FUSE-D: Framework for UAV System-Parameter Estimation with Disturbance Detection

Christoph Böhm and Stephan Weiss

***Abstract*— Modern unmanned aerial vehicles (UAVs) with sophisticated mechanics ask for extended online system identification to aid model-based controls in task execution. In addition, UAVs in adverse environmental conditions require a more detailed environmental disturbance understanding. The necessary combination of online system identification, sensor suite self-calibration, and external disturbance analysis to tackle these issues holistically is currently an open issue.**

Our proposed *FUSE-D* approach combines these elements based on a system model at the rotor-speed level and a single global pose sensor (e.g., a tracking system like Optitrack). Besides sensor intrinsics and extrinsics, the framework allows estimating the UAV's rotor geometry, mass, moments of inertia, and the rotors' aerodynamic properties, as well as an external force and where it acts on the UAV. The general formulation allows us to extend the approach to an N-rotor (multi-rotor) UAV and classify the type of external disturbance. We perform a detailed non-linear observability analysis for the $43 + 7N$ states and do a statistically relevant embedded hardware-in-the-loop performance analysis in the realistic simulation environment Gazebo with RotorS.

I. Introduction

Multi-rotor UAVs became a widely used tool in search and rescue missions, exploration, long-term autonomy, transportation, and entertainment over the last years.

Their deployments and frequent interaction with the environment expose them to various disturbances and uncertainties, raising the need to adapt to different flight conditions and tasks – robust performance and accurate flight paths. This adaptation happens through (i) more sophisticated multi-rotor platforms that allow for high degrees of freedom (DoF) motion [1]–[4] or (ii) controls that take system parameter changes or disturbances into account [5]–[8] or both.

In most cases, approaches assume that the system or environment will not change (or in a known way, e.g., [1]) over time, which can not be guaranteed. Commonly used offline system identification methods' built models could render unusable. The state-of-the-art (SOA) provides a large body of work identifying either changes in the system parameter or changes in the environment. To the best of our knowledge, no work combines both while including system self-calibration in a holistic approach.

A. Related Work

1) System Parameter Estimators: Robustness can be achieved through "self-awareness" of the UAV – estimating system parameters (geometrical, inertial, and aerodynamic properties – such as, e.g., mass or thrust force coefficient).

Both authors are with the Control of Networked Systems Group at the University of Klagenfurt, Austria. email: {firstname.lastname}@ieee.org

Accepted June/2023 for IROS 2023, DOI follows ASAP ©IEEE.

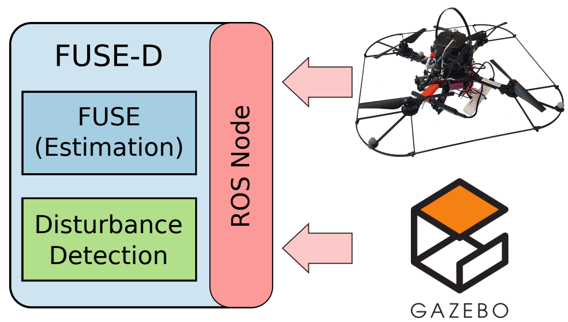

Fig. 1. Overview of all components of FUSE-D.

A couple of offline methods, apart from classical system identification, were presented in [9]–[12]. These apply offline nonlinear least-squares algorithms to do the self-calibration of geometrical, inertial, and aerodynamic properties of the UAV. They produce better estimation results for the control states and system parameters, but can not be used online on small and computationally limited UAVs due to the problem complexity and required data stream length.

Other works solve the estimation problem through the implementation of an extended Kalman filter (EKF) [13], [14] or unscented Kalman filter (UKF) [15], [16]. The recursive nature and inherent "lightweightness" allows for run-time estimation mid-flight as shown by these works. With proper system input (Lissajous figures or observability-aware motions [17]) one can achieve estimation performance close to the optimization-based approaches, c.f. [14]. [16], [18], [19] model additional drag effects, e.g., caused by the UAV's body or the rotors, which can improve estimation quality.

Recently, [20] proposed a Schmidt-Kalman filter (SKF) that propagates the UAV's state through the IMU measurements and updates them based on its dynamics. Their results confirm our previous findings (c.f. [13] and [14]), but do not treat each rotor individually and apply assumptions to the rotor geometry in the estimation process.

Still, all of them do not model external disturbances.

2) Disturbance Estimators: Next are estimators specific to external disturbances or control schemes (i.e., [21]) to reject those during trajectory tracking and, thus, provide indirect information on external disturbances.

Estimators in [21]–[27] allow to estimate wind as disturbance acting on the UAV, impacting the trajectory tracking performance. Many of these chose an abstraction level at force and torque for control inputs (e.g., [24]) and all of them assume system parameters, which we estimate with *FUSE-D* in Sec. I-A.1, to be known and to not change over time. The closest example to our approach can be found in [27], which bases the estimation process on an EKF with force and torque as control inputs, still assuming system parameters to be known and static over time.

All these approaches show impressive results in the estimation of disturbances acting on the UAV, but lack the "self-awareness" of the system self-calibration.

B. Contributions

In this work, we present the Framework for UAV System-Parameter Estimation with Disturbance Detection (FUSE-D); a novel real-time capable holistic framework that combines system-parameter estimation with disturbance detection and sensor-suite self-calibration. More precisely, the contributions presented in the following sections are:

- A significant extension of [14] by adding the online estimation of an external force and its point of application (i.e., lever arm with respect to the center of mass) in a holistic error-state Kalman filter (ESKF) framework.
- A real-time classification model with low latency to distinguish between force (e.g., wind gust) or moment (e.g., collision with environment) acting on the UAV, and ground contact (e.g., landed).
- A detailed non-linear observability analysis revealing that a rotor-speed based UAV model for state dynamics and a single global pose sensor for corrections together with precisely identified easily measurable UAV geometry and mass information is sufficient to render FUSE-D a fully observable system.
- A statistically relevant evaluation and validation of FUSE-D operating on an Odroid XU4 in a hardware-in-the-loop simulation based on Gazebo and RotorS [7] for a highly realistic environment and model settings.

FUSE-D is implemented in C++ and usable on ARM as well as Intel architectures.

II. Preliminaries

Throughout this paper, W is for the fixed world frame, M denotes the center of mass (CoM), P is the frame of the exteroceptive sensor (e.g., position or pose), I labels the inertial measurement unit (IMU)'s reference frame, A_i labels each rotor's center of the tip path plane (TPP), and E refers to the point of application of an external disturbance. Fig. II shows all these reference frames in context.

We define the position (linear velocity, linear acceleration, angular velocity, bias, force, moment, axis) vector as ${}_{[Frame]}\,\mathbf{r}\,_{[From]\,[To]} \in \mathbb{R}^3$ ($\mathbf{v}$, $\mathbf{a}$, $\boldsymbol{\omega}$, $\mathbf{b}$, $\mathbf{F}$, $\mathbf{M}$, $\boldsymbol{\alpha}$). For example, ${}_{W}\mathbf{r}_{WM}$ is the position vector of the UAV pointing from frame W to frame M expressed in the frame W.

The orientation of W with respect to M is represented through a Hamiltonian quaternion $\mathbf{q}_{WM} \in \mathbb{R}^4$ (unit length), and its rotation matrix $\mathbf{R}_{WM}\left(\mathbf{q}_{WM}\right) \in \mathbb{R}^{3\times3}$ applies to a vector the following way: ${}_{W}\mathbf{r}_{WM} = \mathbf{R}_{WM\,M}\mathbf{r}_{WM}$. $\mathbf{q}^*_{WM}$ is its conjugate corresponding to $\mathbf{R}^\intercal_{WM}$. Therefore, quaternions define as $\mathbf{q}\,_{[To]\,[From]}$ throughout this work.

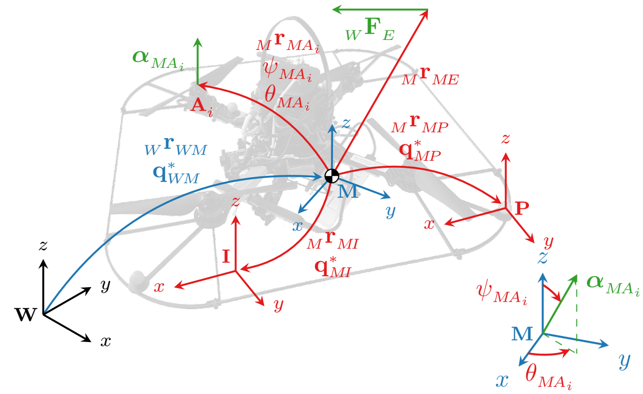

Fig. 2. Reference frames of the proposed UAV model and spherical coordinate definition of the rotor orientation axis $\boldsymbol{\alpha}_{MA_i}$ in FUSE-D.

In this work, we omit the indication of time dependency and write $\mathbf{x} = \mathbf{x}(t)$ to ease the notation and readability.

III. FUSE-D

The Framework for UAV System-Parameter Estimation with Disturbance Detection (FUSE-D) is the result of our steps in [13], [14], [17] and significant advances in this work.

A. External Force & Point of Application Estimation

Instead of estimating the gravitational pull ${}_{W}\boldsymbol{g}$ on the UAV, previous version, we consider an external force ${}_{W}\mathbf{F}_E = (f_x, f_y, f_z) \in \mathbb{R}^3$ (defined in world frame W) acting on the UAV with a point of application, or lever arm, ${}_{M}\mathbf{r}_{ME} = (r_x, r_y, r_z) \in \mathbb{R}^3$ measured from the CoM M (defined in the CoM M). As the force ${}_{W}\mathbf{F}_E$ can act at any point on the UAV, an off-center acting force causes a moment ${}_{M}\mathbf{M}_E$ around the CoM M through the lever arm ${}_{M}\mathbf{r}_{ME}$. The possible application range of estimating these states, e.g., wind gust, collision detection, or landing detection, will be illustrated in more detail in later sections of this paper.

Eq. (1) and Eq. (2) are the sum of all forces (${}_{W}\mathbf{F}_\Sigma$) and moments (${}_{M}\mathbf{M}_\Sigma$) acting on the UAV's CoM, respectively. ${}_{M}\mathbf{F}_{T_i}$ is each rotor's generated thrust (defined in the CoM M) with the magnitude depending on the rotor's squared angular velocity $\omega_i \in \mathbb{R}_+$ and thrust force coefficient $k_{T_i} \in \mathbb{R}_+$, and the direction defined by $\boldsymbol{\alpha}_{MA_i}$. This force causes two moments. The first one is the result of the rotor's drag moment coefficient $k_{M_i} \in \mathbb{R}_+$ times the spinning direction $\epsilon_i \in \{-1, 1\}$, and the second one is due to the distance ${}_{M}\mathbf{r}_{MA_i} \in \mathbb{R}^3$ of the rotor to the CoM.

$$
{}_{W}\mathbf{F}_\Sigma = \mathbf{R}_{WM} \sum_{i=1}^{N} \underbrace{\omega_i^2 k_{T_i} \boldsymbol{\alpha}_{MA_i}}_{{}_{M}\mathbf{F}_{T_i}} + {}_{W}\mathbf{F}_E \tag{1}
$$

$$
\begin{aligned}
{}_{M}\mathbf{M}_\Sigma = & \sum_{i=1}^{N} \left[\left(-\epsilon_i k_{M_i} \mathbf{I}_3 + \left[{}_{M}\mathbf{r}_{MA_i} \right]_\times \right) {}_{M}\mathbf{F}_{T_i} \right] \\
& + \left[{}_{M}\mathbf{r}_{ME} \right]_\times \mathbf{R}^\intercal_{WM\,W}\mathbf{F}_E
\end{aligned} \tag{2}
$$

Note that $[\bullet]_\times$ denotes the vector cross product.

$\boldsymbol{\alpha}_{MA_i}$ is the rotor's TPP alignment (i.e., rotor thrust direction vector) with inclination $\psi_{MA_i} \in \mathbb{R}$ from the z-axis and the azimuth $\theta_{MA_i} \in \mathbb{R}$ from the x-axis of M.

$$\boldsymbol{\alpha}_{MA_i} = \begin{bmatrix} \sin\psi_{MA_i}\cos\theta_{MA_i} \\ \sin\psi_{MA_i}\sin\theta_{MA_i} \\ \cos\psi_{MA_i} \end{bmatrix} \tag{3}$$

B. State-Space Model

The forces' and moments' influence on the trajectory states is modeled through Newton-Euler equations and [28].

The mass of the UAV is denoted with $m \in \mathbb{R}_+$ and scales all forces acting on CoM M. ${}_W\boldsymbol{g}$ is a known parameter with $[0, 0, -9.81]^\mathsf{T}\mathrm{m/s^2}$. We chose to define the inertia tensor in the center of mass (CoM) and aligned with the UAV's principle axis. This makes ${}_M\mathbf{I}$ a diagonal matrix and only the moments of inertia need to be stored in the state vector – ${}_M\mathbf{I} = \mathrm{diag}({}_M\boldsymbol{i})$ with ${}_M\boldsymbol{i} = (I_{xx}, I_{yy}, I_{zz}) \in \mathbb{R}^3_+$.

$$\begin{aligned} {}_W\dot{\mathbf{r}}_{WM} &= {}_W\mathbf{v}_{WM} && (4)\\ {}_W\dot{\mathbf{v}}_{WM} &= \tfrac{1}{m}\,{}_W\mathbf{F}_\Sigma + {}_W\boldsymbol{g} && (5)\\ \dot{\mathbf{q}}_{WM} &= \tfrac{1}{2}\mathbf{q}_{WM} \otimes \left[0, {}_M\boldsymbol{\omega}_{WM}^\mathsf{T}\right]^\mathsf{T} && (6)\\ {}_M\dot{\boldsymbol{\omega}}_{WM} &= {}_M\mathbf{I}^{-1}\left({}_M\mathbf{M}_\Sigma - [{}_M\boldsymbol{\omega}_{WM}]_\times\,{}_M\mathbf{I}\,{}_M\boldsymbol{\omega}_{WM}\right) && (7) \end{aligned}$$

The change of ${}_W\mathbf{v}_{WM}$ referenced in the world frame makes the trajectory states more accessible for controls (common definition in the UAV control community), and the term $[{}_M\boldsymbol{\omega}_{WM}]_\times\,{}_M\mathbf{v}_{WM}$ missing (usually found in body referenced velocities/accelerations [14]) does not cause additional unobservable dimensions for real-systems (c.f. Sec. III-C). A body referenced velocity is needed if velocity-dependant drag parameters would be estimated as well (e.g., [16]).

FUSE-D will use all remaining states and dynamics from [14] to form a $43 + 7N$ long state vector, with N being the number of rotors of the UAV. For a quadrotor case we have 71 states in the estimation process.

$$\mathbf{x} = \left[\mathbf{x}_T^\mathsf{T}, \mathbf{x}_S^\mathsf{T}, \mathbf{x}_I^\mathsf{T}, \mathbf{x}_{G_1}^\mathsf{T}, \ldots, \mathbf{x}_{G_N}^\mathsf{T}\right]^\mathsf{T} \in \mathbb{R}^{(43+7N)\times 1} \tag{8}$$

$\mathbf{x}_T$ contains states for control purposes (${}_W\mathbf{r}_{WM}$, ${}_W\mathbf{v}_{WM}$, $\mathbf{q}_{WM}$, and ${}_M\boldsymbol{\omega}_{WM}$). All sensor self-calibration states of a position or pose sensor and potentially an IMU are in $\mathbf{x}_S$ (${}_M\mathbf{r}_{MP}$, $\mathbf{q}_{MP}$, ${}_M\mathbf{r}_{MI}$, $\mathbf{q}_{MI}$, ${}_I\mathbf{b}_a$, and ${}_I\mathbf{b}_\omega$), with ${}_I\dot{\mathbf{b}}_a = \mathbf{w}_{{}_I\mathbf{b}_a} \sim \mathcal{N}(0, \sigma^2_{{}_I\mathbf{b}_a})$ and ${}_I\dot{\mathbf{b}}_\omega = \mathbf{w}_{{}_I\mathbf{b}_\omega} \sim \mathcal{N}(0, \sigma^2_{{}_I\mathbf{b}_\omega})$, respectively as IMU biases. States that represent physical properties of the UAV and the external disturbance are accounted for in $\mathbf{x}_I$ (m, ${}_M\boldsymbol{i}$, ${}_W\mathbf{F}_E$, and ${}_M\mathbf{r}_{ME}$). Brownian motion models the dynamic behavior of ${}_W\mathbf{F}_E$ and ${}_M\mathbf{r}_{ME}$ with ${}_W\dot{\mathbf{F}}_E = \mathbf{w}_{F_E} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{F_E})$ and ${}_M\dot{\mathbf{r}}_{ME} = \mathbf{w}_{r_{ME}} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{r_E})$, respectively. Both noise terms should allow estimating the external force and its point of application (lever arm) with properly chosen $\boldsymbol{\sigma}^2_{F_E}$ and $\boldsymbol{\sigma}^2_{r_E}$ values (cf. the example in Sec. IV). All rotor parameters, including position, orientation, and aerodynamics, are stored in individual state vectors $\mathbf{x}_{G_i}$ (${}_M\mathbf{r}_{MA_i}$, ψ_{MA_i}, θ_{MA_i}, k_{T_i}, and k_{M_i}) for each rotor i.

All other quantities are assumed to be static and do not change over time unless otherwise defined. This assumption does not contradict their inclusion in the estimation, as such parameters could change during operation (e.g., collisions or changing payload). Applying statistical tests to the estimates, e.g., χ^2, and resetting the covariance in case of failure, could allow for re-estimation of the new values.

The angular velocities of the rotors are the control inputs of the system model and are defined as $\epsilon_i\omega_i$, with ω_i being positive and $\epsilon_i = 1$ (respectively -1) for counterclockwise (CCW) (respectively clockwise (CW)) rotation [29]. The control input vector is $\mathbf{u} = [\omega_1, \ldots, \omega_N]^\mathsf{T} + \mathbf{w}_u \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_\mathbf{u})$, with $\mathbf{w}_u$ being the noise on the control inputs themselves.

C. Observability Properties with the new States

Including the external force ${}_W\mathbf{F}_E$ and its lever arm ${}_M\mathbf{r}_{ME}$ into a holistic estimation framework requires a careful analysis of which states are observable under which conditions similar to the methods shown in [30]–[33].

Compared to [14], our new system model shows several jointly observable states which link the external force, its lever arm and other system parameter of the UAV. In the following, we give an overview of the resulting generalized (N rotors) observability properties in case of (i) only pose sensor measurements, (ii) only position sensor measurements, (iii) pose or position with IMU measurements, (iv) the absents of control inputs, and (v) changes of the former analysis due to the ill-conditioned real-world quadrotor setup of the later used AscTec Hummingbird quadrotor simulation.

Throughout the analysis we assume significant excitation of all rotors (and thus also of the IMU readings, if present, in all axes). In the experiments, we ensure this by flying Lissajous trajectories that provide sufficient motion while keeping the computation time low (compared to estimation optimized trajectories as shown in [17], [34]).

For the observability analysis, apart of the state's dynamic equations, we use (a subset of) the following IMU and pose/position sensor measurement equations:

$$\mathbf{h}_{imu} = \begin{bmatrix} \mathbf{R}^\mathsf{T}_{MI}\,{}_M\mathbf{a}_{act} + {}_I\mathbf{b}_a + \mathbf{v}_a \\ \mathbf{R}^\mathsf{T}_{MI}\,{}_M\boldsymbol{\omega}_{WM} + {}_I\mathbf{b}_\omega + \mathbf{v}_\omega \end{bmatrix}, \text{with} \tag{9}$$

$${}_M\mathbf{a}_{act} = \tfrac{1}{m}\mathbf{R}^\mathsf{T}_{WM}\,{}_W\mathbf{F}_\Sigma + \left([{}_M\dot{\boldsymbol{\omega}}_{WM}]_\times + [{}_M\boldsymbol{\omega}_{WM}]_\times^{\,2}\right){}_M\mathbf{r}_{MI} \tag{10}$$

$$\mathbf{h}_{pose} = \begin{bmatrix} {}_W\mathbf{r}_{WM} + \mathbf{R}_{WM}\,{}_M\mathbf{r}_{MP} + \mathbf{v}_p \\ \mathbf{q}_{WM} \otimes \mathbf{q}_{MP} \otimes \mathbf{v}_q \end{bmatrix}, \mathbf{v}_q = \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{v}_\theta \end{bmatrix} \tag{11}$$

$\mathbf{v}_\bullet \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_\bullet)$ is the noise of the respective measurement and defined as zero-mean Gaussian white noise.

We conducted a symbolic and numerical rank calculation of the observability matrix as well as a numerical study of its null-space in Matlab.

In order to best condition the numerical parts of the analysis, we used random numbers of the same order of magnitude for all states – $\mathcal{N}(1, 0.01)$. This gives us a general idea of the observability properties. Since real system parameters my have several orders of magnitude difference and thus lead to ill-conditioned situations for real-world state estimation, the analysis of the Asctec Hummingbird quadrotor uses real system parameters (cf. Tab. III or [14]

TABLE I. Observability analysis results of given model with a state vector size of $43 + 7N$ with N number of rotors. The observability matrix $\mathcal{O}$ shows observable (green) and jointly observable (blue) subspaces depending on the sensor configuration with the other states independently unobservable (red).

Measurement	observable dimensions	${}_W\mathbf{r}_{WM}$	${}_M\mathbf{v}_{WM}$	$\mathbf{q}_{WM}$	${}_M\boldsymbol{\omega}_{WM}$	${}_M\mathbf{r}_{MP}$	$\mathbf{q}_{MP}$	${}_M\mathbf{r}_{MI}$	$\mathbf{q}_{MI}$	${}_I\mathbf{b}_a$	${}_I\mathbf{b}_\omega$	m	${}_M\boldsymbol{i}$	${}_W\mathbf{F}_E$	${}_M\mathbf{r}_{ME}$	${}_M\mathbf{r}_{MA_i}$	ψ_{MA_i}	θ_{MA_i}	k_{T_i}	k_{M_i}
Position	$24 + 6N$	ok	ok	ok	ok	ok	unobs.	unobs.	unobs.	unobs.	unobs.	J1	J2	see Tab. II		J2	ok	ok	J1	J2
Pose	$28 + 6N$	ok	ok	ok	ok	ok	ok	unobs.	unobs.	unobs.	unobs.	J1	J2			J2	ok	ok	J1	J2
Position & IMU	$37 + 6N$	ok	ok	ok	ok	ok	unobs.	ok	ok	ok	ok	J1	J2			J2	ok	ok	J1	J2
Pose & IMU	$41 + 6N$	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	J1	J2			J2	ok	ok	J1	J2

for the specific numbers). Also in this realistic analysis, a disturbance of $\|F\|_2 = 2\,\mathrm{N}$ with an offset of $\|r\|_2 = 10\,\mathrm{cm}$ both in a random direction were used. All listed observability properties have been tested for 4, 5, 6, and 8 rotor setups.

1) Pose Measurements Only: In case the estimation process only has access to a global pose measurement sensor, e.g., tracking system, we get a rank of $28 + 6N$ (52 for a quadrotor case) which reflects the number of observable dimensions (states or combination of states). These observable dimensions include the full trajectory state vector $\mathbf{x}_T$ $(3+3+4+3 = 13)$ and pose sensor self-calibration states, ${}_M\mathbf{r}_{MP}$ and $\mathbf{q}_{MP}$, $(3 + 4 = 7)$ as well as the rotor rotation axis components ψ_{MA_i} and θ_{MA_i} ($2N$, 8 in quadrotor case).

Naturally, the IMU self-calibration states, ${}_M\mathbf{r}_{MI}$, $\mathbf{q}_{MI}$, ${}_I\mathbf{b}_a$ and ${}_I\mathbf{b}_\omega$, are unobservable due to the lack of information from an IMU $(3 + 4 + 3 + 3 = 13)$. The remaining states span a $2 + N$ dimensional unobservable sub-space (6 for a quadrotor). This unobservable sub-space can be split into two, $J1$ and $J2$. $J1$ contains m, ${}_W\mathbf{F}_E$, and all thrust force coefficients k_{T_i} and has one dimension regardless of the number of rotors N. The remaining $1 + N$ dimensions are spanned by $J2$ which includes ${}_M\boldsymbol{i}$, ${}_M\mathbf{r}_{ME}$, all vectors ${}_M\mathbf{r}_{MA_i}$, and all drag moment coefficients k_{M_i} (see Tab. I).

The unobservable sets $J1$ and $J2$ can be made observable by supplying *a priori* knowledge which covers the $2 + N$ dimensions. The simplest measurable information for $J1$ is the mass m. For $J2$, easily measurable quantities are the rotor-to-rotor distances $\|{}_M\mathbf{r}_{MA_j} - {}_M\mathbf{r}_{MA_i}\|_2$, $i \neq j \in \mathbb{R}$. Measuring any combination of $1+N$ rotor-to-rotor distances, and including the system's mass renders the estimator fully observable. Note that measuring rotor-to-rotor distances limits the number of rotors for this approach to $N > 3$ since only then permutations allow to reach $1 + N$ measurements.

Note that ${}_W\mathbf{F}_E$ and ${}_M\mathbf{r}_{ME}$ are part of $J1$ and $J2$ respectively (see Tab. II). In case ${}_W\mathbf{F}_E = \mathbf{0}$, ${}_M\mathbf{r}_{ME}$ gets unobservable due to the lack of resulting moment, see Eq. (2), and the force itself is observable (not in $J1$ anymore). The last case is where ${}_W\mathbf{F}_E \neq \mathbf{0}$ and ${}_M\mathbf{r}_{ME} = \mathbf{0}$, which results in ${}_M\mathbf{r}_{ME}$ being observable and ${}_W\mathbf{F}_E$ remains as part of $J1$.

TABLE II. Observability analysis details of the states ${}_W\mathbf{F}_E$ and ${}_M\mathbf{r}_{ME}$. Observable states (green) and jointly observable sub-spaces (blue tones) with the other states being independently unobservable (red).

${}_M\mathbf{r}_{ME}$ \ ${}_W\mathbf{F}_E$	$= \mathbf{0}$	$\neq \mathbf{0}$
$= \mathbf{0}$	${}_W\mathbf{F}_E$: ok ${}_M\mathbf{r}_{ME}$: unobs.	${}_W\mathbf{F}_E$: J1 ${}_M\mathbf{r}_{ME}$: ok
$\neq \mathbf{0}$	${}_W\mathbf{F}_E$: ok ${}_M\mathbf{r}_{ME}$: unobs.	${}_W\mathbf{F}_E$: J1 ${}_M\mathbf{r}_{ME}$: J2

2) Position Measurements Only: If the estimation process is only supplied with absolute position measurements, e.g., global navigation satellite system (GNSS), the analysis shows the same observable and jointly observable states as the case of the pose measurement. The only difference is that the sensor self-calibration state $\mathbf{q}_{MP}$ (pose sensor orientation) is now unobservable and the rank is $24 + 6N$ (48 for a quadrotor configuration). Position updates only use the top entry of Eq. (11), hence, the state $\mathbf{q}_{MP}$ is not present.

Interestingly, the sole position sensor setup can estimate the UAV's world attitude $\mathbf{q}_{WM}$ (given enough movement). This is due to pseudo attitude information resulting from the positional changes. Probably even more noteworthy, with only a global position sensor, the UAV model and rotor-speed system inputs, as well as the mass and $1 + N$ rotor-to-rotor measurements, the entire system is fully observable as an online localization, system identification, sensor suite self-calibration, and disturbance estimation framework.

3) Pose or Position with IMU Measurements: Combining an exteroceptive position sensor with an IMU (ego-motion) is one of the most common setups in localization and navigation tasks, therefore, available on most UAVs.

The combined system with pose senor measurements has a rank of $41 + 6N$ and with only position measurements $37 + 6N$ (65 and 61 for a quadrotor configuration). One can see that the only difference is that the IMU's self-calibration states, ${}_M\mathbf{r}_{MI}$, $\mathbf{q}_{MI}$, ${}_I\mathbf{b}_a$ and ${}_I\mathbf{b}_\omega$, are now observable due to Eq. (9). All other observability properties from Sec. III-C.1 and Sec. III-C.2 hold. Although the IMU has no other influence on the observability of states, having an additional sensor can improve the overall estimation quality.

4) No Control Inputs: This edge case happens if the motors are turned off ($\mathbf{u} = \mathbf{0}$). All states in $\mathbf{x}_{G_1}^\mathsf{T}$ to $\mathbf{x}_{G_N}^\mathsf{T}$ get unobservable, while all other states retain their observability properties. Looking at the external force and its lever arm, this situation allows to detect if the vehicle is on the ground (We show this in Sec. IV-C.2). $J1$ now spans m and ${}_W\mathbf{F}_E$; and $J2$ reduces to 1 dimension spanning over ${}_M\boldsymbol{i}$ and ${}_M\mathbf{r}_{ME}$.

5) Hummingbird Quadrotor Case: As mentioned before, a real system can result in an ill-conditioned estimation problem having different observability properties, thus, one needs to look into the specific configuration in combination with the system's parameters. The Gazebo/RotorS model of the Asctec Humminbird quadrotor including its parameters was used for this analysis.

The given system with pose and IMU measurements available during the estimation gives us a rank of 60 of a maximum of 71. The additional 4 unobservable dimensions, compared to Sec. III-C.3, are a result of the z-aligned rotor axis, see Eq. (2). $J1$ remains even in this configuration. 4 additional unobservable dimensions come from θ_{MA_i} for each rotor. $J2$ only contains 1 (only one rotor-to-rotor distance measurement needed) dimension as the 4 ${}_M\mathbf{r}_{MA_i,z}$ states are by themselves unobservable. These unobservable dimensions scale with the number of rotors N. ${}_W\mathbf{F}_E$ and ${}_M\mathbf{r}_{ME}$ behave the same as in the general case.

D. Disturbance Detection & Classification

The disturbance detection uses the estimates of ${}_W\mathbf{F}_E$ and ${}_M\mathbf{r}_{ME}$ to distinguish between the presence of a disturbance or *none*. A disturbance is further grouped into *force* (e.g., wind gust acting on the UAV), *moment* (e.g., contact with an obstacle), or *ground* contact (e.g., landed UAV) based on the norm values $\|{}_W\mathbf{F}_E\|_2$ and $\|{}_M\mathbf{r}_{ME}\|_2$. Such cases are depicted in the example of the evaluation in Fig. 5.

Note, that the detection only checks the threshold of $\|{}_M\mathbf{r}_{ME}\|_2$ if $\|{}_W\mathbf{F}_E\|_2$ is over a user-defined threshold, this avoids false-positives in case ${}_M\mathbf{r}_{ME}$ is unobservable.

The type *none* is defined as case where both norm values are below their respective thresholds. The cases *force* and *moment* need $\|{}_W\mathbf{F}_E\|_2$ to be over the threshold with $\|{}_M\mathbf{r}_{ME}\|_2$ distinguishing them. The UAV is considered to have *ground* contact if the motor speeds are all zero and the mass-normalize estimate ${}_W\mathbf{F}_E$ is close to $-{}_W\boldsymbol{g}$.

E. C++ Implementation with ROS support

We implemented the estimation as an error-state Kalman filter (ESKF) based on the previously described system model in C++ to allow platform and software suite-independent usage. A Robot Operating System (ROS) node gives the framework the ability to be easily integrated into existing software environments.

Following [28], we calculated a proper discrete covariance propagation matrix $\mathbf{F}$ and system noise covariance $\mathbf{Q}$ through second-order truncation compared to other frameworks. The Measurement Jacobian $\mathbf{H}$ and measurement noise covariance $\mathbf{R}$ are based on the error-state representation.

Compared to our previous Matlab implementation of [14], the accuracy is improved by employing linear interpolation of the control inputs from the time step of the last update to the time step of the current control input.

IV. Experimental Results

We chose to use simulations as they allow for good repeatability of experiments and, therefore, ease of evaluation of the estimator. The simulation environment of choice is Gazebo/RotorS [7] with ROS. It provides us with realistic simulations of multi-rotor UAV physics (in our case, a Asctec Hummingbird quadrotor), control input signals (motor speeds), and sensor measurements with noise.

In the following evaluation, Gazebo/RotorS (and necessary components to automate the latter test series) is running on a notebook (i7-7820HQ CPU), while FUSE-D does its estimations on an Odroid XU4 (one commonly used in the community) connected together via Ethernet/ROS.

A considerable advantage of this method is that the ground truth values of the system are known or can be calculated from other properties, done for the moments of inertia ${}_M\boldsymbol{i}$. We set the individual rotor drag force and rolling moment coefficients (velocity induced hub forces and roll moments) in the RotorS model to zero to reduce errors in the evaluation due to such unmodeled effects. These effects are only present during very fast flights for these types of platforms.

A. Simulation Parameters

In our evaluation, we only update the estimate through pose sensor measurements (published through ROS at $50\,\mathrm{Hz}$) as the IMU makes no additional relevant state, no other than its self-calibration states, observable. One could argue that it improves the overall estimation quality. The pose sensor is assumed to be a tracking system like Optitrack with a zero-mean Gaussian white noise and standard deviations of $\sigma_p = 0.001\,\mathrm{m}$ (position) and $\sigma_\theta = 0.1°$ (attitude). Motor speeds get published at $200\,\mathrm{Hz}$ through ROS, and we assume a motor speed noise of $\sigma_{u_i} = 0.15\,\mathrm{s}^{-1}$ (equals the Astec Hummingbird's FCU rpm quantization) in the estimation.

The noise values of the force ${}_W\mathbf{F}_E$ and its point of application ${}_M\mathbf{r}_{ME}$ are chosen to allow the estimates to react to changes in the acting disturbance but not too high to cause a loss in quality. Hence, we calculate its value based on the expected disturbance slope (rate of change, the physical system has first-order characteristic) and the propagation rate (motor speeds): $\sigma_{F_E} = \frac{1.5\,\mathrm{N}}{\sqrt{200\,\mathrm{Hz}}} = 0.15\,\mathrm{N}$ and $\sigma_{r_{ME}} = \frac{0.7\,\mathrm{m}}{\sqrt{200\,\mathrm{Hz}}} = 0.05\,\mathrm{m}$, respectively.

The initial guess of the state values were $20\,\%$ offset to verify that the state is indeed observable. Further ground truth system-parameters of the UAV can be found in Tab. III.

B. Test Case

We tested the observability of Sec. III-C and the performance of the C++ implementation of FUSE-D from Sec. III-E empirically through a combination of five different $120\,\mathrm{s}$ long Lissajous trajectories and six different disturbance-sets which results in 30 test runs in total. These five Lissajous figure-based trajectories are similar to the ones used in [14] (combining a low-frequency high-velocity and a high-frequency low-velocity motion) and give us sufficient excitation in all 6 DoF allowing good convergence of even poorly observable states. The six disturbance-sets contain one force (point of application in CoM) and one moment (force and offset point of application) disturbance each, starting at $90\,\mathrm{s}$ and $105\,\mathrm{s}$ after the trajectory start with a duration of approximately $7\,\mathrm{s}$. The magnitudes for force and point of application are chosen randomly in the range of $\mathcal{U}(1,4)\mathrm{N}$ and $\mathcal{U}(0.1, 0.25)\mathrm{m}$, respectively, with arbitrary orientation.

Every test run starts with the UAV on the ground, followed by a takeoff and flight towards the trajectory's starting point. Reaching this point starts the Lissajous trajectory following

TABLE III. Gazebo/RotorS AscTec Hummingbird model parameters (ground truth) with estimation error and standard deviation of the fully observable state vector based on 30 test runs after 90 s flight time. $\mathbf{q}_{MP,z}$, m, ${}_M\mathbf{r}_{MA_i,z}$, and θ_{MA_i} are not listed as those are assumed known *a priori*.

	ground truth				error				standard deviation			
	x/roll	y/pitch/value	z/yaw	Unit	x/roll/1	y/pitch/2	z/yaw/3	4	x/roll/1	y/pitch/2	z/yaw/3	4
${}_M\mathbf{r}_{MP}$	$2.6 \cdot 10^{1}$	$3.8 \cdot 10^{1}$	$5.9 \cdot 10^{1}$	mm	1.6	3.1	−2.7		1.4	1.8	$7.3 \cdot 10^{-1}$	
$\mathbf{q}_{MP}$	0.0	0.0	0.0	°	$6.0 \cdot 10^{-2}$	$5.4 \cdot 10^{-1}$	known		$6.6 \cdot 10^{-2}$	$6.0 \cdot 10^{-1}$	known	
${}_M\boldsymbol{i}$	$7.5 \cdot 10^{-3}$	$7.5 \cdot 10^{-3}$	$1.3 \cdot 10^{-2}$	$\mathrm{kg\,m^2}$	$1.4 \cdot 10^{-4}$	$8.1 \cdot 10^{-5}$	$3.0 \cdot 10^{-4}$		$1.8 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	
${}_W\mathbf{F}_E$	0.0	0.0	0.0	$\mathrm{m/s^2}$	$-2.8 \cdot 10^{-3}$	$2.1 \cdot 10^{-2}$	$6.3 \cdot 10^{-2}$		$8.7 \cdot 10^{-2}$	$8.2 \cdot 10^{-2}$	$6.9 \cdot 10^{-2}$	
${}_M\mathbf{r}_{ME}$	0.0	0.0	0.0	$\mathrm{m/s^2}$	$3.5 \cdot 10^{-3}$	$-1.1 \cdot 10^{-3}$	$-1.0 \cdot 10^{-2}$		$1.2 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$	
${}_M\mathbf{r}_{MA_1}$	$1.7 \cdot 10^{2}$	0.0	$1.1 \cdot 10^{1}$	mm	2.6	$-1.5 \cdot 10^{-1}$	known		7.8	$8.9 \cdot 10^{-1}$	known	
${}_M\mathbf{r}_{MA_2}$	0.0	$1.7 \cdot 10^{2}$	$1.1 \cdot 10^{1}$	mm	$9.9 \cdot 10^{-1}$	$2.5 \cdot 10^{1}$	known		1.0	$1.0 \cdot 10^{1}$	known	
${}_M\mathbf{r}_{MA_3}$	$-1.7 \cdot 10^{2}$	0.0	$1.1 \cdot 10^{1}$	mm	4.4	$9.7 \cdot 10^{-2}$	known		7.6	$9.2 \cdot 10^{-1}$	known	
${}_M\mathbf{r}_{MA_4}$	0.0	$-1.7 \cdot 10^{2}$	$1.1 \cdot 10^{1}$	mm	$-7.6 \cdot 10^{-1}$	5.7	known		$8.6 \cdot 10^{-1}$	$1.0 \cdot 10^{1}$	known	
$\psi_{MA_{1\dots4}}$		0.0		°	$1.5 \cdot 10^{-1}$	1.4	$8.6 \cdot 10^{-1}$	$-1.7 \cdot 10^{-2}$	1.4	1.3	1.3	$9.5 \cdot 10^{-1}$
$k_{T_{1\dots4}}$		$3.4 \cdot 10^{-4}$		$\mathrm{N/s^{-2}}$	$-2.7 \cdot 10^{-6}$	$-3.4 \cdot 10^{-5}$	$1.1 \cdot 10^{-5}$	$2.1 \cdot 10^{-5}$	$1.7 \cdot 10^{-5}$	$1.5 \cdot 10^{-5}$	$1.5 \cdot 10^{-5}$	$1.7 \cdot 10^{-5}$
$k_{M_{1\dots4}}$		$1.6 \cdot 10^{-2}$		m	$4.3 \cdot 10^{-4}$	$4.5 \cdot 10^{-4}$	$-7.4 \cdot 10^{-5}$	$-3.1 \cdot 10^{-3}$	$7.4 \cdot 10^{-4}$	$3.5 \cdot 10^{-3}$	$7.7 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$

and resets FUSE-D, just before starting the Lissajous trajectory, to its initial estimate $\hat{\mathbf{x}}_0$, guaranteeing the same starting conditions for each run. The first 90 s (transient phase) serve the purpose of convergence evaluation and observability validation. After 90 s, the disturbance force acting on the CoM is applied to the UAV (emulating, e.g., a wind gust) followed by the same force offset with respect to the CoM (moment, emulating a potential contact or collision) at 105 s. The trajectory ends at 120 s and the control lands the UAV 5 s later. Thus, the ground contact is at around 125 s.

C. Evaluation & Discussion

The following evaluation and discussion of the test run results is based on the mean and standard deviation values over all 30 test runs, except Fig. 5 highlighting the disturbance estimation and classification on an example.

1) Transient Phase: The transient phase of the estimation can be thought of as the time it takes the estimator to converge towards a steady-state, in which the estimates will not change drastically anymore. The data in Fig. 4 indicates that all states converge towards appropriate values close to ground truth after 60 s, even with wrong initial estimate $\hat{\mathbf{x}}_0$, confirming the theoretical observability analysis result of Sec. III-C. All error and standard deviation values of all estimates in $\hat{\mathbf{x}}$ at 90 s are reported in Tab. III, except $\mathbf{q}_{MP,z}$, m, ${}_M\mathbf{r}_{MA_i,z}$, and θ_{MA_i}, which are known at the start of the estimation process ($\|{}_M\mathbf{r}_{MA_1} - {}_M\mathbf{r}_{MA_3}\|_2$ for $J2$ is a combination of two states) – see Sec. III-C.5 for the corresponding observability discussion.

The pose sensor self-calibration states, ${}_M\mathbf{r}_{MP}$ and $\mathbf{q}_{MP}$, converge fast and accurate towards ground truth with maximal 8 % positional relative error (and an absolute error in orientation of below 1°). A higher standard deviation of the pose sensor's y-axis orientation calibration state may be due to less exciting motions around this axis.

The lower angular velocity value changes of ${}_M\boldsymbol{\omega}_{WM}$ around the z-axis (low quadcopter yawing) cause a lower quality of the estimate ${}_M\boldsymbol{i}_z$. This can be seen in the bigger z-component error and higher standard deviation. We achieve a high grade of accuracy with not more than 3 % relative error compared to ground truth.

In the undisturbed section of the test flight, it is visible that the force ${}_W\mathbf{F}_E$ gets estimated correctly with an norm

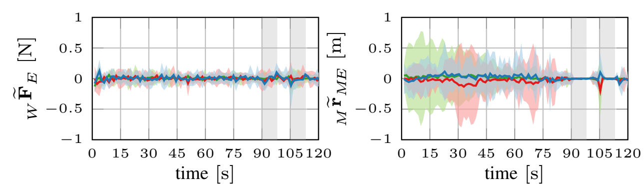

Fig. 3. Estimation errors ($\widetilde{\square}$) of external force ${}_W\widetilde{\mathbf{F}}_E$ and point of application ${}_M\widetilde{\mathbf{r}}_{ME}$ based on 30 Lissajous trajectory flights. The solid lines are the error, and the shaded areas are the 3σ standard deviation over all test runs. Axis x (blue), y (red), and z (green). Gray regions mark the time in which disturbances act. The first one is a force acting on the CoM, and the second is the same force with an offset point of application.

average root mean square error (ARMSE) over the whole 90 s section of 0.11 N. Contrary to the insight of the observability analysis, in an undisturbed case, the point of application ${}_M\mathbf{r}_{ME}$ should be unobservable. We always see a small deviation of the force from this zero line due to uncertainties in the simulation as well as the estimation. Hence ${}_M\mathbf{r}_{ME}$ gets observable with a norm ARMSE of 12.9 mm but exhibits a higher standard deviation.

The inclination angles ψ_{MA_i} can deviate a bit more as they are in the sine components of the axis definition.

Note that the rotor position vectors ${}_M\mathbf{r}_{MA_i}$ exhibit different accuracies depending on if they are part of the *a priori* rotor-to-rotor distance measurement to make $J2$ observable. In this case, ${}_M\mathbf{r}_{MA_2}$ and ${}_M\mathbf{r}_{MA_4}$ show higher relative errors of up to 15 %, while ${}_M\mathbf{r}_{MA_2}$ and ${}_M\mathbf{r}_{MA_4}$ maximum 3 %, which is due to the additional information these "constraints" supply.

All rotor thrust force coefficients k_{T_i} show good convergence with a worst-case 10 % error. The drag moment coefficients k_{M_i} exhibit a higher standard deviation because of the limited yaw motions resulting in an error below 20 %.

2) Disturbance Detection & Classification: The disturbance detection module of our proposed framework applied a smoothing to the estimates and its tests according to Sec. III-D have a success rate of approximately 80 %. The outliers of the detection were of two types: (i) ground contact was mistaken for a moment (happened two times), and (ii) bouncing between the force and moment classes (happened four times). The first error mode happens if the UAV has brief ground contact at a tilted angle – correctly classified as a

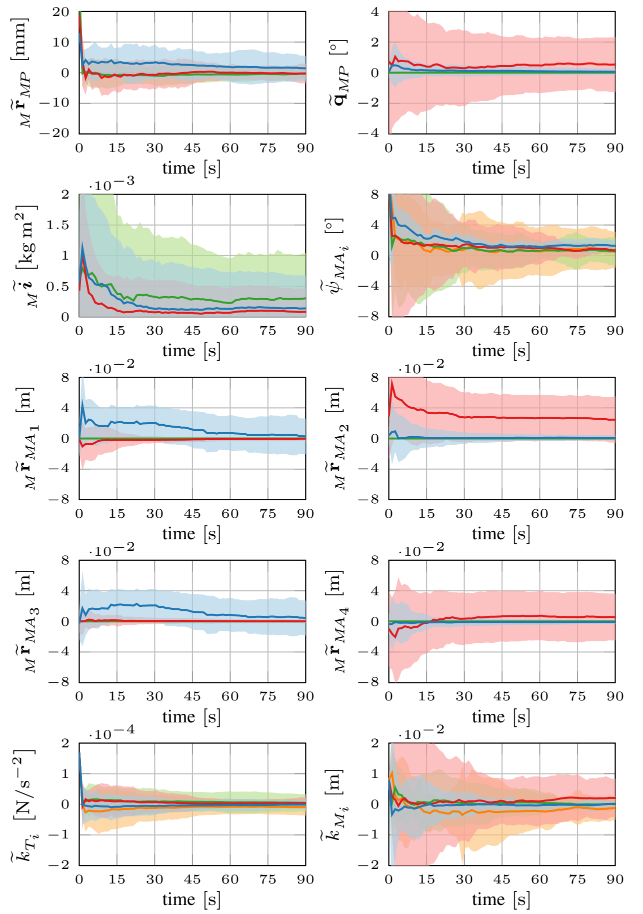

Fig. 4. Convergence of estimated sensor, inertial, and aerodynamic parameters based on the first 90 s of the 30 Lissajous trajectory flights. Known values of m and θ_{MA_i} are omitted. The solid lines are the error ($\tilde{\square}$) and the shaded areas the 3σ standard deviation over 30 test runs. Axis x (blue), y (red), z (green) or rotor 1 (blue), 2 (red), 3 (green), 4 (orange). Although the initial estimate is 20 % wrong, all states converge within 60 s to appropriate values close to ground truth.

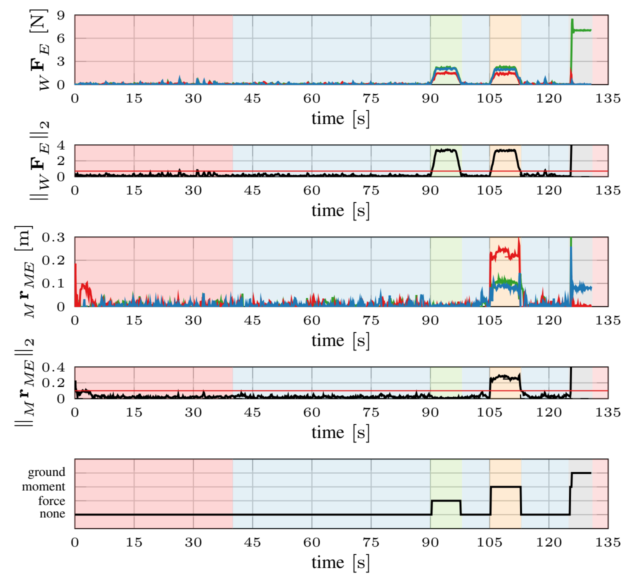

Fig. 5. Example of the disturbance detection and classification based on FUSE-D estimated force and point of application. The solid lines are the estimate ($\hat{\square}$), and the dashed lines are the ground truth of the respective state. Axis x (blue), y (red), z (green), and scalar values (black). A threshold of 0.7 N and 0.1 m for the norm of $\|{}_W\hat{\mathbf{F}}_E\|_2$ and $\|{}_M\hat{\mathbf{r}}_{ME}\|_2$ was used. The colored background depicts the ground truth of the disturbance acting on the UAV. (red) background shows ignored sections of the time-series, (blue) areas mark disturbance-free sections, with (green), (orange), and (grey) areas highlighting a disturbance force (e.g., wind gust), moment (e.g., collision), and ground contact, respectively. As can be seen in the bottom row of this example, the disturbance detection classifies all cases correctly.

moment – but then turns off the motors due to test automation settings. The second error is caused by the norm $\|{}_M\mathbf{r}_{ME}\|_2$ and the predefined thresholds not including a hysteresis.

Fig. 5 shows a test run in detail with the estimates ${}_W\mathbf{F}_E$ and ${}_M\mathbf{r}_{ME}$, their respective norms, and the classification based on them. It can be seen that the landing gets classified as *moment* for a brief moment due to a slight tilt during landing, then it switches to the correct *ground* class.

Nonetheless, the classification achieves good performance and provides valuable data to potential higher level planning.

3) Computation time Odroid XU4: The execution time of each ESKF step (rps-based propagation and pose sensor based update) of the in Sec. III-E discussed implementation has been logged during each experiment. We gathered 10000 samples of each step. In numbers, in the worst case, the worst propagation rate we achieve is approximately 1.4 kHz (mean 0.160 ms, max. 0.693 ms, and min. 0.088 ms) and pose update rate of 685 Hz (mean 0.390 ms, max. 1.459 ms, and 0.256 ms). These timings allow for the estimation to be done online in real-time, potentially supplying controls and higher level decision making during task execution. `htop` reports around 43 % CPU utilization of the fuse_node on the Odroid XU4. Timings and CPU utilization might differ depending on the number of rotors N and hardware used.

V. Conclusion

Our proposed Framework for UAV System-Parameter Estimation with Disturbance Detection (FUSE-D) combines system-parameter identification, sensor suite self-calibration, and navigation state estimation with disturbance detection and classification in a holistic online approach.

We have devised a system model, including a disturbance force acting with a possible offset point of application, of an N-rotor multi-rotor UAV that uses rotor-speed input and position or pose (3D position and potentially 3D attitude) sensor readings to estimate system-parameters and estimate disturbances for further high-level planning or control use. A thorough observability analysis of the system model generalized to N rotors adds to our understanding of which system-parameters or types of disturbances can be estimated. Additional analysis of the observability matrix's null space reveals what *a priori* information needs to be supplied to the estimation process to make the system fully observable.

A realistic simulated case study of the well-known Asctec Hummingbird in Gazebo/RotorS has shown that the insights of the observability analysis are valid. The UAV feeds its motor speeds and pose sensor measurements to FUSE-D running online on an Odroid XU4 (connected via ROS to the Gazebo simulation). We have managed to obtain accurate estimation and classification results that emphasize the usage of FUSE-D in future applications. The disturbance detection module successfully classifies disturbances based on the estimates whether the disturbance is *none*, *force* (e.g., wind), *moment* (e.g., collision), or *ground* contact (e.g., landed).

Further investigation seeks to deploy the proposed approach to closed-loop flown multi-rotor platforms to show the versatility and robustness of FUSE-D.

References

[1] M. Kamel, S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, Z. Taylor, R. Siegwart, and I. Gilitschenski, "The Voliro Omniorientational Hexacopter: An Agile and Maneuverable Tiltable-Rotor Aerial Vehicle," *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 34–44, October 2018.

[2] D. Falanga, K. Kleber, S. Mintchev, D. Floreano, and D. Scaramuzza, "The Foldable Drone: A Morphing Quadrotor That Can Squeeze and Fly," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 209–216, December 2018.

[3] P. Zheng, X. Tan, B. B. Kocer, E. Yang, and M. Kovac, "TiltDrone: A Fully-Actuated Tilting Quadrotor Platform," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6845–6852, July 2020.

[4] M. Ryll, D. Bicego, and A. Franchi, "Modeling and Control of FAST-Hex: a Fully–Actuated by Synchronized–Tilting Hexarotor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, October 2016, pp. 1689–1694.

[5] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, Modeling, Estimation and Control for Aerial Grasping and Manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2011, pp. 2668–2673.

[6] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System," in *Robot Operating System (ROS): The Complete Reference (Volume 2)*. Springer International Publishing, May 2017, pp. 3–39.

[7] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "RotorS - A Modular Gazebo MAV Simulator Framework," in *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Springer International Publishing, February 2016, pp. 595–625.

[8] D. Tzoumanikas, F. Graule, Q. Yan, D. Shah, M. Popović, and S. Leutenegger, "Aerial Manipulation Using Hybrid Force and Position NMPC Applied to Aerial Writing," in *Proceedings of Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, July 2020.

[9] M. Burri, J. Nikolic, H. Oleynikova, M. W. Achtelik, and R. Siegwart, "Maximum Likelihood Parameter Identification for MAVs," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4297–4303.

[10] M. Burri, M. Bloesch, Z. Taylor, R. Siegwart, and J. Nieto, "A framework for maximum likelihood parameter identification applied on MAVs," *Journal of Field Robotics (JFR)*, vol. 35, no. 1, pp. 5–22, January 2018.

[11] D. Six, S. Briot, J. Erskine, and A. Chriette, "Identification of the Propeller Coefficients and Dynamic Parameters of a Hovering Quadrotor From Flight Data," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 1063–1070, January 2020.

[12] M. Dhaybi and N. Daher, "Accurate Real-time Estimation of the Inertia Tensor of Package Delivery Quadrotors," in *2020 American Control Conference (ACC)*, July 2020, pp. 1520–1525.

[13] C. Böhm, C. Brommer, A. Hardt-Stremayr, and S. Weiss, "Combined System Identification and State Estimation for a Quadrotor UAV," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, June 2021, pp. 585–591.

[14] C. Böhm, M. Scheiber, and S. Weiss, "Filter-Based Online System-Parameter Estimation for Multicopter UAVs," in *Proceedings of Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, July 2021.

[15] V. Wuest, V. Kumar, and G. Loianno, "Online Estimation of Geometric and Inertia Parameters for Multirotor Aerial Vehicles," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2019, pp. 1884–1890.

[16] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, "IMU-Based Inertia Estimation for a Quadrotor Using Newton-Euler Dynamics," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 3, pp. 3861–3867, February 2020.

[17] C. Böhm, G. Li, G. Loianno, and S. Weiss, "Observability-Aware Trajectories for Geometric and Inertial Self-Calibration," in *Power On and Go Robots 2020, RSS'20, (Virtual) Workshop*, July 2020.

[18] A. Shastry and D. A. Paley, "Uav state and parameter estimation in wind using calibration trajectories optimized for observability," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1801–1806, November 2020.

[19] B. P. Putra, A. I. Cahyadi, and I. Ardiyanto, "Quadrotor inertial navigation aided by vehicle dynamic model: A nonlinear observability analysis," in *2022 14th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE, October 2022, pp. 42–47.

[20] C. Chen, Y. Yang, P. Geneva, W. Lee, and G. Huang, "Visual-inertial-aided online mav system identification," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, October 2022, pp. 6277–6284.

[21] S. I. Azid, K. Kumar, M. Cirrincione, and A. Fagiolini, "Robust Motion Control of Nonlinear Quadrotor Model With Wind Disturbance Observer," *IEEE Access*, vol. 9, pp. 149 164–149 175, November 2021.

[22] J. Cayero, D. Rotondo, B. Morcego, and V. Puig, "Optimal state observation using quadratic boundedness: Application to uav disturbance estimation," *International Journal of Applied Mathematics and Computer Science*, vol. 29, no. 1, pp. 99–109, March 2019.

[23] K. Guo, J. Jia, X. Yu, L. Guo, and L. Xie, "Multiple observers based anti-disturbance control for a quadrotor uav against payload and wind disturbances," *Control Engineering Practice*, vol. 102, p. 104560, September 2020.

[24] D. Hentzen, T. Stastny, R. Siegwart, and R. Brockers, "Disturbance estimation and rejection for high-precision multirotor position control," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, November 2019, pp. 2797–2804.

[25] J. Jia, K. Guo, X. Yu, L. Guo, and L. Xie, "Agile flight control under multiple disturbances for quadrotor: Algorithms and evaluation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 4, pp. 3049–3062, January 2022.

[26] G. Hattenberger, M. Bronz, and J.-P. Condomines, "Estimating wind using a quadrotor," *International Journal of Micro Air Vehicles*, vol. 14, p. 17568293211070824, January 2022.

[27] L. Sikkel, G. C. de Croon, C. De Wagter, and Q. Chu, "A novel online model-based wind estimation approach for quadrotor micro air vehicles using low cost mems imus," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, October 2016, pp. 2141–2146.

[28] J. Solà, "Quaternion kinematics for the error-state Kalman filter," *arXiv e-prints, arXiv:1711.02508*, November 2017.

[29] P. Martin and E. Salaün, "The True Role of Accelerometer Feedback in Quadrotor Control," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 1623–1629.

[30] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on Automatic Control (TAC)*, vol. 22, no. 5, pp. 728–740, October 1977.

[31] A. J. Krener and K. Ide, "Measures of Unobservability," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC)*, December 2009, pp. 6401–6406.

[32] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," *The International Journal of Robotics Research (IJRR)*, vol. 30, no. 1, pp. 56–79, January 2011.

[33] A. Martinelli, "Continuous Symmetries and Observability Properties in Autonomous Navigation," INRIA, Research Report RR-7049, October 2010.

[34] C. Böhm, P. Brault, Q. Delamare, P. R. Giordano, and S. Weiss, "COP: Control & Observability-aware Planning," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, June 2022, pp. 3364–3370.