inclusion of components with time-varying parameters (resistor, capacitor, voltage and current source) in the linear circuit models. This connection does exist in both directions: a TDF domain signal can control the value of the resistance in a linear circuit, and an electrical value (current or voltage) measured in the linear circuit can be applied at the input of a TDF domain block also. In this paper we present how these features allow building a SystemC-AMS model of the harvester.

The paper is organized in the following way. In section 2 we briefly present the modeled system, in section 3 we present the model of all the blocks of the harvester and in section 4 we discuss the modeling results and compare them with a VHDL-AMS and a Simulink model.

2 Summary of electrostatic harvester operation

The modeled harvester can be seen as a conjunction of three blocks: the electromechanical resonant transducer including a resonator and a variable capacitor, a charge pump and a flyback circuit. The last two blocks are the parts of the conditioning circuit (Fig. 1).

2.1 Resonant electromechanical transducer

The idea of a vibration energy harvester using an electrostatic transducer is the following; the energy stored in the capacitor is given by:

$$W = \frac{Q^2}{2C}, \tag{1}$$

where Q is the charge, C is the capacitance. If C varies thanks to some mechanical force, it is possible to charge the capacitor when C is high (at the price of some energy W_1), and discharge it when C is low (getting back some energy W_2). From the formula (1) one obtains

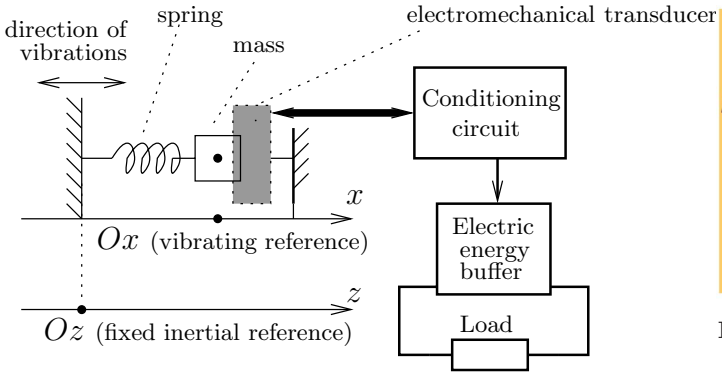

Figure 1 : General structure of the vibration energy harvester.

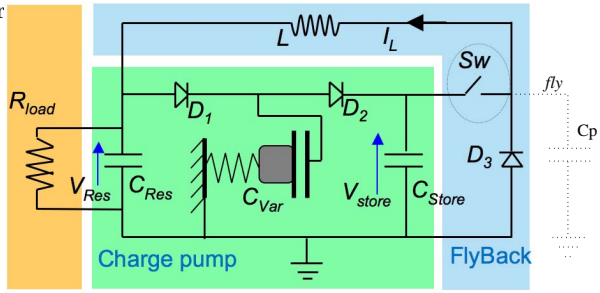

Figure 2 : Modeled conditioning circuit [YEN].

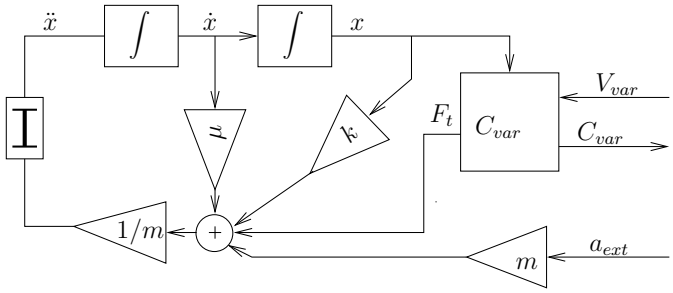

Figure 3 : TDF diagram of the resonator.

that $W_2 > W_1$, and that the difference $W_2 - W_1$ correponds to the energy harvested from the mechanical domain.

Fig. 3 presents the diagram of the resonant transducer block. The mechanical part is modeled by a second-order lumped parameter system, and the capacitive transducer is a parallel-plate capacitor whose mobile electrode can move "in plane".

This system is described by Newton's second law [1]:

$$kx + \mu\dot{x} + F_t(x) + ma_{ext} = m\ddot{x}, \tag{2}$$

where x is the mass displacement from the equilibrum position, m is the mass, k is the stiffness of the spring, μ is the viscous damping constant of the resonator, F_t is the force generated by the capacitive transducer and a_{ext} is the acceleration of the external vibrations.

Transducer force is given by the following formula:

$$F_t = \frac{V^2}{2}\frac{\mathrm{d}C}{\mathrm{d}x}, \tag{3}$$

where V is the voltage applied on the transducer. The gradient of capacitance depends on the transducer geometry and in our case, the capacitance varies linearly with the displacement (the numerical values are valid for the device presented in [6]):

$$C = 75 \cdot 10^{-12} + 1 \cdot 10^{-6}x \ (Farads). \tag{4}$$

2.2 Conditionning circuit operation

The role of the conditioning circuit is to manage the electrical charge flow on the variable capacitor. The circuit proposed in [3] is composed of the charge pump and the flyback circuit (Fig. 2).

The role of the charge pump is to transfer the charges from C_{res} to C_{store}, and since $C_{res} >> C_{store}$, the system will have more electrical energy than before, after the pumping.

The circuit starts the operation from the state where all capacitors are charged at some initial voltage V_0, and the swith is blocked (opened). C_{res}, C_{var} and C_{store} are such that: $C_{res} >> C_{store} >> C_{var}$, and C_{var} is the variable capacitor.

When C_{var} decreases, V_{var} voltage increases, the diode $D2$ becomes on and the charges flow from C_{var} to C_{store}, increasing V_{store}. When C_{var} is minimal and starts to increase, its voltage decreases, $D2$ is off and until $V_{var} = V_{res}$, both the diodes are off. When $V_{var} \leq V_{res}$, $D1$ is on and the charges from C_{res} flow towards C_{var}, discharging C_{res}. This is repeated at the next vibration cycle.

When V_{store} approaches the saturation value [3], the flyback is activated by closing the switch. The natural oscillation period of the LC network is much smaller than the vibration frequency, and very quickly C_{store} discharges on C_{res}. At the moment where $V_{store} = V_{res}$, the switch should open (block), and the system will return to its initial state, where all the capacitors have equal voltages, but slightly higher than those at the start of pumping; the increase in voltage corresponds to the harvested energy.

Switching is driven by the event corresponding to the crossing of threshold values by the voltage V_{store}, and can be modeled by a finite-state automaton with one bit memory register [7].

3 SystemC-AMS modeling of harvester

3.1 Resonator modeling

The equation (2) was modeled as a feedback TDF system, whose scheme is given in Fig. 3.

To define the operation of the resonator, two intput values have to be provided: the voltage of the transducer's capacitor (V_{var}), and the external acceleration (a_{ext}). The output value of this block is the capacitance of the transducer, which is related to the displacement by an algebraic function (block C_{var}).

This block is implemented as a SystemC module (specified as SC_MODULE in the code). The delay at the input of the first integrator is required by the SystemC TDF solver which doesn't tolerate delayless loops.

3.2 Implementation of the conditioning circuit model

The conditioning circuit contains several components whose modeling is tricky. In the following sections we describe it for each block.

3.2.1 Variable capacitor

The SystemC-AMS linear electrical circuit modeling tool provides a model of a variable linear capacitor. This is the most important component for our application, thus we conducted several simple test cases before incorporating it in a complex model. The main question was whether or not the variable capacitor model was charge conservative. For example, if the capacitor voltage is fixed, variations of the capacitance must provoke an electrical current in the voltage source.

The response to this question was positive. Indeed, in SystemC-AMS, when a variable capacitor is added to the circuit, two state variables are created: the capacitor voltage and the charge equal to the product of the capacitance and the voltage.

A variable capacitor is controlled by a signal issued from the TDF domain. Thus, it is natural to connect the signal C_{var} issued from the Fig. 3 model to the input of the variable capacitor.

3.2.2 Diode implementation

The implementation of the diodes is the most challenging part of the modeling task, since they have a very non-linear behaviour. At first, we tried to model the diodes as current sources controlled by their own voltage with a given functional relation between both values. The implemented scheme is given in Fig. 4a. The voltage on the diode is measured in the linear electrical domain and converted to the TDF domain using the predefined SystemC-AMS block *sca_vd2sdf* (Voltage Difference To Signal Data Flow). Then the current is calculated and after a necessary loop delay, the source current is updated.

The 0 Henry inductors are necessary if one wants to encapsulate the diode implementation and allow the higher-level blocks to manage a diode as a two-terminal electrical component. In the SystemC-AMS structure, it is not possible to connect an internal electrical node to an external terminal. Since *np* and *nn* are internal nodes (they implement two electrical connections), a fictive inductor (which behaves as a wire) must be added to allow a connection from one of its terminals to the external terminal. We used inductors instead of zero resistances, since while the conductance matrix is generated, there is a division on the resistor values and a zero resistor yields a division by zero.

Although correct in theory, the presented diode model did not work, mainly because of the very steep exponential I-V diode characteristic in conjunction with the fixed delay between the voltage measurement and the current generation. So, if at step k the voltage becomes, say, 1 V, the corresponding very high current will be generated throughout the whole time step $k+1$,

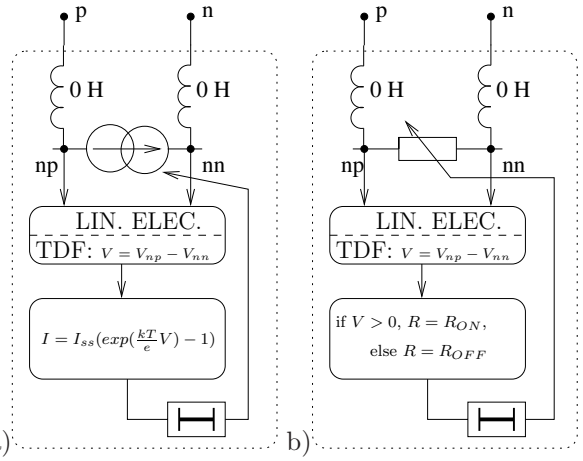

Figure 4 : Different implementations of a diode in SystemC-AMS.

which will definitively cause the modeling process to fail.

For this reason, we proposed a different scheme using a variable resistance (Fig. 4b). In this case, the diode is modeled as a switch with "on" and "off" resistances, and switching is ordered by the voltage on the switch itself. Although the value of the resistance will also be updated with a one time step delay after the voltage change, there is no delay between the current and the voltage calculation, thus the model is accurate enough if the time step is low.

The listing of the diode implementation is given below.

It is also possible to implement a more complex model, taking into account the non-zero diode threshold, adding a voltage source in series with the resistor. In this case, the ON/OFF TDF signal should simultaneously change the resistance value and the source voltage.

```
//TDF block that calculates the state of the diode
SCA_SDF_MODULE(Electrical_diode_function) {
sca_sdf_in<double> voltage;
sca_sdf_out<double> resistance;
void sig_proc()
{
//calculates the resistance using the voltage
}
void attributes()
{
//one delay step for the exit signal
current.set_delay(1);
}
//constructor ...
};
SC_MODULE(Electrical_diode) {
//external connections
sca_elec_port p, n;

Electrical_diode(sc_module_name name_)
                    : sc_module(name_) {
//definition of the 0H inductors
sca_l* in_ind = new sca_l("in_ind",0.);
sca_l* out_ind = new sca_l("out_ind",0.);

//definition of the variable resistor
sca_sdf2r* resistor = new sca_sdf2r("resistor");
```

```
//definition of the converter  LIN ELEC to TDF
//converts the voltage across the resistor
//           (diode) to a TDF signal (double)
sca_vd2sdf* converter =
               new sca_vd2sdf("converter");

//definition of the resistance function
Electrical_diode_function* function =
    new Electrical_diode_function("function");

//definition of internal electrical nodes
sca_elec_node np, nn;

//definition of internal signals
sca_sdf_signal<double> voltage, current;

//connect inductors
in_ind->n(p);
in_ind->p(np);
out_ind->n(nn);
out_ind->p(n);

//connect converter
converter->p(np);
converter->n(nn);
converter->sdf_voltage(voltage);

//connect diode function
function->voltage(voltage);
function->current(current);

//connect the variable resistor
resistor->p(np);
resistor->n(nn);
resistor->ctrl(current);
}
};
```

This model is used as a normal SystemC-AMS component. For example, an insertion of a diode between the nodes p and n is achieved as follows:

```
Electrical_diode diode("diode");
sca_elec_node p, n;
diode.p(p);
diode.n(n);
```

3.2.3 Initial charge of the capacitors

In the modeling of power electronic systems, it is very important to be able to control the initial energy of the reactive elements. In SystemC-AMS, default values are zero, and no mechanism is proposed to modify them.

Thus for each capacitor, we use the scheme shown in Fig. 5. A voltage source was connected during several step periods to the capacitor through a switch (variable resistance). The calculation of the initial pre-charge time is encapsulated in the model. The switch is on during at least the first time step, then, the time constant is calculated (the RC product), and if the latter is high, the switch stays on during the time equal to at least $10RC$.

3.2.4 Conditionning circuit/flyback switch modeling

As shown in [7], the switch should be driven by the energy state of the pump charge. Thus in our model we implemented a switch as a finite-state automaton (with 2 possible states), driven by the events of threshold-crossing by the voltage V_{store}. The switch becomes "on" when it is "off" and when the V_{store} voltage crosses some

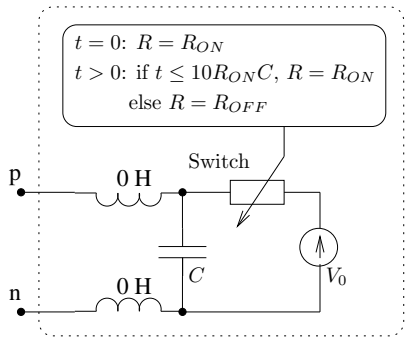

Figure 5 : Implementation of a capacitor with initial pre-charge.

V_1 threshold, and the switch becomes "off" when it is "on" and the V_{store} voltage crosses some V_2 threshold, $V_1 < V_2$.

3.2.5 Modeling of the diode D3 switching

A particular problem arises in the modeling of the circuit behaviour when the the flyback/charge pump switch is turning off. Before this moment, the current in the inductor is maximal. When the switch turns off, the current path is broken, which normally, provokes a high negative voltage on the inductor. Since the voltage of the left node of the inductor is fixed by the C_{res} capacitor, a negative voltage glitch is generated on D3. However, the latter is connected so as to turn on when a negative voltage is generated on the node *fly*. By turning on, the D3 diode allows the inductor current to continue. One can say that the D3 diode "absorbs" the negative voltage glitch.

However, the described phenomenon takes place instantaneously, whereas the model in SystemC-AMS is strictly causal: for the diode D3 to turn on, its voltage should become negative during the preceding step. Since the switching off is abrupt (the off resistance of the switch is high), the generated voltage is very high and being present throughout the step, seriously disturbs the circuit's state vector.

To limit this negative effect, we connected the node *fly* to a small parallel-to-ground capacitor $C_p = 1pF$, whose role is to maintain the voltage on the node *fly* at a reasonable level during the transition step. Adding this capacitor doesn't invalidate the model, since it naturally models the node parasitical capacitance.

When the switch turns off, the current continues to flow through the capacitor C_p, charging it at a negative voltage, and at the next step the diode turns on and the circuit operates normally.

3.3 Model of the complete system

The model of the global system is presented in Fig. 6: it is composed of the conditioning circuit and the resonator models, connected through the input and output values. The delay is necessary since there is a loop, a *sca_v2sdf* block converts the voltage of C_{store} capacitor to the TDF domain.

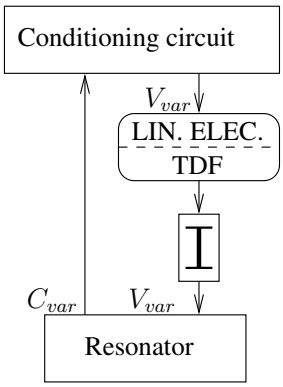

Figure 6 : Diagram of the complete harvester model in the SystemC-AMS language.

Table 1: Numerical values of modeling test case

k, nm^{-1}	m kg	μ, Nsm^{-1}	ω, $\mathrm{rad{\cdot}s}^{-1}$	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	C_p, 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}	1	10^{11}
Switch low threshold, V_1, V		Switch high threshold V_2, V		
6		11		

4 Modeling results

4.1 Description of the modeling experiment

Table 1 gives the numerical values for the circuit modeling.

With a time step of 4 ns, a 1 s simulation requires 250 million steps. Such a small step was chosen to accurately model the quick processes taking place, involving highly non-linear elements (diodes).

In the first version of our model, the capacitance of the variable capacitor C_{var}, calculated by the TDF resonator, was updated in the LIN ELEC domain at each step (every 4 ns). This required a reinitialization of the network matrix at each step, leading to an excessively long simulation time (the linear solver of SystemC-AMS is optimized for modeling time-invariant systems, or systems whose parameters change rarely).

This problem was identified using the GNU profiler (*gprof*). Therefore, we modified the variable capacitor's mixed TDF-LIN ELEC model: in the new version, the LIN ELEC capacitance value is updated only one time every 400 ns. This modification does not affect the accuracy of the model, since the capacitance varies sinosoidally with a frequency of only 298 Hz. This optimisation made the simulation run about 9 times faster.

With this modification, modeling 1 second of system operation required 140 minutes of machine time on Mac OS X 10.4, Intel Core 2 Duo, 2.0 GB, 4 MB Cache, 2 GHz clock computer (only one processor core was used to run the simulation). SystemC 2.2.0 and SystemC-AMS 0.15 RC5 were used.

Table 2: Relative differences in comparison with VHDL-AMS

	V_{store}	V_{res}	I_L
SystemC-AMS	0.495%	1.468%	0.595%
Simulink	0.886%	0.316%	0.100%

Fig. 7 presents the global view of the simulation results, showing the the evolution of V_{store} and V_{res} during 1 s. This is typical behaviour for a charge pump, apparently identical to the results obtained by the VHDL-AMS simulation [7]. The evolution of V_{res} highlights an accumulation of the harvested energy in the system. Fig. 8a presents an enlarged view of the V_{store} and V_{var} voltage evolution at the end of the first cycle, Fig. 8b shows the evolution of the flyback current (in the inductor).

4.2 Modeling results validation

To verify our model we compared the SystemC-AMS model with a VHDL-AMS and a Matlab Simulink model. The former was described in [7], the latter was created using the electrical network equations.

The values of three system quantities were compared: V_{store}, V_{res} and I_L (the quantity defining the energy state of the system). From the results of each model, V_{store} was measured every 10 ms to compare global system operation, V_{res}'s peak value minus 5V (the initial charge) was measured after three cycles of the charge pump/flyback operation to compare only the energy gain, and for I_L the peak value was taken from the first cycle to compare the flyback operation. We also performed a number of other global and detailed tests, in which SystemC-AMS showed simular results as Simulink (under 2% relative difference).

Every model used the same block parameters (resistances, initial voltages,...), but with different diode models. Simulink used a quadratic diode law, whereas VHDL-AMS used a model with 3 zones (linear on and off zones, and a quadratic transition zone to keep the first derivative continious). The VHDL-AMS model did not include the C_p capacitor, as it caused the simulation to slow down too much.

Table 2 presents the relative differences of Simulink and SystemC-AMS modeling versus VHDL-AMS modeling.

These results show that the SystemC-AMS model is correct. The use of various diode models is reflected in the different energy yields (reflected by the values of V_{res} after 4 cycles). Nonetheless, these differences are very small (less than 2%), with Simulink's energy gain being a bit closer to VHDL-AMS's, due to Simulink's diode model being more simular to that of VHDL-AMS.

Simulation times were 3.5 minutes for Matlab Simulink using the *ode23s* solver on a Dual Intel Xeon 3Ghz (4 cores) with 6Gb of memory and 5.75 minutes for VHDL-AMS using the ADVance MS simulator on a Sun Ultra-80.

Compared with SystemC-AMS modeling, Simulink

and VHDL-AMS simulations consume much less machine time. This is explained by the fact, that the corresponding solvers use a variable simulation step, allowing a dramatical time step reduction only during the flyback circuit operation. The SystemC-AMS model uses a fixed step, which is adjusted to accurately model the quickest process of the system.

5 Conclusion

This study presented a complex SystemC-AMS model of a vibration energy harvester with a capacitive transducer. It was demonstrated that complex non-linear electrical circuits coupled with non-electrical domain subsystems can be accurately modeled with SystemC-AMS. The modeling results were compared with VHDL-AMS and Simulink simulation output.

We explored the possibilities of this promising new extension for SystemC. By introducing new reusable components we extended the modeling possibilities of this simulator to non-linear electrical circuits.

The main limitation of the current version of the SystemC-AMS simulator for modeling complex AMS systems is the impossiblity to vary dynamically the simulation step. Also, if a time-variable linear system is modeled with the LIN ELEC solver, the equation system matrix is reinitialized at each variation of the system parameters. This initialization is a very time-consuming operation which, if executed at each time step, can dramatically increase the simulation time.

Nevertheless, SystemC-AMS, being a SystemC extension, offers the unique possibility to model systems from the hardware level (electrical circuit) up to the software level using the same standardised language (C++). As SystemC and SystemC-AMS are both based on C++ and open source, all C/C++ features, tools and libraries are available to the user (eg. the GNU profiler).

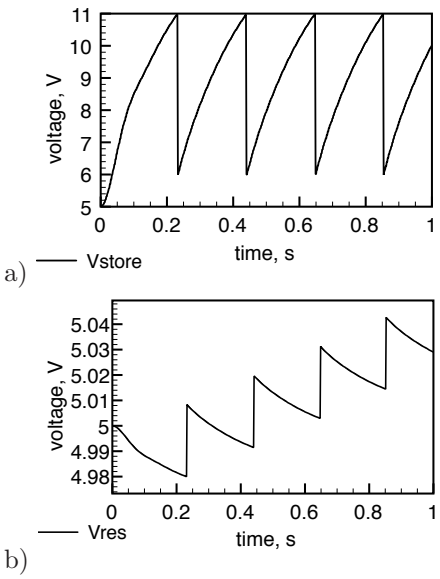

Figure 7 : Global view of the harvesting operation. a) V_{store} voltage, b) V_{res} voltage evolution, highlighting the energy accumulation.

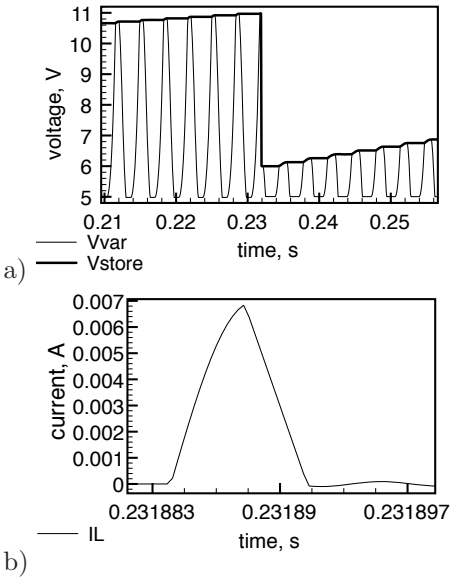

Figure 8 : Zoom on the circuit behaviour: a) V_{var} and V_{store} at the end of the first cycle, b) flyback circuit operation.

References

[1] P. D. Mitcheson *et al.*, "Architectures for vibration-driven micropower generators," *Journal of microelectromechanical systems*, vol. 13, pp. 429–440, june 2004.

[2] G. Despesse *et al.*, "Fabrication and characterization of high damping electrostatic micro devices for vibration energy scavenging," *DTIP of MEMS & MOEMS conference*, 2005.

[3] B. C. Yen *et al.*, "A variable-capacitance vibration-to-electric energy harvester," *IEEE transactions on Circuits and Systems*, vol. 53, february 2006.

[4] A. Vachoux *et al.*, "Extending systemc to support mixed discrete-continuous system modeling and simulation," *ISCAS2005 proceedings*, 2005.

[5] M. Vasilevsky *et al.*, "Modeling heterogeneous systems using SystemC-AMS case study: A wireless sensor network node," *BMAS2007 proceedings*, 2007.

[6] A. M. Paracha *et al.*, "A bulk silicon-based vibration-to-electric energy converter using an in-plane overlap plate (IPOP) mechanism," *PowerMEMS'2006*, 2006.

[7] D. Galayko *et al.*, "AMS modeling of controlled switch for design optimization of capacitive vibration energy harvester," *BMAS'2007*, 2007.