

AMASE Quickstart Guide

Overview

The development of algorithms for command and control of unmanned aerial vehicles (UAVs) can be aided greatly by simulation. The Aerospace Multi-Agent Simulation Environment (AMASE) aims to provide a simulation environment for the demonstration and testing of such algorithms and related technologies. AMASE brings several vehicle motion, system, and control models together to form a single model for UAV simulation. It is designed to allow for command and control of multiple UAVs using control algorithms within the simulation as well as external control via a network interface.

AMASE includes the necessary components to create scenarios, simulate aircraft with basic EO/IR cameras, and interact with control algorithms to command aircraft in a scenario. Graphical user interfaces allow AMASE users to evaluate the actions of one or more aircraft at runtime, and the data output feature saves scenario data for post-processing. An integrated network server allows clients to connect to AMASE in order to send and receive data. AMASE includes three main programs: the simulation, a data playback tool, and a scenario setup tool.

The AMASE simulation models 5-DOF (coordinated turning) flight dynamics with self-configured performance at a set of design points. AMASE UAVs feature an autopilot that manages coordinated turns, altitude hold, heading hold, vertical speed hold, speed hold (auto-throttle), maintain track in wind fields, loiters (Figure-Eight, Orbit (circular), Racetrack) and waypoint following. Additionally, AMASE UAVs can be equipped with gimbaled and fixed sensors, and the simulation performs footprint analysis for target detection and includes line-of-sight calculation for obscuration of sensors by terrain.

Generally, AMASE allows users to:

- Create UAV simulated scenarios
- Define simulated vehicle capabilities
- Connect external applications to provide simulated UAV commands
- Collect, present, and analyze UAV simulated mission performance
- Provide a common interface to connect research on autonomous control systems from industry, academia, and government.
- Close the loop between algorithm development and analysis/demonstration

Running AMASE

The main directory [AMASE_DK_2015](#) contains the AMASE Development Kit. The file [README.txt](#) provides some additional high-level information, and directory [docs](#) contains documentation that goes into more detail about various aspects of AMASE.

To get a demo running quickly on a Linux machine, do the following, with commands issued in a terminal window from the main directory [AMASE_DK_2015](#):

1. Run the command `cd run/linux/Examples/`
2. Run the command `sh TaskAllocator.sh`
3. AMASE should load. Once it does, in the AMASE application, go to File > Open Scenario and choose [example_scenarios/Example4_AssigningTask.xml](#). This should change the map and load some vehicles.
4. Press the play button under the “Simulation Controls” panel at the bottom. The vehicles should start to move.
5. At any point, press the “Assign Task” button in the “Simple Task Allocator” panel on the side to assign tasks to the vehicles.

This demonstrates the general flow for running AMASE. A shell script such as [TaskAllocator.sh](#) loads AMASE, making reference to a location in which configuration files in XML format are stored ([config/examples/TaskAllocator](#) in this case). These configuration files determine which AMASE modules and plugins are loaded and what options they use. A scenario file in XML format such as [example_scenarios/Example4_AssigningTask.xml](#) describes the mission scenario, e.g. how many aircraft are in the scenario and what tasks they should perform. See [LMCP/LMCPGuide.pdf](#) and [LMCP/AMASE LMCP Build/doc/index.html](#) for more information on the message set used to set up scenarios and issue commands during simulated missions.

There are some additional useful shell script files in [run/linux](#) and [run/windows](#). For instance, [SetupTool.sh](#) provides a tool for quickly setting up scenario files.

Developing for AMASE

The directory [AMASE_DK_2015](#) contains a Netbeans project for AMASE.

The tool [LMCP/PackageMaker.jar](#) allows developers to add messages to the LMCP message set. Existing message sets are stored in [LMCP/CMASiv3_2015-04-23.xml](#) and [LMCP/PERCEIVE.xml](#). See [LMCP/LMCPGuide.pdf](#) and [LMCP/AMASE LMCP Build/doc/index.html](#) for more information.

See [AMASE Tutorial.pptx](#) for more information on developing for AMASE.

Public Release

DISTRIBUTION A. Approved for public release: distribution unlimited. Case: #88ABW-2015-4601.
Determination date: 24 September 2015.