

Control of Unknown Quadrotors from a Single Throw

Blaha, Till M.; Smeur, Ewoud J.J.; Remes, Bart D.W.

DOI

[10.1109/IROS58592.2024.10801514](https://doi.org/10.1109/IROS58592.2024.10801514)

Publication date

2024

Document Version

Final published version

Published in

2024 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024

Citation (APA)

Blaha, T. M., Smeur, E. J. J., & Remes, B. D. W. (2024). Control of Unknown Quadrotors from a Single Throw. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024* (pp. 10350-10355). (IEEE International Conference on Intelligent Robots and Systems). IEEE.
<https://doi.org/10.1109/IROS58592.2024.10801514>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Control of Unknown Quadrotors from a Single Throw

Till M. Blaha¹, Ewoud J.J. Smeur and Bart D.W. Remes

Abstract—This paper presents a method to recover quadrotor Unmanned Air Vehicles (UAVs) from a throw, when no control parameters are known before the throw. We leverage the availability of high-frequency rotor speed feedback available in racing drone hardware and software to find control effectiveness values and fit a motor model using recursive least squares (RLS) estimation. Furthermore, we propose an excitation sequence that provides large actuation commands while guaranteeing to stay within gyroscope sensing limits. After 450ms of excitation, an Incremental Nonlinear Dynamic Inversion (INDI) attitude controller uses the 52 fitted parameters to arrest rotational motion and recover an upright attitude. Finally, a Nonlinear Dynamic Inversion (NDI) position controller drives the craft to a position setpoint. The proposed algorithm runs efficiently on microcontrollers found in common UAV flight controllers, and was shown to recover an agile quadrotor every time in live experiments with as low as 3.5m throw height, demonstrating robustness against initial rotations and noise. We also demonstrate control of randomized quadrotors in simulated throws, where the parameter fitting Root-Mean-Square (RMS) error is typically within 10% of the true value.

I. INTRODUCTION

The number of different UAVs and their applications are rapidly increasing, and so are specific drone designs that fulfill various niches. Traditionally, the development of control systems for drones requires system modeling and identification, control design and gain tuning, which takes valuable time and resources. Yet, changes to the hardware configuration again require analysis, re-tuning and testing. Ideally, the controller and its parameters would be identified automatically, without the need for a simulation model or any manual analysis, design and testing. Learning the control of a fixed wing aircraft model after a live drop from a balloon was demonstrated in [1] with a global adaptive control approach. To extend the current state of the art, the purpose of this study is to provide a method to control fixed-motor multirotor UAV using self-collected flight data, without prior knowledge of its flight characteristics.

Reinforcement learning has been used to learn a quadrotor control policy from scratch [2]. This approach is applicable when a model of the quadrotor is already known, such that the learning can happen in simulation. The number of system interactions required to learn a satisfactory policy was $\approx 300,000$ (≈ 1 h of simulated flight time). This is still far beyond what is safely and conveniently achievable without a simulation.

*This work was not supported by any organization

¹All authors are with the Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, Netherlands
Correspondence: t.m.blaha@tudelft.nl

Data: 10.4121/0530be90-cc6c-4029-9774-670657882906

Code: github.com/tudelft/indiflightSupport/tree/iros_imav_2024

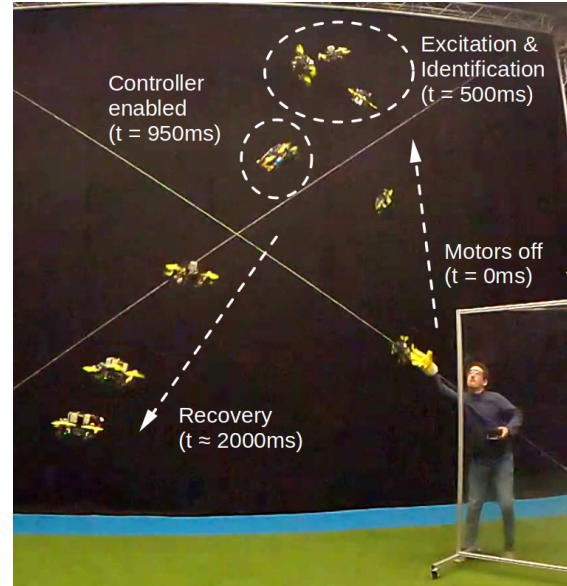


Fig. 1. Composite image: The quadrotor learns to fly before it would hit the ground. Moving picture at <https://youtu.be/CLFPXcpzA14>

White-box adaptive multirotor controllers can be based on Model-Reference Adaptive Control (MRAC), which have been shown to fly with an unknown swinging payload [3]. In this approach, the controller parameters are adapted directly to match a desired system response. Given that quadrotors' performance limits may vary greatly, this reference model itself would likely also be adaptive. Additionally, most MRAC formulations assume knowledge of the signs in the control effectiveness.

The problem therefor appears to lend itself better to controllers based on the physical properties and capabilities of the UAV. Such controllers are also available and are often based on INDI. However, they are usually designed with the purpose to adapt to changes in the control effectiveness, after starting with a reasonable initial estimate [4], [5], or to recover from in-flight failures [6].

Insufficient excitation can have detrimental effect on estimation of the effectiveness for multirotors [5] and, since no prior knowledge is available for the problem at hand, a model-independent excitation method is required. Furthermore, faster adaptation than achieved with Least Mean Squares (LMS) in [4] is needed, as well as to also identify an actuator model and not only effectiveness [4], [5].

In this work, we develop an algorithm that can learn to fly a multirotor directly from a single maneuver without prior platform knowledge, except the location and orientation of its Inertial Measurement Unit (IMU), and that four upwards-

pointing rotors are present. We present (1) an attitude and position controller framework for multirotors, incorporating a novel linearization of the effects of rotor acceleration; (2) an excitation procedure that guarantees remaining within the gyroscope saturation limits; and (3) a Recursive Least Squares (RLS) estimator for the control effectiveness and motor model parameters. It is demonstrated with parametric simulations and real-life throws, that the resulting algorithm reliably stabilizes and controls quadrotors after a throw of a few meters height, with unknown initial parameters (see figure 1). The algorithm computes entirely on onboard microprocessor hardware without simulation.

II. METHODOLOGY

This section describes control loops, parameter identification, and setup of the simulations and live experiments.

The quadrotor system is modeled as a control affine system perturbed by gravity, the thrust generated by the four rotors, and the reaction torque from actuating the rotors.

A. Controller

The flight controls used in the work can be broken down in 3 cascaded loops; (a) the “inner loop” controlling angular accelerations and specific forces, (b) the attitude controller, and (c) the position controller. They are described in detail in the following sections.

a) Inner Loop: PID control is often used on the angular rate error $\Omega_{\text{ref}} - \Omega$. This generates torque commands which are then allocated to the motors in an open-loop fashion (“mixed”). With NDI, the relationship between motor states and angular acceleration may be modeled more accurately and is then inverted to obtain a relationship between so-called pseudo-controls ν and motor commands that linearises the dynamics with respect to the pseudo-controls. The authors of [7] show how INDI can replace a part of the system model with online measurements. This effectively closes the loop for angular acceleration and specific forces, and enables more responsive disturbance rejection and easier tuning, if the remaining model parameters (control effectiveness) are accurate. For an in-depth derivation, refer to [7], [4].

For flying vehicles, a common choice of pseudo-controls are the specific forces and angular accelerations $\nu = (f_x \ f_y \ f_z \ \dot{p} \ \dot{q} \ \dot{r})^T$, because a reference ν_r can be generated conveniently from attitude errors and body rotational velocity errors using linear error feedback. For multirotors with rotor axes aligned with the local z-axis, $f_x = f_y = 0$. To keep this work applicable for vehicles without this property or with different axes definitions however, we will only make this assumption when generating ν_r .

An incremental model for $\Delta \nu$ has to capture the effects of the thrust increments $\Delta T \in \mathbb{R}^4$ produced by each of the motors, but also the rate of change of motor velocity $\dot{\omega}$, which exerts reaction torques on the vehicle. This results in

$$\Delta \nu = \nu - \nu_0 = B_1 \Delta T + B_2 \Delta \dot{\omega}, \quad (1)$$

where ν_0 is a measurement of ν from onboard sensors. B_1 and B_2 are called “control effectiveness” matrices and arise from the geometry and actuator parameters of the vehicle.

Neither T , ω nor $\dot{\omega}$ can be directly commanded with common Electronic Speed Control (ESC) hardware. In the following, it is shown that the choice of thrust-normalized $u \triangleq T_s/T_{\text{max}}$ for the control variables enables a convenient linearized formulation suitable for control and identification. Only the term related to $\dot{\omega}$ is approximated, which only affects the yaw axis of multirotors. Subscript s stands for the steady-state values after cessation of actuator dynamics.

u can be mapped to ESC input $\delta \in [0, 1]$ by inverting a function $u = f(\delta)$, which in our work is assumed to be

$$u = \left[\kappa \delta + (1 - \kappa) \sqrt{\delta} \right]^2, \quad (2)$$

where $\kappa \geq 0$ is an unknown coefficient. This matches motor test bench data and enables a parameter-linear model for identification, as shown in II-B.0.b.

Assume that the motor dynamics are described by a first-order lag, i.e. $\tau \dot{\omega} = \omega_s - \omega$ for some time constant $\tau > 0$ and steady-state motor speed ω_s . The thrust produced by each rotor can be modeled with $T = k\omega^2$ [4], where k is the propeller thrust constant. Combining these two models yields

$$\tau \dot{\omega} + \omega = \omega_s = \sqrt{\frac{T_s}{k}} = \sqrt{\frac{T + \Delta T_s}{k}} = \sqrt{\frac{T}{k}} \sqrt{1 + \frac{\Delta T_s}{T}}. \quad (3)$$

For small steady-state thrust increments ΔT_s , the square root can be eliminated using the binomial approximation. Additionally, substituting $\omega = \sqrt{T/k}$ and $\Delta u = \Delta T_s/T_{\text{max}} = \Delta T_s/k\omega_{\text{max}}^2$ gives

$$\tau \dot{\omega} + \omega \approx \sqrt{\frac{T}{k}} \left(1 + \frac{\Delta T_s}{2T} \right) = \omega \left(1 + \frac{\Delta T_s}{2T} \right) \quad (4)$$

$$\tau \dot{\omega} \approx \frac{\omega}{2T} \Delta T_s = \frac{1}{2k\omega} \Delta T_s = \frac{\omega_{\text{max}}^2}{2\omega} \Delta u. \quad (5)$$

With these approximations, we can write a version of (1) that is linear in the non-dimensional thrust increment Δu

$$\nu - \nu_0 = B_1 \Delta T_s + B_2 (\dot{\omega} - \dot{\omega}_0) \quad (6)$$

$$\nu - \nu_0 + B_2 \dot{\omega}_0 \approx \left(B_1 k \omega_{\text{max}}^2 + \frac{B_2 \omega_{\text{max}}^2}{2\omega_0 \tau} \right) \Delta u. \quad (7)$$

Given a reference $\nu = \nu_r$, this can be solved using e.g. the pseudo-inverse or with active-set methods, if bounds on u must be satisfied [8]. If k is known, then u_0 may be calculated from a measurement of ω . Otherwise, emulating the actuator dynamics $U_0 = \frac{U}{\tau s + 1}$ has shown success, even though this lag actually acts on the rotational speed and not the thrust.

The subscript 0 need not imply the values of the previous time-step. To combat the effects of noise, amplified by the differentiation of ω and Ω , we can replace each $\dot{\omega}_0$, u_0 , and ν_0 with low-pass filtered versions. A suitable choice for $H(s)$ is a second-order Butterworth [9], which has a steeper cutoff compared to a first-order filter, at the expense of larger phase-shift. However, the larger phase delay does not affect the system dynamics since all values are filtered with the same filter, preserving the validity of (1). The filtering however

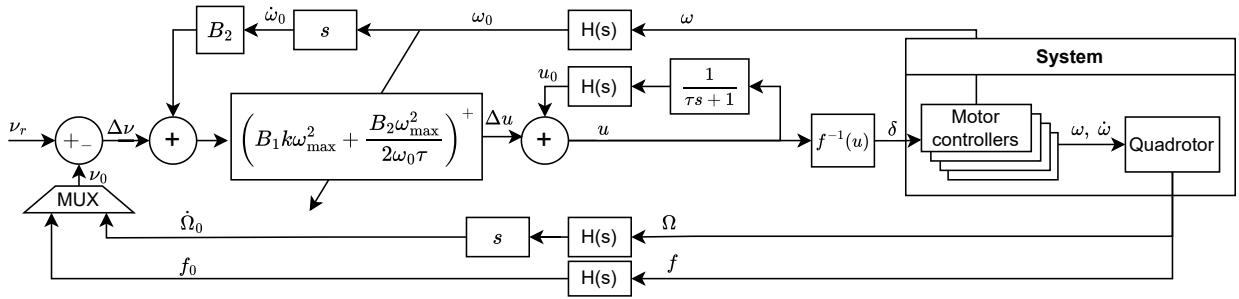


Fig. 2. INDI inner loop similar to [4] with linearized motor acceleration dynamics

does decrease disturbance rejection performance [4]. A cutoff of $f_c = 15\text{Hz}$ yielded good performance in our experiments, but lower values should be used if structural vibrations are a concern, or if smoother motor commands are desired.

Figure 2 shows a diagram of the inner loop.

b) Attitude control: An attitude setpoint given by unit quaternion q_r may be tracked by calculating an error quaternion in a body-fixed frame $q_e^B = q^{-1}q_r$ where q is the current attitude. Reference angular body velocities can then be generated with

$$\Omega_r = 2 \arccos((q_e^B)_w) (\hat{n} \circ A), \quad (8)$$

where \hat{n} is the normalized vector part of q_e^B , subscript w denotes the scalar part of q_e^B , A is a vector of gains, and \circ denotes element-wise multiplication. The last 3 elements of ν_r in (7) are set to $D \circ (\Omega_r - \Omega)$, where Ω is the measured angular body velocity and D another vector of 3 gains.

c) Position control: Attitude and thrust are related to inertial acceleration through:

$$a^I = R_B^I f^B + G^I, \quad (9)$$

in the absence of aerodynamic effects (including wind). Here, $f^B = (0 \ 0 \ f_z^B)^T$ are the specific forces generated by the motors in a body-fixed frame, G^I is the gravity vector, and R_B^I the rotation from the body-fixed to the inertial frame. PID control is used to generate inertial acceleration references a_r^I from position and velocity estimates.

If a reference yaw-angle Ψ is prescribed, then (9) can be solved explicitly for $R_B^{I,*}$ and $f_z^{B,*}$, yielding an NDI law. $R_B^{I,*}$ is the setpoint for the attitude controller, and $\nu_{r,1:3} = (0 \ 0 \ f_z^{B,*})^T$.

d) Gain tuning: With perfect modeling of $B_1 k \omega_{\max}^2$ and B_2 , and in the absence of outside disturbances, it is shown in [4] that the entire inner loop collapses to a first-order transfer function coincident with the actuator model. The inner loop angular rate gains D may then be tuned by pole-placement based on a known actuator time constant τ and a prescribed damping ratio ζ_D . By approximating the resulting 2nd order system by a 1st order linear lag $D/(s + D)$, the pole-placement technique can be successively applied to yield gains for attitude, velocity and position. We choose $\zeta_D = 0.8$, $\zeta_A = 0.7$, $\zeta_V = 0.7$, $\zeta_P = 0.9$.

B. Model Identification

To summarize, the controller depends on the system parameters $B_1 k$, B_2 , ω_{\max}^2 , τ and the mappings $u = f(\delta)$. This section shows a method to identify them from online flight data using recursive least squares estimation (RLS).

a) RLS: For parameter-linear models $Y = X^T \Theta$, with scalar Y , regressor column vector X , and parameter column vector Θ , a common recursive estimator is RLS [10]:

$$e_k \leftarrow Y_k - X_k^T \Theta_{k-1} \quad (10)$$

$$K_k \leftarrow P_{k-1} X_k (\lambda + X_k^T P_{k-1} X_k)^{-1} \quad (11)$$

$$\Theta_k \leftarrow \Theta_{k-1} + K_k e_k \quad (12)$$

$$P_k \leftarrow \lambda^{-1} (P_{k-1} - K_k X_k^T P_{k-1}), \quad (13)$$

where $P_0 = I \cdot 10^2$, $\Theta_0 = 0$.

Since our RLS filters described in the subsequent sections have to compute on single precision floating-point hardware, we avoid the accumulation of numerical errors by pre-scaling the regressors and outputs to be in the order of unity. The gain K_k is decreased, if the subtraction in (13) would otherwise accrue errors.

Finally, choosing $\lambda = \exp(P_s/0.2)$, where P_s is the sampling period, ensures that the error observations e_k decay by 63% after 0.2 seconds, which gave satisfactory performance in the short identification time. To avoid divergence of P_k , its entries are limited to 10^{10} , but future research should replace this by an adaptive forgetting procedure instead.

b) Motor model: Typical ESCs drive the rotors to $\omega_{\text{idle}} > 0$, even when $\delta = 0$. While the thrust produced at that speed has been neglected in (2), the unknown offset needs to be included when working with rotor speed data. With $\omega = \sqrt{T/k}$, $u \triangleq T_s/T_{\max}$:

$$\omega_s = \omega_{\max} \sqrt{u} + \omega_{\text{idle}} = \omega_{\max} (\kappa \delta + (1 - \kappa) \sqrt{\delta}) + \omega_{\text{idle}}. \quad (14)$$

ω_{\max} depends on battery state of charge and aerodynamic flow conditions, but this is neglected as ω_{\max} will be estimated online.

To turn this into a parameter-linear formulation, the parameters $a, b > 0$ are introduced and defined by the transformations $\omega_{\max} = a + b$ and $\kappa = \frac{a}{a+b}$. Combined with the first-order motor dynamics model $\tau \dot{\omega} = \omega_s - \omega$ used above,

this yields

$$\omega = a\delta + b\sqrt{\delta} + \omega_{\text{idle}} - \tau\dot{\omega}. \quad (15)$$

c) *Effectiveness B_1 and B_2* : To identify B_1 and B_2 , we again turn to (1) and notice that the only non-measurable quantity is ΔT . If we write the propeller thrust model $T = k\omega^2$ in linearized differential form $\Delta T = 2k \cdot \omega \Delta\omega$, we obtain

$$v - v_0 = \Delta \begin{pmatrix} f_x & f_y & f_z & \dot{p} & \dot{q} & \dot{r} \end{pmatrix}^T = B_1 k (2\omega \Delta\omega) + B_2 \Delta\dot{\omega}, \quad (16)$$

where only $B_1 k$ and B_2 are unknown and all other quantities can be trivially derived from measurements of specific force f , body rotation rate Ω and motor rotation rate ω . Modern flight control hardware and software is available that provides even the motor rotation rate ω and $\dot{\omega}$ at sufficiently high rates. Again, we can low-pass filter each quantity, in this case chosen as a 20Hz second-order Butterworth low-pass.

The first and last 3 rows in (16) each form a linear estimation problem in 8 parameters. However, since there is no conceivable way in which $\Delta\dot{\omega}$ gives rise to forces, its term is neglected for the first 3 rows, such that they only require 4 parameters. The problem appears to require 6 separate RLS filters, but since the first and last 3 rows share the same regressors, the expensive operations (11) and (13) only have to be computed once for each group of filters.

d) *Excitation*: For the parameters above to converge using RLS, sufficient excitation must be present. A unique challenge with multirotors is that even a short burst (< 100ms) of extreme motor input can induce rotations that exceed the $\pm 2000^\circ/\text{s}$ measurement range of most MEMS gyroscopes. This has to be avoided, as attitude estimation would be lost making recovery impossible. On the other extreme, too conservative excitation may be drowned in measurement noise. Our approach aims at inducing large accelerations on the vehicle without saturating the gyroscope, by providing an adaptive sequence of excitations to the motors. Feedback control is enabled only after excitation (and identification) is complete.

Each motor is successively excited with two step inputs followed by a decreasing ramp. This ensures that many different levels of δ , and the largest possible $\dot{\omega}$ are provided to the system. On entry to the first step of each motor, the gyroscope reading Ω_e and margin $\Delta\Omega$ to the sensor limits is recorded. The excitation is aborted if a gyroscope reading exceeds $\Omega_e + \Delta\Omega/n_l$ where n_l is the number of motors left to excite after the current motor. This is illustrated in figure 3. It could be argued that this is too conservative, since if a quadrotor is able to hover, then 2 of its motors must provide a rotation that de-saturates the sensor, so $0.5n_l$ in the denominator should be sufficient. However, if we assume that rotor acceleration and deceleration have the same dynamics, then in the worst case, the thrust from a rotor imparts as much angular impulse on spin-up as it does while spinning down after the excitation is aborted. This effectively halves the permissible margin $\Delta\Omega$.

It seems generally accepted that orthogonal sine-inputs are more sample efficient inputs for the identification of

MIMO systems [11]. Furthermore, it is common to mix the control signal with the generated signal [1]. However, without prior knowledge of the vehicles capabilities, it seems challenging to guarantee the safety of these inputs while ensuring sufficient excitation.

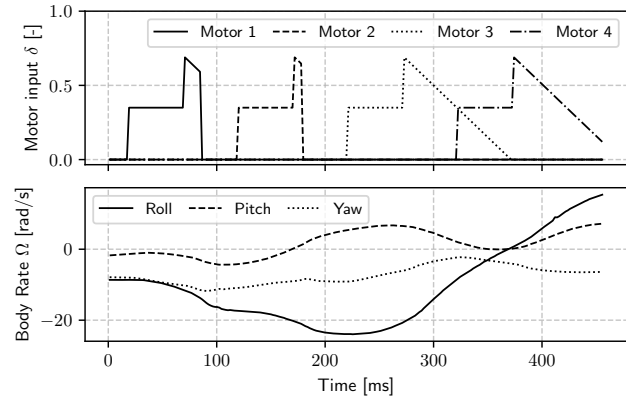


Fig. 3. Excitation after detection of throwing. In this trial, the commands for motors 1 and 2 are truncated to avoid sensor saturation in the roll axis.

C. Experimental Setup

A 3-inch-propeller quadrotor UAV, sporting an STM32H743 microcontroller and TDK InvenSense ICM-42688-P IMU was used for experimental verification. The identification and control algorithm above was implemented onboard at 2kHz sample rate in the open-source software package Betaflight¹. Its focus on performance and computational efficiency, and especially the availability of motor rotation rate feedback at equally high sample rate, make it ideal to research low-level control implementations. Our adaptations were so significant that a new fork INDIfight² was created.

The UAV was instrumented with reflective markers to enable optical position tracking. The positioning data was sent to the UAV, but only used for position control; it does not influence identification. Aside from the positioning triangulation and transmitting that data to the drone, all calculations are performed onboard. The internal attitude estimation remains the Mahony filter provided by Betaflight. Heading, velocity and position use simple constant gain observers to fuse inertial measurements with optical position and velocity data.

a) *Conditions*: The nominal INDI controller parameters were calculated with bench tests of propeller and motors, and measurements of geometry and mass properties. For controlled experiments, this controller took the quadrotor from ground level onto a 3.5-4m high parabolic trajectory and induced a pseudo-random rotation between 0 and 10rad/s. Using the known geometry of the quadrotor, and after performing bench tests with a controller-motor-propeller combination, the parameters were calculated. The engines are then spooled down for 250ms, and the excitation

¹<https://betaflight.com/>

²<https://github.com/tudelift/indiflight>

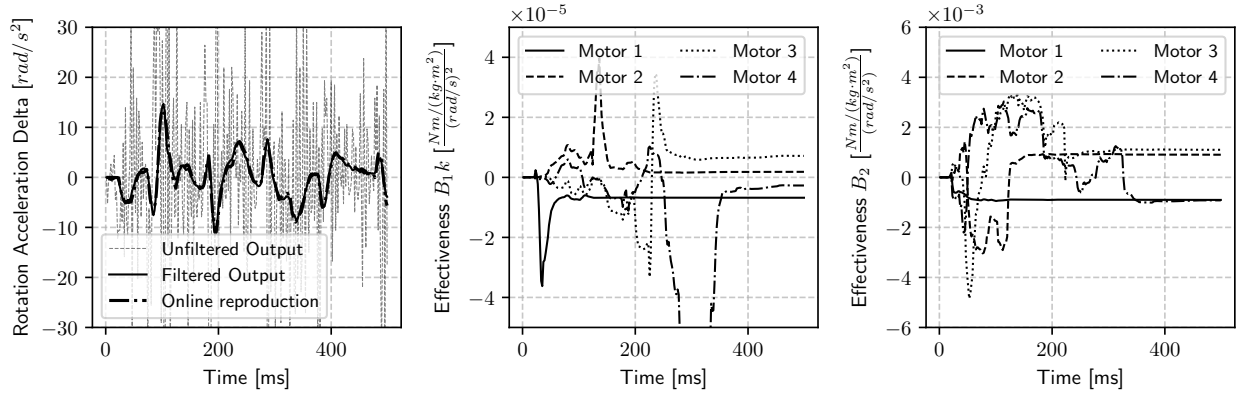


Fig. 4. Time evolution of the control effectiveness around the yaw axis.

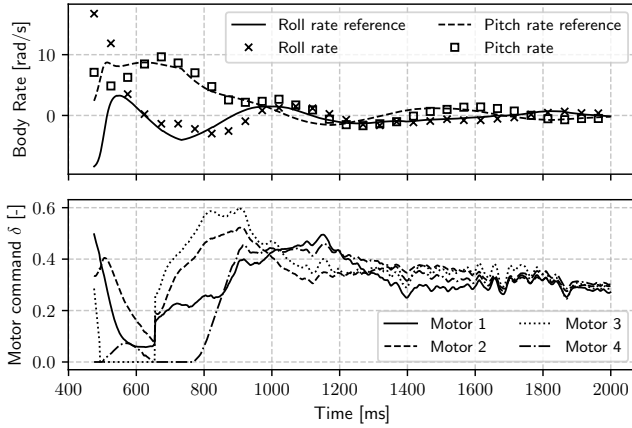


Fig. 5. Angular rate tracking and motor commands during recovery

is triggered. After its completion 450ms later, the fitting is done and switchover to the identified parameter-set occurs.

D. Simulation Setup

1000 asymmetrical multirotor crafts were generated and simulated in randomized throwing conditions of about 4 height. For each rotor of each craft, we varied the following parameters: motor assignment; distance from CoG [0.05, 0.15]*m*; top-view angle of the CoG-to-rotor line from a nominal square quadrotor [−30°, +30°]; non-linearity $\kappa \in [0.25, 0.75]$; time constant $\tau \in [15, 35]$ *ms*; drag-moment coefficient [0.01, 0.02]. If the parameters resulted in a craft that is physically unable to hover, it was regenerated.

The simulation is linked against the same INDIfight software as used in the experiments, and can be reproduced using Docker³. All data files are also available⁴.

III. RESULTS

A. Experiments

A total of 57 controlled launches were performed, and during all of them the parameters were identified well enough

to stabilize the UAVs attitude. Due to an oversight, suitable log files were only obtained in the last 20 runs.

Figure 4 shows the time evolution of the 8 parameters in the model for $\Delta\dot{r}$ during an experiment run (the 4 parameters in the center plot relate to each motor velocity and 4 parameters in the right plot relate to motor acceleration). The low-pass filtering is effective in reducing noise in the measured targets shown in the left plot. From the center and right plot it can be seen that for all motors but motor 1 the parameters initially do not converge. This is due to the presence of only idle rotation speed before the step input. However, once excitation is provided, the parameters immediately converge, and reproduction of the rotation acceleration measurement is unaffected by the initial uncertainty. Nonetheless, control based on these parameters would be impossible.

Despite initial rate errors in excess of 20rad/s, the inner loop tracks the rate reference after less than 200ms, with relatively smooth motor commands, as seen in figure 5. Slight oscillations are present between 1200 and 2000ms, which may indicate that the choices of ζ have been too low.

Finally, table I shows the means and standard deviations of the fitted parameters over the last 20 runs. They are compared to the calculated “reference” value from bench tests, geometry and mass properties measurements. Note, that these calculations also have their own uncertainty and should not be seen as a ground truth. In particular, it is reasonable that motors 1 and 3 (the rear motors) have higher yaw effectiveness $B_{1,r}$ and higher thrust effectiveness $B_{1,z}$ than motors 2 and 4, because the structure that the motors are mounted on is much wider in the front of our quadrotor. This can reduce the downwards flow velocity and swirl.

Any effects from non-zero inflow are neither captured on the static test bench, nor in the simulation. It is therefore possible, that the differences in the motor model parameters in the experiments are due to these effects occurring in-flight. Consequently, the motor model (15) and likely the thrust model $T = k\omega^2$ do not hold globally.

B. Simulations

All 1000 vehicles were successfully stabilized despite errors in the parameter estimates. Table II shows that the root-

³github.com/tudelft/indiflightSupport/tree/iros_imav_2024

⁴[10.4121/0530be90-cc6c-4029-9774-670657882906](https://doi.org/10.4121/0530be90-cc6c-4029-9774-670657882906)

TABLE I

MEAN AND STANDARD DEVIATION FOR B_1 , B_2 AND THE MOTOR MODEL DURING 20 LAUNCHES WITH DIFFERENT INITIAL CONDITIONS.

	Motor 1			Motor 2			Motor 3			Motor 4		
	Reference	Mean	Std	Reference	Mean	Std	Reference	Mean	Std	Reference	Mean	Std
$B_{1,x} \cdot k \cdot 10^6$	0.000	0.056	0.066	0.000	-0.101	0.034	0.000	0.068	0.064	0.000	-0.082	0.046
$B_{1,y} \cdot k \cdot 10^6$	0.000	-0.089	0.046	0.000	0.012	0.039	0.000	0.007	0.049	0.000	-0.019	0.042
$B_{1,z} \cdot k \cdot 10^6$	-0.621	-0.658	0.060	-0.621	-0.592	0.030	-0.621	-0.792	0.069	-0.621	-0.684	0.043
$B_{1,p} \cdot k \cdot 10^6$	-23.440	-29.551	3.194	-23.440	-30.661	2.360	23.440	29.904	2.952	23.440	27.441	2.287
$B_{1,q} \cdot k \cdot 10^6$	-15.718	-18.992	2.040	15.718	21.283	1.498	-15.718	-14.013	1.136	15.718	18.492	1.277
$B_{1,r} \cdot k \cdot 10^6$	-2.989	-6.587	0.673	2.989	2.319	0.480	2.989	6.669	0.792	-2.989	-2.158	0.303
$B_{2,p} \cdot 10^3$	0.000	-0.002	0.079	-0.000	0.185	0.038	-0.000	0.168	0.079	0.000	-0.040	0.074
$B_{2,q} \cdot 10^3$	0.000	0.066	0.049	-0.000	-0.030	0.057	-0.000	0.070	0.125	0.000	0.057	0.084
$B_{2,r} \cdot 10^3$	-1.011	-0.915	0.021	1.011	0.945	0.035	1.011	1.061	0.052	-1.011	-0.961	0.054
ω_{\max}	4113	5253	248.2	4113	5278	199.7	4113	4812	58.3	4113	4919	81.6
κ	0.460	1.118	0.088	0.460	1.184	0.079	0.460	0.994	0.008	0.460	1.000	0.009
ω_{idle}	450	516.7	7.306	450	512.7	8.100	450	505.8	7.194	450	503.7	7.271
τ [ms]	20.00	25.91	0.598	20.00	26.21	0.793	20.00	26.64	0.425	20.00	26.61	0.294

TABLE II

PARAMETER ERROR RMS OVER THE 1000 SIMULATED RUNS.

	Ground truths mean-absolute	RMS error			
		Motor 1	2	3	4
$B_{1,x} \cdot k \cdot 10^6$	0.000	0.037	0.036	0.032	0.037
$B_{1,y} \cdot k \cdot 10^6$	0.000	0.038	0.034	0.032	0.036
$B_{1,z} \cdot k \cdot 10^6$	0.463	0.040	0.036	0.033	0.039
$B_{1,p} \cdot k \cdot 10^6$	18.301	1.147	1.515	1.597	1.379
$B_{1,q} \cdot k \cdot 10^6$	17.538	1.107	1.518	1.535	1.424
$B_{1,r} \cdot k \cdot 10^6$	4.173	0.472	0.436	0.395	0.395
$B_{2,p} \cdot 10^3$	0.000	0.055	0.068	0.085	0.114
$B_{2,q} \cdot 10^3$	0.000	0.055	0.064	0.089	0.116
$B_{2,r} \cdot 10^3$	0.667	0.030	0.032	0.039	0.052
ω_{\max}	4900	390.7	275.7	148.7	47.46
κ	0.499	0.119	0.105	0.114	0.101
ω_{idle}	0.000	2.824	5.568	11.28	4.660
τ [ms]	25.04	0.627	0.689	0.768	1.080

mean-square (RMS) errors between the true and estimated parameters over the runs are typically below 10%. The motor numbers 1 to 4 correspond to the order in which they have been excited. The “nominal” column gives context to interpret the RMS values, and is the mean of the absolute parameter value over all the true generated crafts.

IV. CONCLUSIONS AND FUTURE WORK

The simulation and real-world experiments show that the classical recursive parameter estimation method described, combined with a short and simple excitation sequence lasting only 450ms, is enough to recover agile quadrotor UAVs when thrown to heights as low as 3.5m. It identifies all 52 required control parameters of quadrotor UAV and tunes 4 gains well enough for recovery and position control. In terms of sample efficiency, this beats current reinforcement learning methods by orders of magnitude and does not require simulations. In terms of computational efficiency, the proposed scheme runs at a rate of 2kHz on consumer flight control hardware.

Future research may aim at eliminating dependencies, such as the high frequency motor velocity feedback, and the knowledge of IMU location and orientation. Furthermore, physical experiments with other multirotor designs would validate the generalizability of the proposed method, and

expansions to other types of vehicles should be investigated. Different excitation sequences may increase sample efficiency, and keeping the adaptation active during recovery and subsequent flight may reduce time for recovery, increase controller performance and enable adaptation to different flight conditions.

ACKNOWLEDGMENT

We would like to thank Coen C. de Visser for his helpful suggestions and proofreading.

REFERENCES

- [1] E. H. Heim, E. Viken, J. M. Brandon, and M. A. Croom, “NASA’s Learn-to-Fly Project Overview,” in *2018 Atmospheric Flight Mechanics Conference*, ser. AIAA AVIATION Forum, June 2018.
- [2] J. Eschmann, D. Albani, and G. Loianno, “Learning to Fly in Seconds,” Nov. 2023, arXiv:2311.13081.
- [3] A. P. Erasmus and H. W. Jordaan, “Robust Adaptive Control of a Multirotor with an Unknown Suspended Payload,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9432–9439, Jan. 2020.
- [4] E. J. J. Smeur, Q. Chu, and G. C. H. E. de Croon, “Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles,” *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 3, pp. 450–461, Mar. 2016.
- [5] H. Li, S. Myschik, and F. Holzapfel, “Null-Space-Excitation-Based Adaptive Control for an Overactuated Hexacopter Model,” *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 3, pp. 483–498, 2023.
- [6] C. Ke, K.-Y. Cai, and Q. Quan, “Uniform Passive Fault-Tolerant Control of a Quadcopter With One, Two, or Three Rotor Failure,” *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4297–4311, Dec. 2023.
- [7] S. Sieberling, Q. P. Chu, and J. A. Mulder, “Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction,” *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, pp. 1732–1742, Nov. 2010.
- [8] T. M. Blaha, E. J. J. Smeur, and B. D. W. Remes, “A Survey of Optimal Control Allocation for Aerial Vehicle Control,” *Actuators*, vol. 12, no. 7, p. 282, July 2023.
- [9] B. Bacon, A. Ostroff, and S. Joshi, “Reconfigurable NDI controller using inertial sensor failure detection & isolation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 4, pp. 1373–1383, Oct. 2001.
- [10] V. K. Madiseti and D. B. Williams, *The Digital Signal Processing Handbook, Second Edition - 3 Volume Set.*, 2nd ed., ser. Electrical Engineering Handbook Ser. Boca Raton: Chapman and Hall/CRC, 2009.
- [11] E. A. Morelli, “Practical Aspects of Multiple-Input Design for Aircraft System Identification Flight Tests,” in *AIAA AVIATION 2021 FORUM. VIRTUAL EVENT: American Institute of Aeronautics and Astronautics*, Aug. 2021.