

# BAE SYSTEMS Controls Develops Autopilot for Unmanned Aerial Vehicle Using MathWorks Tools

As air vehicle flight control and avionics systems 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.

Using MathWorks tools, engineers at BAE SYSTEMS Controls in Johnson City, New York, and Santa Monica, California, developed and successfully flight tested a UAV flight avionics system and ground control station in just eight months. The tools enabled the geographically separated teams to design, model, and analyze the control system, simulate the UAV, visualize the simulation data, perform testing, and automatically generate the autopilot code, all in a single, model-based design environment.

“MathWorks products greatly reduced our development cycle time and cut system and software design and testing costs by 50%,” says Feng Liang, principal engineer at BAE SYSTEMS Controls.

## THE CHALLENGE

Engineers at BAE SYSTEMS Controls had to develop and flight test the avionics system and ground control station software for a UAV with an Eagle 150 airframe and a Teledyne Continental Motors IO-240-B engine. Composites Technology Research Malaysia was responsible for overall UAV integration. Excelnet handled the aircraft modification and integration tasks.



An Eagle 150 unmanned aerial vehicle in flight.  
Image courtesy of Composites Technology Research Malaysia.

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 a limited budget and an aggressive schedule, the BAE SYSTEMS team had to deliver reliable, flight-critical software that would operate remotely from a ground control station and autonomously. They also wanted a design that would allow both the hardware and the software to be cost-effectively migrated to new applications.

The BAE SYSTEMS team wanted to use design tools that were widely used, well tested, and fully compatible. They also needed to work in an environment that was well suited to a model-based design approach.

## THE CHALLENGE

To enable teams working in separate locations to design a sophisticated UAV autopilot system quickly and inexpensively

## THE SOLUTION

Using MathWorks tools, modify existing software designs in a model-based design environment and then automatically generate embedded control code

## THE RESULTS

- Design and rework costs substantially reduced
- Testing cycle time minimized
- Coding errors and manual documentation work minimized

“*MATLAB and Simulink greatly reduced development cycle time and cut system and software design and testing costs by 50%.*”

Feng Liang, BAE SYSTEMS Controls

## THE SOLUTION

BAE SYSTEMS chose MATLAB®, Simulink®, and Stateflow® to model the UAV system. They decided to save time and costs by reusing existing software designs. These designs included an autopilot model, developed using MATLAB, Simulink, Stateflow, and an integrated navigation algorithm.

For their new system, the engineers needed to include a six-degrees-of-freedom nonlinear dynamic model of the Eagle 150 aircraft with all variables computed as outputs. They also needed models for 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 existing autopilot model included 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.

Real-Time Workshop® was used to automatically generate the embedded software from the Simulink and Stateflow models. Documentation of the system and software designs was automatically generated from the same models as those used to generate the code and verify the simulation.

The engineers quickly resolved 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 was added to the system in a similar way.

The first monitored flight of the UAV was a great success. The basic autopilot functions were checked and found to be satisfactory. System requirements were validated during additional flight tests.

## THE RESULTS

- **Design and rework costs substantially reduced.** By reusing Simulink models from other BAE SYSTEMS groups, the team created a baseline design for the new UAV autopilot, substantially cutting costs early in the process.
- **Testing cycle time minimized.** Simulation-based testing, which the team described as “much more efficient” than manually unit-testing both the system and the software, allowed the team to meet their aggressive schedules.
- **Coding errors and manual documentation work minimized.** Using a common design tool and common models and automatically generating both the embedded software and design documentation from the same source minimized human translation errors and reduced manual documentation work.

To learn more about BAE SYSTEMS Controls, visit [www.baesystemscontrols.com](http://www.baesystemscontrols.com)

### APPLICATION AREAS

- Aerospace and defense
- Modeling and simulation
- Control design
- Automatic code generation

### PRODUCTS USED

- MATLAB
- Simulink
- Stateflow
- Real-Time Workshop

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