

# Modeling of a Piezoelectric Device with Shocks Management Using VHDL-AMS

Sylvie GUESSAB  
Supélec / Department of Measurement  
Plateau de Moulon  
91192 Gif-sur-Yvette, FRANCE  
+33 (0)1 69 85 14 09  
Sylvie.guessab@supelec.fr

Jean OUDINOT  
Mentor Graphics  
13/15 rue Jeanne Braconnier  
92360 Meudon La Foret, FRANCE  
+33 (0)1 40 94 74 55  
jean\_oudinot@mentorg.com

## ABSTRACT

This paper highlights the facilities of VHDL-AMS to model multi-physical systems and deal with time discontinuities. The modeling and simulation of a piezo-electric device used to spray liquids are presented. Since it is composed of several piezoelectric and metallic parts, a modular approach is proposed. A generic behavioral model for each part type - considered as a mechanical transmission line - is first developed. Then, the whole device is modeled with a structural description. Besides, since there are non-permanent mechanical contacts between some of the parts, a shocks management is added.

## 1. INTRODUCTION

In general, modern signal processing systems include software, electronic hardware and non-electrical components (sensors or actuators). This multi-discipline character combined with the ever-increasing integration complexity, leads to new requirements in design methodologies, and particularly in modeling and simulation capability. Very high accuracy device models are based on partial differential equations that are generally solved by finite-elements tools. Such an approach is not practical because it requires huge computing power and is very time consuming. At a relatively lower level, behavioral models are more efficient for the complete system simulation. They are based on algebraic or ordinary differential equations that can be represented with two approaches: signal-flow block diagrams (using tools such as MATLAB/Simulink) or Kirchhoff networks (or other equivalent networks) with for example SPICE macro-models [1, 2]. The VHDL-AMS language [3, 4] makes it possible to mix both approaches with behavioral as well as structural descriptions. Moreover, since it is adapted to mixed systems, it can deal with the modeling of analog systems with time discontinuities.

This article describes how VHDL-AMS is used to model a piezo-electric device driving a liquid spray. It is composed with several parts, some of which are continuously linked whereas there are non-permanent contacts (that is to say

shocks) between some others. The purpose of the model is to optimize the output flow of the system. In this paper, we will first describe the piezo-electric device and its VHDL-AMS modeling based upon propagation equations in the mechanical transmission lines. A Kirchhoff network approach with a mechanical/electrical analogy is used. Then, we show how the code is completed to model the shocks. We finally give some simulation results. They are compared to those provided by a MATLAB/Simulink model, based on a signal-flow approach.

## 2. MODELING OF THE PIEZO-ELECTRIC ACTUATOR

### 2.1 Device

The piezo-electric device is presented schematically in Fig. 1 (axis of symmetry  $(Ox)$ ). It is composed of an actuator and a needle. The actuator is made of eight joint cylinder parts of which six are piezoelectric ceramics to create movements to the end when applied with a voltage source at a given frequency. The needle, in the middle of the pipe, is linked at one end to a mass. The distance between the end of the actuator and the mass is called the aperture and noted  $x$ . The purpose of the device is to spray a liquid that flows through the central pipe inside the actuator. At rest,  $x$  is zero, which prevents the liquid from flowing.

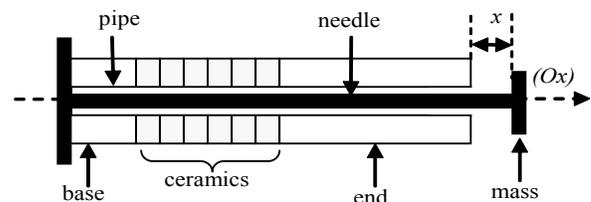


Figure 1 – Piezo-electric device

### 2.2 Methodology

Since the actuator is made of eight parts (metallic and piezoelectric parts), it is natural to adopt a modular modeling approach where following specifications need to be considered: individual part behavior, energy exchange at the interfaces and limit conditions.

To simplify the problem, we assume that all models are one dimension. Moreover, for all parts (except the mass), the elastic deformations are taken into account with propagation times. In this case, we are dealing with two physical quantities, force and speed, at the ends of each part with following conventions:

- The force is considered as applying from left to right. Hence:
  - The force at the left end of a part is considered as a force applied by the part on its left
  - The force at the right end of a part is considered as a force applying to the part on its right.
- The speed is considered as positive if the movement is from left to right.

As we are most familiar with electrical domain, we have adopted a classical analogy between mechanical and electrical quantities:

- Force  $\leftrightarrow$  Voltage (effort quantity)
- Speed  $\leftrightarrow$  Current (flow quantity)

With such analogy, a mechanical system can be represented by a Kirchhoff network (generally used in the electrical domain) with the following equivalent impedances:

- Mass  $\leftrightarrow$  Inductor
- Spring  $\leftrightarrow$  Capacitor
- Fluid friction  $\leftrightarrow$  Resistor

Moreover, each metallic part is comparable to a coaxial cable (both are transmission lines). Indeed, there is an equivalence between the telegraphists' equations used for electrical cables and the continuity equations in a homogeneous medium modeling the metallic parts. Some relations linking electrical and mechanical transmission lines are recalled in (1), (2) and Table 1.

$$C = \frac{1}{E\sigma} \quad (1)$$

$$L = \rho\sigma \quad (2)$$

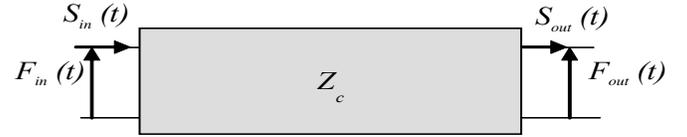
**Table 1 – Electrical/mechanical line equivalences**

	electrical transmission line	mechanical transmission line
characteristic impedance ( $Z_c$ )	$\sqrt{\frac{L}{C}}$	$\sigma \cdot \sqrt{\rho E}$
propagation time ( $\tau$ )	$l\sqrt{LC}$	$l\sqrt{\frac{\rho}{E}}$

where  $l$ ,  $L$  and  $C$  are respectively the length, lineic inductance and capacitance of an electrical cable, whereas  $E$ ,  $\sigma$  and  $\rho$  are respectively the Young's modulus, section and volumic mass of a mechanical line.

### 2.3 Generic models of single parts

We deal directly with the four mechanical quantities associated to each part, which are the force and speed at both ends of the part (Fig 2). We will give below the propagation equations in a metallic or piezoelectric part, as well as their coding in VHDL-AMS.



**Figure 2 – Part representation with an mechanical/electrical analogy**

Regardless of limit conditions and since a metallic part is considered as a transmission line, it is well known [5, 6] that the equations linking the four mechanical quantities are:

$$\begin{cases} F_{in}(t) - G \cdot F_{out}(t - \tau) = Z_c \cdot (S_{in}(t) - G \cdot S_{out}(t - \tau)) \\ G \cdot F_{in}(t - \tau) - F_{out}(t) = Z_c \cdot (-G \cdot S_{in}(t - \tau) + S_{out}(t)) \end{cases} \quad (3)$$

where  $Z_c$  is the characteristic impedance,  $\tau$  is the propagation time,  $G$  is a coefficient (real number between 0 and 1) representing force or speed lost between input and output.

A piezoelectric part behaves just like a metallic part in which a stimuli source  $U(t)$  has been added. The piezoelectric effect  $D(t)$  is generated when the stimuli  $U(t)$  has been applied to the part. The term  $D(t)$  depends on  $U(t)$  as well as the part deformation through following equation (4):

$$D(t) = \frac{\varepsilon s - d^2}{l s} \cdot U(t) + \frac{d}{l s} \cdot \int (S_{output}(t) - S_{input}(t)) dt \quad (4)$$

where  $\varepsilon$  is the electric permittivity (m/V),  $s$  is the elasticity (1/Pa),  $d$  is the charge constant and  $l$  is the part length (m). The elasticity  $s$  depends on the material in use. For a metallic part, it corresponds to the reverse of Young's modulus and in the case of the piezoelectric it is given by (5):

$$s = \frac{d^2}{\varepsilon} + \frac{1}{\rho c^2} \quad (5)$$

where  $c$  is the ceramic propagation speed. The ceramic piezoelectric part equations, when applied by the stimuli  $U(t)$ , then become:

$$\begin{cases} F_{in}(t) - G \cdot F_{out}(t - \tau) = Z_c \cdot (S_{in}(t) - G \cdot S_{out}(t - \tau)) + \frac{d\sigma}{\varepsilon s - d^2} (D(t) - D(t - \tau)) \\ G \cdot F_{in}(t - \tau) - F_{out}(t) = Z_c \cdot (-G \cdot S_{in}(t - \tau) + S_{out}(t)) + \frac{d\sigma}{\varepsilon s - d^2} (D(t - \tau) - D(t)) \end{cases} \quad (6)$$

The previous equations have been coded in VHDL-AMS. The listings are shown below. The entity ports are declared as terminals since we have chosen to model the device with a Kirchhoff network. Note that the code for a metallic part can be re-used for any transmission line whatever the physical domain due to the effort/flow analogies.

**Listing 1. Metallic part model**

```

entity transmission_line is
    generic (Sig, Rc, T, Am : real :=0.0);
    port (terminal E1, E2, S1, S2 : electrical);
end entity transmission_line;

architecture propagation of transmission_line is
    constant Zc: real := Sig*Rc;
    quantity Vin across E1 to E2;
    quantity lin through E1 to E2;
    quantity Vout across S1 to S2;
    quantity lout through S2 to S1;
begin
    Vin==Am*Zc*lout'delayed(T)+Zc*lin+Am*Vout'delayed(T);
    Vout==Am*Zc*lin'delayed(T)-Zc*lout+Am*Vin'delayed(T);
end architecture propagation;

```

**Listing 2. Piezo-electric part model**

```

entity piezo is
    generic (Rc, T, Sig, l, d, g, s, Am : real );
    port (terminal E1, E2, S1, S2, V1, V2 : electrical);
end entity piezo;

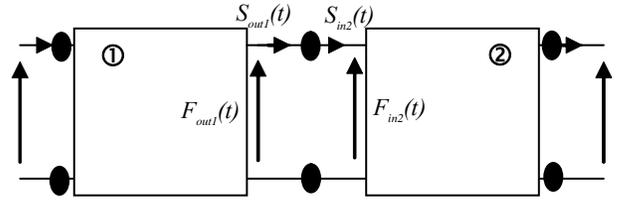
architecture propagation of piezo is
    constant eps : real := d/g;
    constant Zc : real := Sig*Rc;
    quantity Vin across lin through E1 to E2;
    quantity Vout across S1 to S2;
    quantity lout through S2 to S1;
    quantity V across V1 to V2: real;
    quantity D0 : real;
begin
    Vin==Am*Zc*lout'delayed(T)+Zc*lin+Am*Vout'delayed(T)-d*Sig/(eps*s-d*d)*(D0'delayed(T)-D0);
    Vout==Am*Zc*lin'delayed(T)-Zc*lout+Am*Vin'delayed(T)-d*Sig/(eps*s-d*d)*(D0'delayed(T)-D0);
    V==l*s/(eps*s-d*d)*D0-d/(eps*s-d*d)*(lout'integ - lin'integ);
end architecture propagation;

```

**2.4 Interfaces and limit conditions modeling**

The connection of two parts (Figure 3) is modeled by a simple continuity condition of the force and the speed at the interface:

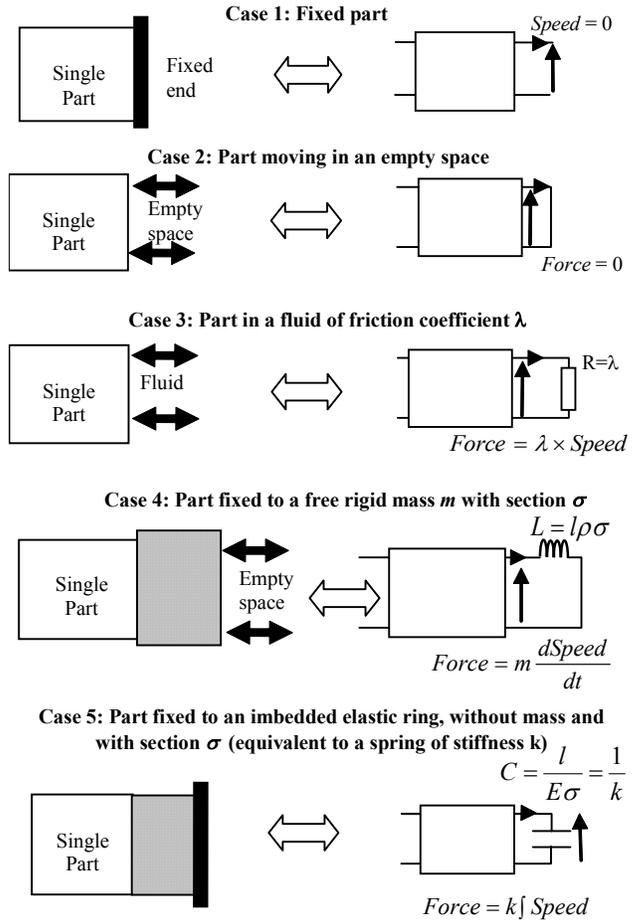
$$\begin{cases} F_{out1}(t) = F_{in2}(t) \\ S_{out1}(t) = S_{in2}(t) \end{cases} \quad (7)$$



**Figure 3 – Interface between two parts**

There is no need to code these equations. Indeed, in a Kirchhoff network, they will be taken into account when connecting terminals of parts 1 and 2. Thus, the actuator model is structural and built upon an association of metallic and piezo-electric parts. It is implemented with a VHDL-AMS structural description.

As far as limits condition are concerned, Figure 4 illustrates some typical cases and recalls the relation that links the force and speed at the concerned limit for a part of length  $l$ .



**Figure 4 – Limit conditions modeling**

### 3. SHOCKS MANAGEMENT

#### 3.1 Actuator/mass shocks

Figure 5 shows the whole device structure. The needle can be represented by a generic model of a metallic part. The needle/mass contact is permanent whereas it is not between the mass and the end of the actuator. Thus, the limit conditions for the mass as well as the shocks management between the mass and the end of the actuator have to be added to the actuator model.

The two possible states of the device are “open” or “closed”:

- If the device is open, the end of the actuator is free (the force applied by the actuator to the front mass is zero) and the needle is fixed to the front mass.
- If the device is closed, the right end of the actuator and the needle are fixed to the mass. The aperture is zero.

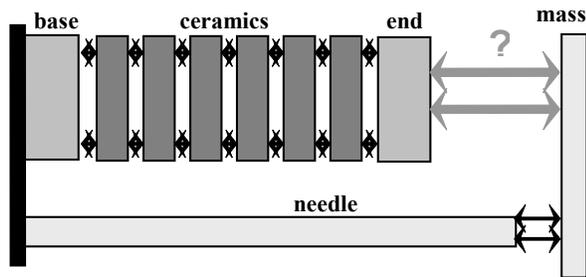


Figure 5 – Non-permanent contact between the actuator and the mass

The two quantities that can change the state are the aperture and the force applied to the mass by the actuator (Figure 6).

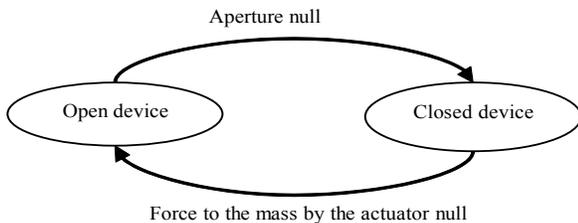


Figure 6 – State changes

#### 3.2 VHDL-AMS coding

The VHDL-AMS code for the shocks management is given below (Listing 3). A binary signal (1 and 0) is being used to represent the state of either the device is open or closed. Such state signal should condition the branching where we will introduce following quantities  $V_{s9}$ ,  $V_{e10}$ ,  $V_{s11}$ ,  $I_{s9}$ ,  $I_{e10}$ ,  $I_{s11}$  (Figure 7).

Besides, we define two quantities which cause the state switching:

- the aperture :

$$aperture = \int (I_{e10} - I_{s9}) dt \quad (8)$$

- the force applied to the mass by the actuator

$$force\_contact = V_{e10} - V_{s11} \quad (9)$$

Finally, a process is activated to change the state, when *aperture* or *force\_contact* becomes zero.

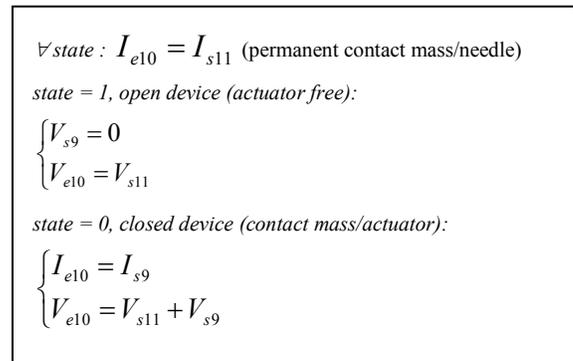
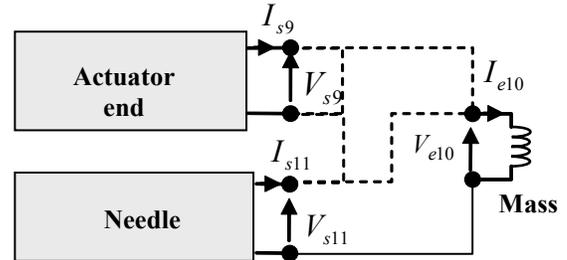


Figure 7 – Conditional branching (----)

#### Listing 3. Shocks management

```

aperture == le10'integ - ls9'integ ;
force_contact == Ve10-Vs11 ;
aperture_s <= '1' when aperture'above(0.0) else '0';
force_contact_s <= '1' when force_contact'above(0.0) else '0';
le10 == ls11 ;
if etat = '1' use
    Vs9 == libre*ls9 ;
    force_contact == 0.0;
else
    ls9 == le10 ;
    Vs9 == force_contact ;
end use;
process
begin
    if etat = '0' then
        if (force_contact_s'event) then etat <= '1' ;
        else etat <= etat ;
        end if ;
    else
        if (aperture_s'event) and (not aperture'above(0.0)) then etat <= '0' ; else etat <= etat ;
        end if ;
        end if ;
        wait for 0.01us ;
        wait on force_contact_s, aperture_s ;
    end process ;

```

## 4. SIMULATION RESULTS

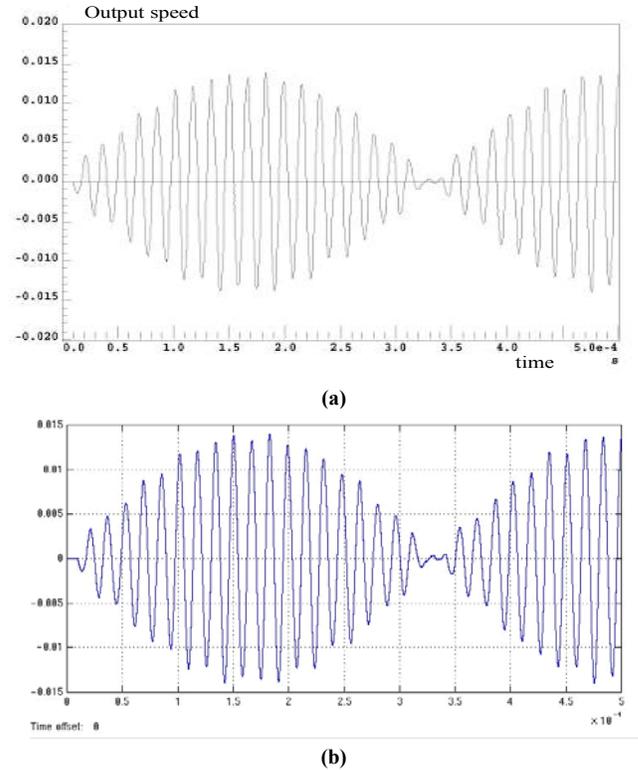
The VHDL-AMS model of the piezo-electric device was simulated using ADVance MS 1.3 from Mentor Graphics. In order to validate the model, the physical coherence of the results was checked for different limit conditions and they were compared to those provided by another model. The latter is based upon a signal-flow approach and is implemented with MATLAB 6.0/Simulink 4.0. Though it is not presented in detail in this paper, what is to highlight is that this model is far less straightforward than the VHDL-AMS one. Indeed, the signal-flow approach allows only one-directional quantities (whereas there are bi-directional in Kirchhoff network). Therefore, each quantity in the previous model has been written as the superposition of two one-directional signals, which makes the model far more complicated, especially for a piezo-electric part and the shock management.

Whatever the test conditions, the simulation results provided by both models proved to be similar, as it is shown in Figures 8 and 9. In both case, a sinusoidal stimulus is applied to every piezo-electric part with a phase shift of  $\pi$  rad between two neighboring parts. Figure 8 corresponds to the simulation of the actuator without the needle and the mass (that is to say without shocks management). The actuator is free, hence, the forces at the two ends, that are not shown here, are null. Figure 9 gives the aperture (distance between the mass and the end of the actuator) when simulating the whole device.

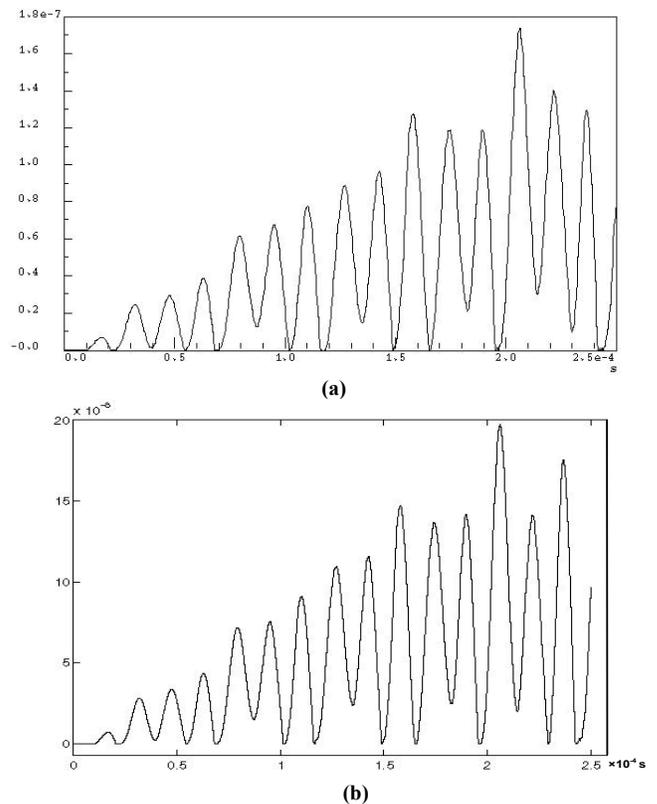
Moreover, when simulating both models with the same test conditions, we got some diverging waveforms with Simulink that are not predicted by the theory. Such instability that might be due the discontinuities did not occur with the VHDL-AMS model.

## 5. CONCLUSION

A piezoelectric device with time discontinuities was modeled with VHDL-AMS. Since it is an assembly of elementary mechanical pieces, the model is built by connecting elementary entities with a Kirchhoff network description. Each part is considered as a transmission line, with an additive stimulus term for the ceramics. The model is finally completed to manage the shocks at one end of the actuator. It has been validated by simulation results and compared to another model implemented in Simulink. Though mathematically equivalent and providing the same results, the latter is far more complicated. This example clearly demonstrates how much VHDL-AMS, which allows to mix structural and behavioral descriptions, makes it easier to model systems involving transmission lines and time discontinuities.



**Figure 8 - Output speed of the free actuator**  
(a) VHDL-AMS model (b) SIMULINK model



**Figure 9 - Aperture**  
(a) VHDL-AMS model (b) SIMULINK model

## 6. REFERENCES

- [1] J. Haase, P. Schwarz. *Behavioral Modeling of Complex Heterogeneous Microsystems*, Proceedings of 1<sup>st</sup> International Forum on Design Languages (FDL'98), Lausanne, 1998.
- [2] P. Schwarz. *Microsystem CAD: From FEM to System Simulation*, Proc. Intern. Conf. "Simulation of Semiconductor Processes and Devices" (SISPAD98), Leuven, 1998.
- [3] IEEE standard 1076.1 VHDL-AMS Language Reference (<http://www.eda.org/>)
- [4] Peter J. Ashenden, Gregory D. Peterson, Darrell A. Teegarden, *The System Designer's Guide to VHDL-AMS*, [Morgan Kaufmann Publishers](#), 2002.
- [5] David K. Cheng, *Field and Wave Electromagnetic*, Addison-Wesley, 1989.
- [6] Giovanni Miano, Antonio Maffucci, *Transmission Lines and Lumped Circuits*, Elsevier, 2001.