

# Hitachi Accelerates the Development of Engine Knock-Reduction Systems

In a well-running automotive engine, the engine control unit (ECU) continually adjusts the spark angle for increased performance and efficiency. Engine knock occurs when a portion of the air-fuel mixture spontaneously combusts ahead of the spark-initiated flame front. A pressure wave is created in the cylinder, which propagates through the engine block and may result in engine damage. Therefore, the ECU has to continually detect any presence of knock for spark feedback control.

Hitachi developed and tuned algorithms that control knock by adjusting engine control and timing in real time. They used Simulink<sup>®</sup> to develop the ECU and MATLAB<sup>®</sup> to analyze data.

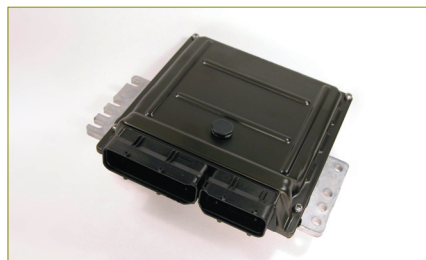
“The ease-of-use and broad functionality of MATLAB and Simulink were instrumental to the timely completion of this project,” says Jonathan Borg, research engineer at Hitachi America.

## THE CHALLENGE

To meet project deadlines and customer expectations, Hitachi would need to rapidly complete two related engineering projects.

For the first project, they had to design and develop an ECU, incorporating algorithms that controlled how much fuel would be injected and when a spark would occur. They would then need to generate code for their prototyping hardware.

For the second project, they would need to calibrate and improve the control algorithms in the ECU by gathering engine knock data from a running engine in real time and then analyze that data.



Hitachi engine control unit.

“For data analysis, the biggest challenge is to get a complete understanding from pressure data, combustion, and other thermodynamic aspects to signals and signal processing techniques,” notes Borg.

## THE SOLUTION

Hitachi used Simulink and Real-Time Workshop<sup>®</sup> to develop and generate code for the rapid-prototyping ECU, and Simulink, MATLAB, and the Signal Processing Blockset to analyze the pressure and accelerometer data gathered from knock tests.

To develop the ECU, Borg assembled the hardware, including signal conditioning, wiring, and harnesses. He then developed the engine-control algorithm with Simulink.

The ECU uses engine position input to determine when to open the fuel injector and how long to leave it open. Crank and cam sensors provide digital inputs and interrupts. ECU filters remove noise and determine engine speed and position.

Borg used Simulink S-functions to develop and integrate algorithms written in C for engine position input (via the cam and crank sensors) and injection and ignition control for incorporation into the ECU model.

### THE CHALLENGE

To develop an ECU for reducing automotive engine knock

### THE SOLUTION

Use MathWorks tools for Model-Based Design and data analysis to rapidly develop sophisticated, real-time control algorithms

### THE RESULTS

- Development time reduced by months
- Proof-of-concept prototype built in two weeks
- Data analysis time halved

To control the engine's electronic throttle, Borg created a proportional integral derivative (PID) controller with Simulink and tuned it in real time.

Once the ECU was built in Simulink, Borg used Real-Time Workshop to automatically generate C code that was run in real time on dSPACE control prototyping hardware. He then validated the algorithm before deploying it on a real engine.

With the ECU in place, Borg gathered data from an engine instrumented with pressure sensors in each cylinder. Data was acquired at fast sampling rates as the engine was operated under various knock conditions. A typical test run produced more than 100 MB of data.

Borg used MATLAB to read the resultant data files and parse, validate, and analyze the data. He also used band pass filters in the Signal Processing Blockset to filter data during analysis.

Once the data was extracted, Borg used the Signal Processing Toolbox to perform fast Fourier transforms (FFTs) and correlate pressure with knock sensor data across the entire data set. This enabled him to calibrate and enhance the knock-control algorithm in the production ECU.

Recently, Hitachi reused a knock analysis tool developed with MATLAB to analyze knock on a six-cylinder engine for the same carmaker.

"Flexibility is a big advantage of MathWorks tools," Borg explains. "When the carmaker provided data for the six-cylinder engine, I just modified my scripts and quickly applied them to the new engine."

With the data analysis completed, Hitachi is using MathWorks tools to develop more advanced knock-detection techniques.

“On another project using another tool, the software development alone took four months. With MathWorks tools, we completed that development in less than two weeks.”

Jonathan Borg, Hitachi

## THE RESULTS

■ **Development time reduced by months.** Using Simulink and Real-Time Workshop, Hitachi developed a working rapid-prototyping controller in fewer than two months. Much of that time was dedicated to acquiring parts and assembling and testing hardware. Using other packages required four months of software development time.

■ **Proof-of-concept prototype built in two weeks.** "During the project, the customer wanted to get a feel for how the knock-reduction algorithm worked. Although it was still in development, I implemented a proof-of-concept entirely in Simulink, including the FFT, without writing a single line of C code. In one week, I had it running and demonstrated it for the carmaker on the dyno-engine. Without Simulink, we would have not been able to do this," explains Borg.

■ **Data analysis time halved.** "Writing code in MATLAB is at least 50 percent faster than in C, and even faster because most data analysis functions are available in the toolboxes," Borg explains.

To learn more about Hitachi, visit [www.hitachi.us/automotive](http://www.hitachi.us/automotive)

### APPLICATION AREAS

- Automotive
- Control design
- Data analysis
- Model-Based Design
- Signal processing

### PRODUCTS USED

- MATLAB
- Simulink
- Real-Time Workshop
- Signal Processing Blockset
- Signal Processing Toolbox

[www.mathworks.com](http://www.mathworks.com)