Team MAVion entry in the IMAV'17 outdoor challenge – A tail-sitting trajectory-tracking μUAV

Leandro R. Lustosa[1], Jacson M. O. Barth[2], Jean-Philippe Condomines[2], François Defaÿ[1], Jean-Marc Moschetta[1] *
[1]Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), Toulouse 31400, France
[2]École Nationale de l'Aviation Civile (ENAC), BP 54005, Toulouse Cedex 4, 31055, France

Abstract

This paper outlines current research conducted on tilt-body micro air vehicles at ISAE, and how we exploit recent advances to provide a tail-sitting flying-wing entry for the IMAV'17 outdoor challenge capable of performing automatic vertical take-off, landing, and trajectory-tracking.

1 Introduction

1.1 A quick contextualization

Since their debut in the 50s, tail-sitting vehicles would only be flown by the most experienced pilots. Recent advances on microelectromechanical (MEMs) inertial sensors and embedded computing, on the other hand, support stability augmentation systems (SAS) in mitigating unstable dynamic modes and allowing for inexperienced (or even autonomous) flight. A large and growing body of literature [1, 2] has investigated underlying modeling, control and planning issues specific to this architecture. Modeling, for instance, has recently called the attention of aerodynamicists due to unfamiliar high incidence propeller operation [3]. Further striking features are its nonlinearity, underactuation and difficult-to-model post-stall aerodynamics, which pose a beautiful challenge for the practicing control and robotics communities [4].

From a practical perspective, far too little attention has been paid to a fundamental question: how should the pilot control such a multifaceted architecture? While it seems appropriate to fly it as a quadrotor when in hover and as a fixed-wing when in cruise, it is unclear how to command it during transition. While platforms that provide an automatic transition phase do not face such dilemma, our control architecture allows for steady-flight in each forward velocity from rest to full forward speed and such question is pertinent. We explore this issue in this paper, and provide an entry to the IMAV'17 competition by applying an adequate control law to our piloting philosophy. We additionally show that guidance law design is independent of vehicle dynamics in our proposed architecture.

1.2 IMAV outdoor challenge specs

The MAVion will take up the outdoor flight performance challenge. It comprises taking-off automatically, flying as many laps as possible around 2 poles during a single flight, and landing automatically (see Fig. 1).

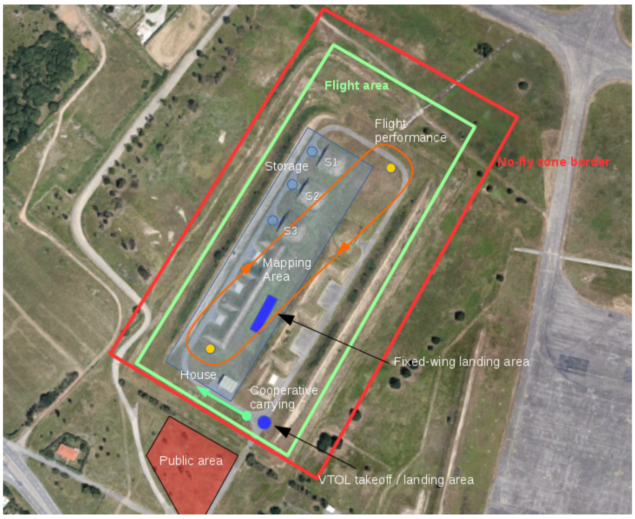

Figure 1: Competition lay-out at the Francazal airport runway.

2 The MAVion project in a nutshell

2.1 A brief history

ISAE started its tilt-body research with a tail-sitter called *Vertigo* that was developed and flown in 2006. Miniaturization of the Vertigo was conducted at University of Arizona to provide the *Mini-Vertigo*, a 30cm span coaxial-rotor MAV capable of transition flight. However, the coaxial rotor driving mechanism represented an additional weight and precluded further miniaturization. Moreover, the Mini-Vertigo suffered from fairly high induced drag in cruise flight due to its low aspect ratio. In view of improving aerodynamic performance in forward flight and simplifying the rotor mechanism, a new tilt-body configuration based on bimotor flying-wings was proposed. The MAVion was initially designed to be a reasonably efficient airplane capable of flying outdoors and requiring a minimum number of moving parts (e.g., no tilting wings, no tilting motors). The main design guidelines were simplicity and transition flight robustness. Currently, our prototypes fly by means of either nonlinear inversion or scheduled-LQR control laws. In parallel, a *Roll and Fly* [5]

*Corresponding e-mail address: l.lustosa@isae.fr version with wheels was designed to pursue indoors exploration.

2.2 Airframe

Fig. 2 illustrates the proposed airframe. In short, the MAVion is a flying-wing capable of tail-sitting. Moreover, it is capable of sustaining trimmed flight from rest up to 25m/s [6]. Equivalently, it sustains hover flight in face of external wind up to 25m/s (see Fig. 3). The MAVion has a 42cm wingspan and 2:1 aspect ratio. It weights (airframe, avionics and batteries) 435g and it is expected to endure a 10-30min flight – depending on mission hovering/cruise ratio requirements – based on wind tunnel measurements [7].

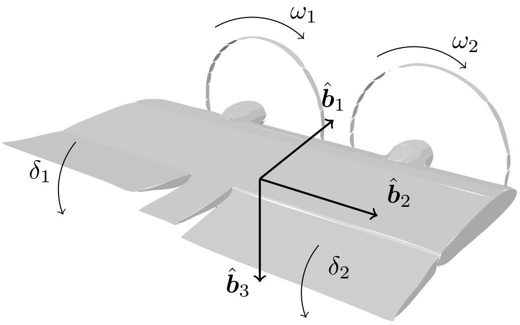

Figure 2: Perspective view, body-axis definition and actuation inputs for the MAVion tail-sitting vehicle. (Adapted from [7].)

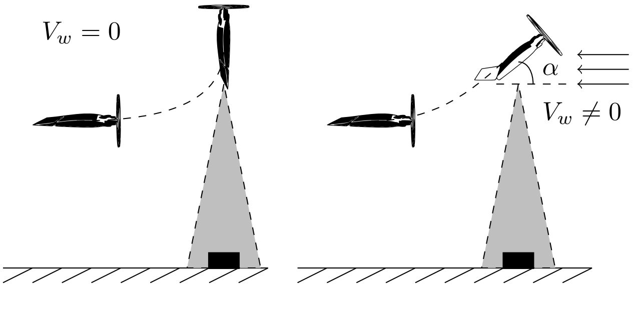

Figure 3: Transition maneuvers and hovering over ground target under non-windy (left) and windy (right) conditions. (Adapted from [8].)

2.3 Avionics

The competition version of the MAVion airframe is equipped with the *Paparazzi Apogee* autopilot board. It contains low-cost 3-axis accelerometers, rate-gyros and magnetometer for navigation purposes, and a MicroSD card slot for logging flight data. The autopilot communicates with an external GNSS receiver, a pitot tube and two radio links (data-link and fail-safe RC control). Digital servos govern elevons while DC motors control propellers motion.

Although not yet natively compatible with convertible architectures at the time of writing this paper, the Paparazzi framework [9] is modular, provides various handy mathematical C libraries, and encourages third-party modifications, such as the guidance, navigation and control techniques proposed herein.

2.4 Mathematical model

We assume MAVion dynamics as described by the state space nonlinear equations set forth in [7] and represented hereafter by $\gamma(\cdot)$ as follows

$$\dot{\boldsymbol{x}} = \gamma(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^{10}$, $\boldsymbol{u} \in \mathbb{R}^4$, $\boldsymbol{w} \in \mathbb{R}^3$ are, respectively, MAVion state, control inputs and wind disturbances, given by

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{v}_l & \boldsymbol{\omega}_b & q \end{pmatrix}^T \tag{2}$$

and

$$\boldsymbol{u} = \begin{pmatrix} \omega_1 & \omega_2 & \delta_1 & \delta_2 \end{pmatrix}^T \tag{3}$$

where $\boldsymbol{v}_l, \boldsymbol{\omega}_b \in \mathbb{R}^3$, $q \in \mathbb{R}^4$, denote, respectively, linear velocity described in local NED frame, angular velocity in body frame, and vehicle attitude in quaternion formulation. The control input $\boldsymbol{u}$ components and its associated sign conventions are depicted in Fig. 2.

The structure of $\gamma(\cdot)$ is fairly complex and, consequently, we point the interested reader to [7] for detailed equations and assumptions. Nevertheless, we remark that $\gamma(\cdot)$ provides an analytic continuous singularity-free model over a full 360^o angle-of-attack and sideslip flight envelope. Additionally, the model incorporates fundamental nonlinear aerodynamics – e.g., post-stall and prop-wash effects – in view of hybrid and high maneuverable vehicles. Incidentally, the tilt-body nature of the vehicle calls for a global numerically stable formulation of attitude and upholds quaternion employment. Another important practical feature of $\gamma(\cdot)$ is its polynomial-like algebraic structure that allows for efficient online trajectory generation by means of semidefinite programming and sum-of-squares (SOS) optimization. The IMAV outdoor challenge, however, proposes a known *a priori* trajectory and, therefore, we benefit from a simpler guidance solution detailed in Section 5.

In summary, for this challenge, whereas $\gamma(\cdot)$ is not used for trajectory generation, it supports simulation and control design (see Section 4). Our MATLAB-based simulator is composed of an outdoor mode which simply integrates (1) numerically, and an indoor mode that relies on the Unreal Engine (UE) physics collision engine. Fig. 5 illustrates the overall architecture: MATLAB computes resultant forces and moments based on γ and sends it to UE for kinematics computation in view of a UE-based map (with obstacles). If existent, collisions are handled and an updated state is then sent back to MATLAB for computing new forces. Additionally, UE provides a beautiful 3D viewer (see Fig. 4) and straightforward map/obstacles editing tools.

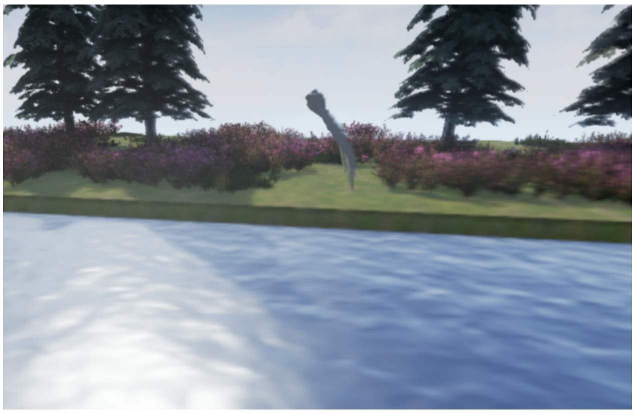

Figure 4: MAVion Unreal Engine-based simulator.

3 Navigation system

In light of the challenge requirements and available on-board avionics, we opted for deploying the subset of navigation sensors described in Table 1. We employ dead-reckoning inertial navigation assuming stationary and flat Earth on account of low-grade inertial sensors. A magnetometer/accelerometer-based complementary filter (CF) bounds the resulting divergent attitude errors. On top of the aforementioned system, a loosely-coupled extended Kalman filter (EKF) corrects for position and velocity errors by means of GNSS receiver measurements. Such architecture (see Fig. 6) benefits from a previous [10] compelling filter formulation that accounts for complementary filter dynamics in the EKF state transition matrix. This section briefly details the overall architecture for the sake of completeness. For a detailed study, we invite the reader to read [10].

Function	Device	Noise (typ)	Bias (typ)
Rate-gyro	MPU-9150	0.005 (°/s/$\sqrt{\text{Hz}}$)	20 (°/s)
Accelerom.	MPU-9150	400 (μg/$\sqrt{\text{Hz}}$)	150 (mg)
Magnetom.	MPU-9150	N/A	N/A
GNSS pos.	NEO-6M	$\sigma = 2.5$ (m)	0 (m)
GNSS vel.	NEO-6M	$\sigma = 0.1$ (m/s)	0 (m/s)

Table 1: Avionics subset for loosely-coupled GNSS and magnetometer aided strapdown inertial navigation. Inertial sensors specifications are readily determined from manufacturer datasheet. On the other hand, our GNSS receiver manufacturer datasheet lacks noise and bias statistics (understandably, since GNSS receiver loosely-coupled position and velocity errors are hardly Gaussian). Therefore, the available information was loosely recast to somewhat equivalent Gaussian noise and bias quantities (in view of suboptimal Kalman filtering implementation).

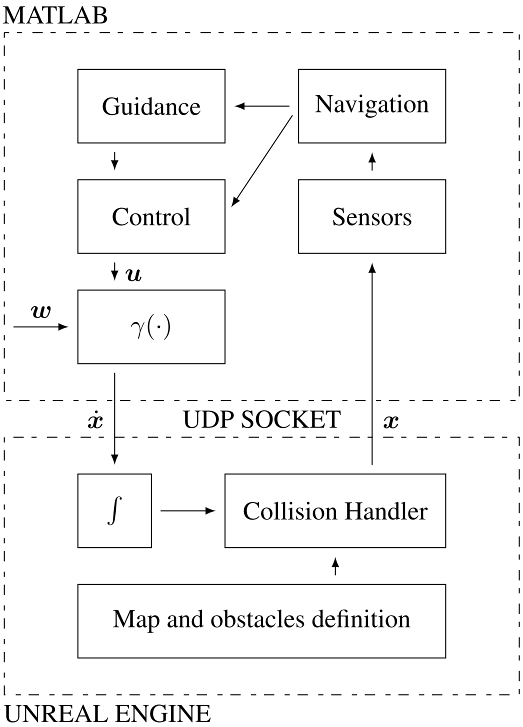

Figure 5: MAVion simulator architecture.

We model and simulate rate-gyro measurements $\boldsymbol{\omega} \in \mathbb{R}^3$ according to

$$\boldsymbol{\omega}(t) = \boldsymbol{\omega}_b^b(t) + \boldsymbol{\varepsilon}_b + \boldsymbol{\nu}_b^g(t) \tag{4}$$

where $\boldsymbol{\omega}_b^b$, $\boldsymbol{\varepsilon}_b \in \mathbb{R}^3$, and $\boldsymbol{\nu}_b^g \sim N(\mathbf{0}, \Sigma_g)$ denote, respectively, nominal angular velocity, rate-gyro drift and rate-gyro white Gaussian noise (all described in body frame). Likewise, accelerometer and magnetometer measurements, namely $\boldsymbol{f}(t)$ and $\boldsymbol{b}(t)$ are modeled as

$$\boldsymbol{f}(t) = \boldsymbol{a}_b^b(t) - \boldsymbol{g}_b(t) + \boldsymbol{\nabla}_b + \boldsymbol{\nu}_b^a(t) \tag{5}$$

and

$$\boldsymbol{b}(t) = \boldsymbol{B}_b(t) + \boldsymbol{\Delta}_b + \boldsymbol{\nu}_b^m(t) \tag{6}$$

where $\boldsymbol{a}_b^b$, $\boldsymbol{g}_b$, $\boldsymbol{\nabla}_b \in \mathbb{R}^3$, and $\boldsymbol{\nu}_b^a \sim N(\mathbf{0}, \Sigma_a)$ denote, respectively, linear acceleration, local gravity field, accelerometer bias and noise. Additionally, $\boldsymbol{B}_b$, $\boldsymbol{\Delta}_b \in \mathbb{R}^3$, and $\boldsymbol{\nu}_b^m \sim N(\mathbf{0}, \Sigma_m)$, denote local magnetic field, magnetometer bias and noise.

Rate-gyro measurements are filtered to bound attitude errors by means of the complementary filter, in such a way that the computed dead-reckoning angular velocity is given by

$$\boldsymbol{\omega}_c^c = \boldsymbol{\omega} - k_a \boldsymbol{f} \times D_c^l \boldsymbol{g}_l + k_m \boldsymbol{b} \times D_c^l \boldsymbol{B}_l \tag{7}$$

where k_a, $k_m \in \mathbb{R}_+$, and $D_c^l \in SO(3)$, are the complementary filter gains and NED-to-body CF-computed direction cosine matrix. Subsequently, $\boldsymbol{\omega}_c^c$ and $D_l^c \boldsymbol{f} + \boldsymbol{g}_l$ are integrated (in the rigid-body motion sense) to yield CF-computed estimates of quaternion attitude, NED-position and velocity, namely, q_c^l, $\boldsymbol{p}_l^c$ and $\boldsymbol{v}_l^c$ (see Fig. 6).

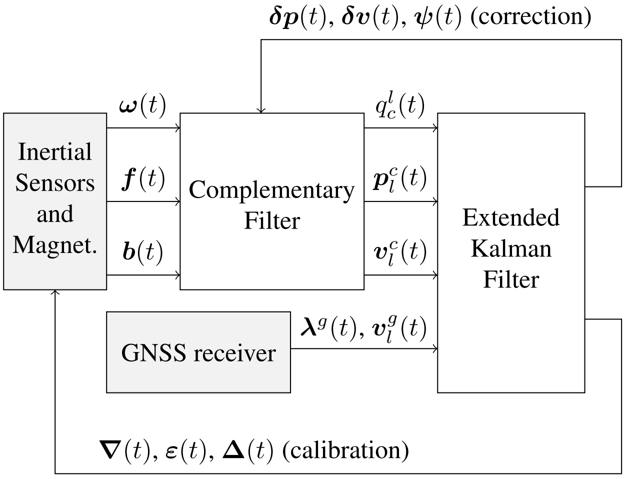
$\boldsymbol{\delta p}(t)$, $\boldsymbol{\delta v}(t)$, $\boldsymbol{\psi}(t)$ (correction)
Inertial Sensors and Magnet.
$\boldsymbol{\omega}(t)$
$\boldsymbol{f}(t)$
$\boldsymbol{b}(t)$
Complementary Filter
$q_c^l(t)$
$\boldsymbol{p}_l^c(t)$
$\boldsymbol{v}_l^c(t)$
Extended Kalman Filter
GNSS receiver
$\boldsymbol{\lambda}^g(t)$, $\boldsymbol{v}_l^g(t)$
$\boldsymbol{\nabla}(t)$, $\boldsymbol{\varepsilon}(t)$, $\boldsymbol{\Delta}(t)$ (calibration)

Figure 6: CF-EKF filter architecture. (Adapted from [10].)

Notice that CF outputs estimates with errors due to sensor imperfections. Previous work [10] modeled how CF errors evolve with time in function of sensor imperfections. That is, given unknown CF errors and sensor biases, namely,

$$\boldsymbol{x}_{EKF} = \begin{pmatrix} \boldsymbol{\delta p}_l & \boldsymbol{\delta v}_l & \boldsymbol{\psi} & \boldsymbol{\nabla}_b & \boldsymbol{\varepsilon}_b & \boldsymbol{\Delta}_b \end{pmatrix}^T \tag{8}$$

and process noise as

$$\boldsymbol{w}_{EKF} = \begin{pmatrix} \boldsymbol{\nu}_b^a & \boldsymbol{\nu}_b^g & \boldsymbol{\nu}_b^m & \boldsymbol{a}_b^b \end{pmatrix}^T \tag{9}$$

[10] computes $g(\cdot)$ such that

$$\dot{\boldsymbol{x}}_{EKF} = g(\boldsymbol{x}_{EKF}, \boldsymbol{w}_{EKF}) \tag{10}$$

The jacobians of $g(\cdot)$ yield the state transition matrix F_k and process noise transport matrix B_k that are employed in an EKF for CF error estimation. Navigation errors $\boldsymbol{\delta p}_l$, $\boldsymbol{\delta v}_l$, $\boldsymbol{\psi}$ are periodically used for correcting the CF while $\boldsymbol{\nabla}_b$, and $\boldsymbol{\varepsilon}_b$ and $\boldsymbol{\Delta}_b$ are periodically used for sensors in-flight calibration (see Fig. 6).

4 Control system

Our approach is to regulate the MAVion to *a priori* computed equilibrium points and limit cycles of $\gamma(\cdot)$ by means of stabilization of unstable transverse dynamics [11] employing scheduled-LQR control. Although often overlooked, we remark that any quaternion-based model linearization yields uncontrollable Jacobians [12] that preclude LQR direct employment. We overcome this shortcoming by employing the strategies set forth in [12] (a commonplace strategy, for instance, is to neglect one out of the four quaternion components in LQR design). Furthermore, quaternion double cover of $SO(3)$ does not pose a problem since we are applying local controllers and reference-to-estimated quaternions unwinding is effortlessly detected by dot product sign inspection.

4.1 Longitudinal equilibrium points

For longitudinal flight, previous work [6] experimentally shows that trimmed flight condition establishes an one-to-one correspondence between pitch angles θ and free-stream velocities $\boldsymbol{v}$. This motivated us to pursue a velocity controlled architecture, where the pilot (or high level guidance loop) does not command angles or angle rates but, instead, desired velocities in an almost-body frame (i.e., forward horizontal, lateral horizontal and vertical velocities). Therefore, a given desired velocity $\boldsymbol{v}_0$ is sufficient to unambiguously define the remaining of the desired equilibrium point $\boldsymbol{x}_0$ (e.g., attitude) in longitudinal flight by means of the (unique) solution of

$$\gamma\left((\boldsymbol{v}_0, \boldsymbol{0}, \boldsymbol{q}), \boldsymbol{u}, \boldsymbol{0}\right) = \boldsymbol{0} \tag{11}$$

The Jacobian of γ evaluated at $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ yields an uncontrollable linear system (A, B). After it is rendered controllable by means of the techniques in [12], we design a LQR controller $\Delta\boldsymbol{u} = -K\Delta\boldsymbol{x}$ to minimize

$$J(Q, R) = \int_0^\infty \left(\Delta\boldsymbol{x}^T Q \Delta\boldsymbol{x} + \Delta\boldsymbol{u}^T R \Delta\boldsymbol{u}\right) dt \tag{12}$$

given an appropriate choice of Q and R. These are tuned by trial-and-error runs in our simulator to account for actuator bandwidth, state estimation imperfections and embedded computer sampling times (all comprised in simulation). Once appropriate Q and R are found for hover operation, they are replicated for all other equilibrium points.

4.2 Lateral limit cycles

In contrast to longitudinal flight, lateral dynamics lacks equilibrium points in view of our choice for $\boldsymbol{x}$. One way to see this is observing that any constant $\boldsymbol{\omega}_b \neq \boldsymbol{0}$ changes q periodically. Therefore, instead of searching for equilibrium points, we search for states $\boldsymbol{x}$ such that the lateral force constitutes a coordinated curve centripetal force while maintaining constant altitude, that is,

$$\gamma\left((\boldsymbol{v}_b, \boldsymbol{\omega}_b, q), \boldsymbol{u}, \boldsymbol{0}\right) = \begin{pmatrix} \boldsymbol{\omega}_b \times \boldsymbol{v}_b \\ \boldsymbol{0} \\ \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}_b^T \\ \boldsymbol{\omega}_b & -[\boldsymbol{\omega}_b \times] \end{bmatrix} q \end{pmatrix} \tag{13}$$

In this way, our search for limit cycles reduces to a nonlinear root solving problem. Once limit cycles are found for a myriad of $(\boldsymbol{v}, \boldsymbol{\omega})$, LQR control can be designed by means of previous (Q, R) and transverse dynamics technology in [11].

Notice that we assume a bijection between equilibrium $(\boldsymbol{v}, \boldsymbol{\omega})$ and the other remaining variables[1]. This is a result which has not yet been established, but it is intuitively reasonable. Further study is required to establish its validity.

[1] Bijection in the sense of classes of equivalence: that is, the set of states $\boldsymbol{x}$ belonging to a same limit cycle composes a class of equivalence, and this class is in an one-to-one correspondence to $(\boldsymbol{v}, \boldsymbol{\omega})$ reference commands.

4.3 RC piloting interface

Notice that control references (equilibrium points or limit cycles) are given in terms of desired velocities $(\boldsymbol{v}, \boldsymbol{\omega})$. This means that a human pilot controls the MAVion thinking on where it needs to go instead of which orientation it needs to be. This abstracts the internal workings of the MAVion and relieves piloting efforts. Fig. 7 illustrates our proposition of RC controller input assignment for commonplace RC radio standards.

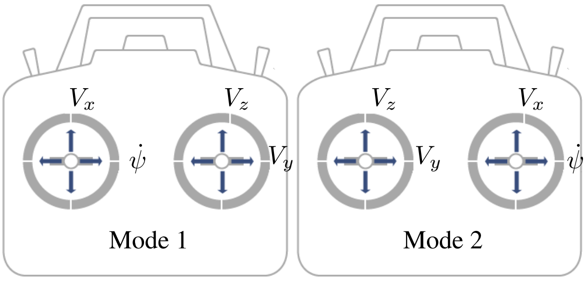

Figure 7: Input assignments of standard RC radio modes. Notice that ψ, V_x, V_z and V_y denote, respectively, desired heading with respect to geographic North, forward velocity, vertical velocity and lateral velocity.

5 Guidance system

We employ the commonplace guidance strategy described in [13], which is fully compatible with our aforementioned velocity-based controller architecture. This strategy is readily implemented in most open-source fixed-wing autopilots (e.g., `Pixhawk`, `Paparazzi`) and can be easily adapted for our purposes in view of our choice of control system reference parametrization – namely, forward and turn velocities (v, ω).

6 Simulation results

Flight simulations were conducted employing the aforementioned guidance, navigation and control strategies. The MAVion model is assumed known but we add wind disturbances up to 5m/s to account for robustness in control design. Additionally, navigation system routines are computed assuming corrupted sensor measurements according to Table 1. Figure 8 illustrates the obtained results. Those were obtained after a few controller tuning iterations. While it took considerable effort to find adequate LQR tuning weights, a promising result – in terms of performance and robustness – was found.

7 Conclusion

Previous research conducted on tilt-body vehicles at ISAE lays the groundwork for our IMAV competition entry. This paper provides general information on our architecture and shows promising results in simulation.

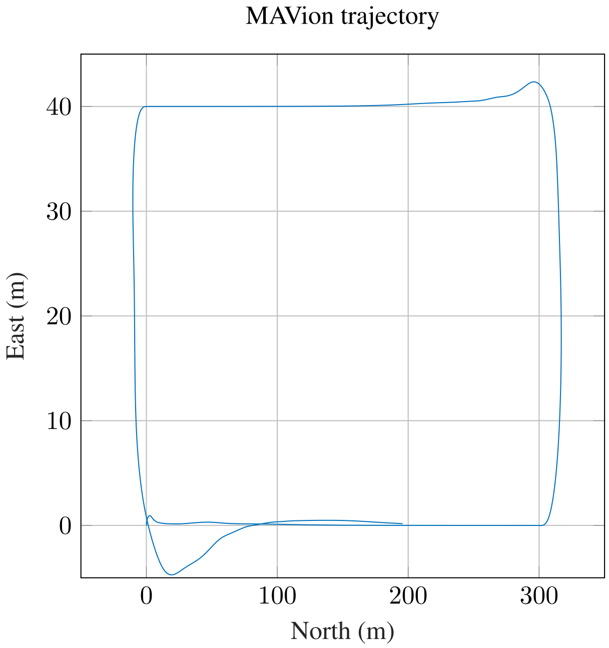

Figure 8: Simulation results: MAV trajectory.

Acknowledgements

The authors gratefully acknowledge the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, (Brazilian National Science Foundation), for partial financial support for this work through the "Ciência sem Fronteiras" program.

References

[1] S. Verling, B. Weibel, M. Boosfeld, K. Alexis, M. Burri, and R. Siegwart. Full attitude control of a VTOL tailsitter UAV. In *IEEE International Conference on Robotics and Automation*, 2016.

[2] T. Matsumoto, K. Kita, R. Suzuki, A. Oosedo, K. Go, Y. Hoshino, A. Konno, and M. Uchiyama. A hovering control strategy for a tail-sitter VTOL UAV that increases stability against large disturbance. In *IEEE International Conference on Robotics and Automation*, 2010.

[3] M. Bronz and A. Drouin. Preliminary design estimation of the V/STOL airplane performance. In *International Micro Air Vehicles Conference and Flight Competition*, 2015.

[4] Joseph Moore, Rick Cory, and Russ Tedrake. Robust post-stall perching with a simple fixed-wing glider using LQR-trees. *Bioinspiration and Biomimetics*, 9(2), 2014.

[5] P. A. Alheritiere, R. Olivanti, L. R. Lustosa, F. Defaÿ, and J.-M. Moschetta. Nonlinear control of a particular

tilt-body MAV: The Roll and Fly. In *The 24th Mediterranean Conference on Control and Automation*, 2016.

[6] M. Itasse, J.-M. Moschetta, Y. Ameho, and R. Carr. Equilibrium transition study for a hybrid MAV. *International Journal of Micro Air Vehicles*, 3(4):229–246, 2011.

[7] L. R. Lustosa, F. Defaÿ, and J.-M. Moschetta. Longitudinal study of a tilt-body vehicle: modeling, control and stability analysis. In *International Conference on Unmanned Aircraft Systems*, 2015.

[8] L. R. Lustosa, F. Defaÿ, and J.-M. Moschetta. Development of the flight model of a tilt-wing micro-air vehicle. In *International Micro Air Vehicle Conference and Competition*, 2014.

[9] G. Hattenberger, M. Bronz, and M. Gorraz. Using the paparazzi UAV system for scientific research. In *International Micro Air Vehicle Conference and Competition*, 2014.

[10] L. R. Lustosa, S. Pizziol, F. Defaÿ, and J.-M. Moschetta. An error model of a complementary filter for use in Bayesian estimation - The CF-EKF filter. In *20th IFAC Symposium on Automatic Control in Aerospace*, 2016.

[11] I. R. Manchester. Transverse dynamics and regions of stability for nonlinear hybrid limit cycles. In *18th IFAC World Congress*, 2011.

[12] L. R. Lustosa, F. Cardoso-Ribeiro, F. Defaÿ, and J.-M. Moschetta. A new look at the uncontrollable linearized quaternion dynamics with implications to LQR design. 2018 (to be published).

[13] Sanghyuk Park, John Deyst, and Jonathan How. A new nonlinear guidance logic for trajectory tracking. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control*, 2004.