

Simulation Enhances Embedded Systems

Jon Titus, Senior Technical Editor

Simulation tools let engineers focus on algorithm designs rather than on debugging complicated C code.



You might wonder whether engineers who design embedded systems can benefit from using simulation software. "If you plan to control something like a nuclear reactor or a large piece of machinery, where you'd pay a high price for creating a bad controller, it's obvious you simulate the algorithm first," explained Pete Darnell, president of Visual Solutions.

But would you simulate a home appliance? "A large manufacture, such as General Electric would," answered Darnell. "They want to produce the most efficient and safe products at the best prices, and the simulation and optimization of a design, with automatic code generation, lets them meet those goals. Our VisSim software, for example, lets designers use an add-on called Motion that includes high-fidelity simulation of many types of electric motors. So, engineers don't have to create their own motor models. They can create and test their control algorithms and then generate C code that runs on, say, a Texas Instruments C2000 processor."

Before engineers turn to simulation, they must examine the complexity of their application," said Tony Lennon, an industry marketing manager at MathWorks, the source of MATLAB and Simulink software. "Simulation makes sense if they have a fairly complex control architecture that might include some supervisory or logic control, or they have several control loops that interact with each other, which makes manual loop tuning difficult."

Simulation Enhances Embedded Systems

Published on Electronic Component News (<http://www.ecnmag.com>)

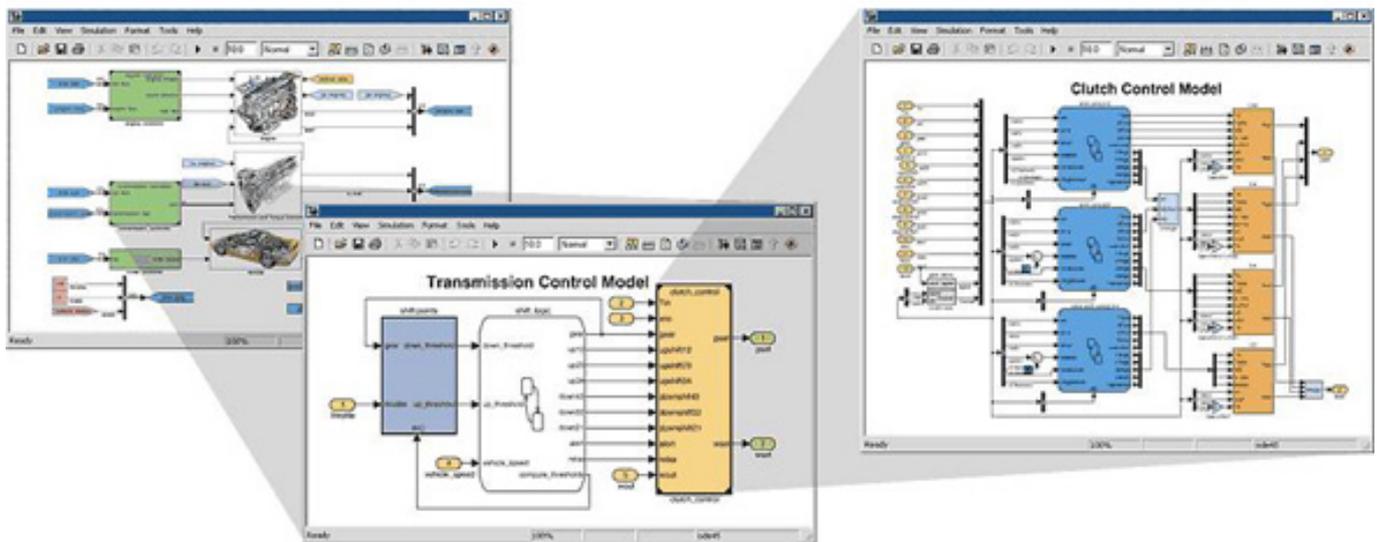


Figure 1. Simulink provides an environment for multi-domain simulation and model-based design for dynamic and embedded systems. To construct a model, you assemble design components, each of which could be a separate model.

"After they realize their inefficient design methods no longer work for increasingly complicated devices, engineers begin to understand how system-level simulation can help," said Lennon. "And they get excited about using a software environment to help with system design. Managers, on the other hand, want to know the impact of implementing simulation as a design tool. The big change is that when companies invest in simulation tools, their engineers and managers must handle design and development in a different way."

"More and more often we hear about the requirement to meet government regulations, 'We need a certain level of safety for x-, y-, and z-type products,' or companies require compliance with a safety standard such as IEC 61508," explained Lennon. "So if you have a safety requirement for a design or product, you need simulation tools that ensure you can meet those specifications and standards."

"Like any software tool, it took simulation time to mature into a mainstream engineering tool," explained Darnell. "But we have seen visual simulation-programming tools such as VisSim and Simulink get to that point. We recently benchmarked VisSim-generated code against code written by a power-supply-control expert for an MCU-based DC-DC buck converter. And we ran a similar comparison for a sensorless field-oriented motor controller. The benchmark showed the C code from VisSim ran only about 4-percent slower than the hand-coded C and assembly-language code created by TI experts, which is very good."

"So there's still a slight penalty, but consider it took the power-supply expert almost a year to handcraft his C code," continued Darnell. "It took us less than a week to construct the VisSim diagram, and less than a second for VisSim to generate the C code. We know certain C constructs will generate faster code than others on the C2000, so our simulation tools create code to get the maximum performance from a system."

Simulation Enhances Embedded Systems

Published on Electronic Component News (<http://www.ecnmag.com>)

Engineers often ask, "Can I slide this piece of code right into my processor?" "They can," answered Lennon. "We have capabilities in our code-generation tools that create code in exactly the format specified by software engineers; just as if they had written it. And we can target any microprocessors with C-code, FPGAs with VHDL or Verilog code, and PLCs with structured text that complies with the IEC-61131 standard."

Visual Solutions has a close relationship with Texas Instruments, and in particular the C2000 MCU group that focuses on digital control of power supplies and motor drives. "We support all of the C2000 devices including their on-chip peripherals as well as most of the TI development kits," said Darnell. "And we help engineers easily test and monitor their code via TI's real-time debug JTAG interface that will exchange data with the target and probe its memory without interrupting the controller code."

VisSim provides a block to let you easily buffer high speed values for any signals such as motor currents, rotor positions, PID gains, Clarke-and-Park-transform values, and so on, in your diagram. After the buffer fills, you examine the data on a scope-like plot and you can change loop gains on the target MCU to immediately see how they affect your controller," said Darnell.

"After people becomes comfortable with automatically generating code from a simulation model, they also can create a real-time simulation of the plant, or device, they want to control," emphasized Lennon. "Then you can test your controller against a real-time plant model. The controller 'sees' the plant model as a real piece of equipment. That's a necessary step because it gives you more insight into your control system and you think more about how you must develop the algorithms and the timing aspects of a real-time system. You also start to consider the system's real I/O requirements and get a better picture of how well you can control your system." This technique works particularly well if you cannot test all of your algorithms on real equipment or if hardware designers don't yet have a prototype.

Many controller-design projects require quite a bit of math, but the price of the end product might not allow for an MCU with floating-point operations. The change from implementing a control algorithm in floating-point to fixed-point math can take a great amount of time and overflows and underflows inherent in fixed-point operations can wreak havoc with control loops. Engineers usually simulate an algorithm first with floating-point math and then switch to fixed-point math before they run code on real hardware.

"That change creates a challenge for many engineers," said Lennon. "The more complicated the math, the more time the engineers must spend on the math conversion. We have tools that will help. A customer in Europe had a PID loop that we converted to fixed-point math in about 10 minutes. It would have taken a programmer more than a day."

"We also provide a complete fixed-point block set for VisSim," said Darnell. "Then when you have a fixed-point version of your algorithms, you can throw in all kinds of

Simulation Enhances Embedded Systems

Published on Electronic Component News (<http://www.ecnmag.com>)

oddball cases and erroneous sensor responses to ensure you've properly guarded against overflow and are using sufficient precision to get the desired response."

For further reading

"PC-Bases Prototyping of Audio Applications Using Model-Based Design,"
www.mathworks.com/mason/tag/proxy.html?dataid=12345&fileid=59582.

"Sensorless 3-Phase PMSM Motor Control,"
www.vissim.com/solutions/field_oriented_motor_control.html.

MathWorks and Visual Solutions offer trial versions of software products:
www.mathworks.com and www.vissim.com.

Source URL (retrieved on 08/21/2014 - 6:08am):

http://www.ecnmag.com/articles/2011/09/simulation-enhances-embedded-systems?qt-most_popular=0