USER STORY

BAE SYSTEMS CNIR Develops On-the-Move Antenna Pointing and Stabilization System

The U.S. Army's Future Combat System will use Joint Tactical Radio System radios and Warfighter Information Network-Tactical (WIN-T) antennas to provide soldiers with more mobile and dependable communications. BAE SYSTEMS Communication, Navigation, Identification, and Reconnaissance (CNIR) is ensuring that the WIN-T antennas maintain reliable communication links among troops under severe disturbances, such as when traveling in Humvees over rough terrain.

Using MathWorks tools, BAE SYSTEMS CNIR met an aggressive four-month project deadline by designing and implementing the controller for a WIN-T on-the-move antenna pointing and stabilization system before field test control hardware was available.

"We validated the automatically generated C code from Real-Time Workshop® Embedded Coder on a prototype controller integrated with the actual antenna pointing system," explains Dr. David Haessig, senior member of technical staff at BAE SYSTEMS CNIR. "This enabled us to save time by thoroughly testing the code in a realistic environment and eliminating recoding and retesting efforts."

THE CHALLENGE

When BAE SYSTEMS CNIR learned that the control hardware would not be available on time, they decided to develop and implement a prototype controller to meet their four-month project deadline.

To save additional time, they sought to reduce redundant development tasks, such as manually writing C code.

Test vehicle for integrating the antenna (upper right).

"On earlier projects, we would write a detailed document defining the algorithm, and this would be used by our software engineers to write and test the code," explains Haessig. "This process took considerable time."

THE SOLUTION

Using MathWorks tools, BAE SYSTEMS CNIR designed their antenna pointing and stabilization system. They used Real-Time Workshop Embedded Coder and xPC Target to implement and test the real-time operation of the prototype system, and to automatically generate embedded C code for the development system.

"We generated an ANSI-C function representing the entire controller," says Haessig. "Our software engineers incorporated this function in their code to implement the controller in the development hardware. This process has worked successfully. In fact, we've since decided to use automatic code generation for other parts of the WIN-T system."

The physical control system consists of a GPS inertial unit, a two-axis pedestal that supports and drives the antenna, and control electronics that tie these components together.

THE CHALLENGE

To design and implement an antenna pointing and stabilization system that maintains a connection from an on-the-move ground vehicle to a target air vehicle, under adverse conditions

THE SOLUTION

Use MathWorks tools to model the system, automatically generate and run code on real-time rapid prototyping hardware, and automatically generate code for production deployment

RESULTS

- Development time reduced
- Documentation and redundant tests eliminated
- Customer needs met

> *MathWorks products have better positioned us by helping us achieve lower costs in a faster timeframe.*
>
> Dr. David Haessig, BAE SYSTEMS CNIR

BAE SYSTEMS CNIR used Simulink® to develop a plant model of the physical antenna system, and to model and simulate the control algorithms that point the antenna. They used the Control System Toolbox to design and analyze the controller.

They used Stateflow® to design their RS-422 interfaces for the antenna pedestal and the GPS inertial system, which enabled them to react to 422 traffic, convert data formats, and synchronize the embedded control software and the hardware.

"This type of system requires accurate synchronization between components," explains Sam Nazari, systems engineer at BAE SYSTEMS CNIR. "Using Stateflow to trigger parts of the model enabled us to synchronize the controller with the GPS inertial system."

Using Real-Time Workshop®, they generated code from their models, and used xPC Target to run the code in real time on a 3 GHz Dell workstation with RS-422 interface modules.

They used Real-Time Workshop Embedded Coder to automatically generate embedded C code for integration with the system.

"The generated code from Real-Time Workshop Embedded Coder has a one-to-one correspondence to the model, including the signal and parameter names," says Nazari. "This was useful in explaining the code to the software engineers and made our implementation easier."

BAE SYSTEMS CNIR validated their test results at MIT Lincoln Laboratory using a six-degree-of-freedom motion simulation table. They are now implementing the system on a test vehicle for live demonstrations and in preparation for the U.S. Army's Development Test/Operational Test (DT/OT) in 2005.

THE RESULTS

- **Development time reduced.** "We asked our software group to determine how much time it would have taken to hand code the algorithm," says Haessig. "Based on their feedback, we believe enough time was saved to recoup the additional investment in MathWorks tools needed for this project. Future projects will yield a cost savings."
- **Documentation and redundant tests eliminated.** "Traditionally, we developed test cases using a Simulink model of the algorithm, which our software engineers used to debug and verify their handwritten C code," explains Haessig. "Real-Time Workshop eliminates hand coding, which removes the need to write a detailed software design document and streamlines the testing work."
- **Customer needs met.** "Getting to the prototype and field-test phases early is critical if you want to establish credibility and win contracts," explains Haessig. "We completed this project with only two people under an aggressive deadline of four months. MathWorks tools enabled that."

For more information on BAE SYSTEMS CNIR, visit **www.baesystems.com**

APPLICATION AREAS

- Automatic code generation
- Model-Based Design
- Rapid prototyping
- Simulation

PRODUCTS USED

- MATLAB®
- Simulink
- Control System Toolbox
- Stateflow
- Stateflow Coder
- Real-Time Workshop
- Real-Time Workshop Embedded Coder
- xPC Target

www.mathworks.com

91260v00 11/04