USER STORY

InterSense Develops an Extensible Architecture for Motion-Tracking Systems

In the 1990s, when InterSense, Inc. was founded, motion-tracking systems were largely developed for virtual reality applications. That focus has broadened to include simulation and training, cockpit helmet tracking, and immersive virtual environments for industrial visualization and design. InterSense has adapted to industry developments by creating a core architecture that supports the requirements of diverse motion-tracking applications.

Using MathWorks tools, InterSense designed and automatically generated the production code for the Sensor Fusion Core, an extensible architecture for motion-tracking applications. The architecture has already been used in a miniature inertial-optical cockpit helmet-tracking system that will assist pilots of emergency medical services and military aircraft in flight.

THE CHALLENGE

InterSense sought to create new business opportunities by developing motion-tracking applications that would leverage their experience and existing technology. Because the company's legacy systems were handwritten in C, however, it was time-consuming to adapt and customize the code for each new application and its specific combination of sensors.

"We wanted to define the architecture for a sensor fusion core that could handle all those configurations and sensors today and tomorrow," explains Eric Foxlin, InterSense founder and CTO. "We needed the next generation of our software to easily handle all the foreseeable applications."

The IS-1200 FlightTracker cockpit helmet-tracking system.

THE SOLUTION

InterSense used Simulink® and the Signal Processing Blockset to model and validate the various units of the Sensor Fusion Core.

The model, which includes more than 6,000 blocks and 590 subsystems, consists of a dynamics reference unit, a measurement management unit, a measurement update unit, the environment map manager, the vehicle map manager, and the sensor/target meta-driver.

"If I want to add an algorithm to the system, I simply add Simulink® blocks and quickly test and validate them," explains Foxlin. "Simulink lets us create a very accurate simulation of the whole system."

Following a spiral development process, the team then developed a working prototype of a wide area mobile tracking system for augmented reality and mobile robotics.

"Simulink encourages spiral development, in which we continually add features," explains Foxlin. "The same platform that we used for algorithmic analysis and simulation is also used for developing a real working system. That gives us continuity throughout each project."

InterSense used the Signal Processing Blockset to simplify the development of

THE CHALLENGE

To build a generalized architecture that enables rapid development of a diverse range of complex motion-tracking applications

THE SOLUTION

Use MathWorks tools for Model-Based Design to model and automatically generate C code for an extensible sensor fusion architecture core

THE RESULTS

- Reduced project delivery times
- SBIR contract
- Foundation for new business opportunities

> *With Simulink, the Signal Processing Blockset, and Real-Time Workshop, it is very rewarding to push the build button and have a working product. I no longer have to wait for the design to be implemented by a separate team of C programmers, and still not know whether the code and model are the same.*
>
> Eric Foxlin, InterSense, Inc.

extended Kalman filters and other intensive linear algebra and estimation algorithms—from position and orientation tracking to autocalibration and environment mapping. Many of the signals are represented as matrices. The Signal Processing Blockset provided InterSense with all the matrix math, linear algebra, and statistics operations, saving hours of tedious hand coding.

"The Signal Processing Blockset provides features such as singular value decomposition, matrix inverses, spectral analysis, and various solvers, each of which would require hundreds of lines of C code," says Foxlin. "For example, a Kalman filter is a complicated algorithm to implement in C, but with MathWorks tools I simply drop a few blocks into my model and tune the parameters dynamically in simulation."

Engineers then used Real-Time Workshop® to automatically generate C code from their Simulink model to run in their embedded hardware.

"Because the real system is built in Simulink and compiled with Real-Time Workshop, I don't have to maintain a simulation of my system and the actual system separately," Foxlin explains. "They are the same, and they are always in sync."

InterSense has produced two applications based on the Sensor Fusion Core: the optical-inertial IS-1200 VisTracker product and the miniature inertial-optical cockpit helmet-tracking system. Under a small business innovative research (SBIR) contract for the U.S. Army, the company is refining the system and flight testing to produce the most advanced and flight-ready cockpit helmet-tracking system.

THE RESULTS

- **Reduced project delivery times.** Using Simulink and the Signal Processing Blockset, InterSense engineers simulated and validated their architecture in days. "I estimate that MathWorks tools saved us several months of manual C coding, and even more months of debugging and tuning, while producing higher-quality and more portable code," says Foxlin.
- **SBIR contract.** InterSense's Sensor Fusion Core has helped the company win a U.S. Army SBIR contract for a miniature inertial-optical cockpit helmet-tracking system. They completed Phase I of the contract in six months—and demonstrated a fully operational prototype that was completed ahead of schedule. This prototype helped InterSense win the Phase 2 contract and generate sales opportunities.
- **Foundation for new business opportunities.** MathWorks tools for Model-Based Design help InterSense simplify the process of porting applications to new platforms and enable them to verify and validate their products for compliance with government regulations.

To learn more about InterSense, visit **www.intersense.com**

APPLICATION AREAS

- Aerospace and defense
- Automatic code generation
- Model-Based Design
- Simulation

PRODUCTS USED

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

www.mathworks.com

91296v00 11/04