UMN UAV Simulation Quick Start Guide

The UMN UAV simulation model is written in the MATLAB/Simulink environment. Three simulation environments are used: the Nonlinear simulation (UAV_NL), the Software-in-the-Loop (SIL) simulation (UAV_SIL), and the Processor-in-the-Loop (PIL) simulation (UAV_PIL). All three simulations share the same plant dynamics, actuator, sensor, and environmental models via Simulink Libraries. Each simulation can be loaded from its respective directory by running "setup.m". Aircraft and environmental parameters are set in m-files and shared between the simulations. Two aircraft models are maintained, one for the Ultra Stick 25e and one for the FASER aircraft.

MATLAB/Simulink Version

The UMN UAV simulation was developed with 32-bit MATLAB R2010a. Users have reported successfully using R2009b; however, R2010a or later is recommended. R2009a is known to fail with the SIL simulation. The following table details the toolboxes and version numbers required for each simulation. You can find your MATLAB version and toolbox information by typing the command "ver" in the MATLAB command window.

Product	Version	UAV_NL	UAV_SIL	UAV_PIL
MATLAB, 32-bit	7.10.0.499	X	X	X
Simulink	7.5	X	X	X
Aerospace Blockset	3.5	X	X	X
Simulink Control Design	3.1	X	X	X
Control System Toolbox	8.5	X	X	X
Real Time Workshop	7.5			X
Real Time Windows Target	3.5			X

Note: the trim and linearize functions supplied with the UMN UAV sim require Simulink Control Design and the Control System Toolbox. These functions use the UAV_NL sim, but can be called from the SIL_Sim and PIL_Sim directories. While not explicitly required to run the UAV_SIL and UAV_PIL simulations, these toolboxes are needed to trim these simulations.

Directory Layout

The simulation directory is arranged into the following subdirectories:

- *UMNUAV_sim**Simulation**Documentation*\- documentation for the simulations
- *UMNUAV_sim**Simulation*\ *Libraries*\- holds all of the common files that are shared across all three simulations
- *UMNUAV_sim**Simulation**NL_Sim*\- Nonlinear simulation model.
- *UMNUAV_sim**Simulation**SIL_Sim*\ - Software-In-the-Loop simulation model.
- *UMNUAV_sim**Simulation**PIL_Sim*\ - Processor-In-the-Loop simulation model.

UAV_NL: Nonlinear Simulation

The nonlinear simulation (UAV_NL.mdl) has the plant dynamics only (no actuators or sensor models). It is used for generating and storing trim conditions and linear models. The trim condition generated with this model is used for the other simulations. The aircraft configuration, trim condition, and linear models are stored in the Libraries directory.

To switch between the FASER and Ultrastick aircraft models, edit line 32 of "setup.m" and change the input to "UAV_config" from:

```
[AC,Env] = UAV_config('Ultrastick');
```

to

```
[AC,Env] = UAV_config('FASER');
```

UAV_SIL: Software-in-Loop Simulation

The SIL simulation (UAV_SIL.mdl) includes the plant dynamics, actuators, sensors, time delays, and a flight control law. The trim condition from the UAV_NL simulation is used to initialize the model.

Two controller implementation options are available: a Simulink-based implementation or a C-code implementation interfaced via a MEX-function. Controller implementations should follow the Interface Control Document "UAV_controllaw_ICD.pdf" located in *UMNUAV_sim**Software**Documentation*\. The controller implementations can be switched by setting the value of the "controller_mode" variable in setup.m . Simulink and C-code implementations of a baseline PI controller and a LQR controller are provided, as well as placeholders for a student controller in Simulink and C-code.

If MATLAB's *mex* has not been configured type "mex -setup" in the Matlab command line and choose the *lcc* compiler for Windows and *gcc* for Linux machines. This configuration needs to be performed just once.

The controllers are implemented using Model Referencing and the switching is done by Variant objects. See the Simulink documentation for more details. This feature may fail in MATLAB versions R2009a and earlier. Removing and reinserting a Model Reference block should fix this error, but switching between controllers will need to be done manually.

The flight controller commands are added to the trim inputs prior to being sent to the actuator model.

UAV_PIL: Processor-in-Loop Simulation

The PIL simulation (UAV_PIL.mdl) is the same as SIL, but instead of the flight control law, this simulation has serial hardware interfaces to connect it to the flight computer, which will be running the flight control law. This simulation also has a connection to FlightGear for visualizing the aircraft state. The trim condition from the UAV_NL simulation is used to initialize the model.

Workflow

A basic control law development workflow is as follows:

1. Start in the NL_Sim directory. Run “setup.m” which will trim and linearize the model at a nominal flight condition.
2. If desired, run the “example1.m” script to see a comparison of the doublet response of the linear and nonlinear models. “model_check” will run the simulation and plot the results compared to a baseline checkcase.
3. Design a control law using the linear models.
4. Implement the control law in the template C-code in *UMNUAV_sim**Software**FlightCode**control*\ and verify the performance by running the SIL sim.
5. In the SIL_Sim directory , edit “setup.m” to specify the name of you C-code in the MEX file compilation step. “simulate_and_save.m” will run the simulation, and “plot_SIL.m” will plot the results. “SIL_montecarlo.m” will vary parameters in the simulation and plot the results. “model_check.m” will run the simulation and compare the results to a baseline checkcase.
6. Implement the control law into the full flight code and compile it to the flight computer.
7. In the PIL_Sim directory, run “setup.m” which will load the PIL simulation, start FlightGear, and start the Ground Station software, and compile and connect the UAV_PIL simulation.
8. Follow the PIL guide to run the PIL simulation.