HAL
open science

# The MAV3DSim: A Simulation Platform for Research, Education and Validation of UAV Controllers

Israel Lugo-Cárdenas, Gerardo Flores, Rogelio Lozano

## To cite this version:

Israel Lugo-Cárdenas, Gerardo Flores, Rogelio Lozano. The MAV3DSim: A Simulation Platform for Research, Education and Validation of UAV Controllers. 19th World Congress The International Federation of Automatic Control (IFAC 2014), Aug 2014, Cape Town, South Africa. pp.713-717, 10.1109/ICUAS.2013.6564761 . hal-01138152

**HAL Id: hal-01138152**

**https://hal.science/hal-01138152v1**

Submitted on 2 Apr 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Proceedings of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

# The MAV3DSim: A Simulation Platform for Research, Education and Validation of UAV Controllers ⋆

**Israel Lugo-Cárdenas** * **Gerardo Flores** ** **Rogelio Lozano** ***

* *Heudiasyc UMR CNRS 7253 Laboratory, Université de Technologie de Compiègne, Compiègne 60205, France (e-mail: icardenas@hds.utc.fr).*
** *Heudiasyc UMR CNRS 7253 Laboratory, Université de Technologie de Compiègne, Compiègne 60205, France (e-mail: gfloresc@hds.utc.fr).*
*** *Heudiasyc UMR CNRS 7253 Laboratory, UTC, Compiègne 60205, France, and UMI-LAFMIA CNRS 3175, Cinvestav 07300, Mexico (e-mail: rlozano@hds.utc.fr).*

**Abstract:**
Despite the substantial interest in UAVs, little attention has been paid to developing realistic simulators which represents a complete mathematical model of the UAV and the external variables, that is the reason why we present a Multi-Aerial Vehicle 3D Simulator (MAV3DSim) that will help with the validation of new controllers. It is multi-aerial because it has the possibility to simulate a fixed-wing or a quadrotor UAV. The multiple display options of the MAV3DSim provides a big help in the development of the controllers. Finally we present the previous implementation of two different controllers, for the fixed-wing a path following strategy is implemented while for the quadrotor an attitude and altitude controller, both strategies are implemented on the same simulation environment.

*Keywords:* UAV simulators, aircraft control, nonlinear control, verification, dynamic models.

## 1. INTRODUCTION

When investigating control algorithms for UAVs, it is important to simulate the effect of the proposed controller before being tested on a real platform. With this aim, it is required that UAV dynamic behavior is reproduced as close as possible to the real one in a simulation platform. Moreover, external conditions such as disturbances and obstacles, must be considered on the controller design. On the other hand, safety issues for operation of UAVs should be taken into consideration with the objective of minimizing risks of handling UAVs. Unfortunately, quite often a simulation framework that satisfies these requirements is not used by the UAV community; certainly a mechanism with the aforementioned properties would be an interesting tool for any researcher involved with UAVs. These observations prompted us to develop a simulation platform with the following features:

- Complete model of the UAV;
- Real parameters and models for each UAV component, i.e. control surfaces, motor and propeller model, etc.;
- Modeling of outdoor real scenarios such as wind gusts and obstacles;
- Possibility of choosing between different UAV models such as: conventional helicopter, fixed-wing UAV and quadrotors. Moreover, diverse kinds of each aforementioned UAV model can be simulated.

Few work has been done on development of complete model-based UAV simulators. For example, a real-time simulation of a quadrotor is presented in I.E.Putro, where the real-time simulation was performed in MATLAB/Simulink by means of the xPC Target, in which a pair of host PC and two PC targets were used. In R.C.B. Sampaio (2013) a commercial flight simulator has been utilized as the simulation engine for the quadrotor Pelican from Asc. Technologies; this fact represents a disadvantage due to the source code is not available for review and/or modification.

The main contribution of this paper is the development of the platform named MAV3DSim, which is capable of simulating realistic scenarios by using elaborated versions of UAV mathematical models. The MAV3DSim simulator allows the user to test controllers before being implemented on the UAV platform; in this manner, the control engineer can design controllers by taking simplified mathematical models and then test such controllers on the complete model provided by the simulator. On the other hand, the MAV3DSim simulator has several characteristics which improves its efficiency, such as the ability of tuning gains online and the visualisation of any variable involved in the system, also it has the possibility to export all the information to a Matlab compatible format for plotting and future analysis.

⋆ Research supported in part by the Mexican National Council for Science and Technology (CONACYT) under grants 314448 and 314456

The remainder of this paper is organized as follows. Section 2 describes the MAV3DSim simulator environment starting with the simulation engine based on a multi vehicle open source simulator which is fully integrated to the MAV3DSim simulator and continuing with the description of visual representation of the data generated by the controller and the simulation engine. Section 3 presents some application examples using the MAV3DSim platform. Finally, conclusions and future work are presented in Section 4.

## 2. MAV3DSIM SIMULATOR

The description of the MAV3DSim simulation environment is addressed in this section. A custom C# .Net based application was developed named MAV3DSim (Multi-Aerial Vehicle 3D Simulator). The MAV3DSim allows read/write operation to/from the simulation engine from which we could receive all the emulated sensor data sent from the simulator, processes it, generates a control input and then send it back to the simulator as depicted in Fig. 1. The MAV3DSim consist of three main components, the simulator engine, the controller computation and the data visualization interface. The simulation engine is in charge of the numeric integration of the dynamic equations of the UAV, here we can choose between a fixed wing UAV and a quadrotor for use it in the simulation. The input of the simulation engine is the controllers output and using the controller's output, it computes the new vehicle state and send it back to the controller as sensor data. The controller computation is the component where the control law is programmed, here it receives the current state of the vehicle as an input, then calculates the controller output and send it to the simulation engine. The data visualization interface looks like a ground station type of application, where all the variables of the state vector of the UAV can be represented in different ways, and with the addition of a 3D visualization of the attitude and position of the UAV in a 3D scenario.

### *2.1 Simulation Engine*

The simulation engine is based on the CRRCSim Simulator [CRRCsim-community (2014)], an open source simulator, which means that we can view and modify the source code of the simulator. The purpose of the original simulator was to aid the radio-controller (RC) enthusiast to learn how to fly an RC airplane or quadcopter. The RC community made a great effort in the development of this simulator to be close enough to reality. We have

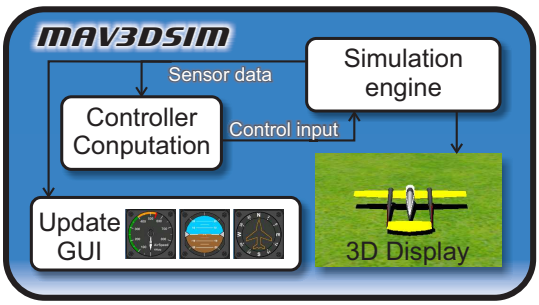


Fig. 1. Communication scheme of the MAV3DSim.

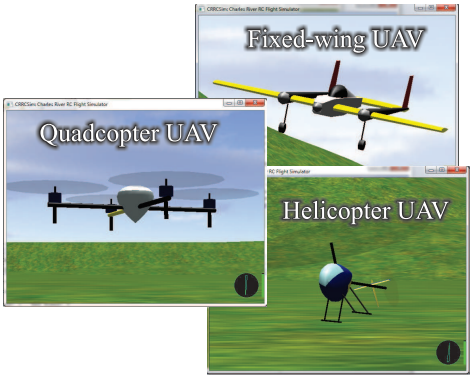


Fig. 2. Different types of UAV available in the CRRCSim simulator.

extended this work by incorporating this simulator into the MAV3DSim solution as seen in Fig. 7 . There are three different types of aircraft or UAV, a fixed wing airplane, a conventional helicopter and a quadcopter (Fig. 2) and, as it is an open source simulator, the user can add as many different UAV types as desired. It is important to mention that the CRRCSim simulator implements the complete nonlinear model in six degrees of freedom (6DoF) and inside the source code we can find that this simulator is based on a NASA report for a standard kinematic model for flight simulation in McFarland (1975).

The simulation engine sends and receive information in a standard data packet used by the MNAV100CA Robotics Sensor Suites in Crossbow Technology (2005). The sensor data that the simulator sends can be taking as the emulation of an inertial measurement unit(IMU) sending inertial gyroscope, accelerometer and magnetometer, a GPS radio in the latitude/longitude format, altitude and airspeed. It can receive commands to move the control surfaces aileron elevators, rudder, and the thrust in the case of the fixed-wing UAV and in the case of the quadrotor UAV it changes the rotational velocity of each motor.

*Mathematical Model* In this section we present the mathematical model used by the simulation engine for the airplane and the quadrotor.

These equations are derived and fully described in McFarland (1975), this reference was found inside the source code of the CRRCSim simulator and we have validated its correct implementation by comparing the programmed source code with the equations described in the NASA report McFarland (1975).

For any aircraft in the simulation engine, the state vector $\mathbf{x}$ is a 12×1 vector representing the vehicle location, the inertial velocity, the vehicle attitude and the vehicle rotational velocity. The rotational and inertial velocities are referenced in the body frame while the attitude and vehicle location are referenced to an inertial frame.

$$\mathbf{x} = [\lambda \ \tau \ R \ V_N \ V_E \ V_D \ \phi \ \theta \ \psi \ p \ q \ r]$$

Translation equations

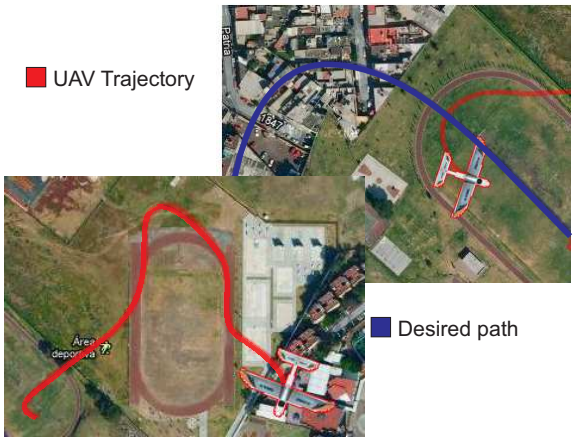


Fig. 3. Map visualization of the UAV desired and actual trayectory.

$$\begin{aligned} \dot{\lambda} &= \frac{V_N}{R} \\ \dot{\tau} &= \frac{V_E}{R\cos\lambda} \\ \dot{R} &= -V_D \end{aligned} \tag{1}$$

where $\lambda$ is the latitude and $\tau$ is the longitude. $R$ is the distance from center of the earth to the vehicle.

Attitude equations

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2}$$

These equations represent the time derivative of the rotation expressed in quaternions, and to obtain an equivalent representation of the angle from the quaternion expressed in the Euler angles$(\phi, \theta, \psi)$ we have the following relations

$$\begin{aligned} \tan\phi &= \frac{2(q_2q_3 + q_0q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \\ \sin\theta &= -2(q_1q_3 - q_0q_2) \\ \tan\psi &= \frac{2(q_1q_2 + q_0q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \end{aligned} \tag{3}$$

The time differential inertial velocity vector $[V_N, V_E, V_D]$ is computed using the following equations

$$\begin{aligned} \dot{V}_N &= \frac{F_N}{m} + \frac{V_N V_D - V_E^2 \tan\lambda}{R} \\ \dot{V}_E &= \frac{F_E}{m} + \frac{V_E V_D + V_N V_E \tan\lambda}{R} \\ \dot{V}_D &= \frac{F_D + F_G}{m} - \frac{V_N^2 + V_E^2}{R} \end{aligned} \tag{4}$$

where $F_N, F_E$ and $F_D$ are the components of the applied force vector on the vehicles center of gravity and $F_G$ is the force of gravity.

The rotational velocity dynamic are presented in the following equations

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (C_1 r + C_2 p)\, q \\ C_7 rp + C_6 \left(r^2 - p^2\right) \\ (C_8 p + C_9 r)\, q \end{bmatrix} + \begin{bmatrix} C_3 & 0 & C_4 \\ 0 & C_5 & 0 \\ C_4 & 0 & C_{10} \end{bmatrix} \begin{bmatrix} L \\ M \\ N \end{bmatrix} \tag{5}$$

where

$$\begin{aligned} C_0 &= \left(I_{xx}I_{zz} - I_{xz}^2\right)^{-1} \\ C_1 &= C_0\left(\left(I_{yy} - I_{zz}\right) I_{zz} - I_{xz}^2\right) \\ C_2 &= C_0 I_{xz}\left(I_{xx} - I_{yy} + I_{zz}\right) \\ C_3 &= C_0 I_{zz} \\ C_4 &= C_0 I_{xz} \\ C_5 &= I_{yy}^{-1} \\ C_6 &= C_5 I_{xz} \\ C_7 &= C_5\left(I_{zz} - I_{xx}\right) \\ C_8 &= C_0\left(\left(I_{xx} - I_{yy}\right) I_{xx} + I_{xz}^2\right) \\ C_9 &= C_0 I_{xz}\left(I_{yy} - I_{zy} - I_{xx}\right) \\ C_{10} &= C_0 I_{xx} \end{aligned}$$

and $I_{xx}$, $I_{yy}$, and $I_{zz}$ are the moments of inertia about the x, y, and z body axes, respectively, and $I_{xy}$ , $I_xz$, and $I_{yz}$ are the products of inertia in the $x-y$, $x-z$, and $y-z$ body axis planes, respectively. $L$, $M$, and $N$ being the aerodynamic total moments about the $x$, $y$, and $z$ body axes.

The fixed-wing and the quadrotor use the same mathematical model, the only difference is in the computation of the forces $[F_N, F_E, F_D]$ and moments $[L, M, N]$, the fixed-wing moments and forces mainly depends on the aerodynamic coefficients and the deflection of the control surfaces and the quadrotor dynamics depends on the moments and forces generated by the rotors.

### 2.2 Graphic User Interface

A very important feature is the display of the state vector $\mathbf{x}$ delivered by the simulator. There are several display option which inlcude a Google Maps display for position the UAV in some point over the surface of the Earth, and it can be seen the path generated by the UAV an

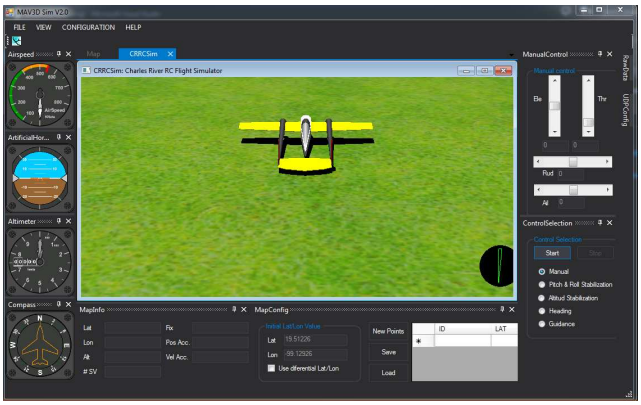


Fig. 4. Integration of the CRRCSim to the MAV3DSim as the simulation engine.

the possibility to add a desired or reference path along with waypoints for trajectory tracking as seen in Fig. 3. Avionics instruments like those used in commercial aircrafts are used to display the state of the UAV, such as an altimeter which indicates the altitude relative to a reference level at which the aircraft is flying. An attitude indicator, which shows the position of the longitudinal and transversal aircraft axes with respect to the natural horizon, this is obtained by reading the roll and pitch angles. A heading indicator, which displays the aircraft heading with respect to magnetic north. And finally it includes an airspeed indicator, which gives us the aircraft speed relative to the surrounding air. Those avionics instruments are depicted in Fig. 4.Gain tuning is a time consuming tasks and to aid reducing the time of the gain tuning it has the possibility to change the gains online and to see the effect of the new gains in the simulation.

## 3. EXAMPLES

Much of the work in the literature considers a simplified system model to investigate a control algorithm. This is a common practice and simplifies the process of control design. However, in a lot of cases the controller must be validated on a real platform, which does not necessarily match the model. In few cases, the designer test the controller on the complete system model, this is due to the difficulty to represent the behavior in any simulation software such as MATLAB SIMULINK.

Therefore, a simulating tool which can represent in an accurate manner the real system behavior is needed. Thus, starting from a reduced model, we develop controllers for two kinds of UAVs, with the aim of showing that such controllers can operate even in the non-reduced system. To this end, the operation of the MNAV3DSim simulator is showed by two examples: the trajectory control problem in a fixed-wing UAV, and the stabilization problem in a Quad-rotor UAV.

*Fixed-wing UAV* In this section a Lyapunov-based control law, developed in our previous work [Flores et al. (May, 2013)], is presented to steer a fixed-wing UAV along a desired path. The key idea behind the proposed strategy, is to minimize the error of the path-following trajectory by using a virtual particle, which should be tracked along the path. For this purpose, the particle speed is controlled providing an extra degree of freedom.

Considering Fig. 5, the key idea behind the path-following controller relies on reducing the distance between the aircraft's center of mass $p$ and the the point $q$ on the path to zero, as well as the angle between the airspeed vector and the tangent to the path at $q$. To accomplish these objectives, we introduce a virtual particle moving along the geometric path at a velocity $\dot{s}$. Consider a frame attached to such particle, this frame plays the role of a body axis of the virtual particle, and is the so called Serret-Frenet frame denoted by $\mathcal{F}$ Micaelli and Samson (1993). It is worth noting that the particle velocity evolves according to a conveniently defined control law $\dot{s}$, yielding an extra controller design parameter.

The Dubins Aircraft model Chitsaz and LaValle (2007) is used to develop the controller. This is a simplified

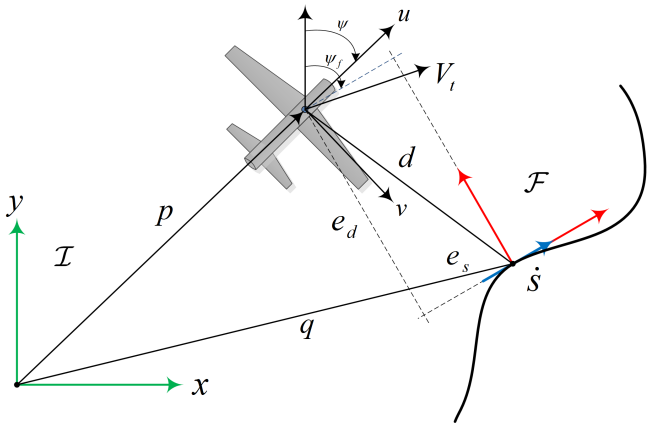


Fig. 5. Path following control problem schema.

kinematic version of the lateral dynamics of an airplane. The Dubins aeroplane is described by the subsequent relations

$$\begin{aligned} \dot{x} &= V_t \cos\psi \\ \dot{y} &= V_t \sin\psi \\ \dot{\psi} &= \omega \end{aligned} \tag{6}$$

where $x$ and $y$ denotes the inertial position of the aircraft, $\psi$ is the heading angle, $\omega$ is the heading rate, $\phi$ is the roll angle, $V_t$ is the airspeed, i.e. the speed of an aircraft relative to the air.

Details regarding the control strategy are omitted in this paper for reasons of brevity, but the reader can find a more detailed explanation in Flores et al. (May, 2013).

We have programmed this path-following technique into the MAV3DSim to validate it before going to the next step that would be the implementation on an experimental platform. Although the controller is designed by using the Dubins model Chitsaz and LaValle (2007), which is a simplified version of the complete model of the aircraft, the simulations performed in the MAV3DSim, which uses the complete nonlinear model described in Section 2.1.1 show a desired performance of the control law (Fig. 6).

*Quadrotor UAV* The Quad-rotor dynamic model presented in Section 2.1.1 incorporates in addition to the actuator action, propellers dynamics and the gyroscopic effects resulting from the rigid body. For purpose of control, we omit the influence of these effects and then, only

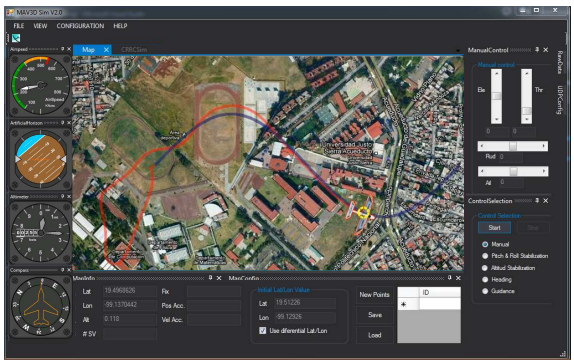

Fig. 6. Path following control problem schema.

the motor's action is considered. Furthermore, a near hover flight scenario is treated; with the aim to simplify the control design, the pitch and roll angle are considered equal to zero for the x-position and y-position control, respectively.

To develop the controller, we will use the bilinear model (7) instead of the nonlinear version, since in general the vehicle operates in areas where $|\theta| \leq \pi/2$ and $|\phi| \leq \pi/2$, these constraints are satisfied even when the nonlinear model is used together with a feedback control Bouabdallah and Siegwart (2005).

$$\begin{array}{rcl|rcl} \dot{x}_1 &=& x_2 & \dot{\theta}_1 &=& \theta_2 \\ \dot{x}_2 &=& -\theta u_1 & \dot{\theta}_2 &=& u_2 \\ \dot{y}_1 &=& y_2 & \dot{\phi}_1 &=& \phi_2 \\ \dot{y}_2 &=& \phi u_1 & \dot{\phi}_2 &=& u_3 \\ \dot{z}_1 &=& z_2 & \dot{\psi}_1 &=& \psi_2 \\ \dot{z}_2 &=& 1 - u_1 & \dot{\psi}_2 &=& u_4 \end{array} \tag{7}$$

The proposed control strategy is based on the idea that the global system (7) is constituted of two subsystems, the attitude dynamics and the position dynamics, each one with a time-scale separation between them Flores and Lozano (2013). From this fact, it is possible to propose a hierarchical control scheme where the position controller outputs desired attitude angles ($\theta_d$, $\phi_d$ and $\psi_d$) which are the angles to be tracked by the orientation controllers.

The movement in the $x - y$ plane is generated by orientating the vehicle's thrust vector in the direction of the desired displacement. Then, $\theta_d$ and $\phi_d$ behaves as virtual controllers for the positioning dynamics. The control proposed for $z$ an $x$, respectively, are defined by

$$u_1 = k_{pz}(z_1 - z_{1_d}) + k_{vz}(z_2 - z_{2_d}) + 1 \tag{8}$$

$$\theta_d = \frac{k_{px}(x_1 - x_{1_d}) + k_{vx}(x_2 - x_{2_d})}{u_1} \tag{9}$$

where $k_{vx}$, $k_{px}$, $k_{pz}$ and $k_{vz}$ are positive real numbers.

Using the control described above we could successfully implement it in the MAV3Dsim for the stabilization of the quadcopter and the control of the altitude. In the MAV3DSim the gain tuning was also implemented and accelerate this process with the online gain tuning feature. Figure 6 shows the simulation performed in the MAV3Dsim.

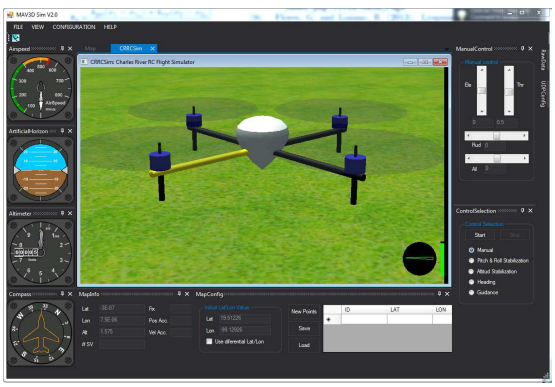

Fig. 7. Integration of the CRRCSim to the MAV3DSim as the simulation engine.

In this section two different control techniques has been applied to the same simulation environment, a path following strategy for the fixed wing UAV and an attitude and altitude controller for the quadrotor. Both simulations can be seen in `http://www.youtube.com/watch?v=L-xaW_Ej-Bg`

## 4. CONCLUSIONS AND FUTURE WORK

A simulation platform is a powerful tool for the development and validation of different controllers. Here we present a simulation environment the MAV3DSim simulator that has proven to be a great candidate for the validation of different controllers on different UAV models, a path following technique implemented for the airplane and an attitude and altitude controller for the quadrotor. Future work will address the problem of robustness in presence of wind gusts. Also, our current work involves the development of a experimental platform that will communicate with the MAV3DSim and perform Hardware-in-the-Loop (HIL) simulations and then use it to test the controllers implemented in the MAV3DSim. As the MAV3DSim is designed as a ground station type of application, the experimental platform will send all the data obtained from the different sensors and display it in the MAV3DSim graphic interface. In the future, other type of aerial vehicles such as coaxial helicopters or a hybrid configuration (airplane-quadrotor), could be implemented using the simulator MAV3DSim.

## REFERENCES

Bouabdallah, S. and Siegwart, R. (2005). Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *In Proceedings of IEEE Int. Conf. on Robotics and Automation*, 2247–2252. Barcelona, Spain.

Chitsaz, H. and LaValle, S. (2007). Time-optimal paths for a dubins airplane. In *Proc. IEEE Conference on Decision and Control (CDC'2007)*, 2379–2384. New Orleans, LA, USA.

Crossbow Technology, I. (2005). Mnav100ca user's manual. 4145 N.First Street,San Jose,CA 95134.

CRRCsim-community (2014). Crrcsim. URL `http://sourceforge.net/apps/mediawiki/crrcsim/index.php?title=Main_Page`.

Flores, G. and Lozano, R. (2013). Lyapunov-based controller using singular perturbation theory: An application on a mini-uav. In *Proc. IEEE American Control Conference (ACC'2013)*, 1599–1604. Washington, DC.

Flores, G., Lugo-Cardenas, I., and Lozano, R. (May, 2013). A nonlinear path-following strategy for a fixed-wing mav. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1014–1021. Grand Hyatt Atlanta, Atlanta, GA.

McFarland, R.E. (1975). A standard kinematic model for flight simulation at nasa-ames. *National Aeronautics and Space Administration.*

Micaelli, A. and Samson, C. (1993). Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Technical Report 2097, INRIA.

R.C.B. Sampaio, M. Becker, A.S.L.F. (2013). Fvms: A novel sil approach on the evaluation of controllers for autonomous mav. *IEEEAC.*