Control of the DarkO Tail-Sitter Drone through an LMI-Based Static Output Feedback Design

Armand-Ioan Curpanaru, Fabrice Demourant, Florian Sansou

Control of the DarkO Tail-Sitter Drone through an LMI-Based Static Output Feedback Design

Armand-Ioan Curpanaru*[1,2] and Fabrice Demourant†[1] and Florian Sansou‡[3]
[1]ONERA/DTIS, Toulouse, France, [2]ISAE-SUPAERO, Toulouse, France
[3] ENAC, Université de Toulouse, F-31400 Toulouse, France

ABSTRACT

This paper deals with the control of a convertible drone through the synthesis of Static Output Feedback (SOF) model-based controllers. To that end, a convex optimization algorithm based on Lyapunov's stability theory and Linear Matrix Inequalities (LMI) available in the literature is employed. To evaluate the algorithm's performance, given that it has never been previously tested on realistic dynamical systems, it is implemented on the experimental model of the DarkO drone. Then, simulations of the closed-loop dynamic of the drone are carried out to assess the performance of the new design. For a preliminary experimental validation, the control architecture is implemented on the real drone system and test flights focused on the hovering phase and reference tracking in terms of position are carried out.

1 INTRODUCTION

1.1 Forewords and problem description

Despite numerous technological advances, today's drones are still limited by their energy consumption and lack of adaptation to complex urban environments. To reduce these weaknesses, a new generation of drones has been developed, namely the convertible drones that offer the capability of taking-off and landing vertically, while having a high aerodynamical efficiency in cruise.

The present article positions itself along a new direction of research aimed at developing control architectures for convertible UAVs (Unmanned Aerial Vehicles). The different designs and control techniques of hybrid and convertible VTOL (Vertical Take-Off and Landing) platforms are presented and discussed in [1]. The decision of employing a particular control scheme is mainly motivated by its performance in stabilizing and controlling the plant's dynamics across multiple phases such as take-off, hovering, transition, horizontal flight and landing.

Research supported in part by ONERA-ISAE-ENAC Research Federation
* armand-ioan.curpanaru@onera.fr, armand-ioan.curpanaru@isae-supaero.fr
† fabrice.demourantr@onera.fr
‡ florian.sansou@recherche.enac.fr

With respect to the controller synthesis process, model-based or model-free/data-driven methodologies are employed. Study [2] presents a detailed comparison between model-based and model-free control. Model-based methods [3, 4, 5] require sophisticated wind-tunnel characterization as well as accurate modeling of propeller-wing interactions, forces and moments of a partially stalled wing and of control surfaces that is highly challenging and time-consuming. As an alternative, model-free control [6, 7] does not need sophisticated models, as it is generally based on sensor measurements for estimating a large part of the dynamic model. As a main drawback, such techniques use test flight data to tune offline the control coefficients. There are also methods like INDI (Incremental Nonlinear Dynamic Inversion) [7] that integrate model-based and sensor-based methods, but are limited by the lack of proof of convergence.

The controller synthesis approach that has been chosen is a model-based one for obtaining SOF (Static Output Feedback) controllers. The majority of controllers implemented for this drone concept are data-driven ones. As a result, designing model-based controllers is a real challenge, having not been addressed exhaustively by the research community. Additionally, considering the intended operation of the DarkO drone in urban and populated areas, it is conceivable that future certification may become necessary. In such a scenario, a model-based control architecture significantly streamlines the certification process.

SOF represents the simplest control law architecture, since only static gains are implemented in the closed loop for generating command signals from measured states. In brief, the implementation is straightforward. Moreover, the fact of explicitly using measurements instead of state variables makes this type of control law extremely attractive from an engineer's point of view.

Beyond the apparent controller simplicity, the stabilizing SOF gain synthesis is a well-known *NP-hard* control problem [8] that cannot be solved by classical synthesis tools [9]. This *NP-hardness* feature comes from the BMI (Bilinear Matrix Inequality) problem formulation. Let us note that any fixed or low-order dynamic controller can be reparametrized as a SOF [10],[11], making this controller structure relatively versatile. More specifically, for a dynamic controller the problem is recast as SOF synthesis by considering the state space representaion of the controller as gain matrices to be optimized. Due to the *NP-hardness* of the SOF problem, it

is difficult to implement a systematic method that can either find a solution to the SOF problem or evaluate the feasibility problem. In the literature, the approaches are mainly based on either non-smooth optimization techniques for obtaining a stabilizing controller under structure constraints [12] or on Lyapunov's stability theory [10], where LMIs are iteratively solved, to treat the BMI original problem. The LMI-based approaches are of interest, since the same framework is used for robust control. Furthermore, the methods for designing a SOF controller can be easily extended to the uncertain and robust (R-SOF) case for a polytopic model or multimodel representation.

1.2 Outlines

The convertible drone design, having a low research maturity in the field of Automatic Control, presents a high degree of potential novelty that is exploited in this article. The main contributions are represented by the implementation of a mathematical formulation for model-based controller synthesis on a practical case study. Thus far, this approach has been tested and evaluated only on theoretical cases [13] with dynamic models that did not reflect real-world systems.

The proposed method for controlling the studied convertible drone model is based on synthesising a SOF controller, that is obtained by algorithmically converting the non-convex BMI problem to 3 auxiliary convex problems i) a State Feedback (SF) problem, ii) an Output Injection (OI) problem, and iii) a State Injection (SI) problem. Solving the SF, OI, and SI problems are necessary conditions for the existence of a solution for the SOF problem. Consequently, solving each of the above problems is addressed through a 3-phase algorithm presented in section 3. This algorithm is implemented on DarkO's augmented dynamics which are incorporated into the open loop to enforce a desired shape, akin to the principles of the loop shaping method [14]. This drives to multiple guarantees, namely in performance, with open-loop high gain at low frequency, and in robustness, with open-loop weak gain strategy at high frequency while also in satisfying the reference tracking requirement.

The resulting SOF controllers for the augmented dynamics will be validated through time-domain simulations. Moreover, the majority of research articles that introduce control system solutions for convertible drones are limited to simulation-only validation, whereas in this article, the validation will also be conducted experimentally through real flight tests with an experimental model of the DarkO drone.

The article is organized as follows. The general methodology is introduced in section I. In section II, the non-linear and linear dynamics modeling is presented. Section III, details the deterministic iterative algorithm for SOF design that will be implemented, as well as the plant's augmentation process. In section IV, the results of the controller synthesis are analysed through time-domain as well as experimental simulations, by implementing the designed control laws directly on the real system.

1.3 Notations

$He\{X\} = X + X^T$, $(.)^T$ is the transpose of a matrix while the identity matrix and the null matrix of dimension p read, respectively, I_p and 0_p, $(X)^\circ$ is the full rank matrix s.t. $X^\circ X = 0$, $(X)^\perp$ is the full rank matrix s.t. $XX^\perp = 0$. $\mathcal{B}$ and $\mathcal{I}$ are the body and inertial reference frames, respectively, while x_b, y_b, z_b and x_0, y_0, z_0 are their associated axis systems.

2 MODELING OF DARKO DYNAMICS

The unmanned VTOL platforms are mainly split into tailsitter-, tiltrotor-, and tiltwing-VTOL aircraft. The model that will be used as a study case is the DarkO drone, a micro convertible tail-sitter flying wing, developed at ENAC Toulouse Fig.1. It is fitted with two propellers (shown in **black**) that generate thrust on the longitudinal axis and moments around the yaw axis, together with two control surfaces, namely elevons (shown in blue) that can be operated independently for generating moments around the pitch and roll axis. The dynamics have been modelled in previous research studies [15] and flight tests have been conducted [5] using the INDI method.

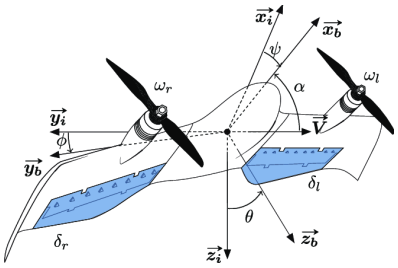

Figure 1: DarkO drone

It has to be stated that the dynamics on the DarkO drone are highly nonlinear. This is due to the aerodynamic forces and moments that vary with the square of the airspeed v^2, as well as couplings between actuators, notably the yawing moment generated by an elevon deflection, that varies with the airspeed of the flow over the control surface which in turn depends on the angular speed of the propeller. Additional nonlinearities emerge from the nonlinear dynamic of the quaternion q as well as products between the states (for example $w_{bx}w_{by}$, $w_{by}w_{bz}$) or higher order states (such as w_{bx}^2). There are even nonlinearities that are not taken into account due to their intrinsic modeling complexity. An example would be the stall phenomenon that appears over a fraction of the wing area during the hovering phase.

The nonlinear model of the DarkO drone is represented by the following equations:

$$\begin{cases} \dot{p} = v \\ \dot{v} = \frac{1}{m} R(q) \sum F_b(x,u,w) + g \\ \dot{q} = \frac{1}{2} q \otimes \omega_b \\ \dot{\omega}_b = J^{-1} \sum M_b(x,u,w) - J^{-1} [\omega_b]_\times J \omega_b \end{cases} \tag{1}$$

where $x = \begin{pmatrix} p & v & q & \omega_b \end{pmatrix}^\mathsf{T}$ - state vector, $u = \begin{pmatrix} \omega_1 & \omega_2 & \delta_1 & \delta_2 \end{pmatrix}^\mathsf{T}$ - command vector, $p \in \mathbb{R}^3$ - position of the centre of mass in the inertial frame, $v \in \mathbb{R}^3$ - velocity, w - external wind perturbation, $g = \begin{pmatrix} 0 & 0 & g \end{pmatrix}^\mathsf{T}$ - (constant) gravity vector, q - unit quaternion characterizing the attitude, ω_b - rotational speed in the body frame, F_b and M_b - forces and moments matrices, m - mass and J - inertia matrix, R - rotation matrix, $p \otimes q$ - quaternion product, $[a]_\times b = a \times b$ - cross product conversion to matrix multiplication.

Quaternions are justified by the tilt-body nature of the UAV that calls for a global numerically stable formulation of attitude. The linearization is conducted around an equilibrium point that corresponds to a particular wind speed, obtaining a discrete set of LTI models. The equilibrium point for the trimming that corresponds to a wind speed equal to 0 is defined by the following system:

$$\begin{cases} \overline{p} = p_e \\ \overline{v} = 0 \\ \overline{q} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \end{pmatrix}^T \\ \overline{\omega}_b = 0 \\ \overline{u} = \frac{mg}{(1 - \frac{S_{wet}}{4S_p} C_{d0})} \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}^T \end{cases} \tag{2}$$

The linearized dynamic is characterized by the state vectors in (3) and by the state-space representation in (4). Note that $B = \begin{bmatrix} B_u & B_w \end{bmatrix}$ where B_u and B_w are the actuator and wind input matrices. For more details related to the nonlinear, linear dynamic modeling and equilibrium points of the DarkO drone, the reader is directed to the articles [5, 15].

$$\tilde{x} = \begin{pmatrix} \tilde{p} & v & \tilde{\epsilon} & \omega_b \end{pmatrix}^T = \begin{pmatrix} p - p_{eq} & v & \epsilon - \epsilon_{eq} & \omega_b \end{pmatrix}^T$$
$$\tilde{u} = u - u_{eq} \tag{3}$$

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}$$
$$y = C\tilde{x} \tag{4}$$

3 SOF CONTROLLER SYNTHESIS

The synthesis of a SOF controller poses a non-convex problem due to multiplications between decision variables, resulting in bilinear matrix inequalities. This complexity renders the optimization problem *NP-hard*. Referring to the BMI formulation in [16], various equivalent reformulations are conducted in [13], leading to the matrix inequality presented in Eq. 5. The proposed method here is extracted from [16]. For further insights into the methods employed, such as the S-variable approach and dual calculations, readers are directed to [16].

$$P \succ 0$$
$$He\left\{ \begin{bmatrix} 0 & 0 & P \\ 0 & 0 & 0 \\ P & 0 & 0 \end{bmatrix} \right\} \prec He\left\{ \begin{bmatrix} I \\ C \\ 0_{p-n,n} \\ -A \end{bmatrix} \left(\lambda \begin{bmatrix} I \\ C \\ 0_{p-n,n} \end{bmatrix} + M \right) \right] S_1 + \begin{bmatrix} 0 \\ S_2 \\ BZ \end{bmatrix} \begin{bmatrix} 0 & I & -H^T \end{bmatrix} \right\} \tag{5}$$

$$F = -ZS_2{}^{-1} \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix} \tag{6}$$

Even though, this new reformulation is still characterized by non-convexity, it is of great interest because it has been proved that if a solution is found for the system in (5) when $\lambda = 1$, $M = 0$ and S_2 is non-singular then, a SOF gain matrix **F** can be computed, that guarantees closed-loop stability [13]. This gain matrix, not being directly optimized in the convex synthesis, is obtained using (6).

This proof serves as the overarching objective for the optimization process. Addressing the challenges stemming from the non-convex nature of problem 5 requires specific mathematical and technical advancements. A mathematical formulation introduced in the literature has shown promising initial theoretical results. The algorithm, to be detailed in the following subsection, will be applied to the DarkO drone's dynamics. The resulting analysis will draw conclusions regarding the algorithm's effectiveness in practical test scenarios.

3.1 Deterministic Iterative Algorithm for SOF Design

The problem of synthesising a SOF gain, is characterized by its intrinsic non-convexity nature that is accompanied by computation difficulties related to *NP-hard* problems. This, coupled with an inherent complexity that comes from the impossibility of optimizing all of the decision variables simultaneously, makes the process of splitting the non-convex problem into multiple auxiliary convex subproblems extremely appealing. The bilinear matrix inequality (5) will be employed to translate these goals into an optimization framework. Each particular goal is attained by optimally fixing a part of the decision variables, while refining the rest through an iterative process. For designing a SOF controller, a 3-phase algorithm introduced in [13] will be implemented Fig. 2 that takes as input the **A**, **B**, **C** matrices of the plant's linearized dynamics.

Figure 2: Structure of Optimization Algorithm

3.1.1 Initialization Phase

The objective of the first phase is essentially to provide an initial estimate for a state feedback gain matrix controller **K** that stabilizes the closed loop $\dot{x} = (A + BK)x$. For this, the

strategy taken is to fix the decision variables λ, M, H present in (5) in the following way:

$$\begin{aligned} \lambda &= \lambda = 0 \\ M &= M_0 = \begin{pmatrix} C^{\circ}C \\ C^{\perp T} \end{pmatrix} \\ H &= H_0 = JM_0^{-1} \\ J &= (-\mu - h)I, \end{aligned} \tag{7}$$

where, h is a positive scalar and $-\mu$ is the maximum real part of the eigenvalues of A. The feasibility study that is conducted in the initialization phase is the following:

$$P \succ 0$$
$$He\left\{\begin{bmatrix} 0 & 0 & P \\ 0 & 0 & 0 \\ P & 0 & 0 \end{bmatrix}\right\} \prec He\left\{\begin{bmatrix} I \\ -M_0 \\ -A \end{bmatrix} S_1 + \begin{bmatrix} 0 \\ S_2 \\ BZ \end{bmatrix} \begin{bmatrix} 0 & I & -H_0^T \end{bmatrix}\right\} \tag{8}$$

If (8) is feasible for a combination of the decision variables S_1, S_2, Z and for the Lyapunov certificate P, then an initial guess of a stabilizing state feedback gain **K** is found, and one can pass to the iteration phase. On the other hand, if a solution is not found, the parameter **h** is increased by 1 and the initialization phase is executed again. This phase solves the (SI) and (SF) problems. The outputs of this phase are:

$$\begin{aligned} S_{1.0} &= S_1 \\ \hat{K}_0 &= -ZS_2^{-1} \\ K &= -ZS_2^{-1}M_0 \end{aligned} \tag{9}$$

3.1.2 Iteration Phase - Step k,1

This phase comprises two iterative steps. In the first step, the optimization variables $\boldsymbol{\lambda}$ and **M** in 5 previously fixed during initialization, are now treated as decision variables. To maintain convexity, the slack variable S_1 is set at the value $\boldsymbol{S_{1,0}}$ determined in the initialization phase. Then, system (10) is solved.

$$P \succ 0, \quad \begin{bmatrix} (1-\lambda)I & M^T \\ M & I \end{bmatrix} \geq 0, \quad \lambda \geq 0$$
$$He\left\{\begin{bmatrix} 0 & 0 & P \\ 0 & 0 & 0 \\ P & 0 & 0 \end{bmatrix}\right\} \prec He\left\{\begin{bmatrix} I \\ -\left(\lambda \begin{bmatrix} C \\ 0_{p-n,n} \end{bmatrix} + M\right) \\ -A \end{bmatrix} S_{1,k-1} + \begin{bmatrix} 0 \\ -I \\ B\hat{K}_{k-1} \end{bmatrix} \begin{bmatrix} 0 & -S_2 & Y^T \end{bmatrix}\right\} \tag{10}$$

In (10), two matrix inequalities appear, in addition to the Lyapunov certificate inequality and the inequality (5). These additional constraints, contribute towards achieving the objective of the iteration phase which is to find a solution for (5) where the variables $\boldsymbol{\lambda}$ and **M** converge to the values of 1 and 0, respectively, this being a mandatory requirement for meeting the main optimization goal. This is accomplished by setting the maximization of $\boldsymbol{\lambda}$ as an objective in the optimization solver. The second matrix inequality works on constraining the norm of **M** and decreasing it to 0, as $\boldsymbol{\lambda}$ increases to 1. The third inequality is merely a constraint on $\boldsymbol{\lambda}$ to be positive.

If a solution is found for (10) and if for this solution $1-\lambda$ is lower than a specific tolerance, this means that one can pass on to the third and last phase, namely the validation. The outputs for this phase are:

$$\begin{aligned} \lambda_k &= \lambda \\ M_k &= M \\ H_k^T &= {S_2}^{-1}Y^T \end{aligned} \tag{11}$$

If not, there is a second step in the iteration phase.

3.1.3 Iteration Phase - Step k,2

Step **k,2** is essentially an initialization step in disguise. Similar to the first phase, where λ, M and H were fixed as inputs to generate a stabilizing feedback gain **K**, the second step fixes the same decision variables with updated values from step **k,1**.

The optimization problem that needs to be solved in this phase is the following:

$$P \succ 0$$
$$He\left\{\begin{bmatrix} 0 & 0 & P \\ 0 & 0 & 0 \\ P & 0 & 0 \end{bmatrix}\right\} \prec He\left\{\begin{bmatrix} I \\ -\hat{M}(\alpha) \\ -A \end{bmatrix} S_1 + \begin{bmatrix} 0 \\ S_2 \\ BZ \end{bmatrix} \begin{bmatrix} 0 & I & -H_k^T \end{bmatrix}\right\} \tag{12}$$
$$\hat{M}(\alpha) = \left((1+\alpha(\lambda_k - 1)) \begin{bmatrix} C \\ 0_{p-n,n} \end{bmatrix} + \alpha M_k\right) \tag{13}$$

In this step, a new term, $\hat{M}(\alpha)$ appears, dependent on a new decision variable α. The goal is to minimize α using the bisection method, converting the non-convex problem into a convex one. Ideally α converges to 0, signaling the end of this iteration phase and progression to the final validation phase. If $\alpha \approx 0$ within a certain tolerance, (13) simplifies to (14) and as a result, the values of the decision variables for the solution that has been found for (12) that correspond to the step **k,2** are also values for which (5) is solved when particularized for $\lambda = 1$ and M = 0. Finding a solution for (5) with $\lambda = 1$ and M = 0, enables the computation of a stabilizing SOF controller.

$$\hat{M}(\alpha) = \left((1+\alpha(\lambda_k - 1)) \begin{bmatrix} C \\ 0_{p-n,n} \end{bmatrix} + \alpha M_k\right) \overset{\alpha \approx 0}{=} \begin{bmatrix} C \\ 0_{p-n,n} \end{bmatrix} \tag{14}$$

If at step **k,2** α is not smaller than a fixed tolerance, the algorithm will start a new iteration step at **k,1**. The outputs of the current phase are:

$$\begin{aligned} \alpha_k &= \alpha \\ \hat{K}_{k-1} &= -ZS_2^{-1} \\ S_{1,k} &= S_1 \end{aligned} \tag{15}$$

3.1.4 Validation Phase

In cases where the iteration stage provides solutions for which $\boldsymbol{\lambda}$ and **M** are exactly equal to 1 and 0, a validation phase is not required because a SOF controller can be computed

directly. In practice, the algorithm exits the iteration phase with solutions for $\boldsymbol{\lambda}$ and $\mathbf{M}$ close to 1 and 0, due to limitations of the optimization solvers and numerical accuracies. In these cases, the conditions for achieving the main optimization objective are not exactly met. As a result, a validation phase is necessary, where the problem to be solved is defined by (16) where the variables $\boldsymbol{\lambda}$ and $\mathbf{M}$ are set to 1 and 0 and $\mathbf{H}$ is set to H_k (obtained during the iteration phase). If a solution is found, the gain matrix $\mathbf{F}$ can be constructed with (6) that is guaranteed to stabilize the closed loop (A+B**F**C). This last phase, provides the solution to the (OI) and (OF) goals.

$$P \succ 0$$
$$He\left\{\begin{bmatrix} 0 & 0 & P \\ 0 & 0 & 0 \\ P & 0 & 0 \end{bmatrix}\right\} \prec He\left\{\begin{bmatrix} I \\ -\begin{bmatrix} C \\ 0_{p-n,n} \end{bmatrix} \\ -A \end{bmatrix} S_1 + \begin{bmatrix} 0 \\ S_2 \\ BZ \end{bmatrix} \begin{bmatrix} 0 & I & -H^T \end{bmatrix}\right\} \tag{16}$$

3.2 *Plant Augmentation and Control Architecture*

The control architecture for the DarkO drone is designed to stabilize it during hovering, both with and without external disturbances. For example, to counteract a headwind affecting linear velocity along the $x_{[b]}$ and $z_{[b]}$ axes and generating a moment about the $y_{[b]}$ axis, the control scheme uses the ailerons and propellers symmetrically to generate a compensatory moment and force. Integral action is employed to counteract the force along the $z_{[b]}$ axis, ensuring asymptotic convergence to the desired force using two integrators, one for the motors and one for the elevons. Other common control methods for this drone configuration include PID scheduling, decoupled speed and attitude controllers, and nonlinear controllers with command allocation. However, none of these approaches directly address wind effects.

The synthesis of a SOF controller for the augmented DarkO dynamics takes its inspiration from [10] where it is shown that a dynamic output compensator of order $q \leq n$ (n-plant order), can be converted to a static output feedback through state-space augmentation. In our case, dynamic terms such as integrators and filters are added to the plant dynamics, resulting in an augmented state-space for which a static feedback law comprised of a pre-compensator static gain matrix will be synthesised. The implemented architecture, presented in Fig. 3, is a reformulation of the control law presented in [17]. It is composed of the linearized DarkO drone dynamic block, the saturation block for the two propellers and the two elevons that introduces limitations for the angular speed and angular rate of the electrical motors as well as angular deflection and angular rate for the control surfaces. The introduction of the actuator dynamics is subject for future studies. The output selection matrix eliminates the measurement of the pitch angle state, $\boldsymbol{\theta}$.

For control design, a PI-like controller structure is employed to minimize static error and enhance rejection of external disturbances. This is done by adding pre-compensator filters at the input of the system, based on

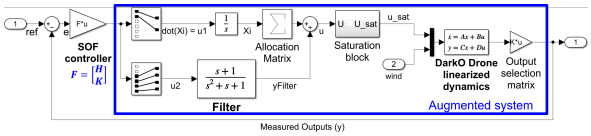

Figure 3: Plant Augmentation

the open-loop shaping methodology [14] where an integrator is added to ensure performance (high gain) at low frequency, which corresponds to the tracking specification, and a second-order filter is added to ensure that the controller is strictly proper and to be robust with respect to neglected dynamics thanks to a roll-off behaviour. The filter's cutoff frequency is set at approximately 5 Hz to align with closed-loop dynamics and reject white noise in measured signals. For limiting the number of integrators to 2 (one that generates the integrative command for the two propellers and the other one for the two elevons) the outputs of the integrator block are doubled by an allocation matrix $\sum$. Implementing a second order filter comes with the extra benefit of avoiding a direct feed-forward term, that may amplify unwanted sensor noise. The SOF matrix $\mathbf{F}$ comprises the gain matrices $\mathbf{H}$ and $\mathbf{K}$ for the integrative and proportional action, respectively. The equations and the state-space representation of the augmented plant are found in (17) and (18).

$$\begin{gathered} \dot{x}_i = u_1 \\ u = \sum x_i + y_{filter} = \sum x_i + Filter \cdot u_2 \\ \dot{x} = Ax + Bu = Ax + B(\sum x_i + y_{filter}) = Ax + B\sum x_i + BCx_{filter} \\ \sum = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = Fe = F\left(\begin{bmatrix} ref \\ \not{r}_{8\times 1} \end{bmatrix} - y\right) \end{gathered} \tag{17}$$

$$\begin{gathered} \begin{pmatrix} \dot{x} \\ \dot{x_c} \\ \dot{x}_{filter} \end{pmatrix} = \begin{bmatrix} A & B\sum & BC_{filter} \\ 0 & 0 & 0 \\ 0 & 0 & A_{filter} \end{bmatrix} \begin{pmatrix} x \\ x_c \\ x_{filter} \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & B_{filter} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ y = \begin{bmatrix} C & 0 & 0 \end{bmatrix} \begin{pmatrix} x \\ x_c \\ x_{filter} \end{pmatrix} \end{gathered} \tag{18}$$

Where $x_i \in \mathbb{R}^2$ - integrative states, $x_f \in \mathbb{R}^4$ - filter states, $r \in \mathbb{R}^3$ - reference signal, $y \in \mathbb{R}^{11}$ - measured states and $u \in \mathbb{R}^4$ - plant input.

4 RESULTS

4.1 *Simulation Results*

The approach from section 3.1 was applied to the augmented system from section 3.2, focusing on the linearized dynamics for a wind speed of 0 (hovering scenario). This resulted in four stabilizing controllers for h=12,13,14, and 15, with h varying between 1 and 40. Multiple iterations with different h values, serving as various initialization points for the optimization algorithm, are necessary to enhance the likelihood of convergence.

Figure 4 shows the closed-loop temporal response of the four controllers, with reference steps for the x-axis

at 5s, y-axis at 90s, and z-axis at 40s. All controllers effectively stabilize the closed-loop dynamics and track position references on the XYZ axes. The response is slow on the X and Z axes and fast on the Y axis. Notably, a position step on one axis does not cause significant offset or change on the other two axes, indicating strong decoupling between the XYZ position states. Additionally, the actuators were never saturated.

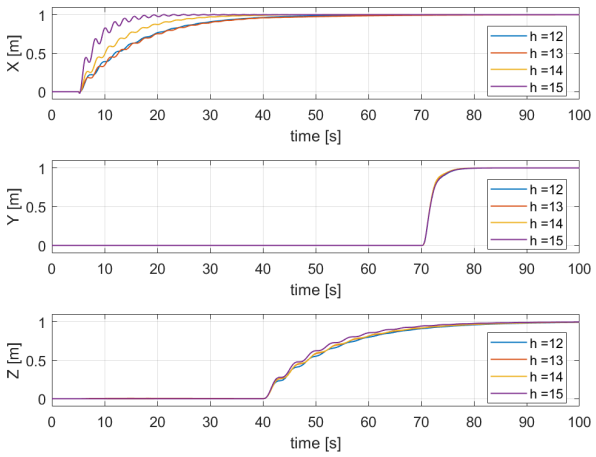

Figure 4: Closed-Loop response of SOF controllers - Linear Dynamics

The first subplot of Fig. 5 shows that the state $\boldsymbol{\theta}$ experiences increasing oscillations as h increases. While the pitch angle $\boldsymbol{\theta}$ is not directly controlled, it naturally converges to 0 as the other states stabilize, indicating hovering flight without wind. The second subplot demonstrates that for dynamics linearized around a specific wind speed, $\boldsymbol{\theta}$ converges to a non-zero value, compensating for external disturbances.

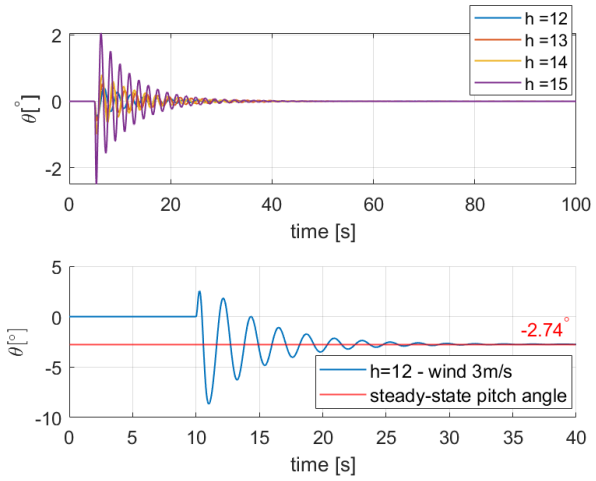

Figure 5: Closed-Loop temporal response - theta state

For analysis purposes, Eq. 19 and 20 present the gains of the synthesized controller **F** (block illustrated in 3) comprising the matrix gains **H** and **K** for h=12 and linearized dynamics at 0 wind speed. By taking a more attentive look at the gain matrix **K** that outputs 4 command signals used for the asymmetric or symmetric control of the drone actuators, one can observe that for the 1^{st} and 2^{nd} rows (that generate a demand of thrust of the two propellers) as well as for 3^{rd} and 4^{th} rows (that generate a demand for a deflection angle for the two elevons) are either nearly equal or opposite, with few exceptions. This pattern emerged naturally from the optimization process without any constraints, reflecting an intuitive and coherent alignment with the drone's dynamics.

$$H = \begin{bmatrix} 3.5e-06 & -3.0e-07 & 1.5e-02 & -6.9e-06 & -7.4e-06 & 4.4e-01 \\ -2.0e-02 & -7.5e-06 & 1.6e-05 & -2.0e-01 & 4.4e-05 & -1.1e-04 \\ 1.8e-04 & 1.1e-04 & -8.4e-05 & -8.6e-05 & -2.3e-04 \\ -9.2e-05 & -9.1e-05 & 7.4e-06 & 7.6e-01 & -3.9e-05 \end{bmatrix} \quad (19)$$

$$K = \begin{bmatrix} 7.6e-03 & -2.0e+01 & 9.9e+00 & 3.1e-02 & -4.5e+01 & 8.2e+01 \\ 5.4e-03 & 2.0e+01 & 9.9e+00 & 8.8e-03 & 4.5e+01 & 8.2e+01 \\ -4.2e+00 & -3.2e-01 & 8.7e-03 & -4.5e+01 & -4.4e-01 & -3.1e-02 \\ -4.2e+00 & 3.2e-01 & 4.8e-03 & -4.5e+01 & 4.3e-01 & 3.3e-02 \\ -3.1e+02 & -3.1e+02 & 6.4e-01 & 3.0e-03 & -1.0e+02 \\ 3.1e+02 & 3.1e+02 & -6.5e-01 & -1.9e-03 & 1.0e+02 \\ -1.2e+01 & 1.6e+00 & -3.1e+01 & 3.9e+01 & -1.6e+00 \\ 1.2e+01 & -1.7e+00 & 3.1e+01 & 3.9e+01 & 1.6e+00 \end{bmatrix} \quad (20)$$

4.2 *Experimental Results*

For experimental validation, the designed controllers and architecture were tested on the real DarkO drone model built at ENAC Fig. 6. DarkO is assembled from multiple 3D printed Onyx parts (a robust material comprising omnidirectional carbon fibres). Flight tests took place in ENAC's volière, equipped with an Optitrack motion capture system providing position and attitude data at 40Hz, eliminating the need for a GPS sensor. Speed is obtained by a finite difference between the position. Data fusion algorithms, including Invariant Filters [18, 19], combined data from Optitrack (position, speed and attitude) and the DarkO drone's IMU unit for improved state estimation. This estimate is used to create the output vector y used by the control law (17). Given the architecture of the control loop, the estimation must be of very high quality with the lowest possible delay. Paparazzi UAV open-source software and hardware packages were utilized [20] and the modular nature of Paparazi permit to use the existing implementation of the invariant filter to estimate the DarkO state vector and to add a stabilisation module based on the control law described above (see 3.2). This software is embedded in an autopilot designed and manufactured at Enac, an Apogee autopilot board[1] for processing. The autopilot board sampled command and control laws at 500Hz, generating appropriate control commands to achieve the desired flight manoeuvres and store all the data for posterior analysis.

The following experimental flight tests mark the first and only successful simulations in ENAC's volière where controllers relying exclusively on model-based synthesis methods were used to control a convertible drone. Prior

[1]Product data sheet - `https://wiki.paparazziuav.org/wiki/Apogee/v1.00`

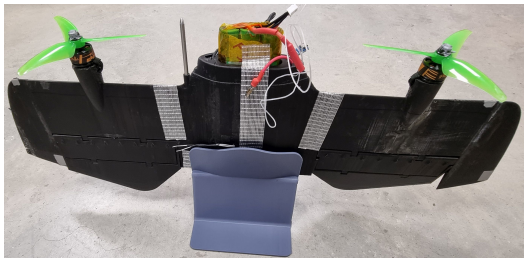

Figure 6: DarkO Drone experimental model

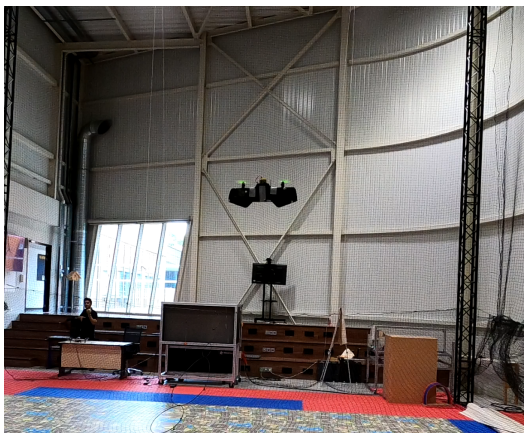

Figure 7: DarkO drone during an experimental flight test

flights for this drone type utilized combined model– and sensor–based control designs like INDI algorithms. Each flight starts with the drone taking off and stabilizing at a reference position using an INDI controller. After reaching the desired hovering location, the INDI controller is replaced with the model-based controller. Figure 8 shows the flight test results of one of the four synthesized controllers for the designed control architecture. Figure 7 depicts the DarkO drone during its test flight. All four controllers were tested and validated on the real system, demonstrating stable performance and accurate position tracking on all three axes.

As it was also observed in the simulations, on the X and Z axes, the time response is relatively high whereas on the Y axis, the time response is drastically quicker, due to the strong differential actuation on this axis backed up by the fast actuator dynamics. On the Z axis, the switch from the INDI controller to the synthesised controller, results in a relatively important vertical descent of the drone. As a consequence, further tuning and modeling needs to be performed in order to correct the state and command input initialization values corresponding to the equilibrium point. The dynamic of the drone on Z is characterized by small oscillations around the reference position, that can also be observed in the command signal of the angular speeds of the propellers, illustrated in Fig. 8. These reduced oscillations are due to the actuator dynamics not being considered in the control law. This problematic will be studied in the

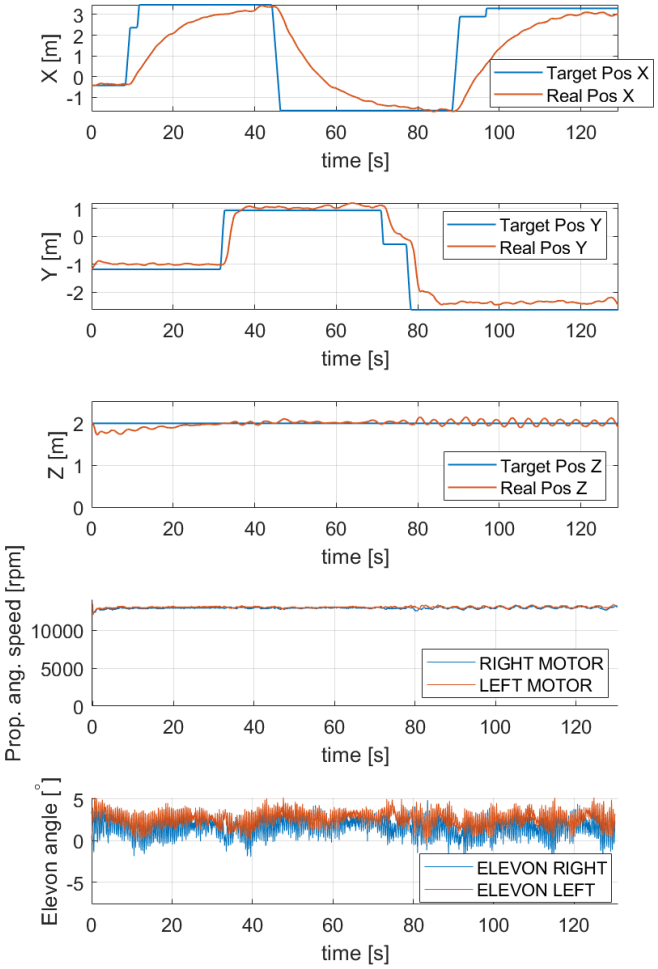

Figure 8: Flight Test, Closed-Loop Response and Command Input, h = 12

future, particularly with the use of a motor speed controller (see AM32-MultiRotor-ESC-firmware [2]). It's worth noting that the optimization algorithm's primary objective is to achieve closed-loop stabilization, disregarding performance or robustness criteria. During experimental flights, the controller did not saturate the actuators (see Fig. 8).

5 CONCLUSIONS

A convex optimization algorithm, utilizing the LMI framework and Lyapunov's stability theory, was employed to synthesize SOF controllers for the DarkO convertible drone model. This model-based synthesis technique effectively stabilized the closed-loop dynamics of the augmented plant, ensuring satisfactory temporal response and reference tracking without actuator saturation. Despite incomplete modeling of nonlinear phenomena, the controllers demonstrated robustness during initial experimental

[2] https://github.com/FlorianSan/AM32-MultiRotor-ESC-firmware

demonstrations on the DarkO drone model. With the designed SOF controller matrices and command law structure, successful experimental flights were conducted for hovering and trajectory tracking. This outcome serves as a solid proof of concept for the developed control law in terms of performance and robustness.

REFERENCES

[1] Guillaume J.J. Ducard and Mike Allenspach. Review of designs and flight control techniques of hybrid and convertible vtol uavs. *Aerospace Science and Technology*, 118:107035, 2021.

[2] Jacson M. O. Barth, Jean-Philippe Condomines, Murat Bronz, Leandro R. Lustosa, Jean-Marc Moschetta, Cédric Join, and Michel Fliess. Fixed-wing uav with transitioning flight capabilities : Model-based or model-free control approach? a preliminary study. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1157–1164, 2018.

[3] Yijie Ke, Kangli Wang, and Ben M. Chen. Design and implementation of a hybrid uav with model-based flight capabilities. *IEEE/ASME Transactions on Mechatronics*, 23(3):1114–1125, 2018.

[4] Yijie Ke, Kangli Wang, Kehong Gong, Shupeng Lai, and Ben M. Chen. Model based robust forward transition control for tail-sitter hybrid unmanned aerial vehicles. In *2017 13th IEEE International Conference on Control and Automation*, pages 828–833, 2017.

[5] Florian Sansou and Luca Zaccarian. On local-global hysteresis-based hovering stabilization of the darko convertible uav. In *2022 European Control Conference (ECC)*, pages 40–45, 2022.

[6] Jacson M. O. Barth, Jean-Philippe Condomines, Jean-Marc Moschetta, Aurélien Cabarbaye, Cédric Join, and Michel Fliess. Full model-free control architecture for hybrid uavs. In *2019 American Control Conference (ACC)*, pages 71–78, 2019.

[7] Rasmus Steffensen, Agnes Steinert, and Ewoud J. J. Smeur. Nonlinear dynamic inversion with actuator dynamics: An incremental control perspective. *Journal of Guidance, Control, and Dynamics*, 46(4):709–717, apr 2023.

[8] O. Toker and H. Ozbay. On the np-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *Proceedings of 1995 American Control Conference - ACC'95*, volume 4, pages 2525–2526 vol.4, 1995.

[9] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. State-space solutions to standard h/sub 2/ and h/sub infinity / control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

[10] V.L. Syrmos, C.T. Abdallah, P. Dorato, and K. Grigoriadis. Static output feedback—a survey. *Automatica*, 33(2):125–137, 1997.

[11] B. Anderson, N. Bose, and E. Jury. Output feedback stabilization and related problems-solution via decision methods. *IEEE Transactions on Automatic Control*, 20(1):53–66, 1975.

[12] P. Apkarian and D. Noll. Nonsmooth h-infinity synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86, 2006.

[13] D. Arzelier, F. Dabbene, S. Formentin, D. Peaucelle, and L. Zaccarian. *Robust Static Output Feedback Design with Deterministic and Probabilistic Certificates*, pages 121–148. Springer International Publishing, Cham, 2018.

[14] D. McFarlane and K. Glover. A loop-shaping design procedure using h infinity synthesis. *IEEE Transactions on Automatic Control*, 37(6):759–769, 1992.

[15] Florian Sansou. Commande hybride d' un drone convertible pour des déplacements sous optimaux. *ArXiv*, abs/2203.15387, 2022.

[16] Yoshio Ebihara, Dimitri Peaucelle, and Denis Arzelier. *S-Variable Approach to LMI-Based Robust Control*. Springer, January 2015.

[17] Florian Sansou, Fabrice Demourant, Gautier Hattenberger, Thomas Loquen, and Luca Zaccarian. Open wind tunnel experiments of the darko tail-sitter longitudinal stabilization with constant wind. *IFAC-PapersOnLine*, 55(22):1–6, 2022. 22nd IFAC Symposium on Automatic Control in Aerospace ACA 2022.

[18] Jean-Philippe Condomines, Cédric Seren, and Gautier Hattenberger. Nonlinear state estimation using an invariant unscented Kalman filter. In *AIAA GNC 2013, AIAA Guidance, Navigation and Control Conference*, pages pp 1–15 ; ISBN : 978–1–62410–224–0, Boston, United States, August 2013. AIAA.

[19] Jean-Philippe Condomines, Cédric Seren, and Gautier Hattenberger. Pi-invariant unscented kalman filter for sensor fusion. In *53rd IEEE Conference on Decision and Control*, pages 1035–1040, 2014.

[20] Gautier Hattenberger, Murat Bronz, and Michel Gorraz. Using the Paparazzi UAV System for Scientific Research. In *IMAV 2014, International Micro Air Vehicle Conference and Competition 2014*, pages pp 247–252, Delft, Netherlands, August 2014.