Optimal Tuning of the Lateral-Dynamics Parameters for Aerial Vehicles with Bounded Lateral Force

Dariusz Horla, Mahmoud Hamandi, Wojciech Giernacki, Antonio Franchi

▶ **To cite this version:**

Dariusz Horla, Mahmoud Hamandi, Wojciech Giernacki, Antonio Franchi. Optimal Tuning of the Lateral-Dynamics Parameters for Aerial Vehicles with Bounded Lateral Force. 2020. hal-02970615v1

HAL Id: hal-02970615

https://univ-smb.hal.science/hal-02970615v1

Preprint submitted on 18 Oct 2020 (v1), last revised 2 Feb 2021 (v2)

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

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

Optimal Tuning of the Lateral-Dynamics Parameters for Aerial Vehicles with Bounded Lateral Force

Dariusz Horla[1*], Mahmoud Hamandi[2*], Wojciech Giernacki[1], Antonio Franchi[2,3]

***Abstract*—This letter shows for the first time why it is important and how to optimize the gains of a position controller on board of a fully-actuated aerial vehicle with bounded lateral force, via an auto-tuning approach. In such vehicles, most of the control authority is expressed along a principal thrust direction, while along the lateral directions smaller forces can be exploited to achieve full-pose tracking. The nonlinear and hard to model interplay between the constraint imposed on the lateral force and the gains of the position controller is overcome employing the OPTIM-tune calibration method. Several experimental tests, performed fully autonomously during flight, clearly show the practicability and benefits of the approach.**

***Index Terms*—Aerial Systems: Mechanics and Control, Aerial Systems: Applications, Motion Control.**

I. INTRODUCTION

UNMANNED Aerial Vehicles (UAV)s have been widely studied in the literature with applications varying between search and rescue [1], fire fighting [2] and, more recently, aerial physical interaction [3].

Most UAV applications rely on the use of *collinear/coplanar* platforms [1], [2], [4], such as quadrotors, hexarotors or octorotors, where all propellers are coplanar and provide thrust in a direction parallel to the platform's vertical axis. While the use of these platforms allowed the advancement of UAV research in the above mentioned fields, they lack the ability of applying lateral forces and thus need to tilt to move laterally.

Different designs from the literature overcome this limitation by adding additional propellers that can apply lateral forces. Romero et al. [5] add four propellers to a quadrotor along the major lateral directions to move sideways. Similarly, Albers et al. [6] add an extra propeller that produces thrust perpendicular to the four main propellers. Conversely, Ryll et al. [7] apply lateral motion by tilting each of the propellers of a quadrotor independently.

Recently, all these platforms that can apply lateral forces in body frame have been grouped in the abstract class of Bounded Lateral Force (BLF) UAVs, firstly introduced in [8]. Unlike the more popular quadrotors, these platforms can apply a lateral force in their body frame thanks to tilted propellers, similar to the example platform shown in Fig. 1. BLF platforms can move laterally without the need of tilting, they can tilt without the need of moving laterally, and can interact with the environment with multiple force directions while maintaining an independent desired orientation. In its simplest and more effective representation, among the ones introduced in [8], a BLF platform's feasible force set is modeled as a cylinder which radius, noted as r_{xy}, which represents the maximum allowable lateral force in any horizontal direction.

The geometric controller presented in [8] requires the inertial parameters of the platform, and tuned gains of the attitude and position controllers; similar controllers have been also proposed in the literature, such as the controller presented in [9]. The inertial parameters of the platform can be easily estimated from the platform geometry, and do not depend on the controller gains. Similarly, the attitude controller has a larger authority than the position controller, and thus can be tuned using standard tuning methods, independently of the chosen position controller. The position controller merits further explanation; where as can be seen from Fig. 2 the lateral force limits are coupled with the applied lift force. Since r_{xy} is an estimate of the actual maximum lateral side force allowed by the platform for the current lift, it is difficult to chose a priori the best value for this parameter.

Furthermore the choice of r_{xy} can substantially affect the platform's performance, where if chosen to be near zero, the platform acts as an underactuated one, with a strong coupling between the position and attitude dynamics. If r_{xy} is chosen to be large enough, the platform behaves as a fully-actuated one, and the position and attitude dynamics become decoupled; special consideration has to be given as not to exceed platform's physical limits at the applied lift.

In this paper, we aim to experimentally study the above-mentioned interplay between the chosen maximum lateral force and the position dynamics. To be able to systematically tune the controller gains for each chosen r_{xy} value, we use the recently introduced Optim-tune algorithm [10]. The method described in [10] performs auto-tuning of controllers, where the tuning is done in cycles, composed of stages where a particular gain is tuned, and the remaining held fixed [10]. The method finds the optimal gains that maximize a designed performance index, and as such can be scaled to tune any of the controller gains affecting the desired performance.

* The two authors contributed equally to the work presented here.

[1] Poznan University of Technology, Faculty of Control, Robotics and Electrical Engineering, Institute of Robotics and Machine Intelligence, ul. Piotrowo 3a, 60-965 Poznan, Poland {dariusz.horla,wojciech.giernacki}@put.poznan.pl

[2] Laboratory for Analysis and Architecture of Systems, CNRS, Robotics and Interactions Laboratory (RIS), 7, avenue du Colonel Roche BP 54200 31031 Toulouse cedex 4, France mahmoud.hamandi@laas.fr

[3] University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science, Robotics and Mechatronics lab, Carré 3609, P.O. Box 217, 7500AE Enschede, The Netherlands a.franchi@utwente.nl

This work was partially funded by the European Robotics Research Infrastructure Network H2020 INFRAIA TERRINet (2017-2020) under Grant Agreement no. 730994 during translational infrastructure access in TERRINet; by Poznan University of Technology under grant no. 214/SBAD/0220; and by the European Union's Horizon 2020 research and innovation programme under grant agreement ID: 871479 AERIAL-CORE.

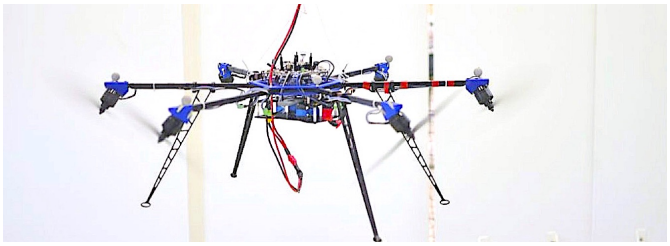


Fig. 1. Considered hexarotor with tilted propellers (Tilt-Hex)

The rest of this letter is structured as follows: Sec. II presents the UAV model and the BLF controller. Sec. III analyzes carefully the parameters of the BLF controller, and discusses the corresponding interdependency. Sec. IV summarizes the tuning algorithm and its use for the tuning of the BLF controller. Finally, Sec. V shows the experimental analysis of the proposed method, and Sec. VI concludes the letter.

II. Modeling and Control of Fully Actuated UAVs with Bounded Lateral Force

In this section, we briefly describe the modeling and control of fully-actuated aerial vehicles with *Bounded Lateral Force*, in order to introduce the parameters whose automatic tuning represents the goal of the proposed method. A BLF model is a powerful and simple abstraction of several different multi-rotor designs, including, e.g., underactuated, fully-actuated, multi-directional thrust, and thrust vectored designs. For a detailed description of the BLF model and its relation with real multirotor designs, we refer the reader to [8], where all these concepts were introduced.

We define an inertial world frame $\mathcal{F}_W$ with origin O_W and axes $\{\boldsymbol{x}_W, \boldsymbol{y}_W, \boldsymbol{z}_W\}$, and a robot frame $\mathcal{F}_R$, attached to the vehicle, with origin O_R and axes $\{\boldsymbol{x}_R, \boldsymbol{y}_R, \boldsymbol{z}_R\}$, where O_R coincides with the Center of Mass (CoM) of the vehicle. We denote by $\boldsymbol{p}_R \in \mathbb{R}^3$ and $\boldsymbol{R}_R \in SO(3)$ the position of O_R in $\mathcal{F}_W$ and the rotation matrix describing the orientation of $\mathcal{F}_R$ with respect to (w.r.t.) $\mathcal{F}_W$, respectively. The linear velocity of O_R in $\mathcal{F}_W$ is denoted with $\boldsymbol{v}_R \in \mathbb{R}^3$ and the angular velocity of $\mathcal{F}_R$ w.r.t. $\mathcal{F}_W$ expressed in $\mathcal{F}_R$ is denoted with $\boldsymbol{\omega}_R \in \mathbb{R}^3$. Finally, $m_R \in \mathbb{R}_{>0}$ and $\boldsymbol{J}_R \in \mathbb{R}^{3\times3}_{>0}$ denote the vehicle mass and moment of inertia w.r.t. to O_R in $\mathcal{F}_R$, respectively.

Following the Newton-Euler formalism, we can write the equations of motion of this rigid body as

$$\dot{\boldsymbol{p}}_R = \boldsymbol{v}_R \tag{1}$$

$$\dot{\boldsymbol{R}}_R = \boldsymbol{R}_R[\boldsymbol{\omega}_R]_\times \tag{2}$$

$$m_R\dot{\boldsymbol{v}}_R = -m_R\, g\, \boldsymbol{e}_3 + \boldsymbol{R}_R\boldsymbol{f}_R\,, \tag{3}$$

$$\boldsymbol{J}_R\dot{\boldsymbol{\omega}}_R = -\boldsymbol{\omega}_R \times \boldsymbol{J}_R\boldsymbol{\omega}_R + \boldsymbol{\tau}_R\,, \tag{4}$$

where $[\bullet]_\times$ is the skew-symmetric operator, $\boldsymbol{e}_3$ is a unit vector along $\boldsymbol{z}_W$, g is the gravitational constant, and $\boldsymbol{f}_R = [u_x,\, u_y,\, u_z]^\top \in \mathcal{U}_f \subset \mathbb{R}^3$ and $\boldsymbol{\tau}_R \in \mathbb{R}^3$ are the total control force and moment applied on O_R in $\mathcal{F}_R$, respectively, and $\mathcal{U}_f$ represents the *set of feasible forces* in the robot frame. A BLF vehicle is characterized by the particular structure of the set $\mathcal{U}_f$, namely, it is $\mathcal{U}_f = \mathcal{U}_{xy} \times \mathbb{R}_{\geq 0}$, where $\mathcal{U}_{xy}$ is the *set of feasible lateral forces* defined as

$$\mathcal{U}_{xy} = \{[u_x,\, u_y]^\top \in \mathbb{R}^2 | u_x^2 + u_y^2 \leq r_{xy}^2\}. \tag{5}$$

The distinguishing feature of the BLF model is the presence of the parameter $r_{xy} > 0$, which represents the *maximum intensity of lateral (horizontal) force* that the BLF vehicle can produce in $\mathcal{F}_R$. The smaller r_{xy}, the closer the BLF vehicle resembles an underactuated multirotor (e.g., a quadrotor) and as a consequence, the more coupled are its lateral motion and attitude – e.g., a lateral acceleration requires a non-zero tilting of the vehicle. The larger r_{xy}, the more decoupled can be its orientation and lateral motion – e.g., the BLF vehicle can accelerate laterally with a small tilting and can tilt with lateral acceleration close to zero.

There are two main advantages of the BLF model compared to a more accurate multi-parametric and coupled model for the particular multi-rotor aerial vehicle at hand. First, the BLF model is much simpler and requires the identification and use of only one actuation parameter – namely r_{xy}. Second, the BLF model can be made asymptotically stable using a controller (see [8]) that is analytically proven to converge and it has been experimentally demonstrated to effectively stabilize real multirotor platforms modeled as BLF. More accurate models are possible [11], however, they require complex identification procedures of many parameters, which may be impractical. Furthermore, due to their complexity, they can be controlled only resorting to numerical optimization-based control, which typically requires a high computational power that may be not available onboard. Last but not least, such numerical methods do not have typically an analytical guarantee of asymptotical stabilization.

A BLF vehicle can be stabilized along a time-varying and full-pose reference trajectory $\boldsymbol{q}^r(t) = (\boldsymbol{p}_R^r(t), \boldsymbol{R}_r(t))$ using the analytically proven control law presented in [8], which has the following form:

$$\boldsymbol{f}_R = \mathrm{sat}_{\mathcal{U}_{xy}}\left((\boldsymbol{f}_r^\top \boldsymbol{R}_R\boldsymbol{e}_1)\boldsymbol{e}_1 + (\boldsymbol{f}_r^\top \boldsymbol{R}_R\boldsymbol{e}_2)\boldsymbol{e}_2)\right) + (\boldsymbol{f}_r^\top \boldsymbol{R}_R\boldsymbol{e}_3)\boldsymbol{e}_3 \tag{6}$$

$$\begin{aligned}\boldsymbol{\tau}_R = {} & \boldsymbol{\omega}_R \times \boldsymbol{J}_R\boldsymbol{\omega}_R - \boldsymbol{K}_{\boldsymbol{R}}\boldsymbol{e}_{\boldsymbol{R}} - \boldsymbol{K}_\omega\boldsymbol{e}_\omega - \\ & - \boldsymbol{J}_R\left([\boldsymbol{\omega}_R]_\times \boldsymbol{R}_R^\top\boldsymbol{R}_d\boldsymbol{\omega}_R^d - \boldsymbol{R}_R^\top\boldsymbol{R}_d\dot{\boldsymbol{\omega}}_R^d\right)\,,\end{aligned} \tag{7}$$

where

$$\boldsymbol{f}_r = m_R(\dot{\boldsymbol{v}}_R^r + g\boldsymbol{e}_3) - \boldsymbol{K}_p\boldsymbol{e}_p - \boldsymbol{K}_v\boldsymbol{e}_v\,. \tag{8}$$

Considering the goal of this paper, we omit the details for the sake of compactness and readability, and we refer the reader to [8] for the exact definition of all the terms in the controller and the stability proof. In the next section, we focus on the discussion of the parameters used in this control law.

III. Discussion on the Control Parameters and need for Automatic Tuning

The BLF model (1)–(5) has three parameters: m_R, $\boldsymbol{J}_R$, and r_{xy}, which are all used in the corresponding controller in (6)–(8) together with the four additional sets of parameters representing the control gain matrices $\boldsymbol{K}_p$, $\boldsymbol{K}_v$, $\boldsymbol{K}_{\boldsymbol{R}}$, and $\boldsymbol{K}_\omega$. In the following, we analyze each parameter from the point

of view of interdependency and easiness of identification with methods available in the state of the art.

A. Inertial Parameters

The inertial parameters m_R and $\boldsymbol{J}_R$ have clear physical meaning and their offline identification or online estimation is rather straightforward and established (see, e.g., [12] and [13]). Furthermore, their nominal values are typically accurate because they can be retrieved from the CAD model of the system. Therefore, one can safely assume their values to be identifiable with good accuracy using standard methods.

B. Gains of the Attitude Control loop

The gain matrices $\boldsymbol{K_R}$ and $\boldsymbol{K}_\omega$ appear in (7) and affect the attitude dynamics (2),(4), which is independent of the rest of the vehicle dynamics (the position dynamics) and contains only the parameter $\boldsymbol{J}_R$, which, as explained before, can be fairly assumed to be known with good accuracy. Furthermore, the attitude dynamics is fully-actuated and no limits in the control moments appear in the BLF model. Thanks to the large control authority of the moment actuation and relatively small inertia of multi-rotor platforms, the gains $\boldsymbol{K_R}$ and $\boldsymbol{K}_\omega$ can be easily tuned independently from the other control parameters by using standard PD tuning methods such as the one presented in [14]. Therefore, we can also in this case safely assume that $\boldsymbol{K_R}$ and $\boldsymbol{K}_\omega$ are tuned with state-of-the-art methods and do not require special attention.

C. Maximum Intensity of the Lateral Force

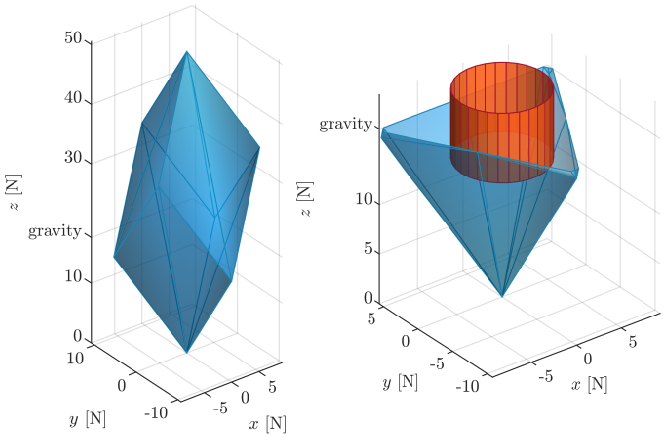
In a real multi-rotor platform, see [15], the maximum intensity of the lateral force depends on the applied vertical component of the force and the applied full moment. Such lateral bound is typically larger when the requested vertical force exactly compensates for the gravity force and the total moment is zero. The farther the vertical force and the moment are from such two neutral conditions the smaller are the lateral bounds on the horizontal component of the force. Figure 2 shows an example feasible force set at hover and a corresponding BLF model calculated when applying a vertical force opposing gravity.

In the BLF model, on the contrary, r_{xy} is a lumped constant value. If r_{xy} is set too small then the controller will let the platform behave too close to an underactuated platform, if r_{xy} is set too large, it could lead to suboptimal behaviors because it may not represent well the lateral bounds induced by the moment and vertical force required by the task.

In conclusion, the parameter r_{xy} plays the role of a 'lateral-actuation slider', which position has to be tuned, in order to optimize the behavior of the real controlled platform for the particular task at hand. Such a need calls for an automatic tuning algorithm that can optimize the value of r_{xy} based on the controller performance.

D. Gains of the Position Control loop

The gain matrices $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ appear in (8) and affect the position dynamics (1),(3). It is standard to assume a diagonal



(a) (b)

Fig. 2. (a) Feasible force set of the platform in Fig. 1 at hover. (b) Same feasible force set with cross section at the gravity plane. The figure also shows the BLF cylinder for the platform applying a lift force $\pm 20\%$ of the gravity opposing lift.

structure of $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ considering the symmetry of the model and in order to avoid an unnecessary cross-direction coupling induced by the controller. Furthermore, thanks to the horizontal symmetry of the model and controller, one can assume that the fist two entries of the diagonals are equal. Therefore, it is reasonable to assume the following structure for $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$:

$$\boldsymbol{K}_p = \mathrm{diag}\left\{k_p,\, k_p, k_{p,z}\right\}, \tag{9}$$

$$\boldsymbol{K}_v = \mathrm{diag}\left\{k_v,\, k_v, k_{v,z}\right\}. \tag{10}$$

The choice of $k_{p,z}$ and $k_{v,z}$ affects the closed-loop vertical dynamics along which the system has a large control authority and is not influenced by the rest of the dynamics. Therefore – similarly to $\boldsymbol{K_R}$ and $\boldsymbol{K}_\omega$ – the gains $k_{p,z}$ and $k_{v,z}$ can be tuned independently, e.g., letting the vehicle move up and down and using standard PD tuning techniques [14].

The remaining parameters, namely k_p and k_v, cannot be chosen independently from r_{xy}, because in (6) there is a nonlinear saturation on the lateral dynamics that depends on r_{xy}. Therefore, k_p and k_v have to be chosen in a way that lets the system behave optimally in the lateral motion, while well coping with the saturation induced by r_{xy}.

To provide an insight into the complexity of such interplay, let us first consider the two extreme cases. If r_{xy} is chosen very small, the platform's lateral dynamics is in practice underactuated and the platform needs to tilt, to move laterally. Therefore, the gains k_p and k_v have to be optimized to let the position dynamics be as fast as possible but 'slower' than the attitude dynamics, as in a quadrotor. On the contrary, if r_{xy} is large, the system can move laterally (up to a certain acceleration) without tilting, therefore there is virtually no need to take into account the attitude dynamics in the tuning of k_p and k_v. On the other side, there is still the dynamics of the motor/propeller to consider. In fact, lateral motions without tilting require a much more ample range of propeller

spinning velocities compared to the case in which the system moves laterally by tilting (with small r_{xy}). This phenomenon can be easily appreciated looking at the experiments done in [8]. Therefore, for large r_{xy} the dynamics to consider is the motor/propeller one, which has of course different characteristics than the attitude one.

For intermediate values of r_{xy}, a mixture of attitude and motor dynamics influences the optimal choice of the gains k_p and k_v in a way that is hard to predict a priori.

E. Conclusions

From the discussion carried out in this section, it emerges that there are two different types of control parameters in (6)–(8). The first type, namely m_R, $\boldsymbol{J}_R$, $\boldsymbol{K_R}$, $\boldsymbol{K}_\omega$, $k_{p,z}$ and $k_{v,z}$ can be tuned mostly independently and resorting to state-of-the-art methods such as, e.g., physical parameter identification using least squares or PD tuning. The second type, namely r_{xy}, k_p, and k_v, are tightly coupled, and their effects on the system behavior are coupled and nonlinear and one cannot use straightforward methods like PD tuning to tune these parameters. In particular,

- different values for r_{xy} may be chosen depending on the motion task at hand, where there is no clear 'best value' until the task is specified;
- for each value of r_{xy} it is expected to obtain different optimized values for k_p and k_v, due to the nonlinear interplay explained before

Therefore, in the following, for the first time in the literature, we focus our attention on the *optimal tuning of k_p and k_v for different values of r_{xy} in real platforms modeled and controlled as BLF*. First, we describe the automatic method used for the tuning (Sec. IV), and then we test the presented method on a real platform. These tests demonstrate the existence of the expected dependency as well as the improvement of the controller performance following the presented tuning method (Sec. V).

IV. Data-based Parameter Tuning Algorithm

The optimization algorithm that is used to tune the gains k_p and k_v, for a given value of the parameter r_{xy}, is an instantiation of the model-free OPTIM-tune algorithm presented in [10] and requires only a measurable metric of the performance of the controller in order to work. Further analysis on the convergence of the OPTIM-tune algorithm can be found in [10]. The overall method is a combination of two nested loops: *i)* An outer loop, described in Algorithm 1, and *ii)* an inner loop, also called *single parameter tuning*, described in Algorithm 2.

Algorithm 1 (the outer loop) receives as input: *i)* the maximum lateral force r_{xy} (which is kept constant during the tuning), *ii)* the two sets $\mathscr{D}_{k_p}^{(1)}$, $\mathscr{D}_{k_v}^{(1)}$ that represent the intervals over which the gains k_p and k_v are optimized, and *iii)* two integers, N_b and N, representing the number of iterations in the outer and inner loop, respectively.

Algorithm 2 (inner loop, or single parameter tuning) receives as input: *i)* r_{xy} and N (the same of Algorithm 1), *ii)* the starting set, denoted with $\mathscr{D}^{(0)}$, in which one of the two gains (either k_p or k_v) will be optimized, and *iii)* the value of the other parameter (either k_v or k_p) that is kept fixed during the execution of Algorithm 2, denoted with ξ. The algorithm provides as output a new set $\mathscr{D}^{(N)}$, which is a contraction of $\mathscr{D}^{(0)}$ and is guaranteed to contain the optimum value of the corresponding parameter.

Algorithm 1 executes N_b times a basic iteration in which two instances of Algorithm 2 are performed sequentially to contract $\mathscr{D}_{k_p}^{(i)}$ and $\mathscr{D}_{k_v}^{(i)}$. In the first instance, k_v is kept fixed to its current estimate and the set to which the optimal k_p belongs is contracted, thus generating a new improved estimate of k_p. In the second instance – symmetrically – the new estimate of k_p is kept fixed while the set to which the optimal k_v belongs is contracted, thus generating a new improved estimate of k_v. At the end of Algorithm 1 the optimized values of the gains are returned in the form of the mid-values of the intervals generated by the contractions of the last (N_b-th) iteration, i.e., $\mathscr{D}_{k_p}^{(N_b)}$ and $\mathscr{D}_{k_v}^{(N_b)}$. Such mid-values are denoted with $\overline{\mathscr{D}}_{k_p}^{(N_b)}$ and $\overline{\mathscr{D}}_{k_v}^{(N_b)}$, respectively.

Algorithm 1: Tuning of controller gains k_p and k_v for a fixed r_{xy} value.

Data: r_{xy}, $\mathscr{D}_{k_p}^{(1)}$, $\mathscr{D}_{k_v}^{(1)}$, N and N_b
Result: optimized $k_p^\star \in \mathscr{D}_{k_p}^{(1)}$, $k_v^\star \in \mathscr{D}_{k_v}^{(1)}$
$i \leftarrow 1$;
while $i \leq N_b$ **do**
  $k_v \leftarrow \overline{\mathscr{D}}_{k_v}^{(i)}$;
  $\mathscr{D}_{k_p}^{(i+1)} \leftarrow \text{algorithm}_2(\mathscr{D}^{(1)} = \mathscr{D}_{k_p}^{(i)}, \xi = k_v, r_{xy}, N)$;
  $k_p \leftarrow \overline{\mathscr{D}}_{k_p}^{(i+1)}$;
  $\mathscr{D}_{k_v}^{(i+1)} \leftarrow \text{algorithm}_2(\mathscr{D}^{(1)} = \mathscr{D}_{k_v}^{(i)}, \xi = k_p, r_{xy}, N)$;
  $i \leftarrow i + 1$
end
$k_p^\star = \overline{\mathscr{D}}_{k_p}^{(N_b)}$;
$k_v^\star = \overline{\mathscr{D}}_{k_v}^{(N_b)}$;

Algorithm 2: Generic single parameter tuning.

Data: $\mathscr{D}^{(1)} = [{}^-\theta^{(1)}, {}^+\theta^{(1)}]$, ξ, r_{xy}, and N
Result: $\mathscr{D}^{(N)} = [{}^-\theta^{(N)}, {}^+\theta^{(N)}]$
$i \leftarrow 1$;
while $i \leq N$ **do**
  calculate contraction factor ρ_i;
  calculate candidate $(i+1)$ bounds:

$$
\begin{aligned}
{}^-\hat{\theta}^{(i+1)} &= {}^-\theta^{(i)} + \rho_i({}^+\theta^{(i)} - {}^-\theta^{(i)}) \\
{}^+\hat{\theta}^{(i+1)} &= {}^-\theta^{(i)} + (1-\rho_i)({}^+\theta^{(i)} - {}^-\theta^{(i)})
\end{aligned}
$$

  execute flight test with params ξ, ${}^-\hat{\theta}^{(i+1)}$, and r_{xy};
  ${}^-f \leftarrow f(\boldsymbol{s}_e)$;
  execute flight test with params ξ, ${}^+\hat{\theta}^{(i+1)}$, and r_{xy};
  ${}^+f \leftarrow f(\boldsymbol{s}_e)$;
  if ${}^-f < {}^+f$ **then**
    $\mathscr{D}^{(i+1)} \leftarrow [{}^-\hat{\theta}^{(i+1)}, {}^+\theta^{(i)}]$;
  else
    $\mathscr{D}^{(i+1)} \leftarrow [{}^-\theta^{(i)}, {}^+\hat{\theta}^{(i+1)}]$;
  end
  $i \leftarrow i + 1$;
end

Algorithm 2 performs the set contraction implementing N smaller consecutive contraction steps. Each step executes two flight tests with the vehicle, using r_{xy} as lateral force bound and ξ as the temporarily fixed gain. The goal of each flight test is to evaluate the effect of a new candidate for the upper and lower bound of the set containing the gain to be optimized. A new upper bound candidate ${}^{+}\hat{\theta}^{(i+1)}$ is tested in the first flight test, while a new lower bound candidate ${}^{-}\hat{\theta}^{(i+1)}$ is tested in the second one. Each flight test is followed by the evaluation of a cost function f that depends on the state error $\boldsymbol{s}_e$, i.e., the vector describing the error between desired and measured state of the system during the execution of a flight test. The candidate bound that corresponds to the test which returned the lower value of f is used as new upper or lower bound for the set of the estimated parameter, thus producing the sought contraction for the particular step. This process is repeated N times. The last obtained set $\mathscr{D}^{(N)}$ is returned as the result of the algorithm.

In our specific case, the goal is to find the optimal controller gains that ameliorate the lateral trajectory tracking while the platform remains as much as possible horizontal – thus exploiting at best the lateral force capability of BLF platforms. In line with such goal, and assuming that the position trajectory of a flight test is composed by N_c reference points, the corresponding cost function f is defined as follows,:

$$f(\boldsymbol{s}_e) = \underbrace{\sum_{k=1}^{N_c} |\boldsymbol{e}_k|}_{f_{\mathbf{e}}} + \frac{1}{Q} \underbrace{\sum_{k=1}^{N_c} |\phi_k|}_{f_{\phi}} = \sum_{k=1}^{N_c} J\,, \tag{11}$$

where $|\boldsymbol{e}_k|$ is the norm of the lateral position error, $|\phi_k|$ is the norm of the tilt angular error and J is the weighted error per reference point. The parameter Q is used to weight the contribution of the angular error in the overall performance index, giving more or less importance to the fact that the platform remains horizontal while following the position trajectory.

V. Test of Tuning on a Real Platform and Validation of the Interdependency Hypothesis

We validate the proposed approach with the BLF platform presented in [8]; the platform is referred to as the Tilt-Hex and is shown in Fig. 1. The platform is a hexarotor constructed from six 12" tilted propellers equally-spaced about the platform center of mass. The platform has a mass of $1.8\,\mathrm{kg}$, and an inertia tensor $\boldsymbol{J}_R = \mathrm{diag}\,\{11.5,\, 11.4,\, 19.4\} \cdot 10^{-6}\,\mathrm{kg}\cdot\mathrm{m}^2$.

In addition, the platform is endowed with an Inertial Measurement Unit (IMU) providing acceleration and angular velocity measurements at $1\,\mathrm{kHz}$, and is tracked with a motion capture system at $100\,\mathrm{Hz}$. Both measurements are fused with an Unscented Kalman Filter running at $1\,\mathrm{kHz}$, providing an estimate of the platform state. The motion controller runs onboard at $1\,\mathrm{kHz}$, and brushless motor controllers (BLDC ESC) regulate the propellers' speed using an in-house developed closed-loop speed controller [16]. Most of the software is developed in C++ and runs on an on-board PC, on the exception of the gain tuning algorithm which runs in Matlab/Simulink on a ground PC. Most of the onboard software are open source, and can be found at https://git.openrobots.org/projects/telekyb3, while the Optim-tune software will be made available at https://github.com/AppliedControlTechniques/Optim-tune.

A simulation and an experimental campaign have been carried out, in which the task has been to tune the controller gains, for selected values of r_{xy} and different Q ratios. The interested reader is referred to the multimedia attachment of this letter for the experiments' videos. In all the experiments and simulations, the lateral acceleration of the reference trajectory has been pushed up to $1.5\,\mathrm{ms}^{-2}$, jerk to $10\,\mathrm{ms}^{-3}$, lateral velocity to $2\,\mathrm{ms}^{-1}$, to bring the platform to its lateral motion limits. During each of the optimizations, the initial domains of the controller gains have been chosen such that $\mathscr{D}^{(1)}_{k_p} = [10,\, 30]$ and $\mathscr{D}^{(1)}_{k_v} = [5,\, 15]$, and each domain has been contracted $N = 12$ times by the inner loop in each of the $N_b = 2$ outer loop iterations.

Finally, to stress the lateral position tracking and horizontality of the platform (zero tilt), the reference trajectory has been chosen as a back and forth path parallel to $\boldsymbol{x}_W$.

A. Simulative Analysis of the Cost Function Landscape

To study the cost function landscape, we simulated the above mentioned platform with the corresponding controller, and computed the different components of the cost function while varying the controller positional gains k_p and k_v over a discretized grid of $\mathscr{D}^{(1)}_{k_p}, \mathscr{D}^{(1)}_{k_v}$.

Fig. 3 shows the contour plots of the two components of $f(\mathbf{s}_e)$ at $r_{xy} = 2$ and $r_{xy} = 8$. The functions $f_{\mathbf{e}}$ and f_{ϕ} are shown separately to understand the effect of each on the performance of the presented tuning scheme. This figure shows that as r_{xy} is increased from 2 to 8, the lateral position error component $f_{\mathbf{e}}$ decreases slightly for almost all values in the given range, and the number of local minima slightly increases and the position changes moderately. On the other hand, f_{ϕ} shows a different behavior, where it can be seen that the number of local minima increases substantially for $r_{xy} = 8$. In addition, as r_{xy} is increased, the values of f_{ϕ} decreases substantially; this can be seen from the different scales of the corresponding contour plots. As such, as r_{xy} is increased the effect of the angular component on the overall cost function decreases; this is similar to an increase in the Q value. This shows the decreased effect of f_{ϕ} (and correspondingly the angular dynamics) on the tuning of the position controller gains as r_{xy} increases, even when Q is kept constant.

B. Experimental Test of the Tuning Algorithm

We conduct an experimental campaign to demonstrate the tuning of the proposed algorithm and to show the relation between the chosen r_{xy} and the optimized controller gains.

Fig. 4 shows the average of the optimized controller gains and standard deviation of each for different values of r_{xy} and Q, where the optimization at each r_{xy} and Q value have been repeated thrice. This figure shows that as r_{xy} increases, the optimized k_p increases and the optimized k_v decreases for both Q values. In the case of $Q = 300$, the contribution of f_{ϕ} is reduced substantially leading to higher optimized k_p and k_v values as the optimization allows for more aggressive

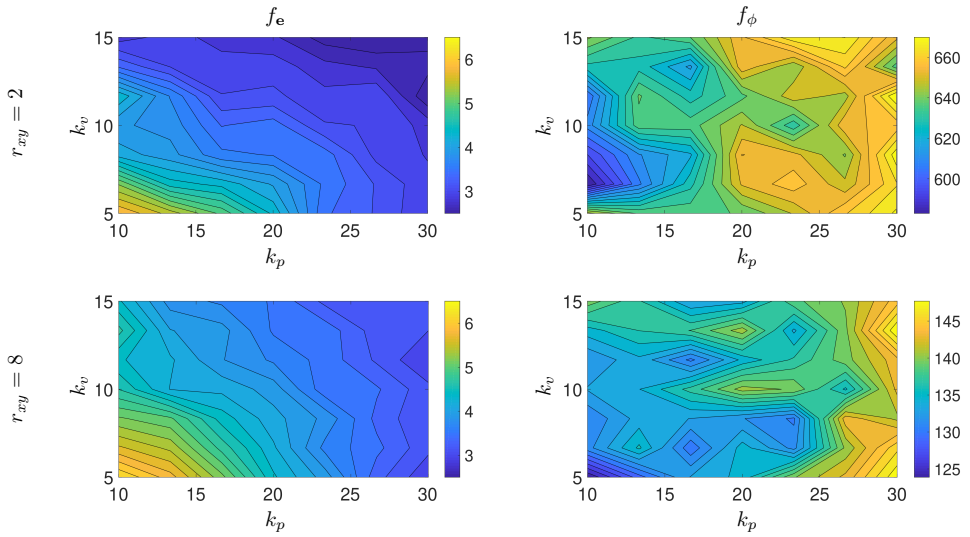
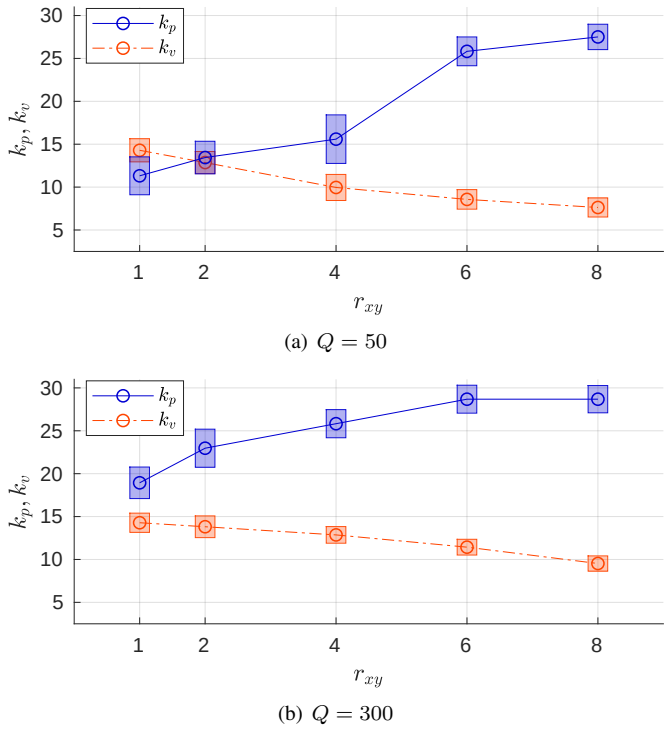


Fig. 3. Contour plots of $f_{\mathbf{e}}(k_p, k_v)$ and $f_\phi(k_p, k_v)$ for two different values of r_{xy} computed via simulation on a discretization of the ranges $\mathscr{D}_{k_p}^{(1)}, \mathscr{D}_{k_v}^{(1)}$.



(a) $Q = 50$

(b) $Q = 300$

Fig. 4. Optimized gains (k_p, k_v) of the position controller versus r_{xy} for two values of the weighting ratio Q. For each optimized gain, we show the average value and the standard deviation bar computed from three repetitions of each experiment.

position tracking without taking the angular tracking error into consideration.

In order to get an improved understanding of how the algorithm operates, we show the evolution of the optimization for two of the above optimization cases, corresponding to $r_{xy} \in \{2, 8\}$ at a constant $Q = 50$. Fig. 5 shows the corresponding evolution for $r_{xy} = 2$ and $r_{xy} = 8$. This figure shows the evolution of the controller gains and the resulting performance index for each of the two r_{xy} values. These experiments show clearly the improved trajectory tracking of the controller with the optimized gains, and that while the initial controller gains are the same for both experiments, the optimized gains vary substantially for different values of r_{xy}, supporting the interdependence of such parameters that is claimed in this work.

VI. CONCLUSIONS

In this letter, we studied the interplay between the parameters and gains of the BLF platform while controlled via a full-pose controller. We showed that the optimally chosen position control gains rely largely on the estimated lateral force limit.

While one can fix the estimated lateral force limit at one of its extreme values, we discussed the effect of these extremes, where if the limit is chosen small enough the platform behaves as an underactuated one, while if chosen large enough it might exceed the physical capabilities of the platform, and thus the controller could behave sub-optimally.

We then presented a detailed method for the auto-tuning of the position control gains for different estimated lateral force limits, and showed how these optimized gains vary accordingly. However, the choice of this parameter is still an open question, where for each application, different values of the lateral force limit should be chosen. Moreover, based on the applied lift force, this parameter has to be changed with the corresponding optimized gains. As such, in the future, we propose to apply a gain scheduling approach to choose online the 'best' value of r_{xy} and the corresponding optimized gains.

ACKNOWLEDGMENT

We thank Anthony Mallet and Dario Sanalitro for their help in the hardware and software implementation of the experimental tests.

REFERENCES

[1] H. Almurib, P. Nathan, and T. Kumar, "Control and path planning of quadrotor aerial vehicles for search and rescue," in *SICE Annual Conference 2011*, 2011, pp. 700–705.

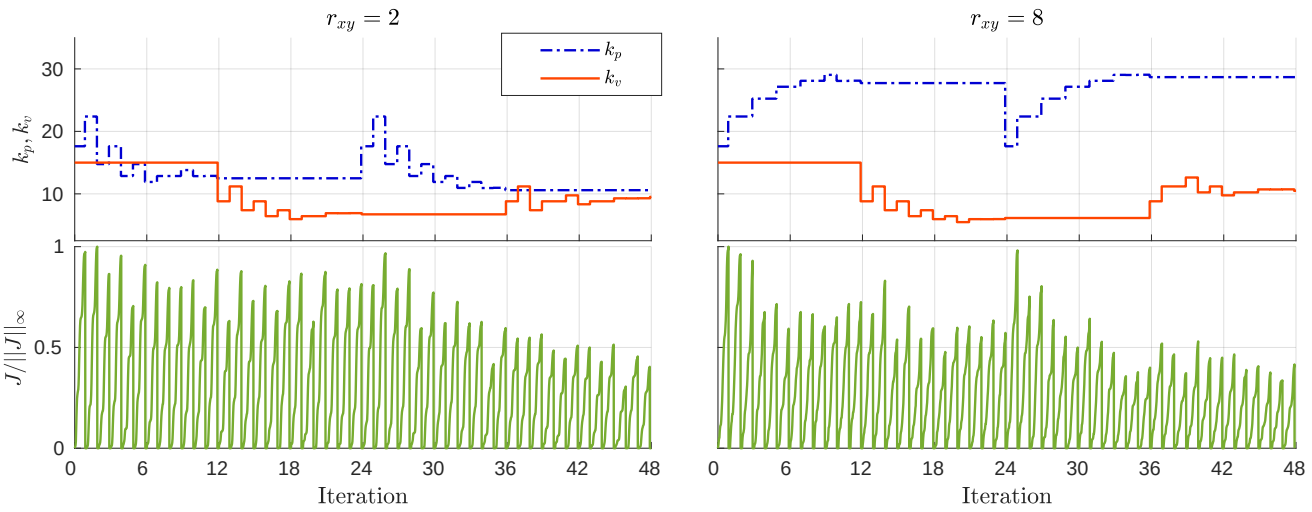


Fig. 5. Evolution of the gain tuning algorithm for $r_{xy} = 2$ and $r_{xy} = 8$ while $Q = 50$ showing the controller gains k_p, k_v, and the cost function $f(\mathbf{s}_e)$ for $N = 12$ and $N_b = 2$.

[2] L. Merino, J. M. de Dios, and A. Ollero, "Cooperative unmanned aerial systems for fire detection, monitoring, and extinguishing," *Handbook of Unmanned Aerial Vehicles*, pp. 2693–2722, 2015.

[3] A. Ollero, G. Heredia, A. Franchi, G. Antonelli, K. Kondak, A. Sanfeliu, A. Viguria, J. R. Martinez-de Dios, F. Pierri, J. Cortés, A. Santamaria-Navarro, M. A. Trujillo, R. Balachandran, J. Andrade-Cetto, and A. Rodriguez, "The AEROARMS project: Aerial robots with advanced manipulation capabilities for inspection and maintenance," *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 12–23, 2018.

[4] T. Báča, P. Stepan, V. Špurny, M. Saska, R. Penicka, G. Loianno, and V. K. J. Thomas J., "Autonomous landing on a moving vehicle with unmanned aerial vehicle," *Journal of Field Robotics*, vol. 36, no. 5, pp. 874–891, 2019.

[5] H. Romero, S. Salazar, A. Sanchez, and R. Lozano, "A new uav configuration having eight rotors: Dynamical model and real-time control," in *IEEE Conference on Decision and Control*, New Orleans, LA, USA, 2007, pp. 6418–6423.

[6] A. Albers, S. Trautmann, T. Howard, T. Nguyen, M. Frietsch, and C. Sauter, "Semi-autonomous flying robot for physical interaction with environment," in *IEEE Conference on Robotics, Automation and Mechatronics*, June 2010, pp. 441–446.

[7] M. Ryll, H. Bülthoff, and P. Giordano, "Modeling and control of a quadrotor uav with tilting propellers," in *IEEE Int. Conf. on Robotics and Automation*, Paul, MN, USA, may 2012, pp. 4606–4613.

[8] A. Franchi, R. Carli, D. Bicego, and M. Ryll, "Full-pose tracking control for aerial robotic systems with laterally-bounded input force," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 534–541, 2018.

[9] D. Invernizzi and M. Lovera, "Trajectory tracking control of thrust-vectoring uavs," *Automatica*, vol. 95, pp. 180–186, 2018.

[10] W. Giernacki, D. Horla, T. Báča, and M. Saska, "Real-time model-free minimum-seeking autotuning method for unmanned aerial vehicle controllers based on fibonacci-search algorithm," *Sensors*, vol. 19, no. 2, p. 312, 2019.

[11] D. B. D., J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs," *Journal of Intelligent and Robotic Systems*, 2020.

[12] V. Wüest, V. Kumar, and G. Loianno, "Online estimation of geometric and inertia parameters for multirotor aerial vehicles," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1884–1890.

[13] R. Spica, P. Robuffo Giordano, M. Ryll, H. H. Bülthoff, and A. Franchi, "An open-source hardware/software architecture for quadrotor UAVs," in *IFAC Proceedings Volumes*, Compiegne, France, Nov. 2013.

[14] H. Wu, W. Su, and Z. Liu, "Pid controllers: Design and tuning methods," in *IEEE Conference on Industrial Electronics and Applications*, 2014, pp. 808–813.

[15] D. Bicego, "Design and control of multi-directional thrust multi-rotor aerial vehicles with applications to aerial physical interaction tasks," Ph.D. thesis, Université de Toulouse, 2019.

[16] A. Franchi and A. Mallet, "Adaptive closed-loop speed control of BLDC motors with applications to multi-rotor aerial vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, May 2017, pp. 5203–5208.