UAV Flight Simulator based on ESA Infrastructure

Flight simulation models compliant with SMP standard

António Daniel Miranda de Almeida

Abstract

Demand for unmanned aerial vehicles (UAVs) has been increasing in the civilian and military sectors. This translates to the need for better tools, such as flight simulators, to support development and operations. Flight simulators can be used to support the design of UAVs and respective control stations, train operators, and perform research in the aerospace field, offering many advantages over the use of real UAVs. Most existing UAV flight simulators have been defined with specific purposes in mind and are not versatile. In this work, a UAV flight simulator compatible with the ESA Simulation Model Portability (SMP) standard is developed. This standard has proven applications in the field of space simulation and promotes modularity and versatility of the simulation. A common model specification compatible with SMP was established.

This common model specification was used for developing a fundamental set of models for simulating the operation of a UAV. The models implemented simulate vehicle dynamics, aerodynamics, propulsion, gravitation, landing gears, and atmosphere. A fundamental requirement for the development of the simulator was the promotion of versatility and scalability, which simplifies the addition of new models and the improvement and/or replacement of existing ones. This requirement had a direct impact in the software implemented. Interfaces with Google Earth and the FlightGear flight simulator have been established for visualizing the simulation output. A number of tests were performed to validate the operation of the simulator. Its applicability to a scenario of fire monitoring support was demonstrated.

1. Introduction

UAVs provide a series of advantages over conventional aircraft. By dealing away with the pilot, a great number of constraints no longer apply to the vehicle. It becomes possible to operate flying vehicles with a very broad range of sizes and shapes, which allows for a wider range of missions to be performed. Additionally, UAVs do not need to carry life support equipment, which potentially reduces both production and maintenance costs. There are also immediate advantages to the absence of the pilot itself. Long, repetitive missions become more effective, since the UAV does not become tired or distracted as a pilot would. The most important reason in many situations, however, is the reduction of the endangerment of human life.

Due to the fact that UAVs can have a very wide range of configurations, they can be adapted to fit very specific needs. As such, each UAV is often designed with a specific range of applications in mind, which can be civilian or military. The most common applications are activities of surveillance and observation. Specific civilian applications range from border patrol, public event security, maintenance and security of oil and gas pipelines, communications, and power lines, early warning against forest fires, coastline patrolling, search and rescue support, and environmental observation [1].

A UAV flight simulator can be a very valuable tool for supporting the development and operations of UAVs. A simulator can be used for testing the UAV control station, which offers an alternative to the use of real UAVs in these applications, possibly reducing their cost. As a design tool, a UAV simulator can be a precious aid in the early phases of the project of a vehicle and all throughout the project of a control system. Also, since it is possible to create a UAV flight simulator that a control system operator cannot distinguish from the real vehicle [2], a UAV simulator can be used as a training tool. This has the advantages of allowing training on the most various situations, simulating failure scenarios, and keeping real UAVs in active duty.

In this work, a series of components of a UAV flight simulator, SimUAV, were developed. The objective of SimUAV was to address the major shortcomings of typical UAV simulators, which are developed with a particular purpose in mind, such as military training for a specific UAV or scientific research in a particular scenario. SimUAV achieves its objective by providing a generic simulation platform – which can be used in a multitude of scenarios –, and a versatile software architecture, which allows for easy adding, updating, and removing of components. The versatility of SimUAV is achieved through its separation into components, which consist of simulation models and interfaces with external visualization tools. This component-based architecture was possible thanks to the support for the ESA SMP standard [3], which defines how the models interact with each other. Supporting SMP also ensured the compatibility with SMP-compliant simulation platforms, such as SIMSAT [4], which was used for running SimUAV.

The specific components of SimUAV developed in this work consist of simulation models and interfaces. The models simulate vehicle dynamics, aerodynamics, propulsion, gravitation, terrain reactions, atmosphere, fire propagation, and an algorithm for fire front following. The interfaces serve to communicate with the tools Google Earth [5] and FlightGear [6], allowing for easier interaction with the simulation as well as a clear understanding of the simulation outputs. The interfaces also provide a valuable validation tool. The remaining components necessary for SimUAV to operate were developed in [7]. They consist of an on-board automatic flight control system, a communications interface between the simulated UAV and a control station, and simulation models for the sensors and actuators.

The aforementioned fire propagation and fire front following models, which were developed in this work, serve to demonstrate the applicability of SimUAV to the scenario of fire fighting support. Examples of the use of UAVs in this scenario are described in [8] and [9]. UAVs can be very helpful for fire-fighters in the field by providing them with aerial imagery of the fire zone, allowing them to better coordinate their efforts and improving their efficiency.

The work developed is described in full in [10]. It was executed during an internship at Critical Software and is part of an internal research project. Due to this, the development of SimUAV followed the software development life cycle process implemented by Critical Software [11], which establishes a procedure to be followed when developing software that promotes the quality of the product. In accordance with this process, a series of tests were executed in order to validate the simulator.

2. Requirements Analysis

In accordance with the software development life cycle implemented by Critical Software, an analysis of the specific requirements for the UAV simulator was performed. The fundamental objective for the project was the development of a generic, SMP compliant UAV flight simulator, but it was necessary to determine which specific functionalities the simulator would support. From a high-level perspective, it was defined that it would be possible for the user to specify a flight plan for the UAV based on waypoints, each with a specific latitude, longitude, and altitude, and have the simulated UAV follow that flight plan. The simulated UAV would also be able to loiter over a specified area, as well as following discrete commands such as climbing to a certain altitude. Additionally, the applicability of the simulator to the scenario of fire fighting support was to be demonstrated by having the UAV follow a simulated fire front.

A specific technological requirement for the simulator was the compliance with the ESA SMP standard. The involvement of ESA in space simulation throughout the years has produced several simulators for its missions. These simulators are used for research, engineering operations preparation, and training. The models that comprise them are often developed in different operating systems and for running in different simulation platforms (these are responsible for managing the running of a simulation). As such, the models developed for a particular application are often incompatible with each other. This greatly limits the possibility of reusing them in other missions. A new simulator often requires models that have already been implemented in another simulator, such as the model for the dynamics of a satellite. A great deal of effort could be saved if these models were reused in the development of other simulators for different missions, which requires compatibility between them. The Simulation Model Portability (SMP) standard was developed in order to standardize the architecture of the models of a simulation, enhancing their reuse [3].

A compatible simulation platform is necessary for running an SMP-compliant simulation. The SIMSAT simulation platform version 3.0 [4] was selected for use in this project because it is provided free of charge by ESA to the European space industry. It is used by ESA in space simulation applications and is an established simulation platform. Version 3.0 is the latest and most advanced version of SIMSAT. It supports the SMP standard, which is a fundamental requirement for this project.

A civil application for UAVs that is rising in popularity is the support of fire fighting activities through the monitoring of forest areas. Examples of applications of UAVs in this scenario are described in [8] and [9]. UAVs can provide for effective surveillance of large areas, warning of fires in their early stages as well as keeping the fire-fighters in the field up to date with the evolution of the fire front, allowing them to better coordinate their efforts and improving their efficiency. Because of its importance, the fire monitoring scenario was selected for the purposes of demonstrating the operation of the simulation and the ease with which it is modified for application to a particular situation. This requirement demanded the implementation of additional models, in order to simulate the scenario.

3. Simulator Architecture Overview

The architecture developed for the simulator promotes the encapsulation of each simulated reality in a model that is as independent from other models as it is possible. Each model receives the data it needs from other models, simulates the reality it represents by processing the data, and outputs the values of that processing. The simulation models that were identified as necessary for simulating a UAV in the scenario of fire monitoring support are depicted in Figure 1, along with the visualization interfaces, the UAV control station, and the XML configuration files which were implemented for configuring the various models.

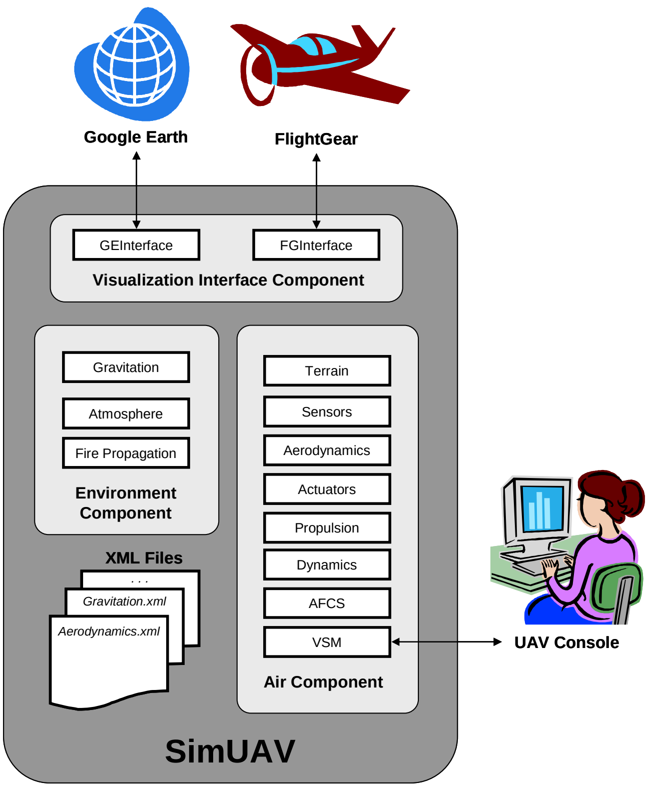

Figure 1: High-level architecture of SimUAV

The simulation models that were implemented in this work are described in the paragraphs that follow.

Dynamics model: this is responsible for receiving forces and torques applied to the UAV by the Aerodynamics, Propulsion, Gravitation, and Terrain models. It determines the evolution of the position of the UAV through time, reporting to other models the values of certain parameters such as velocities and accelerations. Since this model can be considered the heart of the simulation, it was decided that it would be as generic and as a powerful as possible, allowing for its future reuse. Therefore, the UAV is modelled as a rigid body according to the formulation in [12]. The user is able to specify the use of the explicit Euler or 4th order Runge-Kutta methods for propagation of the states in time. UAV data such as inertial properties and initial states are read by this model from an XML file.

Aerodynamics model: the resulting aerodynamic force and torque acting on the UAV at each instant is provided by this model to the Dynamics model. In order to determine these parameters, the Aerodynamics model obtains the current flight conditions from the Dynamics model, the actuator positions from the Actuators model, and the wind velocity and local air density from the Atmospheric model. In order to improve the value of simulation results by increasing their accuracy, both linear and non-linear aerodynamic data are supported. This data will be described in XML files.

Propulsion model: this simulates the behaviour of the UAV propulsion system by providing the Dynamics model with the value of the generated force and torque. The value of the current desired throttle setting is read from the AFCS model (responsible for the autopilot tasks). Due to the wide range of UAV engines and propellers available and the complexity associated with their modelling, it was decided that, in this work, a simple propulsion model would be implemented in the form of a thrust force that is the product of the throttle setting (in percent) by the maximum thrust generated by the UAV being simulated. This maximum value is read from an XML file. To simulate a delay in the response of the engine, the first order lag filter described in [7] is used, with the time constant read from an XML file.

Gravitation model: the Earth gravitational attraction is simulated by this model. The UAV is considered as a point mass, since the alternative of considering higher terms of the gravitational potential is important in the case of satellites but not of UAVs. The gravitational attraction force is calculated and sent to the Dynamics model. In order to improve simulation accuracy, an algorithm based on the Earth Gravitational Model 1996 (EGM96), referred by [12] as delivering a very good level of accuracy, is implemented. In this case the algorithm uses only the largest coefficient of the potential function of EGM96. This algorithm is used instead of considering the Earth as a concentrated mass in order to promote the reuse of the Gravitation model in future applications requiring this level of precision.

Terrain model: this is responsible for providing the Dynamics model with the total force and torque generated by the UAV landing gears. For this, it must read from the Dynamics model the current position of the UAV and must have information available on the layout of the terrain where the UAV is intended to land. The ultimate goal of this model is to allow the possibility of simulating take-off and landing in specified runways. To simulate the landing gears, this model reads from an XML file the necessary data such as landing gear spring and damping constants, as well as position.

Atmospheric model: the values of the atmospheric parameters are determined in this model. They are necessary throughout the simulation. Current density is used by the Aerodynamics model, as well as by the Sensors model. Temperature and pressure are also used by the Sensors model. An ISA model is implemented in order to obtain these parameters. For this, the value of the altitude of the UAV is read from the Dynamics model. The existence of wind in the simulation is performed by reading a constant wind speed and heading from an XML file. This is used by the Aerodynamics model.

Fire Propagation model: this component is responsible for simulating the propagation of a fire in an area, using a very simple algorithm. The algorithm selected was the EMBYR model [13], which is adequate for use in large-scale fires. EMBYR does not simulate the physics behind a forest fire, instead dividing an area into cells and relying on a probabilistic model for the propagation of fire from cell to cell. EMBYR is used instead of a more complex model because the actual simulation of fire behaviour is not the essence of this project. The fire propagation data is only necessary for the purposes of the Fire Front Following model. The data generated by this model is provided to the Fire Front Following model as well as to the Google Earth interface, for visualization purposes.

Fire Front Following model: Based on the data generated by the Fire Propagation model, this model determines, at each instant, the waypoint that the UAV must fly to in order to follow the front of the fire. The algorithm is based on [14]. The path of the UAV becomes a curve approximating the shape of the front. The UAV is automatically commanded to follow the generated waypoints. As the UAV moves and updates its visualization of the fire front area, new waypoints are generated, which results in the UAV being continuously pursing a moving waypoint. The main objective of this model is to demonstrate the applicability of SimUAV in a specific scenario.

The two visualization interfaces in Figure 1 were also implemented in this work. The interface with Google Earth allows for visualizing the exact position of the UAV in the planet, in terms of latitude, longitude, and altitude. The interface with FlightGear, a popular open-source flight simulator, is used for displaying the position and attitude of the UAV.

The remaining simulation models of Figure 1 which are not part of this work were developed in [7]. Those models simulate the on-board flight computer (AFCS), the communications interface (VSM), and the sensors and actuators of the UAV.

4. Validation Results

Validation is a critical part of the software development process, demonstrating whether the software serves the purpose it was created for. SimUAV was subjected to a series of tests, and some of the most relevant results are presented here. The conducted tests consisted of unit and integration tests, focusing on the functionality offered by small groups of components, and on system tests, verifying the behaviour of SimUAV as a whole.

Initial tests focused on the positioning of the UAV. Using the Google Earth interface as input, the position of the UAV was set at a specific location by using a placemark. The placemark was output to an XML file, which was processed by the simulation in order to initialize the Dynamics model. The position data, after being processed by the Dynamics model, was output to the visualization interfaces, FlightGear (Figure 2, left) and Google Earth (Figure 2, right). The two visual representations showed very similar positions. Differences in the UAV position in Figure 2 are mostly due to the separate ways in which the two programs draw the image, to the different camera perspective (which is defined by hand), and to inaccuracies in the positioning of the maps used by the two programs.

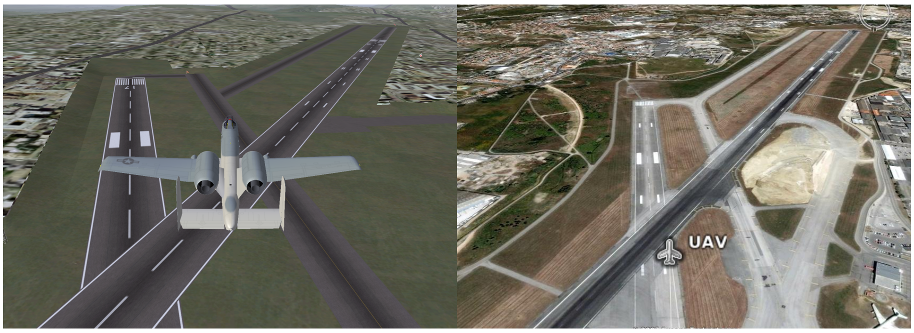

Figure 2: Comparison between positioning in FlightGear (left) and Google Earth (right)

The functionality of following a flight plan and performing loiter was tested by defining a set of placemarks in Google Earth, which were loaded into the simulation. The UAV was commanded to follow the placemarks, each placemark being a waypoint. Upon reaching the final waypoint, the UAV entered loiter mode (see Figure 3).

The application of SimUAV to a fire monitoring scenario was verified by setting a region for simulation of fire propagation, and then defining a flight plan which crossed that region. The UAV autonomously began following the front of the fire, starting at the instant that it detected its presence. The evolution of the fire and the resulting path described by the UAV are depicted in Figure 4.

Figure 3: Flight plan following and loiter mode

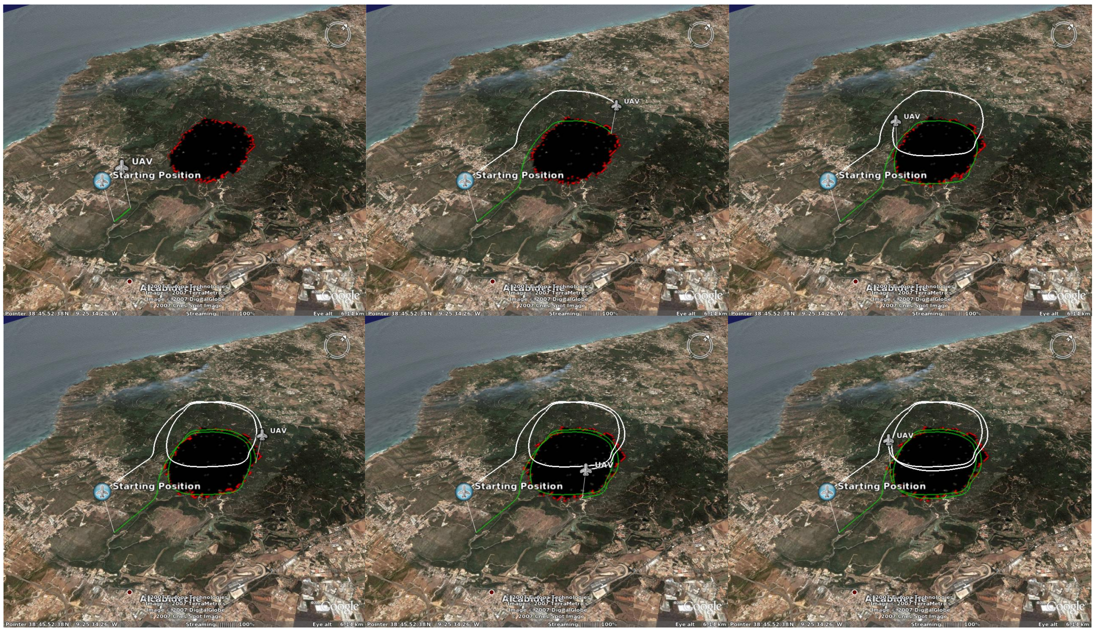

Figure 4: Evolution in time of the trajectory of the UAV and its ground projection in the scenario of fire monitoring

5. Conclusions

This work has resulted in the implementation of an SMP compliant UAV flight simulator, SimUAV. The development of the simulator focused on modularity and versatility, and followed the software development life cycle as implemented by Critical Software, which promoted final product quality.

There are two major contributions in this work. First, the implementation of SimUAV resulted in a platform of fundamental simulation models based on a modular architecture. This platform is generic and highly versatile since it allows implementation of new models and changes to existing ones, and can therefore easily be customized or improved for application to specific UAVs and scenarios. The second contribution is that the ESA SMP standard has been applied to the field of aeronautics for the first time. This standard was developed for use in space simulation but has now been shown to be applicable to UAV simulation.

The implementation of each model answered the requirements laid out in the requirements engineering phase. Generic, fundamental models for UAV simulation were constructed in a mission and vehicle independent way. The models for the demonstration scenario of fire monitoring were also implemented. The independence of each model, thanks to the SMP standard, greatly aided in the implementation phase since development could focus on one model at a time. While some of the models developed offer a very high level of functionality, some are not as advanced. However, they can easily be upgraded or replaced because of the modular architecture. The use of XML configuration files allows for easy definition of vehicle and mission parameters.

Tests were executed on individual models, as well as on groups of models and on the simulation as a whole. The models behaved as expected, allowing for the simulation of waypoint following, loiter over waypoint, and following discrete commands. The applicability of SimUAV to the fire monitoring scenario was demonstrated through the simulation of a fire and the implementation of an algorithm for following its front and monitoring its evolution. The two interfaces developed, with Google Earth and FlightGear, were found to be very valuable tools for visualizing the simulation output. These tools were an invaluable aid to the development process, since the data visualization provided by SIMSAT is extremely limited. SimUAV was also shown to operate correctly when connected to the STANAG 4586 compliant UAV control station developed in [7].

Further work should focus on an upgrade for compliance with the upcoming SMP 2 standard. SMP 2 offers many improvements over the present version of SMP, especially in terms of architecture. The upgrade will ensure that SimUAV will continue to be up-to-date with ESA simulation standards. Additionally, the essential models for UAV flight simulation which were implemented should be improved upon. The most pertinent upgrades at this time are simulation of engine and propeller performance and modelling of complex wind effects. Ground and runway modelling should be implemented for the Terrain model. Finally, new simulation models should be implemented, such as specific payload models (e.g. cameras and other sensors) and a model for situational awareness.

References

[1] M. Okrent (2005), *European Civil Unmanned Air Vehicle Roadmap Volume 3 – Strategic Research Agenda*
http://www.uavnet.com
Retrieved on February 25, 2007

[2] J. Blacklock and L. Zalcman (2006), The ADGESIM, Global Hawk UAV, Testbed Simulator. *Proceedings, SimTecT 2006*

[3] European Space Agency (2003), *Simulation Model Portability Handbook*

[4] J. Eggleston and D. Donald (2005), *SIMSAT Simulator Designer's User Manual*

[5] Google (2007), *Google Earth*
http://earth.google.com/
Retrieved on February 25, 2007

[6] C. L. Olson (2007), *FlightGear Flight Simulator*
http://earth.google.com/
Retrieved on February 25, 2007

[7] J. Mendes (2007), *UAV Flight Simulator based on ESA Infrastructure: Generic graphical user interface for UAV Command and Control,* Master thesis, Instituto Superior Técnico da Universidade Técnica de Lisboa

[8] S. J. Rasmussen and P. R. Chandler (2002), MultiUAV: A multiple UAV simulation for investigation of cooperative control. *Proceedings, Winter Simulation Conference, 2002, 869-877*

[9] Mercury Computer Systems (2007), *Mercury Computer Systems Introduces Complete UAV Solution for Cost-Effective Remote Sensing Application*
http://www.mc.com/mediacenter/pr/news_details.cfm?press_id=2007_01_17_0900_101537_556815pr.cfm
Retrieved on February 25, 2007

[10] A. Almeida (2007), *UAV Flight Simulator based on ESA Infrastructure: Flight simulation models compliant with SMP standard,* Master thesis, Instituto Superior Técnico da Universidade Técnica de Lisboa

[11] *Software Development Process* (2006), Critical Software internal document

[12] B. L. Stevens and F. L. Lewis (2003), *Aircraft Control and Simulation*. Wiley

[13] W. W. Hargrove, R. H. Gardner, M. G. Turner, W. H. Romme, D. G. Despain (2000), Simulating fire patterns in heterogeneous landscapes. *Ecological Modelling 135 (2000) 243-263*

[14] D. W. Casbeer, R. W. Beard, T. W. McLain, S. Li, R. K. Mehra (2005), Forest Fire Monitoring With Multiple Small UAVs. *Proceedings, American Control Conference, 2005, 3530-3535*