

Quantitative systems dynamics: Comparison of modeling techniques for the simulation of electro-mechanical systems

Pablo Viejo Garcia [†]

Jose Maria Gonzalez de Durana [‡]

Oscar Barambones [‡]

Enrique Kremers [†]

[†] EIFER, European Institute for Energy Research (KIT-EDF), Karlsruhe, Germany

[‡] University College of Engineering of Vitoria-Gasteiz (UPV/EHU), Spain

Abstract

This paper presents some comparative examples of the use of system dynamics (SD) for the modeling of electro-mechanical systems. The authors argue that many simulation models coming from sciences can be easily translated to SD, with a large number of advantages. The work has been developed in a multidisciplinary environment, where a lack of knowledge transfer between practitioners of these different disciplines is appreciable. In everyday practice, a clear methodology does not exist to evolve from a classical engineering to a system dynamics approach, from mathematical thinking to SD thinking. As engineers are fixed to quantitative results to specific problems, they need strictly quantitative models. This uses to be a critical point in SD where there is a large amount of qualitative modeling and quantitative modeling with *soft* variables. Through the comparison of the same problems solved with different modeling techniques, it is possible to show the advantages and disadvantages of each of them, and improve to a better understanding of both approaches.

Key words: quantitative model, electro-mechanical systems, hybrid systems, electrical circuits, system thinking.

1 Introduction

The development of computing capabilities and software has enabled in the last years the use of computers for the resolution of complex problems on electro-mechanical systems. The traditional education of engineering schools to solve this kind of problems is the use of mathematical equations under the name of Dynamic Systems (DS). These systems are precise and usually deterministic, purely quantitative, according to the needs of engineering sciences. The objective of these models is to be able to give a concrete answer to a problem, far from planning or strategic purposes.

After the 1950s, the development of the seminal work of Forrester evolved to a new paradigm of modeling techniques called System Dynamics (SD). These systems are based on stock and flows and are more visual and intuitive. Even, while during the first decades of SD development, the main purpose was mainly quantitative, many critics emerged from different disciplines related to a few number of models.

After the 1980s, some modellers derived to the development of pure quantitative models, as explained by Coyle [6]. Other papers have dealt with this topic ([7], [8], [9], [10]; [11]; [12]; [13]; [15], [16], [17]).

From this moment on, models developed with SD have included pure quantitative modes, pure qualitative, and a mixture of both. However, when approaching different scientific domains outside from engineering, like economics and social sciences, there is a profusion of soft variables, that delimitate the engineers work [14]. These approaches are not the aim of the engineer's work, and that is why in this paper the authors do not try to choose one of the approaches (quantitative or qualitative). The aim is rather to compare between quantitative approaches from different disciplines. In the following, a short theoretical overview of solving methods for differential equations will be given, as the examples shown later are concerned with these type of problems.

1.1 Solving methods for differential equations

Differential equations are an important conceptual part not only of automation or control systems but also in many other areas of science and engineering, each of which has developed its own methods to find solutions efficiently. The wheel-and-disc integrator invented by James Thomson [1], brother of Lord Kelvin, was the first device that allowed for (mechanically) the operations of *analog computation*. Using the *integrator* as basic element, the two brothers built a device to calculate the integral of the product of two given functions. Kelvin designed other machines capable of integrating differential equations of any order, but they were never built.

To find the solution of a given explicit ordinary differential equation,

$$\frac{d^n y}{dt^n} = f \left(\frac{d^{n-1} y}{dt^{n-1}}, \dots, \frac{dy}{dt}, y, u, t \right),$$

together with the initial values, the idea of Lord Kelvin was to integrate with his device n times $\frac{d^n y}{dt^n}$, thus obtaining the values $\frac{d^{n-1} y}{dt^{n-1}}, \dots, \frac{dy}{dt}$, and carry out with them the necessary (mechanic) arithmetic operations to obtain $f(\cdot)$, and then close the loop [2].

In 1950s came the *analog computer*, equipped with electronic integrators made by electron valves and based on the same ideas of Kelvin. This device allowed for obtaining solutions of differential equations in the form of electrical signals. Despite its high efficiency (especially compared to the mechanical integration methods), the analog computers lost importance with the advent of computers and now digital methods are predominant. Anyway, Kelvin's method is still applied in the numerical algorithms.

Also on the the 1950s, Jay W. Forrester [3] was confronted with a problem for the company *General Electric* where he understood the need of simulation (of inventory control type). This first simulation that Forrester did, using pencil and paper, can be considered as the beginning of *System Dynamics* [4]. Later he asked Richard Bennett for help to solve the equations using the computer and created a compiler, called SIMPLE, for this purpose. Interestingly, the method of Bennet was the same (adapted to the computer) that Kelvin used to solve differential equations by mechanical methods. Since then, successive generations of SD people have spoken in the *Forrester language*.

2 Simulation and modelling approach

2.1 Why model differential equations in SD

The increasing computation power as well as the rapid development of software and simulation programs allow nowadays to build models of complex technical systems in such areas as architecture, engineering, economics and business, telecommunications, networks and the Internet. The development of these complex models is expensive and requires teamwork among groups of people from different disciplines, often with different academic curricula, what is required to employ a methodology that allows for easy and fast exchange of models and ideas.

However, when multidisciplinary teams are working on models, communication problems arise that may hinder the team integration and then result in a drop of its performance, mainly due to the different training of the persons involved with. So, while people coming from *hard* sciences like physicists, mathematicians or engineers, are used to raise the problems of dynamic systems in terms of differential equations (initial value problem (IVP)); people from economic sciences, biology, architecture and philosophy, feel more comfortable when reasoning using Forrester diagrams, system thinking and stock & flow diagrams. Because these two groups of people have been working for decades in completely disjoint compartments, SD people rarely exchange their views with DS people probably due to the fact that they talk *different languages*.

Regarding methodology, while DS people mainly use block-based tools like Matlab/Simulink and physical modeling tools that allow two-way connections, such as Dymola; SD people typically employ System Dynamics modeling software such as Stella iThink, Vensim, etc. Clearly, an important step to improve the effectiveness of model building in multidisciplinary teams is to try to find a working methodology in which both DS people and SD people feel comfortable. According to the authors, methods used by the SD people are better positioned than DS, as candidates to be elected as common methodology. Some reasons for this are:

1. It is easier to study the SD methods for one who has studied the DS methods than the other way round.
2. While SD systems, depicted as SD diagrams (Forrester diagrams) represent *explicit* differential equations (with derivatives appearing only in the left hand side of the equations) easily resolvable by the numerical calculation solvers; DS systems usually give *implicit* differential equations which results in a set of *differential algebraic equations*, not so easy to solve, mainly for multidimensional, discontinuous and discrete systems.
3. DS, in order to allow more freedom when building models (blocks, bon-graphs and sophisticated components), are a real worry when performing the simulation. For example algebraic loops arise very frequently when modeling systems in Matlab/Simulink. This is noticeable especially in hybrid systems modeling (discontinuous ordinary differential equations) where SD method clearly offers many advantages.
4. A main advantage is the existence of high quality object-oriented software that integrates SD (System Dynamics paradigm) as well as other modeling paradigms explained further, allowing for a multi-approach modeling and high team integration tasks.

The advantages and disadvantages can be summed up and are presented in the following Table 2.1.

Approach	+	-
DS	conciseness reproducibility preciseness	limitations in: space, comprehension, notation non scalable
SD	comprehension scalability multidimensionality	not <i>classical</i> algebra lack of standards unusual representation

Table 1: Advantages and disadvantages of both approaches

2.2 Simulation and modeling tools used in this paper

Matlab is probably the most widely used simulation program for control systems by academics, although there are many other programs like Maple, Mathematica, Octave, Scilab, etc. that are also commonly used and have similar characteristics. With Matlab, simulations are possible in a mathematical sense, i.e. to apply numerical methods for solving some differential equations representing the system. However, being a program designed with the technologies of the 1950-60s, it lacks the advantages of more modern object-oriented based software. These advantages are evident if the system to model is of *discrete event* type and even more if it is a *hybrid* or an *agent-based system*.

AnyLogic is a recognized program in the community of multi-paradigm simulations, but little known in the areas of automation and control engineering, which is based on the latest

advances in object-oriented modeling applied to complex systems [5]. It currently supports three approaches or *modeling paradigms*:

- System Dynamics (SD)
- Discrete Events (DE)
- Agent Based (AB)

These three paradigms are mutually compatible, so that, for example, to model a hybrid system we will use the SD method to model the continuous part of the differential equations and the DE method for modeling the events. AnyLogic models are portable Java applications that can run on their own. They are also multiplatform and can run anywhere a Java Runtime Environment (JRE) is installed. So the models can be also run in a web browser in form of a Java Applet, which allows for an easy way of publishing the models. Moreover, it is very easy to develop animations of active objects: the assembly of the image is done automatically. In this way the animations are highly reusable and can be displayed on the applets.

In AnyLogic scalable models can be easily created, because you can define arrays of objects whose size is a parameter of the model. You can even dynamically change the structure of the model by adding or deleting items or changing their interconnection during runtime to reflect the dynamic changes that can occur in a real system. Regarding the simulation algorithms, some of them have been modified to work in hybrid environments (hybrid state machines).

3 System Dynamics examples

The examples presented below have been chosen to expose some typical problems well known by DS people, especially electrical engineers, accustomed to use sophisticated circuit analysis tools, or mechatronics analysis tools, and to encourage them to use the System Dynamics paradigm. These examples prove that (with a little effort) SD methods are also perfectly valid for analysis of mechanical and electrical systems and that they are easily integrable into multi-approach modeling environments, allowing for integration of DS people within multidisciplinary modeling projects.

3.1 Numerical ODE solution using an SD model

An initial value problem (IVP) is an ordinary differential equation (ODE) with a given initial condition (in form of a specified value) of the unknown function at a given point in the solution domain. In physics and engineering resolving these kind of problems is common, as the differential equation describes a system which evolves with time according to the specified initial conditions. Using SD we can obtain the numerical solution of the first order *control differential equation* (initial value problem),

$$\begin{cases} x'(t) &= f(x(t), u(t)) \\ x(0) &= x_0, \end{cases}$$

where $t \in \mathbb{R}$, $x(t), u(t) \in \mathbb{R}$ and $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$; the (control) function $u(t)$ is given.

Note that the above equation can also represent a multidimensional system if for some given integer n there are $t \in \mathbb{R}$, $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$ and $f : \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^n$

AnyLogic *hyperarrays* allows us to model these multidimensional differential equations, using a very single Forrester diagram, as can be seen in Figure 1.

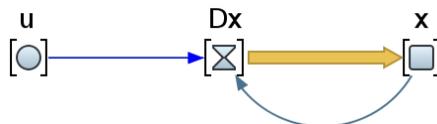


Figure 1: Forrester diagram of an multidimensional ODE

Indeed this diagram can represent a first order system when $n = 1$ or, in general, an order n system for $n > 1$ and where $u(t)$, $Dx(t)$ and $x(t)$ are hyperarrays. In this model, the (given) function $u(t)$ represents the system input (or control input). Dx as a flow variable in DS represents the formal derivative $x'(t)$ of the unknown function $x(t)$, which is represented as a stock.

The thick arrow is nothing more than the *integrator* DS object whereas the thin arrows indicate dependencies of the function

$$f(x(t), u(t)).$$

that is, f depends on x and u (also it depends implicitly on t).

3.2 First order system

To explain a concrete example, a first order ODE given by

$$\text{IVP: } \begin{cases} \frac{dx}{dt} = ax(t) + bu(t), & u(t) \text{ is given} \\ x(0) = x_0 \end{cases}$$

where parameters a, b and control input are given. This system will be modelled as follows.

The way AnyLogic gives to model this system is very easy: after placing the selected objects from Palette into Main window, some Properties should be assigned to them, by *clicking* them. Figure 2 shows a special case from the previous example while its Forrester diagram remains the same (but without array variable settings). After dragging a Stock variable and two Flow variables from Palette to the Main window, by clicking at each one of them we will rename them with appropriate names and assign them some pertinent properties. So in Stock x variable properties we write

$$\frac{dx}{dt} = Dx$$

as well as its initial value x_0 . In Flow Dx variable we should write the function description that in this case is

$$ax + bu$$

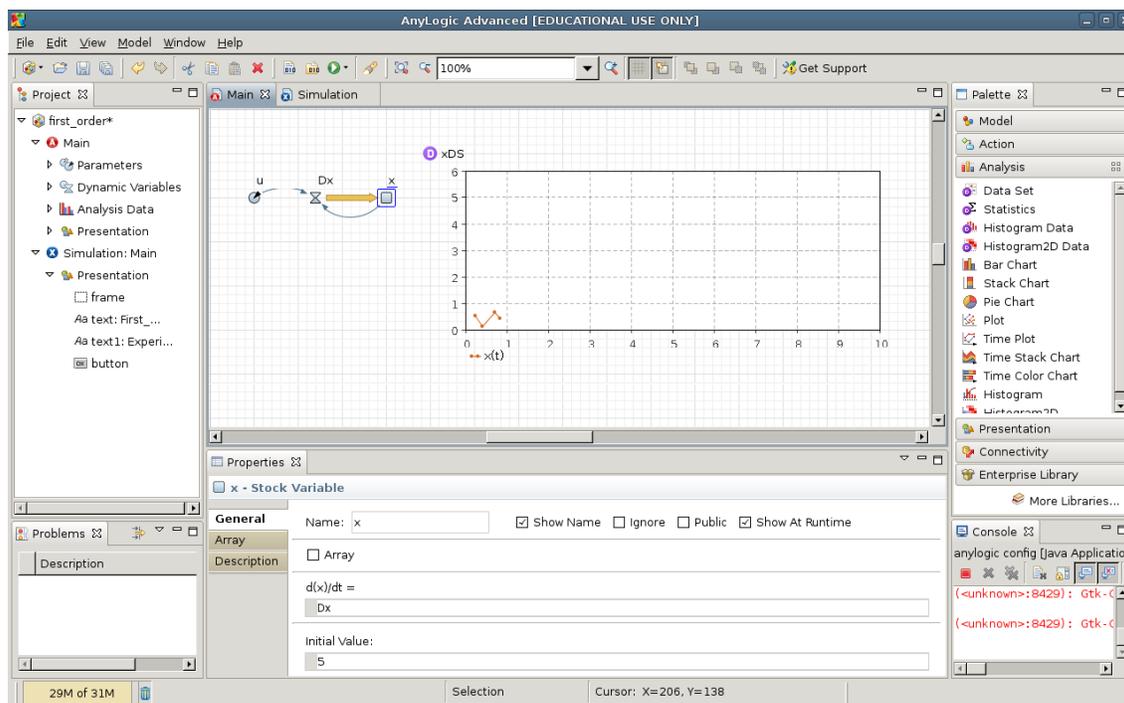


Figure 2: Anylogic interface enables differential equation modeling by drag & drop

where a and b are Java variables of type *double*, declared within the Main window properties. Another double variable x_0 should be declared here. After that, we can run simulation. Also it is possible to adjust some simulation parameters such as plot time window and vertical scale, simulation stop time, etc.

3.2.1 RL circuit

As first example of an electrical system, the *RL* circuit will be discussed. It consists of a resistor, represented by the letter R and an inductor, represented by the letter L . Resistor and inductor are connected in series in this example, as shown in the Figure 3.

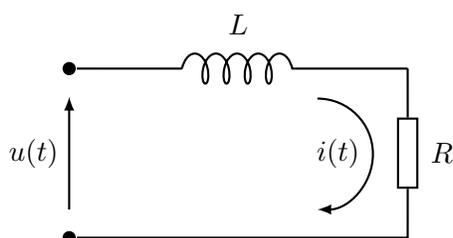


Figure 3: RL circuit

The problem that arises is: for given $u(t)$, $i(0) = i_0$, calculate $i(t)$.

In order to obtain the ODE initial value problem, we use the 2nd Kirchhoff law. Then

we get

$$L \frac{di}{dt} + Ri = u,$$

so the IVP is

$$\begin{cases} \frac{di}{dt} = -\frac{R}{L}i + \frac{1}{L}u \\ i(0) = i_0 \end{cases}$$

and the solution can be computed with AnyLogic in the same way as the previous first order system, taking into account that in this case the unknown function is $i(t)$ and we have

$$a = -\frac{R}{L}, \quad b = \frac{1}{L}, \quad i(0) = i_0$$

The Forrester diagram, similar to the previous generic example, can be seen in figure 4.

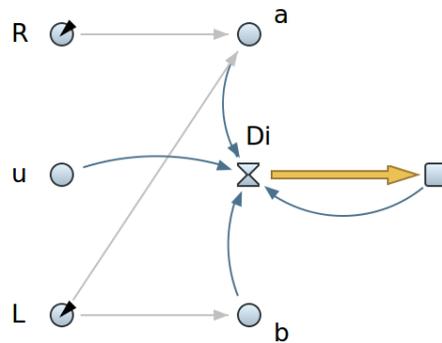


Figure 4: Forrester diagram for the RL circuit

3.2.2 RC circuit

The next example discussed is the RC circuit, which is quite similar to the previous one, but instead of using an inductor, a capacitor, represented by the letter C , is considered here. Like the RL circuit, the RC circuit can be used as filter for signals by letting pass only certain frequencies. Together with the RL circuit, the RC circuit exhibits a large number of important types of behaviour that are fundamental in analog electronics.

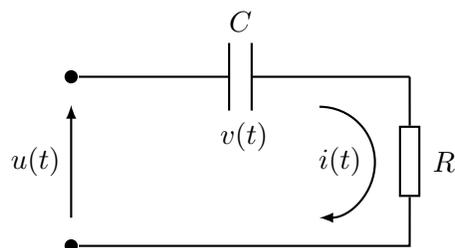


Figure 5: RC circuit

The problem that arises is: for given $u(t)$, $v(0) = v_0$, calculate $v(t)$.

The voltage across the resistor and capacitor are as follows:

$$\begin{aligned} v_r &= Ri \\ \frac{dv}{dt} &= \frac{1}{C}i \end{aligned}$$

Then, from the 2th Kirchhoff law, we get

$$Ri = RC \frac{dv}{dt} = u - v,$$

so the IVP is

$$\begin{cases} \frac{dv}{dt} = -\frac{1}{RC}v + \frac{1}{RC}u \\ v(0) = v_0 \end{cases}$$

and the solution can be computed again with AnyLogic using SD, in the same way as the previous example but in this case the unknown function is $u(t)$ and

$$a = -\frac{1}{RC}, \quad b = \frac{1}{RC}, \quad v(0) = v_0$$

The Forrester diagram is similar as in the previous examples, and can be seen in figure 6.

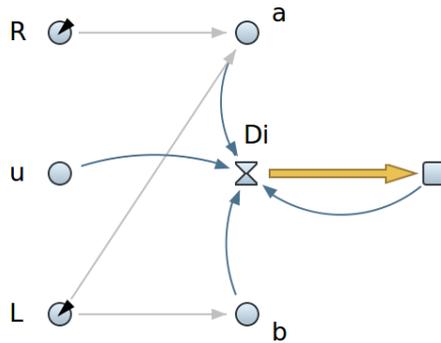


Figure 6: Forrester diagram for the RC circuit

3.3 Second order system

A second order differential equation is an equation involving the unknown function $x(t)$, its first and second derivatives $x'(t)$ and $x''(t)$, and, for *control differential equations*, the given control function $u(t)$. We will consider the Initial Value Problem

$$\text{IVP: } \begin{cases} x''(t) = a_1x'(t) + a_0x(t) + bu(t) \\ x(0) = x_0, \quad x'(0) = v_0 \end{cases}$$

As it is known, any n order explicit differential equation, with some single variable changes, can be translated to a system of n differential equations of first order. So in this second order system, making the changes

$$x := x_1, \quad x' := x_2$$

we will obtain the following IVP with two first order differential equations:

$$\begin{cases} x_1'(t) = x_2(t) \\ x_2'(t) = a_1 x_2(t) + a_0 x_1(t) + b u(t) \\ x_1(0) = x_0, \quad x_2(0) = v_0 \end{cases}$$

It is very easy to model this IVP in AnyLogic, in similar way as the previous first order examples, as it is shown in Figure 7.

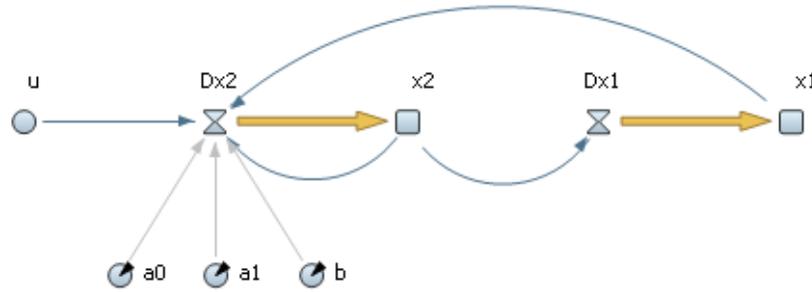


Figure 7: Second order system modelled in AnyLogic

3.3.1 Mechanical system

Second order ODEs appear in many electromechanical systems. As an example, the movement of an object through a viscous fluid tied to a spring will be discussed. The object of mass m is moving through a fluid of viscous damping b and tied to a spring which is fixed on one side and has an elasticity k . The mass is pushed with the force $f(t)$.

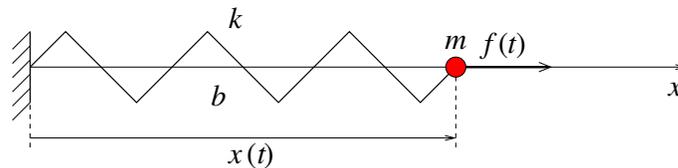


Figure 8: Spring-mass system in viscous fluid

Applying Newton's second law gives

$$m \frac{d^2 x}{dt^2} = -b \frac{dx}{dt} - kx(t) + f(t).$$

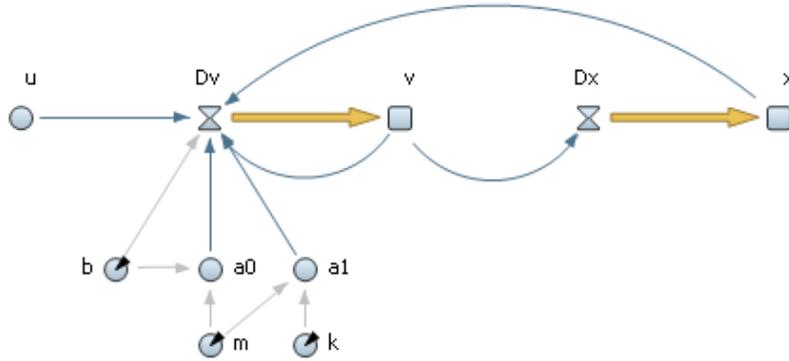


Figure 9: Spring-mass system modelled in Anylogic

Now, with the changes

$$x := x \text{ (no change)}, \quad x' := v$$

we obtain:

$$\begin{cases} x'(t) = v(t) \\ v'(t) = -\frac{k}{m}x(t) - \frac{b}{m}v + \frac{1}{m}u(t) \\ x(0) = x_0, v(0) = v_0 \end{cases}$$

And again it results very easy to model it in AnyLogic, as it is shown in Figure 9. It is even possible to reuse some parts of the previous second order example to build this model.

3.4 Hybrid systems

Power electronic converters as for example buck and boost converters contain switching elements, posing discontinuous differential equations, so always have been difficult to modeling. However, today one can model them using Hybrid System theory. [18, 19]

An hybrid system is composed by two parts: a continuous part and another discontinuous part. The continuous part can be modeled using the System Dynamics paradigm and the discontinuous part can be modeled by means on the so called Discrete Event paradigm. Both of them are provided on the AnyLogic simulation tool.

The analysis that usually is made for the DC-DC converters is based on assuming a priori some operation hypothesis, in order to be able to obtain formulas that therefore will be valid only if such hypotheses are fulfilled. Nevertheless, we will not made previous hypothesis but we will associate the operation modes of the system to different states of an hybrid system.

In the circuits we will suppose switch sw is controlled by a binary periodic signal $\text{clock}(\tau)$, with period T_s , being t_{on} and t_{off} the times during which the function value is 1 and 0 respectively.

We will consider three states, S_{on} , S_{off} and S_{nc} . The states S_{on} and S_{off} will be associate to operation modes with switch sw in states **on** and **off**, respectively, whereas the state S_{nc} will be associate to the mode in which the diode does not conduct (null current). In this

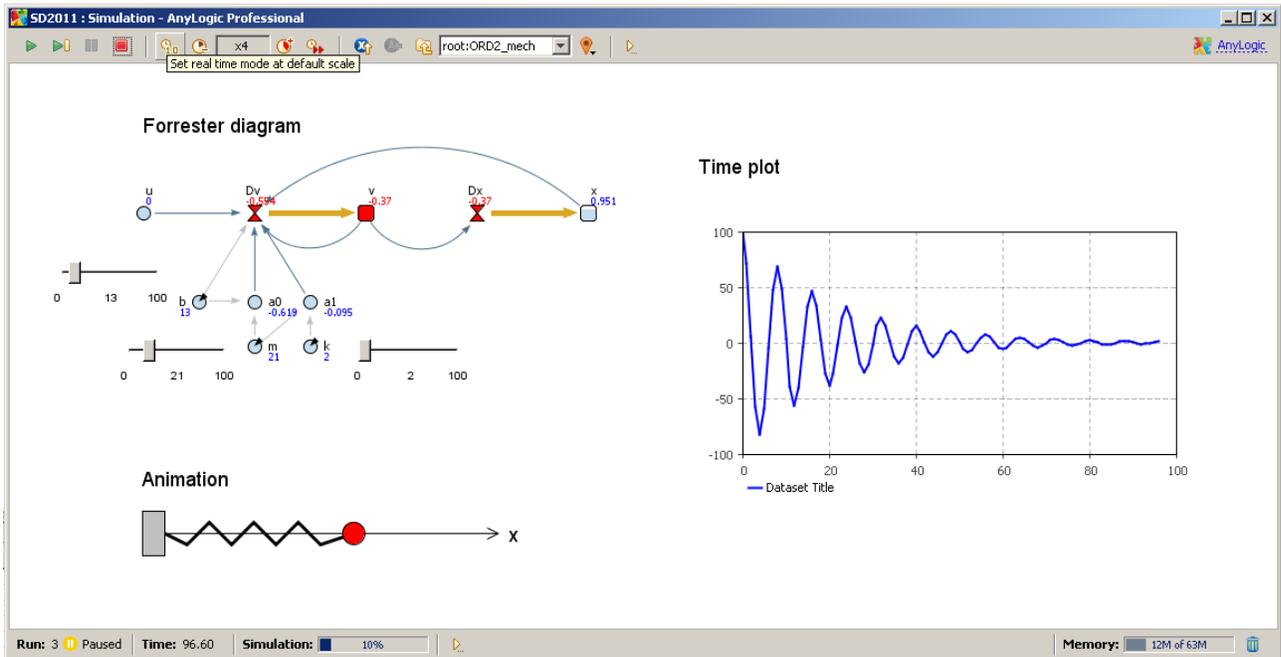


Figure 10: Running simulation of the spring-mass system with time plot and animation

way we are going to analyze the both buck and boost converters. We will denote by i the current through the coil and v the voltage across the capacitor.

3.4.1 Buck converter

This converter gives an output voltage v smaller than the input voltage u . It is based on the circuit of figure 11.

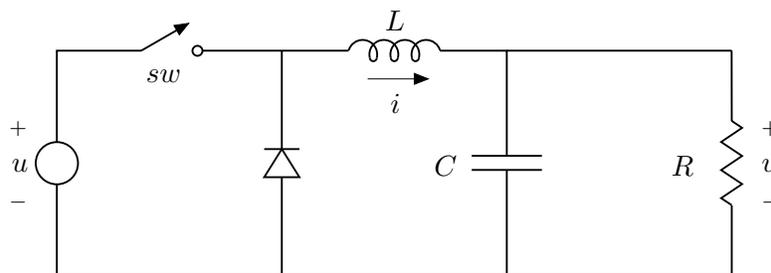


Figure 11: *Buck* converter.

The discrete event system has three states:

S_{on} If switch sw is closed, the diode is on inverse polarization and can be eliminated for analysis. The resulting electrical system, without diode and with sw closed, with two

meshes, is described by the pair of differential equations

$$\begin{aligned}\frac{di}{dt} &= -\frac{1}{L}v + \frac{1}{L}u \\ \frac{dv}{dt} &= \frac{1}{C}i - \frac{1}{RC}v\end{aligned}\tag{1}$$

S_{off} When opening the switch sw , whenever the current is positive, it will also flow through the diode, now directly polarized. For analysis we can replace the diode by a conductor and delete the switch sw and the source of voltage u . The resulting circuit, with two meshes, is described by the equations,

$$\begin{aligned}\frac{di}{dt} &= -\frac{1}{L}v \\ \frac{dv}{dt} &= \frac{1}{C}i - \frac{1}{RC}v\end{aligned}$$

which are the same ones described for the previous mode by doing $u = 0$.

S_{nc} In switch-off mode, with $i > 0$, the voltage v in the capacitor will be increasing and the current will be diminishing; if time is long enough, it will be a moment when i is annulled, later trying the capacitor to discharge through the diode, which is not possible, so there is to be $i = 0$. In this case, with $i = 0$, we can consider the circuit reduced to a single mesh, the one that contains R and C . This circuit is described by

$$\begin{aligned}\frac{di}{dt} &= 0 \\ \frac{dv}{dt} &= -\frac{1}{RC}v\end{aligned}$$

3.4.2 Boost converter

This converter is able to give an output voltage v greater than the input one u . It is based on the circuit of figure 12. The discrete event system has three states:

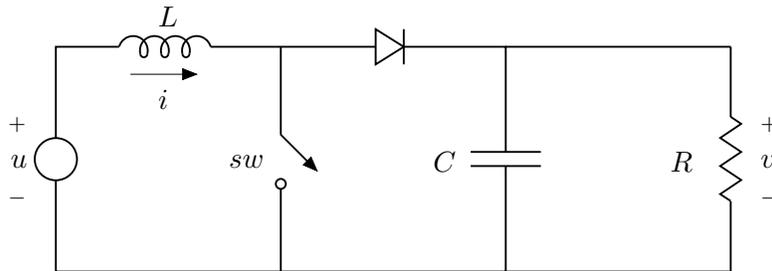


Figure 12: *Boost* converter.

S_{on} When the switch sw is closed, the diode is on inverse polarization (it can be eliminated for analysis), the mesh on the left is isolated and the equations are:

$$\begin{aligned}\frac{di}{dt} &= \frac{1}{L} u \\ \frac{dv}{dt} &= -\frac{1}{RC} v\end{aligned}$$

S_{off} If sw is open, whenever i is positive, the diode is directly polarized. For analysis we can replace the diode by a conductor and eliminate the switch sw . The resulting circuit with two meshes is described by

$$\begin{aligned}\frac{di}{dt} &= -\frac{1}{L} v + \frac{1}{L} u \\ \frac{dv}{dt} &= \frac{1}{C} i - \frac{1}{RC} v\end{aligned}\tag{2}$$

S_{nc} In switch-off mode, with $i > 0$, the voltage v in the capacitor will be increasing and the current i will be diminishing; if time is long enough, it will be a moment when i is annulled, later trying the capacitor to discharge through the diode, which is not possible, so there is to be $i = 0$. In this case, with $i = 0$, we can consider the circuit reduced to a single mesh, the one that contains R and C . This circuit is described by

$$\begin{aligned}\frac{di}{dt} &= 0 \\ \frac{dv}{dt} &= -\frac{1}{RC} v\end{aligned}$$

3.4.3 Hybrid models

Given that equations (1) and (2) are in the standard form of linear systems, i.e. $x'(t) = Ax(t) + Bu(t)$, where $u(t) \in \mathbb{R}$, $x(t) \in \mathbb{R}^2$, $A \in \mathbb{R}^{2 \times 2}$, $B \in \mathbb{R}^{2 \times 1}$, it is very easy to translate the right hand side of differential equations to an AnyLogic Forrester diagram, and their corresponding values on each state of *boost* or *buck* converters, to AnyLogic state actions. Also, the hybrid system transitions are assigned to AnyLogic transitions and the system parameters are assigned to AnyLogic parameters in a straightforward way.

So, for the buck converter, the following statechart properties are chosen:

S_{on} entry action: `a12=-1/L; a21=1/C; a22=-1/(R*C); b1=1/L;`

S_{off} entry action: `a12=-1/L; a21=1/C; a22=-1/(R*C); b1=0;`

S_{nc} entry action: `a12=0; a21=0; a22=-1/(R*C); b1=0;`

$S_{\text{on}} \rightarrow S_{\text{off}}$ transition: `triggered by timeout and timeout=ton`

$S_{\text{off}} \rightarrow S_{\text{on}}$ transition: `triggered by timeout and timeout=Ts`

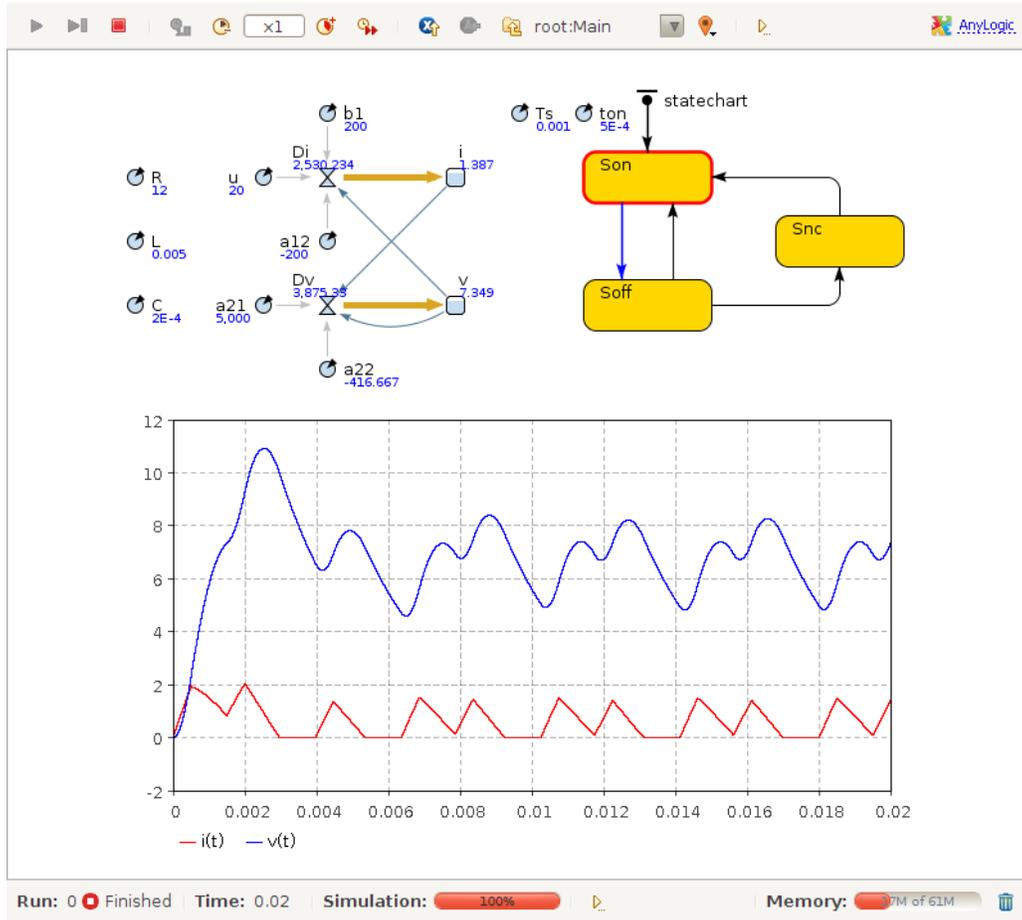


Figure 13: Forrester diagram, statechart and simulation for the buck converter

$S_{\text{off}} \rightarrow S_{\text{nc}}$ transition: triggered by condition and Condition: $i \leq 0$

$S_{\text{nc}} \rightarrow S_{\text{on}}$ transition: triggered by timeout and timeout=Ts

Once made the Forrester diagram for the System Dynamics part and the State Chart for the Discrete Event part, the only thing left is to assign values to parameters. Choosing

$$Ts=1e-3, \quad ton=Ts/2, \quad V=20, \quad R=12.0, \quad L=5e-3, \quad C=200e-6;$$

the simulation in figure 13 has been obtained.

Now for the boost converter, in a similar way, the following statechart properties are chosen:

S_{on} entry action: $a12=0; a21=0; a22=-1/(R*C); b1=1/L;$

S_{off} entry action: $a12=-1/L; a21=1/C; a22=-1/(R*C); b1=1/L;$

S_{nc} entry action: $a12=0; a21=0; a22=-1/(R*C); b1=0;$

$S_{\text{on}} \rightarrow S_{\text{off}}$ transition: triggered by timeout and timeout=ton

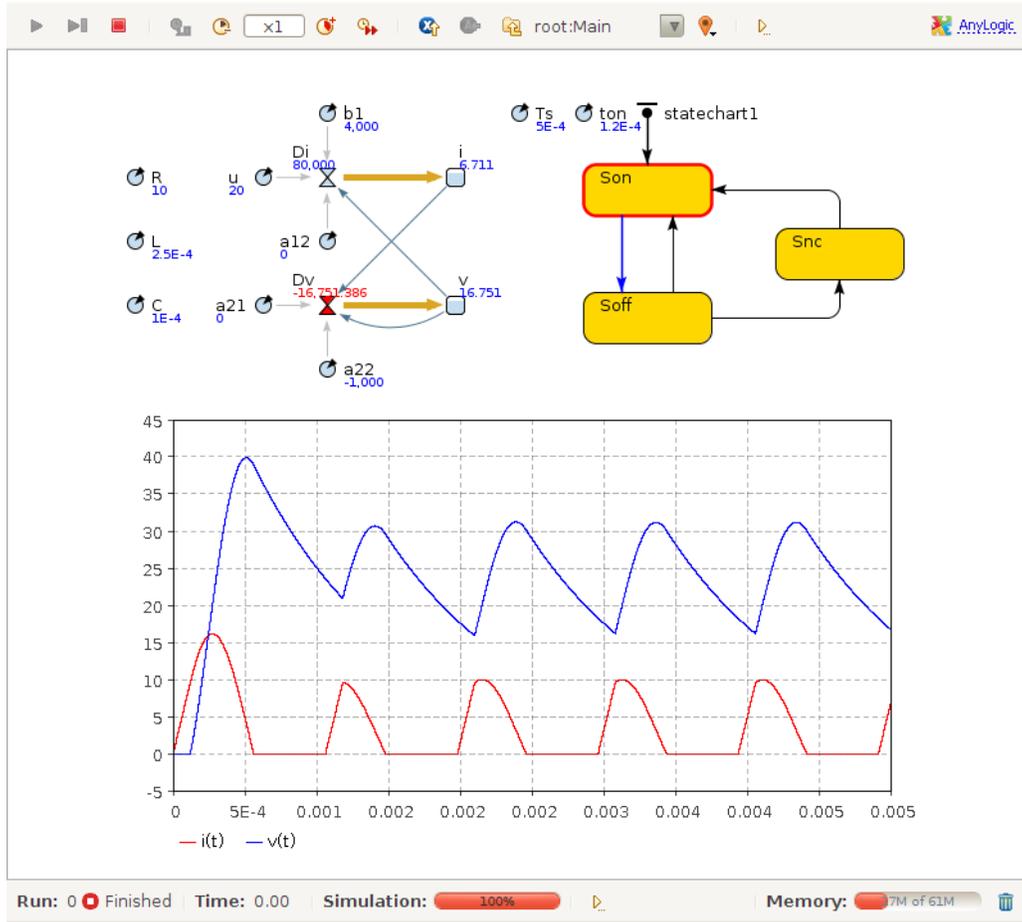


Figure 14: Forrester diagram, statechart and simulation for the boost converter

$S_{\text{off}} \rightarrow S_{\text{on}}$ transition: triggered by timeout and timeout=Ts

$S_{\text{off}} \rightarrow S_{\text{nc}}$ transition: triggered by condition and Condition: $i \leq 0$

$S_{\text{nc}} \rightarrow S_{\text{on}}$ transition: triggered by timeout and timeout=Ts

In this case, with the values chosen for parameters

$$Ts=5e-4, \text{ ton}=1.2e-4, \text{ Vs}=20, \text{ R}=10.0, \text{ L}=250e-6, \text{ C}=100e-6;$$

the simulation in figure 14 has been obtained.

4 Conclusions

As exposed in this paper, SD has many advantages for the resolution of differential equations on electro-mechanical systems, but is not well known by engineers and in more general those who are involved with DS. Probably there is a lack of systematisation and a too large degree of freedom for the engineer approaches. Anyway, if SD models are developed rigorously with

the same coherence than DS models, the advantages in the resolution of problems, and their understanding can be enormously increased.

Moreover, the use of graphical SD interfaces can improve the thinking of DS practitioners and allow them to get integrated into multidisciplinary groups where people of other areas like architects, biologists, economists or philosophers discuss in order to model complex natural phenomena. We hope that these examples can serve as a first step, in the difficult task of communication between SD and DS modellers.

References

- [1] Thomson, J *On an Integrating Machine Having a New Kinematic Principle*. Proc. of the Royal Society of London, 24 : 262-5, 1876
- [2] Kailath, T, *Linear Systems*, Prentice Hall, Inc., 1980
- [3] Forrester, JW, *The Beginning of System Dynamics*, Banquet Talk at the international meeting of the System Dynamics Society, Stuttgart, Germany, 1989
- [4] Gordillo, F, Aracil, J, *Dinmica de Sistemas*, Alianza Editorial, 1998.
- [5] Borshchev, A, Filippov, A *From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools*. The 22nd Int. Conf. of System Dynamics Society, Oxford, England, 2004.
- [6] Coyle, G (2000). *Qualitative and quantitative modeling in system dynamics: some research questions*. System Dynamics Review 16(3): 225-244.
- [7] Coyle R G (1983), *Who rules the waves? A Case Study in System Description*, Journal of the Operational Research Society, Vol.34, No. 9, pp 885-898.
- [8] Coyle R G (1984A), *East and West and North and South*, FUTURES, Vol.16, No. 6, pp594-609.
- [9] Coyle R G (1984B), *A systems approach to the management of a hospital for short- term patients*, Socio-economic Planning Sciences, Vol. 18, No. 4, pp 219-226.
- [10] Coyle R G (1985), *A system description of counter-insurgency warfare*, Policy Sciences, 18, pp 55-78.
- [11] Coyle R G and C J Millar (1996), *A methodology for understanding military complexity: the case of the Rhodesian counter-insurgency campaign*, Small Wars and Insurgencies, Vol. 7, No. 3, pp 360-378.
- [12] Coyle R G and M W D Alexander (1996), *Two approaches to qualitative modeling of a nation's drugs trade*, System Dynamics Review, Vol. 13, No. 3, pp 205-222.
- [13] Coyle R G, M J Bate and S Hamid (1999), *Strategic assessment of present and future security environments: A systems approach*, submitted to Defense Analysis, draft available from authors.

- [14] Doman, A, M Glucksman, et al (1994) *The Dynamics of Managing a Life Insurance Company*. International System Dynamics Conference. Stirling, Scotland.
- [15] Wolstenholme E F, (1983), *The development of system dynamics as a more complete methodology for practical system enquiry*, Proceedings of the System Dynamics Conference, Boston, USA.
- [16] Wolstenholme E F, (1985), *A methodology for qualitative system dynamics*, Proceedings of the System Dynamics Conference, Denver, Colorado, USA.
- [17] Wolstenholme E F (1999), *Qualitative vs quantitative modeling: the evolving balance*, Special issue on system dynamics of Journal of the Operational Research Society, Vol. 50, No. 4, pp 422-428.
- [18] Simić, S N Johanson, K H, Lygeros, J & Sastry, S: *Towards a Geometric Theory of Hybrid Systems* Hybrid Systems: Computation and Control, vol. 1790 of Lecture Notes in Computer Science, pp.421–436, Springer Verlag, Pittsburg, PA, 2000.
- [19] Zulueta, E, Rico, T & González de Durana, J M, *Hybrid modeling of open loop DC-DC converters*. Rev. Fac. Ing. - Univ. Tarapacá [online]. 2003, vol.11, n.2, pp. 41-47. ISSN 0718-1337.
- [20] Garn, H.A., Wilson, R. H. (1972), *A Look at Urban Dynamics: The Forrester Model and Public Policy*, Systems, Man and Cybernetics, IEEE Transactions on 2(2): pp 150-155.