Multi-domain Modeling and Simulation of a Linear Actuation System

Deepika Devarajan, Scott Stanton, Birgit Knorr

Ansoft Corporation
Pittsburgh, PA, USA

Abstract

In this paper, VHDL-AMS is used for the behavioral modeling of a linear actuation system that contains subsystems and components belonging to different physical domains, such as the solenoid (electromechanical), the valve (hydraulic), and the driver circuit (electrical). The paper describes the modeling methodologies applied to each subsystem and focuses on additional modeling approaches for capturing the non-linear behavior of the solenoid. Some extensions to the VHDL-AMS language for modeling non-linear effects through table lookup are suggested.

Introduction

The IEEE standardization of VHDL-AMS (1) has encouraged the automotive industry to take major steps in adopting it as a common modeling language. Previously, the industry was tied to proprietary languages which in turn implied dependence on simulation tool vendors. The introduction of VHDL-AMS now offers the automotive industry a choice between simulation tools from different vendors and promises to solve the problem of model delivery from suppliers to OEMs. While proprietary languages allowed only a small community to contribute models, VHDL-AMS enables a number of simulation tool providers to support the language and opens the market for more competition. Additionally, a large community of model developers will be able to provide public domain models.

VHDL-AMS is ideal for modeling complex multi-domain applications. It encourages a modular approach by dividing the entire system into sub-systems. With the definition of *natures*, users can easily develop models for mechanical, electrical, hydraulic, thermal, and other domains. Linear actuation systems typically include electrical, mechanical, and hydraulic components as well as control systems. The solenoid plays the role of an interfacing component between the electrical domain and mechanical loads. It is very challenging to model the solenoid accurately because its operation is based on complex electromagnetic principles. It is a multi-domain component in itself comprising the electrical driver circuit and the mechanical plunger behavior. We recommend a stepwise approach towards modeling the solenoid, with each step serving to improve the accuracy and refine the model. As a first step, a mathematical model for the component is developed representing its basic functionality. This model can be easily generated using differential algebraic equations in VHDL-AMS. One of the main parameters used in such a math-based model is the inductance of the solenoid winding. In the first modeling step, this inductance is assumed to be linear and is derived from analytical calculations. VHDL-AMS is ideal for this modeling stage since it allows an easy and fast model implementation.

However, the actual inductance of the solenoid is typically determined by a variety of physical effects that cannot be easily included in analytical calculations. Most of these effects are non-linear and are primarily due to the magnetic saturation of the iron within the solenoid. The saturation characteristics are non-linear both temporally as well as spatially and they also depend on the position of the plunger. Due to the dependency of the inductance on a variety of geometric and material parameters, a better prediction of the real value of the inductance can be obtained using a Finite Element Analysis (FEA) based tool. The FEA results deliver a more accurate constant lumped parameter value for the inductance. In the second modeling method, this inductance value is used to parameterize the behavioral math-based VHDL-AMS model.

In reality, the inductance value varies with the position of the plunger and the coil current. This cannot be represented by a constant lumped value of the inductance and further model refinement is required. In an ideal case, the FEA system of equations would be solved together with the system equations described in VHDL-AMS. However, this is not practically feasible due to the large number of equations generated by the FEA method. Therefore, a model extraction technology may be used to refine the model behavior. The FEA solver can be used to generate a lookup table that includes the non-linear effects of the solenoid by performing a parametric sweep. This lookup table can then be used to more accurately simulate the solenoid behavior in a multi-domain system.

While VHDL-AMS has several high-level constructs that aid in behavioral modeling, it lacks standardized table elements and interpolation functions. In our example, the table and interpolation functions for such an FEA-based

solenoid are characterized in SML (SIMPLORER Modeling Language). We suggest an extension to the VHDL-AMS language that allows the easy and standardized inclusion of tables and interpolation functions for modeling non-linear effects. This extension could then be implemented as efficient simulator-specific routines without jeopardizing model exchangeability.

Multi-domain System Design

The linear actuation system described below integrates modeling and simulation using digital control, electrical, mechanical, and hydraulic subsystems. The system employs the mechanical motion of a solenoid to control hydraulic flow. A digital subsystem controls the excitation of the solenoid's electromagnet by switching a pair of transistors on the electrical side. The force from the solenoid displaces a plunger in the mechanical domain. This translational motion of the plunger serves to block and unblock the flow of a fluid in the hydraulic domain. The models of the various subsystems in this multi-domain system have been developed using VHDL-AMS components (see Figure 1) and are described in detail below.
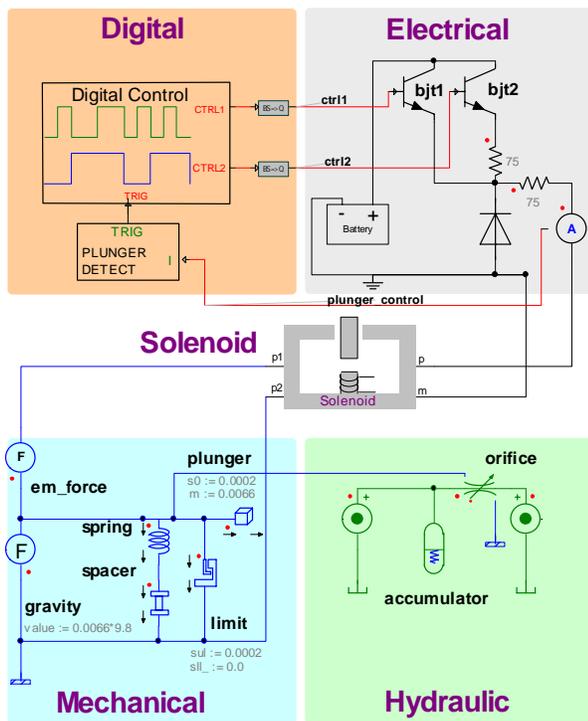
Figure 1: Linear Actuation System

A. Electrical Subsystem

A simple dynamic battery model is used to power two bipolar junction transistors. The transistors have been modeled at the system level and operate as on-off switches based on their input control signals. The transistors along with the associated resistor behave like a voltage divider and provide the voltage control for the solenoid's electromagnet. A free-wheeling diode is provided between the solenoid's electrical terminals to ensure a gradual decay of the solenoid's coil current when the plunger opens to unblock the fluid flow. An ammeter is used to measure and feedback the solenoid's coil current to the digital control subsystem.

B. Digital Control Subsystem

The digital subsystem consists of two parts – a pure digital model called "digital control" developed in VHDL and a mixed signal VHDL-AMS model called "plunger detect" that identifies the closing of the plunger. The control signals for the two transistors are called ctrl1 and ctrl2 and the control stimuli are modeled using behavioral PROCESS constructs in VHDL. Since VHDL-AMS is a strongly typed language, special interface models called omnicasters are used to connect the digital control signals to analog input of the transistor switches. Omnicaster models are used to interface ports of different data types in analog and digital domains. In this example, the omnicasters function like one-bit D/A converters that transform the digital BIT data to an analog quantity based on a threshold value.

C. Mechanical Subsystem

The plunger is modeled in the mechanical domain as a mass model with a single conservative node (position is the *across* quantity and force is the *through* quantity). A pre-loaded spring is represented by a spacer-spring combination and the maximum displacement of the plunger is limited with a limit stop model. When the solenoid is energized initially, the plunger closes the gap beginning with an initial position of 0.0002m to a final position of 0m as specified in the limit stop model. A force source is used to model the force of gravity on the plunger with a force value set to 9.8 m/s^2 times the plunger mass in kg. The force at different points in the mechanical subsystem may be measured using force sensor models.

D. Hydraulic Subsystem

A simplified hydraulic system is modeled using an accumulator, a hydro-mechanical variable orifice model, and a pair of pressure sources that provide a differential pressure to enable hydraulic flow. The plunger displacement in the mechanical domain is used by the variable orifice model to either block or unblock the hydraulic flow due to the differential pressure.

Modeling Approaches for the Solenoid

Various methods may be employed to model solenoids for simulation depending on the required level of detail, accuracy and simulation run-times. An abstract behavioral

model of a solenoid for simplified system design can be developed by using equations that govern its behavior. The parameters for the model may be derived either through hand calculations or by extracting a single lumped value from a FEA tool. Another approach is to develop a detailed model based on parametric sweeps that accounts for the geometry and excitation of the model. This method is employed in FEA tools to generate piecewise linear lookup tables from which the exact non-linear solenoid behavior may be captured. Each of these methods has its advantages and disadvantages in terms of accuracy, simulation speed etc. In this section, we develop a math-based VHDL-AMS model for the solenoid, and determine its parameters through hand calculations as well as FEA analysis.

A. Behavioral Modeling using VHDL-AMS

The solenoid model is developed in VHDL-AMS using idealized equations (2). The equations determine the flux linkage in the solenoid and the corresponding force on the plunger, based on the input current value and the position of the plunger. The VHDL-AMS model of the solenoid's interface (entity) and behavior (architecture) are developed below.

The solenoid model has four conservative pins: two electrical pins (with voltage as the *across* quantity and current as the *through* quantity) and two mechanical pins (with position as the *across* quantity and force as the *through* quantity). The model accepts three parameters: the maximum inductance value per turn at minimum air gap (L_0), an inductance coefficient (K), and the number of coil turns (N). The entity description of the model is as follows:

```
LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
USE IEEE.MECHANICAL_SYSTEMS.ALL;

ENTITY solenoid IS
  GENERIC (
    L0: REAL := 4.05e-7;
    K: REAL := 6328.0;
    N: REAL := 1500.0);
  PORT (
    TERMINAL p,m: ELECTRICAL;
    TERMINAL pos1,pos2: TRANSLATIONAL);
END ENTITY solenoid;
```

The model behavior based on the following equations is described in the VHDL-AMS architecture.

$$L_{\max} = N^2 * L_0$$
$$L(x) = \frac{L_{\max}}{1 + Kx}$$
$$F_k = \frac{K}{2 * L\max}$$
$$\phi = L(x) * i$$

$$v = \frac{\partial \phi}{\partial t}$$
$$F = \phi^2 * F_k$$

where:

N = Number of coil turns
L_0 = Max inductance value/turn at min air gap, H
K = Inductance coefficient
x = Plunger position, m
$L(x)$ = Instantaneous inductance value at x, H
$Lmax$ = Max inductance for all turns, H
Fk = Force at max inductance for given K, N
ϕ = Instantaneous flux through solenoid, Wb
i = Current through solenoid, A
v = Voltage across solenoid, V
F = Force output from solenoid, N

From this set of equations, the architecture for the model may be described as follows:

```
ARCHITECTURE behav OF solenoid IS
  CONSTANT Lmax : REAL := L0 * N * N;
  CONSTANT Fk : REAL := K / (2.0 * Lmax);
  QUANTITY v ACROSS i THROUGH p TO m;
  QUANTITY position ACROSS force THROUGH pos1 TO
pos2;
  QUANTITY L, flux : REAL;
BEGIN
  IF (position > 0.0) USE
    L == Lmax / (1.0 + K * position);
  ELSE
    L == Lmax;
  END USE;
  flux == L * i;
  v == flux'dot;
  force == flux * flux * Fk;
END ARCHITECTURE behav;
```

The value of L_0 can be calculated by hand using the classical equation for solenoid inductance (3) as follows:

$$L_0 = \frac{\mu N^2 A}{l}$$

where:

$\mu = \mu_0 \mu_r$ = Permeability of medium, H/m
N = Number of turns of solenoid
A = Cross-sectional area of solenoid, m^2
l = Length of solenoid, m

The parameters used for calculating the inductance L_0 are
$\mu_0 = 4\pi*10e^{-7}$H/m
$\mu_r = 800$
$N = 1$ turn
$A = 1.2*10^{-5}m^2$
$l = 1.5*10^{-2}$m
and this yields a value of L_0 as 0.8μH.

This value of inductance is easy to calculate, although it is

not always accurate. One of the approximations made in this method is that the value of the permeability is constant while typically permeability varies non-linearly (see Figure 2). Additionally, this calculation assumes that the solenoid is very long and end-effects (fringing effects) can be neglected, which is not true in most cases.
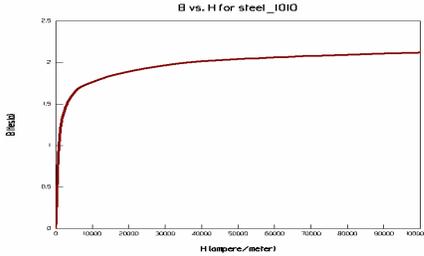
Figure 2: Non-linear Permeability Curve

B. Extracting a Single Lumped Parameter Value using FEA
In order to account for the non-linearities of the device as well as fringing effects, it is more accurate to use finite element methods to calculate inductance. FEA allows the exact geometry of the structure to be used, and includes the exact non-linear material definition (see Figure 3).

A static magnetic field problem may be solved using the following equations:

$$(a) B = \nabla X A$$

$$(b) H = \frac{1}{\mu} B$$

$$(c) W = \frac{1}{2} \int_v B \bullet H dv$$

where:

B = Magnetic flux density, Tesla
H = Magnetic field intensity, A/m
W = Apparent stored field energy, J

Taking into consideration that current flowing in a single coil has created the magnetic field, the inductance of that coil (L) may be determined using equation (c) above:

$$(d) L = 2 \frac{W}{i^2} = \int_v B \bullet H dv \quad \text{(for i=1 Ampere)}$$

The inductance of the coil is LN^2 and the permeability is linear. When the current is greater than 1 ampere and the permeability is non-linear, two steps are needed to obtain the inductance:

1. A non-linear magnetostatic solution is generated with the coil current at values specified by the designer. This establishes a value of μ that varies with each mesh triangle, since the degree of saturation varies throughout the device.

2. These self-consistent μ values are used in a linear solution of equation (d) for the inductance calculation, with the coil current set to one ampere.

Using the finite element method, the value calculated for L0 is 0.405μH. This value of inductance may be used in the VHDL-AMS model to improve the accuracy, but this still assumes that the total inductance of the system behaves linearly as a function of stroke and coil current. In reality, this is not the case; the inductance will vary as a function of stroke and coil current. To include this effect, a parametric finite analysis can be employed.
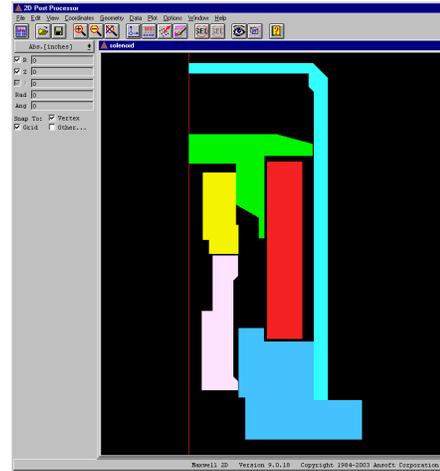
Figure 3: Cross section of the Solenoid

B. Creating State Space Type model with Parametric FEA
A parametric FEA is a series of magnetostatic solutions that varies the values of coil current and position (stroke) through the entire operating range of the device. At each value of coil current and position, the magnetostatic non-linear solver calculates the value of coil inductance and plunger force (see Figure 4).
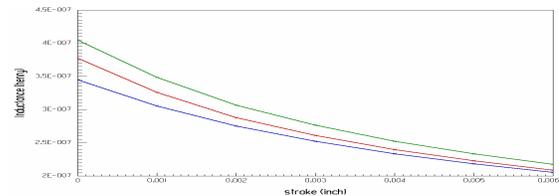
Figure 4: Per turn Inductance vs. Stroke for Various Coil Currents

These values are assembled together to create a lookup table model. This model has two inputs, coil current and stroke, and two outputs, force and inductance (see Figure 5). The table lookup functionality involves complex interpolation routines and is implemented in SML for fast and efficient simulation.
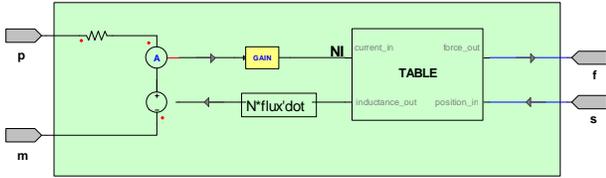
Figure 5: Table lookup Model

Simulation and Interpretation of Characteristics

Two simulations of the multi-domain linear actuation system are performed to obtain the solenoid current characteristics. First, the solenoid model developed in VHDL-AMS with the inductance parameter derived from FEA is used in the multi-domain system. The second simulation is performed using the lookup table model provided by the FEA tool.

The system dynamics due to the opening and closing of the plunger are observed by carrying out a transient simulation for 80ms. Both control signals are reset for the first 10µs of simulation time and then ctrl1 is set to 1. This causes electromagnetic excitation of the solenoid and as soon as a trigger from the plunger detect model is received indicating that the plunger has closed at 10.5ms, ctrl2 was set to 1 and ctrl1 was reset to 0 (see Figure 6). This ensures that the minimum current required to keep the plunger closed is provided. There is a higher voltage drop when the second transistor is switched on due to a larger resistance in the circuit. The initial closing of the plunger causes a back emf that is reflected as a transient dip in the solenoid coil current (see Figure 7). The force exerted on the plunger displaces it from its initial position of 0.0002m to 0m and the plunger is held in this position until about 50ms after its closing. The hydraulic flow through the variable orifice is blocked during the period that the plunger is closed and is continued again when the plunger opens. After the second transistor is switched off, the plunger opens and moves back to its original position of 0.0002m due to the force of the spring, and a transient increase in the solenoid coil current may be noticed due to the back emf.

The solenoid current characteristics from the lookup table are found to be more accurate than that from the lumped inductance parameter (see Figure 8) while the simulation times for both solenoid models were almost identical. One of the critical parameters in solenoid design is the "pull-in time" that defines the time taken for the plunger to close. According to the simulation results, the linear solenoid model takes 12% longer to close when compared with the non-linear table lookup model. A second consideration for solenoid design is the time at which the solenoid current dips and then begins to increase. Again, there is a 12% deviation in the characteristics of the linear solenoid model when compared with the non-linear table lookup model.

Since these are important design characteristics, accurate modeling of solenoid behavior is essential in its applications and a 12% difference could detcide if the design is within specifications.
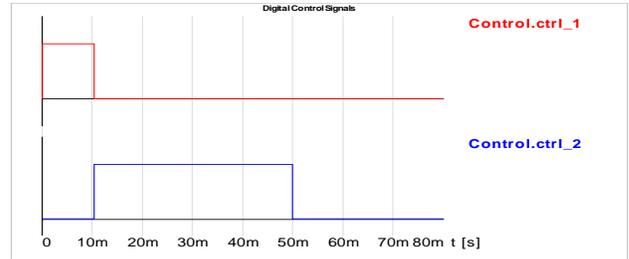
Figure 6: Control Signal Characteristics
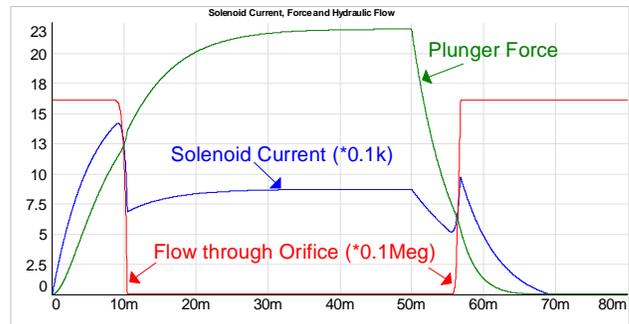
Figure 7: Solenoid Current, Force, and Hydraulic Flow
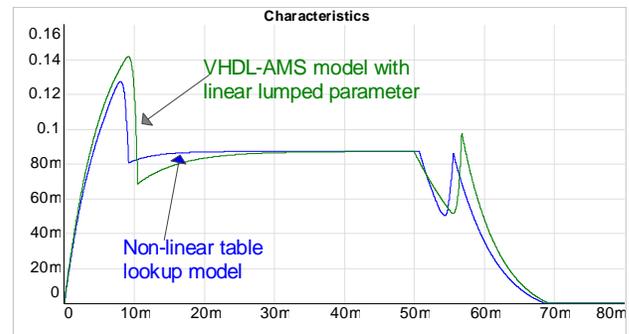
Figure 8: Comparison of Solenoid Current Characteristics

Suggestions for Language Extension

The simulation results indicate that a higher level of accuracy can be obtained from using a table lookup method that incorporates non-linearities and fringing effects on the model characteristics. The state space type parametric model can be easily exported in a standardized format from the FEA tool. It is possible to develop a VHDL-AMS model that can read the data from the exported table model. In order to be able to access a table model using VHDL-AMS, a separate package with user-defined table datatypes and lookup functions need to be defined. While this is possible in the standardized VHDL-AMS format, it is not optimized for individual simulators and will result in large simulation times. If the table lookup functionality were provided using

simulator-specific functions, a significant improvement in speed can be realized at the expense of not being able to exchange models across simulators. We suggest that the VHDL-AMS language is extended so that it would not only maintain model exchangeability across simulation tools but also be optimized for individual simulator cores. If a table element along with VHDL-AMS subprogram declarations for interpolation and extrapolation were standardized, then the functions may be implemented internally in the simulator. This would allow a fast and standardized method for modeling non-linear behavior.

Conclusions

VHDL-AMS is a powerful modeling language and is well suited for multi-domain mixed-signal automotive applications. A linear actuation system involving electrical, mechanical, and hydraulic domains was developed and simulation results were explained.

Several approaches for detailed modeling of the solenoid used in the linear actuation system were outlined. In order to accurately characterize the solenoid's non-linear behavior, a table lookup method may be used. The solenoid model based on table-lookup was simulated using SIMPLORER Modeling Language. Suggestions to extend the VHDL-AMS language was made to support the table lookup functionality.

References

1. IEEE Std. 1076.1 VHDL–AMS Language Reference Manual
2. H. Woodson and J. Melcher, Electromechanical Dynamics Part 1: Discrete Systems, 1968, John Wiley & Sons, pp. 84-86
3. J. D. Kraus, Electromagnetics, 4th ed., 1991, McGraw Hill, pg 235