

AVL Develops Dynamic Controller for Engine Conditioning System Using Embedded Code Generation for PLCs

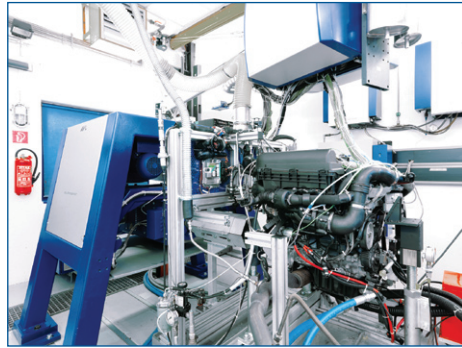
Engine temperature variations of even a few degrees can increase exhaust emissions beyond mandated limits. To develop and calibrate engine management systems that meet standards around the world, automotive engineers need to simulate the precise temperature conditions experienced by an engine. With an engine test bed, they can evaluate control strategies in a repeatable environment and access engine performance across a range of ambient temperatures and driving conditions.

Building on years of experience in developing test beds for Formula 1 and NASCAR race cars, engineers at AVL List GmbH developed a control system for a dynamic engine conditioning system. The conditioning system is part of a passenger vehicle test bed that emulates the engine's operating environment. Using MathWorks tools for Model-Based Design, they completed the project in just six weeks.

"We had to choose between modifying an existing steady-state implementation written in C code and starting fresh using MathWorks tools," notes Michael Hofer, solution engineer at AVL. "We chose to completely redevelop the system using Simulink, Stateflow, and Real-Time Workshop®. This approach was about twice as fast as hand-coding, which enabled us to complete the project on time."

THE CHALLENGE

The engine conditioning system was required to maintain engine temperature within 1 degree Celsius of the target temperature across the operating range of -20 to +110 degrees Celsius. AVL had an exist-



AVL passenger vehicle engine test bed.

ing steady-state controller for an engine conditioning system. While its steady-state accuracy was 1 degree Celsius, it had a dynamic error of 5 degrees Celsius.

"We had some novel control strategies that we were using for our racing customers, but they were for more sophisticated conditioning systems and would need to be modified for passenger cars," notes Josef Mayrhofer, lead engineer for Racing Applications at AVL.

Because they had just six weeks to deliver a working system to the customer, AVL engineers needed a development environment that would enable them to quickly adapt and try out a new control strategy. Furthermore, they needed to rapidly implement a prototype of the system to enable real-time testing on production hardware.

THE SOLUTION

AVL used MathWorks tools for Model-Based Design to model, simulate, implement, and deliver the control system on schedule.

AVL engineers used MATLAB®, Simulink®, and Stateflow® to develop an initial model of

THE CHALLENGE

Develop an engine conditioning control system for a passenger vehicle test bed in just six weeks

THE SOLUTION

Use MathWorks tools for Model-Based Design to design, simulate, and implement the control system on a PLC

THE RESULTS

- Development time cut in half
- Design iterations completed in hours, not days
- Documentation time eliminated

“ Using MathWorks tools, we verified that our initial control design was on track in the first week and delivered a complex control system in only six weeks. Model-Based Design enables us to simulate complex designs, conduct software-in-the-loop testing, and then rapidly generate efficient code for multiple target hardware platforms. ”

Josef Mayrhofer, AVL

the control system that comprised a state machine and four proportional-integral (PI) controllers. Completed in just one week, this model enabled the team to conduct open-loop simulations in Simulink and verify that the basic control strategy would work.

After using Stateflow to develop supervisory logic, the team simulated safety-critical functionality to ensure that internal pressure and temperatures never reached unsafe levels.

With Real-Time Workshop Embedded Coder™, the team generated C code from their Simulink control system models. They then deployed the control algorithms on a B&R X20 programmable logic controller (PLC) using B&R Automation Studio™ Target for Simulink.

Testing on the PLC real-time prototype revealed that the new design performed better than the steady-state design, but still could not control temperature within the required 1 degree.

After AVL mechanics added new valves to the conditioning system, the engineers modified the Simulink and Stateflow models to incorporate the new hardware and refine the state machine, which ultimately contained 10 parent states, some with as many as 80 sub-states. They then re-generated the code with Real-Time Workshop Embedded Coder.

After verifying that the design met performance specifications, the team used Simulink to implement safety features, add an interface to the CAN bus, and further stabilize the design.

The team re-generated code for the final control system and tested the conditioning system on the full test bed against a European drive cycle before delivering a production system to the customer.

THE RESULTS

■ **Development time cut in half.** “With MathWorks tools we completed the project on time, in just six weeks,” says Hofer. “We would have needed 12 to 15 weeks if we were hand-coding in C.”

■ **Design iterations completed in hours, not days.** “When our mechanical experts added valves to the physical conditioning system, I used Simulink and Stateflow to change some states and add PI controllers to the model,” Hofer reports. “With two more clicks, I had a new prototype ready to test. A change that would have taken two or three days of manual coding we completed in half a day.”

■ **Documentation time eliminated.** “Stateflow models make it easy to communicate our design to other engineers within AVL and to our customers, who accept the models as documentation,” says Hofer. “On the conditioning system, this saved us a week of documentation effort, and on larger projects, two or three months.”

Learn more about AVL: www.avl.com

INDUSTRY

- Automotive

APPLICATION AREAS

- Control systems
- Embedded systems
- System design and simulation
- Embedded code generation
- Rapid prototyping

PRODUCTS USED

- MATLAB®
- Simulink®
- Stateflow®
- Stateflow Coder™
- Real-Time Workshop®
- Real-Time Workshop Embedded Coder™
- B&R Automation Studio™ Target