# Flying a quadrotor with unknown actuators and sensor configuration

Blaha, T.M.; Smeur, E.J.J.; Remes, B.D.W.; de Visser, C.C.

**Publication date**
2024

**Document Version**
Final published version

**Published in**
IMAV 2024: Proceedings of the 15th annual International Micro Air Vehicle Conference and Competition

**Citation (APA)**
Blaha, T. M., Smeur, E. J. J., Remes, B. D. W., & de Visser, C. C. (2024). Flying a quadrotor with unknown actuators and sensor configuration. In T. Richardson (Ed.), *IMAV 2024: Proceedings of the 15th annual International Micro Air Vehicle Conference and Competition: September 16-20, 2024 Bristol, United Kingdom* (pp. 192-198). Article IMAV2024-23

# Flying a Quadrotor with Unknown Actuators and Sensor Configuration

T.M. Blaha[*], E.J.J. Smeur, B.D.W. Remes and C.C. de Visser
Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands

## Abstract

Though control algorithms for multirotor Unmanned Air Vehicle (UAV) are well understood, the configuration, parameter estimation, and tuning of flight control algorithms takes quite some time and resources. In previous work, we have shown that it is possible to identify the control effectiveness and motor dynamics of a multirotor fast enough for it to recover to a stable hover after being thrown 4 meters in the air. In this paper, we extend this to include estimation of the position of the Inertial Measurement Unit (IMU) relative to the Center of Gravity (CoG), estimation of the IMU rotation, the thrust direction of all motors and the optimal combined thrust direction. In order to guarantee a correct IMU position estimation, two prior throw-and-catches of the vehicle with spin around different axes are required. For these throws, a height as low as 1 meter is sufficient. Quadrotor flight experimentation confirms the efficacy of the approach, and a simulation shows its applicability to fully-actuated crafts with multiple possible hover orientations.

## 1 Introduction

Drones are used for an ever increasing number of applications, with variations in the number and placement of propellers, the craft's inertia and sensor location and orientation. Careful controller design is needed to ensure new platforms are stable and demonstrate good flight performance. However, the process of developing, testing, and fine-tuning flight control algorithms for new drones is both time-consuming and expensive, which poses a significant barrier to the advancement of new drone applications.

In recent work [1], we have shown that it is possible to learn all required control parameters in the span of a single throw, such that a multirotor learns to fly without prior knowledge of its motor dynamics time constant, actuator position and effectiveness. In that paper, two important assumptions were made. First it was assumed that the motors were providing upward thrust in a body frame, simplifying the position control to that of a standard multirotor. The motors may not always be mounted in this configuration, or in the case of fully or overactuated multirotors, independent thrust can be generated in any direction, allowing multiple hover attitudes [2]. The second assumption was that the Inertial Measurement Unit (IMU) orientation and position offset relative to the Center of Gravity (CoG) were known. This assumption was needed, because of the high angular rates the vehicle obtains during throwing and excitation, which necessitate compensation of the acceleration measured by the IMU to transform it to the CoG.

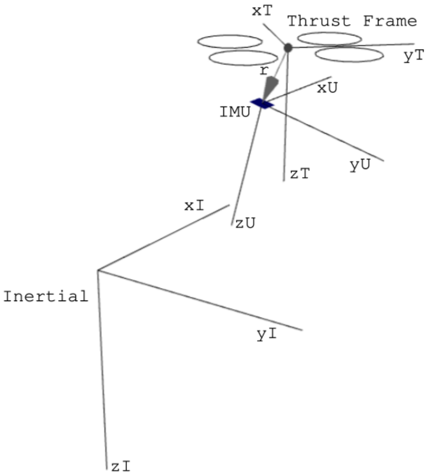

Figure 1: Coordinate Systems: Identifying the transformation between thrust frame $T$ and IMU frame $U$ from less than 1 second of flight data is the main contribution of this work.

Online estimation of the position of the CoG relative to the IMU was addressed in [3], but this was done relying on actuator inputs with a predetermined effectiveness. Svacha et al. [4] implemented a 32 state Unscented Kalman Filter (UKF) estimating all mass properties, including IMU offset. However, this was done with offline data, with known mass and motor locations. Lukacs and Lantos [5] provide an algorithm for determining the position of the CoG together with a force model for a sailplane, but practical results are not provided. [6] estimated the moments of inertia, CoG position, and aerodynamic parameters of a quadrotor model using a Minimum Likelihood Estimator, but this can only be

[*] Email address(es): {t.m.blaha; e.j.j.smeur; b.d.w.remes; c.c.devisser}@tudelft.nl
Data: 10.4121/0530be90-cc6c-4029-9774-670657882906
Code: github.com/tudelft/indiflightSupport/tree/iros_imav_2024

run offline. The typical approach in the literature seems to be to estimate the CoG offset at the same time as the control effectiveness, which either requires some assumptions on the geometry of the drone, or extends the estimation time significantly.

In this paper, we extend the work of [1] to unknown IMU orientation and position, and unknown thrust directions of the actuators. However, the basic assumption that the vehicle has to be capable of static hover is maintained. The only user input to the system is two throw-and-catches of about 1 meter height with sufficient spin and one throw-and-fly of around 3.5 meter height. The applicability of the algorithm is shown through real life experiments with a quadrotor.

## 2 Frames of Reference

We distinguish 3 coordinate frames, illustrated in Figure 1, and relate them using unit quaternions. The frame of the IMU with respect the inertial frame is described by $q_I^U$. The "thrust frame" $T$ is defined with its origin in the center of gravity of the vehicle, and the $z^T$-axis pointing opposite a chosen thrust axis. This axis is obvious for quadrotors, but e.g. for fully actuated hexarotors that can hover in many different orientations[2], the choice may not be obvious, so that we define $-z^T$ as the axis of *most efficient* thrust generation capacity while simultaneously not causing any body-moments. The choice of $x^T$ is commonly taken as the front of the vehicle, but in our study this choice is arbitrary.

The thrust frame relates to the IMU frame with $q_U^T$, and to the inertial frame with the intrinsic rotation sequence $q_I^T = q_I^U\, q_U^T$.

A problem with the arbitrary definition of the $x^T$ and $y^T$ axes, is that the yaw angle needs to be known in order to control the position and velocity of a multirotor. We see two solutions for this problem.

The first is an Extended Kalman Filter (EKF) that estimates the heading by fusing Global Navigation (GNSS) measurements and IMU acceleration measurements projected onto the inertial xy-plane. The second is to include a magnetometer, of which the axes are aligned with the IMU.

In this study we only consider IMU rotations around an axis in the thrust-xy plane. This guarantees that a desired inertial accelerations can be translated to the correct attitude changes for closed loop control.

## 3 Methodology

This section gives a brief overview of the state estimation, controller structure and system identification onboard of the UAV. Then our method to determine the IMU location and the thrust frame is explained with focus on real-time implementation.

### 3.1 Controller Summary and Effectiveness identification

The same controller and estimator structure of [1] is also used in this work. A short summary:

A complementary attitude filter is used to estimate $q_I^U$. The yaw angle computed by an external motion capture system is fused into this estimate.

An Incremental Nonlinear Dynamic Inversion (INDI) controller is used to give motor commands to achieve a desired $q_I^T$, which in turn is computed by a Nonlinear Dynamic Inversion (NDI) position controller.

The identification of the effectiveness matrix and motor model for the INDI controller results from a Recursive Least Squares (RLS) implementation based on the measured angular rate derivatives and specific forces in response to a sequential open-loop excitation of each motor.

### 3.2 IMU location estimation

The inertial acceleration $a_r$ of a point $r$ expressed in the coordinates of a local, rotating frame $T$ is given by:

$$a_r = a_T + \ddot{r} + 2\Omega_T \times \dot{r} + \dot{\Omega}_T \times r + \Omega_T \times (\Omega_T \times r)\ , \tag{1}$$

where $a_T$ is the inertial acceleration of the frame, $r,\ \dot{r},\ \ddot{r}$ is the location, velocity and acceleration of $r$ relative to frame $T$, and $\Omega_T,\ \dot{\Omega}_T$ the rotation rate and acceleration of frame $T$. All quantities are expressed in the coordinates of frame $T$.

A rigidly mounted IMU is now placed at $r$, and the origin of the rotating frame is chosen at the vehicle's center of gravity: $\dot{r} = \ddot{r} = 0$. Since accelerometers do not measure gravity (but rather all other specific forces), its reading for a free-tumbling rigid body on which no external forces act will follow this reduced equation with $a_T = 0$:

$$a_r = \dot{\Omega}_T \times r + \Omega_T \times (\Omega_T \times r) \tag{2}$$

$$\underbrace{a_r}_{y_i} = \underbrace{\begin{pmatrix} -\Omega_y^2 - \Omega_z^2 & \Omega_x\Omega_y - \dot{\Omega}_z & \Omega_x\Omega_z + \dot{\Omega}_y \\ \Omega_x\Omega_y + \dot{\Omega}_z & -\Omega_x^2 - \Omega_z^2 & \Omega_y\Omega_z - \dot{\Omega}_x \\ \Omega_x\Omega_z - \dot{\Omega}_y & \Omega_y\Omega_z + \dot{\Omega}_x & -\Omega_x^2 - \Omega_y^2 \end{pmatrix}}_{X_i} \underbrace{\begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}}_{r} . \tag{3}$$

This equation is linear in the parameters $r$, and so if $\Omega,\ \dot{\Omega},\ a_r$ are measured during such a tree-tumble, the observations $X_i$ and $y_i$ can be aggregated in matrix $X$ and vector $y$. Least Squares (LS) can then be used to obtain an estimate of $r$.

The conditions of a tumbling rigid body with minimal external forces (e.g. from aerodynamic drag) can be achieved by throwing it in the air with sufficient angular velocity. One single throw of the vehicle would be ideal, however it is generally not enough to reliably fit $r$: if the vehicle is rotating around a single axis (without loss of generality, assume the axis $(0,\ 0,\ 1)$), then the resulting equation is

$$a_r = \begin{pmatrix} \Omega_z^2 & -\dot{\Omega}_z & +\dot{\Omega}_y \\ \dot{\Omega}_z & -\Omega_z^2 & -\dot{\Omega}_x \\ -\dot{\Omega}_y & \dot{\Omega}_x & 0 \end{pmatrix} \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}\ , \tag{4}$$

and so if $(0,\ 0,\ 1)$ happens to be close to a principle inertial axis, then $\dot{\Omega}$ is expected to be small and $r_z$ cannot be estimated. It is later experimentally illustrated that two throws about significantly different axes provide enough excitation.

### *3.3 Thrust Direction Estimation*

Instead of estimating the IMU's orientation with respect to some other body-fixed frame, we estimate the direction in which force can be generated with the motors, without producing body-moments. For conventional quadrotors there is only one such direction (coinciding with the axes of each motor), but many more directions are possible for configurations with more actuators and tilted rotors. If we knew a suitable such direction (e.g. the most power-efficient one), we can use this to construct a reference frame in which a conventional multirotor controller can operate.

Recall that we assume the multirotor configuration to be able to achieve steady-state hover. We now approximate the specific forces and torques acting on the vehicle as linear combinations of the motor input $u \in \mathbb{R}^m$, where $m$ is the number of motors on the vehicle:

$$\begin{pmatrix} f \\ \tau \end{pmatrix} = \begin{pmatrix} G_{1,f} \\ G_{1,\tau} \end{pmatrix} u \, . \tag{5}$$

Here, the $f \in \mathbb{R}^3$ and $\tau \in \mathbb{R}^3$ are the specific forces and torques, respectively, and $G_{1,f} \in \mathbb{R}^{3\times m}$ and $G_{1,\tau} \in \mathbb{R}^{3\times m}$ are the corresponding steady-state effectiveness matrices. In [1], these matrices are identified locally using the increments of $u$ and $(f,\ \tau)^T$, but here we assume that they hold in the global sense of Equations 5. Note that Equation 5 is only valid in steady-state when $\dot{u} = 0$.

The most efficient distribution of control effort in hover can be defined as the one that minimizes some square norm of $u$, while ensuring (1) there are no torques acting on the vehicle ($\tau = 0$), (2) the gravity is canceled by the forces ($||f|| = g$), and (3) the bounds on $u$ are respected. Introducing the diagonal actuator weighing matrix $W$, these statements are summarized in the non-convex quadratically constrained quadratic problem

$$\begin{aligned} \min_u \quad & u^T W u \\ s.t. \quad & u^T (G_{1,f}^T G_{1,f}) u = g^2 \\ & G_{1,\tau} u = 0 \\ & \underline{u} \leq u \leq \overline{u} \, . \end{aligned} \tag{6}$$

If a solution $u = u^*$ exists, then the vehicle is indeed capable of static hover and the most efficient direction to generate thrust without causing rotations is $d = G_{1,f} u^*$.

General quadratic programming solvers readily solve this problem, however, their implementation in embedded hardware is difficult, and computation time is slow and convergence time may not be guaranteed. If $G_1$ changes, or at least its estimate, then Problem 6 would have to be solved again and again, which quickly becomes infeasible.

**Real-Time Implementation** We now introduce a heuristic and a simplification that allows development of an efficient algorithm to solve Problem 6 in sufficient cases.

Firstly, assume that the bounds $\underline{u} \leq u \leq \overline{u}$ are always fulfilled if the first two constraints are met, and can thus be dropped. This may cause the found solution to be infeasible if some motors are at their limit or not used at all at true solution $u^*$. The rationale for this simplification is that it is unlikely that a vehicle capable of static hover requires motors at maximum or minimum thrust even in its most efficient hovering attitude.

Secondly, assuming identical motors, impose $W = I$. This will cause sub-optimal hover distributions when the motors do not have the same power consumption for a given input $u$.

The algorithm follows these steps

1. Introduce new variables $\eta \in \mathbb{R}^{m-3}$ by calculating an orthogonal basis $N_\tau$ for the $(m-3)$-dimensional nullspace of $G_{1,\tau}$ (full-rank is guaranteed by the assumption that the vehicle can achieve static hover). This can be efficiently achieved by a column-pivoting complete QR decomposition of $G_{1,\tau}^T$, e.g. with the LAPACK routine SGEQP3[1].

2. Now, any choice of $u = N_\tau \eta$ satisfies the constraint $G_{1,\tau} u = 0$, and what remains is

$$\begin{aligned} \min_\eta \quad & \eta^T N_\tau^T W N_\tau \eta \\ s.t. \quad & \eta^T (N_\tau^T G_{1,f}^T G_{1,f} N_\tau) \eta = g^2 \, . \end{aligned} \tag{7}$$

   Defining $H \triangleq N_\tau^T W N_\tau$, and $A \triangleq N_\tau^T G_{1,f}^T G_{1,f} N_\tau$, and introducing the Lagrange multiplier $\lambda$, we can write the KKT system of Equation 7 as

$$\begin{aligned} 2H\eta + 2\lambda \cdot A\eta &= 0 \\ \eta^T A \eta &= g^2 \end{aligned} \tag{8}$$

3. It is now clear why we imposed $W = I$: $N_\tau$ is an orthogonal basis for the nullspace, due to our choice of the QR factorization. Consequently, $H = I$ as well, and the first line of Equation 8 is an eigen-problem $Av_i = \tilde{\lambda}_i v_i$, where we explicitly, but without loss of generality assume $||v_i|| = 1$. Now we know that $\eta^* = k v_i$ with one of the eigenvectors of $A$ and some scaling factor $k$, which can be computed by substituting this relation into the second line of 8:

$$k^2 v_i^T \underbrace{A v_i}_{\tilde{\lambda}_i v_i} = g^2 \implies k = \frac{g}{\sqrt{\tilde{\lambda}_i v_i^T v_i}} = \frac{g}{\sqrt{\tilde{\lambda}_i}} \, . \tag{9}$$

[1] netlib.org/lapack/explore-html/d0/dea/group__geqp3

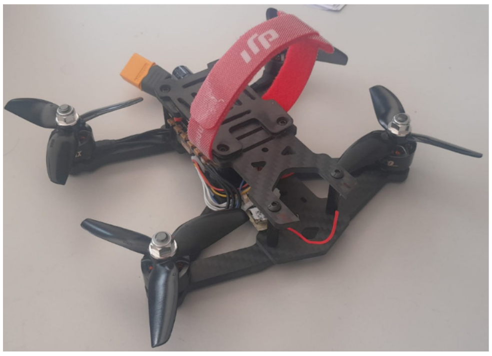

Figure 2: Compact 3-inch quadrotor used for the flight experiments.

Consequently, $u^* = \pm N_\tau \eta^* = \pm N_\tau k v_i = \pm N_\tau g/\sqrt{\lambda}_i v_i$, where the sign must be chosen so that the components of $u^*$ are within $\underline{u} \leq u \leq \overline{u}$. Following our heuristic, if this is impossible, then the vehicle is very likely not capable of static hover in the first place. This relation also shows that we require not just any, but the largest eigen-pair (or one of the largest eigen-pairs, if there are multiple equal ones) of $A$, as that results in the lowest magnitude of $u^*$.

4. Since $A \succ 0$, we can compute the largest eigen-pair of $A$ using the power iteration $v_{j+1} = \frac{Av_j}{||Av_j||}$. In practice an iteration limit should be imposed, and since the parameters of the problem are not expected to change rapidly, subsequent invocations of step 4. will continue to improve the guess.

5. Finally, the axis of most efficient thrust generation is then $d = G_{1,f} u^*$, where $u^* = \pm N_\tau g/\sqrt{\lambda} v_i$, with $\pm$ and $v_i$ as determined above. Then, $q_U^T$ can finally be assembled as the shortest rotation that maps $d$ to $(0,\ 0,\ -g)$.

Most of the computation time of the algorithm above is expected in the computation of the QR-decomposition which increases linearly with the amount of motors $m^2$. The subsequent matrix calculations increase with $(m-3)^2$.

### *3.4 Experiment Design – Quadrotor*

The estimation and control algorithm is verified experimentally, in the same fashion as done in [1], by implementing the computations in the firmware INDIflight[2] onboard of an STM32H743 flight controller (480 MHz clockspeed), and using gyroscope and accelerometer data from a TDK InvenSense ICM-42688-P IMU. The vehicle is pictured in Figure 2. The update rate of the attitude control and actuator effectiveness estimation is $2kHz$. Even though it is not necessary to run the thrust frame estimation at the same rate, this is still performed for ease of comparison of the runtimes later. For safety reasons, the quadrotor is launched and given an initial rotation of $\Omega_0 = (400,\ 400,\ 100)^T$ deg/s to simulate a human throw. The invariance of the estimations to this initial condition has been shown in [1].

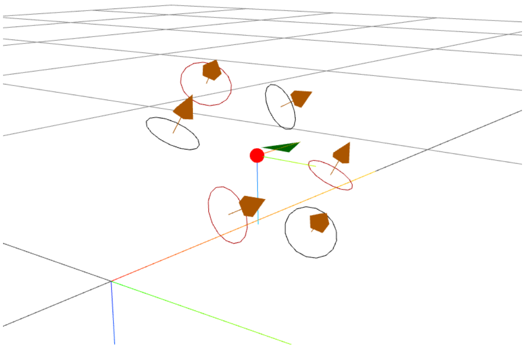

Figure 3: Fully-actuated hexarotor that can hover in multiple orientations.

In order to test different IMU orientations without making changes to the hardware, the IMU measurements are rotated by the inverse of $q_U^T = 0.837 - 0.491i + 0.242j$, which corresponds to Euler angles of $-15°$ Yaw, $24°$ Pitch, $-64°$. As explained in the limitation of Section 2, this rotation results in correct yaw alignment when rotated back using the algorithm of Section 3.3.

Though it is entirely possible to do the estimation of the IMU location $r$ onboard, for the demonstration of this work it has been performed offline with logged sensor measurements, and was then manually programmed into the memory of the flight controller. Onboard estimation of the parameter would merely require detection of the 2 required throws and casting the Least Squares (LS) estimator as a Recursive Least Squares (RLS) estimator to eliminate the memory requirements of the offline LS.

### *3.5 Simulation Design – Fully Actuated Hexarotor*

Steps 3 through 5 of the algorithm in Section 3.3 only need to involve the matrix algebra if the rotational nullspace $N_\tau$ spans multiple dimensions. In the case of a quadrotor, $\dim(N_\tau) = 1$ and steps 3-5 are trivial.

The hexarotor ($\dim(N_\tau) = 3$) shown in Figure 3 has an arrangement such that there is one energy-optimal hover orientation, obvious by inspection, but many other suboptimal ones [2]. To test our algorithm, a simulation model has been created with the added difficulty that the entire actuator constellation is rotated by $45°$ with respect to the IMU (the green arrow in Figure 3). This illustrates again the equivalence of unknown IMU orientation, and unknown actuator locations; our approach does not need an explicitly defined "body-frame".

The simulation is run with the same control code as in the

[2] github.com/tudelft/indiflight

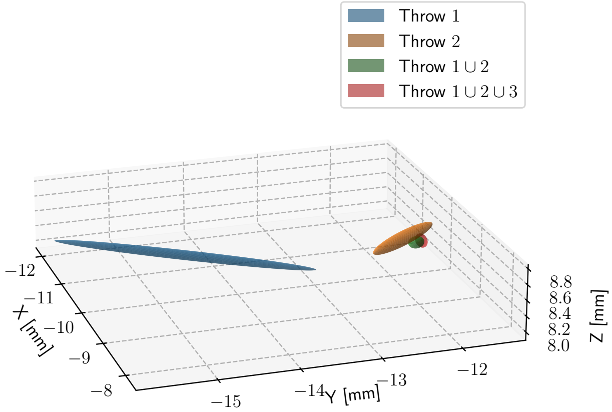

Figure 4: 95% percent confidence ellipsoids of the IMU offset parameter for a sequence of throws, where $1 \cup 2$ stands for the union of the data obtained from throws 1 and 2. Least Squares fails for just 1 throw, but succeeds for multiple.

flight-experiment, and may be reproduced using Docker[3].

## 4 RESULTS

All data used in this work is available for reproduction in a dataset[4].

### 4.1 IMU location

Figure 4 shows the 95% percentile confidence ellipsoids for the parameters $\hat{r}$ after LS fitting, defined by $(r - \hat{r})^T \Sigma^{-1} (r - \hat{r}) < iCDF(\chi^2_3, 0.95)$. Here, $\Sigma \approx \text{SE}^2 \cdot (X^T X)^{-1}$ and the standard error of the residuals $\text{SE}^2 = ||y - X\hat{r}||_2^2 / (N-3)$ [7].

These ellipsoids capture the effects of correlation in the problem data $X$ and the residual error variance in the predictions $y - X\hat{r}$. Systematic errors like those arising from, e.g. inaccurate scaling of the IMU measurements, are not captured.

Using only the observations from a single throw yields uncertain estimates in one direction, as indicated in Section 3.2. However, combining two throws with spin around two different axes reduces the confidence bounds to the sub-millimeter range and is not much improved by adding data from a third throw. The remaining uncertainty may be due to sensor bias, aerodynamic drag being a factor ($a^T \neq 0$), and the individual test masses of the 3 Micro Electro-Mechanical System (MEMS) accelerometers not being exactly in the same location.

### 4.2 Flight Experiment Results – Quadrotor

The recovery of the vehicle and attaining a position setpoint follows a similar trajectory to the results shown in [1]. The excitation and system response are shown on the left in Figure 5. Due to the alignment of the IMU, each motor gives rise to specific forces in all three axes. The right side of Figure 5 shows the identification of the translational part of the effectiveness matrix $G_{1,f}$, where every vertical dashed line corresponds to the start of the excitation of the next actuator. Once all actuators have been triggered (roughly 300ms into the sequence), the computation of the algorithm determining the thrust direction (Section 3.3) runs to completion, as $G_{1,\tau}$ now has a nullspace of dimension greater than 0.

The estimated and true alignment of the IMU is given in Table 1. The alignment is sufficiently accurate to not affect manual or automatic position control performance.

| | Ground truth | Estimated |
|---|---|---|
| Roll | -15° | -14.1° |
| Pitch | 24° | 23.7° |
| Yaw | -64° | -61.0° |

Table 1: Quadrotor Flight Experiment: Estimated rotation of the IMU compared to the ground truth.

### 4.3 Execution Timings

All execution on the Flight Controller is single-thread, which means that execution time can be measured accurately, with only interrupt handling being a source of error. Figure 7 shows the cumulative execution times of the components of the system identification and thrust axis determination algorithms. The compilation was done with `gcc 10.3.1` at optimization level `-O3`.

The thrust axis estimation adds only a small delay, amounting to an additional $2kHz \cdot 20e-6s = 4\%$ CPU load, over the other identification components present in [1].

### 4.4 Simulation Results – Hexarotor

Also in the simulated throw-to-hover, the results show that the optimal thrust frame is estimated correctly at the end of the fitting process. In the ground truth, this frame of most efficient hover has been rotated in roll by 45°, and so the attitude quaternion that our algorithm seeks is $q_U^T = 0.924{+}0.383i$. The lower right subplot in Figure 6 shows that this is reached at the time the last actuator has been excited. In particular, our method found $\hat{q}_U^T = 0.930 + 0.367i - 0.026j$, which is shown in Euler angles in Table 2.

| | Ground truth | Estimated |
|---|---|---|
| Roll | 45° | 43.1° |
| Pitch | 0° | -2.74° |
| Yaw | 0° | -1.08° |

Table 2: Hexacopter Simulation: Estimated rotation of the IMU compared to the ground truth.

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

[4] 10.4121/0530be90-cc6c-4029-9774-670657882906

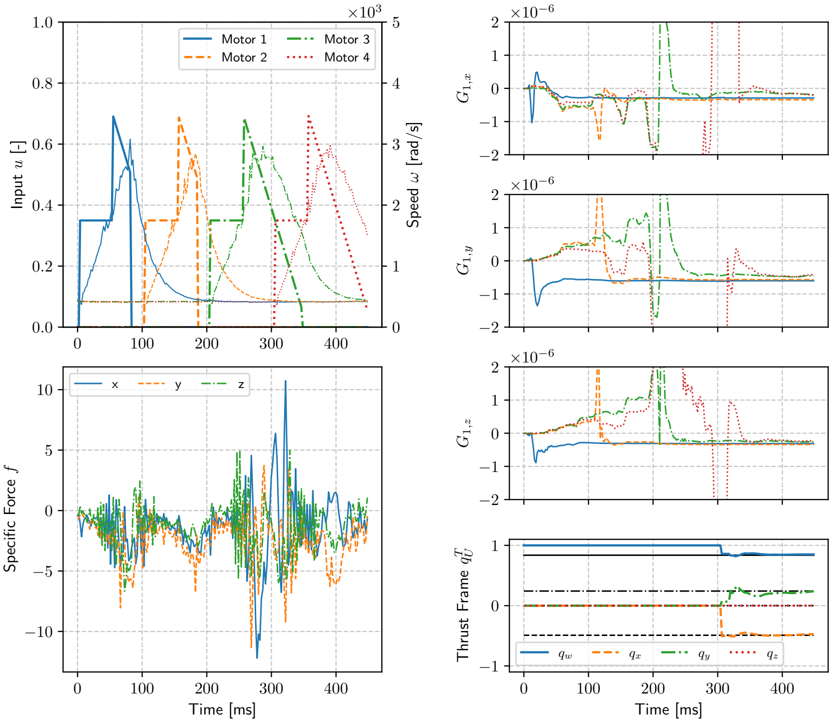

Figure 5: Flight experiment: Inputs $u$, motor response $\omega$, accelerometer measurements $f$ and fitting of the linear part of the Effectiveness Matrix ($G_{1,xyz}$). The last plot shows that the true IMU alignment is approached closely, once the last motor is excited.

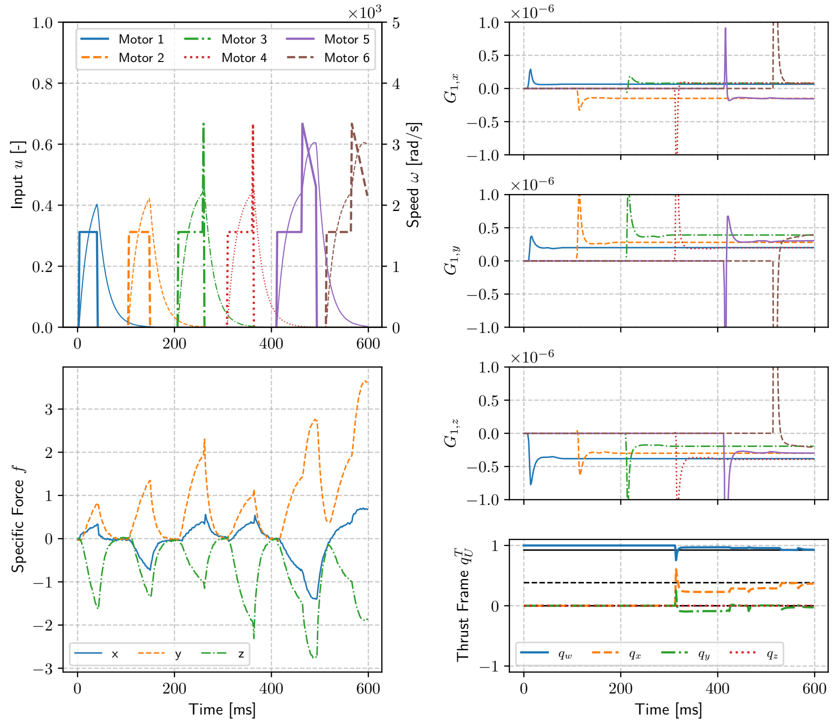

Figure 6: Hexarotor simulation: The parameters of the 6 actuators are identified in the span of 600ms. The specific-force part of $G_1$ (xyz-components shown on the top right for all 6 actuators) show the significant contributions in all IMU axes, not just $z$. From these measurements, our method estimates the optimal thrust frame.

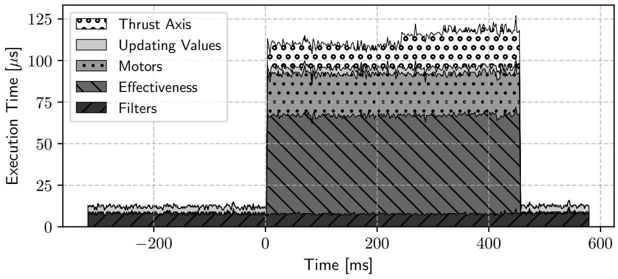

Figure 7: Cumulative execution time on the flight controller show that 2kHz update rates are feasible for quadrotors.

## 5 Conclusion

In this study, we successfully demonstrated attitude control of a quadrotor where the motors' location and strength are unknown, as well as the alignment and location of the IMU. The location of the IMU has been estimated from two prior throw-and-catches, after which the quadrotor was able to recover from being launched on a 3.5m parabolic trajectory.

Recursive Least Squares was used to identify the motors' rotational and linear effectiveness in-flight, which in turn were used to calculate an optimal thrust direction in the frame of the IMU which can be actuated without giving rise to rotations. The latter is the main contribution of this paper and enables position control if an external source of position and heading (GPS+magnetometer, optical motion capture) is available and the alignment of the IMU $q_U^T$ is restricted to yaw-free rotations. The presented algorithm computes in $20\mu$s on consumer-grade flight control hardware, however computation time will grow with $(m-3)^2$, where $m$ is the number of actuators.

The ability of our method to compute the most efficient hover direction, as opposed to just any, was confirmed in a simulation of a fully actuated hexarotor.

In future work, position control should be enabled even in the cases where the IMU-to-hover frame alignment includes yaw components, either by including a magnetometer sensor aligned with the IMU, or by calculating heading from comparing the vehicles GPS-track with the acceleration measured onboard.

Furthermore, the dependence on prior throws may be eliminated by eliminating the need to know the IMU location before identification of the motors' effectiveness. This could be achieved by providing additional excitation in the nullspace of $G_{1,\tau}$ once rotations have been arrested. In this case, the accelerometer will measure the true specific forces, as every point on the vehicle experiences the same acceleration.

## Acknowledgements

The authors would like to thank Elijah Hao Wei Ang for his insights.

## References

[1] Till Blaha, Ewoud Smeur, and Bart Remes. Control of Unknown Quadrotors from a Single Throw. *arXiv*, arXiv:2406.11723, 2024.

[2] Sujit Rajappa, Markus Ryll, Heinrich H. Bülthoff, and Antonio Franchi. Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4006–4013, May 2015.

[3] Emil Fresk, David Wuthier, and George Nikolakopoulos. Generalized center of gravity compensation for multirotors with application to aerial manipulation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4424–4429, September 2017.

[4] James Svacha, James Paulos, Giuseppe Loianno, and Vijay Kumar. IMU-Based Inertia Estimation for a Quadrotor Using Newton-Euler Dynamics. *IEEE Robotics and Automation Letters*, 5(3):3861–3867, July 2020.

[5] L. Lukacs and B. Lantos. Nonlinear dynamic model identification of airplanes including inertial parameters and force-Torque relations based on real flight data. In *SACI 2015 - 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics, Proceedings*, pages 393–398, 2015.

[6] M. Burri, J. Nikolic, H. Oleynikova, M.W. Achtelik, and R. Siegwart. Maximum likelihood parameter identification for MAVs. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2016-June, pages 4297–4303, 2016.

[7] Larry Wasserman. Lecture 18: Tests and Confidence Sets for Multiple Coefficients. Lecture Series, Carnegie Mellon University, Pittsburgh, PA 15213, USA, 2017.