JSBSim Open-Source Flight Dynamics Model for Fixed-Wing Unmanned Aerial Vehicle Applications

O. Cereceda
Memorial University of Newfoundland
St. John's, NL, Canada

L. Rolland
University of West Scotland
Glasgow, Scotland, U.K.

S. O'Young
Memorial University of Newfoundland
St. John's, NL, Canada

Abstract- **Simulation packages provide a valuable framework or environment to study the interaction between aircraft, including Unmanned Aerial Vehicles (UAVs), in the existent air traffic in near mid-air collision (NMAC) scenarios. The proposed simulation package is based on the open-source JSBSim Flight Dynamics Model (FDM) and our developments represent the UAVs with the required precision to model those interactions. This tool has been validated and tested in UAV computer models and is currently being used in 4D encounters and avoidance maneuvers. The final objective of this paper is to provide a simplified model version of the current package, the including the minimum requirements for the design of a UAV in JSBSim, and guide any modeler through the UAV computer design task.**

Index Terms— Unmanned Aerial Vehicles, JSBSim FDM, UAV modelling and simulation.

I. INTRODUCTION

A trusted computer environment is crucial for the correct implementation of dynamics simulation models in aviation whereby small and large Unmanned Aerial Vehicles (UAVs) pose a threat to other aircraft. The recent UAV integration into the airspace has resulted in a wide range of studies about its role and impact including Sense and Avoid (SAA) strategies. Near mid-air collisions (NMAC) and SAA methods are events and processes that need to be studied under simulated environments as real tests remain limited in occurrence as they are regulated by national (Transport Canada) and international aviation administrations. Computer models must provide sufficiently high levels of fidelity with a performance proven to be the sufficiently precise to the real UAV under certain conditions. In particular, its dynamic performance is given by the Flight Dynamics Model (FDM), which is the combination of physical and mathematical equations that express the aircraft dynamics in a simulator.

JSBSim is a 6-DoF FDM that has been utilized in aeronautics since 1996 [1]. It is an open-source software with a large library in constant development that makes the dynamic model versatile and easy to use by any designer; the library includes general aircraft, jet planes, airliners, rockets, zeppelins, etc. The programming code is C++ and the configuration files are implemented in XML-format. It runs with most of the operating systems, per contra, it was initially developed for general aircraft and although it has been used in UAV modelling in previous projects [2]–[4], there still exists

a lack of reliability analysis for UAV models in JSBSim [5]. The most remarkable work done on JSBSim can be found in the motion base simulator at the University of Naples, Italy [6] and the human pilot math model with JSBSim as the 6-DoF simulation core developed by the U.S. Department of Transportation [7].

II. SIMULATION STRUCTURE

The versatility of JSBSim is one of its highlights and strongest attributes; the software can be downloaded from its website [8]. In this section and along this paper, two different options of its implementation are described. Both require a second software: FlightGear [9]. It is an open-source flight simulator which has been in constant development since 1997 (its last version, FlightGear 2017.3.1, was released on September 20th, 2017). The aircraft visual models and the real world including scenery and airports are open to any design included in its online library or can be designed from scratch by the user. It is usually run along with JSBSim as their roles are complementary; JSBSim provides the dynamics of the system and FlightGear the visual performance.

A. Standalone mode. Scripting

The standalone simulation mode of JSBSim is straightforward once the full package including the source code and the set of engines and aircraft is downloaded. The source code is considered stable and to be kept the same whereas new aircraft and other models are uploaded to the repository once they are stable and verified.

A JSBSim standalone simulation can be summarized by the following blocks (also shown in Figure 1):

The *JSBSim source code (B1)* is formed by the full 6-DoF FDM including the dynamics of the system and all the models that interact with the aircraft or computer model.

The *Script file (B2)* is expressed in XML-format and it describes the tasks to be performed by the computer model. It is based on events and conditions.

The *Initialization file (B3)* includes the information related to the initial state of the aircraft. It can be called either from *B2* or *B1*.

The *Aircraft configuration file (B4)* contains all the parameters representing a specific aircraft. The main file includes, among others, metrics, mass and aerodynamics. The

propulsion system formed by the external *Engine (B4.1)* and *Thrust (B4.2)* files is called from the block *B4*. Additionally, some *extra files (B4.3)* can be added depending on the final purpose of the simulation. Some examples are an autopilot, a guidance system or more specific information about elements such as sensors or a control system.

The type of *Output (B5)* is defined at the end of the aircraft configuration file *(B4)* and generated by the source code. The outputs can be generated in series of datalogs that can be later processed and plotted, or come in the form of a visual performance using a complementary software such as FlightGear or OpenEaagles.

An *additional datalog (B6)* with specific parameters or set of parameters to a particular package; this feature has not been tested yet.

The minimum and most basic configuration consists on the source code *(B1)* with the aircraft configuration file including the propulsion system *(B4, B4.1, B4.2)*, the script files defining the task *(B2)*, the initialization file *(B3)* and the output generation files or interface *(B5)*. The corresponding blocks are highlighted in light blue whereas the white ones are complementary depending on the task to be completed.
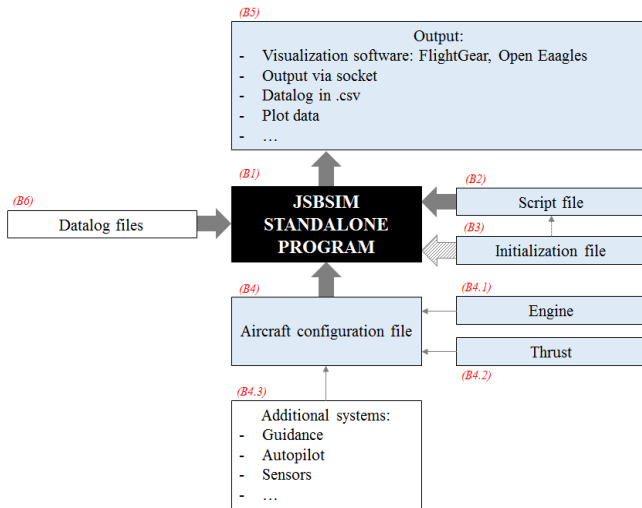

Figure 1. JSBSim standalone mode structure.

B. Integration in FlightGear

In this simulation setup, FlightGear hosts the simulation execution. In its early developments, JSBSim was the main representative aircraft FDM. The software can be structured as shown in Figure 2:
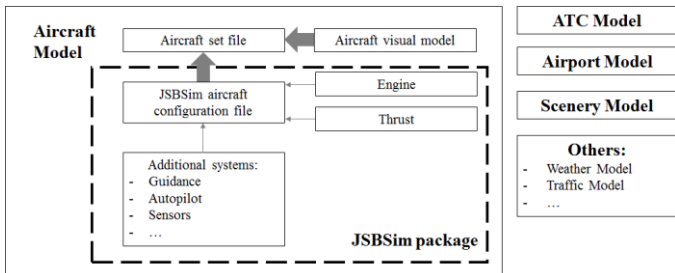

Figure 2. FlightGear block structure.

FlightGear is a built-up flight simulator that only needs the user to select an aircraft, scenario and conditions in the launching interface. Contrary to the standalone mode, there is no need to download the entire JSBSim package as only the aircraft configuration and the propulsion files are needed. The base package only includes basic aircraft but a full list of available models to download can be found online on its website [9]. Unfortunately, the current online aircraft library is the only one available for downloading and it does not include any UAV; the graphic UAV model introduced in this paper was developed using another software and later imported.

To run the software, copy the required files in the source folder that includes the aircraft information in FlightGear. Then, the selected aircraft should appear in the list of available aircraft. The model will start on the ground or in the air and can be initiated and controlled by a flight controller, a joystick, or with the computer keyboard.

III. MATHEMATICAL BASIS

The mathematical model used in JSBSim comes from the Newton's laws of motion where the position, velocities and accelerations are solved from the total forces and moments assuming that the aircraft is a 6-DoF rigid body. The purpose of this section is to highlight the most relevant aspects of the flight dynamics in JSBSim, not providing the full mathematical background for the understanding of the flight dynamics; for that purpose, consult the references [10]–[12].

A. Frames of reference

The frames of reference are utilized to model the different elements during flight. For more information about their definition in JSBSim, consult its Manual [1].

B. Forces and Moments

JSBSim obtains the aerodynamic forces and moments using the coefficient buildup method, meaning that all the contributions to the generation of a specific force or moment are calculated and summed in order to obtain that total force/moment. The coefficients can be taken from flight tests, calculated by hand, or by approximation, using a computer software. JSBSim on its website offers the option of using its Aeromatic tool v2.0 [8] for the generation of an approximate set of coefficients calculated from the aircraft dimensions.

In a 6-DoF aircraft model, *Lift*, *Drag* and *Side* forces and *Rolling (L)*, *Pitching (M)* and *Yawing (N)* moments will fully express the dynamics of the system. These are expressed by the following:

$$
\begin{aligned}
Lift &= q\,S\,C_L & L &= q\,S\,b\,C_l \\
Drag &= q\,S\,C_D & M &= q\,S\,c\,C_m \\
Side &= q\,S\,C_Y & N &= q\,S\,b\,C_n
\end{aligned}
\tag{1}
$$

Where q is the dynamic pressure, S is the wing area, b the wing span, c the wing chord and C_L, C_D, C_Y, C_l, C_m and C_n are the *Lift*, *Drag*, *Side force*, *Rolling*, *Pitching* and *Yawing moment* coefficients respectively.

JSBSim calculates each force/coefficient from all the contributions to the force expressed as <function> within the

correspondent <axis> in the aircraft configuration file. Taking *Lift* as example:

```
<axis name="LIFT">
 <function name="aero/coefficient/CLwbh">
 <!--Lift_due_to_alpha-->

 <function name="aero/coefficient/CLadot">
 <!--Lift_due_to_alpha_rate-->

 <function name="aero/coefficient/CLq">
 <!-- Lift_due_to_pitch_rate -->

 <function name="aero/coefficient/CLDe">
 <!-- Lift_due_to_Elevator_Deflection -->
</axis>
```

Note that the Mach number and altitude elements are neglected from the general case because their dependencies are not significant in low altitude and slow airspeed.

Following the same idea as expressed for the *Lift* coefficient, the *Drag* coefficient C_D, the *Side force* coefficient C_Y, the *rolling moment* coefficient C_l, the *pitching moment* coefficient C_m and the *yawing moment* coefficient C_n are also calculated.

C. *Equations of Motion (EOM)*

The performance of the aircraft is defined by its motion variables expressed in the body frame by the linear velocities *[u, v, w]* and the angular rates *[p, q, r]* obtained from the forces $[F_X, F_Y, F_Z]$ and moments $[M_X, M_Y, M_Z]$ involved in the flight. Especially to understand the aircraft performance, the Euler angles *Roll*, *Pitch* and *Yaw*[1] *[ϕ, θ, ψ]*, are also directly obtained from the linear velocities frame transformation executed in the *FGAccelerations* file under the *FGModel* class.

IV. CLASSES, CLASS HIERARCHY AND MODEL CLASS

The JSBSim complete package includes all the source codes for a basic simulation; however, the user can create new systems depending on the required task. These systems might include Autopilot, GNC (Guidance, Navigation and Control) and specific on-board instruments.

The code is distributed in classes depending on their function. The collection of described high level classes could be classified in Executive class (*FGFDMExec),* Model class (*FGModel),* Math classes (*FGColumnVector3, FGQuaternion, FGTable,* etc), I/O and Initialization classes *(FGInputSocket, FGOutputTextFile*, etc.) and Basic classes (*FGJSBBase* and *FGState*). Focusing on the UAV computer model design, only the Model classes differ from the general aviation aircraft case; the rest of the classes include operators and simulation characteristics required for the correct implementation of JSBSim and are compulsory. The Model classes inherited from *FGModel* are listed below including a brief description and special conditions (if applicable) for the UAV case.

- The Input class (*FGinput*)[2] manages the inputs to the model with <input> elements in the aircraft configuration file.
- The Atmosphere class (*FGAtmosphere*)[2] models the 1976 standard atmosphere. Beware that the UAV case

will only consider low altitude (under 1000ft)[3] and the model could be simplified considering only the troposphere conditions. However, the authors recommend keeping this model as it is for future possible use.

- The FCS class (*FGFCS*) manages a collection of flight control classes defined by the aircraft components (surface control elements, throttle, autopilot). In a simple UAV simulation, the primary controls for the model are the surface command elements of the aircraft which include ailerons, elevator and rudder.
- The Propulsion class (*FGPropulsion*) manages from 0 to n number of engines (*FGEngine*). JSBSim includes different kinds of propulsion systems depending on the engines used to generate thrust. Only the piston (*FGPiston* → *FGEngine*) and electric models (*FGElectric* → *FGEngine*) would be used in case of UAV modelling. The thrust generation (*FGThruster* → *FGForce*) in the fixed-wing UAV case is given by the propeller (*FGPropeller* → *FGThruster*).
- The MassBalance class (*FGMassBalance*)[2] calculates the moments of Inertia, Center of Gravity (CG) and mass over time. This class is relevant in the UAV case when carrying a piston engine as the fuel consumption will highly change those values.
- The Aerodynamics class (*FGAerodynamics*)[2] is a collection of manager classes with individual force and moment definitions. When <aerodynamics> is called in the aircraft configuration file, this class handles the aerodynamic calculations obtaining the correspondent force and moment calculated for each of the axes.
- The Inertial class (*FGInertial*)[2] initializes the radius and reference acceleration values.
- The Ground Reactions class (*FGGroundReactions*) models the ground reactions defined in the aircraft configuration file as <ground_reactions>. JSBSim models the landing gear set as a spring/damper model with <spring_coeff> and <damping_coeff>. It also models retractable landing gear for contact matters but it is not applicable to most UAVs as their landing gear is non-retractable.
- The Aircraft class (*FGAircraft*)[2] gathers all systems together; it initializes the aircraft model, loading its properties, and <metrics> and it also obtains the contribution of each of the systems in the generation of forces and moments.
- The Propagate class (*FGPropagate*)[2] models the EOM giving the state of the vehicle from the forces and moments generated.
- The Auxiliary class (*FGAuxiliary*) models pilot sensed accelerations and other auxiliary parameters used for acceleration calculations in inertial space. This class is

---

[1]Which should not be confused with Rolling, Pitching and Yawing moments.
[2]This class is required in all types of aircraft and fixed-wing UAVs.

[3]This value might differ with the upcoming Transport Canada regulations to be implemented in 2018.

only required in case the UAV carries motion-based sensors onboard.

- The Output class (*FGOutput*)[2] handles the simulation output, including: CSV (Datalog in csv), SOCKET (Data sent to a socket output defined by IP address), FLIGHTGEAR (socket to FlightGear) and TABULAR (Columnar data).

When JSBSim is initialized, *FGFDMExec* is executed generating all the model's objects that will be uploaded with the information contained in the aircraft configuration file. This allows the assembly of the 6-DoF aircraft computer model. For a full collection of JSBSim classes, visit [8].

V. GIANT BIG STIK FIXED-WING UAV 6-DOF FDM

The Giant Big Stik R/C UAV is an aerobatic sport-scale aircraft belonging to the "Stik" series manufactured by GreatPlanes and powered by a fuel engine (Zenoah 26A). This model airplane has been widely used in the RAVEN group at Memorial University for many years. Its aerobatic characteristics are the most significant highlight of this model. The Giant Big Stik features obtained from the manufacturer's information [13] and the real model are the following:

TABLE I
GIANT BIG STIK PARAMETERS

<metrics>		<mass_balance>	
Wing area (ft^2)	10.538	Ixx (Slug*ft^2)	0.3046
Wing span (ft)	6.709	Iyy (Slug*ft^2)	0.4752
Chrod (ft)	1.148	Izz (Slug*ft^2)	0.7036
H tail area (ft^2)	1.69	Ixy (Slug*ft^2)	0
H tail arm (ft)	2.36	Ixz (Slug*ft^2)	0.0951
V tail area (ft^2)	1.05	Iyz (Slug*ft^2)	0
V tail arm (ft)	2.27	Empty weight (lbs)	13
AERORP (in)	[18.4961, 0, 2.5591]	CG (in)	[14.4881, 0, 0]
EYEPOINT (in)	[7.09, -18, 45]		

TABLE II
GIANT BIG STIK AERODYNAMIC COEFFICIENTS

	<aerodynamics>				
	Lift Coefficient		Drag Coefficient		Side Coefficient
	C_{L0}	0.0	C_{D0}	0.1	C_Y^β	-0.83
	C_L^α	5.32	D_i	0.0877		
	C_L^α	1.7				
	C_L^q	3.9				
C_L^δ	Elevator	-5.0	C_D^δ	0.0135	C_Y^δ	0.0
	Aileron	0.0		0.0302		-0.075
	Rudder	0.0		0.0303		0.1914
	Roll Coefficient		Pitch Coefficient		Yaw Coefficient
	C_l^β	-0.074	C_{m0}	0.15	C_n^β	0.071
	C_l^p	-0.41	C_m^α	-7.8	C_n^p	-0.0575
	C_l^r	0.107	C_m^α	-3.5	C_n^r	-0.12032
			C_m^q	-6.813		
C_l^δ	Elevator	0.0	C_m^δ	-0.458	C_n^δ	0.0
	Aileron	-0.2		0.0		0.0108
	Rudder	-0.107		0.0		-0.062

A. JSBSim standalone

As an example of the use of JSBSim in UAV applications, a simple task is defined where the model turns left using the ailerons at maximum deflection in open-loop. This simulation has been used in previous work [5] for validation purposes.

According to Figure 1, the blocks listed are developed:

1) *B1:* JSBSim full package; downloaded from the original source.

2) *B2*: Script file where the task to be carried is described. In this example, the aileron is set to it maximum value in 5secs and returns back to 0.0 in 6.5sec.

```xml
<use aircraft="GBS" initialize="reset00"/>
<run start="0" end="35" dt="0.01">

    <event name="Set engine throttle">

    <event name="Set aileron max. Turn left">
        <condition> simulation/sim-time-sec ge 5.0 </condition>
        <set name="fcs/aileron-cmd-norm" action="FG_STEP" value="-0.569" tc="1"/>
        <notify/>
    </event>

    <event name="Set aileron to zero">
        <condition> simulation/sim-time-sec ge 6.5 </condition>
        <set name="fcs/aileron-cmd-norm" action="FG_STEP"
value="0.0" tc="1"/>
        <notify/>
    </event>

</run>
```

3) *B3*: Initialization file. The simulation is held near Clarenville, Newfoundland and Labrador, Canada.

```xml
<initialize name="reset00">
    <vc unit="KTS"> 38.87689112646 </vc>
    <longitude unit="DEG"> -53.91752033 </longitude>
    <latitude unit="DEG">  48.2778129  </latitude>
    <phi unit="DEG">        0.0   </phi> <!-- Roll -->
    <theta unit="DEG">      0.0   </theta> <!-- Pitch -->
    <psi unit="DEG">        0.0   </psi> <!-- Yaw -->
    <altitude unit="FT">    984.252   </altitude>
    <hwind> 0.0 </hwind>
</initialize>
```

4) *B4*: Aircraft configuration file, including all the coefficients and parameters specific for the Giant Big Stik

```xml
<fdm_config name="GBS" version="2.0" release="ALPHA"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="http://jsbsim.sourceforge.net/JSBS
im.xsd">

    <fileheader>
        <author> Oihane Cereceda </author>
        <filecreationdate> 2015-11-11 </filecreationdate>
        <version> v4 </version>
        <description> Giant Big Stik JSBSim model
    v1: Files and values created from Aeromatic v0.82
    v2: Files modified according to more specific data
    v3: Validate model using scripts and plotting data
</description>
        v4: Output to FlightGear to visualize the performance
    </fileheader>

    <metrics>, <mass_balance>, <ground_reactions>, <propulsion>,
<flight_control name="FCS: unnamed">, <aerodynamics>

    <output name ="localhost" type="FLIGHTGEAR" port="5500"
protocol="UDP" rate="10"> </output>
    <output name="GBS_Out.csv" type="CSV" rate="100">

</fdm_config>
```

UAV described in Tables I and II.

5) *B5: Output.* Note that this is defined in the aircraft configuration file. For this example, two cases are considered where JSBSim provides an output file when the simulation has ended and a visualization while the simulation is running.
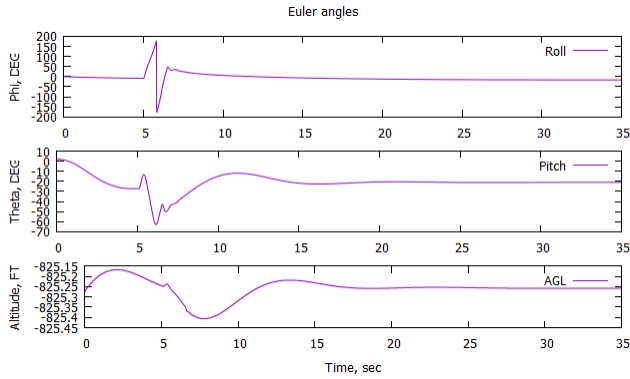
Figure 3. UAV Roll, Pitch and Altitude (gnuplot from JSBSim datalog output).

The simulation starts with the UAV in the air, needing a few seconds to get it to a stable situation before any command is sent to the model. This is also reflected when, in case of real flight, the R/C model switches from manual to autonomous mode. The simulation test shown here expresses a special case where the ailerons are pushed to its maximum value in order to evaluate the dynamics of the system when performing an extreme maneuver. Figure 3 shows how the Roll angle gets affected by changes in the aileron deflection; the aircraft quickly spins around and stabilizes as soon as the ailerons are back to 0.0. The coherence of the system is noticeable: when the aircraft rolls, the lift vertical component is not anymore balanced with the weight creating a loss in altitude and, therefore, in pitch angle. The response is pictured in Figure 4 where FlightGear v2.0.0 has been used to visualize the performance of the aircraft during the simulation.
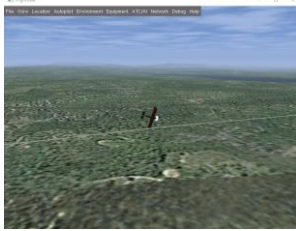

Figure 4. Giant Big Stik performing an aerobatic maneuver.

B. JSBSim incorporated into FlightGear

Following the setup description in *Section II.B* and using the Futaba Interlink Elite Controller to manually fly the computer model, some screenshots from the flight were taken and added below:
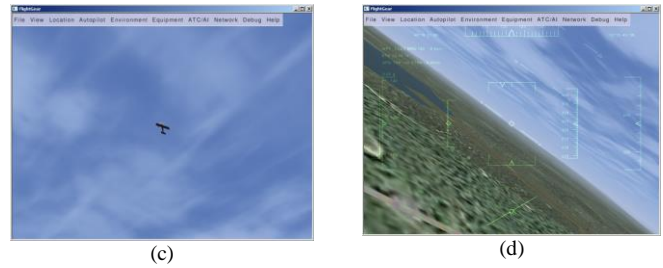

(a)

(b)

(c)

(d)

Figure 5. Giant Big Stik manual flight in FlightGear v2.0.0. (a) Chase view of a turning maneuver. (b) Straight flight over the runway. Chase view. (c) Fly-by view. (d) Cockpit view during flight.

VI. CONCLUSION AND FUTURE WORK

The JSBSim 6-DoF FDM offers a valuable and functional method for simulating manual and autonomous flights. The standalone mode offers the possibility of integrating UAVs with general aircraft in the same synchronized simulated environment. This special feature is currently being used in simulations to determine the requirements for extreme SAA maneuvers in NMAC scenarios.

Starting from a physics model of the Giant Big Stik expressed in JSBSim and adjusted from real flight data, the current line of research work intents to show that with differential dynamics between a manned and an unmanned system, the UAV provides the best avoidance leaving the piloted aircraft out of a collision risk at all times.

REFERENCES

[1] J. S. Berndt and JSBSim Development Team, "JSBSim, An open source, platform-independent, flight dynamics model in C++." 2011.
[2] C. Coopmans, M. Podhradsky, and N. V. Hoffer, "Software- and hardware-in-the-loop verification of flight dynamics model and flight control simulation of a fixed-wing unmanned aerial vehicle," in *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, 2015, pp. 115–122.
[3] S. K. Lim and C. C. Hao, "Modeling Unmanned Vehicle System," 2012.
[4] T. Vogeltanz and R. Jašek, "JSBSim library for flight dynamics modelling of a mini-UAV," in *International Conference on Numerical Analysis and Applied Mathematics 2014 (ICNAAM-2014)*, 2015.
[5] O. Cereceda, L. Rolland, and S. O'Young, "Validation discussion of an Unmanned Aerial Vehicle (UAV) using JSBSim Flight Dynamics Model compared to MATLAB/Simulink AeroSim Blockset," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 3989–3994.
[6] Universita degli Studi di Napoli Federico II, "ADAG | Aircraft Design & AeroFlightDynamics Group." [Online]. Available: http://www.adag.unina.it/english/index.html. [Accessed: 10-Oct-2017].
[7] Y. Zhang and S. Mcgovern, "Mathematical Models for Human Pilot Maneuvers in Aircraft Flight Simulation."
[8] JSBSim Development Team, "JSBSim Open Source Flight Dynamics Model." [Online]. Available: http://jsbsim.sourceforge.net/. [Accessed: 09-Oct-2017].
[9] "FlightGear Flight Simulator." [Online]. Available: http://www.flightgear.org/. [Accessed: 09-Oct-2017].
[10] W. Durham, *Aircraft Flight Dynamics and Control*. Wiley, 2013.
[11] R. H. Barnard and D. R. Philpott, *Aircraft flight. A description of the physical principles of aircraft flight*, 4th ed. Pearson Education, 2010.
[12] D. Allerton, *Principles of Flight Simulation*. 2009.
[13] Great Planes, "Giant Big Stik ARF Instruction Manual," no. 217, pp. 13–15, 2005.