2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

In-flight Efficient Controller Auto-tuning using a Pair of UAVs

Wojciech Giernacki[1], Dariusz Horla[1] and Martin Saska[2]

Abstract— **In the paper, a pair of auto-tuning methods for fixed-parameter controllers is presented, in application to multirotor unmanned aerial vehicles (UAVs) control. In both cases, the automatized process of searching the best altitude controller parameters is carried out with the use of a modified golden-search method, for a selected cost function, during the flight of a pair of UAVs. All the calculations are performed in real-time in the iterative manner using only basic sensory information available concerning current altitude information for a pair of UAVs. The auto-tuning process of the controller is characterized by neglectfully low computational demand, and the parameters are obtained rapidly with no dynamic model of a UAV needed. In both methods, by using a pair of UAVs in tuning process, the level of control performance can be increased, what has been proved by means of multiple outdoor experiments. The first method increases precision of the obtained controller parameters by averaging sensory information over a pair of UAVs, whereas in the second, by exchanging measurement information between the units, the search space is explored faster. The latter is of special importance when seeking the best controller parameters, what is especially expected when a limited experiment duration of multirotor UAVs is taken into account.**

SUPPLEMENTARY MATERIAL

This letter has supplementary video material available at `https://youtu.be/Lvj60i3HnqI`

I. INTRODUCTION

A rapid growth in a number of unmanned aerial vehicles is observed in the last decade, both in multirotor constructions, as well as in fixed-wing UAVs [1]. The UAVs are increasingly more often used in missions requiring precision, robustness and high level of safety during manual and autonomous flights. These requirements can be satisfied by the use of various control techniques and by incorporating advanced controller structures [2], [3], [4], [5]. Between the latter, fixed-parameter controllers are still an attractive with a maximum of a couple of parameters, such as these used broadly in commercial auto-pilots as Pixhawk, Naze32, Open Pilot, CC3D. This is due to the simplicity of their structure and high attenuation of disturbances, as well as resilience against changes in environmental conditions [6], [7], [8]. Currently, in UAV research field, optimal auto-tuning methods are sought for such controllers [9], [10], [11], as manual choice of parameters is ineffective and time-consuming, and further in-flight tuning is simply dangerous [12], [13].

[1]Wojciech Giernacki and Dariusz Horla are with Faculty of Control, Robotics and Electrical Engineering, Institute of Robotics and Machine Intelligence, Poznan University of Technology, ul. Piotrowo 3A, 60-965 Poznan, Poland `wojciech.giernacki@put.poznan.pl`

[2]Martin Saska is with the Faculty of Electrical Engineering, Czech Technical University in Prague, 166 36 Prague 6, Czech Republic

In a world literature, there is a great attention paid to auto-tuning procedures at the stage of prototyping of flying units and their further tuning for the specific mission undertaken, i.e. carried on-board equipment or load. Two main approaches can be identified here, namely: model-based [13], [14], [15], [16] and model-free (data-driven) approach [6], [7], [9], [12]. In the first case, for the newly-designed UAV, computer aided design techniques are widely used, and the obtained 3D model provides further information concerning moments of inertia of the UAV with certain mass. When this information is merged with specification data of the driving units, complete information is obtained, in order to parametrize a basic dynamical model of a UAV. It can be further implemented using such programming platforms as MATLAB/Simulink or Robot Operating System (ROS), enabling selection of controller parameters for UAVs by various auto-tuning techniques, with the further possibility to use these results in real-world experiments on a test bed, as well as during flight conditions. Such an approach to auto-tuning, presented in detail in [13] (online) and [16] (offline), increases safety of carried-out tests and enables one to use multiple control theory methods to obtain stabilizing ranges for controller parameters. From the other side, there are such inferior factors as the necessity to have a model adequate to dynamics of a real UAV, and increased computational burden connected with using this model. When design time is considered, model-based optimal tuning techniques for a given cost function and constraints taken into account, this approach should be still used as an off-line (batch) method. This feature makes the use of the above-mentioned methods during in-flight conditions impossible, or much harder when, e.g. parameters of a single flying unit or a group of transporting UAVs must be fine-tuned, in order to ensure exertion of nominal forces or achieving precision of manipulation of grippers attached (Fig. 1). It can be considered, in addition, when the energy in batteries of UAVs decreases to generate thrust force during flight, or in order to allow one to mount additional sensors or actuators, which are not taken into account in the available dynamical model.

Both in fine tuning and auto-tuning of UAVs their model is not highly necessary. Currently, a strong research trend can be observed in model-free techniques used to find the minimum of the selected performance index in an iterative manner, in experimental conditions [17]. In [13], two algorithms for real-time gain tuning of the PD (proportional-derivative) controller on a quadrotor system are shown. They are based on the deterministic method of the steepest descent and Newton's minimization of an objective function. Instead of using gradients, in [18], the authors presented

Fig. 1. Precise, autonomous flights: a) cooperative transport of objects during research camp in the South Bohemia `https://youtu.be/Pdg3j791I9c`, b) Challenge 1 (https://youtu.be/2-cLSjRCKDg), c) Challenge 2 (https://youtu.be/1-aRtSarYz4), d) Challenge 3 (https://youtu.be/O8QBiAyP2c0), during the Mohamed Bin Zayed International Robotics Challenge. Experiments highlighted the strong need for auto-tuning of UAV controllers and motivated research of this paper.

five deterministic methods based on zero-order iterative algorithm (Fibonacci-search, golden-search, equal division and dichotomy in two configurations). They compared their efficiency in real-world experiments for altitude controller of a custom hexacopter. The methods allow to adapt PD controller's gains by using only altitude measurements. In [7], the Matlab-Simulink Graphical User Interface (GUI) is presented that allows execution and automatic tuning of the fixed-wing Volcan UAV controllers by means of well-known Åström and Hägglund's method based on repeating step-type signal during the flight and due comparing the following step responses iteratively to achieve better performance declared via GUI. In the state-of-the-art therein, several probabilistic approaches to finding optimal controller sets for minimizing the time-varying control costs with the use of machine learning techniques are also reported – for example as Bayesian optimization [12], [19], [20], [21], [22], where usually performance function is model as a Gaussian process. Among these methods, the most important one from aspect of applications in mobile robotics, can be these presented in [12], [20], which modify the Bayesian optimization algorithm from [22] called SAFEOPT which guarantees *"(...) that the performance of the system never falls below a critical value; that is, safety is defined based on the performance function."* The SAFEOPT-MC generalized algorithm from [20] allows for Multiple Safety constraints separate from the objective. In the algorithm, still some initial expert knowledge concerning the process is necessary, to define initial set of safe parameters.

The main point of the current paper, is to perform efficient and model-free auto-tuning of the used two-parameter altitude controller of a hexacopter. The main novelty of the research and the result of the work carried out with reference to above-mentioned approaches, is the use of a pair of flying units, communicating with one another during flight in a noisy environment, in order to achieve the best altitude controllers' parameter tuning in the both UAVs using the proposed zero-order deterministic optimization algorithm. The algorithm minimizes a selected function of the tracking error with respect to the given reference altitude profile. In addition, the proposed approach is characterized by no need to approximate gradients, which are difficult to obtain from noisy measurements. Zero-order (gradient-free) optimization methods are not as prone to convergence to local optima and *"(...) explore the parameter space globally in a more data-efficient manner."* [20], as the methods that use gradient information. To the best of the knowledge of the authors, there are no design methods available, which by increasing the number of the flying units used during in-flight auto-tuning procedure result in a positive impact with increasing measurement precision, translating to an improvement in the quality of controller parameters (approach no. 1) or by shortening the tuning time (approach no. 2), welcomed e.g. when loads of different weights are transported or on-board equipment is replaced.

The paper is structured as follows: in Section II, the proposed methods of parallel tuning based on the golden-search algorithm are introduced and their mathematical basics are presented. The experimental testbed is shown in Section III. The evaluation of the performance of proposed methods in real-world experiments on a custom hexacopter is presented in Section IV, while conclusions are provided in Section V.

II. Golden-search based search algorithm

A. Introduction to the method

The optimization method presented below is based on the following assumptions:

- a mathematical model of an UAV is either imprecise or incomplete and finding the controller parameters is not possible using analytical calculations,
- a cost function $f(\cdot)$ can be defined to evaluate whether the current solution $\hat{x}$, i.e. gains of a fixed-parameter controller with at most few parameters, is optimal,
- a cost function is related to the performance of the UAV, i.e. tracking quality in a given time horizon for the considered parameters of a controller,

The calculation of the cost function value (e.g., the integral of the absolute error – IAE) is done repeatedly during the same flight, repeating its stages, by the evaluation of repeatable UAV behavior for consecutive controller gains. In this approach, the optimal solution $\hat{x}^*$ to the minimization problem (minimizer, treated as the optimal controller gains) is sought in the iterative manner, using an efficient, zero-order iterative algorithm (branch-and-bound algorithm) to find the best solution.

The information needed across iterative steps of the algorithm is reduced to altitude sensory information obtained every sampling period, what visibly decreases the burden of calculations of the method. By defining proper stopping criteria of the algorithm (e.g. a given tolerance to find the minimizer), the method has a deterministic running time. It is to be stressed that the method can be used to virtually tune any controller with a low number of parameters, what is a major improvement in comparison to other approaches, dedicated to particular controller structures.

A simple one-dimensional search method used here to tune controllers, is a well-known golden-search procedure [23], modified in the research to be used in parallel, where the basic principles of the method do not change, and are cited below for the sake of clear presentation of the results.

The considered algorithm is based on identifying the minimum of a predefined one-argument unimodal cost function f within some predefined bounds (from initial range $\mathscr{D}^{(k)}$) of its parameter, presented as $\left[x^{(0^-)},\, x^{(0^+)}\right]$ with $x^{(0^+)} > x^{(0^-)}$. From control systems viewpoint, the parameter of f can be understood as a gain of some controller, i.e. k_P gain in a PD controller. The above assumption, concerning applicability to unimodal functions, results in finding a local minimum, whenever the cost function turns out to be multimodal.

In the case of application to real-world, cost function is built on the basis of measurements of, e.g. tracking error squares within some time horizon. To find its minimum within certain range, when considering unimodal functions only, a pair of two interior points should be repeatedly selected from the current range, to reduce its length. By choosing the parameters as:

$$x^{(1^-)} - x^{(0^-)} = x^{(1^+)} - x^{(0^+)} = \rho(x^{(0^+)} - x^{(0^-)}), \quad (1)$$

where $\rho < \frac{1}{2}$ is reduction factor, a decision can be made if:

- $f(x^{(1^-)}) < f(x^{(1^+)}) \rightarrow \hat{x}^* \in \left[x^{(0^-)},\, x^{(1^+)}\right]$,
- $f(x^{(1^-)}) \geq f(x^{(1^+)}) \rightarrow \hat{x}^* \in \left[x^{(1^-)},\, x^{(0^+)}\right]$.

The pair of interior points are selected repeatedly.

The golden-search procedure [23] can be presented for arbitrary initial range where the optimum is sought with $x \in \mathscr{D}^{(0)} = \left[x^{(0^-)},\, x^{(0^+)}\right]$, as in Algorithm 1.

Whenever $k = 1, \ldots, N$ it holds that:

$$\hat{x}^{(k^-)} = x^{(k-1^-)} + \rho(x^{(k-1^+)} - x^{(k-1^-)})\,, \quad (3)$$
$$\hat{x}^{(k^+)} = x^{(k-1^-)} + (1-\rho)(x^{(k-1^+)} - x^{(k-1^-)})\,, \quad (4)$$

where the golden-search reduction factor $\rho = \frac{3-\sqrt{5}}{2}$ and N satisfy $(1-\rho)^N \leq \epsilon$.

The golden-search method, unlike some other branch-and-bound methods, due to the rule of selecting ρ allow either to reduce running time of the algorithm or average over past measurements, since at k-th iteration one point from a pair of points is known from the $k-1$-th iteration.

The proposed approach will use this one-dimensional search in a 2D search space, on the basis of the properties of the optimization method. Let $g(x_1,\, x_2)$ be a unimodal function within acceptable ranges of its parameters. For arbitrary $\alpha \in \mathscr{D}_1^{(0)}$ and $\beta \in \mathscr{D}_2^{(0)}$ the single-argument functions $g(x_1,\, \beta)$ and $g(\alpha,\, x_2)$ are obviously unimodal (for brevity denoted in both cases as $\varphi(x)$).

Algorithm 1 Golden-search algorithm.

Step 1. Calculate the required number of iterations N to achieve expected tolerance ϵ of the solution $\hat{x}^*$ with

$$|x^* - \hat{x}^*| \leq \epsilon(x^{(0^+)} - x^{(0^-)}), \quad (2)$$

where $\hat{x}^*$ is in the center of $\mathscr{D}^{(N)}$, $\hat{x}^*$ is the actual (unknown) minimum, and superscripts $^{(0^-)}$ and $^{(0^+)}$ depict left and right range limits in the predefined range at initial iteration.

Step 2. Iterate over $k = 1, \ldots, N$,

1) select a pair of points $\hat{x}^{(k^-)}$ and $\hat{x}^{(k^+)}$ ($\hat{x}^{(k^-)} < \hat{x}^{(k^+)}$, $\left\{\hat{x}^{(k^-)},\, \hat{x}^{(k^+)}\right\} \in \mathscr{D}^{(k-1)}$),
2) reduce the range to $\mathscr{D}^{(k)}$, where:
   a) $x^{(k+1)} \in \mathscr{D}^{(k)} = \left[x^{(k-1^-)},\, \hat{x}^{(k^+)}\right]$ for $f(\hat{x}^{(k^-)}) < f(\hat{x}^{(k^+)})$,
   b) $x^{(k+1)} \in \mathscr{D}^{(k)} = \left[\hat{x}^{(k^-)},\, x^{(k-1^+)}\right]$ otherwise,
   c) jump to the next iteration by putting $k := k + 1$.

Step 3. Stop the algorithm; put $\hat{x}^* = \frac{1}{2}\left(x^{(N^+)} + x^{(N^-)}\right)$.

The latter function clearly has a minimum at $x^* = \arg\min_{a \leq x \leq b} \varphi(x)$, thus the relation $a \leq x^{(1)} < x^{(2)} \leq x^*$ implies that $\varphi(x^{(1)}) > \varphi(x^{(2)})$, and $x^* \leq x^{(1)} < x^{(2)} \leq b$ implies that $\varphi(x^{(1)}) < \varphi(x^{(2)})$. After N function evaluations, $\hat{x}$ is within the contracted range, just as the true minimiser. Performing the same procedure at second stage of a bootstrap, enables improvement in a minimum search procedure in the 2D space (g is assumed to be unimodal).

B. Optimal gain tuning of a two-parameter controller

Optimal tuning of a two-parameter controller is based, firstly, on stipulating the tolerance ϵ which is the expected reduction ratio of the initial ranges for controller parameters. Secondly, the tuning is built of bootstrap cycles, which number N_b is defined by the user. A single bootstrap is formed up by two (in a two-parameter case) runs of the algorithm on the basis of an incremental calculation of values of f. One from a pair of parameters is fixed when the remaining one is tuned, starting from user-provided initial values, and vice versa.

For a pair of tuned parameters, forming the decision variables vector $\underline{x} = \begin{bmatrix} x_1, & x_2 \end{bmatrix}^T$ and $f(\underline{x})$ it is assumed:

- admissible ranges of $\underline{x}$, i.e. $\mathscr{D}_i^{(0)} = \left[x_i^{(0^-)}, \quad x_i^{(0^+)}\right]$, $i = 1, 2$, are known, and are usually related to ranges of stable behaviour of the UAV,
- a golden-search algorithm is used to find $\hat{x}^*$, and is based on bootstraps and algorithms presented hitherto,
- the cost function f is related to control performance using e.g. any control-based performance criteria, forming non-decreasing functions across time;
- number N is calculated for given ϵ, where tuning of a pair of controller parameters requires $2N$ steps in a single bootstrap sequence,
- the algorithm is terminated after N_b bootstraps, which can be stipulated e.g. based on the time when UAV

can safely operate on a single set of batteries. In the other case, the algorithm can be run in stages, after the batteries have been replaced.

Two-parameter controller tuning (for given N_b) can be summarized as in Algorithm 2.

Algorithm 2 Two-parameter controller tuning.

Step 0. Set bootstrap cycles counter to $l = 0$; for initial $\mathscr{D}_i^{(l)}$ $(i = 1, 2)$ define ϵ, N_b, and initial value of the second parameter $x_2^{(l)}$ (take $\hat{x}_2^{(l)^*} = x_2^{(l)}$), put $l := l + 1$.

Step 1. Calculate the optimal $\hat{x}_1^{(l)^*}$ using golden-search method, keeping the second parameter fixed at $\hat{x}_2^{(l-1)^*}$.

Step 2. Calculate the optimal $\hat{x}_2^{(l)^*}$ as in Step 1, keeping the first parameter fixed at $\hat{x}_1^{(l)^*}$.

Step 3. Increase the bootstrap cycles counter $l := l+1$ whenever when $l < N_b$, and proceed to Step 1, otherwise stop the algorithm – the optimal solution $\underline{\hat{x}}^* = \left[\hat{x}_1^{(l)^*}, \quad \hat{x}_2^{(l)^*}\right]^T$ has been obtained after N_b bootstrap cycles, as desired.

Finally, a general scheme to perform controller tuning using a pair of twin UAVs is as follows:

Algorithm 3 Controller tuning using a pair of twin UAVs.

Step 0. Initialize the tuning algorithms by sending controller parameters' values to both UAVs.

Step 1. Send a single reference primitive to both UAVs. Perform a single iteration at the bootstrap cycle considered using both UAVs and their parameters; Collect a pair of performance indices; Return performance indices to the ground station; Hover;

Step 2. Depending on the stage of the optimization process: increase bootstrap cycle counter, reduce the range or stop the algorithm; if applicable – return to Step 1.

At Step 0 or 2 of Algorithm no. 3 it can be assumed that:

- both the UAVs receive the same controller parameters, and in Step 2 the possible reduction of the ranges is performed taking averaged value of $f(\cdot)$,
- both the UAVs receive two different controller parameters, and in Step 2 the possible reduction of the ranges takes halved number of iterations (done in parallel), without averaging.

As it is impossible to claim that the tuning result is always optimal, one can clearly see that the method always terminates in a local minimum, what is a common case in local-search based algorithms. To avoid potential problems, especially in the situation when the system is highly sensitive to slight gain changes, the user would need to run the algorithm either from various initial points or by stipulating various ranges for gains sought. In addition, as a pair of UAVs used in subject to the same environmental conditions, the impact of the latter affects both the UAVs, usually for a longer period of time. To minimize the environmental conditions impact on the tuning procedure, it is easy to average performance index values at certain already-considered gain configuration over a number of prior trials.

C. On calculation of cost function values

In order to evaluate the cost function value $f(\cdot)$ when tuning controllers, the performance index must be calculated, and represented, e.g., as an incremental function computed at sampling time instants, equally spaced every T_S during the tuning procedure. If the sampling period T_S is short enough to capture dynamics of a system, then a single cost function value evaluation from Step 2 of the tuning procedure, might be accompanied by a single parameter change of the controller.

Such a change can be imagined as the source of transient behaviour in the control system. If N_c denotes the number of sampling periods necessary to calculate the performance index at l-th iteration of the tuning procedure, and $N_{\max}$ denotes the number of sampling periods related to the length of this procedure , with n as the sampling period counter incremented for every calculation of $f(\cdot)$, then the following applies:

- for $n = 1, \ldots, N_c - 1$ with the controller parameters are updated during $l-1$-th iteration, the performance index is evaluated by adding its increment, related to performance;
- for $n = N_c$ a single iteration of golden-search procedure is initialized, performance index is stored, and if applicable – the range for controller parameters is reduced or bootstrap is done;
- for $n = N_c + 1, \ldots, N_{\max}$ no parameter changes are done – this is the period when transient behaviour should decay, no performance index is collected.

Knowing the number of bootstrap cycles, number of iterations at single golden-search method run, and number of steps in every bootstrap, the running time of the tuning procedure is deterministic. The time can be further reduced by omitting some combinations of parameters that have been considered in previous iterations of the tuning algorithm.

In addition, the tuning method can be further modified either to take the value of the cost function as the averaged value over prior cost function values for previously-considered controller gains, or this combination of gains can be skipped. Also, as the presented tuning method must be applicable to real-world problems, the impact of measurement uncertainty, due to e.g. noise acting on the signals must be reduced. The latter can be done by low-pass filtration of the signals, where the filter is in the current application given applied as a ZOH-discretized first-order inertia with time constant of $0.1\,\mathrm{sec}$.

III. EXPERIMENTAL TESTBED

Altitude control system architecture for a hexacopter UAV is considered, as presented in [24] and [25], and is based on PID controller with I term fixed, and P/D parts tuned.

During real-world Experiments (no. 1-3) discussed in the further part of the paper, the custom hexacopter construction was used (designed originally for the MBZIRC2017

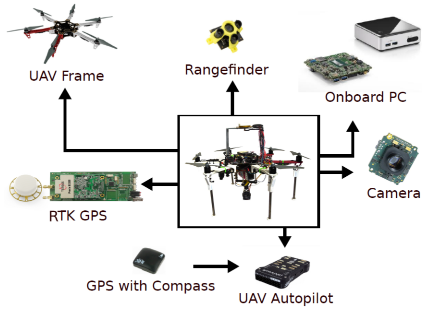


Fig. 2. Diagram of the customized UAV hexacopter modules.

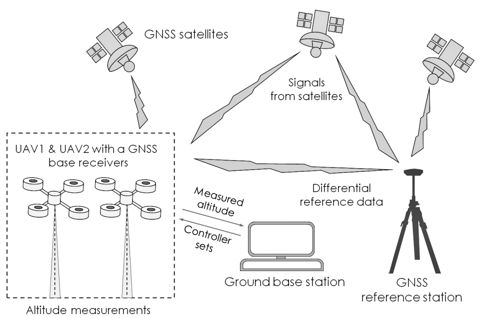


Fig. 3. Centralized measurements architecture used during the experiments with parallelized auto-tuning

TABLE I

CONFIGURATION PARAMETERS & RESULTS OF EXPERIMENTS NO. 1-3

	Experiment no. 1	Experiment no. 2	Experiment no. 3
UAV dynamics ($v_{\max}$ / $a_{\max}$)	3.33 / 11.11	3.33 / 11.11	3.33 / 11.11
no. of bootstrap cycles	2	2	1
Low-pass filtering	yes	yes	yes
Accuracy of calculations (ϵ)	0.05	0.05	0.05
no. of main iterations*			
planned / performed	56 / 45	56 / 45	28 / 14
no. of Li-Po batteries			
Gens Ace Tattu 6750 mAh	1	2 (1 for each UAV)	2 (1 for each UAV)
$\underline{x}^{(0)} = \left[k_P^{(0)}, \quad k_D^{(0)}\right]^T$ (initial values)	7, 9	7, 2	7, 2
Ranges: k_P	[4, 12]	[4, 12]	[4, 12]
Ranges: k_D	[2, 7]	[2, 7]	[2, 7]
Real tuning time [sec]	540	540	168
k_P and k_D final values	7.8095 / 5.0902	8.4323 / 4.6393	11.8622 / 6.8936
J_1 (after 1st iteration)	3.2926	1.4460	1.5804
J_I (after 1st bootstrapping)	1.9950	1.2004	2.7536
J_{end} (after the tuning proc.)	1.3652	1.1612	2.7536
J_{best} (best during the entire tuning)	0.9719 (in 25th iter.)	0.8784 (in 25th iter.)	1.0385 (in 5th iter.)
$J_{I\text{avg}}$ average (for 1 bootstrap)	1.7646	1.1317	2.4133
$J_{II\text{avg}}$ average (for 2 bootstrap)	1.6206	1.1401	$--$
Iteration no. for $J < 1.0385$	25	2	5
$(1 - J_{II_{\text{avg}}}/J_{I_{\text{avg}}}) * 100\%$	8.16 %	-0.74 %	$--$
$(1 - J_{I_{avgEXP(i)}}/J_{I_{avgEXP(1)}}) * 100\%$ **	$--$ %	35.87 %	-36.76 %
$(1 - J_{II_{avgEXP(i)}}/J_{I_{avgEXP(1)}}) * 100\%$	$--$ %	29.65 %	$--$

* In algorithm the same parameters combinations are omitted

** $J_{I\,avgEXP(i)}$ average (for 1 bootstrap) in i-th experiment

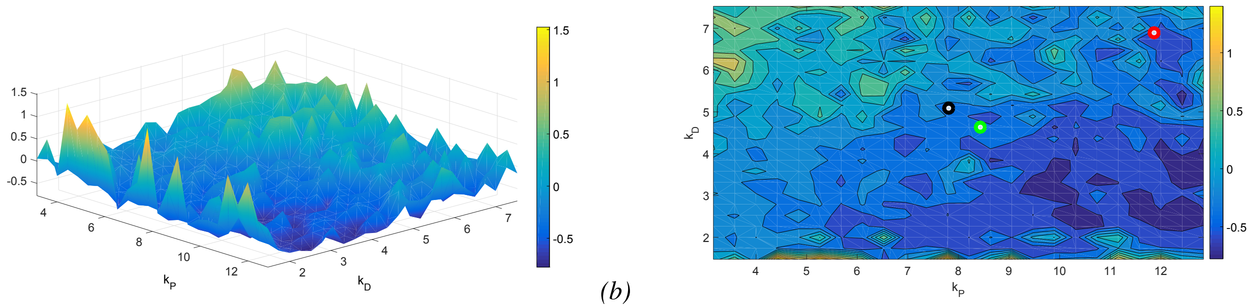
competition). For more information see [26], [27]. The full hardware specification of the considered platform is given in [28], and its block diagram including key components has been depicted in Fig. 2. Every UAV is built on the DJI F550 frame and using the DJI E310 driving units. The hexacopters are equipped with PixHawk flight controllers to stabilize orientation and the onboard computer Intel NUC-i7 with GNU Linux Ubuntu 16.04 and Robot Operating System (ROS) in Kinetic version. The estimate of the current altitude during the experiments is provided from an extended Kalman filter based on measurements from the precise TeraRanger rangefinder, IMU unit, pressure sensors, magnetometers and GPS units. To improve the altitude accuracy of the UAV in 3D space, the PRECIS-BX305 GNSS RTK BOARD differential GPS receiver was added. Differential Real Time Kinematics (RTK) GPS is using a ground base station to transmit the corrections to the particular UAV in order to eliminate the GPS drift. During Experiments (no. 2-3) where auto-tuning on a pair of UAVs is conducted, the architecture depicted in Fig. 3 has been used, with cross-communication through base station equipped with a WiFi module.

IV. REAL-WORLD EXPERIMENTS

A. Initial research

In order to evaluate the performance of the proposed auto-tuning methods of a two-parameter altitude controller of the hexacopter UAV, the initial tests carried out in ROS/Gazebo have been used. It has been done in the purpose to perform comparison with the tests carried out in [18]. For the dynamics model of the UAV a sequence of 576 pairs (combinations from the admissible range) of the altitude controller parameters have been set, i.e. $24 \times k_P$ (equally-spaced between 4 and 12) and $24 \times k_D$ (between 2 and 7), to obtain performance index (cost function) values, as the sum of absolute tracking errors performance index in a long time horizon, averaged over 20 periods of the reference signal (trajectory primitive with 12 sec length, changing in the range $2 \div 3$ m) for each of the 576 combinations of controller parameters. The obtained surface of the performance index ($J = f(k_P, k_D)$) is shown in Fig. 4 (in logarithmic scale). As can be vividly seen, the function $J = f(k_P, k_D)$ for the dynamics model of the UAV has a very flat nature and many local minima – especially problematic when using gradient-based optimization algorithms.

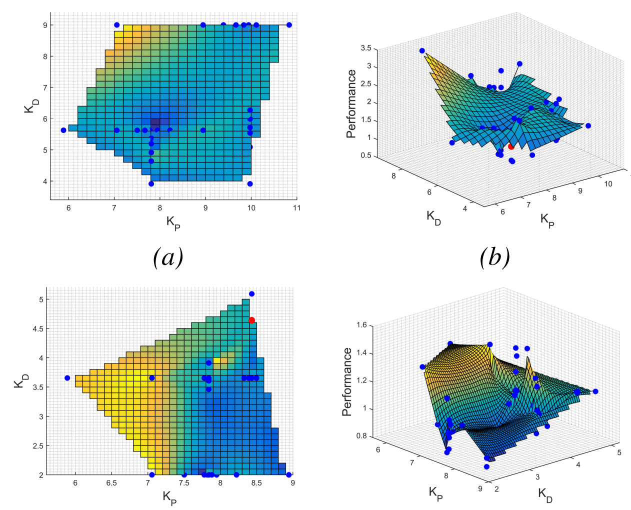
The performance index surface from Fig. 4b has been marked with results obtained from real-world Experiments no. 1-3 during the research camp in South Bohemia (Czech Republic) during April 2018. The video report from the experiments is available under (`https://youtu.be/Lvj60i3HnqI`). Configuration parameters taken during the experiment, and the selected, major, results are presented in Table I. Listing from subsequent steps of the used method can be found in `http://uav.put.poznan.pl`. During the recorded experiments, a single or a pair of UAVs (see Fig. 5) have been commanded to autonomously fly to the desired position using initial controller parameters, to start the tuning procedure. The procedure has been composed of cyclic changes in altitude, by repeating 12 sec long reference altitude profiles, changing altitude between 4 m and 3 m, with performance index J values stored after every trajectory primitive. After these step-by-step values of f have been used by the optimization algorithm to alter the gains (k_P, k_D) for the following iteration. With the sampling period of $T_S = 0.2$ sec, there were 60 samples recorded for one period of the trajectory primitive ($N_c = 50, N_{\max} = 60$). Within every 12 sec reference primitive, 10 sec was used to collect J, and last 2 sec allow transients decay past each optimization stage.



(a) *(b)*

Fig. 4. The two-parameter function surface $J = f(k_P, k_D)$ obtained on the basis of initial simulation tests: a) 3D view, b) 2D view with the altitude controller gains marked and obtained from real-world Experiments: no. 1 (black dot), no. 2 (green) and no. 3 (red), at the last iteration of the tuning

Fig. 5. Screenshots from experiment with the parallel tuning of hexacopter altitude controller



(a) *(b)* *(c)* *(d)*

Fig. 6. Upper and 3D views for a surface approximation of $J = f(k_P, k_D)$: a)–b) Experiment no. 1, c)–d) Experiment no. 2

B. Experiment no. 1 Model-free auto-tuner (2 parameters, 1 UAV, low-pass filtering) based on modified zero-order golden-search algorithm

During the Experiment no.1, the performance of the altitude controller tuning for a single UAV has been verified in real-world conditions. Due to a limited duration of the experiment, as Li-Po has been used, a pair of bootstraps has been performed, i.e. to the maximum number of 56 iterations of the algorithm. As the previously-applied gain combinations have been skipped, the results presented in Table I and in Figs. 6a,b, 8, 9 have obtained after 45 iterations, corresponding to 540 sec of experiment duration.

The proposed tuning algorithm is very efficient in exploring the search space especially in the initial bootstrap phase. Statistical characteristics (see Table I) reveals that the second bootstrap improves the average in J_{avg} by 8.16 %, however, it does not directly imply that $J_{\min}(t)$ is achieved, as visible in bootstrap no. 1 at iteration no. 25.

C. Experiment no. 2 Model-free auto-tuner (2 parameters) based on zero-order golden-search method parallel version (2 drones, with averaging and low-pass filtering)

Having in mind that during the flight, estimated altitude is obtained on the basis of the measurements carried out in a noisy environment, and possible noise impact is reduced by the use of the low-pass filtration, the second, twin unit of the UAV is added in Experiment no. 2. Its main purpose has been to supply a parallel information to the tuning algorithm in order to average values of $J(t)$ between a pair of UAVs in every iteration. The results of the experiment have been presented in Table I and in Figs.: 6c,d, 7, 8 and 9.

The averaged sensory information between a pair of UAV combined with the use of the initial value of k_D from within the defined safe range resulted in:

- major improvement of tracking performance of the reference primitive after the proposed tuning method has been used (compare 1st and 45th iteration in Fig. 7), where no initial overshot is visible, and the reference altitude profile is tracked smoothly and fast,
- obtaining the lowest values of J_{end} (after the final iteration of the algorithms), and $J_{\min}$ (among all iterations of the considered algorithm), from experiments no. 1-3; The best combination of controller parameters allowed to achieve $J_{\min} = 0.8784$, obtained just in 25 iterations,
- obtaining decreased values for J_{avg} (see Fig. 8), as well as $J_{I\,\text{avg}}$ (past first bootstrap) and $J_{II\,\text{avg}}$ (past second bootstrap) – improved by 35.87 % and 29.65 %, respectively, with respect to the corresponding method

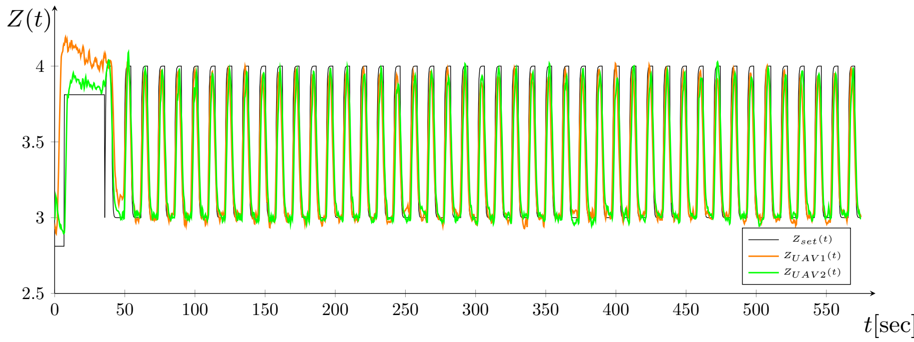


Fig. 7. Tracking of the reference altitude $Z_{set}t$ by a pair of UAVs ($Z_{UAV1}(t)$ and $Z_{UAV2}(t)$) in Experiment no. 2

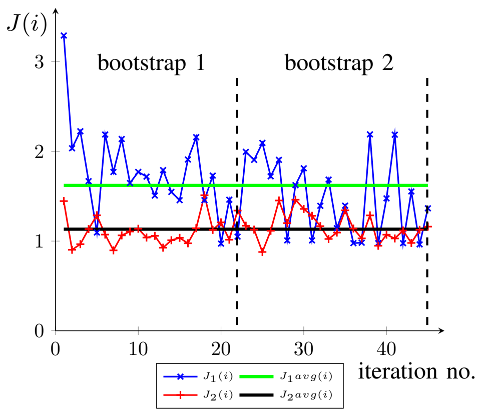


Fig. 8. $J(i)$ in consecutive steps i from Exp. no.1 (blue), no.2 (red)

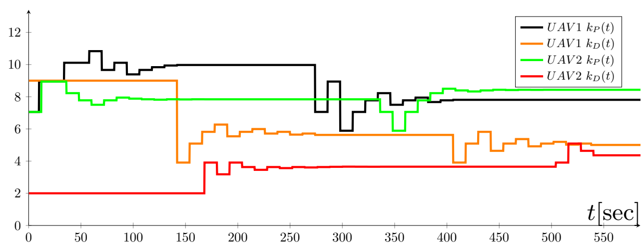


Fig. 9. Controller gains as a function of time, obtained from Exp. no. 1-2

for a single UAV unit (see Tab. I),

- from a statistical viewpoint, the second bootstrap corresponds to a 0.74 % increase in performance indices in comparison to the first bootstrap, though the final result ($J_{II\,\mathrm{avg}}$), past 2-nd bootstrap improves.

D. Experiment no. 3 Model-free auto-tuner (2 parameters) based on zero-order golden-search method parallel version (2 drones, with low-pass filtering, no averaging)

Promising results of Experiments no. 1 and no. 2 obtained in the very first bootstrap from the proposed methods, have given rise to a slight modification of a parallel tuning method. The expected features of the methods from the implementation viewpoint mentioned in the Introduction with respect to high precision and speed of tuning, resulted in the seek to find the compromise between classical optimality (understood as the stroll to find the best, globally optimal, controller parameters) and optimal tuning (understood as a visible improvement in tracking by a substantial improvement in performance index due to iterative-based optimal tuning in a limited time).

In Experiment no. 3, a single bootstrap is performed, composed of 14 parallel iterations when upper and lower bounds for the ranges of parameters have been obtained on the basis of the current iteration conducted on UAV1 and UAV2. This has led to the following conclusions based on Table I and results from Fig. 10:

- duration of Experiment no. 3 is reduced to 31.11 % of the duration of the previous experiment and has taken 168 sec; just after the 5-th iteration the minimal value of $J_{\min}$ has been achieved, with the corresponding value of J from 1-UAV tuning has been achieved after 25 iterations (see Table I), and the parallel tuning method with averaging the results after the 2-nd iteration!
- the gains k_P and k_D to converged to a minimum point different then in the two earlier cases (see Fig. 4),
- the first iteration of the algorithm (see Fig. 10) explains the need to use asynchronous communication between units and ground station (7 sec delay is observed),
- the reason for the behaviour observed at the 9-th iteration may only be found in the future research by conducting a number of experiments taking safety features and fault-tolerant control into consideration.

V. CONCLUSION

A pair of optimal iterative-based fixed-parameter controller gains with a small number of parameters presented in the paper allows one to conduct efficient, parallel tuning of altitude controllers of a pair of twin UAV units during in-flight outdoor conditions. By using a pair of twin autonomous units in a centralized architecture with the optimizing unit, enables to perform automatic and iterative improvement of the gains of their controllers, on the basis of sole altitude measurements, without the use of any dynamical model of the UAVs. The latter feature is currently highly-demanded in applications to transporting tasks, where it is necessary to tune controller parameters in various environmental conditions, such as in changing of a gross UAV weight, in order to achieve flight safety by appropriate UAV stabilization.

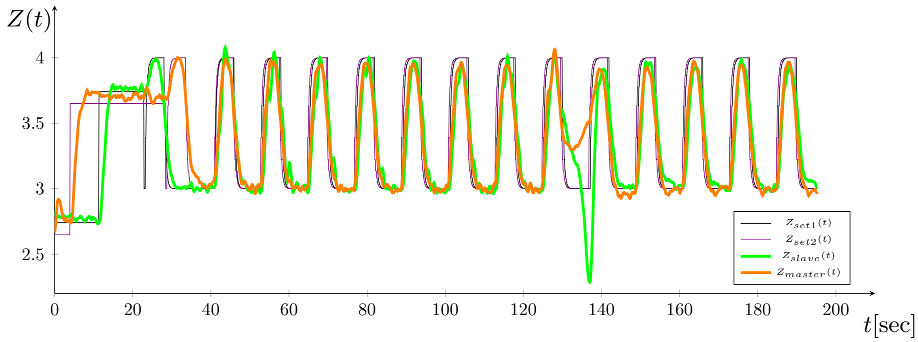


Fig. 10. Tracking of the reference altitude $Z_{set}t$ by a group of UAVs ($Z_{UAV1}(t)$ and $Z_{UAV2}(t)$) in Experiment no. 3

The obtained results at this point, form a path to further research where a complete verification of the efficiency of the proposed methods is planned, as well as the methods adapted from [29] for remaining position and orientation controllers, including the experiments with lifting load from the ground. In this context, it is interesting to compare the results with one and only currently available commercial solution for PIDs auto-tuning for roll, pitch and yaw angles of one UAV (CleanFlight G-Tune) for Naze/Multiwii platforms.

The obtained results (see last iteration in Fig. 10) also imply that it is necessary to introduce or to consider other methods of tracking performance evaluation during the flight, e.g. by taking energy consumption related to control signal value, despite taking tracking error information only.

ACKNOWLEDGMENT

The authors with to acknowledge a great deal of help from the side of Vít Krátký, Petr Štibinger and Pavel Petráček in conducting experiments.

REFERENCES

[1] K. Valavanis and G. V. (Eds.), *Handbook of Unmanned Aerial Vehicles*. Springer Publishing Company, 2015.

[2] R. Beard and T. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.

[3] E. Ebeid, M. Skriver, K. Terkildsen, K. Jensen, and U. Schultz, "A survey of open-source uav flight controllers and flight simulators," *Microprocessors and Microsystems*, vol. 61, Sep. 2018.

[4] M. Koppany and M. Busoniu, "Vision and control for uavs, a survey of general methods and of inexpensive platforms for infrastructure inspection," *Sensors*, vol. 15, no. 7, pp. 14 887–14 916, 2015.

[5] T. Espinoza, A. Dzul, F. Cortez-Martinez, and W. Giernacki, "Real-time implementation and flight tests using linear and nonlinear controllers for a fixed-wing mav," *International Journal of Control, Automation & Systems*, vol. 16, no. 1, pp. 392–396, 2018.

[6] D. Tesch, D. Eckhard, and W. Guarienti, "Pitch and roll control of a quadcopter using cascade iterative feedback tuning," in *IFAC-PapersOnLine*, pp. 30–35, 2016.

[7] A. Catena, C. Melita, and G. Muscato, "Automatic tuning architecture for the navigation control loops of unmanned aerial vehicle," *Journal of Intelligent and Robotic Systems*, vol. 73, pp. 413–427, 2014.

[8] A. Julkananusart and et al., "Quadrotor tuning for attitude control based on pid controller using fictitious reference iterative tuning (frit)," in *Proc. of the IEEE 6th IC-ICTES conference*, pp. 1–5, 2015.

[9] J. Kim, S. Wilkerson, and S. Gadsden, "Comparison of gradient methods for gain tuning of a pd controller applied on a quadrotor system," in *Proc. of the SPIE conference*, vol. 9837, 2016.

[10] J. Junell, T. Mannucci, Y. Zhou, and E. van Kampen, "Self-tuning gains of a quadrotor using a simple model for policy gradient reinforcement learning," in *Proc. of the AIAA guidance, navigation, and control conference*, pp. 1–15, 2016.

[11] W. Giernacki, T. Espinoza-Fraire, and P. Kozierski, "Cuttlefish optimization algorithm in autotuning of altitude controller of unmanned aerial vehicle," *ROBOT'17 Iberian Robotics conference*, pp. 841–852.

[12] F. Berkenkamp, A. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes," in *2016 IEEE ICRA Conference*, pp. 491–496, 2016.

[13] F. Goodarzi, "Self-tuning geometric control for a quadrotor uav based on lyapunov stability analysis," *International Journal of Robotics and Automation*, vol. 5, no. 3, pp. 1–15, 2016.

[14] L. Ravanbod and D. Noll, "Gain-scheduled two-loop autopilot for an aircraft," *Journal of Dynamical Systems, Measurement, and Control*, vol. 136, no. 4, pp. 1–3, 2014.

[15] Y. Alothman and B. Gu, "Quadrotor transporting cable-suspended load using iterative linear quadratic regulator (ilqr) optimal control," in *Proc. of the IEEE 8th Comp. Science & Electr. Engin.*, pp. 168–173.

[16] W. Giernacki and J. Goslinski, "Coaxial quadrotor - from the most important physical aspects to a useful mathematical model for control purposes," *in review*, 2020.

[17] L. Rios and N. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, pp. 1247–1293, 2013.

[18] D. Horla, W. Giernacki, T. Baca, V. Spurny, and M. Saska, "Al-tune: In-flight optimal tuning of an altitude controller for uavs," *in review*.

[19] J. Su, J. Wu, P. Cheng, and J. Chen, "Autonomous vehicle control through the dynamics and controller learning," *IEEE Transactions on Vehicular Technology*, DOI 10.1109/TVT.2018.2819806, 2018.

[20] F. Berkenkamp, A. Krause, and A. Schoellig, "Bayesian optimization with safety constraints," *arXiv*, DOI arXiv:1602.04450v2, 2018.

[21] A. Alametalu, S. Kaynama, J. Fisac, M. Zeilinger, J. Gillula, and C. Tomlin, "Reachability-based safe learning with gaussian processes," in *53rd IEEE CDC Conference*, pp. 1424–1431, 2014.

[22] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with gaussian processes," in *Proc. of the International Conference on Machine Learning*, pp. 997–1005, 2015.

[23] E. Chong and S. Zak, *An Introduction to Optimization. Second Edition*. John Wiley & Sons, 2001.

[24] T. Lee, M. Leok, and N. McClamroch, "Nonlinear robust tracking control of a quadrotor uav on se(3)," *Asian Journal of Control*, vol. 15, no. 2, pp. 391–408, 2013.

[25] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2520–2525, 2011.

[26] G. Loianno and at al., "Localization, grasping, and transportation of magnetic objects by a team of mavs in challenging desert like environments," *RA-L Journal*, vol. 3, no. 3, pp. 1576–1583, 2018.

[27] T. Baca and at al., "Autonomous landing on a moving vehicle with an unmanned aerial vehicle," *Journal of Field Robotics*, vol. 36, no. 5, pp. 874–891, 2019.

[28] V. Spurny and at al., "Cooperative autonomous search, grasping and delivering in a treasure hunt scenario by a team of uav," *Journal of Field Robotics*, vol. 36, no. 1, pp. 125–148, 2019.

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