Free Software for the Modelling and Simulation of a mini-UAV

An Analysis

Tomáš Vogeltanz, Roman Jašek
Department of Informatics and Artificial Intelligence
Tomas Bata University in Zlín, Faculty of Applied Informatics
nám. T.G. Masaryka 5555, 760 01 Zlín, CZECH REPUBLIC
vogeltanz@fai.utb.cz, jasek@fai.utb.cz

***Abstract*—This paper presents an analysis of free software which can be used for the modelling and simulation of a mini-UAV (Unmanned Aerial Vehicle). Every UAV design, construction, implementation and test is unique and presents different challenges to engineers. Modelling and simulation software can decrease the time and costs needed to development of any UAV. First part of paper is about general information of the UAV. The fundamentals of airplane flight mechanics are mentioned. The following section briefly describes the modelling and simulation of the UAV and a flight dynamics model. The main section summarizes free software for the modelling and simulation of the UAV. There is described the following software: Digital Datcom, JSBSim, FlightGear, and OpenEaagles. Other software is mentioned only. Finally, the results of the analysis are discussed in conclusion.**

Keywords—aircraft; airplane; analysis; flight dynamics model; free software; modelling; simulation; UAV

I. INTRODUCTION

Mini-UAVs are a relatively inexpensive alternative to manned aircraft for a variety of applications, including aerial reconnaissance, environmental monitoring, agriculture, surveying, and safety. [1] [2] [3]

The research community does not clearly define the term mini-UAV (or small-UAV). [2] Instead of creation new definition about weight and proportions of the mini-UAV, we can simply say, a UAV is mini (or small) if the strength of an average man suffice for lifting and moving the UAV without any trouble. For the remainder of this paper, the term mini-UAV is implied.

The flight-testing of an aircraft is a well-documented engineering procedure. However, every aircraft design, construction, implementation and test is unique and presents different challenges to engineers, pilots, and test team. The same can be said about UAVs, but the criteria for UAVs can be different from manned aircraft. [2] [4] [5]

This work was supported by the Internal Grant Agency at Tomas Bata University in Zlín, project No. IGA/FAI/2014/006.

Current standards for handling qualities apply to only piloted aircraft and there are no specific standards for UAV handling qualities. Relevant article discussing dynamic stability and handling qualities for small UAVs is [4]. [2]

Any UAV system depends on its mission and range, however, most UAV systems include: airframe and propulsion systems, control systems and sensors to fly the UAV, sensors to collect information, launch and recovery systems, data links to get collected information from the UAV and send commands to it, and a ground control station. [3] [5]

UAVs promise greater precision, but the auto-pilot system which keeps the vehicle in the air and in control is critical to the success of UAV systems. The ability to test autopilot systems in a virtual environment is significant for development. A reliable UAV simulation process which can be adapted for different aircraft would provide a platform for developing autopilot systems with reduced dependence on expensive field trials. In many cases, testing newly developed autopilot systems in a virtual environment is the only way to guarantee absolute safety. Additionally the model allows better repeatability in testing. [1] [2] [4]

II. FUNDAMENTALS OF AIRPLANE FLIGHT MECHANICS

The basic aeronautical concepts and definitions required to be understood in order to deal with the flight dynamics model. Flight mechanics is the application of Newton's laws (1) and (2) to the study of vehicle trajectories (performance), stability, and aerodynamic control. [3] [6]

$$F = m \cdot a \tag{1}$$

$$M = I \cdot \alpha \tag{2}$$

The equations of motion are composed of translational (force) equations (1) and rotational (moment) equations (2) and are called the six degree of freedom (6DOF) equations of motion. The aircraft can move in three dimensions in space and can rotate about three axes. Motion caused by gravity, propulsion, and aerodynamic forces contribute to the forces

ISBN: 978-1-61804-247-7

and moments which act upon the body of the airplane. Fig. 1 shows the three axes and the forces and moments acting on an aircraft. The center of gravity of the aircraft is at the intersection of the axes. [1] [2] [3] [6] [7]

For trajectory analysis (performance), the translational equations are uncoupled from the rotational equations by assuming that the airplane rotational rates are small and that control surface deflections do not affect forces. The translational equations are referred to as the three degree of freedom (3DOF) equations of motion. [6] [7]

The forces of lift, weight, drag, thrust, and side force act along the axes, forcing the aircraft to move in the axes direction. On the other hand, the three moments, yaw, roll, and pitch force the aircraft to turn around the axes. [1] [2] [3]

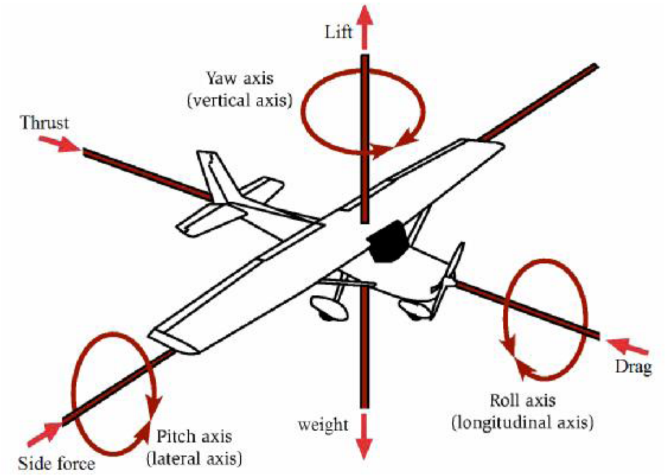

Fig. 1. The forces and moments acting on an aircraft

Table I. defines each of the state variables. Although the forces and moments are relative to the atmosphere, the state variables are defined relative to the earth. [2]

TABLE I. FLIGHT PATH COMPONENTS VARIABLE DEFINITION

Variable	***Symbol***
Roll Rate (rad / sec)	P
Pitch Rate (rad / sec)	Q
Yaw Rate (rad / sec)	R
Velocity (m / sec)	V
Sideslip Angle (rad)	β
Angle of Attack (rad)	α
Bank Angle (rad)	μ
Flight-Path Angle (rad)	γ
Heading Angle (rad)	χ
North Position (m)	ξ
East Position (m)	η
Altitude (m)	h

V, χ, and γ represent the magnitude of the velocity vector, heading angle, and flight path angle respectively. P, Q, and R represent the components of angular velocity; roll, pitch, and yaw. The position of the aircraft relative to the earth in Cartesian coordinates is ξ, η, and h. Body attitude relative to the velocity vector are μ, β, and α. [2]

All longitudinal motion occurs in the xz-plane of the aircraft. Stability along the longitudinal axis is both static and dynamic. Longitudinal static stability is the tendency of the airplane to return to pitch equilibrium following an angle of attack disturbance. Static stability is the aircraft's initial response to an input command. In the case of flight test, aircraft considered statically stable immediately tend to return to its steady level flight condition. The airplane is statically stable if the center of gravity is located at the wing aerodynamic center. When viewed over time, the aircraft is dynamically stable if it tends to return to steady level flight condition. [2] [6] [7]

III. THE MODELLING AND SIMULATION OF A UAV

The modelling and simulating of a UAV accurately is not an easy task, due to the need to calculate many parameters either by physical measurements, experiments, or estimation from available data of similar UAV or by software tools. One of the big challenges is calculating aerodynamic coefficients. Aerodynamic coefficients characterize the response of the proposed vehicle based on its geometry. [1] [3]

Several major assumptions are often made for the modelling and simulation of the aircraft. First, the aircraft is rigid. Although aircraft are truly elastic in nature, modelling the flexibility of the UAV should not contribute significantly to the research. Second, the earth is an inertial reference frame. Third, aircraft mass properties are constant throughout the simulation. For UAV modelling, it can be assumed the aircraft has constant mass over a flight. Finally, the aircraft has a plane of symmetry. The first and third assumptions allow for the treatment of the aircraft as a point mass. This assumption is a satisfactory approximation for UAV models. [1] [2]

For the modelling and simulation of a UAV the following items must be created: [3]

- A complete Flight Dynamics Model (FDM)
- A 3D graphical model (only if 3D visualization of the UAV is needed)
- An effective autopilot
- A way to identify flight route
- Autonomous flight simulation

The FDM is the physics/math model which defines the movement of an aircraft under the forces and moments applied to it using the various control mechanisms and from the forces of nature. The FDM includes development of a physical, inertial, and aerodynamic model representing the UAV. The FDM processes parameters from all input information. By manipulating input variables mathematically, an FDM predicts the future states of an aircraft. [1] [2] [3]

ISBN: 978-1-61804-247-7

The best equations to use to completely and accurately model an aircraft's true motion are nonlinear fully coupled ordinary differential equations. With these equations of motion, UAV response to any commanded inputs or wind disturbances is accurately modeled. [2]

However, a software model developed from first principles has unknown accuracy. For the model which is supposed to be used for real simulation, it is necessary to include implementation, verification and validation into its development process. [1]

The 3D graphical model is necessary only if 3D visualization of the UAV is needed. 3D Visualization can give us a better view of the simulation than numbers and graphs alone. [3]

The effective autopilot is very important for the good control of the UAV. Autonomous flight and stability of the UAV depend on the autopilot. If the autopilot is designed and implemented wrongly, the UAV can crash very easily. Every crash can increase distrust of the UAV and of its using in cities. [1] [2] [4] [7]

The way to identify flight route means to simulate system similar to the GPS or the GPS itself. [3]

Autonomous flight simulation depends on all things above. Additionally, use cases and activity diagrams of simulation have to be designed to define what should be found and how it should be found. Next, the settings of simulation parameters (e.g. density, gravity, airspeed, and altitude) must be defined and then the simulation itself can be run with or without visualization. [2] [3]

IV. Free Software

There is a lot of free software for the modelling and simulation of the aircraft on the Internet. Most of it can be also used for the modelling and simulation of the UAV. This paper cannot include and analyze all free software and therefore the most interesting free software only is described in detail. Other software is mentioned only.

A. Public Domain Aeronautical Software (PDAS)

For many years the Air Force, Navy, NASA and educational institutions have sponsored the development of computer software that is useful to aeronautical engineers, airplane designers, and aviation technicians. [8]

Public Domain Aeronautical Software (PDAS) was founded to make this valuable software available to the aeronautical community for use on desktop computers. These programs include descriptions and complete public domain source code. The source code is not copyrighted and may be used in whole or in part in any of aeronautical studies. Many programs have sample cases (both input and output). Some of the programs on the website are noted as work-in-progress, indicating that they are lacking in instructions or documentation or do not run properly. [8] [9]

There are many interesting applications, e.g. Atmosphere which characterizes the 1976 standard atmosphere to 1000 km altitude, including nonstandard atmosphere routines (hot, cold, polar, tropical), Aeroelastic Analysis which estimates the planform and aero-elastic effect on stability derivatives and induced drag, Flutter Analysis by Strip Theory which has been developed for rapidly predicting flutter of finite-span, swept or unswept wings at subsonic to hypersonic speeds, Induced Drag from Span Load Distribution which is a popular algorithm for computing the span load distribution on a planar wing when only a few sparse values of the loading are known, NACA Airfoils in which the coordinates of 4-digit, 4-digit-modified, 5-digit, 6-series, and 16-series airfoils may be accurately calculated, PANAIR which is a state of the art computer program developed to predict inviscid subsonic and supersonic flows about an arbitrary configuration by means of a higher order panel method, and more. [9]

There is no space to describe all applications in detail. Only Digital Datcom is described more because it has been used many times to real development of aircraft. [1] [2] [9]

1) Digital Datcom

The Stability and Control Data Compendium, Datcom for short, provides a systematic summary of methods for estimating basic stability and control derivatives. Datcom is over 1500 pages of detailed methodology to determine stability and control characteristics of a wide variety of aircraft and aircraft configurations. For any given flight condition and configuration the complete set of derivatives can be determined without resort to outside information. [2] [10]

Primarily intended for preliminary use, ahead of test data, it is designed to give an initial look at the stability performance of an aircraft design. In 1979, the Datcom was rewritten in FORTRAN IV computer language. Renamed the USAF Stability and Control Digital Datcom, it became an efficient, user-oriented computer program. However, it is not intended for use instead of wind tunnel or flight test data. [2]

In [1], the wind tunnel testing was performed in a low speed wind tunnel so accuracy was slightly compromised. There also has been said: "The results were sufficient to show that the aerodynamics coefficients determined by Datcom do not accurately represent the actual values. The experimental drag coefficients are higher than those predicted by the software model and this has a large effect on the accuracy of the flight dynamic model."

By interfacing Datcom with the FDM in the frontend, an aircraft model for any fix wing UAVs can be rapidly developed without wind tunnel testing. This feature significantly increases the repeatability of flight simulation and is found very useful for UAV preliminary designs where only a rough estimate of the vehicle's stability is required. [1]

Digital Datcom is used to calculate aerodynamic coefficients from first principles. By writing an input file containing all essential geometries of an aircraft, Datcom produces an output file with aerodynamic coefficients. The coefficients in the six degrees of freedom are drag, lift, side, pitching moment, rolling moment, and yawing moment coefficient. [1] [10]

Inputs to Datcom include desired flight conditions, aircraft attitudes, physical geometry, and desired outputs. Datcom treats inputs which represent a traditional wing-body-tail configuration and any control or high lift devices. Some nonstandard geometry can be treated as well. Datcom inputs were assumed for a straight-tapered or nonstraight-tapered wing. For the longitudinal characteristics, the program assumes a mid-wing configuration. [2] [10] [11]

B. *JSBSim Flight Dynamics Model*

JSBSim is open source software which allows users to access the internal of the models. JSBSim is a 6-DOF nonlinear flight dynamics model. JSBSim is generally considered as a very accurate FDM. It is the default FDM for the FlightGear flight simulator and for the OpenEaagles Simulation Framework. [1] [3] [12] [13] [14]

The open source feature of JSBsim has gained a lot of attention from researchers. A full 6-DOF simulator for flight simulation and pilot training was constructed at the University of Naples using JSBSim as its physics engine. JSBSim is also used to drive the motion-based research simulators in the Institute of Flight System Dynamics and Institute of Aeronautics and Astronautics at RWTH Aachen University in Germany. [1] [3] [12]

JSBSim is written in the C++ programming language. The use of an object-orientated approach makes JSBSim a generic simulator, and allows the modelling of any aircraft, missile, or rotorcraft. Particular aircraft flight control systems, propulsion, aerodynamics, landing gears, and autopilot are defined in eXtensible Markup Language (XML) format configuration files. [3] [12]

JSBSim can be run as a stand-alone application, or it can be run as an integrated part of the flight simulator which provides visual output. JSBSim models the rotational earth effects on the equation of motion and supports many data output formats such as to screen, socket, and file. [3]

The first step of the use of JSBSim to model and simulate a small autonomous UAV is to create the required JSBSim aircraft configuration files by using the Aeromatic. Aeromatic is a free web application and it is also included in source code of JSBSim. The next step is to make educated guesses to refine important sections in the created configuration files with the assistance of available data of similar UAV. [3] [13]

The Aeromatic takes inputs from the user such as system of measurements used, the aircraft name, type of aircraft, maximum take-off weight, wing span, length, wing area, landing gear layout, number of engines, engine type, and engine layout. Some values can be estimated, e.g. wing chord, wing area, inertia, and more. By clicking on the generate button, the main JSBSim configuration file will be generated in another window. The next step is to save the generated file by using save as, and giving the file a name. [3] [13]

The aircraft's metrics, the aircraft's airframe geometry, mass and inertia properties, landing gear positions and their ground reactions, flight control system, and aerodynamic characteristics are specified in the main aircraft configuration file. [3]

JSBSim can model different types of engines, for instance electric, piston, rocket and turbine engines. The aircraft's propulsion system is specified in two files, one for the engine, and the other for the thruster. These files are referred to in the propulsion section in the main aircraft specification file which allows the researcher to assign different kinds of engines and thrusters to the aircraft. The propulsion section specifies information about engine, thruster, and fuel tank. Aeromatic takes the engine power, maximum engine RPM, pitch condition, and propeller diameter from the user. [3] [13]

JSBSim provides components which can be connected together to model a flight control system for an aircraft. The flight control surfaces are elevator, right and left ailerons, and rudder. [3]

C. *FlightGear Flight Simulator*

FlightGear is an open-source flight simulator, written in the C++ programming language, to model and simulate an aircraft. Data visualization is another aspect considered while building the flight dynamics model. FlightGear supports many 3D formats. The common format used in FlightGear has "ac" extension. The FlightGear can produce a 3D graphic animation in real time and is connected to a FDM. The animation facility allows the UAV to be viewed from any angles, and provides absolute visual information on the UAV attitude and stability. Moreover, it models real world instrument behavior, and system failures. [1] [3] [14] [15]

FlightGear allows the user to access the internal properties and monitor any of its internal state variables. By editing configuration files it is possible to create sound effects, model animations, instrument animations and network protocols for approximately any situation. [3] [15]

FlightGear can communicate with external flight dynamics models, GPS receivers, external autopilot, control modules, other instances of FlightGear, and other software. [3] [15]

It is possible to choose between two primary FDMs: JSBSim and YASim. Actually, there is a third FDM named UIUC and based on LaRCsim originally written by the NASA but UIUC is no longer supported by default FlightGear builds. It is possible to add new dynamics models or even interface to external "proprietary" flight dynamics models. [15] [16]

YASim is an integrated part of FlightGear and uses a different approach than JSBSim by simulating the effect of the airflow on the different parts of an aircraft. The advantage of this approach is that it is possible to perform the simulation based on geometry and mass information combined with more commonly available performance numbers for an aircraft. This allows for quickly constructing a plausibly behaving aircraft that matches published performance numbers without requiring all the traditional aerodynamic test data. [15]

Except of FDM configuration files, other files are required for use with FlightGear flight simulator which include the electric system file, autopilot file, and 3D graphical model specification file. The final file required is a file to tie the previous files together in order to perform complete simulation using FlightGear. [3]

The electrical system file specifies the battery characteristics, and the lights and other parameters. [3]

ISBN: 978-1-61804-247-7

To fly the modelled UAV autonomously, a tuning process have to be made for the built-in generic PID (proportional, integral, and derivative) autopilot of FlightGear which has the ability to hold aircraft velocity, vertical aircraft speed, altitude, pitch angle, angle of attack, bank angle, and true heading. [3]

FlightGear implements a PID algorithm in a flexible way which makes it reusable with similar aircraft. Any number of PID controllers can be defined in the autopilot configuration file. A process value, reference point, any number of output values, and other tuning constants can be assigned to each controller. Cascading controllers can be implemented by specifying multiple PID controllers in which the output of the current stage is used as the input to the next stage. [3]

A flight path which contains a number of waypoints chosen over a selected area using Google Earth map can be constructed. In order to use the chosen waypoints with FlightGear navigation system, a unique ID can be assigned to each waypoint, and the FlightGear database can be altered to include the new waypoints with their IDs. [3]

The fixed waypoints are determined by latitude and longitude. When a waypoint is entered in the aircraft route during the simulation time, FlightGear checks the database to see if it is a valid fixed point or not. This data is stored in the compressed file called fix.dat which can be found in the directory FG_ROOT\ FlightGear\data\Navaids. [3]

D. OpenEaagles Simulation Framework

OpenEaagles is an open source multi-platform simulation framework targeted to help simulation engineers and software developers rapidly prototype and build robust, scalable, virtual, constructive, stand-alone, and distributed simulation applications. OpenEaagles is written in the C++ programming language and it has been used extensively to build applications that demand deterministic real-time performance or simply execute as fast as possible. This includes applications used to conduct human factor studies, operator training, or the development of complete distributed virtual simulation systems. OpenEaagles has also been used to build stand-alone and distributed constructive applications oriented at system performance analysis. Constructive-only simulation applications that do not need to meet time-critical deadlines can use models with even higher levels of fidelity. [17] [18]

It should be emphasized that OpenEaagles is a cycle or frame-based system, not a discrete-event simulator. This approach satisfies the requirements for which it is designed; namely, support for models of varying levels of fidelity including higher level “physics based” models, digital signal processing models and the ability to meet real-time performance requirements. Model state can be captured with state machines and state transitions can use the message passing mechanisms provided by the framework. [19] [20]

The framework embraces the Model-View-Controller (MVC) software design pattern by partitioning functional components into packages (see Fig.2 - packages with white/clear background indicate the use of a third party open source tool.). This concept is taken a step further by providing an abstract network interface so custom protocols can be implemented without affecting system models. Framework utilizes a number of other third party open source tools such as FLTK, Fox and wxWidgets for cross-platform GUI applications, and JSBSim as a high quality flight dynamics model. [18] [19] [20]

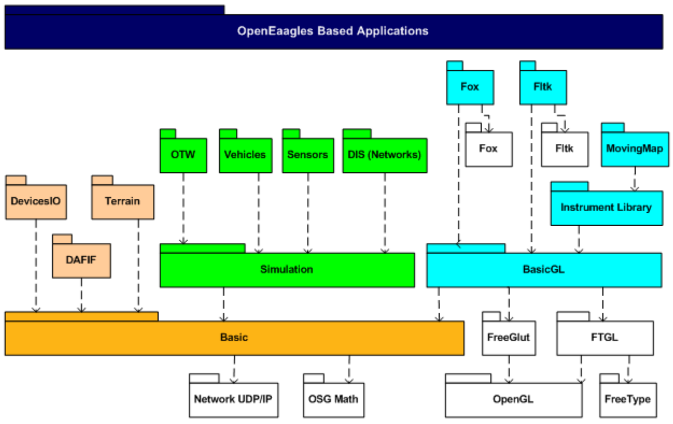

Fig. 2. OpenEaagles Package Hierarchy

The simulation side provides a wealth of capabilities including abstract classes for representing a variety of entity types such as aircraft, ships, tanks, ground vehicles, space vehicles and even lifeforms. A complete radar modeling environment is also included in the simulation hierarchy. [20]

The graphics hierarchy provides a mature collection of classes that can be used to render instruments that are commonly used in operator-vehicle interface displays. A sampling of the instrument objects available include: analog dials for altimeters, dials for direction finders, speedometer dials, and even landing gear indicators. [20]

Distributed applications can interoperate with other systems and simulations through Distributed Interactive Simulation (DIS) and/or High Level Architecture (HLA) interfaces. Numerous DIS compliant distributed simulation applications have been built using this framework as the foundation. [17] [18] [19]

Specific applications using the framework to support simulation activities include representative F-16 cockpits, an Unmanned Aerial Vehicle (UAV) ground control station (Predator MQ-9), Integrated Air Defense Systems (IADS) and a futuristic battle manager. [19]

The framework is routinely compiled with Microsoft Visual Studio for the Windows environment and GCC for Linux. [19]

Project files for Codelite and Codeblocks development environments can be generated by OE_source_ROOT\build\premake\make.bat file. For the Codelite project files generation, “%Premake% codelite” have to be added to the end of make.bat file.

E. Other Software

XFOIL is an interactive program for the design and analysis of subsonic isolated airfoils. It consists of a collection of menu-driven routines which perform various useful functions. [21]

ISBN: 978-1-61804-247-7

Tornado for Octave is a Vortex Lattice Method for linear aerodynamic wing design applications in conceptual aircraft design or in aeronautical education. By modelling all lifting surfaces as thin plates, Tornado can solve for most aerodynamic derivatives for a wide range of aircraft geometries. [22]

Aerospace blockset for the Scilab/XCos is an external module providing aerospace palette. It is based on CelestLab aerospace library. Although Scilab/XCos and aerospace blockset are very interesting compensation for Matlab/Simulink, aerospace blockset is designed rather for satellites than aircraft nowadays. [23]

V. Conclusion

This paper has described the most interesting free software for the modelling and simulation of any UAV. The fundamentals of airplane flight mechanics and the basics of the modelling and simulation of a UAV have also been mentioned.

UAVs can be used in civilian operations such as searching for missing people, aerial photography, monitoring of illegal activities, traffic control, and monitoring of pollution. The future of any UAV program is open to a wide range of research topics: collision avoidance, autonomous formation flight, navigation without the use of GPS, autonomous detection of the potential criminal etc.

It has been explained that FDM and Flight Simulators are used in the development process of a UAV to test its design and control systems. In any FDM, the coefficients of lift, drag, and side forces and roll, pitch, and yaw moments have to be estimated in order to calculate the forces and moments acting on the UAV.

Public Domain Aeronautical Software contains many interesting applications. Especially Digital Datcom has been used in real projects many times. Digital Datcom gives very good way how to calculate aerodynamic coefficients from first principles but the creation of an input file may be complicated.

The combination of JSBSim Flight Dynamic Model and FlightGear Flight Simulator provide an excellent base for building the simulation environment. To model the UAV in JSBSim Flight Dynamic Model and simulate it with FlightGear Flight Simulator, essential configuration files have to be constructed. These files include main UAV configuration, engine configuration, propeller configuration, electric system configuration, 3D model configuration, and an autonomous flight autopilot configuration file. All these files are tied together in the top level configuration file.

OpenEaagles framework is designed for the simulation application developer; it is not an application itself. OpenEaagles is a cycle-based system, not a discrete-event simulator. OpenEaagles utilizes a number of other third party open source tools, e.g. JSBSim. OpenEaagles has been used extensively to build applications that demand deterministic real-time performance. This includes applications used to conduct human factor studies, operator training, or the development of complete distributed virtual simulation systems.

References

[1] X.Q. Chen, Y.Q. Chen and J.G. Chase, Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions. Croatia: In-Teh, May 2009, pp.177-201.

[2] N. M. Jodeh, Development of Autonomous Unmanned Aerial Vehicle. Ohio, USA: Wright-Patterson Air Force Base, March 2006, 185 p.

[3] T. Abdunabi, Modelling and Autonomous Flight Simulation of a Small Unmanned Aerial Vehicle. Sheffield, UK: The University of Sheffield, August 2006, 61 p.

[4] T. M. Foster, Dynamic Stability and Handling Qualities of Small Unmanned-Aerial-Vehicles. Brigham, USA: Brigham Young University, April 2005, 125 p.

[5] R. Austin, Unmanned Air Systems: UAVs Design, Development and Deployment. Wiltshire, UK: Wiley, 2010, 332 p.

[6] D. G. Hull, Fundamentals of Airplane Flight Mechanics. Springer, 2007, 298 p.

[7] T.V. Chelaru, V. Pana, A. Chelaru, "Dynamics and flight control of the UAV formations," WSEAS Transactions on Systems and Control, vol. 4 no. 4, pp. 198-210, April 2009.

[8] R. Carmichael. (2013, January 31). Public Domain Aeronautical Software. [Online]. Available: http://www.pdas.com

[9] R. Carmichael. (2013, March 28). Public Domain Aeronautical Software: Contents. [Online]. Available: http://www.pdas.com/contents15.html

[10] R. Carmichael. (2013, April 21). Description of Digital Datcom. [Online]. Available: http://www.pdas.com/datcomDescription.html

[11] R. Carmichael. (2013, February 7). Addressable Configurations in Digital Datcom. [Online]. Available: http://www.pdas.com/datcomTable1.html

[12] JSBSim contributors. JSBSim Open Source Flight Dynamics Model. [Online]. Available: http://jsbsim.sourceforge.net/

[13] JSBSim contributors. (2005, December 31). Aeromatic. [Online]. Available: http://jsbsim.sourceforge.net/aeromatic2.html

[14] FlightGear contributors. FlightGear Flight Simulator: Introduction. [Online]. Available: http://www.flightgear.org/about/

[15] FlightGear contributors. FlightGear Flight Simulator: Features. [Online]. Available: http://www.flightgear.org/about/features/

[16] FlightGear contributors. (2014, May 12). FlightGear Wiki - UIUC. [Online]. Available: http://wiki.flightgear.org/UIUC

[17] D. Hodson. (2014, January 10). OpenEaagles Simulation Framework. [Online]. Available: http://www.openeaagles.org

[18] D. Hodson. (2012, December 03). OpenEaagles Simulation Framework: Overview. [Online]. Available: http://www.openeaagles.org/wiki/doku.php?id=overview:overview

[19] D. Hodson, D. Gehl and R. Baldwin, "Building Distributed Simulations Utilizing the EAAGLES Framework," Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), vol. 5, no. 2, May 2006.

[20] D. Hodson, "OPENEAAGLES, An Open Source Simulation Framework," A Publication of the AIAA Modeling and Simulation Technical Committee, vol. 1, no. 1, January 2008.

[21] M. Drela. (2013, December 23). XFOIL: Subsonic Airfoil Development System. [Online]. Available: http://web.mit.edu/drela/Public/web/xfoil/

[22] Redhammer Consulting Ltd. (2010). TORNADO. [Online]. Available: http://www.redhammer.se/tornado/index.html

[23] P. Zagórski and C. David. (2013). Aerospace Blockset for Xcos. [Online]. Available: http://forge.scilab.org/index.php/p/aerospace-blockset/