A Robust Control Strategy with Perturbation Estimation for the Parrot Mambo Platform

Ignacio Rubio Scola, Gabriel Alexis Guijarro Reyes, Luis Rodolfo Garcia Carrillo, *Member, IEEE*, João P. Hespanha, *Fellow, IEEE*, and Laurent Burlion, *Member, IEEE*

***Abstract*—This paper address theoretical and practical challenges associated with a commercially–available and ready–to–fly small scale unmanned aircraft system developed by Parrot SA®: the Mambo™quad rotorcraft. The dynamic model and the structure of the controller running onboard the unmanned aircraft system autopilot are not disclosed by its manufacturers. For this reason, a novel robust controller for discrete–time systems under time–delays and input saturation is first developed for this platform. Then, three fundamental estimation and control challenges are addressed. The first challenge is the system identification of the X and Y translational dynamics of the unmanned aircraft system. To accomplish this goal, input-output data pairs are collected from different unmanned aircraft system platforms during real-time experimental flights. A group of dynamic models are identified from the data pairs through an extended least squares algorithm. The obtained models are similar in nature but exhibit parametrical variations due to uncertainties in the fabrication process and different levels of wear and tear. Using a time-varying modeling approach, the second challenge addressed consists on the development of a robust controller which guarantees the stability of all the identified dynamic models. The third challenge addresses the development of a nonlinear controller enhanced with a perturbation estimation which can reject, from the nominal model, the effects of model uncertainties and perturbations. These theoretical developments are presented in the form of two original theorems. The proposed strategies are ultimately validated in a set of real-time experiments, demonstrating their effectiveness and applicability.**

***Index Terms*—Unmanned aircraft systems, Robust control, Nonlinear control, Perturbation estimation.**

I. Introduction

UNSATISFACTORY performance and instability in feedback systems are commonly related to model uncertainties. For this reason, a control design considering robust stability is relevant and nontrivial. Towards this objective, H_∞ control strategies for robust stability of linear and nonlinear systems have been extensively studied over the past decades and many interesting results have been introduced, see for example [1]–[4] and the references therein. In order to apply robust control techniques a model of the system to be controlled is needed. However, when dealing with commercial systems, it is rare that the manufacturer makes available the equations describing the corresponding dynamic model. This is the case with most, if not all commercially–available ready–to–fly unmanned aircraft system (UAS) platforms. Commercial UASs are equipped with an onboard autopilot executing a low–level control strategy whose structure is also not disclosed. The main function of the autopilot is to stabilize the attitude of the platform during autonomous flights. In some instances, the autopilot also allows a human user to control the motion of the UAS, making use of a remote controller or via a pre–selected set of way-points that the UAS must follow. Due to the lack of knowledge in terms of dynamic models and control structures, commercial UAS platforms are rarely used for research purposes. However, these limitations represent in fact an ideal scenario for developing novel system identification techniques in combination with robots control strategies allowing the implementation of commercial systems in research–oriented frameworks [5].

Fig. 1. The Mambo™multicopter: a small UAS developed by Parrot SA ®.

Taking inspiration from the results in the area of H_∞ control as well as from the limitations encountered in numerous commercial UAS platforms, the research presented in this paper focuses on the development and implementation of an H_∞ robust control strategy for the X–Y translational dynamics of the Mambo™multicopter, a commercially–available ready–to–fly small scale UAS developed by Parrot SA®, see Figure 1. Not surprisingly, the equations describing the Mambo UAS dynamics are not disclosed by the manufacturer, neither is the structure of the low–level control strategy running onboard its autopilot. As a first step towards developing a high–level control strategy for stabilizing the Mambo during real–

I. Rubio Scola is with CIFASIS (CONICET-UNR), Department of Mathematics, FCEIA-UNR, Rosario, Argentina, and also with the Unmanned Systems Laboratory. Engineering and Computing Sciences, Texas A&M University - Corpus Christi, Texas, USA. e-mail: irubio@fceia.unr.edu.ar

G.A. Guijarro Reyes is with the Unmanned Systems Laboratory. School of Engineering and Computing Sciences, Texas A&M University - Corpus Christi, Texas, USA. e-mail: gguijarroreyes@islander.tamucc.edu

L.R. Garcia Carrillo is with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, USA. e-mail: luisillo@nmsu.edu

J. Hespanha is with Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara, USA. e-mail: hespanha@ece.ucsb.edu

L. Burlion is with Rutgers, The State University of New Jersey, USA. e-mail: laurent.burlion@rutgers.edu

Manuscript received xxxx

time autonomous missions, a model identification procedure is conducted, which consists of an extended least-squares (ELS) identification technique, applied to a set of data pairs containing input–output system information. The interested reader is directed to [6]–[8], where this kind of technique has been successfully applied to real–time systems.

The main challenge associated with system identification is that any identification procedure for a real-time system would be commonly affected by errors and uncertainties. As a matter of fact, by performing multiple instances of a system identification procedure for the exact same system, the designer will commonly obtain similar but slightly different results. On top of that, it is often the case that a system identification must be performed not just for one, but for two or more *identical* real–time platforms, e.g. in our case, for a group of four Mambo UAS. This situation is commonly faced by practitioners when it is time for them to replace (probably broken) platforms, and also by researchers performing real-time experiments related to Multi-Agent Systems (MAS). Indeed, a broad parametrical variation is encountered when performing multiple identification experiments involving multiple platforms similar in nature. This is the situation addressed in this paper, since multiple uncertain models are obtained from the set of Mambo UAS platforms available at the TAMU-CC Unmanned Systems Laboratory [9], and currently used for performing MAS research activities.

A. Related Work

A survey of current methods and applications of system identification techniques for small low-cost UASs is provided in [10]. In [11] a linear matrix inequality (LMI)-based procedure is proposed for synthesizing the gains of a closed–loop controller for stabilizing a quad rotorcraft UAS using an approximate feedback linearizing controller. The synthesis procedure generates suboptimal gains with respect to H_2 and H_∞ performance cost functions, and a pole placement to mixed region constraint. In [12] a nonlinear disturbance observer–based robust attitude tracking controller was proposed for quad rotorcraft UASs.

1) Robust control for uncertain dynamics and perturbations: In [13] an H_∞ control approach is adopted in order to cope with the presence of uncertainty such as unmodeled dynamics, unknown parameters, and inputs constraints. First, a continuous-time system identification is performed on the experimental data to encapsulate a nominal model of the system as well as a multiplicative uncertainty. By doing this, H_∞ control strategies for both roll and pitch angles are synthesized. Recently, in [14], [15] the authors proposed a robust output tracking controller design for a class of uncertain discrete-time systems subjected to actuator saturation. The work in [16] addressed the control problem of a quad rotorcraft in the presence of input constraints. In [17] the authors addressed the stability and tracking control problem of a quad rotorcraft UAS, where adaptation laws are designed to learn and compensate modeling errors and external uncertain disturbances.

2) Systems with time–delay: In [18], [19] a delay estimation method is introduced by using transient responses of a quad rotorcraft UAS and analytical solutions of delay differential equations. Collected altitude responses in the time domain are compared to the predicted ones, which were obtained from analytical solutions. The effects of the time delay on the responses are analyzed through the locations of the characteristic roots in the complex plane. Based on the estimation result, proportional plus velocity controllers are proposed to improve transient altitude responses. In [20] the authors presented an application of model reference adaptive control (MRAC) to quad rotorcraft UASs, considering the time–delay in the altitude control system. The MATLAB system identification toolbox is applied to obtain the altitude motion model, without time–delay, for the quad rotorcraft. Then proportional–plus–velocity (PV) and PV-MRAC altitude control systems are designed, by incorporating an estimated constant time delay. In [21] a robust trajectory tracking control problem is developed for quad rotorcraft UASs with multiple uncertainties and time–delays. The UAS model is described as a multiple–input multiple–output time–varying system subject to parametric perturbations, nonlinear and coupled dynamics, external disturbances, and state and input delays. In [22] a robust control strategy based on a predictor and an uncertainty and disturbance estimator is developed for a class of uncertain nonlinear systems with input/output delays.

3) Literature on commercial UAS platforms: The work in [23] exposed the navigation and control technology embedded in the commercial AR.Drone UAS platform developed by Parrot. Still, the system identification and translational position control problem for a commercial UAS platform with a builtin low–level controller that stabilizes and decouples the velocities in each translational direction has not yet been studied. In general, the stability and decoupling advantages of the builtin controller represent an advantage for certain users. However, when it is desired to use these kind of platforms in research oriented experimental scenarios, diverse undesired disadvantages become evident, such as dead–times, unknown dynamics, saturation in the control inputs, and fixed sampling times.

B. Main Contributions

Taking into account the practically–oriented challenge of enabling commercially–available UASs to perform autonomous tasks, the research conducted here addresses a set of fundamental estimation and control challenges related to any commercially–available UASs in general, and to the Parrot Mambo UAS quad rotorcraft in particular. The first contribution consists on the development of a novel robust controller for discrete-time systems under time delays and input saturation. Two original theorems are presented, in particular *Theorem 3* and *Theorem 4*, which together summarize the theoretical contribution of this research work. Next, and building upon these theoretical results, three fundamental estimation and control challenges are identified, addressed, and successfully solved. The first challenge is the system identification of the X and Y translational dynamics of the UAS. To accomplish this goal, input-output data pairs are

collected from different Mambo UAS platforms during real-time experimental flights. A group of dynamic models are identified from the data pairs through an extended least squares algorithm. The obtained models are similar in nature but exhibit parametrical variations due to uncertainties in the fabrication process and different levels of wear and tear. Using a time-varying modeling approach, the second challenge addressed consists on the development of a robust controller which guarantees the stability of all the identified dynamic models. The third challenge addresses the development of a nonlinear controller enhanced with a perturbation estimation which can reject, from the nominal model, the effects of model uncertainties and perturbations. The proposed strategies are validated in a set of real-time experiments, demonstrating their effectiveness and applicability.

The rest of the manuscript is organized as follows. First, the problem addressed is formally introduced in Section II. Next, Section III describes the mathematical background of the proposed identification technique, which is based on an ELS methodology for linear systems with dead–time delays. The theoretical background for the H_∞ robust control strategy and the additional nonlinear control law with disturbance estimation is introduced in Section IV. The main contributions are introduced next. First, Section V presents the model identification procedure for the Mambo UAS platform. Next, Section VI presents the H_∞ corresponding nonlinear robust controller synthesis, which makes use of the identification results. Real–time experiments are presented in Section VII, demonstrating the satisfactory performance of the identification and robust nonlinear control techniques. Concluding remarks and future research directions are presented in Section VIII.

II. Problem Statement

The Mambo quad rotorcraft is equipped with an onboard autopilot that decouples and controls its motion in three axis, i.e., *longitudinal*, *lateral*, and *vertical* translational directions. The autopilot also decouples and controls the platform *heading orientation*, i.e., the yaw angle. As with most commercially–available ready–to–fly UAS platforms, the architecture of the low–level estimation and control strategies running onboard the Mambo autopilot are unavailable to the final user.

As a first step towards addressing the estimation and control challenges at hand, it is assumed that the Mambo's autopilot architecture is similar to the architecture encountered in the AR.Drone autopilot, which has been previously studied in [23]. In particular, the assumption is that both platforms have similar (but not identical) estimation, control, and decoupling strategies running onboard. This conclusion was reached after performing similar identification tests with both the AR.Drone and the Mambo, and observing similar results and behaviors. Furthermore, from an inspection of the internal components, it can also be inferred that both platforms share a similar architecture in terms of hardware.

Despite the observed similarities, a dedicated identification of the Mambo's closed-loop system is needed in order to ensure appropriate stabilization during autonomous real–time missions. The model to be identified considers the *velocity set–point* as the *control input*, while the *translational position* in the corresponding axis of motion is considered to be the *output*. An additional challenge that is worth mentioning is due to the fact that the amplitude of the control input signal is limited (saturated) by the manufacturer via software. This saturation has the objective of avoiding pitch and roll attitude angles compromising the appropriate operation of the UAS.

Fig. 2. The TAMU-CC Unmanned Systems Laboratory experimental test-bed.

A. The UAS experimental test–bed

The UAS test–bed available at Texas A&M University – Corpus Christi (TAMU-CC) Unmanned Systems Laboratory (USL) is shown in Figure 2. The following discussion refers to this system in particular. Still, these ideas apply to any test–bed regardless of the UAS platforms, motion capture system (MCS), ground station computer (GSC), and communication links selected for performing the real–time experiments.

In the TAMU-CC USL, the 3-dimensional position and heading dynamics of the Mambo are measured using a 12–cameras MCS developed by Vicon. Then, a GSC receives the system states, and uses this information to compute the corresponding control signals. A wireless link is then used for sending the control commands to the UAS. Figure 2 illustrates the interaction between the MCS, GSC, and the Mambo. Notice that any communication process involved in the estimation and control tasks will introduce a time–delay. Ultimately, the control strategy should stabilize the corresponding states of the UAS using the velocity set-points, while taking into account input saturation, communications delays, as well as model uncertainties coming from manufacturing heterogeneity between similar platforms.

III. System Identification

This section introduces an ELS algorithm designed to identify a linear system affected by output additive noises. Real–time systems are commonly affected by time delays in the signals used for state estimation and control purposes. To address this additional challenge an extension of the identification methodology for linear systems with time–delays is provided. Relevant practical considerations for implementing

the identification approach are discussed also, which simplifies the identification task under certain conditions.

A. The proposed ELS algorithm for system identification

The ELS algorithm introduced in [6], [7] allows identifying Single Input – Single Output (SISO) linear time invariant (LTI) discrete–time systems. Towards this goal, consider an n-th order difference equation model of the form

$$A_q z_k = B_q u_k, \tag{1}$$

where the subscript k denotes the time sample, u_k and z_k are scalar input and noiseless output, respectively, q is the unity advance operator, and A_q and B_q are the linear n-th order polynomials in q such that

$$q^{-1} z_k = z_{k-1} \tag{2}$$

$$A_q = 1 + a_1 q^{-1} + \cdots + a_n q^{-n} \tag{3}$$

$$B_q = b_1 q^{-1} + \cdots + b_n q^{-n} \tag{4}$$

Assumption 1: The system in equation (1) is stable, and the coefficients a_i and b_i, for $i = 1, 2, \ldots, n$, are constant.
For the research challenges addressed in this work, the validity of *Assumption 1* comes from two reasons: (i) the controller on board the Mambo's autopilot, which was developed by the manufacturer of this system, is able to stabilize the rotorcraft's translational velocity and, (ii) this control strategy decouples the motion of the platform in four independent axis, i.e., longitudinal, lateral, vertical, and heading.

Assumption 2: The system output $y_k$is corrupted by an additive Gaussian zero-mean white noise ν_k as

$$y_k = z_k + \nu_k. \tag{5}$$

From equations (1) and (5) one obtains

$$A_q y_k = B_q u_k + \epsilon_k \tag{6}$$

with $\epsilon_k = A_q \nu_k$. The following matrices are now introduced

$$\theta = [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n]^\top, \tag{7}$$

$$y = [y_{n+1}, y_{n+2}, \cdots, y_{N_{ls}}]^\top, \tag{8}$$

$$\epsilon = [\epsilon_{n+1}, \epsilon_{n+2}, \cdots, \epsilon_{N_{ls}}]^\top, \tag{9}$$

$$\phi_i = [-y_{i-1}, -y_{i-2}, \cdots, -y_{i-n}, u_{i-1}, u_{i-2}, \cdots, u_{i-n}]^\top, \tag{10}$$

$$\Phi = [\phi_{n+1}, \phi_{n+2}, \cdots, \phi_{N_{ls}}]^\top, \tag{11}$$

where the superscript $\top$ stands for matrix transpose, and N_{ls} is the number of measurements available. Now, it is possible to write the equation

$$y = \Phi\theta + \epsilon \tag{12}$$

The optimal least square (OLS) solution $\hat{\theta}_{OLS}$ that minimizes the norm $\epsilon^\top \epsilon$ is then given by

$$\hat{\theta}_{\text{OLS}} = (\Phi^\top \Phi)^{-1} \Phi^\top y. \tag{13}$$

In order to eliminate the noise–induced bias in $\hat{\theta}_{OLS}$, ϵ_k is modeled as $C_q \epsilon_k = e_k$, with $C_q = 1 + c_1 q^{-1} + \cdots + c_n q^{-n}$, and e_k as an independently distributed random sequence. Furthermore, the order of the constants terms c_i is m.

Introducing now the matrices

$$\Pi = [c_1, c_2, \cdots, c_m]^\top, \tag{14}$$

$$e = [e_{n+1}, e_{n+2}, \cdots, e_{N_{ls}}]^\top, \tag{15}$$

$$w_i = [-\epsilon_{i-1}, -\epsilon_{i-2}, \cdots, -\epsilon_{i-m}]^\top, \tag{16}$$

$$\Omega = [w_{n+1}, w_{n+2}, \cdots, w_{N_{ls}}]^\top, \tag{17}$$

the term ϵ_k can be estimated by means of

$$\hat{\epsilon}_k = \hat{A}_q y_k - \hat{B}_q u_k, \tag{18}$$

with $\hat{A}$ and $\hat{B}$ being the estimation of A and B, respectively. Putting together the estimation problems for the matrices θ and Π, the following nonlinear problem is obtained

$$y = [\Phi \ \Omega] \begin{bmatrix} \theta \\ \Pi \end{bmatrix} + e \tag{19}$$

whose solution is obtained by solving the following equations

$$\hat{\theta} = \hat{\theta}_{OLS} - \hat{\theta}_{\text{Bias}}, \tag{20}$$

$$\hat{\theta}_{\text{Bias}} = (\Phi^T \Phi)^{-1} \Phi^T \Omega \hat{\Pi}, \tag{21}$$

$$\hat{\Pi} = [\Omega^T \Omega]^{-1} \Omega^T \epsilon, \tag{22}$$

The nonlinear problem at hand must be solved iteratively. In particular, the term $\hat{\theta}_{\text{OLS}}$ is first obtained at each iteration using equation (13). Next, $\hat{\epsilon}$ is computed using equation (18). Finally, $\hat{\theta}$ is calculated using equations (20)-(22). This procedure is repeated until the value $\|\hat{\theta}^i - \hat{\theta}^{i+1}\|$ is sufficiently small, where $\hat{\theta}^i$ is computed at the iteration i, and $\hat{\theta}^{i+1}$ is computed in the consecutive iteration.

The identification procedure is now extended to address time–delays induced by hardware and software, which are no negligible since they directly affect the system dynamics.

B. Extension for dead–time identification

A system with a dead–time (i.e., a delay) of τ samples can be represented by modifying equation (6) as

$$A_q y_k = z^{-\tau} B_q u_k + \epsilon_k \tag{23}$$

In order to develop useful modeling and control results, the value of τ must be identified first. A solution for the dead–time estimation problem was originally proposed in [24], where the ELS algorithm is combined with an optimization procedure. In particular, given the expected minimum and maximum delay samples, i.e., τ_{min} and τ_{max}, the ELS algorithm is applied to a set of $(\tau_{\text{max}} - \tau_{\text{min}} + 1)$ models, all of which have polynomials with the same order in A_q and B_q, but different dead–times given by

$$\tau_i = \tau_{\text{min}} + i, \quad i = 0, 1, 2, ..., (\tau_{\text{max}} - \tau_{\text{min}}). \tag{24}$$

A performance index I_{i} is computed for each model by comparing its output with respect to (w.r.t.) the ground truth obtained from the real process. Specifically

$$I_{\text{i}} = \|y - \hat{y}_{(i)}\|^2, \tag{25}$$

where $\hat{y}_{(i)}$ is the output of the i-th model. The best model is chosen as the one which gives the lowest I_{i}. Then, the dead–time of the best model is considered as the best estimation of the dead–time afecting the system.

Once the time–delay has been properly identified, the corresponding dead-time problem is addressed by making use of the methodology presented in [25]. Towards this end, the technique to stabilize a SISO LTI system with time–delays using an equivalent SISO LTI system which is not affected by time delays is introduced next.

C. A reduction for discrete-time LTI systems with delay

The authors in [25] proposed a method for reducing a system with delayed dynamics into a new system which is free of delays and equivalent – in terms of stabilization – to the original one. Following this approach, consider first the reduction technique for a SISO discrete-time LTI system of the form

$$\eta_{k+1} = A_d\eta_k + B_d u_{k-\tau}. \tag{26}$$

with η_k the state of the system, and τ the dead–time at the control input u.

Remark 1: Equation (23) can be written in the form of equation (26) (which is not unique) using one of the methods for going from a transfer function to a state space representation, as described in [26].

A change of variable is now introduced, which has the form

$$x_k = \eta_k + \sum_{j=k-\tau}^{k-1} A_d^{k-1-\tau-j} B_d u_j \tag{27}$$

and evolves with dynamics

$$x_{k+1} = A_d x_k + A_d^{-\tau} B_d u_k \tag{28}$$

Lemma 1: (From [25]) The dynamics in equation (28) are controllable if and only if the following conditions are satisfied

- the pair (A_d, B_d) is controllable;
- A_d has no zero eigenvalue.

Also, any feedback $u = Kx$ ensuring that $A_d + A_d^{-\tau} B_d K$ is Schur, asymptotically stabilizes the system in equation (26).

Remark 2: The real–time system identification procedure of the Mambo UAS involves the execution of a set of (very similar) experiments. From these results, a collection of linear models with the same structure but with different parameters are obtained.

D. A discussion on important practical considerations to ease the system identification

The following practical considerations, which are important to achieve a suitable identification of the real system, were also taken into account for the real–time implementation of our theoretical results.

- **Signal scaling:**for the computation of the ELS in equations (20)-(22) to be numerically well conditioned, it is important that the numerical values of both the inputs and the outputs have roughly the same order of magnitude. Because the units of these variable are generally quite different, this often requires scaling of these variables. It is good practice to scale both inputs and outputs so that all variable take "normal" values in the same range, e.g., the interval $[-1, 1]$.
- **Down–Sampling:** if the data acquisition system can sample the system at higher frequencies than the identification needs, a down–sample can be applied to the signals. This helps removing measurement noises from the signals.
- **Dealing with known parameters:** due to physical considerations, one often knows one or more poles/zeros of the process. Imposing the predefined structure with its known parameters simplifies the problem.
- **Quality of Fit:** the quality of fit can be checked by computing the Mean-Square Error (MSE) achieved by the estimate, normalized by the Mean-Square Output (MSO). In particular: $\frac{MSE}{MSO} = \frac{\|\phi\hat{\theta}-y\|^2}{\|y\|^2}$.

The interested reader is referred to [27] for additional details concerning the implementation of identification methods.

IV. Robust Control

As a preliminary step towards the development of a robust control strategy, it is required to mathematically represent all the obtained LTI reduced models by means of a single unified linear time varying (LTV) model.

A. Parameter dependent LTV discrete-time systems

Consider a parameter dependent LTV discrete-time system of the form:

$$\begin{aligned} x_{k+1} =& A(\xi_k)x_k + B(\xi_k)u_k + E(\xi_k)d_k \\ y_k =& C(\xi_k)x_k + D(\xi_k)u_k + F(\xi_k)d_k \end{aligned} \tag{29}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector and $d \in \mathbb{R}^p$ a perturbation. Also, the matrices $A(\xi_k)$, $B(\xi_k)$, $C(\xi_k)$, $D(\xi_k)$, $E(\xi_k)$, and $F(\xi_k)$ are assumed to depend affinely on the time-varying parameter ξ_k, with values assumed in the unit simplex

$$\Xi = \left\{ \xi \in \mathbb{R}_+^N : \sum_{i=1}^{N} \xi_i = 1 \right\} \tag{30}$$

The affine assumption means that the matrices $A(\xi_k)$, $B(\xi_k)$, $C(\xi_k)$, $D(\xi_k)$, $E(\xi_k)$, and $F(\xi_k)$ can be written as

$$\begin{bmatrix} A(\xi_k) & B(\xi_k) & E(\xi_k) \\ C(\xi_k) & D(\xi_k) & F(\xi_k) \end{bmatrix} = \sum_{i=1}^{N} \xi_{i,k} \begin{bmatrix} A_i & B_i & E_i \\ C_i & D_i & F_i \end{bmatrix} \tag{31}$$

where each subindex $i =\in \{1, 2, \ldots, N\}$ is associated with each one of the N vertice generated by combinations of all the extreme values of each matrix element from the LTI models previously obtained from the identification procedure.

B. The proposed H_∞ controller

For the purpose of defining the H_∞ cost criterion, consider the asymptotically stable open-loop dynamics of the form

$$\begin{aligned} x_{k+1} =& A(\xi_k)x_k + E(\xi_k)d_k \\ y_k =& C(\xi_k)x_k + F(\xi_k)d_k \end{aligned} \tag{32}$$

for which the H_∞ performance μ is defined by means of the L_2-to-L_2 gain

$$\|H\|_\infty = \mu = \sup_{\|d_k\|_2 \neq 0} \frac{\|y_k\|_2}{\|d_k\|_2} \tag{33}$$

A linear condition for obtaining a static controller ensuring a predefined H_∞ performance μ is discussed in *Lemma 2*.

Lemma 2: (From [3]) The system in equation (29) is poly-quadratically stabilizable with H_∞ performance bound μ, if and only if there exist matrices $Q_i = Q_i^\top \succ 0$ in $\mathbb{R}^{n,n}$, and $X \in \mathbb{R}^{n,n}$, $L \in \mathbb{R}^{m,n}$, $Y_i \in \mathbb{R}^{m,n}$, $W_i \in \mathbb{R}^{m,p}$, $Z_i \in \mathbb{R}^{m,m}$, such that

$$\begin{bmatrix} X + X^T - Q_i & \star & \star & \star \\ A_i X & Q_j - B_i Y_j - Y_j^T B_i^T & \star & \star \\ -L & Z_j^T B_i^T - Y_j & \star & \star \\ C_i X & -D_i Y_j - W_j^T B_i^T & Z_j + Z_j^T & \star \\ 0 & E_i^T & D_i Z_j - W_j^T & \mu I - D_i W_j - W_j^T D_i^T \\ & & 0 & F_i^T \end{bmatrix} \begin{matrix} \star \\ \star \\ \star \\ \star \\ \star \\ \mu I \end{matrix} \succ 0 \tag{34}$$

for all vertices represented by $i, j = 1, ..., N$. In equation (34) the symbol $(\star)$ denotes the symmetric transpose, and $\succ$ expresses the positive definite condition. The robust control law obtained is therefore given by

$$u_k = Kx_k, \quad \text{with} \quad K = LX^{-1}. \tag{35}$$

Remark 3: The problem defined in equation (34) is composed by an inequality, therefore, there exists an infinity number of solutions. Ultimately, it is possible to minimize μ in order to obtain the solution to a linear optimization problem that finds a static linear controller with minimal H_∞ performance bound.

C. Nominal Dynamics and Disturbance Identification

This section builds upon the procedure developed in [14] to minimize the effects of model uncertainties and external perturbation over a nominal model. To begin, consider a nominal model of the form

$$\begin{aligned} x_{k+1} &= A_N x_k + B_N \text{sat}(u_k) + d_k \\ y_k &= C_N x_k \end{aligned} \tag{36}$$

with time invariant matrices A_N, B_N, and C_N and d_k a perturbation term that is composed by external perturbations and mismatch due to model uncertainties. The term $\text{sat}(u_k)$ is the saturation function defined as

$$\text{sat}(u_k) = \begin{cases} u_{\max}, & \text{for } u_k > u_{\max} \\ u_k, & \text{for } |u_k| \leq u_{\max} \\ -u_{\max}, & \text{for } u_k < -u_{\max} \end{cases} \tag{37}$$

where $u_{\max} > 0$ is the maximum amplitude allowed of u_k due to the actuator saturation. A couple of relevant assumptions are now introduced.

Assumption 3: The pair (A_N, B_N) is controllable, and the realization (A_N, B_N, C_N) is invertible and has no zeros when the complex variable, of the Z-transform, $\mathbf{z} = 1$.

Assumption 4: The upper bound of d_k is $d_{\max}$, and it is assumed that $|(B_N^T B_N)^{-1} B_N^T d_k| < d_{\max}$, $\forall k \in \mathbb{Z}^+$. The term $d_{\max}$ is a known positive constant and $d_{\max} < u_{\max}$.

Due to the nature of the problem at hand, the required control signal must be composed by the sum of three parts: (i) a linear feedback law, (ii) a nonlinear feedback law, and (iii) a perturbation estimation for a feed–forward law. The design of each component is introduced next.

D. Linear Feedback Law

The linear part of the controller is computed as

$$u_{L,k} = Kx_k + Gr \tag{38}$$

where r is the amplitude of the reference, and $K \in \mathbb{R}^{1\times n}$ stabilizes the system. Also, with an identity matrix I_d having appropriate dimensions, $G = 1/(C_N(I_d - A_N - B_N K)^{-1} B_N)$ is well defined because all eigenvalues of $(A_N + B_N K)$ are inside the unit circle and (A_N, B_N, C_N) is invertible and has no zeros at $\mathbf{z} = 1$. Furthermore, there exists an equilibrium state x_e associated with r, which allows defining the error vector $\tilde{x}_k = x_k - x_e$, then the controller $u_{L,k}$ is

$$\begin{aligned} u_{L,k} &= K\tilde{x}_k + (1 + K(I_d - A_N - B_N K)^{-1})Gr \\ &= K\tilde{x}_k + Hr \end{aligned} \tag{39}$$

The following assumptions are now introduced

Assumption 5: For any $\delta_1 \in (0, 1)$, then c_δ be the largest positive scalar such that

$$|K\tilde{x}_k| \leq \delta_1 u_{\max}, \qquad \forall \tilde{x}_k \in \mathbb{X}_\delta := \{\tilde{x} : \tilde{x}^T P \tilde{x} \leq c_\delta\} \tag{40}$$

Assumption 6: The initial condition $\tilde{x}_0$ and r satisfy

$$\tilde{x}_0 \in \mathbb{X}_\delta, \qquad |Hr| \leq \delta_2 u_{\max} \tag{41}$$

where the positive scalar δ_2 is such that $\bar{\delta} = \delta_1 + \delta_2 \in (0, 1)$.

Therefore, $\forall \tilde{x}_k \in \mathbb{X}_\delta$ one has

$$|u_{L,k}| \leq |K\tilde{x}_k| + |Hr| \leq \bar{\delta} u_{\max} < u_{\max} \tag{42}$$

Consequently, the linear part of the controller will never exceed the saturation.

E. Nonlinear Feedback Law

The nonlinear part of the controller has the structure

$$u_{N,k} = \rho(r, y) B_N^T P (A + BK) \tilde{x}_k \tag{43}$$

with P the positive matrix solution of the Lyapunov equation

$$P = (A_N + B_N K)^T P (A_N + B_N K) + W \tag{44}$$

and $W \in \mathbb{R}^{n\times n}$ a positive definite matrix.

Theorem 1: (From [14]) Consider the system in equation (36) with $d_k = 0$, and with $u = u_{L,k} + u_{N,k}$ as described in equations (39) and (43), respectively. Then, for any non positive $\rho(r, y)$, locally Lipchitz in y, and

$$|\rho(r, y)| \leq \rho^* = 2(B_N^T P B_N)^{-1}, \tag{45}$$

the controlled output y_k asymptotically tracks the step command input of amplitude r from an initial state x_0, provided that the conditions in equations (40) and (41) are satisfied.

F. Perturbation Estimation for a Feed–forward Law

Assuming continuity assumptions on d_k and a small enough sampling time, it is possible to estimate the perturbation d_k using its previous value d_{k-1}. Towards this goal, consider that

$$\hat{d}_k = d_{k-1} = x_k - A_N x_{k-1} - B_N \text{sat}(u_{k-1}) \tag{46}$$

where $\hat{d}_k$ is the estimation of d_k.

When the system in equation (36) is put together in closed-loop with the feedback control law in equation (45), it is not robust against uncertain terms or external perturbations, which in this case are modeled by d_k. To overcome this problem, the following theorem from [14] is recalled.

Theorem 2: (From [14]) Consider the nominal system model in equation (36). The control law u_k in equation (47) guarantees the output tracking of a step command input of amplitude r, provided that the conditions in equations (40) and (41) are satisfied, and also that $d_{\max} \leq (1 - \bar{\delta}) u_{\max}$

$$u_k = u_{L,k} + u_{N,k} - (B_N^T B_N)^{-1} B_N^T \hat{d}_k \tag{47}$$

where $u_{L,k}$, $u_{L,k}$ and $\hat{d}_k$ are defined in equations (39), (43), and (46), respectively, and $\bar{\delta}$ comes from the conditions in equations (40) and (41).

Remark 4: In order to reduce the noise commonly encountered in measurements obtained during real-time applications, the estimation of the perturbation can be filtered using a low-pass filter. This allows obtaining an exponentially weighted moving average as [27]

$$d_k^f = \alpha_f \hat{d}_k + (1 - \alpha_f) d_{k-1}^f \quad \text{with} \quad d_0^f = \hat{d}_0, \tag{48}$$

with d_k^f being the filtered disturbance, and $\alpha_f \in (0, 1]$ the filter constant.

G. Main Theorems: Robust controller with perturbation estimation

The main theoretical contribution of this work is provided next, in the form of two Theorems. In particular, Theorem 3 addresses a robust stability for a set Ω_d of SISO LTI systems including dead-times, while Theorem 4 addresses a nonlinear controller guaranteeing the output tracking of a step command input of amplitude r for the systems in Ω_d.

Theorem 3: (H_∞ *Control for LPV systems with dead-time*) Consider a set Ω_d of SISO LTI systems with dead–time of the form

$$\begin{aligned} \eta_{k+1} &= A_d \eta_k + B_d \text{sat}(u_{k-\tau}) \\ y_k &= C_d \eta_k \end{aligned} \tag{49}$$

and identical C_d matrices. Introducing the change of variable defined in equation (27) for each system in Ω_d, this lasts are reduced creating a new set Ω of reduced LTI systems. The set of systems from Ω can now be rewritten as a parameter variant LTV system with no dead–time of the form

$$\begin{aligned} x_{k+1} &= A(\xi_k) x_k + B(\xi_k) \text{sat}(u_k) \\ y_k &= C x_k \end{aligned} \tag{50}$$

Then, considering matrices $D_i = 0$, $E_i = B_i$, and $F_i = C$, and considering that equation (34) holds, a linear robust controller with the form of equation (35) stabilizes the set of systems Ω_d.

Proof: The stability of the closed-loop for each system in the set Ω is given by *Lemma* 2. Then, the stability of the closed-loop for each system in the set Ω_d derives from *Lemma 1*. ■

Theorem 4: (*Robust Control for LPV systems with dead-time and input saturation*) Consider a set Ω_d of SISO LTI systems with dead–time having the form described in equation (49), with a feedback controller K having the form of equation (35) and computed as described in Theorem 3. Rewrite the set Ω_d in the form of equation (50). Define a reference r, an equilibrium point x_e, and an initial condition $\hat{x}_0$ in such a way that Assumptions 5 and 6 hold for all the N vertices. If necessary, for these assumptions to hold the term D_i in equation (34) is modified to a new $D_i = D_{opt}$ for a new synthesis of the control matrix K. Define a value for ξ_k in equation (50). By doing this, a nominal model (A_N, B_N, C_N) is obtained, and supposing that *Assumption 3* and *Assumption 4* hold, the system in equation (50) is rewritten in the form

$$\begin{aligned} x_{k+1} &= A_N x_k + B_N \text{sat}(u_k) + d_k \\ y_k &= C_N x_k \end{aligned} \tag{51}$$

where the term d_k concentrates the differences between the parameter–variant LTV model and the nominal model, together with the possible external disturbances. Define $\rho(r, y)$ in such a way that equation (45) holds, and select W as the required positive definite matrix in equation (44). Then, the control law u_k in equation (47) guarantees the output tracking of a step command input of amplitude r.

Proof: The overall closed–loop stability proof is straightforward from *Theorem 1*, *Theorem 2* and *Theorem 3* . ■

Taking into account these theoretical results, the real–time identification and control of the Mambo UAS platform is now formulated and addressed from a practical point of view.

V. System Identification for the Parrot Mambo

This section provides an overall description of the software and hardware that make up the experimental test–bed available at the TAMU–CC USL. A description of the experiment designed to accomplish the system identification of the Mambo UAS platform is also provided.

Fig. 3. The flying area of the TAMUCC's Unmanned Systems Laboratory.

A. The Parrot Mambo UAS Multicopter

The Mambo quad rotorcraft UAS, see Figure 1, is part of the Parrot's MiniDrones family [28]. The platform dimensions are 180mm x 180mm x 40mm, with a weight of 70 grams. The Mambo sensor suit includes a downwards facing camera, an ultrasonic sensor, a barometric sensor, and an Inertial Measurement Unit (IMU), all of which are used for stabilization purposes. Power is provided by a 3.7V 550 mAh 2Wh Li-Po battery, enabling a flight time of 7min to 10min, depending on the nature of the task to be performed and the accessories implemented. Making use of the pitch and roll attitude angles as virtual control inputs, the maximum flight speed is internally governed by the onboard autopilot at a magnitude of ≈ 8 m/sec. The communication with the vehicle is established via Bluetooth Low Energy (BLE), or via WiFi if the First Person View (FPV) flight accessories are used. As reported in [23], the control architecture running on the autopilot of the Parrot AR.Drone, a UAS platform similar to the Mambo and developed by the same manufacturer, is composed by two nested loops. The first loop corresponds to the Attitude Control Loop, which computes an angular rate set–point from the difference between the estimated attitude and the attitude set–point. This angular rate is tracked with a proportional integral (PI) control. The second loop is the Angular Rate Control Loop, which controls the platform motors with simple proportional controllers.

B. The Experimental Test-bed

The experimental test-bed available at the TAMU-CC USL is supported by a 12-camera MCS developed by Vicon, see Figure 3 for a snapshot of the flying area, as well as Figure 2 for a corresponding block diagram representation. This system is capable of providing the position and orientation (ground truth) of multiple robots at rates of up to 500 measurements per second, and with a 0.1mm precision. In other words, the MCS essentially provides fast and reliable *indoor GPS* for testing and validation of modeling and control solutions. A computer is used as GCS, and it is here that the user–developed estimation and control strategies are executed. For the experimental application addressed in this research, the GCS computer receives the states of the robotic agent(s) via the MCS at a sample rate of 100Hz. Next, the GCS use this data to calculate the corresponding control signals, which are then normalized in the set $[-1, 1]$, and then sends them back to the robotic agent(s) via BLE.

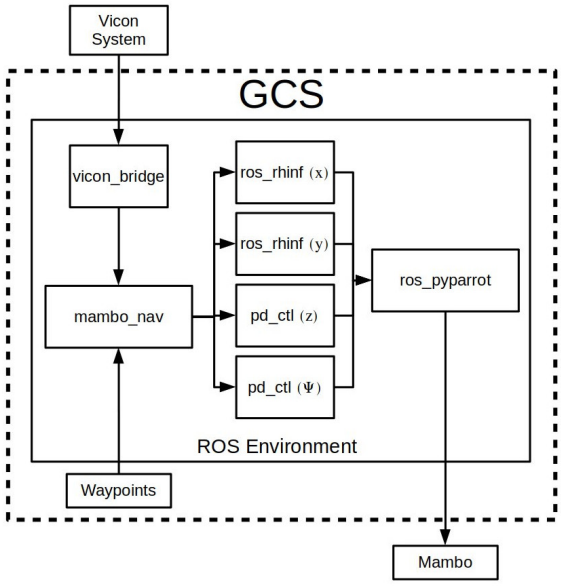

Fig. 4. The Robot Operating System (ROS) platform and virtual (software) modules implemented in the experimental application. The block diagram also shows how these modules interact with the physical system.

C. Software running in the GCS of the test-bed

The Robot Operating System (ROS) platform is implemented in the GCS for controlling the UAS during the real-time experiments. ROS allows joining multiple *virtual modules* (i.e., software modules) as shown in Figure 4, for estimation and control purposes. A description of these modules is provided next.

- **ros_pyparrot**: enables the communication between ROS and the UAS driver (pyparrot). Developed in [29].
- **vicon_bridge**: translates data coming from the MCS data stream (DataStreamSDK) and converts it into the appropriate ROS format. Developed in [30].
- **ros_rhinf**: a module developed by TAMU-CC USL, containing the novel H_∞ controller proposed in Section IV. This module governs the X and Y translational dynamics of the UAS.
- **pd_ctl**: a module containing a PD Controller that governs the vertical and heading dynamics of the Mambo UAS.
- **mambo_nav**: a module developed by TAMU-CC USL, in charge of the interaction between the MCS signals and the control signals, ensuring both frames of coordinates are consistent.

The source code that controls the communication with the Mambo UAS, i.e., **ros_pyparrot**, is further modified in such a way that the user can design and introduce a known and repeatable perturbation to the system. This additional signal is considered in this research as an undesired control signal affecting the system. By injecting the signal in this way, the control strategy in charge of stabilizing the translational

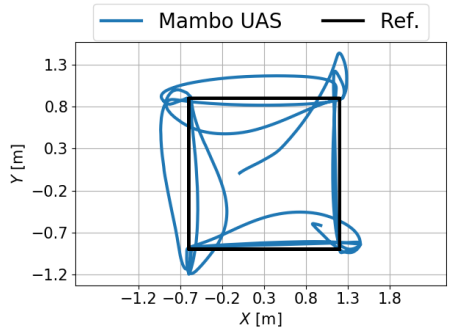

Fig. 5. Experiment performed to extract the input–output data pairs for model identification. Making use of PD controllers, the UAS flew a squared trajectory at 1m above the floor (i.e., the X–Y plane) of the laboratory flight arena.

dynamics of the UAS is unaware of the perturbation, nor its specific characteristics. Notice that, for the internal controller running onboard the Mambo autopilot, this perturbation will be observed as part of the desired set–point affecting the behavior of the system in a similar way than the control input.

D. Design of the Identification Experiment

A set of real-time experiments were designed with the objective of obtaining rich and sufficient information about the Mambo UAS. The goal is to generate input–output data pairs, in such a way that it is possible to guarantee the convergence of the identification algorithm. In these tests, the Mambo UAS was tasked to fly in a squared trajectory at an altitude of 1m above the floor of the laboratory flying arena, i.e., above the the X-Y plane of the inertial coordinates frame. The control strategy implemented for the identification experiments corresponded to a PD controller, which was tuned heuristically.

Four different Mambo platforms were used in the identification tests. Specifically, a set of two similar flights were performed, each one of these sets with a different Mambo platform. During each flight, the GCS generated and recorded the control inputs required for performing the squared motion in the X–Y plane. Simultaneously, the GCS obtained the Mambo's translational states via the MCS. The input–output data pairs were collected at a rate of 100Hz. For illustrative purposes, the recorded UAS states showing the squared motion, as well as the corresponding control input signals are shown in Figure 5 and Figure 6, respectively.

The practical considerations previously introduced in Section III-D to ease the system identification are now referred to. Taking into account that the quad rotorcraft model from translational *velocity* to *position* is described by a pure integration (see for example our previous works [5], [31], and [32]), the position of the UAS, which is provided by the MCS, is differentiated once in order to identify the model from the *input* to the *velocity*. This identification will also contain the dynamics of the internal control strategy running onboard the Mambo autopilot. When choosing the order of the model, the trade off between an *accurate but complex* model and a *uncertain but simple* model is also taken into account. Furthermore, there are two important aspects to be considered

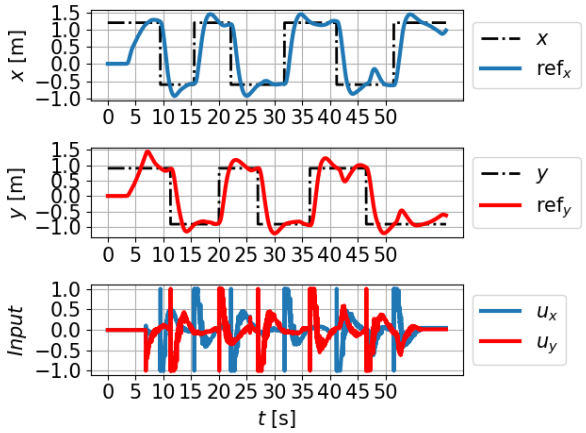

Fig. 6. An example of the signals representing the X and Y states of the Mambo the flight experiment. The blue and red colors are associated with the X and Y dynamics, respectively. The upper plot shows the X translational motion and its corresponding reference. The center plot shows the Y translational motion and its corresponding reference. The lower plot shows the corresponding control signals.

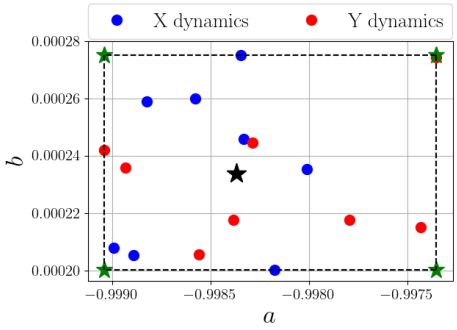

Fig. 7. Parameters for the identified X–Y translational models. The results come from two experiments, from each one of the four Mambo UAS. The dots blue ● and red ● are associated with X and Y dynamics, respectively.

when doing the system identification: (i) it is more desirable to numerically solve a simpler LMI problem, and (ii) since uncertainties in the model are considered, it is also preferred to choose a low order model.

Taking into account these previous considerations, a first order model is selected for the translational dynamics identification, together with a first order model for the noise identification. The proposed model is then of the form

$$y_k = -ay_{k-1} + bu_{k-\tau-1} + ce_{k-1} + e_k \tag{52}$$

From each experiment performed, four parameters were identified, specifically: a, b, c, and the dead-time τ. In particular, for the X translational model identification, the a and b parameters obtained are between the maximum and minimum values of

$$\begin{aligned} a_{\max} &= -0.9973 \qquad b_{\max} = 0.2752 \cdot 10^{-3} \\ a_{\min} &= -0.999 \qquad b_{\min} = 0.2003 \cdot 10^{-3} \end{aligned} \tag{53}$$

These numerical values are plotted in Figure 7 as blue dots. The numerical values of the c parameters are approximately $c = 1 \cdot 10^{-3}$. It is also verified that these parameters minimize the MSE/MSO ratio, i.e., they provide the best quality of

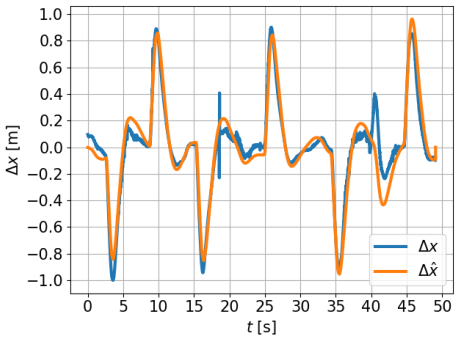

Fig. 8. Identification results for the Mambo UAS dynamics in the X coordinate with a quality of fit $MSE/MSO = 0.32$. The blue line (i.e., Δx) represents ground truth for the velocity. The orange line (i.e., $\Delta\hat{x}$) represent the output obtained from the identified first order nominal model.

fit. Furthermore, from the analysis of multiple experimental outcomes, it is concluded that a value of $\tau = 17$ is the one that minimizes the performance index I_i as calculated from equation (25).

From a practical point of view, the Mambo UAS robotic platform is symmetric. Therefore, the behavior in the X and Y dynamics should be very similar. For this reason, the parameters obtained in both the X and Y directions were combined and used as a single and richer set of information. The overall set of a and b parameters obtained from the X and Y identification experiments are show in Figure 7. In this plot, the blue dots correspond to the parameters for the X dynamics, while the red dots correspond to the parameters for the Y dynamics. The four green stars ($\star$) placed at the corners of the rectangle are the vertices associated with the combination of the maximum and minimum values for the a and b parameters. The black star ($\star$) close to the center represents the mean (i.e., the average) model, which is now called the nominal model. The nominal, the minimum, and the maximum numerical values, as shown in Figure 7, will be hereafter used in equation (31).

E. Validation of the identification results

Figure 8 shows the validation of the identification results for the Mambo UAS dynamics in the X translational displacement. The blue line (i.e., Δx) represents ground truth for the velocity in the X motion. The orange line (i.e., $\Delta\hat{x}$) represent the output obtained from the identified first order nominal model, specifically $a_m = -0.9984$ and $b_m = 0.2343 \cdot 10^{-3}$. Notice that the identified model behaves very similar to the real velocity dynamics, i.e., to the Δx obtained from ground truth. A similar identification procedure was performed for identifying and validating the Y translational dynamics. The outcomes of this experiment exhibited also a good agreement between the identified model and the ground truth.

VI. Controller Synthesis

The identification of the nominal model and associated variations make it possible to synthesize a robust controller for the Mambo UAS platform. The first step consists on writing the Mambo UAS model as in equation (31). To accomplish this goal, the model is written as a state space model, based on the knowledge that the Mambo UAS translational system, for the X and Y dynamics, is composed by an integrator plus a first order system. Specific details about this procedure are provided in [26].

Defining the state vector $\bar{x}_k = [x_k, \Delta x_k]$, and considering the input u_k and the output y_k, the following system structure is obtained

$$\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}u_k, \quad y_k = C\bar{x}_k + Du_k \tag{54}$$

with matrices

$$\bar{A} = \begin{bmatrix} 1 & 1 \\ 0 & -a \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0 \\ b \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0 \tag{55}$$

where $a \in (a_{min}, a_{max})$ and $b \in (b_{min}, b_{max})$ are the model parameters, with its minima and maxima, as identified in Section V.

Taking into account the practical considerations introduced in Section III-D, and also for enabling a better numerical conditioning when computing equations (27) and (28), the model is down–sampled to a number equal to τ. By doing this, the new time delay (τ_n) is equal to just one sampling time. This change implies that the matrices of the down–sampled model $A_{d\tau}$ and $B_{d\tau}$ are now computed as

$$A_{d\tau} = \bar{A}^{\tau} \quad \text{and} \quad B_{d\tau} = \sum_{j=0}^{\tau-1} \bar{A}^j \bar{B} \tag{56}$$

To remove the delay we use the reduction method explained in section III-C. With the downsampled delay $\tau_n = 1$ sampling time, the system is reduced using equation (28) as

$$A = A_{d\tau} \quad \text{and} \quad B = A_{d\tau}^{-1} B_{d\tau} \tag{57}$$

The numerical value of the vertices A_i and B_i are obtained from the varying parameters a and b. As mentioned in *Theorem 4*, in order to guarantee the conditions in equations (40) and (41), the D_i matrix is heuristically tuned to a value of $D_{\text{opt}} = 0.75$. Finally, the optimization problem in equation (34) is solved using the CVXPY [33] and CVXOPT [34] libraries for Python language with SciPy [35]. The controller gain matrix obtained is $K = [-1.0825 \; -72.0064]$, with an H_∞ performance $\mu = 0.9333$.

A. Robust controller with perturbation estimation

To compute the controller from *Theorem 4*, the numerical values for the nominal model in equation (36) are obtained from matrices A and B using the nominal values a_m and b_m

$$A_N = \begin{bmatrix} 1 & 16.8 \\ 0 & 0.975 \end{bmatrix}, \quad B_N = \begin{bmatrix} 0.0319 \\ 0.00397 \end{bmatrix}, \quad C_N = [1 \; 0] \tag{58}$$

and the function $\rho(r, y_k)$ is adopted from [14] as

$$\begin{aligned} \rho(r, y_k) &= -3.686|e^{-1.1|e_k|} - e^{1.1|e_0|}| \\ e_k &= y_k - r \end{aligned} \tag{59}$$

with $e_0 = 0$, the term $W = 0.1I_d$ is heuristically tuned, and the discrete-time Algebraic Riccati Equation (44) is solved with a Scipy tool in order to obtain

$$P = \begin{bmatrix} 0.0264 & 0.2878 \\ 0.2878 & 5.0584 \end{bmatrix} \tag{60}$$

Finally, the filter from equation (48) is tuned with an $\alpha_f = 0.1$, in order to filter the measurements noise.

Next, the performance of the proposed control strategy is validated in a set of real–time experiments making use of four different Mambos UAS platforms.

VII. Real-time implementation of the estimation and control strategies

The proposed control strategy is implemented and tested in real–time experiments making use of the UAS experimental test-bed available at the TAMU-CC USL, see Figures 2 and 3. These tests, which involved four different Mamo UAS platforms, are described next.

A. Real–time control strategy for the Mambo UAS platform

The experimental scenario adopted is similar to the one used in Section V-D. Each one of the four UASs is independly tasked to fly a squared trajectory in the X–Y plane, at $1m$ above the floor. The outcome of these tests are shown in Figure 9, where it can be observed that all the UAS follow satisfactory the given reference. Figure 10 shows the reference signals as well as the trajectories flown for the X and Y directions. The corresponding control signals are shown in Figure 11. In these two Figures, the data associated with each one of the four UASs is labeled as M_i for $i = [1, 2, 3, 4]$. Notice that some of the control signals in Figure 11 exhibit saturation. Still, all the identified models present marginally stable dynamics, allowing the presence of saturation in the control signals, without leading to an unstable behavior.

Remark 5: The saturation effect can be explained by considering the uncertainty existing in the high frequency range for the identified UAS models. Indeed, since no identification was performed for high frequencies, the controller observes the rapid change of navigational waypoints as a perturbation. This ultimately leads to the control signal saturation.

A video showing the execution of this real–time experiment can be found in the following link: https://youtu.be/ZNKXFGFx2Xo

B. Real–time control without perturbation estimation

A second experiment is performed under a similar scenario, but now with the additional challenge that a perturbation is added to the control signal of the X-axis. The addition of the perturbation control signals was performed based on the procedure explained in Section V-C. For these tests, a constant value corresponding to 30 percent of the input saturation value for the X-axis control is considered.

Figure 12 shows the X–Y trajectories of the four Mambo UAS platforms, as well as the reference signal (square). Notice that the proposed controller stabilizes the drones, but the

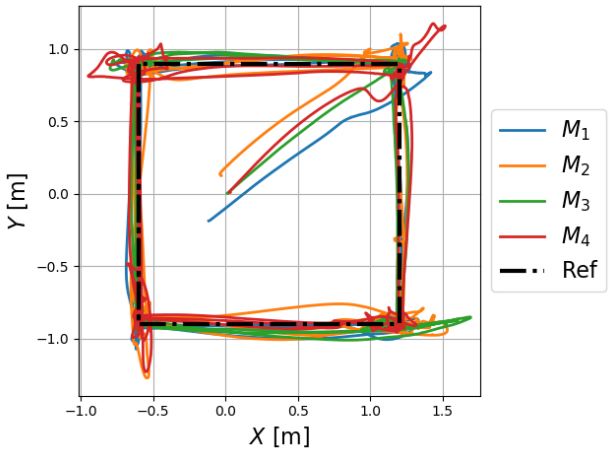

Fig. 9. Experiments performed to illustrated the performance of the robust controller on four different Mambo UAS. The vehicles were flown in a squared trajectory the X–Y plane, at $1m$ above the floor. The trajectory was performed counter–clockwise, staring from the top-right corner.

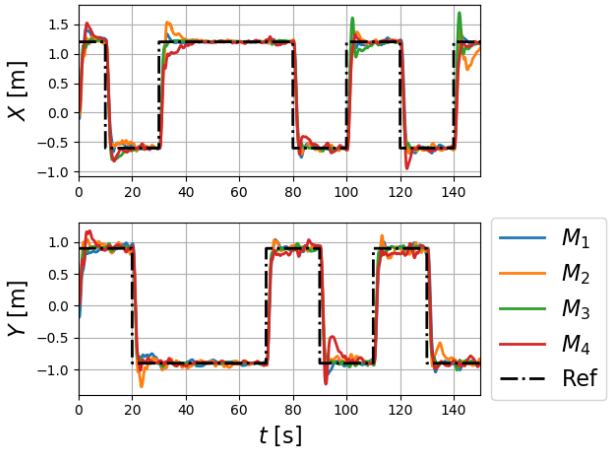

Fig. 10. The X–Y motion of four different Mambo UAS platforms during a real–time experiment. The nonlinear controller in equation (47) is used during this test. The reference tracking is effectively performed by all the vehicles, with slightly different levels of overshoot.

perturbation produces an offset in the X-axis. Additionally, Figures 13 and 14 show the X and Y dynamics of the UASs, as well as the corresponding control inputs when the linear controller from *Lemma 1* is applied.

A video showing the execution of this real–time experiment can be found in the following link: https://youtu.be/cCiboUIuOSs

C. Real–time control with perturbation estimation

Figure 15 shows the X–Y trajectories of the four Mambo UAS platforms as well as the squared reference signal when the nonlinear controller in equation (47) is applied, which includes the perturbation estimation. Figures 16 and 17 show these trajectories independently, as well as the corresponding control inputs. From these results, it is possible to observe that there is no offset between the UAS trajectories and the reference signal. Notice that some overshoots can be observed

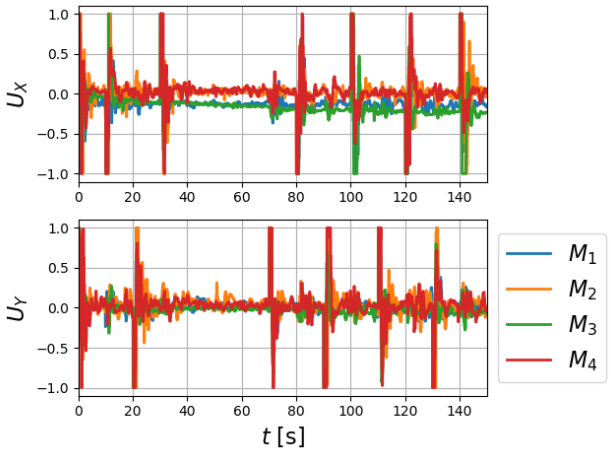

Fig. 11. Control signals generating the the X–Y motion of the four Mambo UAS platforms. Some of the control signals exhibit saturation, and for two of the UASs, an offset can be observed in the X-axis control.

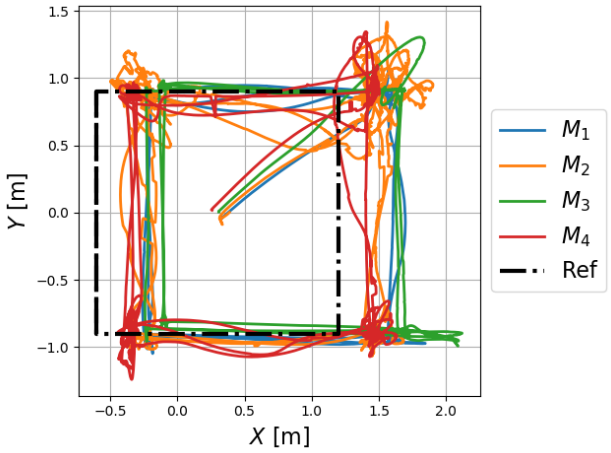

Fig. 12. The performance of the linear robust controller without perturbation estimation, executed on four different Mambo UAS. All UAS were flown in square trajectory the X–Y plane at $1m$ above the floor. Notice that the controller stabilizes the drones, but the perturbation produces an offset in the X-axis.

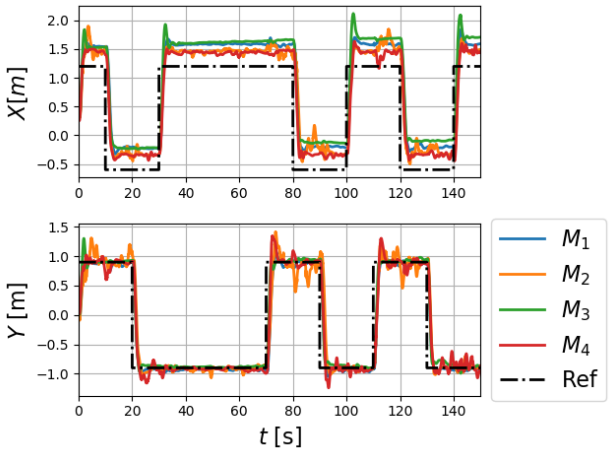

Fig. 13. Experimental results obtained with the linear controller introduced in Lemma 1. The trajectories are stable, but an offset appears due to the unknown perturbation.

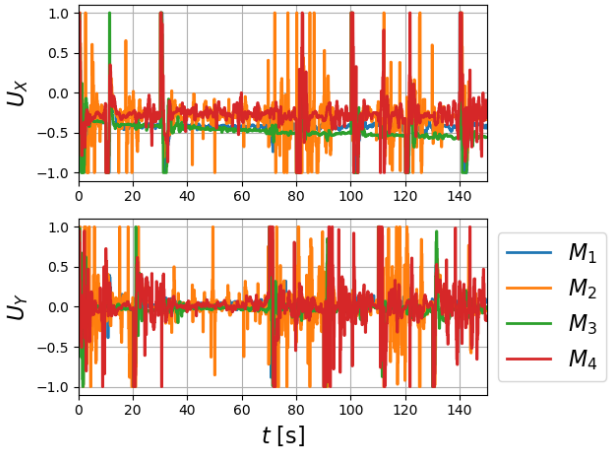

Fig. 14. Control inputs for the X and Y dynamics of four Mambo platforms. The offeset observed in the X-axis control input is due to the perturbation added in the communication module. The perturbation has a magnitude of 0.3 (30% of the maximum value of the control input). This addition virtually changes the saturation limits for the controller action, going from the nominal range of $[-1, 1]$ values to $[-1.3, 0.7]$ values.

for the X-axis dynamics. This behavior comes from the fact that the perturbation added inside the UAS driver changes the control action from its original interval of $[-1, 1]$ values (a hardware limitation) to a new one of $[-1.3, 0.7]$ (a hardware limitation plus a software alteration).

A video showing the execution of this real–time experiment can be found in the following link: https://youtu.be/3VxMfX5imto

D. Validation of the perturbation estimation technique

To validate the perturbation estimation approach a couple of experiments were designed and executed. The first one corresponds to the *no perturbation* case and the second one to the scenario where a perturbation affects the system. For these tests, the Mambo platform associated with the label M_4 was implemented. Figure 18 shows the (filtered) values of the perturbation estimation obtained by applying equation (48). The upper plot illustrates the first component of d_k^f, while the lower part shows the second component.

In the case studied during the first experiment (no perturbation), some values different from zero are observed in the (blue) signal. These disturbance values are observed every time the reference is changed. Therefore, the perturbation estimated at those moments corresponds to the effect from high frequencies, which were not totally identified as explained in Remark 5. In the second experiment, a perturbation of 0.3 (30% of the maximum value of the control input) is applied and combined with the control input, with the procedure explained in Section V-C. From Figure 18 it can be observed that the first and second components of d_k^f converge to values of 0.01 and 0.001, respectively. Recalling that the perturbation is added to the control input, the theoretically values of d from equation (36) can be computed multiplying the value of B_N

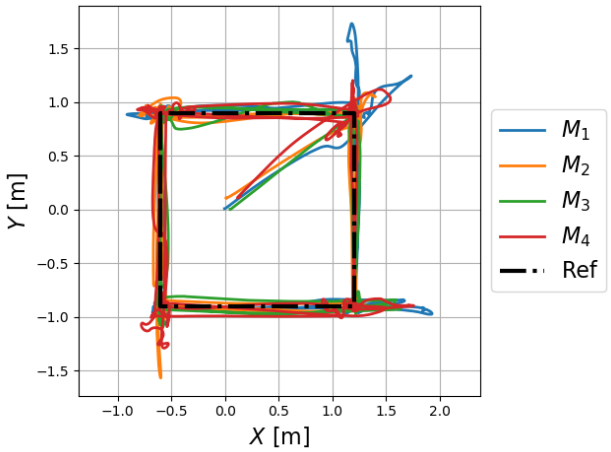

Fig. 15. Real–time performance of the linear robust controller without perturbation estimation. These tests were performed in four different Mambo UAS platforms. The consist a

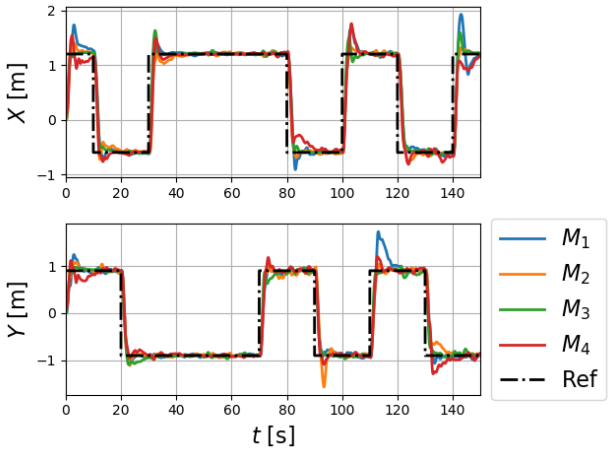

Fig. 16. Experimental results illustrating the control performance of the robust controller over four different Mambo UAS platforms. Overshoots can be seen when the X-axis reference is positive, which is due to the combination of the hardware saturation and the addition of the perturbation to the input.

by the value of the perturbation. This procedure results in the numerical values of 0.00957 and 0.001191, which are very closed to the estimated values.

As an additional measurement of performance, the energy spent by the control action is evaluated using the performance classical index *IVU*, which stands for Integral of Time-weighted Variability of the Signal Control, defined as in [36]:

$$IVU = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(u_i - u_{mean})^2} \tag{61}$$

with u_i being the input to evaluate, $u_{i,mean}$ the mean value of that input, and N the lenght of the input signal. In order to compare the linear controller with respect to the nonlinear one including the disturbance estimation, the *IVU* is computed using the control input signals generated by both techniques for the particular case of the second experiment, i.e., for the experiment including the perturbation. The data therefore cor-

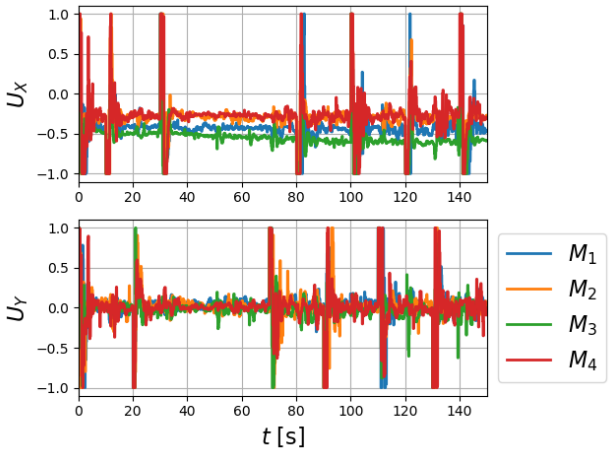

Fig. 17. Control inputs for stabilizing the the X and Y dynamics of the UAS in real–time. The offset is due to the perturbation effect. Note the different mean values of these input signals are obtained for each UAS. This is due to the fact that, despite being similar platforms, their particular manufacturing process and different levels of war and tear ultimately lead to different dynamic models.

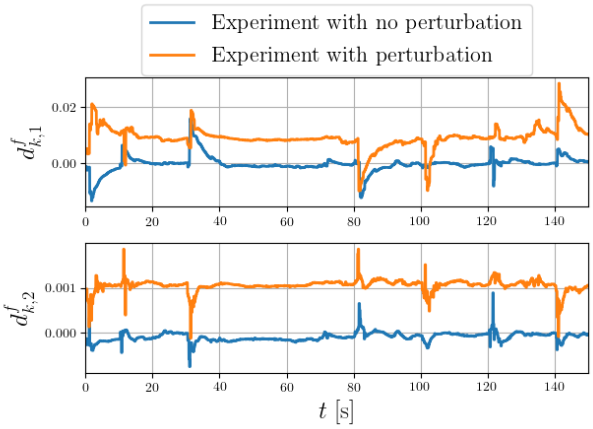

Fig. 18. Disturbance estimation applied to the Mambo M_4 platform. Two experiments where performed, one with no perturbation affecting the system and another with a perturbation having a magnitude of 30% of the maximum value of the control input. The upper plot shows the first component of the perturbation, while the lower plot shows the second component. The first and second components of d_k^f converge to values of 0.01 and 0.001, respectively. In both experiments the perturbation estimation algorithm captures the effect from high frequencies not identified during the system identification procedure (as detailed in Remark 5).

responds to the signals in Figures 14 and 17. The conclusion is that, in average, there is an efficiency improvement of 30%.

E. Performance evaluation for tracking a circular trajectory

An additional experiment is executed, where each UAS is individually tasked to follow a smooth circular trajectory in the presence of a perturbation in the X dynamics. The circular trajectory has a radius of 0.75m, centered at coordinates $(0, 0)$, parallel to the X–Y plane, and placed at an altitude of 1.1m. The way-points of the trajectory are updated at a speed of 0.20944 rad/sec. The control strategies implemented in this task correspond to (i) linear robust controller without

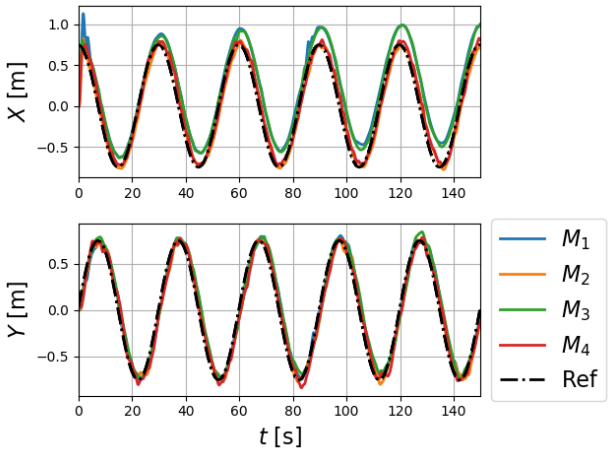

Fig. 19. Performance of the linear robust controller over four different Mambo UAS platforms. The UASs execute a circular trajectory in the Plane X–Y. Notice the presence of an offset in the X direction due to a drift phenomena.

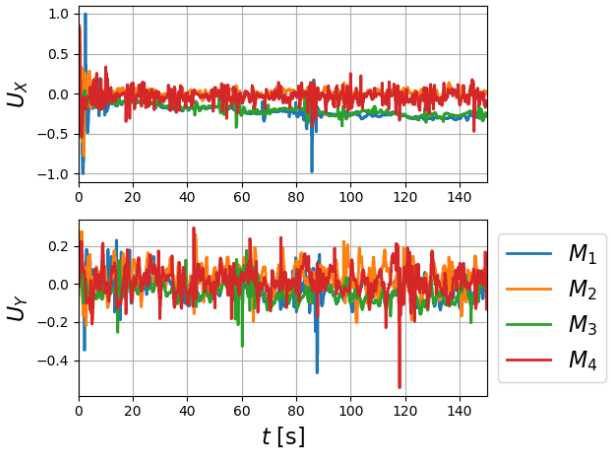

Fig. 20. Control inputs generated with the linear robust controller for the circular trajectory tracking task.

perturbation estimation, and (ii) non linear robust controller with perturbation estimation.

The performance of the linear robust controller is discussed first. Figure 19 shows the sinusoidal reference signals and the corresponding states of each one of the four UASs. The information corresponding to the X and Y dynamics are shown in the upper and lower plots, respectively. The corresponding control signals generated during this experiment are shown in Figure 20. Finally, Figure 21 shows the 2-dimensional representation, in the X–Y plane, of the trajectories executed by each one of the four UASs. From this results, it can be observed the presence of a drift phenomena in the X dynamics that generates an offset. Therefore, the circular trajectories are not centered at the origin. Not surprisingly, and similar to the results obtained while performing the squared trajectory, the linear robust controller is not able to overcome this issue when tracking the circular reference.

The performance obtained when using the non linear robust controller with perturbation estimation is now discussed. Figure 22 shows the sinusoidal reference signals and the corresponding states of each one of the four UASs. The information corresponding to the X and Y dynamics are shown in the upper and lower plots, respectively. The corresponding control signals generated during this experiment are shown in Figure 23. Finally, Figure 24 shows the 2-dimensional representation, in the X–Y plane, of the trajectories executed by each one of the four UASs. From this results, it can be observed that the drift phenomena in the X dynamics are successfully handled by the non linear robust controller with perturbation estimation. The circular trajectories are now tracked effectively, and they are centered at the origin.

A video showing the performance of the four different Mambo UAS platforms while performing the circular trajectory, and while using the two controllers just described, can be found in the following link: https://youtu.be/VW3uOdUgaac

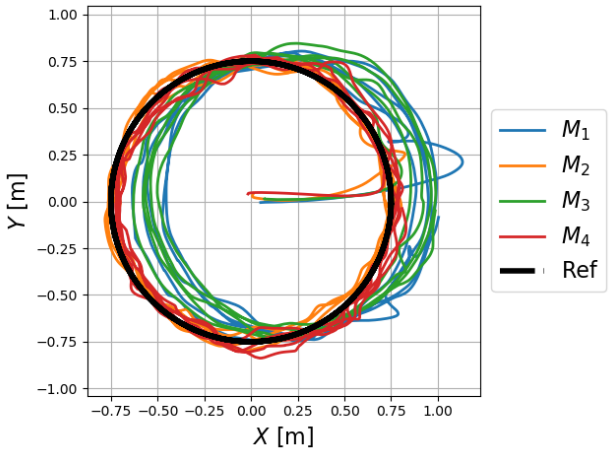

Fig. 21. Each one of the four Mambo UAS platforms performed 5 circular trajectories in the X– Y plane. The real–time experiment was executed using the proposed linear robust controller without the perturbation estimation technique. Due to the drift phenomena, an offset can be observed in the X coordinate.

VIII. Conclusions and Future Research

This paper presented the real-time implementation of a novel X–Y translational robust control strategy for a commercially–available and ready–to–fly UAS platform: the Mambo^{TM}quad rotorcraft developed by Parrot SA®.

First, and making use of an UAS experimental test–bed equipped with an MCS, real–time input–output data–sets were obtained from multiple real-flight experiments, which involved a set of four Mambo UAS platforms. An integrator plus a first order structure with dead-time delay was chosen as the mathematical model for the translational dynamics of the Mambo UAS. Next, the X and Y translational dynamics of each UAS were identified applying an ELS algorithm to the input–output data pairs. From the collection of linear models – all of which have the same structure but exhibit different parameters – and making use of a reduction technique, a new collection of linear models without dead-time delay was obtained.

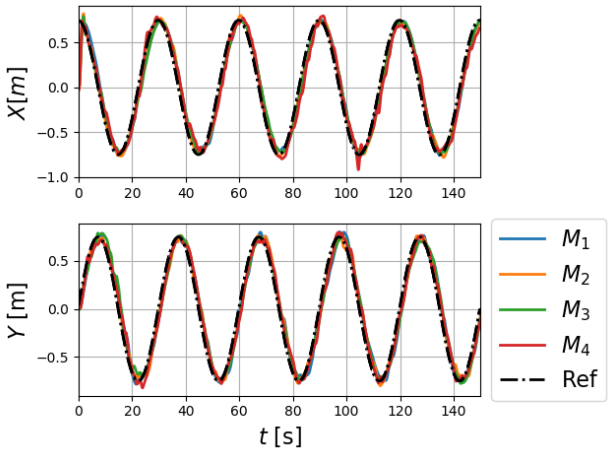

Fig. 22. Performance of the nonlinear robust controller with perturbation estimation over four different Mambo UAS platforms. The UASs execute a circular trajectory in the Plane X–Y. Notice that the offset in the X direction, which was due to the perturbation, has been satisfactorily removed.

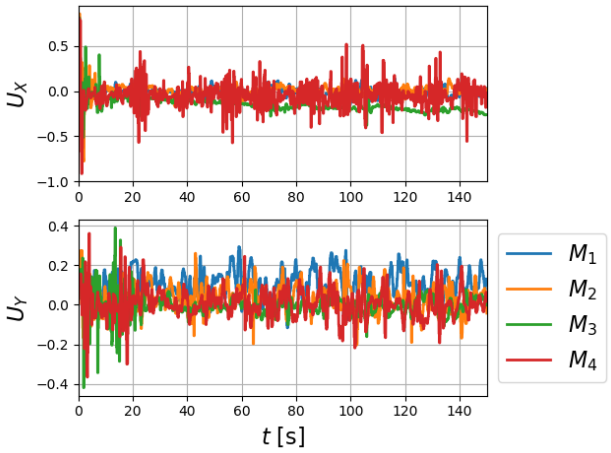

Fig. 23. Control inputs generated with the nonlinear robust controller with perturbation estimation for the circular trajectory tracking task.

The control strategy was then designed based on a parameter–variant discrete–time linear system obtained from all the reduced X–Y translational models. On a first stage, a robust controller to guarantee the global stability for the translational dynamics of the UAS was proposed, which was synthesized with a minimal H_∞ norm. Next, a robust nonlinear controller enhanced with a disturbance estimation was developed in order to guarantee effective reference tracking in the presence of (i) disturbances and (ii) parameter deviation from the nominal model. These novel theoretical results were introduced as a couple of original theorems.

From a technological point of view, real–time experimental flights were performed making use of the proposed robust control techniques over four different Mambo UAS platforms. The effectiveness of the the proposed robust control approach for stabilizing these systems during real–time flights, despite the presence of modeling uncertainties, delays, and perturbations, was successfully accomplished.

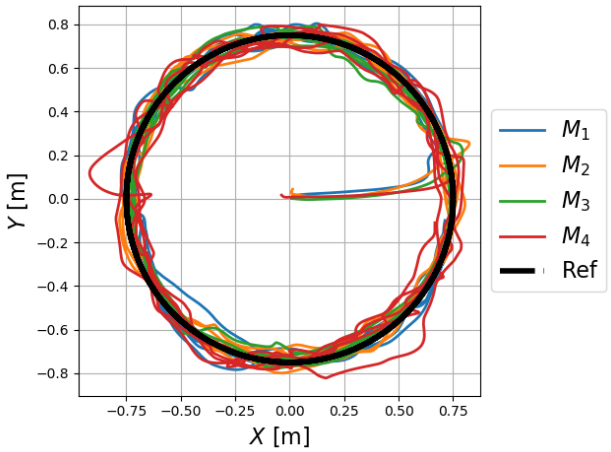

Fig. 24. Each one of the four Mambo UAS platforms performed 5 circular trajectories in the X– Y plane. The real–time experiment was executed using the nonlinear robust controller enhanced with the perturbation estimation technique. Notice that the offset in the X direction, which was due to the drift phenomena, has been satisfactorily removed.

Current directions of this research aim at the development of a nonlinear observer to perform the online estimation of the model parameters, ultimately leading to an improvement of the control performance.

ACKNOWLEDGMENT

This work was supported by ARO under grant W911NF1810210 and by NSF under grant EPCN-1608880. The work of G.A. Guijarro Reyes is supported by the Mexican National Council of Science and Technology (CONACYT) under grant 709368.

REFERENCES

[1] J.C. Geromel, P.L.D. Peres and S.R. Souza, "H_∞ control of discrete-time uncertain systems", IEEE Transactions on Automatic Control, Vol 39(5), pp 1072–1075, 1994.

[2] J. Daafouz and J. Bernussou, "Poly-quadratic stability and H_∞ performance for discrete systems with time varying uncertainties. In Decision and Control, Proceedings of the 40^{th} IEEE Conference, Vol. 1, pp. 267–272, 2001

[3] A.P. Pandey and M.C. de Oliveira "Discrete-time H_∞ control of linear parameter-varying systems", International Journal of Control, Taylor & Francis, pp. 2750-2760, April 2018.

[4] X. Chang, R Liu and J.H. Park "A Further Study on Output Feedback H_∞ Control for Discrete-Time Systems, IEEETransactions on Circuits and Systems II: Express Briefs", Early Access, 2019

[5] L.R. Garcia Carrillo, A. Dzul, R. Lozano and C. Pégard. Quad Rotorcraft Control: Vision-Based Hovering and Navigation. *Springer*, ISBN-10: 1447143981, September 2012.

[6] T.C. Hsia, "On least squares algorithms for system parameter identification", IEEE Trans. Automat. Control, pp 104–108 1976.

[7] M.S Ahmed "Rapid parameter estimation algorithms using the ELS principle", Systems & Control Letters, Elsevier, Vol 2, Num 4, pp 209–216, 1982

[8] L. Ljung "System Identification: Theory for the User", Prentice Hall, 1999

[9] Unmanned Systems Laboratory at Texas A&M University - Corpus Christi: https://www.tamucc.edu/~lgarciacarrillo/laboratory.html

[10] N.V. Hoffer, C. Coopmans, A.M. Jensen and Y. Chen, A Survey and Categorization of Small Low-Cost Unmanned Aerial Vehicle System Identification, Journal of Intelligent & Robotic Systems, Vol. (74), pp 129–145, 2014.

[11] T. Ryan and H. Jin Kim, "LMI-Based Gain Synthesis for Simple Robust Quadrotor Control", IEEE Transactions on Automation Science and Engineering, VOL. 10(4), 2013
[12] K. Lee, J. Back, and I. Choy, "Nonlinear Disturbance Observer based Robust Attitude Tracking Controller for Quadrotor UAVs", Int. Journal of Control, Automation, and Systems, Vol. 12(6), pp 1266-1275, 2014.
[13] A. Noormohammadi-Asl, O. Esrafilian, M.A. Arzati and H.D. Taghirad, "System identification and $H\infty$-based control of quadrotor attitude", Mechanical Systems and Signal Processing, Elsevier, Vol. 135, 2020.
[14] T., Binazadeh and M. Bahmani, "Design of Robust Controller for a Class of Uncertain Discrete–Time Systems Subject to Actuator Saturation", IEEE Transactions on Automatic Control, Vol. 62, No. 3, 2017.
[15] T. Binazadeh and M. Bahmani, "Robust Controller Design for a Class of Nonlinear Systems Subject to Actuator Saturation", 26th Iranian Conference on Electrical Engineering, IEEE, 2018.
[16] T. Tran, S.S. Ge, W. He, P. Luu and N. Truong, "Trajectory Tracking Control of a Quadrotor Aerial Vehicle in the Presence of Input Constraints", International Journal of Control Automation and Systems, Springer, Vol. 16, pp 1–11, 2018.
[17] S. Islam, P.X. Liu and A.E. Saddik, "Robust Control of Four Rotor Unmanned Aerial Vehicle with Disturbance Uncertainty", Transactions on Industrial Electronics, IEEE, Vol. 62(3), pp. 1563–1571, 2014.
[18] S.K. Armah, and S. Yi, "Analysis of Time Delays in Quadrotor Systems and Design of Control", Advances in Delays and Dynamics, Springer, pp 299–313, 2017.
[19] S.K. Armah, S. Yi and W. Choi, "Design of feedback control for quadrotors considering signal transmission delays", Int. Journal of Control Automation and Syst., Springer, Vol. 14, pp 1395–1403, 2016.
[20] S.K. Armah, and S. Yi, "Adaptive Control for Quadrotor UAVs Considering Time Delay: Study with Flight Payload" Robotics & Automation Engineering Journal, Vol. 2(5), 2018
[21] H. Liu, W. Zhao ,Z. Zuo, and Y. Zhong, "Robust Control for Quadrotors With Multiple Time-Varying Uncertainties and Delays", Transactions on Industrial Electronics, IEEE, Vol. 64(2), 2017.
[22] R. Sanz, P. García, Q. Zhong and P. Albertos, "Predictor-Based Control of a Class of Time-Delay Systems and Its Application to Quadrotors", in IEEE Trans. on Industrial Electronics, Vol. 64(1), pp. 459–469, 2017.
[23] P.-J. Bristeau, F. Callou, D. Vissière and N. Petit "The Navigation and Control technology inside the AR.Drone micro UAV", Proceedings of the 18th World Congress, IFAC, 2011.
[24] J.E. Normey-Rico and E.F. Camacho, "Control of Dead-time Processes" in Advanced Textbooks in Control and Signal Processing, Springer, 2007
[25] M. Mattioni, S. Monaco, D. Normand-Cyrot, "Nonlinear discrete-time systems with delayed control: A reduction" in Systems & Control Letters Vol. 114, pp. 31–37, 2018.
[26] Ogata, K., "Discrete-Time Control Systems, 2nd Edition", Pearson, 1995.
[27] J.P. Hespanha, "Advanced Undergraduate Topics in Control Systems Design", 2019 https://www.ece.ucsb.edu/~hespanha/published/allugtopics-20190508.pdf
[28] Parrot Mambo UAS website: www.parrot.com/global/drones/parrot-mambo-fpv
[29] A. McGovern, "pyparrot: Python interface for Parrot Drones", Available at https://github.com/amymcgovern/pyparrot.
[30] M. Achtelik, "vicon_bridge", version 1.5.19, Available at https://github.com/ethz-asl/vicon_bridge.
[31] L.R. Garcia Carrillo, G. Flores, G. Sanahuja, and R. Lozano, "Quad Rotorcraft Switching Control: An Application for the Task of Path Following", *IEEE Transactions on Control Systems Technology*, vol. 22, issue 4, pp. 1255-1267, July 2014.
[32] L.R. Garcia Carrillo, I. Fantoni, E. Rondon, and A. Dzul, "3-dimensional Position and Velocity Regulation of a Quad-rotor Using Optical Flow", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, issue 1, pp. 358-371, January 2015.
[33] A. Agrawal, R. Verschueren, S. Diamond and S. Boyd, "A Rewriting System for Convex Optimization Problems", Journal of Control and Decision, Vol 5(1), pp 42–60, 2018
[34] M. S. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: A Python package for convex optimization", version 1.2. Available at cvxopt.org, 2019.
[35] E. Jones, T. Oliphant, P. Peterson and others, "SciPy: Open source scientific tools for Python", scipy.org, 2019
[36] I. Rubio Scola, M.M. Quadros, V.J.S. Leite, "Robust Hybrid PI Controller with a Simple Adaptation in the Integrator Reset State" IFAC-PapersOnLine, Vol.50(1), 1457-1462, 2017.

Ignacio RUBIO SCOLA received an Electronic Engineering diploma from Universidad Nacional de Rosario (UNR), Argentina in 2010, and a Ph.D. degree in Control from Université Grenoble Alpes, France in 2015. He was a Postdoctoral Researcher at CEFET-MG, Brasil between 2015 and 2018 and at Unmanned Syst. Lab. at Texas A&M-CC, USA in 2019. Since 2018, he is a Postdoctoral Researcher at CIFASIS and an Assistant Professor at UNR. He is interested in nonlinear, hybrid and unmanned systems, particularly in low-complexity observers-based controllers.

Gabriel Alexis GUIJARRO REYES redeived his B.Sc. in Mechatronics Engineering from La Salle Laguna University in 2018. In 2020 he obtained his M. Sc. in Computer Science from Texas A&M University - Corpus Christi. At present, he is pursuing a PhD in Electrical Engineering at New Mexico State University where he continues his research in applied control strategies for UAS.

Luis Rodolfo GARCIA CARRILLO received the Ph.D. degree in Control Systems from the University of Technology of Compiègne, France. He was a Postdoctoral Researcher at the Center of Control, Dynamical systems and Computation at UC Santa Barbara, USA. At present, he serves as a faculty member with the Klipsch School of Electrical and Computer Engineering at New Mexico State University, USA. His research interests include control systems, multi-agent systems, UAS, game theory, and the use of vision in feedback control.

João P. HESPANHA received the Licenciatura in electrical and computer engineering from the Instituto Superior Tecnico, Lisbon, Portugal in 1991 and the Ph.D. degree in electrical engineering and applied science from Yale University, New Haven, Connecticut in 1998. From 1999 to 2001, he was Assistant Professor at the University of Southern California, Los Angeles. He moved to the University of California, Santa Barbara in 2002, where he currently holds a Professor position with the Department of Electrical and Computer Engineering. His current research interests include hybrid and switched systems; multiagent control systems; game theory; optimization; distributed control over communication networks (also known as networked control systems); the use of vision in feedback control; stochastic modeling in biology; and network security

Laurent BURLION obtained his Ph.D. from the University of Paris Saclay in 2007. He joined the Rutgers University Department of Mechanical and Aerospace Engineering as an assistant professor in 2019, after serving as a scientific researcher at the French Aerospace Lab. His research is in the area of nonlinear control design to uncertain systems and is motivated by aerospace applications including spacecraft and aircraft. He is a member of the IFAC Technical Committee on Aerospace and of the IEEE-CSS Technical Committee on Aerospace Controls.