Today's automotive powertrain control systems must meet a wide variety of performance requirements, from improving the necessary inherent vehicle performance to satisfying demands for lower fuel consumption, cleaner exhaust emissions and improved safety. Control-system specifications have thus become extremely complex. The article presents an overview of a powertrain control-system development support environment created to facilitate more efficient development of optimum control systems.

**Powertrain Control-System Development Characteristics and Issues**

An automotive powertrain control system is typically developed through a repeated process of identifying and resolving problems based on actual vehicle tests until the desired performance is attained. The flow of this development process is shown in Fig. 1.

Global environmental concerns and other factors today mean that the specifications of a powertrain control system must meet a widening range of performance requirements. As a result, the control specifications have become more complex and require much greater development time and expense. However, because of intensifying global competition, the development lead time of vehicles must be shortened. Accordingly, more efficient development of powertrain control systems is a high priority.

**Improving the Development Process (see Fig. 2)**

**Automating Hardware Calibration.** A powertrain control system contains numerous control parameters generally set according to calibrations based on tests conducted with the actual vehicle. An automatic calibration envi-

---

*Jiro Sumitani is with Himeji Works and Kiyoharu Anzai is with Mitsubishi Electric Control Software Corp.*
The advanced logic integrator for car electronics (ALICE) environment (see Fig. 3) has been developed for powertrain control-system development. ALICE is built around MATLAB/Simulink (from The MathWorks, Inc.), a commercially available control-system design support tool, and the TargetLink code generator (from dSPACE GmbH) that runs on MATLAB and generates fixed-point program code. To these core capabilities we have added the above-mentioned unique functions developed by Mitsubishi Electric Corporation.

**ALICE Workbench**

This is an integration tool built to enable various development support tools created in-house to be used with consistent operating procedures. These tools include ALICE Designer, which incorporates additional functions for specification descriptions based on control-model diagrams, for comparing control-model diagrams and for conducting information searches. It improves the development efficiency and reliability of control-model diagrams using the ALICE Auto Coder that automatically generates optimized object code based on control-model diagrams; the
ALICE Doctor (document generator) that automatically generates detailed specification documents from control-model diagrams and other information; and the ALICE Caliber that automatically generates the information for calibrating the actual powertrain hardware.

**ALICE Data Server**
A control model generally describes the processing operation in terms of actual physical quantities. However, functional limitations of the microcomputer that is used require conversion to a fixed-point processing description. Conversion to a fixed-point description can be accomplished with TargetLink, but the various types of information required must be entered manually in each mathematical function block. Therefore, an environment has been created that facilitates automatic entry all at one time through centralized management of the information needed for conversion to a fixed-point description.

**ALICE Library**
It is difficult to describe interrupts and the like with just the standard control-model description blocks of MATLAB. To make such descriptions possible and reduce the size of the program that is generated, we created an independent block for powertrain-control descriptions and incorporated it into the ALICE Library.

**Development of an Automatic Calibration Environment**
Calibration has traditionally been performed as shown in Fig. 4. Control parameters were changed manually and measured for each operating state of an actual vehicle, and the process of changing the parameters was repeated until the measured results coincided with the target control values. One problem with this approach has been the enormous time and cost involved in processing the data.

To resolve this problem, input/output functions for controlling the operating state of the actual vehicle were added to the previously used calibration tool along with an automatic processing capability, as shown in Fig. 5. This resulted in the creation of an automatic-calibration environment configured as shown in the diagram.