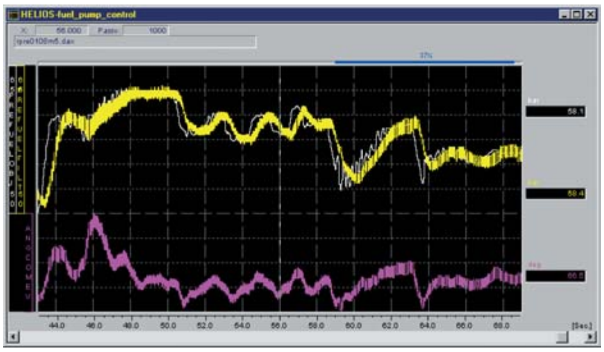
Magneti Marelli's Design of a GDI Engine Controller

- TargetLink supports Magneti Marelli's model-based design methodology
- Improved time-to-market with gasoline direct injection (GDI) project

The design of powertrain controllers is among the most challenging problems in automotive electronics because of the complexity of the functions that the system has to support, like safety aspects, and tight cost limits. Time-to-market requirements and continuously changing specifications have contributed to the migration of most functions to software implementations. Magneti Marelli Powertrain is developing a model-based design methodology clearly defined by a set of successive refinement steps from a level of abstraction as high as possible down to the details needed for the final implementation. In this methodology framework, TargetLink has been successfully used.

In our approach, we identify five main levels of abstraction: system level, function level, operation level, architecture level, and component level. At the system level, the car manufacturer specifications are analyzed and expressed in an analytical formalism. At the function level, the system is decomposed into a set of functions. At the operation level, those functions are built by a network of operations. An operation is an algorithm described in an analytical (at least executable) form, typically Simulink®/Stateflow®. At the architecture level, when an operation is committed to a particular architecture, it produces a software or hardware component.



Fuel pressure control displayed with Magneti Marelli's calibration tool, Helios: Fuel pressure target (white), fuel pressure measure (yellow), control valve angle command (magenta).

Convincing Features

In September 2000, Magneti Marelli Powertrain started to evaluate dSPACE's TargetLink to automatically transpose a Simulink/Stateflow operation specification into a software component. The first application chosen for this evaluation was a part of a gasoline direct injection (GDI) control system: An electronic control unit (ECU) controls a high-pressure fuel pump in a closed loop. After 3 -months of evaluation, we conclude that the tool is:

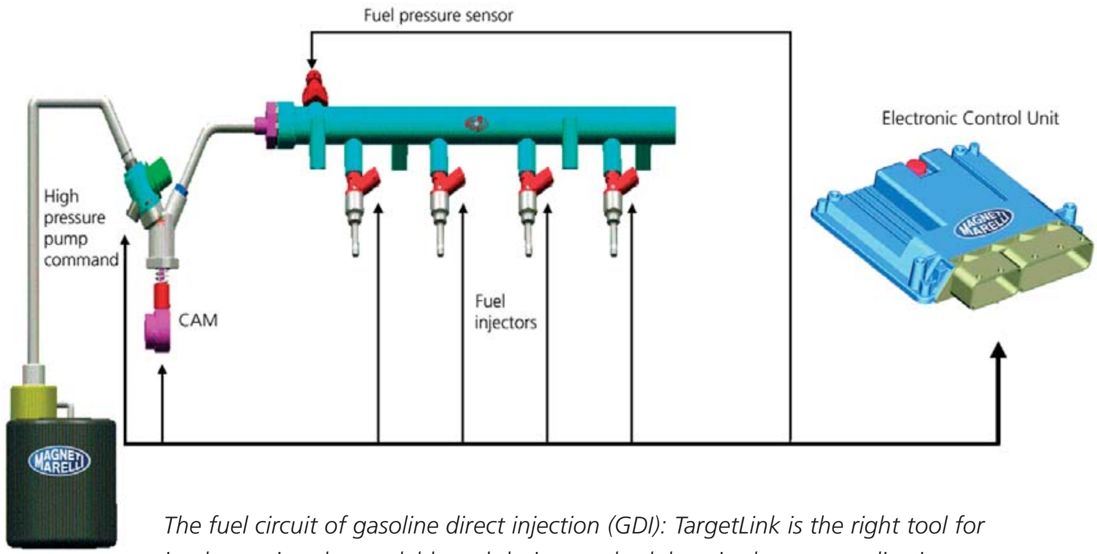
- Applicable to our model-based methodology
- User-friendly

And the generated code is:

- Suitable for our legacy software architecture
- Efficient enough in terms of size and speed

TargetLink As a Model Compiler

Today, we are using TargetLink as our main Model Compiler for the entire GDI engine control application. Any operation designed and verified by our control engineers using Simulink/Stateflow and assigned to be a software component is automatically translated to C code using TargetLink. The functional and implementation verification are still fundamental steps, for which the plant (i.e. engine) and driver inputs are modeled in Simulink/Stateflow using traces and/or complex models. The translation of a Simulink/Stateflow operation to C code might require different design steps. Moreover, the translation is part of a qualified software design process in which each step must be fully traceable. If the target architecture does not support native floating-point operations, the Simulink/Stateflow specification must be correctly and efficiently translated to the fixed-point



The fuel circuit of gasoline direct injection (GDI): TargetLink is the right tool for implementing the model-based design methodology in the new application.

representation of the target microcontroller. This floating-to-fixed point transformation must be verified by simulation (using a host machine), since no formal equivalence between the two representations can be proved. The last design step requires the definition of all the target dependencies, such as target software architecture hooks, RAM and ROM mapping. The outputs of this phase are the C code modules and the ASAP2 file, generated by TargetLink for fine-tuning activities in the application department.

Shorter Time-to-Market

By using the model-based design methodology, we aim to drastically decrease the time-to-market of a new powertrain controller. TargetLink has been a fundamental factor in achieving this goal.

It is important to emphasize that the correctness of the final implementation is closely linked to the correctness of the model, i.e., to the high-level description of the implementation, and hence to the ability to validate the specification. This validation requires the use of complex models of the environment (engine and car driver), and the more correctly addressed it is, the less usage of prototypes or targets is required, resulting in an enormous reduction in design time and cost.

Nowadays, this goal is still far from being completely achieved because of the difficulties of modeling the physical powertrain processes and the other electromechanical parts of the environment interacting with the controller in real time. To overcome these difficulties, prototyping and several design cycles are still required. An important characteristic of the model compiler is that it hides the details of the target hardware/software architecture. For this reason we are also expecting to reduce the design time of other engine control systems due to extensive reuse of operation specifications.

The Right Tool for Model-Based Design

In conclusion, TargetLink has been the right tool for implementing our model-based design methodology, which is essential if we are to improve the time-to-market and cope with the complexity of modern powertrain controllers. In the future, we plan to extend the use of the model-based design methodology to other powertrain applications, and to exploit the new features of the coming releases of TargetLink.

The GDI project team has made extensive use of the model-based design methodology, mainly because of fast evolving system requirements, which create the need for a quick design cycle. Moreover, given the fairly new application, the methodology was applied from the earliest phases onward, allowing the GDI project team to manage complexity at a higher level of abstraction.

Magneti Marelli B.U. Systems:
Giacomo Gentile, Giovanni Stara, Luigi Romagnoli
Parades EEIG:
Alberto Ferrari
Italy