

Model-Based Powertrain Control: Many Uses, No Abuse

A. Dauron

Renault, Centre Technique de Lardy, 1 allée Cornuel, 91510 Lardy - France
e-mail: alain.dauron@renault.com

Résumé — Contrôle moteur « Model-based » : à consommer sans modération ! — Derrière l'expression « model-based » se cachent de nombreuses acceptions, comme nous l'a confirmé une étude bibliographique simplifiée. Les « Modèles » décrivent soit le Groupe Moto-Propulseur, soit son système de contrôle. À côté de ce premier axe de diversité, on constate aussi des usages très variés, depuis des activités de conception initiale jusqu'à la validation finale d'un logiciel, en passant par les tests « Model In the Loop » et « Software In the Loop », la Mise au Point Assistée par Ordinateur, ... Nous illustrons les principales acceptions par plusieurs travaux menés chez Renault. Il apparaît clairement que l'intégration de modèles dans le produit et dans les méthodes d'ingénierie est de première importance pour maîtriser la complexité croissante des systèmes de Contrôle du Groupe Moto-Propulseur et les technologies à venir : la convergence de ces démarches locales conditionne notre capacité à concevoir et valider les groupes moto-propulseurs comme de véritables systèmes mécatroniques.

Abstract — Model-Based Powertrain Control: Many Uses, No abuse — Many meanings exist behind this expression. We have conducted a mini-survey to confirm and describe this diversity. "Models" mainly stands for powertrain-models or Engine Management System (EMS) models. Uses are also very diverse, from upstream design activity to final software validation, through Model In the Loop and Software In the Loop testing, Computer aided tuning, ... We propose illustrations of the main meanings through work conducted at Renault. It clearly appears that putting models in products and in methods is of great importance to master future technologies and increasing complexity of powertrain control area: the target is to mix both worlds, towards mechatronic systems.

INTRODUCTION

It is a commonplace in Powertrain Engineering to state that powertrain control is of more and more importance in providing customers with vehicles that fulfill drastically increasing requirements. As a result, Engine Management System (EMS) complexity exhibits rapid growth. A direct consequence is that engineers need better methods and tools to develop efficient & robust engine control in an efficient & robust way. “Model-based” is often presented as the best way to reach efficiency & robustness. But depending on the speaker, many meanings can be hidden behind the expression and sometimes misunderstanding can occur.

We shall give in this paper more insight on what is behind the expression, and illustrate the main meanings by classical or recent works.

1 MANY USES...

In order to confirm and render somewhat explicit the qualitative feeling that the term “model-based” is clearly not enough to guess the content, we conducted a (limited) bibliography survey with main keywords “model” “based” “engine” “control”.

TABLE 1
Results of mini “model-based” survey

Model embedded in production software or model in development process	explicitly embedded		9
	only process		10
Embedded algorithm design	control	5	10
	diagnostic	2	
	estimator	3	
Computer aided tuning			2
EMS or engine optimum design			1
Engine or PT* model type	physical	8	21
	non physical	13	
Engine, PT* model for algorithm or software simulation	non real-time	3	7
	real-time (HIL*)	4	
EMS* or software model	software	8	10
	System	2	

* PT: Powertrain, EMS: Engine Management System, HIL: Hardware In the Loop.

After minimal filtering, limited to eliminate irrelevant papers, such as “... model based on the principle of...”, we retained 19 references [1-19], dated from 2002 to 2006. Splitting of this small sample is summarized in Table 1.

Note that criteria are not completely exclusive, so the sum of numbers is greater than 19: for instance, some papers deal

with both innovative algorithms and tool-chain, so are counted both in “Embedded algorithm design” and in “Engine, PT Model for algorithm or SW simulation”. Similarly, some papers are surveys, and point to different model types (physical & non-physical).

The main results concerning type of models and type of activities described in publications are listed below.

Two main “objects” are modeled: Engine or Powertrain (“the plant”) and control software or EMS (“the controller”).

For plant models, main uses are:

- algorithm development (design and simulation), in which case algorithm and related sensors/actuators are of course also modeled),
- software or EMS simulation,
- computer aided tuning,
- engine + system optimum design (only one case).

For controller models, main uses concern control systems (ECU – Electronic Control Unit, including hardware & software–, sensors and actuators) or only software development process support through both formal approaches and tool-chains descriptions. The scope is mainly focused on software, sometimes partially enriched with control system perimeter items (ECU and rarely sensors and actuators). Objectives include formalizing, linking and/or equipping with tools:

- different abstraction levels, from global functional to detailed logical & technical system architectures,
- different tasks of development process, from capture of requirements to software module re-use,
- different software components (or much less frequently: different sub-systems).

Concerning model-based **algorithms design**, the vast majority uses explicit embedded models (IMC: Internal Model Control, explicit adaptive control, MPC: Model Predictive Control, etc.), rather than controllers indirectly designed from the plant model. We shall later illustrate both design axes.

Concerning engine model categories, **physical models** are principally “0D” (zero dimensional), including dynamics and main non-linearities, and **non physical models** are of many types:

- black-box linear dynamics,
- static mappings,
- neural nets (with or without dynamics),
- statistics without dynamics.

Considering the small number of papers examined, quantitative interpretation is not worthwhile, but spread of “model-based” meaning is confirmed. Two simple questions allow identification of the concerned area:

- What is modeled?
- What are the uses of the model?

Those questions will supply the organization for this paper.

In Section 2, we shall look at some work using “controller models” (software or control system).

In Section 3, the main part of the paper, we shall examine work based on “Plant” models (Engine or Powertrain), and some mixed or less classical approaches.

Before concluding, Section 4 will propose some general remarks and perspectives.

2 MODEL = SOFTWARE OR CONTROL SYSTEM MODEL

In Computer Engineering areas, model-based expressions often take software or the electronic system (or their abstract representations) as the object of modeling activity.

The need for having formal software or system description (“model”) is directly linked to the drastic complexity increase we face in engine control, as stated in the Introduction. We show in Figure 1 the evolution of software + calibration size from the beginning of Diesel engine control up to 2008. System complexity, in terms of number of sensors and actuators, size of ECU connector, microcontroller capacities, etc. follows the same macroscopic evolution. This evolution calls for major changes, in both product and development process and in the associated tools.

In the mini-survey we conducted, paper contents range from theoretical/formal concepts (*e.g.* [16]) up to mainly practical tool-chain descriptions (*e.g.* [12]), but the majority of papers include both process/abstraction levels/formal architecture description, and tools supporting them. As we pointed out above, very few papers deal with the complete system perimeter. A large majority is concentrated on software, even if the general formalism described could be applied to software + ECU + sensors + actuators, or even to complete engine + EMS.

We have at Renault an important internal program whose objective is to make the necessary changes in EMS, both in terms of product (functional and software architecture) and in terms of process (system and software development and validation). One can find in [23] a description of our functional and software architecture, and of some associated process and tools.

The main change decided for our EMS product definition is the evolution towards modular architecture, along three main dimensions:

- functional grouping with 3 levels of granularity (functions, sub-functions, modules),
- description of functional layers splitting specifications between **sensors & actuators interface**, including local control and diagnostics (*e.g.*: air-flow sensor signal processing, throttle actuator position control), **sub-system control** (*e.g.*: air control) and **whole_engine control** (*e.g.*: torque set-point calculation),

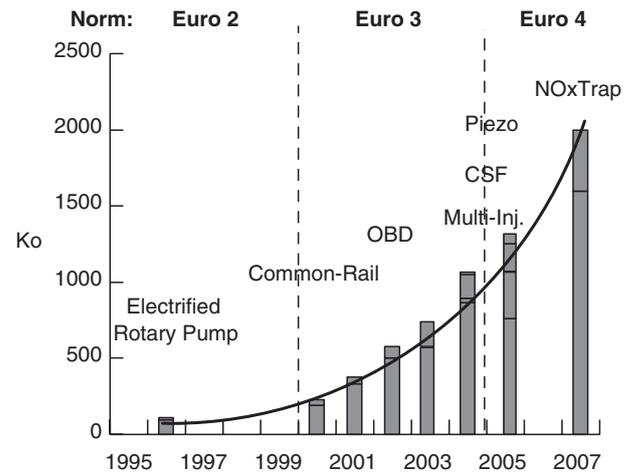


Figure 1

Diesel engine control software + calibration size evolution (Kbytes).

- software modularity dynamic mechanisms describing the way functional requirements concerning real-time, data transfer between module, etc. are implemented in C code.

This formalism improves two important characteristics: confinement and re-use:

Confinement: functional modular architecture brings better de-coupling between each module. It allows developing and modifying modules more independently, as functions interact less with each other.

Re-use: the ability to define standard modules (up to C code) and to use them in different software codes and on different EMS platforms, for a list of pre-defined compatible configurations. (re-use is also a goal for *Autosar* program, see [31], even if *Autosar* scope is much wider as it considers whole vehicle electronics networks, so in terms of re-usability prepares adequate abstraction mechanism to place a module on any vehicle ECU).

These two characteristics have direct positive effect on:

Quality: We expect easier and more complete validations for local modifications as they are designed to have less “global effects” thanks to confinement. As re-use will induce less “re-coding” of specifications, we shall suffer less coding errors.

Cost: Stronger upstream processes including architecture impact analysis for a new function development will help in defining directly good specifications with less code correction loops, and re-use will avoid re-coding the same module for different platforms.

Time to market: validation time without modular architecture would grow exponentially as complexity increases: modular architecture and associated process and tools will help maintaining reasonable durations, and allow partly parallel module validations.

In parallel to this modular architecture, we are working on many process improvements and associated tools for robust EMS development. One is described in [23]: Model In the Loop (MIL) and Software In the Loop (SIL) validations. The idea consists in developing the module test environment in the upstream phase, in parallel with algorithm design. Tests are developed to fit with use-cases and requirements. Specification (“Model”, written in Simulink – see [23] – environment) is tested in this environment (“Loop”). Depending on module type, the test environment may represent either a stimuli generator, or a real loop including some part of an engine model. Then, later in the process, substituting software code to specification allows us to confirm that the written code reacts as specified.

Other important methods and tools are described in [34]. They deal with Safety, Availability, Maintainability of whole EMS: a process is proposed, and a system simulation is described, covering both functional and dysfunctional behavior, and simulating the system either at functional or at organic level. Of course, this implies having formalized functional and dysfunctional behavior specifications, for the entire system and for a cascade of embedded sub-systems, which is not usual in EMS development. This is the first step towards efficient and robust system development with modern supporting tools.

3 MODEL = ENGINE OR POWERTRAIN MODEL

As already mentioned in Chapter 1, in “model-based” control, engine/powertrain models serves for **algorithm design**, for **algorithm, software** or **EMS simulation**, for **aided tuning** and for **engine/powertrain + system optimum design**. We shall follow this list in next few paragraphs.

3.1 Algorithm Design

The mini-survey in Section 1 tends to show that majority of model-based algorithms encountered consist in embedding directly or almost directly a simplified explicit model of an adequate part of the engine in control software. Going through different algorithm types and model uses, we shall propose some explanations for this tendency, and some illustrations from work conducted at Renault.

First, estimators and diagnostic algorithms often integrate **explicit models** (see [3, 9, 14, 2, 11]). One example will help to illustrate on-board estimators: it deals with a valuable way to reduce Diesel EMS cost. Classically, turbo diesel engine EMS use both airflow-meter and manifold pressure sensors. The first is mainly used for Exhaust Gas Recirculation (EGR) precise closed-loop control and Air/Fuel Ratio (AFR) open-loop limitation, whereas the second is more dedicated to closed-loop boost control. We have worked at Renault to

develop a dynamic air-flow estimator, that replaces the expensive sensor. The solution (see [20]), using a simplified embedded model, is already used in production for our 1.5l engine, in both Euro 3 and Euro 4 versions.

Second, a large proportion of engine control algorithms are open-loop feedforward. Historical tuning maps are replaced by **models or inverse models**. Even if some feedback is used, feedforward is often present to compensate for system and feedback dynamics with varying set-points, or to take into account measured disturbances and compensate (statically or dynamically) for their effects on desired states. This compensation includes as much knowledge as possible, often through an inverse model. In [38], we find such feedforward on measured disturbance. This work, conducted at Renault, deals with Diesel Particulate Filter (DPF) regeneration control. On such Diesel engines, periodic regeneration is needed to eliminate accumulated soot, through increase of DPT inlet gas temperature during several minutes. This is obtained by switching to special engine working points with high exhaust temperature, combined with a closed-loop control of DPF inlet temperature. This closed-loop uses HC emission as an input variable, in order to drive exothermic combustion in Diesel Oxidation Catalyst (DOC). But DOC thermal inertia is high and prevents rapid rejection of disturbances, whereas Input DOC temperature has a large influence and is easily measured. We therefore built a feedforward on this measured disturbance, using two models, one linking disturbance to output and another linking input to output. Results shown in [38] are promising.

Third, some control algorithm design methods lead to **explicit model** embedding into the control algorithm (IMC: see [17], MPC: see [5]). Another example is found in [26] where we describe some work on AFR control in gasoline engines with an AFR linear sensor. The main characteristic of dynamics between injected fuel mass and measured AFR is a greatly variable time delay, due to engine combustion cycle and gas transport in the exhaust line. We propose to use some variant of a simple controller with the “Smith Predictor” (see *Fig. 2* for standard Smith predictor) combined with online model identification.

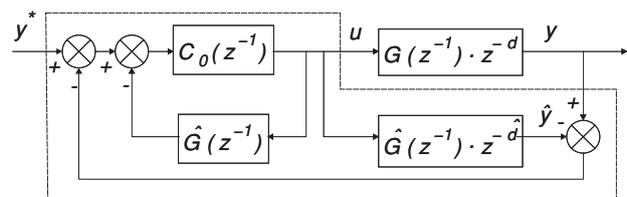


Figure 2

Smith Controller architecture (usual notations).

Smith Predictor Controller is dedicated to control systems with important time delay. It uses two versions of the system model: one with modeled delay: $\hat{G}(z^{-1}) \cdot z^{-\hat{d}}$ and the other with no delay at all, on which some closed-loop design is applied, with $C_0(z^{-1})$. As delay and system zeros vary according to engine working point, controller structure is adapted closer to IMC, and the “no-delay model” is replaced by a simplified version with no zeros, the same poles and same static gain. Recursive Least Squares are used to identify on-line model parameters: the adaptive method is both explicit (the model is explicitly identified) and almost direct (controller parameters are simple functions of model parameters), see [30]. Such AFR Control proved efficient in upstream development, even in simplified versions, but value/cost ratio is still lower than for a classical binary sensor and its associated “limit cycle” PI control.

We see many examples of algorithms with explicitly embedded models, but the majority of design methods described in Automatic Control literature use model structure and parameters as inputs, and deliver as outputs control algorithm structure and parameters through non-trivial calculations. These calculations are driven by performance/robustness specifications such as expected dynamics for closed-loop (pole placement method, etc.), criteria weights (LQG, etc.). We shall illustrate such methods by a simple Single Input Single Output (SISO) example, extracted from components control activities. Many “air actuators” such as Gasoline and Diesel throttle valves, EGR valves, use DC motors + gears as basic actuators. This technology is appreciated for robustness, high torque and widespread standard status, inducing limited costs. That is why we decided to maximize standardization in software drivers for such components.

For the position control algorithm, analysis of performance needs (time response, no overshoot, zero offset), but also of the variety of components to be treated with minimum engineering effort, of limits of previous methods (mainly PID + heuristic complements), and of system complexity (transfer function order) led to choosing a classical pole placement method (see for instance [22]). This is clearly an indirect method, using, for SISO dynamic linear systems, the resolution of a Bezout polynomial equation to calculate R , S and T polynomials shown in Figure 3 with usual meaning for u , y and y^* .

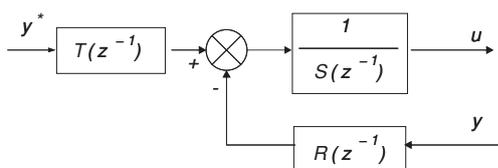


Figure 3

Pole Placement RST controller structure.

Qualitatively, the few more degrees of freedom even in low order RST allows a better compromise than PID between rise-time and overshoot, by taking into account system zeros.

A complete environment was built with Matlab-Simulink including:

- a simulator using a physical model created from past experience, with main non-linearities (saturations, friction, spring(s) kinematics), with parameters directly coming from a components database built with actuators suppliers information (inertia, motor constants, etc.),
- procedures to assist construction of linear models from the same actuator parameters, to specify expected performances and implementation parameters such as sampling time, to solve the Bezout equation, to evaluate robustness through classical gain & phase margins, plus time domain simulations for nominal and “worst-cases“, and if necessary to iterate to obtain a good performance/robustness compromise.

This complete computer aided environment for design, tuning and validation is now operational and gives satisfactory results, improving both closed-loop performances and engineering efficiency (straightforward tuning method and less components real-world testing). Beyond simple algorithm design, it gives an adequate introduction to the next paragraphs dedicated to other model uses.

3.2 Other Engine or Powertrain Model Uses

3.2.1 Algorithms Simulation

For every algorithm development activity, simulation is a classical way to obtain early evaluation, at least of some qualitative viability, up to precise performance assessment, depending on model accuracy. The above example described such a process and tool.

To be of some interest for algorithm development itself, the simulator must implement a more detailed description than the one used to design the algorithm. We find for instance in [39] a detailed description of a tool based on Simulink implementing 0D Diesel engine model.

Validation of algorithm integration in global engine control is another important task, on which we shall say some words later, as it is related both to engine modeling and also Control applicative software modeling.

3.2.2 SW & EMS Simulation

When complete software needs to be tested on a production platform, the usual solution is to use Hardware In the Loop (HIL) testing. Renault HIL solution for EMS is described in [25]. Simulink real-time engine and Automatic Transmission (AT) models runs on dSpace hardware (see [28]), and are functionally connected to the ECU through hardware sensors and actuators emulators, so that engine ECU and AT ECU software codes behave as if they were controlling a real

powertrain. Many tests are already performed in an automated way, and extensions are under study, in order to take full advantage of such a tool, that allows control of the complete software (applicative and basic) and also software-hardware interface, while MIL and SIL methods concentrate more on single module (or module groups) of applicative software.

3.2.3 Computer Aided Tuning

The tuning task follows the same evolution as software functions in terms of complexity increase. We saw in paragraph 3.1 an example of computer aided tuning using polynomial calculation with RST controller for a DC Motor.

More often, aided tuning uses optimization methods to minimize some performance index, (weighted sum of individual criteria) with some constraints on control parameters, states, or other criteria. For a long time, the increasing number of degrees of freedom (see *Fig. 4*) made it mandatory to identify models and then optimize criteria, rather than doing test and trial or direct local optimization on the Engine Bench. (see [7] and [10]).

We see in [24] an approach developed at Renault that combines global black-box rather than local modeling, and online adaptive Design of Experiment (DOE). The optimization calibration process, and the improvement of learning zone and of model accuracy, are iteratively mixed to minimize engine bench use while improving the optimization result.

3.2.4 Optimum Design

System optimum design is less frequent than tuning parameter optimization as described in the previous paragraph. We

see in [10] a first example where the same model is used to evaluate both some tuning parameter evolutions (transient engine speed set point) and some design variants (intake air flap position relative to compressor of an additional turbocharger) in a sequential turbocharger architecture for marine engine.

Closer to automotive needs, [40] describes an Infinitely Variable Transmission (“power-split” type with electric variator), and its evolution into a hybrid powertrain with high electric energy storage addition. The chosen object is the result of a systematic exploration conducted at Renault, of all possible architectures, including variator technologies, one or two modes, etc. This systematic evaluation was conducted thanks to a physical first order model simulator, taking into account necessary phenomena, without any expensive and lengthy prototype comparison phase. Choice criteria such as transmission efficiency, and mechanical + system constraints allowed choice of physical architecture and of some control system requirements (for instance to implement smooth and rapid mode change). Further on-going vehicle prototype testing confirms the potential for this architecture.

3.2.5 Combined Approach: Powertrain + Software Modeling and Simulation

Some examples already cited were in fact using models from both “worlds”, coming from “plant” and from “controller” sides. Environments combining a complete software model and a complete engine model are however much rarer. A HIL simulator facing an ECU which hosts a modular software, as described in Section 2, should be a first way to get a global engine + EMS simulator.

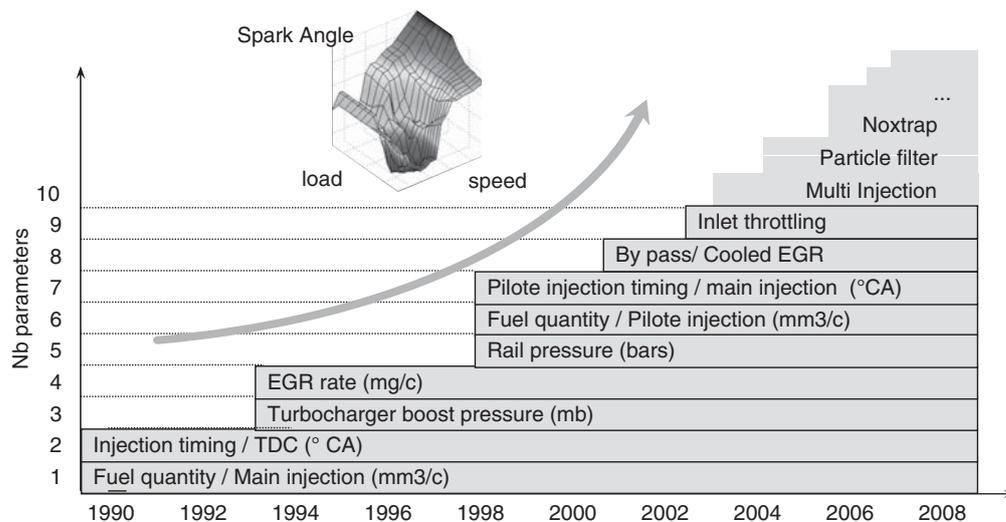


Figure 4

Number of degrees of freedom (from [24]).

Another example is described in [37]. It presents a platform for Automated Transmission (AT) integration, consisting of modular description of both AT control software and the AT model itself. It includes also some simplified Engine+EMS, vehicle and driver models. Different complexity levels exist for the AT model. Although the tool was mainly dedicated to algorithm validation and Software integration, it gives first sight of what could be a simulation tool for development of mixed “mechatronic” sub-systems, or even complete AT + ATMS or Engine + EMS. Common or compatible handling of both software and physical objects, through architecture configuration management at different aggregation levels, should be in the longer term a characteristic target for the CAE tool allowing unified development of the powertrain and its electronic management system.

3.3 ... No Abuse

We have seen many types of models, and many uses for these different types. Some examples still could not enter the above analysis, and I would like to emphasize some positive side-effects of the “quest for model” that is part of the Automatic control engineer initial work.

There is a long history at Renault in trying to get physical knowledge about the system to control (engine, after-treatment device, specific component, ...), in order to develop or tune control algorithms (or estimators, or diagnostics). See for instance [26, 21, 32, 28].

Depending on the context, in particular the existence or not of available physical models with the right complexity level, model reduction is sometimes necessary, and sometimes “by-products” appear, not directly linked to engine control or tuning. For example, in [20], simple models for three-way catalysts (TWC) had to be derived from complex 3D model of physical & chemical phenomena. This led to:

- very simple models (state dimension 1 or 2) capturing mainly non-linear Oxygen storage dynamics, that we could use in on-board control or diagnostics,
- 0D or 1D TWC model, precise enough to be included in a simulation environment dedicated to exhaust line optimum design, including engine control strategies for light-off duration reduction. This tool has since been used at Renault, allowing rather cost effective technical definitions of the “after-treatment system”.

The same interdisciplinary approach with multiple “modeling products” currently takes place, for newer technology (see for example [36] for HCCI reduced combustion models, and [27] for fuel cell modeling, control and fault detection).

4 LIMITS & PERSPECTIVES

Of course, some limits, or “good/bad choices”, exist that can reduce in **some** cases the interest of **some** model-based approaches. Two examples:

As many model types exist, some will be more or less relevant depending of expected use. It is beyond the scope of this paper to propose precise rules. A general statement would be to use physics whenever available at the right level of accuracy and complexity, either as final equations or as structuring guide-lines for a “grey-box” model that should need more accuracy at the expense of more calibration effort.

EMS architecture alternatives may sometimes make “model based” less useful, when comparing global system performance and engineering effort. For example, sometimes an efficient closed loop based on a relevant sensor and a simple model, with limited feed-forward effort, will be more suitable in terms of EMS cost, control accuracy and development + tuning workload, than a detailed open loop control involving a precise inverse-model and many disturbance measurement sensors.

Nevertheless, “model-based” has proved definitely positive in all the examples discussed here. Considering those multiple results and on-going works, are we close to asymptotic use of models in powertrain control system developments?

No, for at least two reasons:

First, as Vehicle requirements (emissions, fuel economy, performance, etc.) always pull heavy technological innovations, modeling activity is always struggling to get the right equations before new solutions becomes old or even obsolete! That is why, even in very upstream technology assessment, modeling activity is important, as it will help every “métier” to enter quicker any new technical domain, and develop better products with more efficiency. Even if no technological change occurs on the “Plant side”, the introduction of a control system can justify new modeling effort because some characteristics (dynamics, ...) just become mandatory to understand and compensate. An example for this is clutch friction characteristics, both in terms of temperature and force dynamics. These phenomena first became of interest when Automated Manual Transmissions appeared, substituting for the driver a control system with various sensing, computing and actuating characteristics.

Second, even when a “Plant model” exists, we have illustrated the common statement that the control system is becoming more and more complex, so structured approaches using “Controller models” are needed to keep quality at the right level and reduce development costs and time to market. We have seen that things are just beginning, and mainly for the software part of EMS. For full EMS, and even more for mechatronic “controller-plant” integrated development, we are still far from the status of other industries such as Aerospace, and revolution is still in front of us.

CONCLUSION

We have seen that many meanings can exist behind the two words “model-based”, even when limited to an engine

control perimeter. Two main perimeters are modeled: powertrain and control system. Activities range from mechatronic system optimum design up to real-time software architecture definition, through control algorithm design and tuning, software and system validation. We have given some illustrations in both domains, and hope that the importance of the activities described was clearly demonstrated. When considering both product design and development process, the ultimate aim of “model-based” is the result of combining the two domains: plant + controller. This target has many stakes. In terms of product optimization, it will help to achieve maximum mechatronic integration inducing better value to cost ratio. In terms of productivity, mechatronic CAE environments will be the key for inter-disciplinary efficient work. In terms of quality, a formal mechatronic system description will bring the ability to describe functional and dysfunctional requirements, to distribute them to a hierarchy of sub-systems, and to support this hierarchy of objects during the whole process of development design and validation in a safe and complete way.

ACKNOWLEDGEMENTS

I would like to thank all Renault engineers and authors cited in the paper, whose publications illustrate so well our ambitions in EMS. Thanks also to Michel Sorine, from INRIA, long term partner of these ambitions and director of many upstream studies that were cited.

REFERENCES

Papers from “model-based” mini-survey:

- 1 Anders, J.W. and Franchek, M.A. (2005) Nonlinear adaptive engine speed control using an instrumental variables approach and a truncated volterra series. *Proceedings of IMECE*.
- 2 Antory, D. (2005) Application of a data-driven monitoring technique to diagnose air leaks in an automotive diesel engine: A Case study. *Mech. Syst. Signal Process*, **21**, 2, 795-808.
- 3 Arsie, I., Pianese, C. and Sorrentino, M. (2006) A procedure to enhance identification of recurrent neural networks for simulating air-fuel ratio dynamics in SI engines. *Eng. Appl. Artificial Intelligence*, **19**, 1, 65-77.
- 4 Bauer, A., Broy, M., Romberg, J., Schätz, B., Braun, P., Freund, U., Mata, N., Sandner, R. and Ziegenbein, D. (2005) AutoMoDe-Notations, Methods and Tools for Model-based development of automotive software. *SAE paper No. 05AE-268*, in Proceedings of the 2005 SAE World Congress, Detroit, MI, April 2005, Society of Automotive Engineers.
- 5 Bengtsson, J., Strandh, P., Johansson, R., Tunestal, P. and Johansson, B. (2006) Hybrid control of homogeneous charge compression ignition (HCCI) engine dynamics. *Int. J. Control*, **79**, 5, 422-448.
- 6 Cipollone, R. and Sughayyer, M. (2003) *Transient air/fuel ratio control in si engines*, The Institute of measurement and control.
- 7 Hafner, M. and Isermann, R. (2003) Multiobjective optimization of feedforward control maps in engine management systems towards low consumption and low emissions. *ICEF. Trans. Inst. Meas. Control*, **25**, 57-74.
- 8 Jeutter, R. and Heppner, B. (2004) Model-based system Development – Is it the solution to control the expanding system complexity in the vehicle. *SAE World Congress*.
- 9 Johnsson, R. (2005) Cylinder pressure reconstruction based on complex radial basis function networks from vibration and speed signals. *Mech. Syst. Signal Process*, **20**, 8, 1923-1940.
- 10 Kech, J.M. and Klotz, H. (2002) Model-based sequential turbocharging optimization for series 8000 M70/M90 engines, *SAE World Congress*.
- 11 Kimmich, F., Schwarte, A. and Isermann, R. (2005) Fault detection for modern diesel engines using signal-and process model-based methods. *Control Eng. Pract.*, **13**, 2, 189-203.
- 12 Jong, Min Choi, Il Sang, Yoo, Young, Won Park and Joong, Yong Park (2002) Model-based architecting of reliability, availability and maintainability applications. *SAE World Congress*.
- 13 Mutz, M., Huhn, M., Goltz, U. and Krömke, C. (2003) Model based system development in automotive. *SAE World Congress*.
- 14 Park, S., Yoo, H., Park, Y., Lee, S., Park, S., Kim, S. and Lee, K. (2006) Model-based development of engine control system from concept to vehicle. *SAE World Congress*.
- 15 Ranville, S. (2002) Practical application of model-based software design for automotive. *SAE World Congress*.
- 16 Rappl, M., Braun, P., Von Der Beeck, M. and Schröder, C. (2002) Automotive software development: A model based approach. *SAE World Congress*.
- 17 Schwarzmann, D., Nitsche, R. and Lunze, J. (2006) Diesel boost pressure control using Flatnes-based internal model control. *SAE World Congress*.
- 18 Shaver, G.M., Roelle, M.J. and Gerdes, J.C. (2006) Modeling cycle-to-cycle dynamics and mode transition in HCCI engines with variable valve actuation. *Control Eng. Pract.*, **14**, 3, 213-222.
- 19 Turin, R., Mills, J., Fowlern, S. and Myers, C. (2005) Enabling virtual development for practical engine control. *SAE World Congress*.

OTHER REFERENCES:

- 20 Abida, J. (2000) Brevet FR2789731 : Procédé de détermination du débit d'air entrant dans un moteur à combustion interne équipé d'un circuit de recirculation des gaz d'échappement.
- 21 Aimard, F. (1995) Modélisation de pots catalytiques trifonctionnels. Utilisation en contrôle moteur et diagnostic. *PhD thesis*, Université Paris IX Dauphine.
- 22 Astrom, K.J. and Wittenmark, B. (1984) *Computer Controlled Systems, Theory and Design*, Prentice Hall.
- 23 Boutin, S. (2005) A modular architecture for embedded applicative software. *VDI Congress*, Baden.
- 24 Buzy, A., Cotte, P. and Visconti, J. (2005) Online Adaptive DOEs and global modelling for Diesel engines and DPF calibration. *IAV Berlin Conference: Design of Experiments in Engine Development*.
- 25 Colin, H. (2004) Renault's Universal HIL for Powertrain. *Conférence utilisateurs dSPACE*.
- 26 Dauron, A. (1991) Modélisation et commande d'un moteur à combustion interne : application à la régulation de richesse. *PhD Thesis*, Université Paris IX Dauphine.

- 27 Di Penta, D., Bencherif, K. and Sorine, M. (2005) A Reduced Fuel Cell Stack Model for Control and Fault Diagnosis. *ASME European Fuel Cell Technology and Applications Conference*.
- 28 Dspace: <http://www.dspaceinc.com>
- 29 Emery, P. (2002) Modélisation simplifiée des moteurs essence à injection directe par réduction des modèles physiques multidimensionnels. *PhD Thesis*, Université de Paris VI.
- 30 Goodwin, G.C. and Sin, K.S. (1984) *Adaptive, filtering and Control*, Prentice Hall.
- 31 Heinecke, H. et al. (2006) AUTOSAR – Current results and preparations for exploitation. *7th EUROFORUM conference 'Software in the vehicle'*.
- 32 Ketfi-Cherif, A. (1999) Modélisation mathématique d'organes de véhicules automobiles à basse consommation. Applications en simulation, estimation et commande. *PhD Thesis*, Université Paris IX Dauphine.
- 33 Ketfi-Cherif, A., von Wissel, D., Beurthey, S., and Sorine, M. (2000) Modeling and control of a NOx trap catalyst. *SAE World Congress*.
- 34 Martinez, A., Rieuneau, F., Cochevin, D. and Vallée, F. (2006) Construction des méthodes et outils pour la SdF des systèmes de contrôle embarqués. *Lambda-Mu 15 to be presented in Poster session*.
- 35 Matlab/Simulink: <http://www.mathworks.com>
- 36 Millet, J.-B., Maroteaux, F., Emery, P. and Sorine, M. (2006) A reduced model of HCCI combustion in view of application to model based engine control systems. *to be presented in SAE Powertrain & Fluid Systems Conference*, Toronto.
- 37 Royer, L. and Le Vaillant, C. (2006) Pilot software Integration platform for transmission. *Conference: 'Virtual Product Development (VPD) in Automotive Engineering'*.
- 38 Sadai, S. (2006) Improved diesel particle filter thermal management using simulation tools. *to be presented in IFP Conference: New Trends on Engine Control, Simulation and Modelling*.
- 39 Talon, V., Tabet, T., Castric, S. and Emery, P. (2006) Engine control model based design with achille Library. *to be presented in IFP Conference: New Trends on Engine Control, Simulation and Modelling*.
- 40 Villeneuve, A. (2004) Dual mode electric infinitely variable transmission. *Aachener Kolloquium Fahrzeug und Motorentechnik*.

Final manuscript received in October 2006

Copyright © 2007 Institut français du pétrole

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than IFP must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee: Request permission from Documentation, Institut français du pétrole, fax. +33 1 47 52 70 78, or revueogst@ifp.fr.