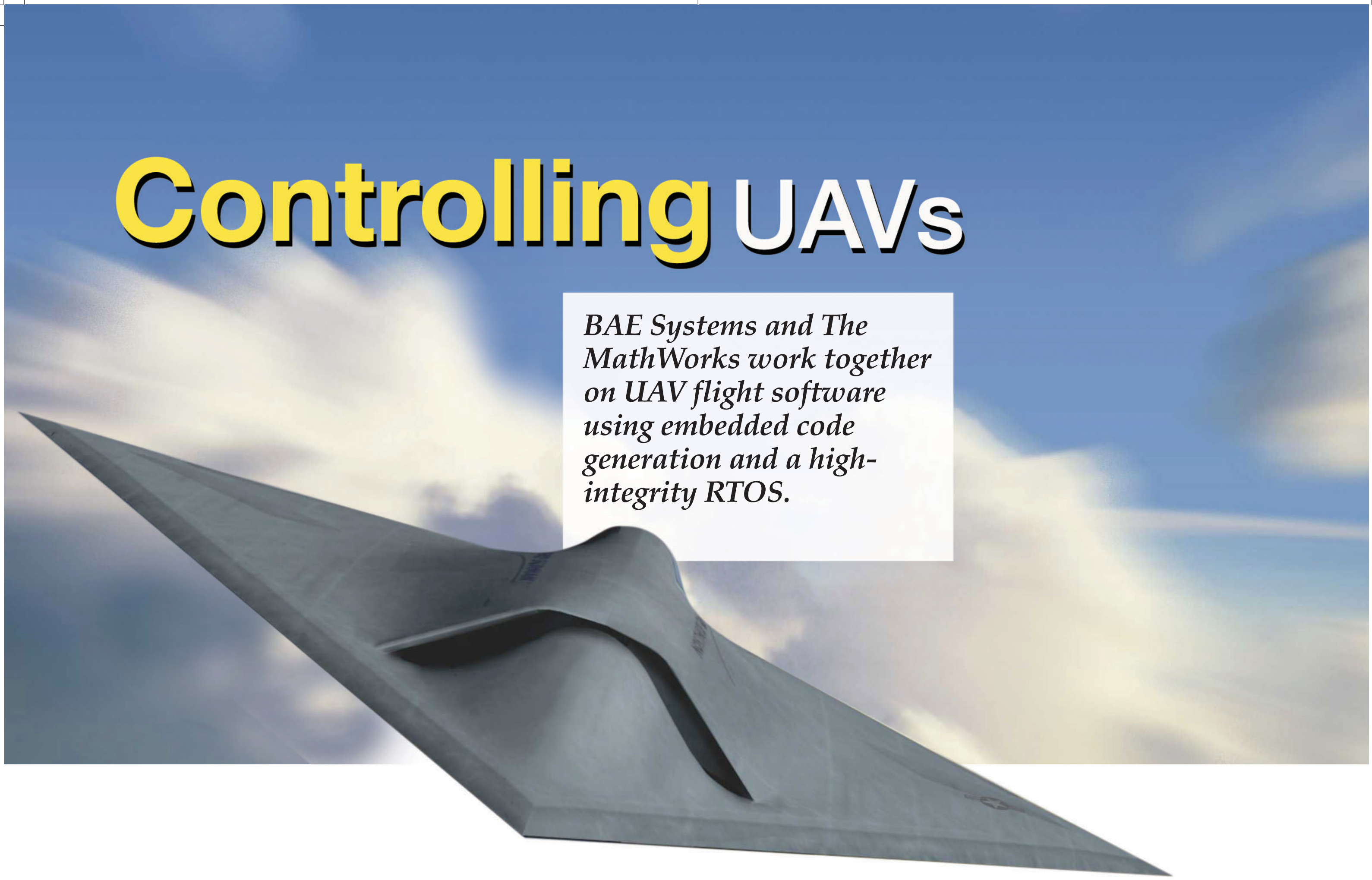
Controlling UAVs



BAE Systems and The MathWorks work together on UAV flight software using embedded code generation and a high-integrity RTOS.

As flight-control and avionics systems of military and commerical aircraft are required to deliver more functionality and perform more complex missions, their onboard software becomes increasingly complicated to design and produce. Flight-control systems for unmanned aerial vehicles (UAVs) have the added complexity of autonomous or remote control.

UAV system architecture is highly sophisticated and requires the configuration of such items as a vehicle management controller, actuators, data-link receivers and transmitters, payload assembly, a GPS receiver, a power generator, and a battery. Within limited budgets and aggressive schedules, **BAE Systems** Controls in Johnson City, NY, and Los Angeles, CA, routinely has to deliver reliable, flight-critical software that operates remotely and autonomously from a ground control station. The company must also create designs that allow both the hardware and the software to be cost-effectively integrated and migrated to new applications.

Integration of new and existing applications with a real-time operating system (RTOS) can be very complicated and labor intensive. BAE has successfully integrated its CsLEOS RTOS with The **MathWorks** modeling and code generation tools on a variety of manned and unmanned aircraft programs.

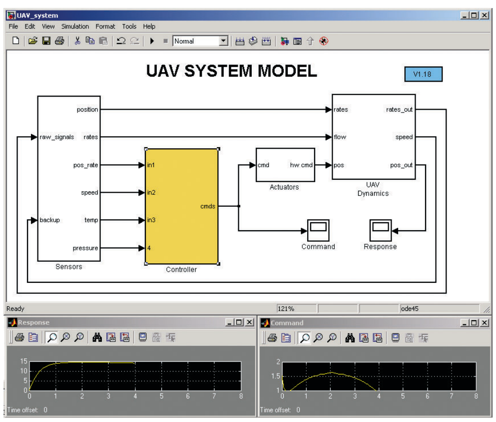
Within the last year, CsLEOS and automatically generated code has been deployed in the integrated vehicle management system computer on the **Northrop Grumman** Pegasus X-47A UCAV (unmanned combat aerial vehicle); the flight-control computer designed for the **Boeing**-built C-17 Globemaster III **U.S. Air Force** transport; fly-by-wire flight control for **Sikorsky**'s S-92 and H-92 helicopters; and various other current and future systems being developed.

The tools

The CsLEOS RTOS, designed specifically for embedded applications, employs "brick-wall" time and space partitioning to operate multiple systems independently of each other. If one system experiences a failure, the others are unaffected.

The system's ARINC-653 compliant applications programming interface ensures ease of use. Its RTCA/DO-178B, Level A, certification support package provides reliability standards for safety-critical use. The CsLEOS is a scaleable, real-time, deterministic, multi-tasking operating system targeted for applications that operate in a flight/safety critical environment.

The CsLEOS provides temporal and spatial partitioning for each separately loadable application. Each application is called a partition and it is allocated a reserved area of memory that is not available for use by other partitions. The CsLEOS executes the partitions according to a predefined schedule. Within these partitions exist one or more tasks that carry out the functional requirements of the partition. Reserved operating windows for each partition ensure applications have a fixed period of uninterrupted access to the proces-



Executable UAV model with sensors, actuators, plant, and controller model, shown via a typical Simulink model.

sor. Since the CsLEOS separates partitions from each other in both time and space, applications of different critical levels can be executed on the same processor without interfering with each other.

MathWorks products are used extensively in the control industry. COTS (commerical off-the-shelf) tools based on Simulink and Stateflow allow model-based design for many different types of real-time systems. From these designs, efficient, readable code can be auto-generated for an array of target processors by using the Real-Time Workshop Embedded Coder. Documentation can also be auto-generated using the same set of designs from which the code was generated.

Flight software design

To reduce overall integration time of the embedded application, BAE uses MathWorks' MATLAB, Simulink, and Stateflow tools to create executable mathematical models that accurately simulate UAV system behavior. BAE uses these models to design UAV embedded software components, including autopilot models and integrated navigation algorithms. The autopilot model includes flight phase and mode logic; longitudinal, lateral, and directional loop control; throttle and fuel mixture control; guidance and navigation data computation; and engine status monitoring.

The UAV system models also contain components of the plant environment, which include six-degrees-of-freedom nonlinear dynamic models of the aircraft with all variables computed as outputs. The plant model also includes atmosphere and turbulence, the landing gear, the steering wheel, control surfaces, actuators, sensors, engine speed, exhaust gas temperature, propeller thrust, and a data link.

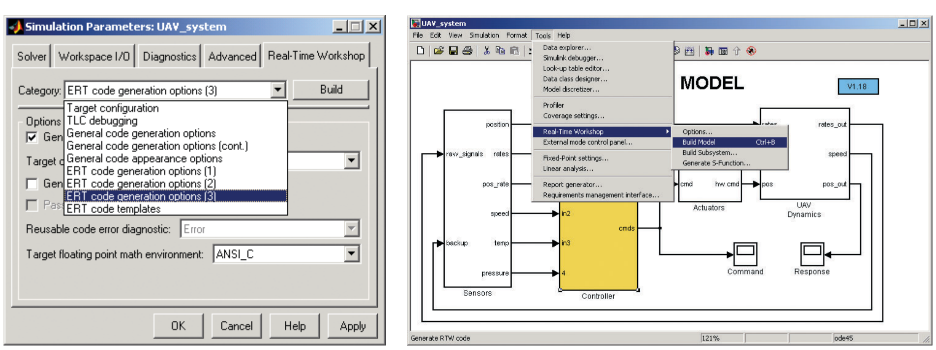
The control system is modeled in discrete time and uses fixed sample rates. The plant is modeled in continuous time and may or may not use variable step rates. The overall system model is hybrid since it involves a combination of discrete and continuous time dynamics. The typical model also contains finite state machines within Stateflow for mode logic in addition to the classic feedback control block diagrams provided by Simulink. Table lookups, legacy code insertion, detailed data types, and function and file partitioning are all handled within the modeling and simulation environment.

During simulation, model profiling information can be collected to evaluate performance and optimize loading. Model coverage tools are used to reveal possible errors by providing structural coverage such as modified condition/decision coverage. If the system does not meet the requirements, changes can easily be made in the model-based environment and the tests run again. Once the system meets all requirements, code can be generated for the application partition, which can be loaded into the target running CsLEOS.

Implementation

After using Simulink and Stateflow for design and development UAV developers then automatically generate flight code with the Real-Time Workshop Embedded Coder. The code generated is ANSI-C and can be run on any system that can compile standard C code.

After the application code is generated, the next step is to integrate the generated code with BAE's CsLEOS. Its brick wall partitioning scheme allow multiple applications to run in separate partitions with no spatial or temporal interference. The total flight software consists of application layer software provided by Real-Time Workshop, which is controlled and scheduled by the underlying CsLEOS RTOS layer software.



The process for generating the code involves selection of appropriate code optimization switches and configuration settings.

Control systems designed with CsLEOS and MathWorks products are often comprised of a set of control laws embedded in CsLEOS partitions. Included in each partition are CsLEOS tasks (threads) that can control the input and output from the control laws. This I/O could be from a supported external device or from another partition in the system. The CsLEOS toolbox for Simulink includes blocks for partition creation, task creation and I/O through the use of built-in intrapartition communication services.

To simplify integration, a set of Simulink blocks was developed for CsLEOS, allowing developers to create and test entire applications as models, which can then be transferred directly to the target system for implementation.

Testing and documentation

Within Simulink, documentation of the system and software designs is automatically generated from the same models as those used to generate the code and verify the simulation. Code reports are generated for the application code and include HTML traceability links back to the corresponding Simulink blocks. This documentation and traceability feature facilitate UAV design reviews and analysis.

BAE engineers quickly resolve flight test issues by logging and identifying problems and then modifying the Simulink model, testing the new requirements by simulating the model, automatically generating code, and conducting hardware-in-the-loop testing and a flight test checkout. New functionality is added to the system in a similar way.

Model-based design saves time during software development and the ability to auto-generate entire CsLEOS applications contributes to less overall design time. The built-in test and simulation tools of the MathWorks software products reduced testing efforts by allowing design tests to be run without the hardware. Once the design met the requirements, the Real-Time Workshop generated an entire CsLEOS application that was downloaded to the target processor.

This article was written for *Aerospace Engineering* by **William G. Barnes**, Senior Principle Software Engineer, BAE Systems Controls, and **Tom Erkkinen**, The MathWorks, Embedded Applications Manager.