Translational Model Identification and Robust Control for the Parrot Mambo UAS Multicopter

Ignacio Rubio Scola, Gabriel Alexis Guijarro Reyes, Luis Rodolfo Garcia Carrillo, Joao Hespanha, and Junfei Xie

Abstract— **Two main problems are addressed in this paper. The first one is the model identification for a commercial unmanned aircraft system (UAS): the Parrot Mambo multicopter. The second one aims at synthesizing a robust controller for guaranteeing the stability of the X and Y translational dynamics of this UAS. To accomplish these goals, we first collect input-output data from a set of real-time flight experiments. Next, by applying an extended least square (ELS) algorithm to the data, a group of dynamic models are identified. Due to uncertainties, the obtained models are similar in nature but exhibit parametrical variations. For this reason, from the set of identified models, a unique nominal (i.e., average) parameter-dependent linear model is built, which also takes into account the minimum and maximum values defining the model parametrical variations. Finally, a static linear controller is synthesized for the dynamics of interest, guaranteeing global stability for every model. The identification results and the performance of the closed-loop controller are validated in a set of numerical simulations, demonstrating the effectiveness of the proposed modeling and control approaches.**

Fig. 1. The Mambo Drone. A commercially available mini UAS Platform developed by Parrot.

I. Introduction

Uncertainties are one of the main causes for poor performance and instability in feedback systems, therefore, robust stability is an important and nontrivial issue for any control design. Over the past decades, H_∞ control strategies for robust stability of linear and nonlinear systems have been extensively studied, and many interesting results have been introduced. In particular, some classic and recent works in this area can be found in [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 real-time systems, it is rare that the manufacturer of a system makes available the equations describing the corresponding dynamic model. This is the case with most, if not all of the Unmanned Aircraft System (UAS) platforms commercially available [5].

In this work, we focus on the development of an H_∞ robust control strategy for the Parrot Mambo UAS multicopter, see Figure 1. Not surprisingly, the equations describing the dynamics of such system are not provided by the manufacturer. Therefore, as a first step towards developing the control strategy, a model identification procedure is conducted. We start by applying an extended least-squares (ELS) identification technique to a set of data containing input-output system information. This kind of technique has been studied in [6], [7], and [8].

It is well known that any identification procedure of a real-time system is commonly affected by uncertainties and errors. Indeed, if we perform multiple instances of an identification procedure applied to the exact same system, we will ultimately end up with similar but slightly different results. For this reason, we decided to conduct multiple identification tests, collecting each time input-output data pairs. With this procedure, instead of obtaining just a single accurate model, multiple solutions are obtained. As mentioned before, each one of the identified models is somehow affected by uncertainties and errors. This is where the H_∞ robust control strategy comes into play. The proposed controller is designed in such a way that it is able to stabilize the real-time UAS platform, by taking into account the worst case parametric variations in the models obtained from the system

I. Rubio Scola is with CIFASIS (CONICET-UNR), Department of Mathematics, FCEIA-UNR, Rosario, Argentina and with the Unmanned Systems Laboratory. School of Engineering and Computer Sciences, Texas A&M University - Corpus Christi, Texas, USA. 78412-5797. ignacio.rubioscola@tamucc.edu

G.A. Guijarro Reyes and L.R. Garcia Carrillo are with the Unmanned Systems Laboratory. School of Engineering and Computer Sciences, Texas A&M University - Corpus Christi, Texas, USA. 78412-5797. Guijarro Reyes email: gguijarroreyes@islander.tamucc.edu, Garcia Carrillo email: luis.garcia@tamucc.edu

J. Hespanha is with the Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara, CA 93106, USA. hespanha@ucsb.edu

J. Xie is with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA 92182. jxie4@sdsu.edu

This work was supported by ARO under grant W911NF1810210 and by NSF under grant EPCN-1608880 and CI-1730589.

978-1-7281-0960-2/19/$31.00 ©2019 IEEE

identification procedure.

The rest of the manuscript is organized as follows. Section II describes the identification technique, which is based on a recursive ELS methodology, and is also extended for linear systems with output delays. The H_∞ robust control strategy is introduced in Section III. Next, Section IV presents the model identification procedure to identify the dynamics of the Mambo UAS test-bed. Section V complements the identification results with a robust controller for the UAS. Numerical simulations are presented in Section VI to demonstrate the performance of both the proposed identification and robust control techniques. Finally, concluding remarks and future research directions are presented in Section VII.

II. Identification

This section summarizes a recursive least square algorithm to identify a linear system affected by output additive noises. Next, we show an extension of such methodology for linear systems with time delays. Finally, we introduce relevant practical considerations for implementing the identification approach, which simplify the identification task under certain conditions.

A. Recursive Least-Squares

Following the methodology introduced in [6], [7], we summarize an ELS algorithm for identifying a Single Input, Single Output (SISO) linear time-invariant (LTI) discrete-time system. 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 is the time sample, u_k and z_k are scalar input and 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 are constant (with $i = 1, 2, \cdots, n$). Also, the output is corrupted by an additive Gaussian zero-mean white noise ν_k.

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

From equations (1) and (5), we obtain:

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

with $\epsilon_k = A_q \nu_k$.

Introducing the matrices:

$$\theta = [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n]^T \tag{7}$$
$$y = [y_{n+1}, y_{n+2}, \cdots, y_{N_{ls}}]^T \tag{8}$$
$$e = [e_{n+1}, e_{n+2}, \cdots, e_{N_{ls}}]^T \tag{9}$$
$$\phi_i = [-y_{i-1}, -y_{i-2}, \cdots, -y_{i-n}, u_{i-1}, u_{i-2}, \cdots, u_{i-n}]^T \tag{10}$$
$$\Phi = [\phi_{n+1}, \phi_{n+2}, \cdots, \phi_{N_{ls}}]^T \tag{11}$$

where the superscript T mean the matrix transpose and N_{ls} the number of measurements available, it is possible to obtain the following equation

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

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

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

In order to eliminate the bias induced by the noise in $\hat{\theta}_{OLS}$, ϵ_k is modeled as:

$$C_q \epsilon_k = e_k \tag{14}$$

with $C_q = 1 + c_1 q^{-1} + \cdots + c_n q^{-n}$, and e_k is an independently distributed random sequence. The order of the constants terms c_i is m.

Defining:

$$\Pi = [c_1, c_2, \cdots, c_m]^T \tag{15}$$
$$e = [e_{n+1}, e_{n+2}, \cdots, e_{N_{ls}}]^T \tag{16}$$
$$w_i = [-\epsilon_{i-1}, -\epsilon_{i-2}, \cdots, -\epsilon_{i-m}]^T \tag{17}$$
$$\Omega = [w_{n+1}, w_{n+2}, \cdots, w_{N_{ls}}]^T \tag{18}$$

the term ϵ_k can now be estimated by:

$$\hat{\epsilon}_k = \hat{A}_q y_k - \hat{B}_q u_k \tag{19}$$

with $\hat{A}$ and $\hat{B}$ the respective estimation of A and B. Combining the estimation problems for matrices θ and Π, we arrive at a nonlinear problem:

$$y = [\phi\ \Omega] \begin{bmatrix} \theta \\ \Pi \end{bmatrix} + e \tag{20}$$

whose solution is provided as follows:

$$\hat{\theta} = \hat{\theta}_{OLS} - \hat{\theta}_{\text{Bias}} \tag{21}$$
$$\hat{\theta}_{\text{Bias}} = (\Phi^T\Phi)^{-1}\Phi^T\Omega\hat{\Pi} \tag{22}$$
$$\hat{\Pi} = [\Omega^T\Omega]^{-1}\Omega^T\epsilon \tag{23}$$

This problem is nonlinear and must be solved iteratively. In particular, at each iteration, $\hat{\theta}_{\text{OLS}}$ is first found using equation (13), $\hat{\epsilon}$ is then computed using equation (19), and finally $\hat{\theta}$ is calculated using equations (21)-(23). This procedure is repeated until the norm of the difference of two $\hat{\theta}$ corresponding to two consecutive iterations is small enough.

In the following analysis, we extend the identification procedure to address hardware-induced time delays that affect the measurements.

B. Dead-Time Identification

A system with an output time-delay of τ samples can be represented as

$$y_k = z_{k-\tau} + \nu_k \tag{24}$$

Therefore, in order to obtain meaningful modeling and control results, the value of τ must be identified first. We now recall a solution for this time-delay estimation problem,

which was originally proposed in [9]. This technique combines the ELS and an optimization procedure. In particular, given the expected τ_{min} and τ_{max}, the ELS algorithm is first applied in parallel to a set of $(\tau_{\text{max}} - \tau_{\text{min}} + 1)$ models, all of which have the same A_q and B_q polynomial orders, but different dead times given as follows

$$\tau_i = \tau_{\text{min}} + i, \quad i = 0, 1, 2, ..., (\tau_{\text{max}} - \tau_{\text{min}}). \tag{25}$$

Then, a performance index is computed for each model by comparing its output with the ground truth obtained in the real process using

$$I_{\text{pi}} = \frac{1}{\tau_{\text{max}} - \tau_{\text{min}}} \sum_{t=1}^{t=\tau_{\text{max}}-\tau_{\text{min}}} \|y - \hat{y}_i\|^2, \tag{26}$$

where $\hat{y}_i$ is the output of the i-th model. Finally, the best model is chosen as the one which gives the lowest I_{pi}. The dead time of the best model is then considered as the best estimation of the dead time.

C. Practical Considerations

Below we recall some practical considerations from [10], which are important to achieve a better identification.

- **Signal scaling:** The least-squares are, generally, numerically well conditioned when the numerical values of both the inputs and the outputs have roughly the same order of magnitude.
- **Down-Sampling:** If the data acquisition system can sample the system at higher frequencies than the identification needs, then a down-sample can be applied to the signals, which actually helps remove 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) as

$$\frac{MSO}{MSE} = \frac{\|\phi\hat{\theta} - y\|^2}{\|y\|^2} \tag{27}$$

Remark 1: The first objective of our research concerns with the system identification of the Parrot Mambo UAS. Executing several identification experiments for this platform, a collection of linear models with the same structure but different parameters are obtained. Our next objective is then to express all those models as a single model with variant parameters.

The following section explains the synthesis for a robust controller for a class of linear time variant (LTV) discrete systems.

III. Robust Control

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

$$\begin{aligned} x_{k+1} =& A(\xi_k)x_k + B(\xi_k)u_k + E(\xi_k)d_k \\ z_k =& C(\xi_k)x_k + D(\xi_k)u_k + F(\xi_k)d_k \end{aligned} \tag{28}$$

where the state $x \in \mathbb{R}^n$, the perturbation $d \in \mathbb{R}^p$ and 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

$$\Xi = \left\{ \xi \in \mathbb{R}_+^N : \sum_{i=1}^{N} \xi_i = 1 \right\} \tag{29}$$

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

$$\begin{bmatrix} A(\xi_k) & B(\xi_k) & C(\xi_k) \\ D(\xi_k) & E(\xi_k) & F(\xi_k) \end{bmatrix} = \sum_{i=1}^{N} \xi_{i,k} \begin{bmatrix} A_i & B_i & C_i \\ D_i & E_i & F_i \end{bmatrix} \tag{30}$$

where N is the number of vertices and the subscript $_i$ denote each of the vertices. For the purpose of defining the H_∞ cost criterion, consider the asymptotically stable open-loop dynamics of the form:

$$x_{k+1} = A_k x_k + E_k d_k \tag{31}$$

$$y_k = C_k x_k + F_k d_k \tag{32}$$

for which the H_∞ performance is defined by the l_2-to-l_2 gain:

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

The following lemma provides a linear condition for obtaining a static controller, ensuring a predefined H_∞ performance (μ).

Lemma 1 (from [2]): The system in equation (28) is poly-quadratically stabilizable with H_∞ performance bound μ, if and only if there exist $Q_i = Q_i^T \succ 0$ and X, L such that

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

where the $\star$ is used for the symmetric transpose, $\succ$ for the positive definite condition, and for all $i, j = 1, ..., N$. The robust control law is given by $u_k = K x_k$, with $K = L X^{-1}$.

It is possible to minimize μ and obtain a linear optimization problem that finds a static linear controller with minimal H_∞ performance bound. We next show the numerical solution of the identification problem for the Parrot Mambo Multicopter UAS.

IV. Application: System Identification for the Parrot Mambo Multicopter UAS

The Parrot Mambo multicopter UAS was first launched by Parrot in 2017, as part of the Parrot's MiniDrones

family [1]. When equipped with the protective casing, the UAS dimensions are 180mm x 180mm x 40mm, with an average weight of 70 grams. A group of on-board sensors are used to stabilize the UAS, specifically, a downwards-looking camera, an ultrasonic sensor, a barometer, and an Inertial Measurement Unit (IMU). The vehicle is powered by a 3.7V 550 mAh 2Wh Li-Po battery, enabling a flight time of 7-10 min, depending on the nature of the task to be performed and the accessories implemented. The maximum flight speed is 8 meters/sec. The communication with the vehicle is established by means of two different protocols, BLE and WiFi. The first protocol is used by default, and the second one is applied when using the First Person View (FPV) flight accessories.

For measuring ground truth when performing the identification procedure, we made use of a 10-camera Motion Capture System from Vicon. This system is cable of providing the position and orientation of the robot at rates in excess of 500 measurements per second, with a 0.1mm precision. This system essentially provides fast and reliable *indoor GPS* for testing and validation of modeling and control solutions.

A. Design of the Identification Experiment

The real-time experiments were designed with the objective of obtaining sufficient information from the UAS, in such a way it is possible to guarantee the convergence of the identification algorithm. In these tests, the UAS was flown in a squared trajectory, at 1m above the floor (i.e., the X-Y plane) of the laboratory flight arena, see Figure 2. While the system was flying, the Motion Capture System and a ground control station were used for collecting input-output information for the motions in the X and Y inertial coordinates, at a rate of 100Hz. An illustration of the collected squared motion of the UAS, as well as the corresponding control signals are shown in Figures 3 and 4.

At this point, the practical considerations in Section II-C become handy. Knowing that the model from velocity to position is described by a pure integration, we proceed to derive the position, and to identify the model from the input to the velocity. Since a simpler model is more desirable, we choose a first order model for the dynamics identification, in combination with a first order model for the noise identification. The corresponding model is then

$$y_k = -ay_{k-1} + bu_{k-1} + ce_{k-1} + e_k \tag{35}$$

For each experiment, we identified 4 parameters: a, b, c, and the output time delay τ. After analyzing various experiments, we concluded that the value of $\tau = 17$ minimizes the MSE/MSO ratio. Figure 5 illustrates the real data validation of the first order model for the velocity dynamics (Δx) in the X coordinate. A similar plot and result were obtained for the Y dynamics. In theory, the UAS multicopter platform is symmetric. Therefore, the X and Y dynamics should be very similar. For this reason, we decided to combine the parameters obtained in both X and Y directions, and use

[1] www.parrot.com/global/drones/parrot-mambo-fpv

Fig. 2. For system identification purposes, the UAS was flown in a squared trajectory, at 1m above the floor (i.e., the X-Y plane) of the lab flight arena. A Motion Capture System and a ground control station were used for collecting input-output information for the UAS motions in the X and Y directions, at a rate of 100Hz.

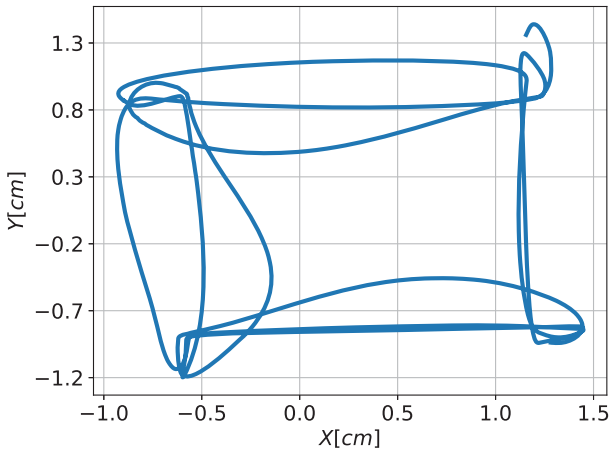

Fig. 3. Experiment used to extract the data for model identification. The UAS was flown in a squared trajectory, at 1m above the floor (i.e., the X-Y plane) of the laboratory flight arena.

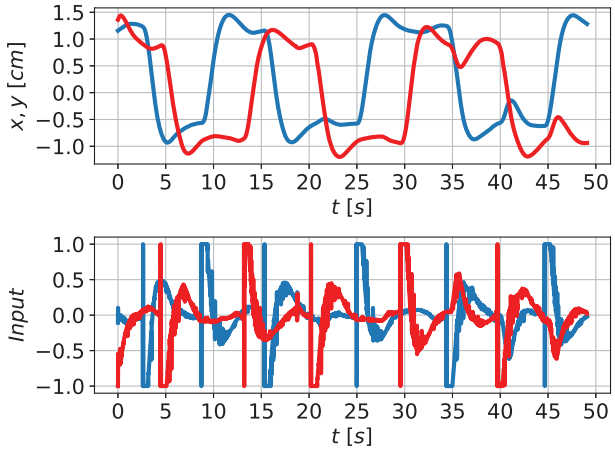

Fig. 4. An example of the X-Y dynamics during the experiment. The upper plot shows the translational motion, while the lower plot shows the corresponding control signals. In the plots, the blue and red colors are associated with the X, and Y dynamics, respectively.

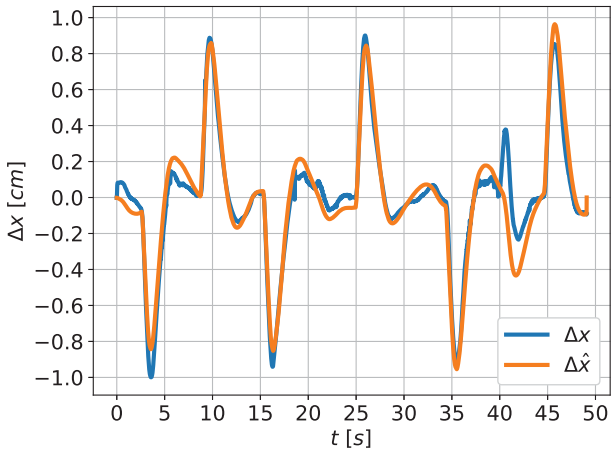

Fig. 5. Identification of the UAS dynamics in X. Real-data validation of the first order nominal model for the X coordinate.

them as a single larger set. As an illustration of the mixing procedure, the overall a and b parameters obtained from the experiments are show in Figure 6. 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 crosses are the vertices associated with the combination of the maximum and minimum values for a and b parameters. Also, the black cross represents the mean (i.e., the average) model, which represents the nominal model. The nominal, minimum, and maximum values will be used hereafter in equation (30).

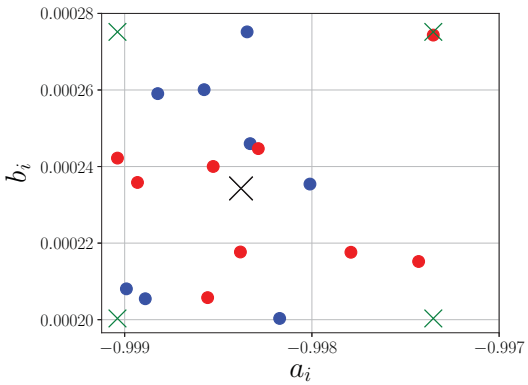

Fig. 6. Model parameters for the translational model.

V. Controller Synthesis

Now that the nominal model and associated variations are identified, we can proceed to synthesize the controller. First, we write the model as in equation (30). Using the knowledge that the system is composed by an integrator plus a first order system, it is possible to rewrite the model as a state space model. Defining the state vector $\bar{x}_k = [x_k, \Delta x_k]$, the input u_k, and the output y, we obtain

$$\bar{x}_{k+1} = A\bar{x}_k + Bu_k, \quad y = C\bar{x}_k + Du_k \tag{36}$$

$$A = \begin{bmatrix} 1 & 1 \\ 0 & -a_m \end{bmatrix}, B = \begin{bmatrix} 0 \\ b_m \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix}, D = 0 \tag{37}$$

where a_m and b_m are identified as the nominal model, and $a_{\min} < a_m < a_{\min}$ and $b_{\min} < b_m < b_{\min}$. Numerically:

$$\begin{aligned} a_m &= -0.99837 \\ a_{\max} &= -0.99735 \quad a_{\min} = -0.99903 \\ b_m &= 0.23428 \cdot 10^{-3} \\ b_{\max} &= 0.27517 \cdot 10^{-3} \quad b_{\min} = 0.20032 \cdot 10^{-3} \end{aligned} \tag{38}$$

Recall that the output of the UAS model is affected by a time delay τ, we propose to down-sample the model to a number equal to τ, such that the time delay is equal to the sampling time and hence does not affect the dynamics of the new model. This implies that the matrices of the down-sampled model, A_d and B_d, are computed as:

$$\begin{aligned} A_d &= A^\tau, \quad B_d = \sum_{j=0}^{\tau-1} A^j B \\ C_d &= C \quad \text{and } D_d = D \end{aligned} \tag{39}$$

To complete the model as in equation (28), we need the information of $E(\xi_k)$ and $F(\xi_k)$, which are used for tuning purposes. Here, we select:

$$E = [0\ 1]^T \text{ and } F = [1] \tag{40}$$

To minimize the energy spent by the controller, we replace the D matrix with $D_{\text{opt}} = 0.075$. Using equation (39), the tuning matrices E and F in equation (34), and the matrix D_{opt} instead of D, we then solve the optimization problem using the CVXPY [11] and MOSEK [12] libraries for Python language with SciPy [13]. The following controller gain matrix is obtained:

$$K = [-3.52718, -269.24932] \tag{41}$$

In the following section, we validate the performance of the proposed control strategy by means of numerical simulations.

VI. Simulations

The simulations were performed using Python language and the library SciPy [13]. We simulate in closed-loop, using the controller K obtained from Section III, and the models identified in Section IV. As mentioned in Section III, the controller is built for a down-sampled system, i.e., the model is sampled at 100Hz, but for each time sample multiple of τ (0.17s) the input is updated. Figure 7 shows the X coordinates of 9 models, when the UAS is following the same sequence of set-points with the same initial conditions given below:

$$\begin{aligned} x &= 0 \quad \forall t : t < 2 \\ x &= -1 \quad \forall t : 2 \leq t < 10 \\ x &= 0 \quad \forall t : 10 \leq t < 17 \\ x &= 4 \quad \forall t : 17 \leq t < 25 \\ x &= 0 \quad \forall t : t \geq 25 \end{aligned} \tag{42}$$

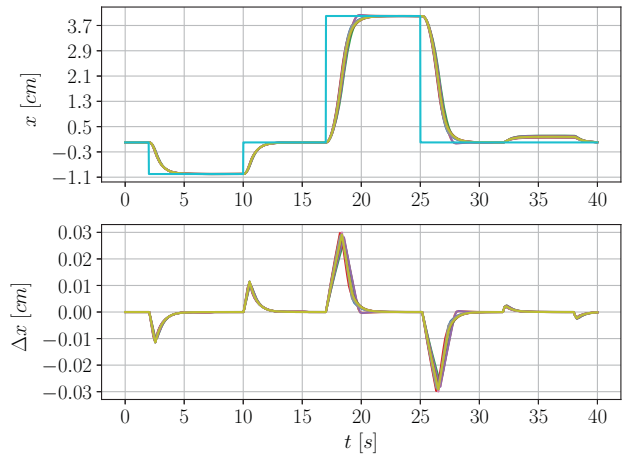

Fig. 7. Simulation results for position and velocity. The plot shows the closed-loop time response for X dynamics, for all models. The set-points are shown in cyan color.

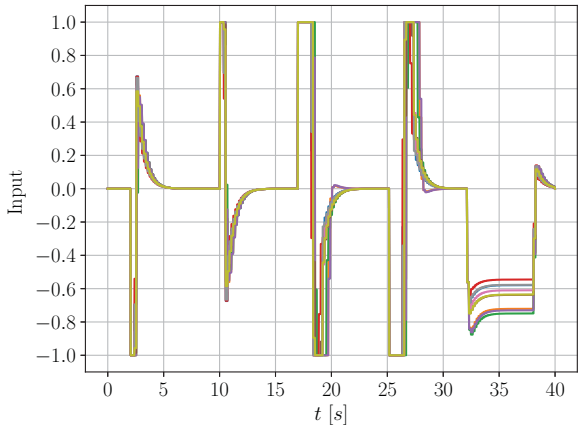

Fig. 8. Simulation results for position and velocity. The plot shows the closed-loop input signal for the X translational motion, for all models.

Figure 8 illustrates the input of each model over time. Notice that for each system model, the controller saturates the input and reaches each set-point in about 4 seconds. In addition, the saturation does not make the systems, which are marginally stable, unstable. To validate the robustness of the controller, we introduced a perturbation in the velocity, with a magnitude $p = 0.31 \cdot 10^{-4}$, between 32s and 38s. As soon as the perturbation is applied, the velocity is immediately stabilized at the origin, but it can be seen that an offset appears in the position. From Figure 8, we can see that when the perturbation is applied, each system has a different input to stabilize the velocity.

VII. Conclusions and Future Research

This paper presented a model identification procedure for a parameter-variant discrete-time linear system, and applied this methodology to a commercial UAS: the Parrot Mambo multicopter. An integrator plus a first order structure with output time-delays was chosen as the nominal model for the Mambo. Real-time input-output data-sets from multiple real-flight experiments were obtained from a Motion Capture system. Then, a collection of linear models with the same structure but different parameters were obtained. These models were expressed as a single model with variant parameters. A robust controller to guarantee the global stability for the translational dynamics of such system was also proposed, which was synthesized with a minimal H_∞ norm. Numerical simulations show the effectiveness of the proposed approach, against model uncertainties, delays, and disturbances.

A. Future Work

Future work will address the real-time implementation of the proposed controller using the Robot Operating System (ROS) environment. The difficulties associated with the real-time implementation will also be considered. Also, a perturbation estimation is envisioned, in order to feed-forward the perturbation information, and equip the controller with the capability of rejecting certain classes of perturbations.

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, 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", 1EEE 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] J.E. Normey-Rico and E.F. Camacho "Control of Dead-time Processes" in Advanced Textbooks in Control and Signal Processing, Springer, 2007

[10] J.P. Hespanha "Advanced Undergraduate Topics in Control Systems Design", 2019 `https://www.ece.ucsb.edu/~hespanha/published/allugtopics-20190508.pdf`

[11] 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

[12] MOSEK, "Optimizer API for Python 9.0.96", `docs.mosek.com/9.0/pythonapi/index.html`, 2019

[13] E. Jones, T. Oliphant, P. Peterson and others, "SciPy: Open source scientific tools for Python", `scipy.org`, 2019