Improvement of powertrain mechatronic systems for lean automotive manufacturing

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 33 (2015) 53 – 58

www.elsevier.com/locate/procedia

9th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '14

Improvement of powertrain mechatronic systems for lean automotive manufacturing

P. Chiabert [a], G. D'Antonio [a*], J. Inoyatkhodjaev [b], F. Lombardi [a], S. Ruffa [a]

[a]*Department of Management and Production Engineering, Politecnico di Torino, Torino, Italy*
[b]*General Motors Powertrain Uzbekistan,Tashkent, Uzbekistan*

* Corresponding author. Tel.: +39-011-090-7295; fax: +39-011-090-7299; *E-mail address*: gianluca.dantonio@polito.it

Abstract

In recent years, the increasing severity of emission standards forced car manufacturers to integrate vehicle powertrains with additional mechatronic elements, consisting in sensors, executors and controlling elements interacting with each other. However, the introduction of the best available ecological devices goes hand in hand with the legislation and/or limitations in different regional markets. Thus, the designers adapt the mechatronic system to the target emission standards of the produced powertrain. The software embedded into the Engine Control Unit (ECU) is highly customized for the specific configurations: variability in mechatronic systems leads to the development of several software versions, lowering the efficiency of the design phase.
Therefore the employment of a standard for the communication among sensors, actuators and the ECU would allow the development of a unique software for different configurations; this would be beneficial from a manufacturing point of view, enabling the simplification of the design process. Obviously, the new software must still guarantee the proper level of feedbacks to the ECU to ensure the compliance with different emission standards and the proper engine behavior. The general software is adapted to the powertrain: according to the specific target emission standard, some control elements may not be necessary, and a part of the software may be easily removed.
In this paper, starting from a real case-study, a more general methodology is proposed for configurations characterized by different powertrain sets and manufacturing line constraints. The proposed technique allows to maintain the accuracy of the control system and improve process efficiency at the same time, ensuring lean production and lowering manufacturing costs. A set of mathematical techniques to improve software efficacy is also presented: the resulting benefits are enhanced by software standardization, because the design effort may be shared by the largest possible number of applications.

Selection and peer-review under responsibility of the International Scientific Committee of "9th CIRP ICME Conference"

Keywords: Lean Manufacturing; Mechatronic Systems; Powertrain; Product Development

1. Introduction

Mechatronic components have been first employed in automobiles in the 1970s, when the electronic voltage regulator and electronic ignition were introduced. The employment of mechatronic systems increased over time, with particular concern for engine management and control [1]. Today, mechatronic still represents a significant growth area in the motor vehicle branch, even if many engine and safety systems are considered standard equipment in vehicles [2]. The reason for this trend is twofold: on one side, customers require automobiles equipped with high efficiency and low fuel consumption engines; on the other side, as the number of vehicles in the world increases, stricter emission standards are necessary to reduce pollutant emissions [3]. Furthermore, electronic components are usually lighter than the mechanical components they replace, leading to lower fuel and power needed. However, customer satisfaction is not the only target in the development of mechatronic systems; they also must be implemented into the manufacturing processes in an efficient way, with minimum costs for the manufacturer.

Selection and peer-review under responsibility of the International Scientific Committee of "9th CIRP ICME Conference"
doi:10.1016/j.procir.2015.06.011

The general configuration for an engine management control system is shown in Figure 1. The Engine Control Unit (ECU) is in charge of combustion optimization. It receives a set of electric signals from the sensors, which collect information to assess the current state of the engine. The quality of measured data is the result of a compromise among sensors cost, precision and reliability; further, several quantities like torque or emission concentrations are not measurable because of high measurement cost or short life of the data; hence, the ECU has to extract information from indirect measurements [4]. Data are analyzed by the ECU embedded software, which also includes a model of the engine operating conditions; the result of ECU calculation is translated into a new set of electric signals transmitted to the actuators, which determine engine behavior [3,5]. Mechatronic systems ensure benefits compared with merely mechanical frameworks, including higher engine performance and reduced risks for engine damages [6]; such systems are also able to adapt the fuel injection strategy according to the current state of the engine, taking into account, for example, the warm-up phase or the regeneration of particulate traps [5].

Fig. 1. Interconnection of engine mechatronic components.

The improved engine calibration reduces pollutant emissions; moreover, electronics allow to calibrate and control engine functional parameters in order to comply with different emission characteristics. Actually, the prescribed standards for pollutants produced by passenger cars exhibit large differences according to the specific regional markets; hence, worldwide production lines are required to assemble engines with different emission requirements, and the introduction of the best available devices is strictly tied to the legislations holding in the different target markets.

In this paper, we consider the case-study of a plant for car engine production, located in Central Asia; these engines are installed on vehicles sold mostly in CIS countries and on Uzbekistan market. On the Russian market, the Euro 4 standard is currently holding, but conformity to Euro 5 will be mandatory from 2015. In Uzbekistan, the compliance with the Euro 2 standard is currently mandatory, and no additional restrictions are expected.

The employment of mechatronic systems enhances the flexibility of an already developed engine: a flexible monitoring and control system would allow to comply with different emission standards, thus making an engine available for different countries. The adaptation of this system may also allow a manufacturer to extend the production of an existing engine, even if new restrictions on pollutant emissions are prescribed. For example, the introduction of a Diesel Particulate Filter (DPF) allows to easily reach Euro 4 prescriptions, while the installation of an Exhaust Gas Recirculation (EGR) valve on gasoline engines is necessary to comply with the Euro 4 standard.

Unfortunately, in many cases, the ECU software is highly customized for specific applications; it is adapted to the specific platform, the target emission standard, and the employed set of sensors and actuators. This customization occurs because designers adapt the set of mechanical components to the specific application, and rarely reuse an existing engine control system; the software is, in turn, adapted to the employed hardware. This application-oriented approach leads to disadvantages: a single change in the requirements or in the employed system may lead to a completely new software development. Additionally, the software must be redesigned when a new vehicle is developed, even if the hardware set exhibits small variations and the target emission standard does not vary. According to [7], the R&D cost of an engine control system for a diesel passenger vehicle is approximately 14 million dollars. This amount of money includes the combustion optimization phase (design of mechanical components and of engine algorithms), the emission testing of the whole system (engine, after-treatment, and ECU), and the development of new models for the ECU.

The aim of our work is to propose a methodology for the identification of a general formulation of the software embedded into the ECU: the replacement of an application-oriented approach with a functional-oriented one may allow to use the same software on different configurations, enhancing its flexibility, with several advantages for the manufacturer and no quality loss for the customer. In an additional step, we will propose a methodology to develop new ECU models and improve the efficacy of the algorithms.

In Section 2 we will introduce our case-study; in Section 3 we will present the two steps of our methodology. In Section 4, we will explain which are the main advantages of the proposed approach in a lean manufacturing perspective. Finally, in Section 5, we will discuss the impact of this work and provide some hints for future extensions.

2. Case-Study

In this paper, we consider different configurations – currently manufactured – of a four cylinders, gasoline 1.5L engine. In the following, in place of the real names of car models we simply use the tags “Car A” and “Car B”. The engine has been installed on Car A since 2011, and complies with the Euro 2 and Euro 4 standard. Two years later, Car B was updated and the same engine was adapted to this platform, proposed both in the Euro 2 and the Euro 5 versions. The employed sensors are the same in all the cases; conversely, there are some differences in the sets of actuators: the EGR valve is not used onto the Euro 2 vehicles, and other components, including the ECU, change with the platform. Finally, different software have been developed for these configurations. In Table 1, a subset of sensors and actuators is reported: different letters represent different components.

Table 1: Configurations of the engine management system in the case study.

	Car A		Car B	
	Euro 2	Euro 4	Euro 2	Euro 5
Sensors				
T/Body	a	a	a	a
Manifold air pressure	a	a	a	a
Air temperature	a	a	a	a
Coolant temperature	a	a	a	a
Heated oxygen	a	a	a	a
Crank position	a	a	a	a
Cam position	a	a	a	a
Knock	a	a	a	a
Actuators				
Injectors	a	a	a	a
Fuel rail	a	a	a	a
EGR valve	NO	a	NO	b
Ignition coil	a	a	b	b
Canister purge solenoid	a	a	b	b
PDA solenoid assembly	a	a	a	a
VIM solenoid assembly	a	a	b	b
Oil pressure switch	a	a	a	a
Oil control valve	a	a	b	b
Electric control thermostat	a	a	b	b
Spark plug	a	a	a	a
Starter motor A	a	a	b	b
ECU				
Hardware	a	a	b	b
Software	a	b	c	d

3. Methodology

Currently, because of the differences in the actuators sets, a completely new software is developed for each application. The aim of our work is twofold: first, we improve the flexibility of the employed software, so that it may be (at least partially) reused for different configurations; to do this, we will introduce a standard for the communication among sensors, executors and the ECU. Second, we identify a set of smart mathematical techniques that may improve the efficacy of the ECU embedded software, leading to a more efficient use of the engine and enhanced performances.

3.1. Software standardization

First of all, the employment of a standard communication protocol for the information transfer among the components of the monitoring and control system is necessary. For a given type of device, the kind and the format of the information provided (or requested) must be the same; the number of variants due to the specific hardware component leading to variability in software design must be minimized.

A standard procotol named Autosar (AUTomotive Open System ARchitecture) has already been defined. Autosar is a partnership launched in 2003, composed of worldwide OEM manufacturers and automotive suppliers working together to establish and develop an open standard for automotive software architecture [8].

Autosar standard comprises a set of specifications that describe software architecture components and defines their interfaces. A layered structure is defined (see Figure 2), in which the software for car functionalities is separated from hardware-related basic software. The architecture distinguishes among three main software code groups: the Basic Software, the Runtime Environment, and the Application Software. The Basic Software provides essential functionalities for ECU hardware; the Runtime Environment is in charge of the integration among application software functionalities, and handles the exchange of data between Application and Basic software. The Application layer is composed of the software components related to ECU functionalities. In this approach, the functional content of the application software is OEM specific, and is related to the features desired by the car manufacturer. However, even if brand-oriented, also these applications are reusable for all models [9,10].

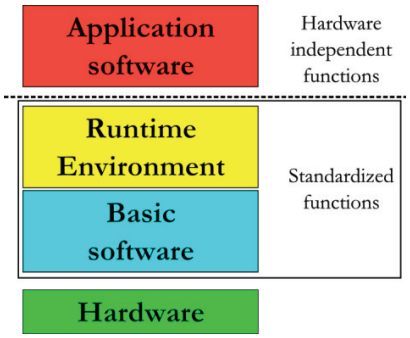

Fig. 2. Autosar software architecture.

The first step of our project consists in adapting the currently employed software to Autosar standard. In this way, the application software may be immediately reused for several configurations; the elimination of variations due to the specific hardware components will lead to reduced variability in software design and lower efforts for software maintenance.

However, the composition of the set of actuators varies with the configuration: for example, the EGR valve is employed on Euro 4 and Euro 5 applications, but it is not used on Euro 2 vehicles. Thus, in the adaptation to Autosar standard, we will also have to separate the software for single functionalities, in a sort of Object-Oriented Programming approach [11]. This methodology will allow us to reuse the most of the software on Euro 2, Euro 4 and Euro 5 vehicles, and employ the EGR functionalities only whether necessary. The net separation of the functionalities makes the software easier to understand, reduces its complexity and enhances its management: software modifications and extensions are easier, and code corruptions or failures are easily solvable, resulting in easier maintenance operations.

3.2. Software efficacy improvement

Once the adaptation to the Autosar standard will be fulfilled, the software for functionalities will consist in a set of independent, communicating code-objects, that may be easily replaced with more suitable ones. Hence, if a performance improvement is necessary, a new algorithm may be developed and integrated into the existing software. However, the algorithms to be implemented must meet the following requirements [12]:

- Reliability: the software must be extraordinarily reliable over the full vehicle lifetime; for example, rebooting during an operation is not feasible;
- Functional safety: ECU also controls functionalities tied to safety systems: software errors may result in inability to change the throttle level, uncontrolled acceleration or degraded engine performance [6], hence a meticulous programming is mandatory and emergency procedures to be used in case of failure must be developed, according to the ISO 26262 standard.
- Real-time behavior: the software must be able to analyze data and provide an output in a few milliseconds, hence the algorithms must be highly optimized;
- Resource consumption: the amount of resources (processor and memory) necessary for the software must be minimized. A common ECU is provided with a few Megabytes of embedded memory, and a CPU with less than 100 MHz.

In the ECU, software for the following functionalities must be embedded: evaluation of the amount of fuel to be injected into the engine; determination of the right moment to deliver the air and the fuel to the engine; control of the idle speed, the ignition timing of the engine, the engine speed, the coolant temperature, the gas emission content.

Thus, the software must deal with a complex system characterized by interconnected dynamics; further, several bounds and constraints, due to physical limits and to the current state of the engine, must be taken into account to avoid damages. Hence, the ECU must deal with an optimal multi-objective problem, but the commonly employed software is not able to take into account the necessary constraints [13].

The ECU may be represented through the model of Nonlinear Model Predictive Control proposed in [14], in which the controller is composed of a system model, an objective function with a set of constraints and a dynamic optimizer (see Figure 3).

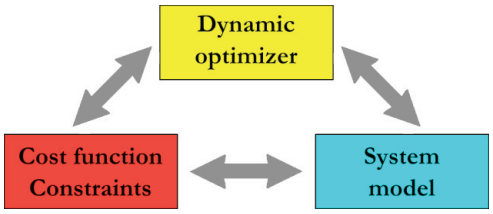

Fig. 3. Model of Nonlinear Model Predictive Control.

Two approaches are available for the formulation of the model describing engine behavior. The first option is a fixed-model, containing all the information which is known a priori. However, this approach does not allow to take into account the deviations from the expected behavior. The second approach is a direct-feedback model; this is a totally adaptive methodology, but a large quantity of data is necessary to build the model, and the extrapolation rules may be unknown, as in the Artificial Neural Networks. A third possibility consists in the employment of a hybrid approach, given by the combination of the two techniques: the software to be developed will be given by a fixed-model and a data-mining technique able to identify global patterns in the observed data. The integration of these two methods allows data-driven changes in the structure of the model, without the drawbacks of a totally direct-feedback model. Another advantage of the hybrid approach is that model adaptability can also take into account deviations due to non-measurable events, such as components aging, environmental effects, or combustion deposits.

The choice of the optimization method is also critical for the performance of the system. Three methods are proposed in [13]. The explicit Model Predictive Control exhibits a low computational effort, but is only able to deal with linear problems; the parameterized MPC allows to derive a low dimensional optimization problem, that can be solved using simple solutions. According to us, the more promising technique is the online Nonlinear Model Predictive Control (NMPC), which can deal with complex nonlinear problems; it is able to evaluate the optimal solution and to stop after a fixed time if the convergence takes a long time. Because of its formulation, MPC explicitly takes into account the physical limits of the system (for example the throttle opening) and the constraints to comply with in order to avoid engine damages. Conversely, the standard ECU software tries to compensate these limits through additional feed forward functions, and the mechatronic system cannot be fully exploited. The importance of this feature is particularly relevant in the transients: the MPC shows a more intensive use of the actuators, leading to faster responses. However, the features of the ECU hardware must be taken into account: the implementation of MPC may require high computational cost and memory loads.

4. The Lean Manufacturing approach

The innovations previously explained can provide different advantages in the automotive lean manufacturing context. According to [15], lean production is a multi-dimensional approach consisting in a large variety of management practices with the aim of creating a quality system able to produce finished products with little or no waste. Lean principles have also been applied to the field of software design [16]. Similarly to the manufacturing framework, lean software development is not a conventional software engineering methodology, but a set of practices for building software systems. In [17], seven principles for lean software development are listed; in particular, the principles that our approach is able to deal with are:

- Optimization of the whole: the value of a software does not only result from the development phase; the capability to modify a code assumes increasing importance over time. Software standardization and the separation of codes related to different functionalities allow to improve the efficiency of the maintenance phase: a problem in the software or a functionality to be enhanced may be quickly identified, and a code-object may be easily replaced with a more adequate one. Further, the improvement of software

efficacy is helpful to optimize the utilization of the mechatronic system, leading to better engine performance.

- Waste elimination: in the lean approach, it is necessary to minimize the processes that do not add value to the product or do not help to reach that value more effectively. First, the identification of a standard protocol contributes to reduce waste, because the same software may be reused for different configurations. Furthermore, the separation of code-objects allows to reduce the maintenance time spent to identify bugs and failures. The improvement of software efficacy leads to better exploitation of the mechatronic system; if it is oversized, a system with lower performances may be employed, leading to money savings without performance downgrade.
- Build quality in: when a new functionality is developed, it may be immediately integrated into the existing software; it is not necessary to wait until the end of a development cycle. Further, if the new functionality does not require additional mechanical components, it may also be integrated into the software of already existing vehicles.
- Deliver fast: the increased rapidity in software design and maintenance allows to think development as a flow system in which the software is designed, developed and delivered in a continuous, steady flow of small changes. This approach is completely different from thinking software development as a project to be completed, or a series of periodical releases.

5. Expected impact

The innovation strategy we presented in this paper is divided into two steps. First, we want to adapt the currently employed ECU software to the Autosar standard: benefits for the manufacturer may be reached in a short-medium term outlook, because software components may be shared among different configurations, and they will not be linked to specific sensors or actuators. This also allows the manufacturer to have already available software for new projects. In this way, the initial investment for software development may be amortized over a larger quantity of vehicles. Further, software designers may neglect the redevelopment of basic code parts, and are able to focus their work on the study of new functionalities and on algorithm refinements, for performance improvement or due to the introduction of stricter norms or to new customers necessities. The second step, consisting in software efficacy improvement, will lead to long-term benefits, both for the manufacturer (cost savings due to an eventual downsizing of the mechatronic system) and the customer (improved vehicle performance). To approach this step, software standardization is mandatory: the development of new mathematical techniques may require a huge effort, thus the resulting algorithm must be employed on the largest possible number of configurations. Further, each algorithm will deal with a particular functionality; hence, the separation of the codes related to different functionalities is necessary, because it makes easier software objects replacement.

In this paper, we employed the ECU as a case-study, but the same approach may be used on other electronic control units. Actually, cars may have up to 70 control units [10] to manage, control and coordinate different functionalities, such as safety (ABS, ASR, ESP, airbags,…), comfort (climate, infotainment systems,…), and vehicle management (transmission, automatic gear, …). The proposed methodology may be profitably applied to standardize the software embedded in the single control units, but is also helpful to improve the communication among different electronic systems.

Another application may concern the transformation of a vehicle from mono- to bi-fuel. When the vehicle undergoes the conversion, it is equipped with another ECU, because some functional parameters (such as the ignition spark) must vary with the employed fuel. With the proposed methodology, software designers may easily prepare the ECU for the information exchange with the secondary control unit. In this case, the standardization of the communication protocol has a significant importance because, during the conversion, some mechanical components (such as the injectors or the sensors for pressure and temperature) are replaced; hence, the ECU must be able to work even if non-original components are installed on the vehicle. In addition, our approach may allow to merge the software of the secondary ECU with the original manufacturer's one, to reach a better integration among the two systems.

References

[1] Resistance is futile – Electronics are on the rise: electronic units and communication protocols. IHS Global Insight, 2009 April.
[2] Automotive technology: greener vehicles, changing skills. Center for Automotive Research, 2011 May.
[3] Ribbens WB. Understanding automotive electronics. 7th ed. Waltham: Butterworth-Heinemann; 2012.
[4] Isermann R. Design of computer controlled combustion engines. Mechatronics. 2003;13(10):1067-89.
[5] Bauner D, Laestadius S, Iida N. Evolving technological systems for diesel engine emission control: balancing GHG and local emissions. Clean Technologies and Environmental Policy. 2009;11(3):339-65.
[6] George E, Pecht M. Tin whisker analysis of an automotive engine control unit. Microelectronics Reliability. 2014;54(1):214-9.
[7] Sanchez FP, Bandivadekar A, German J. Estimated cost of emission reduction technologies for light-duty vehicles. International Council on Clean Transportation, 2012 March.
[8] Automotive standard open architecture [cited 2014, March 20]. Available from: www.autosar.org.
[9] Bunzel S. Autosar – The standardized software architecture. Informatik spectrum. 2011;34(1):79-83.
[10] Fürst S. Challenges in the design of automotive software. In: Design, Automation, and Test in Europe, DATE2010. 2010 Mar 14-18; Grenoble, France.
[11] Booch G. Object-Oriented development. IEEE Transactions on Software Engineering. 1986;SE-12(2):211-21.
[12] Mossinger J. Software in automotive systems. IEEE Software. 2010;27(2):92-4.
[13] Del Re L, Allgöwer F, Glielmo L, Guardiola C, Kolmanovsky I. Automotive model predictive control: models, methods and applications. Berlin-Heidelberg: Springer; 2010.
[14] Findeisen R, Allgöwer F. An introduction to nonlinear model predictive control. In: 21st Benelux meeting on systems and control; 2002 Mar 19-21; Veldhoven, The Netherlands.
[15] Shah R, Ward P. Lean manufacturing: context, practice bundles, and performance. Journal of Operations Management. 2003;21(2):129-49.

[16] Poppendieck M, Cusumano A. Lean software development: a tutorial. IEEE Software. 2012;29(5):26-32.

[17] Poppendieck M, Poppendieck T. Lean software development: an agile toolkit. Boston: Addison-Wesley; 2003.