A Model-Based Development Environment and Its Application in Engine Control

Shugang Jiang, Michael Smith, Charles Halasz
A&D Technology Inc.

ABSTRACT

To meet the ever increasing requirements for engine and vehicle performance, and at the same time shorten product development time, model-based development methodologies have been quickly adopted by OEMs and suppliers in recent years. With model-based development, new algorithms are specified in a high-level graphical language and directly compiled into executables to be implemented and tested on rapid control prototyping hardware. This seamless transition from design to implementation and testing greatly improves the efficiency of the development process for modern automotive control strategies, especially in engine control system development.

A model-based development environment and its application in engine control functions development are presented. The environment consists of the ADX rapid prototyping controller, various functional boards, necessary signal conditioning and power electronics modules, and graphical user interface (GUI) software VirtualConsole. Control algorithms to realize various fundamental features necessary for engine operation are developed. To show the features of the environment, the development of two control functions — A/F ratio modulation, and electronic throttle control (ETC) — are selected as examples and their development processes are described in detail. The applications demonstrate the great benefits of the development environment and the rapid control prototyping methodology.

INTRODUCTION

Driven by the ever increasing requirements of fuel economy, safety and environmental protection, more and more electronic systems are being installed on vehicles. Despite the increase in number and complexity of the electronic systems, OEMs are continually seeking to shorten the product development cycle in order to survive the fierce market competition. Under such a situation the market sees a great need for methods to rapidly develop, test and calibrate the vehicle subsystems. This has led to the quick adoption of model-based development methodologies [1-9], whose basic concepts are shown in Figure 1. With model-based development, new algorithms can be specified in a high-level graphical language. Then with the aid of an automatic code generation tool, the algorithms are converted and compiled into executables to be implemented and tested on real-time hardware. Because of the seamless transition from design to implementation and testing, model-based development provides solutions to quickly prototype complex control strategies and carry out testing and verification in an efficient way. System behavior can be assessed and tested in every development phase. Control software can be evaluated and verified early in the design process. Due to these advantages, model-based development has become an essential part of the development process for modern automotive system control strategies, especially in engine control system development.

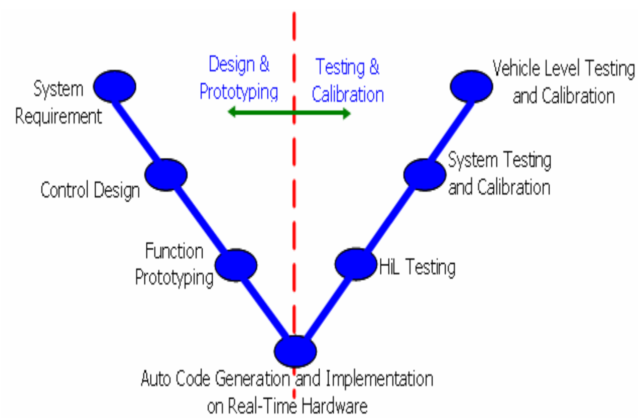

Fig. 1 Model-Based Development Process

The effort to set up such a development environment is presented. For brevity, only the design and rapid prototyping portion of the process is emphasized in this paper. The environment is based on the ADX rapid prototyping controller that includes various functional boards, and necessary signal conditioning and power electronics modules. The ADX rapid prototyping controller has a dual CPU architecture which enables highly efficient digital signal processing operations. It has a modular design and can be configured for a wide variety of applications by combining multiple functional boards. MathWorks products — MATLAB®/Simulink®/Stateflow® — are used to develop control logic in the form of block diagrams.

Real-time interface blocksets for various functionalities are developed and can be integrated in the model. Real-Time Workshop® converts the block diagram into C code, which is then compiled, linked, and downloaded automatically to the target platform for execution under a real-time operating system. The environment uses VirtualConsole for the GUI. Screen elements can be associated with the parameters and variables of the model and thus provide an interface for real-time parameter setting and signal monitoring.

Application of the environment in engine control features development is presented. Necessary functional boards, including engine timing detection and control board, analog input board, and PWM board, are integrated in the system. Signal conditioning for crank and cam sensors, as well as power electronics modules, such as low-side drive, high-side drive, and H-bridge, for ignition, injection, and ETC are included. Control algorithms to realize various fundamental features necessary for engine operation, such as engine timing detection, speed calculation and filtering, look-up-tables for basic spark advance and fuel injection duration, ignition and injection implementation, closed loop A/F ratio control, etc. are developed. To show the features and the great benefits of the environment, the development of two control functions — A/F ratio modulation, and electronic throttle control — are selected as examples and their development processes are described in detail.

The paper is organized into four sections. First, an overview of the ADX rapid prototyping controller is given. The features of the controller, including its hardware and software configurations, are introduced. In the second section the integrated environment for engine control development, the function modules and the overall engine control features are described. The next section explains in detail the process of algorithm development for steady state A/F ratio modulation, and ETC. Finally a conclusion section summarizes the paper.

RAPID PROTOTYPING CONTROLLER OVERVIEW

The ADX rapid prototyping controller is a universal high-speed measurement and control system. Its hardware framework is shown in Figure 2. The controller has a dual CPU architecture, using a PentiumM processor for high-speed simulation and control, and a Renesas SH4 processor for running the human-machine interface, including color touch-screen LCD, programmable function keys, and Ethernet communication with the host PC. A bus controller on the active back plane handles the data transfer between the interface boards and the CPUs. This design frees up the PentiumM CPU and enables execution of highly efficient digital signal processing operations. The model cycle of the system can be as high as 20 kHz. The controller has a modular design. Seven slots are available for any combination of dedicated function boards. Various user selectable general-purpose (A/D, D/A, digital I/O, PWM, thermocouple, serial communication, etc.) and special function interface boards (engine timing detection and control, three-phase PWM motor control, networking, ECU bypass, etc.) are available. Simulink S-functions to access board functions are provided. By combining necessary function boards, the controller can be configured for a wide variety of applications, providing very powerful and flexible solutions for control system design, prototyping, simulation, and testing.

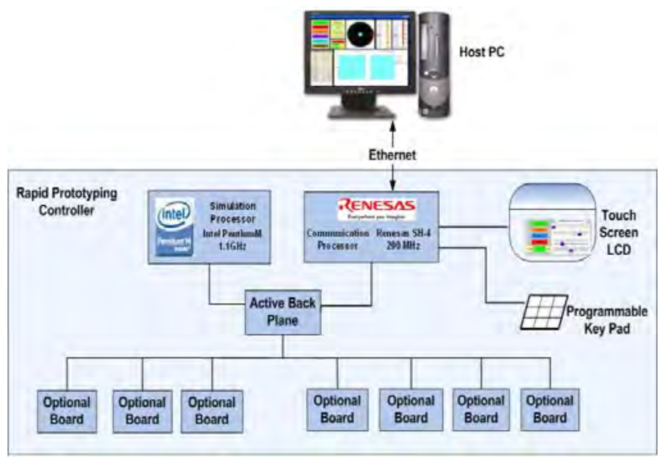

Fig. 2 Hardware Architecture of the Rapid Prototyping Controller

Figure 3 shows the software architecture of the controller and the specific role of each software component. Simulink and Stateflow are used to develop control logic in the form of model block diagrams. The S-functions for system hardware and function boards can be integrated in the block diagram. Real-Time Workshop converts the Simulink model into C code, which is then compiled, linked, and downloaded automatically to the target platform for real-time execution. VirtualConsole is the GUI software. It enables arrangement of various screen elements on the host PC and on the color touch-screen LCD. These screen elements can be associated with the variables or parameters of the model, providing users with an interface for real-time parameter setting and signal monitoring.

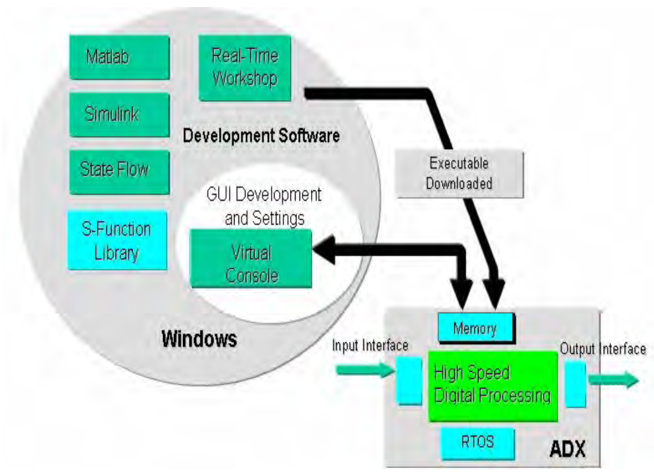

Fig. 3 Specific Roles of the Software Components

ENGINE CONTROL DEVELOPMENT ENVIRONMENT

As shown in Figure 4, the ADX rapid prototyping controller together with necessary function boards, signal conditioning and power stage modules form the hardware platform for engine control. For the design and implementation of basic engine control features, three function boards are integrated in the controller: a general purpose A/D input board, an engine timing detection and control board, and a PWM input and output board. The analog input board is used for measuring such variables as throttle position, manifold absolute pressure, intake mass air flow rate, intake air temperature, and engine coolant temperature, etc. For the environment presented in this paper, the A/D input board is also used to measure the A/F ratio, which is analog signal from A/F ratio meter. The engine timing detection and control board is used for processing the crankshaft and camshaft input signals, obtaining engine speed and timing, as well as sending out control commands for ignition and fuel injection. The PWM board can be used for solenoid or DC motor control, for such functions as EGR valve control, variable cam phasing, and ETC. Necessary signal conditioning modules for the engine timing sensors, as well as power electronics modules with proper functions for actuator operation, e.g., high-side or low-side drive for ignition and injection, H-bridge driver for ETC, are also integrated in the system.

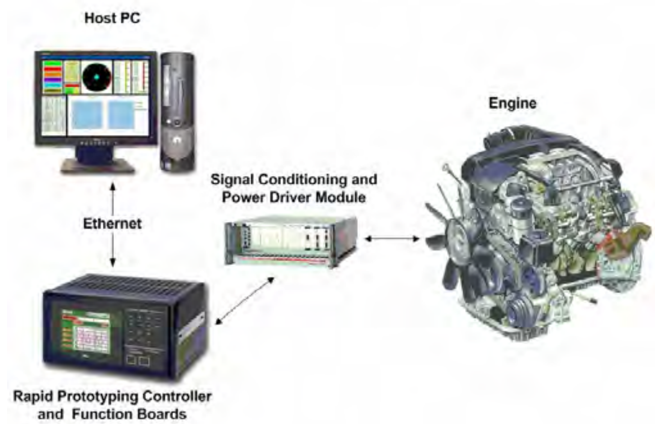

Fig. 4 Engine Control Development Environment

Real-time crank angle detection and cylinder identification are critical for engine control. The engine timing mechanism is specific to customers and platforms. There are various types of crankshaft and camshaft signals that are currently used by OEMs and suppliers. The engine timing detection and control board of the ADX controller can handle different types of input signals, e.g., encoder signal, missing tooth signal, chasing tooth signal, etc. The controller can be configured to use the specific timing mechanism via parameter settings of the S-functions. An FPGA is used for signal processing and the output of control signals for accurate timing resolution as high as 1 μsec or 0.1° crank angle.

Engine control algorithms are designed using Simulink and Stateflow. Based on the hardware platform described above, control algorithms are developed using a model-based methodology to realize various fundamental features necessary for engine operation, such as analog input signal sampling, filtering and conversion, engine timing detection, speed calculation and filtering, look-up-tables for basic spark advance and fuel injection duration, dwell angle calculation, spark and fuel adjustment based on temperature, pressure, and battery voltage, ignition and injection implementation, engine cold start strategy, closed loop A/F ratio control, A/F ratio modulation etc. These basic control features are used to implement the more complex engine operation modes such as idling, steady state operation, and transient operation. To demonstrate the control function development process using the ADX, the algorithm development for A/F ratio modulation and ETC are selected as examples and described in detail in the next section. For brevity, only the most basic portion of the control algorithms is shown.

A/F RATIO MODULATION AND ETC ALGORITHMS DEVELOPMENT

Steady State A/F Ratio Modulation

To reduce engine exhaust emissions, a great many studies on engine control and catalytic converter performance have been made. It is well known that A/F ratio has great effect on catalyst conversion efficiency. Some research shows that modulating the A/F ratio to make use of the oxygen storage capacity of the catalyst yields better conversion efficiency than just keeping the constant stoichiometric value upstream of the converter. To study the relationship between forced A/F ratio modulation and catalyst efficiency, an A/F ratio control algorithm to realize A/F ratio modulation of various frequencies and amplitudes is needed.

We design the algorithm shown in Figure 5 to realize steady state A/F ratio modulation. First the real average A/F ratio is calculated by performing a rolling average calculation of the sampled A/F ratio, with the time window equal to the period of the desired A/F ratio modulation. Then the base fuel width which corresponds to the target average A/F ratio is obtained through a PI controller based on the deviation between the targeted and the real average A/F ratio during the engine operation. The A/F ratio modulation is realized by modulating the base fuel width with a waveform of proper amplitude and frequency. The control parameters, such as the target average A/F ratio, P gain, I gain, and the signals such as the real A/F ratio, fuel injection width, etc. are designated so as to be accessed in the VirtualConsole GUI.

We set the control loop timing to be 10 msec. After the algorithm is designed, it is compiled, downloaded and run on the ADX hardware. A GUI window is designed using VirtualConsole. The screen elements, such as numerical input boxes, oscilloscope, etc. are associated with the control parameters and signals. Through the GUI, the parameters can be adjusted in real time and the signals monitored simultaneously.

Figure 6 shows the results of applying the algorithm on a 4.6 liter V8 engine. The goal is to achieve stoichiometry as the target average A/F ratio, and realize 1 Hz modulation of 5% perturbation amplitude. The left and right banks of the engine are controlled individually and both lambda values are shown. We see the designed algorithm gives excellent control performance and realizes very precise A/F ratio modulation as desired.

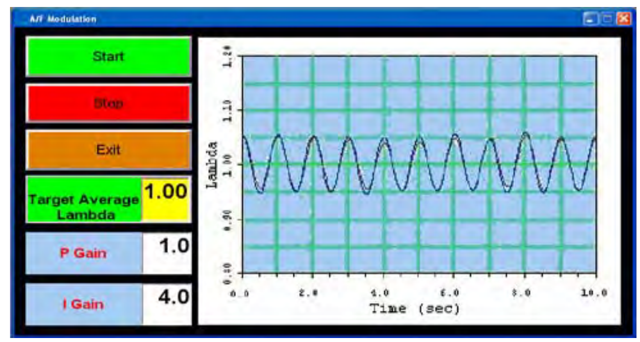

Fig. 6 VirtualConsole GUI for Steady State A/F Ratio Modulation

Electronic Throttle Control

Electronic throttle control system is more and more used and becoming a standard part of the engine. It controls the amount of air intake into the cylinders by precisely positioning the throttle valve at the desired position. An ETC system provides the possibility of improving the overall engine and vehicle performance because with such a mechanism, the engine controller can decide and set the throttle position not only based on driver intention (the acceleration pedal behavior), but also taking into consideration the specific engine

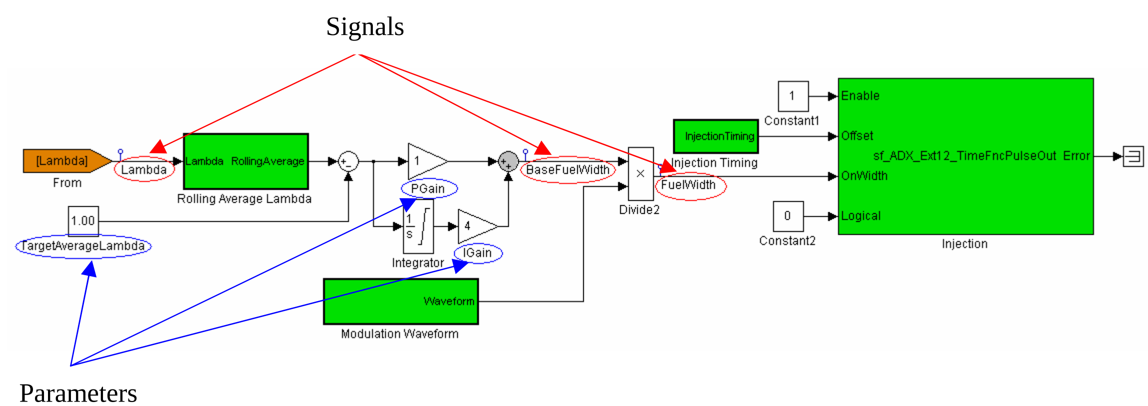

Fig. 5 Algorithm to Realize A/F Ratio Modulation

operation mode information, such as safety factors, emission limitation constraints, etc. After the throttle position target is determined, the requirement for the ETC system is that the throttle should achieve the commanded position as accurately and as quickly as possible. Some usually adopted specifications for ETC include, e.g., step response settling time of less than 100 msec, and steady state tracking error of less than 2%. Comprehensive control of a throttle body needs algorithms that take care of such nonlinear effects as friction, backlash, and return spring nonlinearity. For simplicity, we only focus on the linear operation mode in this paper.

As shown in Figure 7, the electronic throttle body consists of a DC motor, gearbox, throttle plate, return spring for failsafe, and throttle position sensors (usually two of them). The rapid prototyping controller samples the acceleration pedal sensor input, and together with other information of the engine, it determines the required throttle opening. Then based on the throttle position target and the measured real throttle position, the controller determines the PWM signal parameters and sends out the signals to control the operation of the motor, and consequently the throttle plate. By using an H-bridge driver, the motor and the throttle plate can rotate in both directions.

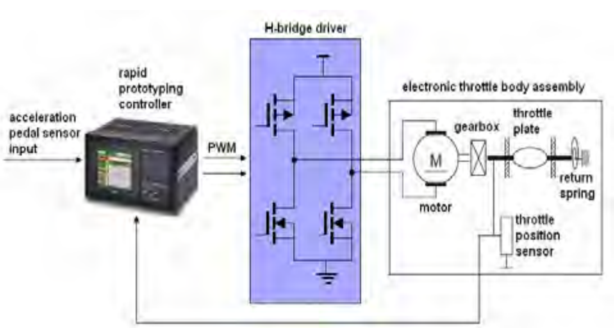

Fig. 7 ETC Hardware

A two degree-of-freedom PID controller is adopted for the throttle control. As shown in Figure 8, the proportional and integral controls are on the setpoint error and the derivative control is on the process variable. The control voltage value of the motor is calculated using the PID control gains, which is then turned into PWM signal duty cycle for motor control. A saturation block is used in the model to take care of the upper and lower limits of the duty cycle. The signals of interest and the control parameters that need be tuned are labeled and can be accessed when the algorithm is implemented on the hardware target.

We set the control loop timing to be 1 msec. Figure 9 shows the throttle response to a 50 degree step command. The trace of the control variable, i.e. the duty cycle, is also shown. Since the control gains can be adjusted in real time and the system behavior monitored simultaneously, the environment provides a very convenient and efficient way for control algorithm tuning to fulfill such specifications as settling time, overshoot, etc.

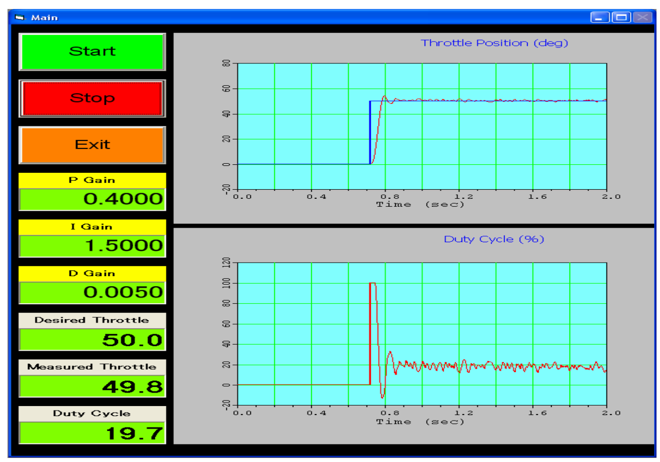

Fig. 9 VirtualConsole GUI for ETC

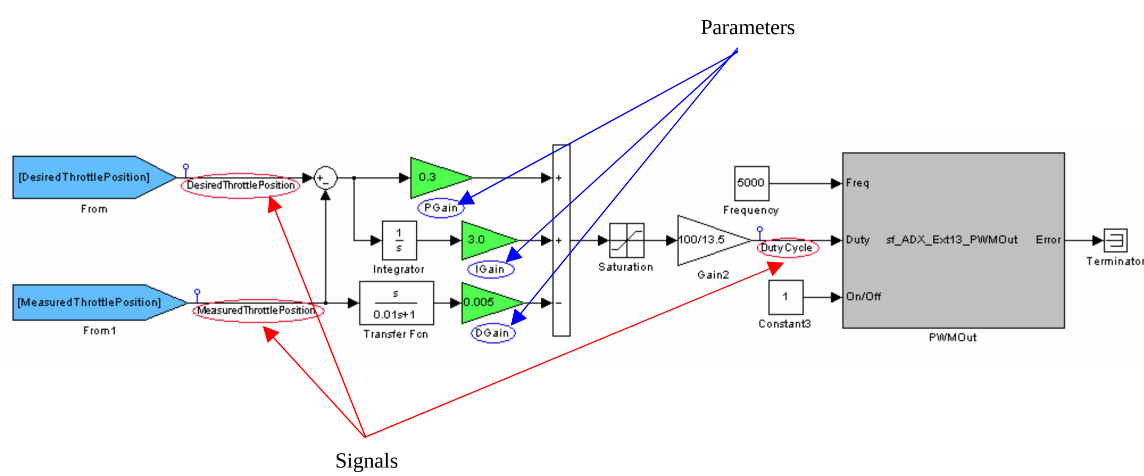

Fig. 8 Two Degree-of-Freedom ETC Algorithm

CONCLUSION

To rapidly develop, test and calibrate the vehicle subsystems, model-based development methodology has been widely adopted, especially for engine control. In this paper, a model-based development environment based on the ADX rapid prototyping controller is established. Its application in engine control functions development is presented. The development process of two control functions — A/F ratio modulation and electronic throttle control — are described in detail to show the features of the system. These examples clearly demonstrate the great benefits of the development environment and the rapid control prototyping methodology.

REFERENCES

1. Jooyoung Ma, Jeamyoung Youn and Minsuk Shin, et al., "A Design Approach using Seamless Development Environments, SILs/RCP, for Real-Time Control Systems," SAE 2006-01-0310.
2. Andrew Beaumont, Joe Lemieux and Paul Battiston, et al., "Design of a Rapid Prototyping Engine Management System for Development of Combustion Feedback Control Technology," SAE 2006-01-0611.
3. Tom Erkkinen and Koos Zwaanenburg, "A Seamless Implementation of Model-Based Design Applied to a New Fuel Control Feature for an Existing Engine ECU," SAE 2006-01-0612.
4. Seungbum Park, Hakmo Yoo and Youngbug Park, et al., "Model-based Development of Engine Control System from Concept to Vehicle," SAE 2006-01-0856.
5. Jeffrey D. Naber, Eric K. Bradley and John E. Szpytman, "Target Based Rapid Prototyping Control System for Engine Research," SAE 2006-01-0860.
6. Michael Beine and Rainer Otterbach, "Development of Safety-Critical Software Using Automatic Code generation," SAE 2004-01-0708.
7. Paul Cook, Peter James and Mark Willows, "Rapid Prototyping of Generic Hybrid Concept Vehicles," SAE 2002-01-0755.
8. Joseph J. Lemieux, "Foundation for Rapid Embedded Development," SAE 2001-01-0022.
9. Shugang Jiang, Dharshan Medonza, et al., "Design and Implementation of an Integrated Development Environment Consisting of Engine Rapid Control Prototyping and Real Time Vehicle Simulation," SAE 2007-01-0515.

CONTACT

Dr. Shugang Jiang
Technical Specialist
A&D Technology Inc.
4622 Runway Blvd., Ann Arbor, MI 48108 USA
Email: sjiang@aanddtech.com