

## Linear Analysis Approach for Modelica Models

Loig Allain

Stéphane Neyrat

Antoine Viel

LMS Imagine

La Cité Internationale, 84 quai Charles de Gaulle, F-69006 LYON

### Abstract

This paper is dedicated to the Linear Analysis of dynamic systems represented by Modelica models. The simulation of dynamic systems helps the engineer in his design activities giving him a better understanding of the system he is in charge. Additionally to time-domain simulations, frequency domain analysis provides a complementary view of the systems. This paper gives a general overview of the Linear Analysis methods that can be performed on Modelica models using the existing facilities in LMS Imagine.Lab platform. It highlights the use of Eigenvalues, Modals Shapes, Transfer Functions or Root Locus to fully understand the intrinsic dynamic behaviors of the systems, including non-linear systems. Different multi-physics examples show how these tools can be used practically.

*Keywords: Modelica; linear analysis; eigenvalues; modal shapes; transfer function; root locus*

### 1 Introduction

The responsibility of any engineer is to find a technical solution to a given problem. Most issues appearing around dynamics systems are related to vibrations or instabilities of the control laws used to pilot these systems. In order to analyze the response of their systems, the engineers can use the computer efficiency. With virtual prototypes, engineers can explore different design options such as evaluating the influence of parameters variations in time-domain simulations. However, more powerful approaches such as frequency-domain analysis can reveal the intrinsic dynamic properties of these systems, independently from their time-domain excitations, and with very few computation times. Finally, the full understanding of the in-depth dynamic behavior of the systems allows making the most appropriate design choices with an important efficiency in the design process (fewer iterations, then reduced development time).

In order to improve the behaviour of the system, several possibilities could be tested:

Some engineers could:

- perform several batch runs in the simulation tool, changing a set of parameters and looking at the corresponding time-responses, with quite long CPU-times.
- launch Design Exploration tools such as Optimization, Design Of Experiments, Monte-Carlo, Pareto Plots, ... with quite long CPU-times.

Or, others engineers would prefer to:

- analyze the root locus of the system as function of each parameter, and to check if the eigenvalues (real and imaginary parts, that provide natural frequency [Hz] and damping ratio [null]) move towards more stable areas.
- analyze the modal shapes in order to understand how each mode contributes to the motion of each output observer variable.
- analyze the transfer functions of the system, in particular the one linking some output observer variables to some input control variables ( $O(s) / I(s)$ ), and to check which are the conditions of resonance (peaks in Bode plots for example), when natural frequencies can be excited by the inputs.

This paper emphasizes the Linear Analysis approach as a very powerful method. State space representation and the Linearization process used in the LMS Imagine.Lab AMESim tool supporting Modelica models are presented. The second section shows some practical applications on multi-domains systems such as mechanical, thermal and electrical models. Finally, the Linear Analysis approach is applied to a Diesel Common Rail Injection System. This general approach is discussed to conclude on its benefits in system modeling.

## 2 Linearization Process of Nonlinear Systems

### 2.1 Level and Formalism for physical modeling

Various formalisms exist to represent physical models. For example, Vangheluwe [1] proposed a classification with a multi-criteria analysis that shows that each formalism has its advantages and its flaws.

	Popularity	Modularity	Adaptability	Numerical Analysis	Topology Analysis
Equations	5	4	3	1	1
State-Space	4	3	2	5	1
Transfer Function	5	3	2	5	1
Bond-Graphs	2	5	5	3	5

Legend:

5	4	3	2	1
very good	good	neutral	bad	very bad

Figure 1: Comparison of formalisms/languages, extract of some criterias from [1]

With the comparison of the formalisms, it then becomes clear that Modelica, with its high level of abstract for the models description, can help the *Equations* formalism to fulfill non-good criteria such as Numerical Analysis or Topology Analysis, to bring *Equations* formalism combined with Modelica to the highest values.

However, it may still be interesting to use formalisms such as *State-Space equations* or *Transfer Functions* commonly used in the Control community, to add valuable information about the system's dynamics (Otter [2]). Since these ways of representation are only usable in linear cases, engineers could think this would strongly limit their applicability. But it is not actually the case since non-linear systems can be linearized around some operating points. It is also very useful to make analogies on complex physical systems to represent them as simpler equivalent linear mass-springs models. Additionally, the utilities developed by control-specialists on the basis of these formalisms are numerous and extremely useful (e.g., the study of dynamics and stability on Bode, Nyquist or Black-Nichols charts, the location of Evans poles, etc. ), which largely justifies their use in the context of process control problems.

### 2.2 Numerical linearization of the system

One of the principal techniques for the analysis of nonlinear system is to approximate them with a proper linear system and then to use the linear sys-

tem theory, which is fully established since about three decades (Russell [3]).

The very interesting point is that whereas engineering systems are never linear, they can practically be approximated as linear, and it appears that such a linear system description is sufficient enough around selected operating points (N [rpm], P [bar], T [°C], ...). It is particularly true when the amplitude and frequency of the excitation signal (the inputs) are kept in between certain limits, which define the domain of linearity.

The linearization process needs the determination of the Jacobian matrix of the system at the desired operating point.

The evaluation of the [A, B, C, D] matrices is quite equivalent to considering the linear behaviour tangent to the nonlinear system in correspondence with an operating point which has also to be an equilibrium point.

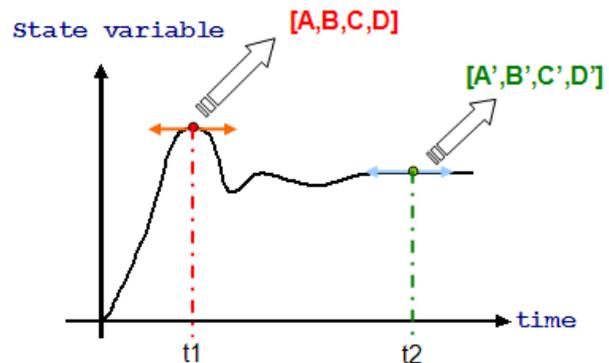


Figure 2 : linearization of a nonlinear system around two different local maximum points

Finally, a full description of the non-linear system around various operating points can be obtained considering several times the linear description of the non-linear system around several equilibrium operating points. Ones could imagine it would lead to a large number of linear analysis to be done, but in practice, only the extreme operating points, such as an injector fully opened and an injector fully closed, are considered.

The non-linear system may be written as:

$$\frac{dx}{dt} = f(x,t)$$

The linearization of these differential equations gives the Jacobian matrix A:

$$A = \left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]$$

Each term of the  $i^{\text{th}}$  vector is evaluated, applying a numerical perturbation to the state variable  $x_i$ :

$$\frac{\partial f}{\partial x_i} = \begin{bmatrix} \frac{\partial f_1}{\partial x_i} \\ \vdots \\ \frac{\partial f_2}{\partial x_i} \\ \vdots \\ \frac{\partial f_n}{\partial x_i} \end{bmatrix} = \begin{bmatrix} \frac{f_1(x_1, \dots, x_i + \delta, \dots, x_N, t) - f_1(x_1, \dots, x_i - \delta, \dots, x_N, t)}{2\delta} \\ \dots \\ \frac{f_n(x_1, \dots, x_i + \delta, \dots, x_N, t) - f_n(x_1, \dots, x_i - \delta, \dots, x_N, t)}{2\delta} \end{bmatrix}$$

The other B, C and D matrices are determined in the same way. For an ordinary differential equation system, the linearized system is represented under the common A, B, C, D state-space representation:

$$\begin{cases} \frac{dx}{dt} = Ax + Bu \\ y = Cx + Du \end{cases}$$

Where:

- $u$  is the vector of the control variables
- $x$  is the vector of the state variables
- $y$  is the vector of the observer variables

From this control representation, it is possible to apply linear algebra algorithms to compute the eigenvalues (that are included in A matrix) representing the system natural modes, the modal shapes of the frequencies (eigenvectors from matrixes A and C) representing the distribution of a frequency all along the system, the transfer functions (linking A, B, C, D matrices) that represent the frequency response in magnitude and phase of an observer variable due to the excitation of another control variable, and the root locus (from A matrix) that represent the evolutions in frequency and damping ratio of the natural modes to some parameters changes in a real/imaginary plot.

For differential algebraic equation (DAE) systems, we first consider a semi-explicit DAE, as obtained by topologically sorting the constitutive equations in LMS Imagine.Lab AMESim:

$$\begin{aligned} \frac{dx}{dt} &= f(x, z, u, t) \\ 0 &= g(x, z, u, t) \\ y &= h(x, z, u, t) \end{aligned}$$

$x$  (resp.  $z$ ) being the differential (resp. algebraic) state variables, in addition to the already defined input and observer variables.

This system may be linearized around an equilibrium point:

$$\frac{dx}{dt} = \nabla_x f \cdot x + \nabla_z f \cdot z + \nabla_u f \cdot u \quad (1)$$

$$0 = \nabla_x g \cdot \frac{dx}{dt} + \nabla_z g \cdot \frac{dz}{dt} + \nabla_u g \cdot \frac{du}{dt} \quad (2)$$

$$y = \nabla_x h \cdot x + \nabla_z h \cdot z + \nabla_u h \cdot u \quad (3)$$

If the system index is one, then the partial Jacobian matrix  $\nabla_z g$  is non-singular. Taking the Laplace transform of (1)-(3), and introducing the augmented state  $\phi = (X(s), Z(s))^T$ , the state-space equations are given by:

$$s\phi(s) = A\phi(s) + B_0 U(s) + B_1 sU(s)$$

with

$$A = \begin{bmatrix} \nabla_x f & \nabla_z f \\ -(\nabla_z g)^{-1} \nabla_x g \nabla_x f & -(\nabla_z g)^{-1} \nabla_x g \nabla_z f \end{bmatrix}$$

$$B_0 = \begin{bmatrix} \nabla_u f \\ 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 \\ -(\nabla_z g)^{-1} (\nabla_x f \nabla_u f + \nabla_u g) \end{bmatrix}$$

As the output of the system is

$$Y(s) = C\phi(s) + D_0 U(s) + D_1 sU(s)$$

with the following additional matrix definitions

$$C = [\nabla_x h \quad \nabla_z h]$$

$$D_0 = \nabla_u h$$

$$D_1 = 0$$

the transfer matrix of the system is finally given by:

$$Y(s) = \sum_{i=0}^1 (C(sI - A)^{-1} B_i + D_i) s^i U(s) \quad (4)$$

The above reasoning can be extended to higher index systems. In this case, the equation (2) is differentiated up to the index of the DAE. In the resulting linearization process, higher order derivatives of the input variables are retained, and thus the transfer matrix defined by (4) is generalized by extending the summation order up to the system index.

Finally, the linear analysis tools used for differential algebraic equation system are the same as the ones used for ordinary differential equation systems.

### 3 LMS Modelica libraries overview

The LMS Modelica libraries are an in-house library that presents several packages as shown below for the first hierarchical level:

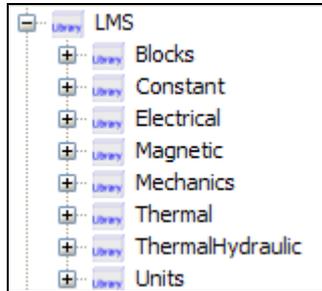


Figure 3: screen shot of some Imagine.Lab libraries in the package browser

The packages are structured in functional sub-packages (Tummescheit [7]). Some of them include quite simple models. The intention there was mainly to support the development of the Modelica compiler embedded in Imagine.Lab AMESim.

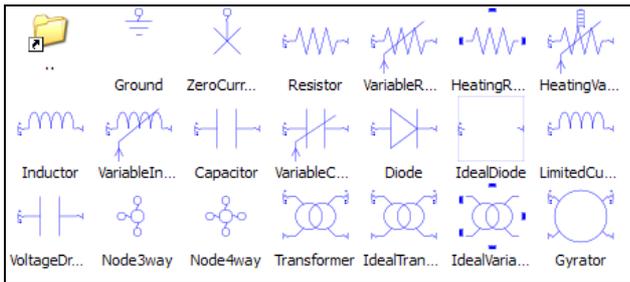


Figure 4: example of Imagine.Lab library for Electrical / Basics components

Other libraries offer more complex models. These LMS Modelica models will be used to illustrate the Linear Analysis approach described in the next sections.

## 4 Linear analysis in various physical domains

The Linear Analysis approach can be applied to any physical system modeled with Modelica models, whatever are the involved domains of physics (Karnopp [4]).

### 4.1 Mechanical systems

Let's consider the example of the longitudinal vibrations of a rod in fixed conditions. We sample the

continuous system in distributed parameters nodes with 7 masses and 8 stiffnesses with dampers:

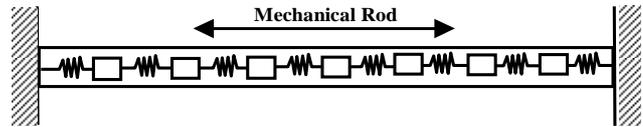


Figure 5: distributed rod in fixed conditions

We consider the Modelica model of the system, built by connecting together the simple Modelica models of mass elements with spring and damper elements:

```

model distributed_rod
  parameter Integer n(fixed=true) = 7;
  LMS.Mechanics.Translational.Mass mass_element[n](each
    m=1.290569);
  LMS.Mechanics.Translational.Spring
    spring_element[n+1](each k=12692085.0);
  LMS.Mechanics.Translational.Damper
    damper_element[n+1](each d=1000.0);
  LMS.Mechanics.Translational.Ground ground1;
  LMS.Mechanics.Translational.Ground ground2;
equation
  connect(ground1.q, spring_element[1].q1);
  connect(ground1.q, damper_element[1].q1);
  for i in 1:n loop
    connect(spring_element[i].q2, mass_element[i].q);
    connect(mass_element[i].q, spring_element[i+1].q1);
    connect(damper_element[i].q2, mass_element[i].q);
    connect(mass_element[i].q, damper_element[i+1].q1);
  end for;
  connect(ground2.q, spring_element[n+1].q2);
  connect(ground2.q, damper_element[n+1].q2);
end distributed_rod;
    
```

Figure 6: Modelica text for the rod model

To get the modal shapes of the system, we need first to set the mass velocities [m/s] as state observers. This is done through the simulation tool GUI. Note that no modification of the Modelica models within the sketch is needed for accessing the linear analysis settings, neither any specific library with added blocks on the sketch. A selection list is directly available at the tool level, being in the Linear Analysis mode:

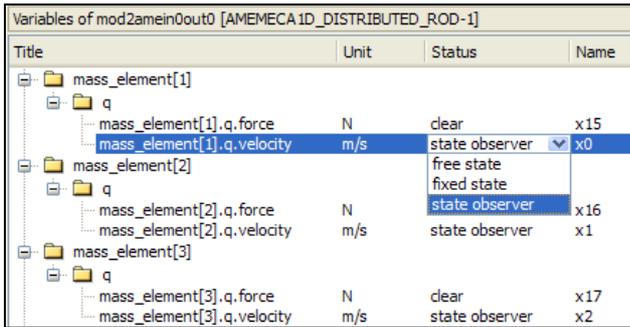


Figure 7: selection list for observer variables in Linear Analysis mode

An additional Linear Status window summarizes all the status (observer / control / ...) of the variables in the Modelica model:

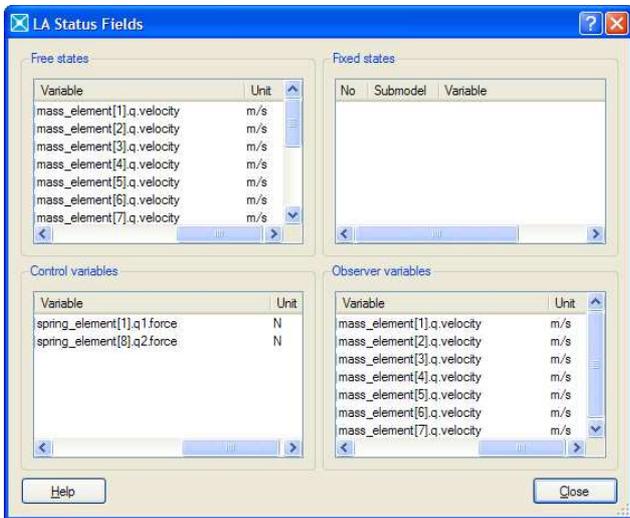


Figure 8: Linearization Status window

These modal shapes provide the distribution of the natural mode along the different observed parts. It is easy to recognize here the well-known normal modal shapes of the continuous rod system (dashed line):

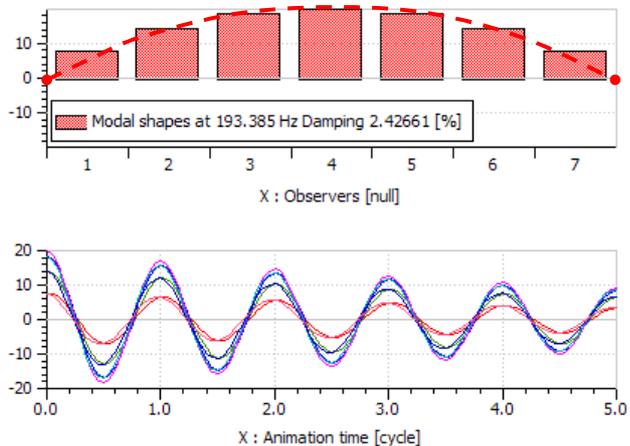


Figure 9: first modal shape ( $f_1=193$  Hz,  $\zeta=2.4\%$ ) of a mechanical rod with fixed-fixed boundary conditions

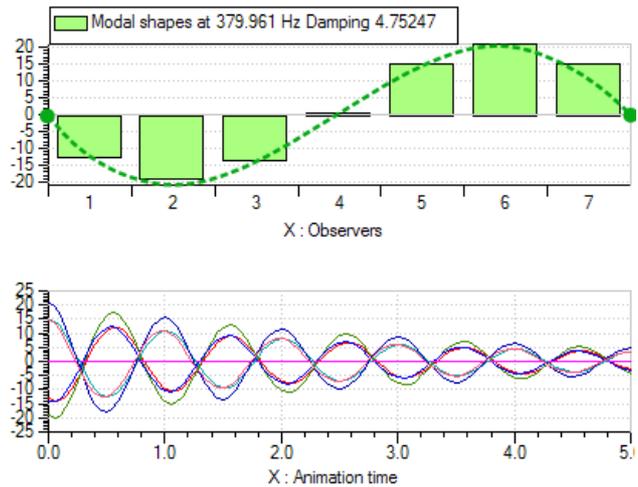


Figure 10: second modal shape ( $f_2=379$  Hz,  $\zeta=4.7\%$ ) of a mechanical rod with fixed-fixed boundary conditions

This approach greatly helps the engineer understanding which parts of the systems are involved in the frequencies present on experimentally measured oscillations.

## 4.2 Thermal systems

We start here from a yet existing Modelica model and we will reuse it to determine the eigenvalues of the system:

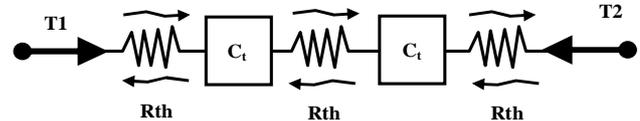


Figure 11: two thermal capacities with conductance

The Modelica model was built as an AMESim supercomponent to be able to mix the Modelica model with non-Modelica (standard C) models.

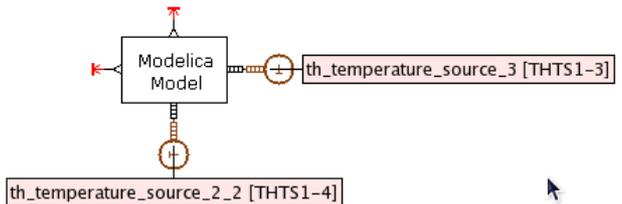
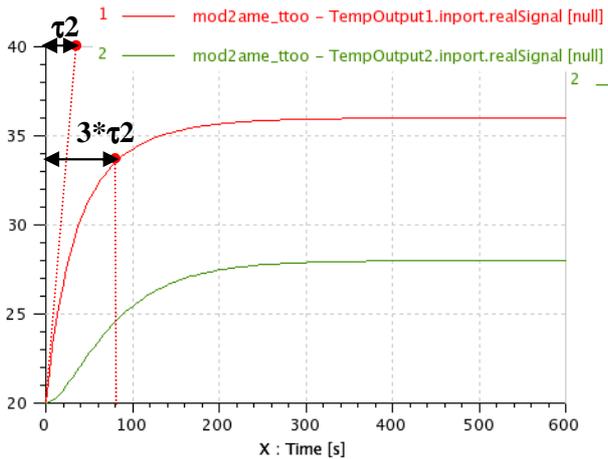


Figure 12: simple Modelica thermal model connected to non-Modelica thermal sources and signal

The Modelica text is quite simple, just calling the Modelica components in libraries to connect them together to get the assembled circuit of the thermal system.

The time-response of the temperature in the thermal capacities are the following:



**Figure 13: evolution of the temperatures Temp [degree] in the two masses in time-domain**

Looking at the eigenvalues shows that there are two time constants  $\tau$ :

Eigenvalues			
Frequency	Damping ratio	Real part	Imaginary part
0.002540	1.000000	-0.015961	0.000000
0.006674	1.000000	-0.041934	0.000000

**Figure 14: eigenvalues of the thermal system**

Then:

- $\tau_1 = 1/(2\pi \cdot 0.002540) = 62.6 \text{ s}$
- $\tau_2 = 1/(2\pi \cdot 0.006674) = 23.8 \text{ s}$

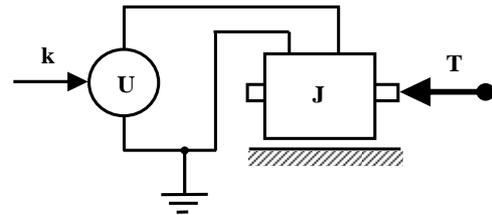
It is clear that these two time constants  $\tau_1$  and  $\tau_2$  directly represent the system dynamics. It is typical to get slow dynamics for thermal systems (range is several [s]).

Alternatively, the analysis of the eigenvalues of this thermal model could help the engineer finding out the best sampling for distributed thermal capacities in his thermal model, in order to reduce the number of states variables, still being sure that the system dynamics are preserved and not altered by the model reduction. This is practically very helpful.

### 4.3 Electromechanical systems

We now show the interest of eigenvalues analysis for electromechanical systems. We consider there the electrical motor of a tailgate motorized opening system.

An electrical motor is loaded by the rotor inertia with an external torque applied as input:



**Figure 15: electrical motor with inertia and torque**

The Modelica text for the electrical model is presented below:

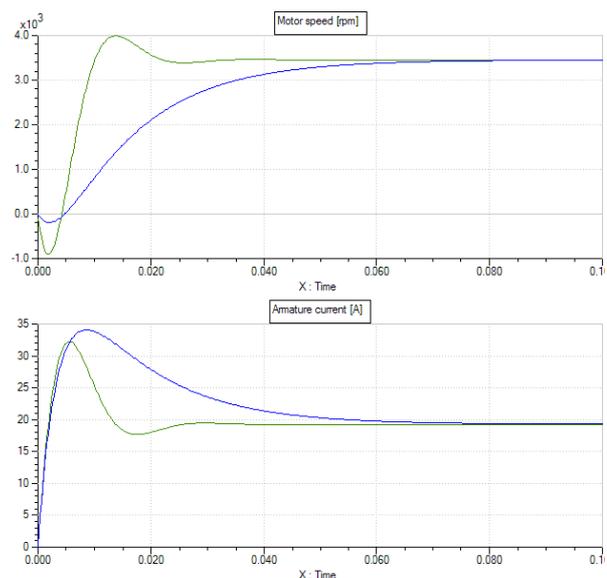
```

model PermanentMagnetDCMotorWithLoad
  LMS.Electrical.ElectricMotors.PermanentMagnetDCMotor
  motor;
  LMS.Electrical.Sources.SignalVoltage Source;
  LMS.Mechanics.Rotational.Inertia J;
  LMS.Mechanics.Rotational.SignalTorque Tau;
  LMS.Blocks.Sources.ConstantSig torque(k(fixed=false)=0.33);
  LMS.Blocks.Sources.ConstantSig voltage(k(fixed=false)=13);
  LMS.Electrical.Ground g;

equation
  connect(g.p, Source.n);
  connect(g.p, motor.n);
  connect(Source.p, motor.p);
  connect(motor.rotor, J.flange_b);
  connect(J.flange_a, Tau.flange);
  connect(Tau.inputTorque, torque.output);
  connect(Source.inputVoltage, voltage.output);
end PermanentMagnetDCMotorWithLoad;
    
```

**Figure 16: Modelica text for the model of electrical motor with inertia and torque**

When the  $U = 13 \text{ V}$  voltage source is activated, the inertia starts increasing its shaft speed up to its steady state value. Depending on the inertia  $J$ , we can see that the transient behavior completely differs:



**Figure 17: evolution of the motor speed  $N$  [rev/min] (top) and the armature current  $I$  [A] (bottom) in time-**

domain –  $J=3 \cdot 10^{-6} \text{ kg.m}^2$  (green curve) and  $J=15 \cdot 10^{-6} \text{ kg.m}^2$  (blue curve)

It is commonly considered that such a system has an electrical time-response  $\tau_1$  and a mechanical time-constant  $\tau_2$ . Actually, the eigenvalues reveal that there are two separated time-constants  $\tau_1$  and  $\tau_2$ :

Eigenvalues			
Frequency	Damping ratio	Real part	Imaginary part
11.511499	1.000000	-72.328879	0.000000
44.192731	1.000000	-277.671121	0.000000

Figure 18: eigenvalues of the electrical system -  $J=15 \cdot 10^{-6} \text{ kg.m}^2$

But these considerations of time-constants being separated are quite abusive since these two time-constants are linked together (it is a second order system). These two time-constants are therefore not uncoupled. For example, considering a change in the moment of inertia from  $3 \cdot 10^{-6} \text{ kg.m}^2$  to  $15 \cdot 10^{-6} \text{ kg.m}^2$  would reveal that the dynamic behavior shifts from an oscillatory mode around  $f = 50 \text{ Hz}$ , well damped  $\zeta = 55\%$ , to the two separated time-constants observed previously. The best way to follow this change of dynamics is to plot a Root Locus after making batch runs with modified inertia values  $J$  [ $\text{kg.m}^2$ ]:

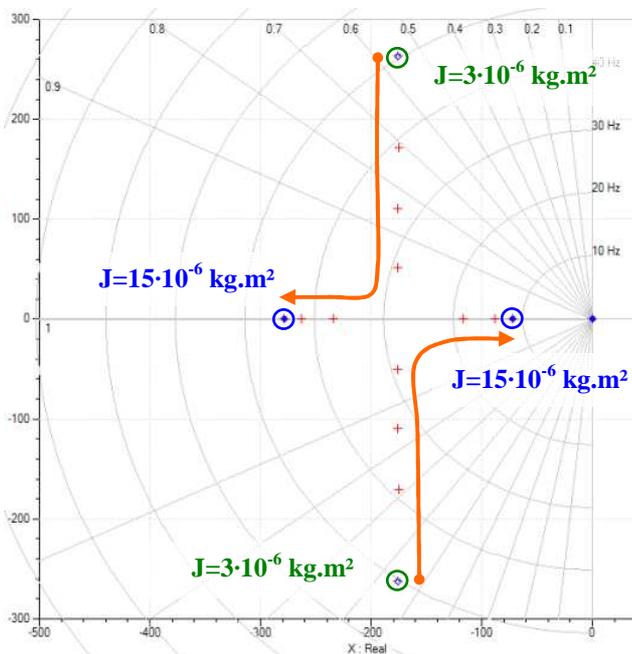


Figure 19: Root-Locus of the electrical motor with inertia -  $J = 3 \cdot 10^{-6} \text{ kg.m}^2$  to  $15 \cdot 10^{-6} \text{ kg.m}^2$

One has to remember the corresponding time-responses associated to the location of the eigenvalues:

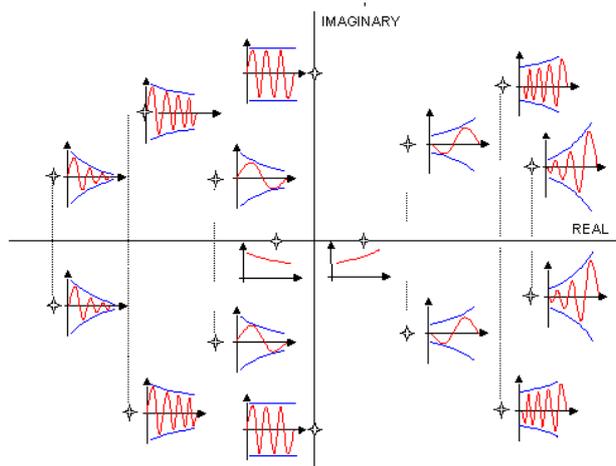


Figure 20: root locus and typical equivalent time-responses

It is finally easy to follow the evolution of the natural modes (frequency  $f$  [Hz] and damping ratio  $\zeta$  [% or null]) to any change of parameters. Root Locus analysis are very appropriate for solving optimal design issues.

## 5 Case Study: Diesel Common Rail Injection System

Diesel engines need to meet reduced fuel consumption and pollutants emissions. In order to reach these targets, Diesel Common Rail injection systems have been introduced some years ago, with various evolutions of their design and architecture. Below is presented a typical Common Rail injection system with a Bosch CP3 pump that delivers the required amount of fuel to be injected by the injectors, depending on the operating conditions ( $N_{\text{engine}}$  [rpm],  $P_{\text{rail}}$  [bar],  $T_{\text{rail}}$  [ $^{\circ}\text{C}$ ], ...).

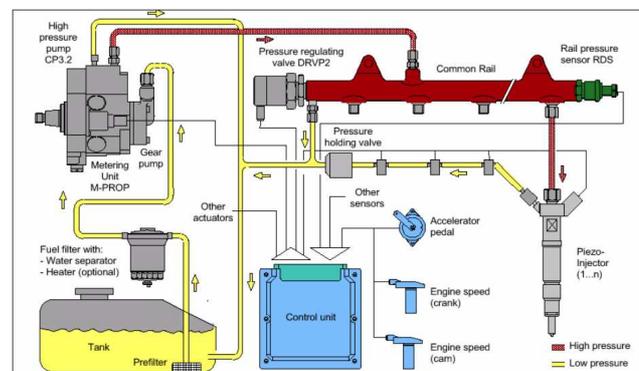


Figure 21: Diesel Common Rail Injection System – Bosch GmbH - Second generation with CP3 pump

Nowadays, the control strategies of injection include multiple-injections to be able to pilot the injected flow rate  $Q$  [mm<sup>3</sup>/ms] that directly impacts the combustion in the engine cylinder:

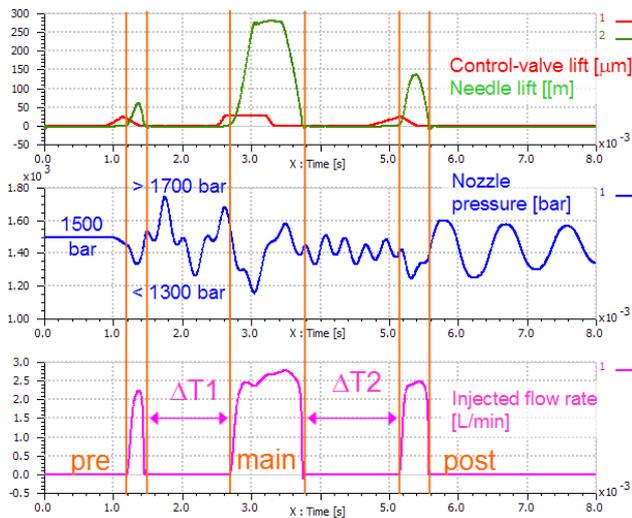


Figure 22: Typical multiple-injection with pre/main/post-injections

The fuel quantities [mg/stroke] to be injected are very small with low deviations admitted from stroke to stroke and from cylinder to cylinder. Therefore, any pressure oscillations as the ones appearing during the injection (typically from 800 Hz to 1000 Hz) have to be understood and controlled, if not damped enough. We propose below to use the Linear Analysis approach on a simple injection system to demonstrate where these frequencies come from. It provides a complementary view to time-domain analysis such as the one proposed by Corno, Casella and All [5] for Gasoline injection system in Modelica.

### 5.1 Coupling between L4 cylinders

We propose to start with a L4 engine.

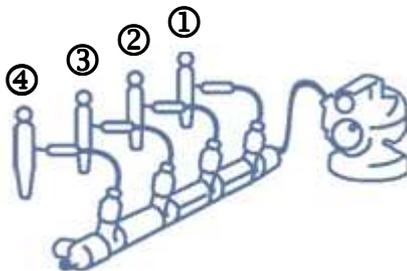


Figure 23: schematics of a L4 injection system: pump with common rail and 4 injectors

The injector is detailed below. Note that an inner pipe is located around the injector needle, which runs from the filter at one end up to the nozzle at the other

end. The dimensions of this inner injector line are usually quite similar to the dimensions of the connecting line from the rail to the injector ( $\varnothing 2.4$  mm –  $L = 150$  mm to 200 mm):

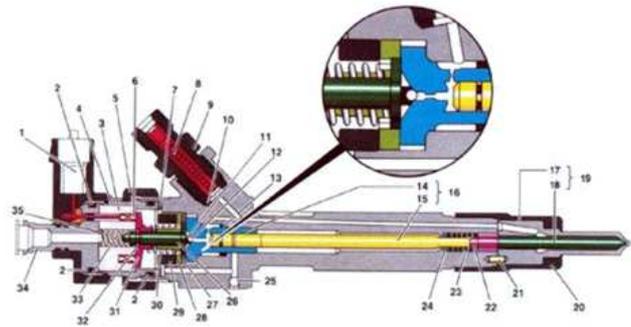


Figure 24: Common Rail Injector – cross-sectional view

The injector ends by the nozzle that delivers the amount of fuel in the cylinders. As a first approach, the nozzle can be modeled as a simple orifice.

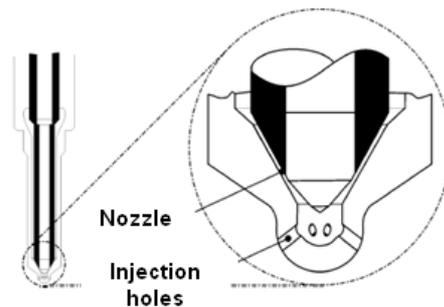


Figure 25: Nozzle with injection holes

The fuel used is a standard ISO4113 diesel. The media properties are based on measurements in the temperature range of +10 to +120°C and pressure range of 0 to 2000 bar:

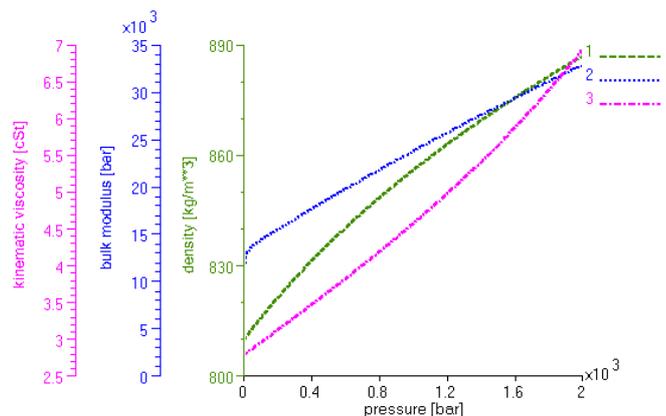


Figure 26 : ISO4113 properties – 0 to 2000 bar – +40°C, from Chaufour and All [6]

For computing the linear state-space representation of the injection system parts in the next section, we will only consider the  $P_{rail} = 1000$  bar and  $T = +40$  °C operating conditions.

ISO4113 media properties @ +40 °C	Iisentropic bulk modulus [bar]	Density [kg/m <sup>3</sup> ]	Kine-matic viscosity [cSt]
$P_{rail} = 0$ bar	9 700	810	2.8
$P_{rail} = 1000$ bar	23 700	855	4.6
$P_{rail} = 2000$ bar	32 800	886	6.9

Figure 27 : ISO4113 fuel properties

The L4 injection system for the common rail with 4 injectors sub-system without the pump can be simplified into the following system, the rail being considered as a distributed volume with closed-closed boundary conditions, and the injectors with connecting pipes being represented by their hydraulic inertances and hydraulic stiffnesses:

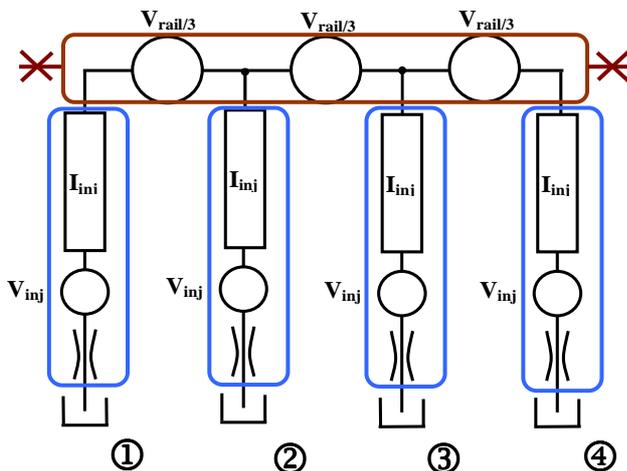


Figure 28: simplified injection system with 4 injectors

The model is based on a prototype LMS thermal-hydraulic library which is still under development. The components are limited to thermal-hydraulic capacity C, resistance R and inertance I, with usual equations. The specific point here is that the ISO4113 Diesel properties are called from the standard Imagine.Lab AMESim Bosch properties with external C-coded functions:

```
function mo_tfrhopti_
  input Real P;
  input Real T;
  input Real fluid_index;
  output Real rho;
  external "C";
end mo_tfrhopti_;
```

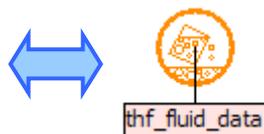


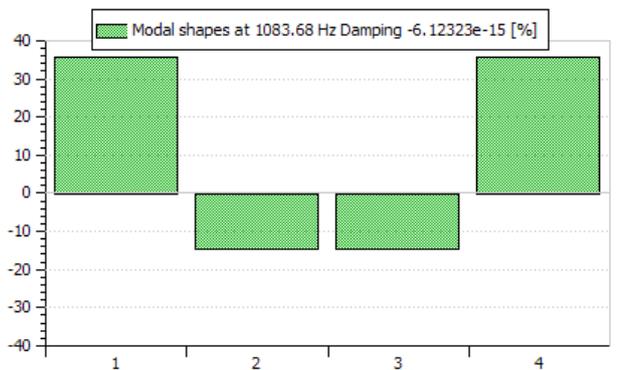
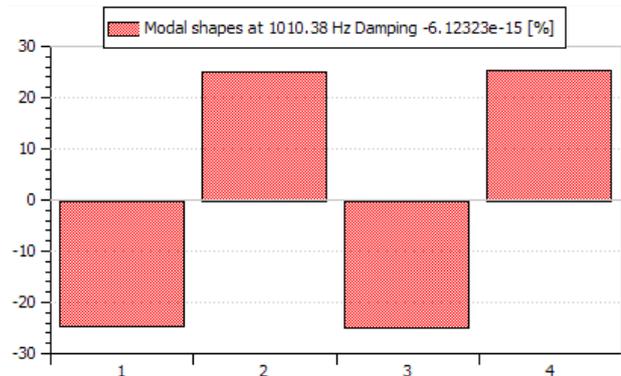
Figure 29: Modelica text for external function call of ISO4113 media properties

Currently, the library is not finalized then its design could be highly improved in the future.

Title	Value	Unit
Capa2p		
p1		
Capa2p.p1.T - Temperature in the connection point	313 K	
Capa2p.p1.p - Pressure in the connection point	1000 barA	
Capa2p.fi - fluid index	1 null	
Capa2p.vol - capacity volume	0.03 L	
ConstantDMTSource		
ConstantDMTSource.fi - fluid index	1 null	
ConstantDMTSource.T0	313 K	
ConstantDMTSource.dm0	0.05 kg/s	
Res		
Res.fi - fluid index	1 null	
Res.lamc - Critical flow number	1000 null	
Res.cqmax - Maximum flow coefficient	0.7 null	
Res.deq - Equivalent orifice diameter	0.00014 m	
ConstantPTSource		
ConstantPTSource.T0	313 K	
ConstantPTSource.P0	50 barA	

Figure 30: example of parameters window for a thermal-hydraulic system with the prototype library

This library is used for applying the Linear Analysis tools. For example, the modal shapes are computed for the first 4 natural modes. They highlight that the 4 injectors are actually coupled together, even through the large diameter rail, since the 4 inertances contributions are always combined together in every modal shape. Note also that their frequencies are very close.



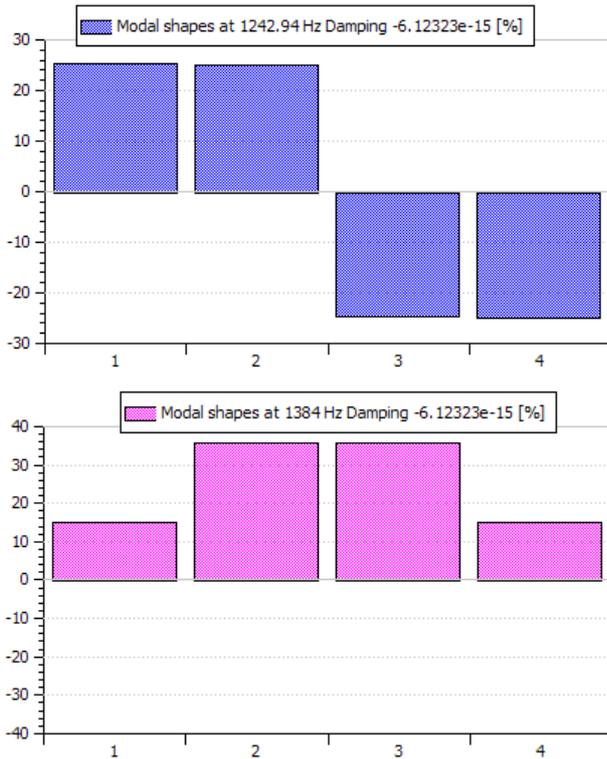


Figure 31: modal shapes of the first 4 modes – simplified injection system with 4 injectors

These values from  $f_1 = 1010$  Hz to  $f_4 = 1384$  Hz are finally close to the oscillations observed in experiments as presented in Figure 22 with a 1 ms period for one oscillation, which corresponds to approximately 1000 Hz.

Additional transfer functions could give more information. They could help determining if the associated gains [dB] of the nozzles pressures are important when an excitation occurs in one cylinder. It is indeed likely that the natural mode the closest to where the injection occurs would be the more excited one.

Finally, the modal shapes help understand how the system dynamics are organized. Some hydraulics/mechanics analogies (Viersma [8]) through the hydraulic stiffness  $K_{hyd}$  [Pa/m<sup>3</sup>] and the hydraulic inertance  $I_{hyd}$  [kg/m<sup>4</sup>] would show that the rail with the 4 injectors and connecting pipes behave as a simple masses-springs system:

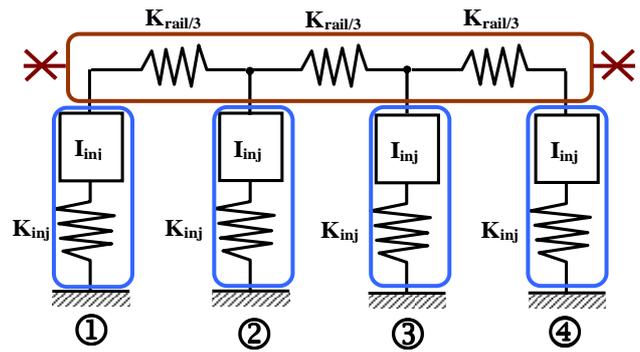


Figure 32: equivalent mass-spring model for common rail with 4 injectors

For the injector with connecting line included, we have:

$$I_{hyd\_inj} = \frac{\rho \cdot Length}{Area} = \frac{855 \cdot 0.385}{\frac{\pi (2.4 \cdot 10^{-3})^2}{4}} = 7.27 \cdot 10^7 \text{ kg / m}^4$$

$$K_{hyd\_inj} = \frac{Bulk}{Volume} = \frac{23700 \cdot 10^5}{0.81 \cdot 10^{-6}} = 2.93 \cdot 10^{15} \text{ Pa / m}^3$$

From this equivalent model, we would find directly the observed characteristic frequencies as:

$$f = \frac{1}{2\pi} \sqrt{\frac{K_{hyd\_inj}}{I_{hyd\_inj}}} = \frac{1}{2\pi} \sqrt{\frac{2.93 \cdot 10^{15}}{7.27 \cdot 10^7}} = 1010 \text{ Hz}$$

## 6 Conclusions

This paper demonstrated how Linear Analysis can be used to understand the dynamic behavior of a Modelica model. The view is independent from the time excitation so that the dynamic answer of such system can be understood for any type of excitations, with very few associated CPU-time. After a brief theoretical overview, the Linear Analysis approach has been applied to a series of different Modelica models to demonstrate that it is valid on different fields of physics, and that it can also be used for non-linear systems with a great efficiency. Next steps would be to highlight the use of such Linear Analysis tools in the context of models reduction (to reduce CPU-time to reach Real-Time performances during the MIL/SIL/HIL process) or for the design of the Control laws of closed loop systems (including Black-Nichols or Nyquist charts for control stability analysis). Alternatively, some works around formal linearization to get the Jacobian with Modelica libraries would bring even more added-value in the linearization process of nonlinear dynamic systems.

## References

- [1] “Multi-Formalism Modelling and Simulation” – H. Vangheluwe, Thesis, Universiteit Gent, Faculteit Wetenschappen, 2000-2001
- [2] “The LinearSystems Library for Continuous and Discrete Control Systems” – M. Otter, Modelica Conference, 2006.
- [3] “Mathematics of finite-dimensional control systems: theory and design” - D. L. Russell, Marcel Dekker, 1979.
- [4] “System Dynamics – Modeling and Simulation of Mechatronics Systems” - D.C. Karnopp, D.L. Margolis, R.C. Rosenberg, 3rd Edition, John Wiley & Sons Inc., 2000
- [5] “Object Oriented Modeling of a Gasoline Direct Injection System” - M. Corno, F. Casella, S. M. Savaresi, R. Scattolini, Modelica Conference, 2008.
- [6] “Advanced Modeling of a Heavy-Truck Unit Injector System and its Application in the Engine Design process” - P. Chaufour, G. Millet, M. Hedna, S. Neyrat, E. Botelle – SAE2004-01-0020, 2004
- [7] “Design and Implementation of Object-Oriented Model Libraries using Modelica” - H. Tummescheit, Lund, Sweden: PhD thesis, Department of Automatic control, Lund Institute of Technology, 2002.
- [8] “Analysis, Synthesis and Design of Hydraulic Servosystems and Pipelines”- T.J. Viersma, Elsevier Scientific Publishing Company, 1980