Object Oriented Modeling of a Gasoline Direct Injection System

Matteo Corno Francesco Casella Sergio M. Savaresi Riccardo Scattolini
Dipartimento di Elettronica e Informazione, Politecnico di Milano,
Piazza Leonardo da Vinci, 32. 20133 Milano

Abstract

The topic of this paper is the object oriented modeling of a Common Rail Direct Injection System of a gasoline engine. The injection system of a gasoline engine is described; the main functional elements are detailed and an object oriented implementation using the Modelica language is proposed. The availability of a fast and easily reconfigurable simulator allows to study how different parts of the system interact and notably speeds up the design of the final system. The use of the Modelica language allows to seamlessly put together mechanics, fluid dynamics, and control algorithms. The design problem can be therefore approached as a whole, in a genuine and modern codesign approach.

Keywords: automotive; fluid dynamics; common rail injection system simulation.

1 Introduction

In this work we present an object oriented simulator of a Direct Common Rail Injection System of a gasoline engine.

The key to designing a clean and efficient ICE (Internal Combustion Engine) lies in precise control of the combustion. This can be achieved by accurate control of the flow of fuel and air in the combustion chambers. Pre-2000 injection systems (such as mechanical carburetors [10] and Multi Point Injection technology [9]) cannot meet today's stringent pollution regulations [8]. The introduction of the Common Rail Injection System technology for Diesel engines [1, 7] in the 90's represented a great breakthrough. Now, it is possible to precisely mix fuel and air directly in the combustion chamber. Only a few years had to pass before the same technology could be applied to gasoline engines [4, 5, 12], thus increasing fuel efficiency and reduce emissions. The cost of these advantages is a more complex system, both from the standpoint of mechanics and electronics. The higher complexity makes it more difficult to foretell the effects of a modification of the elements of the system. The design of such a complex system can greatly benefit from the availability of a reconfigurable simulator. The design process can be sped up and the cost cut down.

The goal of this work is to describe an object oriented simulator of a modern Common Rail Injection System and to show how it can be helpful in the design of the injection system.

The work is structured as follows. Section 2 describes the overall architecture of the system. In Section 3, the mathematical model of the system is derived and its Modelica [2] implementation illustrated. In Section 4, it is shown how the model reconfigurability can be exploited for fast sensitivity studies. Finally, conclusions are drawn in Section 5.

2 Common Rail Injection System

A Common Rail Injection System schematics is depicted in Fig.1; the injection system goal is to deliver fuel to the injectors at a desired high pressure. The system can be divided into two sections; a high pressure circuit and a low pressure one. The low pressure circuit is composed of the fuel tank, a low pressure pump, filters and a pipeline. The low-pressure circuit is not critical for the overall engine performance and therefore it is out of the scope of this work. From the system dynamics standpoint, the most interesting part is the high-pressure circuit; it goes from the the high-pressure pump to the injectors. Its main elements are now briefly described:

- HIGH PRESSURE PUMP. It is a volumetric pump that connects the two main circuits of the system. It is used to increase gasoline pressure from 6 bar to [30-150] bar, according to the working load and engine speed. The piston of the pump is mechanically connected to the engine camshaft through a cam and follower system. In modern common rail pressure control system the control valve is built in the high pressure pump.

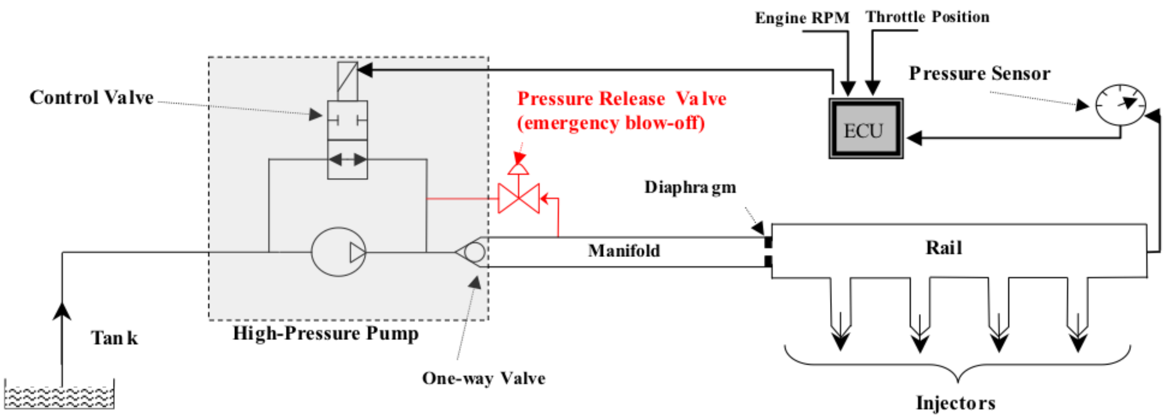

Figure 1: The Common Rail Injection System architecture.

Three main elements compose the pump: the piston, the control valve and the one-way valve. When the control valve is open, the output of the pump is redirected back to the low-pressure circuit. It has only two states: open or closed and the switching instant is the control variable of the system. The one-way valve is used to avoid unwanted refluxes.

Two phases are periodically alternated: an aspiration and a compression phase. In the first phase, the piston moves downward while the control valve is open. This allows the gasoline to flow from the low pressure circuit to the pump chamber. After the piston has reached its lower dead point, the compression phase starts. Initially the control valve is open and the gasoline flows back to the low pressure circuit. At any time during this phase the controller can command the closure of the control valve. When it is done, the pressure in the chamber increases. As soon as it surpasses the pressure in the high pressure circuit, the one way valve opens, letting the gasoline flow into the high pressure circuit. Notice that the control variable of the system is the closing instant of the control valve. The controller must be synchronized with the pump: when the piston is at its upper dead point (end of compression), the ECU (Electronic Control Unit) samples the pressure sensor output and computes the control action. The control action will be executed, at the earliest, when the piston reaches its lower dead point.

- MANIFOLD. The outlet flange of the high-pressure pump is connected to the common rail through the manifold. The one-way valve prevents the gasoline from flowing back from the manifold to the pump. The manifold is equipped with a safety valve that opens whenever the pressure in the manifold reaches a threshold. This valve is designed to function as a safety device and its opening should be avoided in nominal conditions.

- DIAPHRAGM. The common rail and the manifold are connected through a diaphragm that reduces the diameter of the pipeline. This component provides for a better damping of the system and a partial decoupling between the pressures of the manifold and the pressure of the rail. This is achieved at the price of a decreased energy efficiency of the whole system.

- COMMON RAIL. It is the core of the system. It is connected to the manifold through the diaphragm. All the injectors are connected to the common rail. In order to achieve precise injection control, the pressure in the common rail must be regulated at the desired reference value, minimizing oscillations.

- PRESSURE SENSOR. It measures the pressure in the common rail at its end. It is the only pressure measure available for control.

3 Object Oriented Modeling

In this section the mathematical modeling of the system and its implementation in the Modelica language are described. First, the core of a simplified fluid dynamics library is described and then it is shown how it is possible to use the library (along with the Modelica Standard Library) to implement the injection system. Modelica turns out to be well suited for aiding the design of injection systems. Injection systems are inherently multi-domain systems and the possibility to change each element of the system without having to redesign all the interconnections allows to easily study the effects of different design choices. The use of such a fast and user-friendly virtual prototyping system can save numerous iterations of the design-prototyping-testing cycle.

3.1 Simplified Fluid Dynamics Library

Being fluid dynamics the most important dynamic phenomenon of the system, a new simplified fluid dynamics library has been developed. The library is built on the two new connectors: the `flange_in` and `flange_out` defined as:

```
connector flange_in
 "Connector for the simplified
 fluid dynamics library"

 import Modelica.SIunits.AbsolutePressure;
 import Gasoline_turbo.Types.MassFlow;

 flow MassFlow q
    "Mass flow into the flange";
 AbsolutePressure p
    "Pressure at the flange interface";

end flange_in;
```

The absence of a temperature variable in the connector deserves a comment; the library is not designed to model thermal interaction. This choice is based on the consideration that, although thermal interaction does take place in the system, the focus of this study is on pressure waves. Wave propagation is faster than any thermal interaction that may happen in the system and therefore heat exchanges can be neglected.
In addition to the connectors, two partial packages have been introduced: the `BaseFluid` and `BaseMaterial`. The former representing the fluid properties and the second representing the properties of the material of the pipeline.

```
partial package BaseFluid

import Modelica.SIunits.AbsolutePressure;
import Modelica.SIunits.BulkModulus;
import Modelica.SIunits.KinematicViscosity;
import Modelica.SIunits.Density;

  replaceable partial function getDensity
    "Return density as function
    of absolute pressure"
    extends Modelica.Icons.Function;
    input AbsolutePressure p "Pressure";
    output Density rho;
  end getDensity;

  replaceable partial function getBulk
    "Return BulkModulus as function
     of absolute pressure"
    extends Modelica.Icons.Function;
    input AbsolutePressure p "Pressure";
    output BulkModulus beta;
  end getBulk;

  replaceable partial function getViscosity
  "Return Viscosity"
    output KinematicViscosity vcin;
  end getViscosity;

end BaseFluid;
```

The main properties used in the modeling are density, bulk modulus and kinematic viscosity. The package allows to add new properties where needed and redefine how the properties are computed. For example, the designer can choose to use a constant value for the density or model it as a function of the pressure. This gives the user a certain amount of freedom whenever the fluid properties are not readily available. In this injection system study a gasoline package has been implemented assuming a linear dependence of the bulk modulus on the pressure.
Similarly the `BaseMaterial` package is defined as:

```
  partial package BaseMaterial

    import Gasoline_turbo.Types.*;

    replaceable partial function getYoung
      extends Modelica.Icons.Function;
      output Young y;
    end getYoung;

    replaceable partial function getPoisson
       extends Modelica.Icons.Function;
       output Poisson p;
    end getPoisson;

  end BaseMaterial;
end BaseFluid;
```

The main properties of the material are assumed to be the Young and Poisson moduli. They characterize the material elasticity, which plays an important role in pressure waves propagation. In the final implementation, the two moduli are assumed constant, but the replaceable feature allows to take into account more complex models. The partial package has been extended into two different types of steel: the manifold steel and the common rail steel.

3.2 Models Description

As already mentioned the most important dynamics affecting the system is the propagation of pressure waves in narrow, long, circular section pipelines filled with a compressible fluid. According to the description given in Section 2, the injection system is obtained by connecting the following elements: pipelines, diaphragm, high pressure pump, and valves. The closed-loop pressure control and synchronization algorithms also need to be modeled.
Fig. 2 is a graphical representation of the Common Rail Injection System. The most important elements of the system are depicted in figure and are described in the following.

3.2.1 Common Rail and Manifold

The `DistributedPipe` is constituted by two connectors, a `flange_in` and a `flange_out` and the two replaceable packages, `BaseFluid` and `BaseMaterial`; this allows to specify the fluid and material properties independently of the wave dynamics model.
The distributed pipe model equations are derived under the following assumptions:

1. single component, single phase fluid;
2. one-dimensional spatial model;
3. negligible heat exchange phenomena;
4. straight and constant section pipelines.

Under these assumptions, the dynamics of the fluid in the pipeline are described by mass and momentum balances, which can be written as [11]:

$$\begin{cases} \frac{A}{c^2}\frac{\partial P}{\partial t} + \frac{\partial w}{\partial x} = 0 \\ \frac{\partial}{\partial t}\left(\frac{w}{A}\right) + \frac{\partial}{\partial x}\left(\frac{w^2}{A^2\rho}\right) + \frac{\partial P}{\partial x} + F_f = 0 \end{cases} \tag{1}$$

In system (1), ρ is the gasoline density expressed in kg/m^3; w is the gasoline mass flow expressed in kg/s; P is the gasoline pressure expressed in Pa; c is the sound velocity in the fluid expressed in m/s; A is the section area of the pipe expressed in m^2; and F_f is the load loss due to friction. The sound velocity term c depends on the properties of the fluid and the elasticity of the pipe and it is given by:

$$c = \sqrt{\frac{\beta}{1+K\beta}\frac{1}{\rho}} \tag{2}$$

where β is the bulk modulus that describes the compressibility of the fluid and K is the stiffness of the pipe that depends on its material and on its geometric properties. In addition, assuming turbulent flow (this assumption is verified a-posteriori) the frictional load loss can be written as:

$$F_f = 2\frac{w|w|}{\rho A^2 D} f \tag{3}$$

where D is the inner diameter of the pipe and f is the pipe Fanning friction factor. The final equations are partial differential equations. The infinite-dimensional system can be transformed in a finite-dimensional one by means of the finite-difference method [3, 6]. The pipeline is divided in N cells, each assumed to have a uniform pressure and an inflow and outflow. The finite-difference approximations used herein are (the dependence of P and w on t is omitted for the sake of notational simplicity) :

$$\begin{array}{l} \frac{\partial P}{\partial x} \approx \frac{P(i+1)-P(i)}{\Delta x} \\ \frac{\partial w}{\partial x} \approx \frac{w(i)-w(i-1)}{\Delta x} \end{array} \quad \text{for} \quad i = 2...N \tag{4}$$

In (4), Δx represents the cell length; P is the mean pressure within the i-th cell; and $w(i)$ and $w(i-1)$ are the inlet and the outlet flows of the i-th cell, respectively, and N is the number of cells considered. The discretization must be completed by boundary conditions; this is done using the `flange_in` and `flange_out` connectors already introduced:

$$\begin{array}{l} w(1) = \text{flange\_in.q} \\ w(N+1) = -\text{flange\_out.q} \\ p(1) = \text{flange\_in.p} \\ p(N+1) = \text{flange\_out.q} \end{array} \tag{5}$$

These boundary conditions allow to freely connect different elements without having to worry about causality of the elements, as one would have to do in a signal based modeling environment.
The distributed pipe is used to model both the manifold and the common rail. The manifold is simply modeled by a long distributed pipe; it is connected to

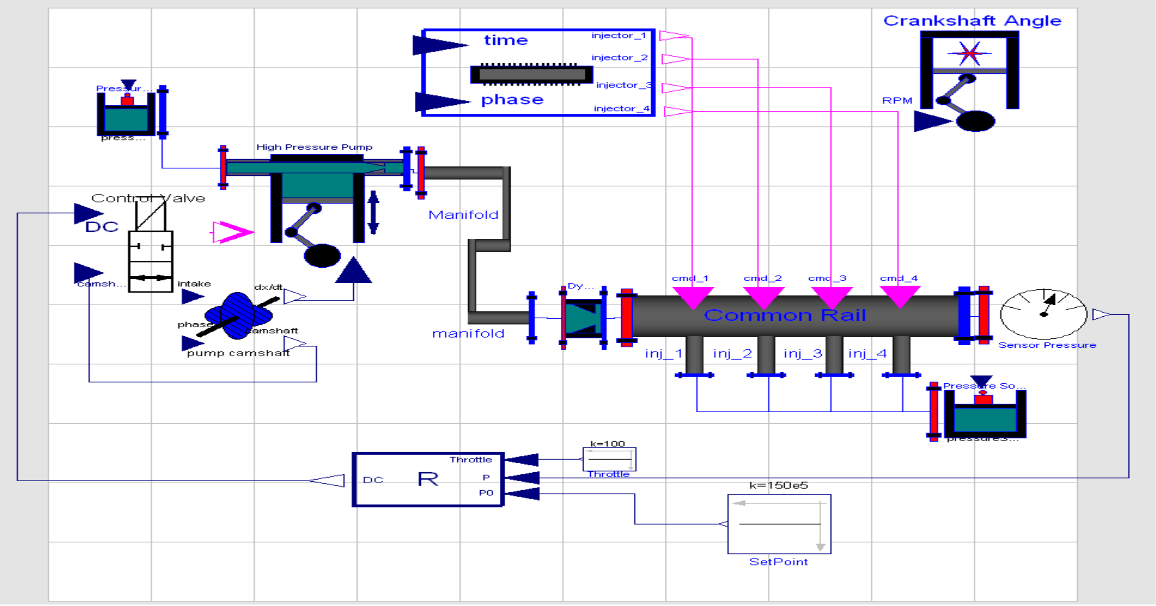

Figure 2: A graphical representation of the Injection System Model.

the high pressure pump on one side and to the orifice on the other. The modeling of the common rail is more interesting because it must be provided with the connectors used by the injectors. Fig. 3 is a graphical representation of the `CommonRail` model. It shows the model interface and its components. The model is composed of 5 distributed pipes and 4 injectors. The interface is represented by the two main flanges of the common rail, the 4 flanges of the injectors (which will be described later) and 4 logical signals representing the command signals to the injectors. The model encapsulates the geometric properties of the rail (length, diameter and inclination) and the two replaceable packages `Basefluid` and `BaseMaterial`. As shown in Fig. 1, in the final assembly, one of the main flanges of the common rail is connected to the diaphragm, the other to the pressure sensor. The injectors flanges are connected to a constant pressure representing the piston chambers.

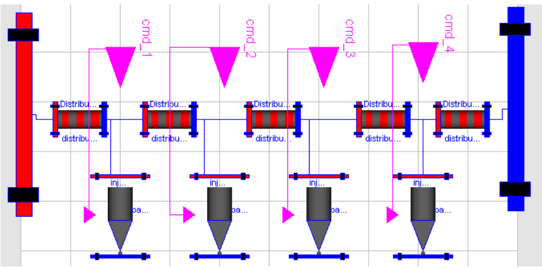

Figure 3: A graphical representation of common rail model.

3.2.2 Diaphragm

The diaphragm is a choking of the pipe. Its function is to increase the damping and by doing so it also achieves a certain amount of decoupling between the manifold and the rail dynamics. The narrowing is coaxial to the fluid flow; thus it can be modeled as concentrated load loss [11]. Having defined $\Delta p = P_1 - P_2$, the flow characteristic can be written as

$$w = \begin{cases} \rho\pi\frac{D_i^2}{4}\sqrt{\frac{2|\Delta p|/\rho}{1-(D_i/D_o)^4}} & \text{if} \quad \Delta p > 0 \\ -\rho\pi\frac{D_i^2}{4}\sqrt{\frac{2|\Delta p|/\rho}{1-(D_i/D_o)^4}} & \text{if} \quad \Delta p \leq 0 \end{cases} \tag{6}$$

where D_i is the diameter of the inlet and D_o the diameter of the choking.
The resulting model has two flanges; one is connected to the manifold and one to the common rail. It is important to notice that this element must manage flow inversion. Flow inversion happens when the pressure gradient between the two flanges changes sign. From equation (6), it is immediate to see that when ΔP approaches 0 the Jacobian of the function approaches singularity. In order to treat this case a linear junction has been implemented. When $|\Delta P| < \varepsilon$, the flow characteristic is approximated by a linear function.

3.2.3 Injectors

The injectors are one of the elements of the system most likely to be subject of study; in order to increase the configurability of the model, the partial model `BaseInjector` has been implemented. Its interface is composed of two flanges and of a boolean signal. In a standard configuration, one of the flanges is connected to the common rail and the other to the cylinder chamber.

For this study a simple injector model has been implemented. It is modeled as a valve whose opening request is assumed to be a two-valued variable (open-close). The dynamics of the orifice area is approximated by a first order filter with a pure delay. The relationship between the actual orifice area and the output fuel flow is assumed to be a non-dynamic relationship and modeled by a static customizable map depending on the pressure difference between the two flanges.

Notice that the cylinder chamber can be modeled as a constant pressure. Although the pressure in the chamber changes during a piston revolution; the high pressure in the common rail guarantees that the flow through the injectors is independent from the chamber pressure. It is also interesting to note that the injectors are not subject to flow inversion and thus their implementation is straightforward.

3.2.4 Engine Carrier

In engine control applications, it is a common practice to write all the control logic algorithms in term of the engine crankshaft angle. In order to meet this standard, a mechanism to relate the time independent variable (needed to simulate wave propagation) and the engine crankshaft angle independent variable (needed to simulate injection, spark, pumping and control) is necessary. The `EngineCarrier` model achieves this goal. The crankshaft is computed as a function of the instantaneous RPM of the motor (an user supplied variable). By defining an inner `EngineCarrier` instance in the model, the user can conveniently provide all the models with the needed independent variable.

3.2.5 High Pressure Pump and Control Valve

The high-pressure pump is one of the most important elements of the system; and, according to the object oriented paradigm, it has been defined as a partial model `BasePump`. The partial model only defines its interface (2 flanges, a command signal, and a real signal describing the velocity of the piston) and its main components (the replaceable packages fluid and material). This approach allows to easily define and implement different models of the pump.

For the goals of this study the compression dynamics inside the pump chamber is neglected along with the dynamics of the low pressure circuit. This is done because the chamber volume is negligible with respect to the volume of the rest of the system. As described in Section 2, the piston is driven by a camshaft and the instantaneous gasoline flow is determined by the velocity of the piston.

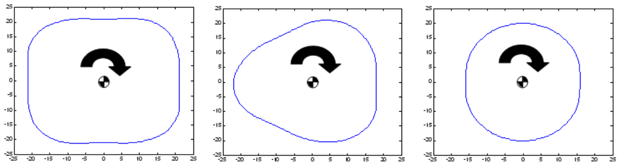

Figure 4: High pressure pump cam profiles (4-lobes, 3-lobes, 2-lobes).

$$
\begin{aligned}
&w_h = 0, \quad w_l = -\rho A_{pist}\frac{dh}{dt}, \quad \text{if control = open}\\
&w_h = \rho A_{pist}\frac{dh}{dt}, \quad w_l = 0, \quad \text{if control = closed}
\end{aligned}
$$

where w_l and w_h are, respectively, the flow through the flange connected to the low and high pressure circuit; A_{pist} is the area of the piston; h is the height of the piston and control is the state of the control valve.

The other two models needed in order to have a functional high pressure pump are the `PumpCamshaft` and the `ControlValve` models. These two models require the crankshaft angle provided by an `EngineCarrier` instance. `PumpCamshaft` computes the velocity of the piston as a function of the crankshaft angle; this is important because different high pressure pump camshaft profiles may be available, and the ability to easily switch between them is helpful. Three different cam profiles have been implemented: 2, 3, and 4 lobes. They are depicted in Fig. 4. The `ControlValve` model determines the state of the control valve. Its interface is defined by 2 input signals (DC and camshaft angle) and one boolean output signal which represents the state of the valve. As explained in Section 3.2.7 the DC signal is the control variable; it is the closing time of the valve expressed as a ratio to the whole high pressure pump period. The model can be extended to define more complex behaviors.

3.2.6 Injection logic

The `InjectorLogic` model generates the command signals for the injectors. It is possible to specify a time variant or constant injection time, in seconds, and an injection phase, in crankshaft degrees. It is therefore possible to simulate engine steady state or transient regimes. It requires the presence of an inner

`EngineCarrier` instance in the model. In the present implementation, which is based on real-world specifications, the four injection times are equal and sequentially delayed by 180°.

3.2.7 Control Algorithm

The goal of the common rail pressure controller is to regulate the pressure to a set point which is computed as a function of engine rpm and the throttle position. A known and steady pressure in the common rail allows to control the injected gasoline with precision by action of the injection time. This can be explained by the above considerations regarding the injectors; thanks to the great pressure difference in the rail and in the piston chambers, the amount of injected fuel depends mainly on the pressure in the rail and the injection time. If the pressure in the common rail is maintained constant without oscillations, then the injected fuel can be controlled by varying the injection time. Regulation of the gasoline pressure in the common rail is achieved in closed-loop, with an additional feed-forward component. The only measurement available for feedback is provided by the rail pressure sensor whose reading is sampled when the high pressure pump piston reaches its upper dead point (start of the expansion phase). The sampling time of this sensor hence is time-varying (whereas it is camshaft-angle-invariant). The control variable is the closing time of the valve measured as a Duty Cycle. The Duty Cycle (DC) is referred to the whole high pressure pump cycle, starting with the aspiration phase; therefore a DC of 50% means that the control variable is closed as soon as the compression phase begins, whereas a DC of 75% implies that the control valve is closed in the middle of the compression phase. Hence, this actuator implements a sort of Pulse-Width-Modulation (PWM). It is clear that, according to this control strategy, the aspiration phase is a pure delay. Also in this case, notice that the duty-cycle is time-varying but camshaft-angle-invariant.

The control algorithm is implemented in the `Controller` model. It has 3 inputs (pressure measurement, pressure set point and throttle position) and one output, the Duty Cycle, an `Controller` instance provides the variable for scheduling. Fig. 5 depicts the control system block diagram. The control algorithm is based on a Proportional + Integral control action whose parameters are scheduled on the engine speed. The feed-forward component allows to compensate for the injected fuel. The injected fuel can be seen as a load disturbance acting on the plant; although, technically, this disturbance cannot be measured, it can be estimated by knowing the gasoline pressure in the common rail and the injection time. The feed-forward term guarantees a faster disturbance rejection than the one achievable only by the closed loop term.

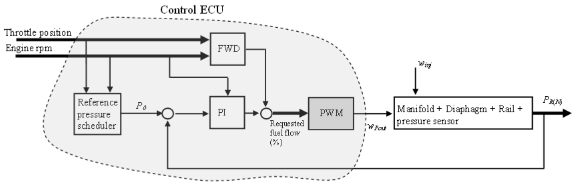

Figure 5: Control system block diagram.

4 A Case Study

The implemented model can be used for different kinds of sensitivity analysis. In this section the simulation results are presented and commented.

Before doing any sensitivity analysis, the fluid parameters have been identified and the model validated. The model is characterized by a large number of parameters. They belong to two categories: geometric and fluid-dynamic parameters. Geometric parameters are easily known; fluid dynamic parameters cannot be measured easily and inexpensively and therefore they were experimentally estimated. The parameters that have been estimated from data are: the gasoline bulk modulus and the Fanning friction factor of the manifold and of the rail. Note that, although the bulk modulus of the gasoline is known [11], it must be corrected to account for the elasticity of the pipes which is hard to be directly measured. All model uncertainties and simplifications are concentrated in the identified parameters. Fig. 6 shows the final validation results, by plotting the manifold and rail pressures. Pressures are normalized with respect to the set point pressure. The validation data is the result of a workbench test on a modern turbocharged direct ignition gasoline engine. It is important to note that only closed loop tests are available for validation; therefore all the following considerations are to be referred to the closed loop case. The results show that the model is able to accurately replicate the main resonances and damping. The relevant dynamics are correctly captured. Spectral analysis confirms that the frequency range of validity of the model is approximately 0-1000Hz. It can be seen from figure that the model is able to reproduce higher frequency dynamics, but the fitting between the simulated and the measured data is not as

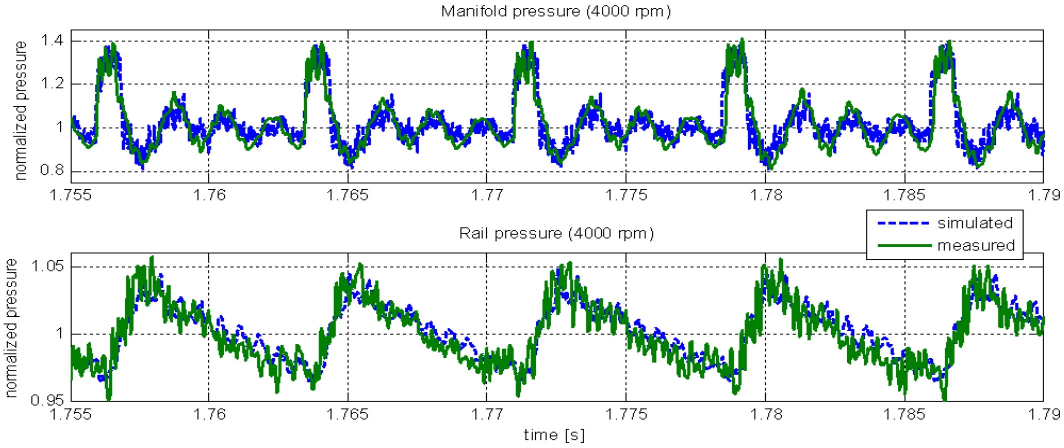

Figure 6: Simulated and measured pressures in the manifold and in the rail (time-domain; 4000rpm).

good as with lower frequency dynamics. In order to improve the model accuracy beyond 1KHz, the Fanning friction coefficient dependence on the Reynolds number needs to be accounted for. Although the identification of such dependence requires considerable efforts; the model can be easily modified to account for that dependence.

This validation test also shows that the prototype suffers from high pressure peaks. In fact, the pressure in the manifold raises to levels up to 1.4 times the nominal pressure. The final system will be equipped with an exhaust valve set to open at about 1.35 time the nominal pressure. The opening of the safety valve is to be avoided.

This problem is well suited to be studied with the described model. The dependence of the pressure peaks has been studied as a function of the diameter of the manifold. Results are show in Fig. 7, where pressures are normalized with respect to the safety valve threshold. Thanks to this analysis, it is possible to draw some interesting guidelines for the design of the manifold:

- the pressure peak problem is sensitive to the engine speed. The higher the engine speed is the higher the pressure peak is;
- the maximum pressure depends hyberbolically on the diameter of the manifold;
- an increase of the manifold diameter of 2mm with respect to the baseline manifold solves the problem.

This is an example of the kind of possible sensitivity analyses that can be run using the model. Other elements that can be easily studied are: the radius of the rail, the number of lobes of the high pressure pump and the static characteristic of the injectors.

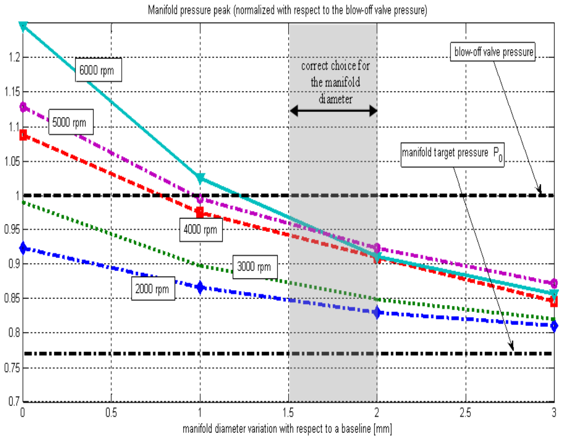

Figure 7: Manifold maximum pressure as a function of the manifold diameter increment for different engine speeds.

5 Conclusions

In this work an object oriented model of a Common Rail Injection System for a gasoline engine has been proposed. The focus has been on providing an easily reconfigurable simulation tool that can help the design process. The system has been described in detail and a 1-Dimensional model of the pressure waves propagation has been derived from fluid dynamics basic principles. The simulator has been validated using bench tests. A case of study in which the model is used to study the dependence of the maximum pressures reached in the manifold as a function of the manifold diameter has been presented. The case of study shows that the proposed model is accurate enough to help co-design of the system, where mechanics, fluid dynamics and control logic are looked at as an ensemble. These kind of analyses allowed to draw useful guidelines for the design of such a complex system.

6 Acknowledgments

The Authors would like to thank Emilio Comignaghi, Marco Sofia (from FIAT Powertrain srl), Antonio Palma and Eduardo Sepe (from Élasis SCpa) for the insightful discussions and valuable suggestions.

References

[1] Kouremenos D. A. and Hountalas D. T. Development and validation of a detailed fuel injection system simulation model for diesel engines. *SAE Technical Papers*, 01(0527), 1999.

[2] Modelica Association. Modelica - a unified object-oriented language for physical system modelling. language specification. technical report, 2002.

[3] Bertin D., S. Bittanti, and S.M. Savaresi. The decoupled cushion control in ride control systems for air cushion catamarans. *Control Engineering Practice*, 08, 2000.

[4] Alabastri E., L. Magni, S. Ozioso, R. Scattolini, C. Siviero, and A. Zambelli. Modeling, analysis and simulation of a gasoline direct injection system. In *Proceedings of the IFirst IFAC Symposium on Advances in Automotive Control*, 2004.

[5] D'Errico G. and Onorati A. An integrated simulation model for the prediction of gdi engine cylinder emissions and exhaust after-treatment system performance. In *SAE Int. Congress & Exp. Detroit*, 2004.

[6] Strikwerda J. *Finite Difference Schemes and PDE*. Wadsworth Brooks/Cole., 1980.

[7] Chiavola O. and Giulianelli P. Modeling and simulation of common rail systems. *SAE Technical Papers*, 01(199), 1999.

[8] Official Journal L 044, 16/02/2000. *Directive 1999/96/EC of the European Parliament and of the Council.*

[9] Baker P. and Watson H. Mpi air/fuel mixing gaseous and liquid plg. *SAE Technical Papers*, 01(246), 2005.

[10] Covington J. P. Modernizing the fixed-venturi carburetor. *Automotive Engineering*, 82(7), 1974.

[11] Thomas P. *Simulation of Industrial Processes for Control Engineers*. Butterworh Heinemann, Oxford, England., 1999.

[12] Li J. Z., Treusch C., Honel B., and Neyrat S. Simulation of pressure pulsations in a gasoline injection system and development of an effective camping technology. *SAE Technical Papers*, 01(1149), 2005.