This paper has been accepted for publication at the Robotics: Science and Systems conference (RSS), Daegu, 2023.

HDVIO: Improving Localization and Disturbance Estimation with Hybrid Dynamics VIO

Giovanni Cioffi*, Leonard Bauersfeld*, Davide Scaramuzza

***Abstract*—Visual-inertial odometry (VIO) is the most common approach for estimating the state of autonomous micro aerial vehicles using only onboard sensors. Existing methods improve VIO performance by including a dynamics model in the estimation pipeline. However, such methods degrade in the presence of low-fidelity vehicle models and continuous external disturbances, such as wind. Our proposed method, HDVIO, overcomes these limitations by using a hybrid dynamics model that combines a point-mass vehicle model with a learning-based component that captures complex aerodynamic effects. HDVIO estimates the external force and the full robot state by leveraging the discrepancy between the actual motion and the predicted motion of the hybrid dynamics model. Our hybrid dynamics model uses a history of thrust and IMU measurements to predict the vehicle dynamics. To demonstrate the performance of our method, we present results on both public and novel drone dynamics datasets and show real-world experiments of a quadrotor flying in strong winds up to 25 km/h. The results show that our approach improves the motion and external force estimation compared to the state-of-the-art by up to 33% and 40%, respectively. Furthermore, differently from existing methods, we show that it is possible to predict the vehicle dynamics accurately while having no explicit knowledge of its full state.**

Supplementary Material

A narrated video illustrating our approach is available at: https://youtu.be/CrnINdJS3s4

I. Introduction

Visual-inertial odometry (VIO) has become the de-facto standard for state estimation of consumer and inspection drones. To improve the performance of the VIO pipeline, multiple approaches that tightly couple the drone dynamics in VIO systems have been recently proposed [1, 2, 3]. Including the system dynamics in the VIO formulation brings in new information, which allows the VIO system to distinguish between motion due to actuation and motion due to perturbations (external forces). This results in an increased accuracy of the pose estimates and the possibility to estimate an external force acting on the robot.

Despite working well in many situations, the performance of state-of-the-art methods degrades drastically if the model mismatch is large (high speeds, systematic noise in the actuation inputs) or if continuous external disturbances are present (continuous wind). This is because their simplifying assumptions—no aerodynamic drag and zero-mean noise in the system dynamics—no longer hold. Simply reusing a state-of-the-art dynamics model [4, 5] inside a VIO pipeline is difficult as the dynamics model should only depend on measurements and not rely on the full robot state, otherwise, a feedback loop is introduced where the VIO output influences the system model which then, in turn, affects the VIO.

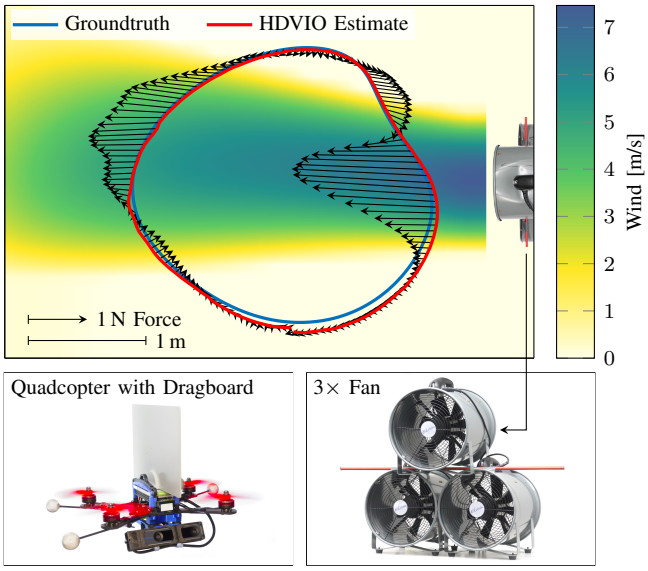


Fig. 1: A quadrotor with a dragboard attached is flown on a circular trajectory through a wind field generated with three industrial fans. Our HDVIO is used to estimate the position of the drone (shown in red) and the external disturbance force (black arrows) acting on the vehicle. The ground-truth position of the vehicle is shown in blue.

Addressing these limitations enables the deployment of model-based VIO estimators in applications where aerodynamic effects are significant, such as during fast flights [6] and in windy conditions [7], or in cases where modeling inaccuracies are present.

The state-of-the-art approach VIMO [1] integrates the drone dynamics in an optimization-based VIO [8] system by defining a new residual term derived from the propagation of the drone model. The drone dynamics are based on a point-mass model that neglects aerodynamic effects, leading to the aerodynamic drag being estimated as part of the external force. Another limitation of VIMO is that any systematic offset in the actuation inputs (vehicle miscalibration, such as wrong rotor lift coefficients) is estimated as an accelerometer bias. This leads to erroneous information being introduced into the inertial residuals, resulting in reduced motion estimation

*Equal contribution.

The authors are with the Robotics and Perception Group, Department of Informatics, University of Zurich, Switzerland, http://rpg.ifi.uzh.ch.

This work was supported by the National Centre of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation (SNSF) and the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 871479 (AERIAL-CORE) and the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT).

accuracy.

All the above limitations can be addressed by improving the dynamics model of the drone included in the VIO estimator. Current methods to estimate high-fidelity drone models [4, 9, 10] typically require knowledge of the full drone state, including velocity and attitude. However, the full state is part of the VIO output. Consequently, the prediction of the drag force would depend on the estimator state, introducing a feedback loop that can lead to a diverging estimator.

Contribution

We present HDVIO, the first VIO pipeline, which uses a neural network to refine the drone dynamics model. In contrast to prior work focusing on drone modeling [4], our learned dynamics model does not require knowledge of the full drone state (for example, velocity). Instead, by using a temporal convolutional network [11], it only needs thrust (commands or measurements) and angular velocities from a gyroscope to estimate the aerodynamic forces. We integrate our hybrid drone model into an optimization-based VIO system [12, 13], and leverage the preintegration theory [14] to efficiently compute dynamics residuals between consecutive camera frames. The dynamics residuals are optimized together with monocular camera and IMU residuals in the VIO backend.

We evaluate our method against the same VIO system without the proposed hybrid dynamics model and also against VIMO [1] in multiple experiments using the public real-world datasets *Blackbird* [15] and *VID* [16]. The results show that our method overcomes the limitations of current state-of-the-art methods and estimates the robot states and the external force more accurately, achieving up to 33% and 40% improvement, respectively. Furthermore, we evaluate the performance of HDVIO in a set of experiments where the drone is flown in a known wind field as shown in Fig. 1. Our method is able to outperform VIMO in the prediction of the external force due to wind.

We also evaluate the accuracy of the learned dynamics model on the *NeuroBEM* [4] dataset, where it provides competitive performance compared to state-of-the-art aerodynamics models. This, for the first time, shows that the forces acting on the vehicle can be predicted without access to its full state, i.e., without knowing its ground-truth linear and angular velocity. Furthermore, to the best of our knowledge, our learning-based model is the first data-driven dynamics model that requires no ground-truth force measurements for training but only position and velocity supervision signals. We show that this removes the need for a motion-capture system to record training data, as simultaneous localization and mapping (SLAM) methods are sufficiently accurate to estimate the position and velocity ground truth for training.

By having both a precise VIO pipeline and an accurate estimate of the external force, we believe that this work is a stepping stone towards the use of *autonomous* drones in safety-critical applications like surveying a disaster site and public air transport[1] which, to date, still require a human pilot.

[1]https://www.volocopter.com/newsroom/first-crewed-evtol-flight/

II. Related Work

Works on visual-inertial odometry with external force estimation can be separated into two groups: loosely-coupled and tightly-coupled methods. In the first category, the external force estimation for flying robots is decoupled from the motion estimation [17, 18, 19, 20, 21, 22], whereas tightly-coupled approaches propose to simultaneously estimate the motion of the robot and the external perturbance.

A. Loosely-Coupled Methods

Initially, all developed methods [17, 18, 19] were based on deterministic approaches. They propose nonlinear force and torque observers derived from the robot dynamics model and assume that the estimates of the robot state are available from another estimator.

Differently from these methods, the probabilistic approaches proposed in [20, 21, 22, 23] account for the sensor noise and, consequently, achieve increased accuracy. They are based on the Extended Kalman Filter (EKF) [21, 23] and the Unscented Kalman Filter (UKF) [20, 22]. The work in [3] uses the quadrotor model to update an EKF-based VIO estimator [24] in order to perform online system identification as well as state estimation. They show that in the case of noisy dynamics measurements, the best solution is to decouple the estimation of the state variables from the measurement update based on the quadrotor dynamics. The approach in [25] uses a UKF to estimate external disturbances such as wind and interactions with humans. The filter is updated with the output of a neural network that processes airflow measurements from a bio-inspired airflow sensor and pose measurements from a motion capture system.

As loosely-coupled approaches neglect the correlations among the estimated variables and their noise characteristic, they suffer from a decreased performance unless the signal-to-noise ratio of the sensor data is very high.

B. Tightly-Coupled Methods

To overcome the limitations of loosely-coupled methods, more recently [1, 2, 26] propose tightly-coupled approaches to simultaneously estimate the motion of the robot and the external perturbance. VIMO [1] is the first work that tightly couples the robot dynamics in an optimization-based VIO system [8]. The main contribution is the addition of a residual term that represents a motion constraint based on the robot dynamics, including external forces, to the VIO problem formulation. The derivation of this dynamic residual term is inspired by the IMU preintegration theory [14]. In this case, high-rate thrust inputs are pre-integrated, resulting in residual terms between consecutive camera frames. The external force is modeled as a zero-mean Gaussian variable since its dynamics are unknown. In this way, VIMO jointly estimates the external force in addition to the robot state. However as discussed in Sec. I, this method is subject to limitations when the external force is continuous or there is a model mismatch.

The more recent method, VID-Fusion [2], proposes an algorithm very similar to VIMO where only the model of

the external force is different. In VID-Fusion the mean of the Gaussian distribution used to represent the external force is equal to the average difference between the accelerometer and thrust measurements in the preintegration window. An extension of VIMO for legged robots is proposed in [26].

C. Drone Modeling

At the core of our HDVIO approach, we need an accurate model of the drone dynamics. Prior work exclusively addresses this in a setting where the vehicle state is available. In a VIO pipeline, this would introduce a feedback loop inside the estimator and such a dynamics model is not suitable. Nevertheless, a brief review of quadrotor modeling literature is presented for completeness.

Often, quadrotors are modeled as a simple rigid body with mass and inertia. In this model, the robot can only exert a force in the body-z direction and has either no aerodynamic drag or linear drag [9, 27, 28, 29]. Such basic models can be refined based on first-principles, leading to blade-element-momentum (BEM) theory [4, 10, 30, 31]. On the other hand, purely data-driven models have been developed [5] which typically outperform first-principle based methods as quadrotor aerodynamics are highly complex. The state-of-the-art model, NeuroBEM [4], combines a physical model and a learning-based component. The success of such a method has inspired us to use a learned component in our HDVIO for the drone dynamics.

III. Methodology

In this section, we describe our visual-inertial-hybrid drone dynamics odometry algorithm. First, we introduce the notation used throughout the paper and the drone dynamics. We focus our derivation on a quadrotor platform, however, our approach could be extended to any other robotic platform. Second, we formulate the estimation problem. Third, we present a concise derivation of the dynamics residual term. This derivation is based on the preintegration theory [14] and is also used in VIMO. Last, we present our learning-based module.

A. Notation & Quadrotor Dynamics

Throughout this paper, scalars are denoted in non-bold $[s, S]$, vectors in lowercase bold $\boldsymbol{v}$, and matrices in uppercase bold $\boldsymbol{M}$. World $\mathcal{W}$, Body $\mathcal{B}$, IMU $\mathcal{I}$, and camera $\mathcal{C}$ frames are defined with an orthonormal basis, i.e. $\{\boldsymbol{x}^{\mathcal{W}}, \boldsymbol{y}^{\mathcal{W}}, \boldsymbol{z}^{\mathcal{W}}\}$. The frame $\mathcal{B}$ is located at the center of mass of the quadrotor. For simplicity, the IMU frame $\mathcal{I}$ is assumed to be the same as $\mathcal{B}$.

We use the notation $(\cdot)^{\mathcal{W}}$ to represent a quantity in the world frame. A similar notation is utilized for every reference frame. The position, orientation, and velocity of $\mathcal{B}$ with respect to $\mathcal{W}$ at time t_k are written as $\boldsymbol{p}^{\mathcal{W}}_{\mathcal{B}_k} \in \mathbb{R}^3$, $\boldsymbol{R}^{\mathcal{W}}_{\mathcal{B}_k} \in \mathbb{R}^{3\times3}$ part of the rotation group $SO(3)$, and $\boldsymbol{v}^{w}_{\mathcal{B}_k} \in \mathbb{R}^3$, respectively. The unit quaternion represenation of $\boldsymbol{R}^{\mathcal{W}}_{\mathcal{B}_k}$ is written as $\boldsymbol{q}^{\mathcal{W}}_{\mathcal{B}_k}$. The accelerometer and gyroscope bias are written as $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$, respectively. The gravity vector in the world frame is written as $\boldsymbol{g}^w$. We use the symbol $\hat{\cdot}$ to indicate noisy measurements.

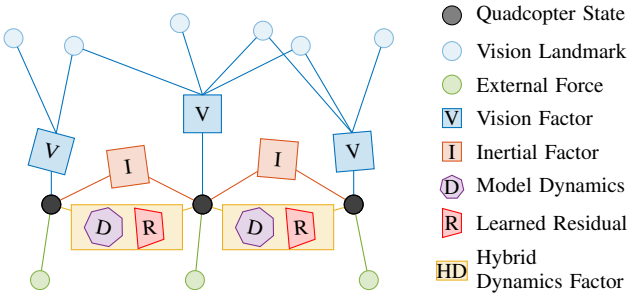


Fig. 2: Factor graph representation of HDVIO with visual, inertial, and hybrid dynamics factors.

The quadrotor is assumed to be a point of mass m. The evolution of the position and velocity of the quadrotor platform is described by the following model:

$$\dot{\boldsymbol{p}}^{\mathcal{W}}_{\mathcal{B}_k} = \boldsymbol{v}^{\mathcal{W}}_{\mathcal{B}_k}, \quad \dot{\boldsymbol{v}}^{\mathcal{W}}_{\mathcal{B}_k} = \boldsymbol{R}^{\mathcal{W}}_{\mathcal{B}_k}(\boldsymbol{f}^{\mathcal{B}}_{t_k} + \boldsymbol{f}^{\mathcal{B}}_{res_k} + \boldsymbol{f}^{\mathcal{B}}_{e_k}) + \boldsymbol{g}^{w}, \tag{1}$$

where $\boldsymbol{f}^{\mathcal{B}}_{t_k} = [0, 0, T_k]^\top$ is the mass-normalized collective thrust and $\boldsymbol{f}^{\mathcal{B}}_{e_k}$ is the external force acting on the quadrotor platform. To account for aerodynamic effects and unknown systematic noise in the thrust inputs, we introduce a residual term $\boldsymbol{f}^{\mathcal{B}}_{res_k}$. We will drop the term mass-normalized when referring to the collective thrust hereafter for the sake of conciseness. The external force is assumed to be a random variable distributed according to a zero-mean Gaussian distribution. As pointed out in [1], this allows the estimator to distinguish between the slowly changing accelerometer bias and the incidental external forces. The dynamics motion constraint is derived by leveraging the preintegration theory [14]. This requires the separation of the residual terms dependent on optimization variables from the terms dependent on the measurements. The rotational dynamics of the quadrotor are not considered here because the torque inputs cannot be separated from their dependency on the robot orientation. Instead, the evolution of the orientation of the quadrotor is obtained from the gyroscope rotation model:

$$\dot{\boldsymbol{q}}^{\mathcal{W}}_{\mathcal{B}_k} = \frac{1}{2}\boldsymbol{q}^{\mathcal{W}}_{\mathcal{B}_k} \otimes [0, \boldsymbol{\omega}^{\mathcal{B}_k}]^\top, \tag{2}$$

where $\otimes$ is the quaternion product and $\hat{\boldsymbol{\omega}}^{\mathcal{B}_k} = \boldsymbol{\omega}^{\mathcal{B}_k} + \boldsymbol{b}_{\omega_k} + \boldsymbol{n}_\omega$ is the gyroscope measurement. The gyroscope noise is modeled as additive Gaussian noise $\boldsymbol{n}_\omega \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_\omega)$. The bias is modeled as a random walk $\dot{\boldsymbol{b}}_{\omega_k} = \boldsymbol{n}_{b_\omega}$, with $\boldsymbol{n}_{b_\omega} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{b_\omega})$.

B. Estimation Problem Formulation

We implement our hybrid drone dynamics in a sliding-window optimization-based VIO system based on the non-linear optimization proposed in [13]. An overview of the proposed optimization-based VIO with hybrid drone dynamics using a factor graph representation is in Fig. 2. The sliding window contains the most recent L keyframes and K drone states. In this work, we use $L = 10$ and $K = 5$. The optimization variables are: $\mathcal{X} = \{\mathcal{L}, \mathcal{X}_\mathcal{L}, \mathcal{X}_\mathcal{B}\}$, where $\mathcal{L}$ comprises the position of the 3D landmarks seen in the sliding window, $\mathcal{X}_\mathcal{L}$ the poses of the keyframes: $\mathcal{X}_\mathcal{L} = [\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_l], l \in [1, L]$, and

$\mathcal{X}_\mathcal{B}$ the poses of the drone: $\mathcal{X}_\mathcal{B} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_k], k \in [1, K]$. The l^{th} keyframe pose is $\boldsymbol{\zeta}_l = [\mathbf{p}^\mathcal{W}_{\mathcal{B}_l}, \mathbf{q}^\mathcal{W}_{\mathcal{B}_l}]$. The k^{th} drone state is $\boldsymbol{x}_k = [\boldsymbol{p}^\mathcal{W}_{\mathcal{B}_k}, \boldsymbol{q}^\mathcal{W}_{\mathcal{B}_k}, \boldsymbol{v}^\mathcal{W}_{\mathcal{B}_k}, \boldsymbol{b}_{a_k}, \boldsymbol{b}_{g_k}, \boldsymbol{f}^\mathcal{B}_{e_k}]$.

The visual-inertial-model estimation problem is formulated as a joint nonlinear optimization that solves for the maximum a posteriori estimate of $\mathcal{X}$. The cost function to minimize is:

$$\mathcal{L}^{HDVIO} = \sum_{h=0}^{L+K-1} \sum_{j \in \mathcal{J}_h} \left\| \boldsymbol{e}_{\boldsymbol{v}}^{j,h} \right\|^2_{\boldsymbol{W}_{\boldsymbol{v}}^{j,h}} + \sum_{k=0}^{K-1} \left\| \boldsymbol{e}_i^k \right\|^2_{\boldsymbol{W}_i^k} + \sum_{k=0}^{K-1} \left\| \boldsymbol{e}_d^k \right\|^2_{\boldsymbol{W}_d^k} + \left\| \boldsymbol{e}_m \right\|^2. \tag{3}$$

The cost function in Eq. 3 contains the weighted visual $\boldsymbol{e_v}$, inertial $\boldsymbol{e}_i$, dynamics $\boldsymbol{e}_d$, and marginalization residuals $\mathbf{e}_m$. The visual residuals are formulated as $\boldsymbol{e}_{\boldsymbol{v}}^{j,h} = \mathbf{z}^{j,h} - h(\mathbf{l}_j^\mathcal{W})$, which describes the re-projection error of the landmark $\mathbf{l}_j^\mathcal{W} \in \mathcal{J}_h$, where $\mathcal{J}_h$ is the set containing all the visible landmarks from the frame h. The function $h(\cdot)$ denotes the camera projection model and $\mathbf{z}^{j,h}$ the 2D image measurement. We refer to [13] for further details. The inertial residuals $\boldsymbol{e}_i$ are formulated using the IMU preintegration algorithm [14]. The dynamics residuals are presented in Sec. III-C. The error term $\boldsymbol{e}_m$ denotes the prior information obtained from marginalization. We adopt the marginalization strategy proposed in [13].

We based our implementation of the sliding-window optimization on [13]. We merge this VIO backend with the visual frontend proposed in [12]. The code of this VIO pipeline is available open-source[2]. We implement the drone model of VIMO and our hybrid-dynamics model in this VIO system.

C. Dynamics Residual

To derive the dynamics motion constraint, we use the collective force measurement model: $\hat{\boldsymbol{f}}^\mathcal{B}_k = \boldsymbol{f}^\mathcal{B}_{t_k} + \boldsymbol{f}^\mathcal{B}_{res_k} + \boldsymbol{n}_{f_t}$. Hence, in addition to the residual force $\boldsymbol{f}^\mathcal{B}_{\text{res}_k}$, we also consider a zero-mean gaussian noise $\boldsymbol{n}_{f_t} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{f_t})$ to account for uncertainty in the force direction. Given two consecutive states at t_k and t_{k+1}, the dynamic motion constraint is:

$$\boldsymbol{e}_d^k = \begin{bmatrix} \boldsymbol{\alpha}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}} - \hat{\boldsymbol{\alpha}}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}} \\ \boldsymbol{\beta}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}} - \hat{\boldsymbol{\beta}}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}} \\ \boldsymbol{f}^\mathcal{B}_{e_k} \end{bmatrix}, \boldsymbol{W}_d^k = \begin{bmatrix} \boldsymbol{P}^{\mathcal{B}_k - 1}_{\mathcal{B}_{k+1}[0:5]} & \mathbf{0} \\ \mathbf{0} & w_f \boldsymbol{I} \end{bmatrix}. \tag{4}$$

The quantities $\boldsymbol{\alpha}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}}$ and $\boldsymbol{\beta}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}}$ are the position and velocity change in the time interval $[t_k, t_{k+1}]$ and are written as:

$$\begin{aligned} \boldsymbol{\alpha}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}} &= \boldsymbol{R}^{\mathcal{B}_k}_\mathcal{W} (\boldsymbol{p}^\mathcal{W}_{\mathcal{B}_{k+1}} - \boldsymbol{p}^\mathcal{W}_{\mathcal{B}_k} - \boldsymbol{v}^\mathcal{W}_{\mathcal{B}_k} \Delta t_k - \frac{1}{2} \boldsymbol{g}^\mathcal{W} \Delta t_k^2) \\ &\quad - \frac{1}{2} \boldsymbol{f}^\mathcal{B}_{e_k} \Delta t_k^2 \\ \boldsymbol{\beta}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}} &= \boldsymbol{R}^{\mathcal{B}_k}_\mathcal{W} (\boldsymbol{v}^\mathcal{W}_{\mathcal{B}_{k+1}} - \boldsymbol{v}^\mathcal{W}_{\mathcal{B}_k} - \boldsymbol{g}^\mathcal{W} \Delta t_k) - \boldsymbol{f}^\mathcal{B}_{e_k} \Delta t_k. \end{aligned} \tag{5}$$

The quantities $\hat{\boldsymbol{\alpha}}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}}$ and $\hat{\boldsymbol{\beta}}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}}$ are the preintegrated position and velocity in $[t_k, t_{k+1}]$. They can be calculated in the discrete-time using Euler numerical integration over the timestep δt:

$$\begin{aligned} \hat{\boldsymbol{\alpha}}^{\mathcal{B}_k}_{i+1} &= \hat{\boldsymbol{\alpha}}^{\mathcal{B}_k}_i + \hat{\boldsymbol{\beta}}^{\mathcal{B}_k}_i \delta t + \frac{1}{2} \boldsymbol{R}(\hat{\gamma}^{\mathcal{B}_k}_i) \hat{\boldsymbol{f}}^\mathcal{B}_i \delta t^2 \\ \hat{\boldsymbol{\beta}}^{\mathcal{B}_k}_{i+1} &= \hat{\boldsymbol{\beta}}^{\mathcal{B}_k}_i + \boldsymbol{R}(\hat{\gamma}^{\mathcal{B}_k}_i) \hat{\boldsymbol{f}}^\mathcal{B}_i \delta t \\ \hat{\gamma}^{\mathcal{B}_k}_{i+1} &= \hat{\gamma}^{\mathcal{B}_k}_i \otimes \begin{bmatrix} 1 \\ \frac{1}{2} (\hat{\boldsymbol{\omega}}^\mathcal{B}_i - \boldsymbol{b}_{\omega_k}) \delta t \end{bmatrix}, \end{aligned} \tag{6}$$

where the initial conditions are: $\hat{\boldsymbol{\alpha}}^{\mathcal{B}_k}_{\mathcal{B}_k} = \hat{\boldsymbol{\beta}}^{\mathcal{B}_k}_{\mathcal{B}_k} = 0$ and $\hat{\gamma}^{\mathcal{B}_k}_{\mathcal{B}_k}$ is equal to the identity quaternion. $\boldsymbol{R}(\hat{\gamma}^{\mathcal{B}_k}_i)$ is the rotation matrix representation of $\hat{\gamma}^{\mathcal{B}_k}_i$. We run the propagation algorithm at the rate of the IMU, which is the faster sensor in our experiments. The quantity $\boldsymbol{W}_d^k$ is the weight of the residual term. It can be calculated from the 6×6 top-left block of the covariance $\boldsymbol{P}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}}$. This covariance is derived by linearizing the error, $\delta \boldsymbol{z} = [\delta \boldsymbol{\alpha}, \delta \boldsymbol{\beta}, \delta \boldsymbol{\theta}, \delta \boldsymbol{b}_\omega]^\top$, and noise, $\boldsymbol{n} = [\boldsymbol{n}_{f_t}, \boldsymbol{n}_\omega, \boldsymbol{n}_{b_\omega}]^\top$ in δt. Where $\delta \boldsymbol{\theta} = \boldsymbol{\gamma}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}} - \hat{\boldsymbol{\gamma}}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}}$, with $\boldsymbol{\gamma}^{\mathcal{B}_k}_{\mathcal{B}_{k+1}} = \boldsymbol{q}^{\mathcal{B}_k}_\mathcal{W} \otimes \boldsymbol{q}^\mathcal{W}_{\mathcal{B}_{k+1}}$, and $\delta \boldsymbol{b}_\omega = \boldsymbol{b}_{\omega_{k+1}} - \boldsymbol{b}_{\omega_k}$. It is important to note that the preintegrated terms depend on the gyroscope bias. In order to avoid repropagating each time that the estimate of the gyroscope bias changes, we employ the strategy proposed in [14]. Namely, the preintegration terms are corrected by their first-order approximation with respect to the change in the gyroscope bias. We refer to [14] for a detailed derivation of the preintegration theory.

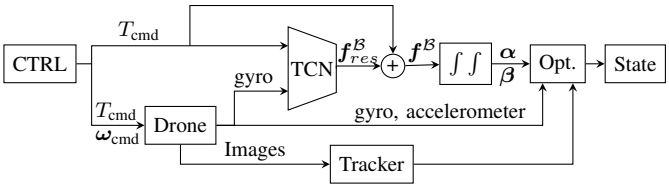


Fig. 3: Our novel learning-based model learns to predict the forces acting on the vehicle only based on the thrusts commanded by the controller and the gyro measurements from the IMU onboard the drone. The commanded collective thrust is added to the predicted residual force $\boldsymbol{f}^\mathcal{B}_{res}$ and then integrated to obtain the velocity and position updates. The optimizer then estimates the vehicle state in a tightly-coupled fashion, taking into account the displacements predicted by the hybrid dynamics model, the displacements predicted by the IMU, and the visual features.

D. Learning Residual Dynamics

The dynamics residual term presented above relies on an accurate estimate of the forces acting on the vehicle. In previous works, modeling aerodynamic effects, such as drag forces, requires knowledge of the linear velocity of the vehicle, which is not measured, but is part of the state to be estimated. Therefore, simply reusing a state-of-the-art quadcopter dynamics model is not possible.

In our setting, we have access to thrusts and gyroscope measurements as these values are directly measured. The goal is to estimate a residual force $\boldsymbol{f}^\mathcal{B}_{res}$ that accounts for aerodynamic effects and model mismatches (systematic noise) between the commanded or measured thrust T and the actual force acting on the robot (when no disturbance is present). For this, we propose the system architecture shown in Fig. 3. At the core of the learning-based component, we use a temporal

[2]https://github.com/uzh-rpg/rpg_svo_pro_open

convolutional network (TCN). TCNs have been shown to be as powerful as recurrent networks to model temporal sequences [32] but require less computation. The network takes a buffer of collective thrust and gyroscope measurements as input. The bias is removed from the gyroscope measurements. During training, we assume that the bias behaves as a random Gaussian variable with zero mean and standard deviation equal to 1e-3. At deployment time, we use the current bias estimate. Given as input a buffer of measurements in the time interval $\Delta t_{i,j} = t_j - t_i$, the neural net output is the residual force $\boldsymbol{f}^{\mathcal{B}}_{res_i}$. This residual is added to the thrust inputs $\boldsymbol{f}^{\mathcal{B}}_{t_k}$ with $k \in [t_i, t_j]$ to yield forces $\hat{\boldsymbol{f}}^{\mathcal{B}}_k$ that take aerodynamics and robot miscalibration into account. These forces are then used inside the preintegration framework, see Sec. III-C, to obtain relative velocity and position measurements.

We train the neural network to minimize the MSE loss:

$$\mathcal{L}^{HD}(\Delta\boldsymbol{\alpha},\Delta\hat{\boldsymbol{\alpha}},\Delta\boldsymbol{\beta},\Delta\hat{\boldsymbol{\beta}}) = \frac{1}{N}\sum_{n=1}^{N}\left(\left\|\boldsymbol{\alpha}^{\mathcal{B}_j}_{\mathcal{B}_i} - \hat{\boldsymbol{\alpha}}^{\mathcal{B}_j}_{\mathcal{B}_i}\right\|^2 + \left\|\boldsymbol{\beta}^{\mathcal{B}_j}_{\mathcal{B}_i} - \hat{\boldsymbol{\beta}}^{\mathcal{B}_j}_{\mathcal{B}_i}\right\|^2\right) \quad (7)$$

where $\boldsymbol{\alpha}^{\mathcal{B}_j}_{\mathcal{B}_i}$ and $\boldsymbol{\beta}^{\mathcal{B}_j}_{\mathcal{B}_i}$ are the ground-truth velocity and position changes, and N is the batch size. In order to learn aerodynamic effects and systematic noise in the input thrust measurements, there is no external force acting on the drone in the training data. Also, our training does not necessarily require force ground-truth data, which removes the need for a high-resolution motion-capture system. Instead, our training data could be collected using a SLAM pipeline.

Our TCN architecture consists of four temporal convolutional layers with 64 filters, followed by three temporal convolutional layers with 128 filters each. A final linear layer maps the signal to a 3-dimensional vector representing the learned residual thrust. The thrust and IMU measurements are sampled at 100 Hz and are fed to the TCN in an input buffer of a length of 100 ms. Consequently, each input buffer contains 10 thrust and 10 gyroscope measurements. We use the Gaussian Error Linear Unit (GELU) activation function. We train our neural network on a laptop running Ubuntu 20.04 and equipped with an Intel Core i9 2.3GHz CPU and Nvidia RTX 4000 GPU. Training is performed using the Adam optimizer with an initial learning rate of 1e-4. To test the feasibility of deploying the neural network onboard the quadrotor, we test the neural net inference on an NVIDIA Jetson TX2, which is the computing platform onboard the quadrotor. The neural net inference runs at ≈180 Hz on an NVIDIA Jetson TX2 which exceeds the required 100 Hz state-update rate of our controllers for agile flight.

IV. Results on Datasets

In our experiments, we evaluate our method against the same VIO system without the proposed hybrid-dynamics model (from now on referred to as VIO) and also against VIMO. We refer the reader to the Appendix for an evaluation against VID-Fusion [2]. Following the best practices in the evaluation of VIO algorithms [33], we use the evaluation metrics: translation absolute trajectory error (ATE_T [m]), rotation absolute trajectory error (ATE_R [deg]), and relative translation and rotation errors. These error metrics are computed after aligning the estimated trajectory with the pose-yaw method [33]. We refer the reader to [33] for a detailed description of these metrics. In addition, we use the RMSE between the ground truth and predicted force to evaluate the accuracy of force estimation.

TABLE I: Comparison in terms of RMSE of the force estimates on the test set of the NeurBEM dataset. Methods of Type DD are data-driven, as opposed to FP first-principles methods. Our method performs remarkably well given it has no information about the velocity or orientation of the vehicle and only falls short of the NeuroBEM method which has access to the full vehicle state. The values for the first four methods are taken from [4].

Model	Type	Inputs	F_{xy} [N]	F_z [N]	F [N]
Quadratic Fit	FP	thrust	1.536	1.381	1.486
BEM [4]	FP	full state	0.803	1.265	0.982
PolyFit [5]	DD	full state	0.453	0.832	0.606
NeuroBEM [4]	DD	full state	**0.204**	**0.504**	**0.335**
HDVIO (Ours)	DD	thrust + gyro	0.402	0.672	0.491

A. NeuroBEM Dataset

Experimental Setup: In the first set of experiments, we want to evaluate the hybrid dynamics model without the full VIO pipeline. That is, the estimate of the total force $\boldsymbol{f}^{\mathcal{B}}$ (see Fig. 3) acting on the quadcopter from a history of thrusts and gyroscope measurements and compare it to ground-truth data. To perform this evaluation, we use the challenging NeuroBEM [4] dataset. It features data from indoor drone flights at speeds up to 65 km/h. This dataset provides rotor speeds, from which we compute thrust measurements, and gyroscope measurements alongside ground-truth forces.

We compare the force estimation accuracy of our learned dynamics model to the state-of-the-art baselines. The Quadratic Fit is the model used in VIMO. BEM is a state-of-the-art first-principles model which models the force and torque acting on a propeller by integrating over infinitesimal area elements of the propeller [10, 4]. The PolyFit model [5] is a data-driven model which relies on polynomial basis functions to model the drone dynamics. Finally, NeuroBEM is a hybrid model which augments the BEM model with a learning based component. Note that all baselines except the Quadratic Fit model require the full vehicle state as an input, including the linear and angular velocities.

Evaluation: The results of the comparison are summarized in Tab. I. As expected, the NeuroBEM method that has access to the full robot state outperforms ours. However our HDVIO outperforms BEM by a factor of three and PolyFit by a factor of two. Figure 4 illustrates this: it shows the ground-truth forces and the forces estimated by BEM and our HDVIO during the first five seconds of a very fast flight where the vehicle accelerates to 15 m/s on a lemniscate track. The accuracy of our HDVIO is remarkable because, in contrast to the baselines, it has no access to the ground-truth state information like the linear or angular velocity of the vehicle.

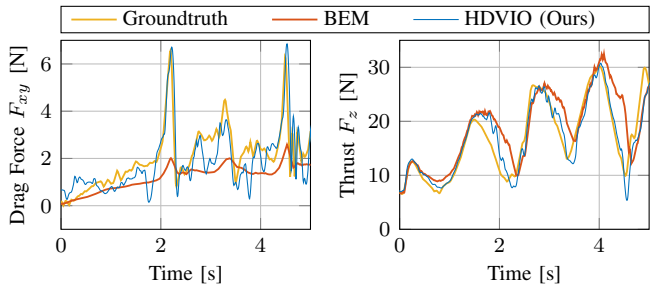


Fig. 4: This figure illustrates the results shown in Tab. I by exemplarily showing the force estimates on a very fast trajectory from the NeuroBEM dataset. Our HDVIO clearly outperforms the state-of-the-art first-principle model BEM and is able to model aerodynamic effects accurately on short timescales.

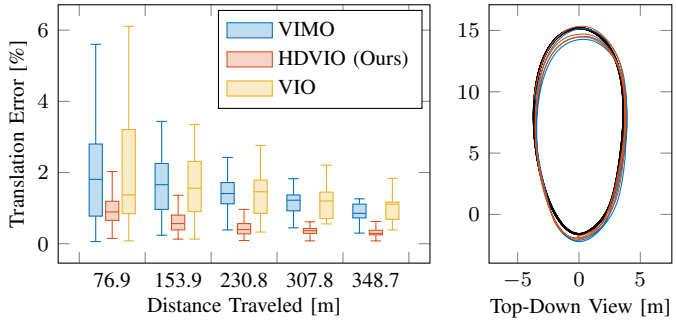


Fig. 5: The performance of VIO, VIMO and HDVIO is compared on the *Egg* 8 m/s trajectory (ground truth in black) from the Blackbird dataset. Our proposed method outperforms VIMO and the VIO clearly.

From this experiment, we conclude that our learning-based component is able to accurately capture the aerodynamic forces acting on the quadrotor and, consequently, is suitable for integration into a VIO pipeline. The fact that it is possible to estimate the aerodynamic forces so accurately only through a history of thrusts and gyroscope measurements is an interesting finding by itself.

B. Blackbird Dataset

Experiment Setup: In this second set of experiments, we evaluate our system and the baselines on the Blackbird dataset [15]. The Blackbird dataset provides rotor speed measurements recorded onboard a quadrotor flying in a motion-capture system. We use these measurements to compute mass-normalized collective thrust measurements. In addition, this dataset also contains IMU measurements and, in some of the sequences, photorealistic images of synthetic scenes. In total, the Blackbird dataset contains 18 different trajectories with speeds from 0.5 m/s to 9.0 m/s. Since this dataset does not contain external disturbances, we only evaluate the accuracy of the pose estimates. Among the trajectories with available camera images, we select 6 for evaluation: *Bent Dice*, *Clover*, *Egg*, *Mouse*, *Star*, and *Winter*. The remaining 80 % of the trajectories, amounting to approximately 2 hrs of flight data, are used for training and 20 % for validation of our neural network. In addition, we also train our network on a reduced training dataset that contains speeds up to 2 m/s to evaluate the generalization performance to higher speeds than the ones present in the training data.

TABLE II: Evaluation of the trajectory estimates in the Blackbird dataset. In bold are the best values, and in underlined are the second-best values. HDVIO* (ours) is trained on a reduced training set, with speeds up to 2 m/s to evaluate generalization performance.

Trajectory Name	$v_{\max}$ [m/s]	Evaluation Metric: ATE_T [m] / ATE_R [deg]			
		VIO	VIMO	HDVIO (ours)	HDVIO* (ours)
Bent Dice	3	**0.20** / 1.78	0.31 / 1.53	0.21 / 1.53	0.23 / **1.46**
Clover	5	0.90 / 3.52	0.88 / 3.66	0.60 / **2.08**	**0.59** / 2.95
Egg	5	1.07 / 1.54	0.75 / 1.34	**0.59** / **1.21**	0.68 / 1.28
Egg	6	1.40 / 2.35	0.98 / 4.89	0.83 / **1.62**	**0.81** / 2.26
Egg	8	1.79 / 4.55	1.57 / 3.69	**1.06** / **2.89**	1.22 / 3.52
Mouse	5	1.10 / 4.54	0.76 / 2.14	**0.36** / **1.40**	0.40 / 1.75
Star	1	0.17 / 0.78	0.18 / 1.05	**0.16** / **0.58**	**0.16** / 0.62
Star	3	0.62 / 3.50	0.43 / **1.38**	0.38 / 1.40	**0.34** / 1.42
Winter	4	0.97 / 2.92	0.69 / 2.46	0.57 / **1.54**	**0.50** / 2.27

Evaluation: We present the ATE_T and ATE_R on the evaluation sequences in Table II. Our approach outperforms the VIO and VIMO baselines showing the effectiveness of our hybrid drone model. As expected, the performance increase becomes larger at higher speeds, reaching an improvement of 41 % and 33 % against the VIO and VIMO respectively, at the max peak velocity of 8 m/s. This result is explained by the fact that our method includes the learned drag forces as measurements in the dynamics motion constraint. We include the relative translation error and the top-down view of the estimated trajectory in Fig. 5. Remarkably, our system still outperforms the baselines in almost all the sequences when the network is trained on the reduced training dataset, see the last column of Tab. II. This result shows that HDVIO is able to generalize to velocities up to 4x larger than the ones included in the training data.

C. VID Dataset

Experiment Setup: In this third set of experiments, we are interested in evaluating the ability of our system in estimating an external force acting on the quadrotor and in testing our learned component when ground-truth data from a motion capture system is not available. The VID dataset [2] contains visual, inertial, actuation inputs, and ground-truth force measurements. In this dataset, the data is recorded onboard a quadrotor flying in an office room equipped with a motion-capture system and in an outdoor parking area. We use the provided rotor speed measurements to compute the thrust measurements. We use the indoor sequences to evaluate the estimation of the external force. Parts of these sequences include ground-truth force data. We use the outdoor sequences to validate our learned module when ground-truth position and velocity training data is obtained from a visual-inertial SLAM system [34] instead of a motion-capture system. Since outdoor sequences do not include ground-truth force measurements, we are interested in the estimation of the drone poses. The quadrotor mass changes between the indoor and outdoor sequences, for this reason, we train two different neural networks, one for the indoor drone configuration and one for the outdoor drone configuration. We use the indoor sequences without any external perturbance to train our neural

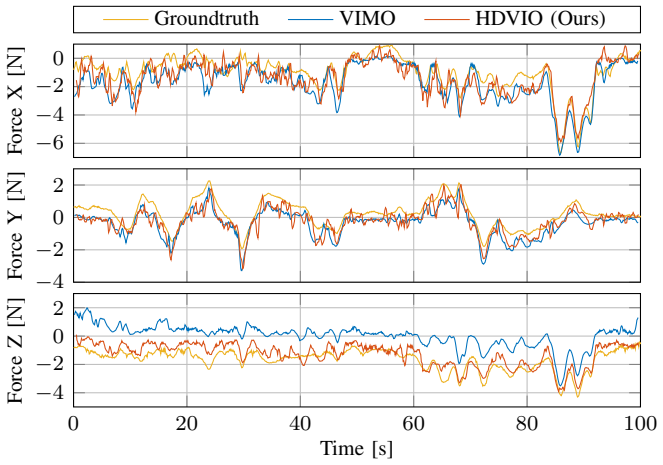
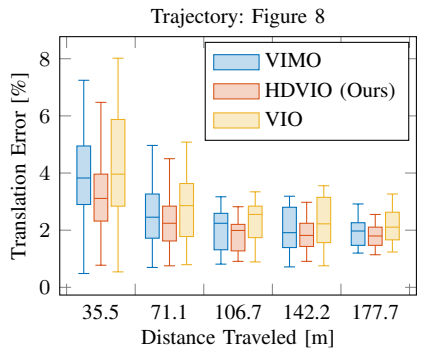


Fig. 6: Comparison of the external force estimate in the *sequence 17* of the VID dataset. HDVIO drastically improves the force estimation along the z axes resulting in a 40% reduction of the RMSE.



	Method	Figure 8	Rectangle
ATE pos [m]	VIO	1.89	2.09
	VIMO	1.67	1.89
	HDVIO	**1.48**	**1.72**
ATE rot [deg]	VIO	4.02	2.01
	VIMO	**3.70**	1.92
	HDVIO	**3.70**	**1.86**

Fig. 7: The plot and accompanying table show how our HDVIO performs in a setting where the training data for the learning-based component is gathered from a vision-based SLAM system in the outdoor sequences of the VID dataset. The flown trajectories are at low speeds below 3 m/s, which is why all three methods show good performance, with HDVIO being the more accurate.

network for the indoor configuration. They consist of a hover, a circle, and, a figure 8 trajectory amounting to only 6 min of flying data. 80% of this data is used for training and 20% for validation. The outdoor sequences contain circle, figure 8, and rectangle trajectories. We use one figure 8 and one rectangle trajectory for testing. The remaining data, amounting to only 11 min of flying data, is used for training, 80%, and validation, 20%.

Evaluation: We include in Fig. 6 the external force estimates of VIMO and our method in the *sequence 17* of the dataset. In this sequence, the quadrotor flies at low speed, below 1 m/s, and is attached to a rope. The estimates are aligned to the motion-capture reference frame. To perform the alignment, we use the *posyaw* alignment method [33]. The RMSE achieved by VIMO is 1.08 N, and the RMSE the along x, y, and z axis are 0.89, 0.63, and, 1.73 N, respectively. The RMSE achieved by HDVIO is 0.65 N resulting in a 40% reduction. The RMSE along the x, y, and z axis are 0.81, 0.61, and, 0.55 N, respectively. Clearly from the plot in Fig. 6, we see that our method drastically improves the force estimation along the z axes. Our neural network has learned to compensate for a systematic residual error affecting the thrust inputs. We believe that the cause of this error is inaccuracy in the thrust coefficients used to compute the collective thrust inputs from the rotor speed measurements. In this sequence, the slow motion of the vehicle and the rich texture environment renders the pose estimation problem simple. The VIO algorithm, VIMO, and HDVIO achieve similar performance. Namely, the ATE_T of the three methods is 0.02 m. The plot and accompanying table in Fig. 7 shows the evaluation of the pose estimates in the outdoor sequences. The flown trajectories are at low speeds below 3 m/s, which is why all three methods show good performance, with HDVIO being the more accurate. This experiment shows that our learning-based dynamics model can be purely trained without using an external motion-capture system.

V. Real-World Experiments

In this set of experiments, we demonstrate that HDVIO is able to estimate continuous external disturbances, such as continuous wind, outperforming the state-of-the-art method VIMO. To achieve so, we fly a quadrotor in a wind field as shown in Fig. 1. Details on the quadrotor platform are given in [35]. We obtain camera and IMU measurements from an onboard Intel RealSense T265[3] camera. Although this camera provides stereo fisheye images, we only use images from the left camera. Rotor speed measurements are not available on our quadrotor platform. Instead, we use the collective thrust commands that are output by the MPC controller [35] used to control the vehicle. The quadrotor is flown in a motion-capture system that provides pose at 200 Hz update rate.

We conduct the experiments with two different quadrotor configurations: one where the vehicle was in its nominal state and one with a 22 cm×16 cm large dragboard attached to the vehicle. The dragboard is attached such that its normal corresponds to the body y-axis of the drone. The dragboard increases the drag in the y-direction over a factor of 2 and additionally makes the vehicle much more sensitive to crosswinds.

A. Wind Generation

To generate a windfield, we place three axial fans (Ekström 12 inch, see Fig. 1) in an office-like room of 8x10 m at a height of 1.6 m above the ground, in a way that the fans are slightly angled inwards. This ensures high windspeeds across a virtual tube in front of the fans. Each fan has an (advertised) air circulation of $1.3\,\text{m}^3/\text{s}$ and a measured wind speed of up to 8 m/s at the front grill.

To quantitatively evaluate the performance of our method, ground-truth data for the external wind forces is required. Following, we describe how we obtain the ground-truth wind forces.

1) Wind Speed Map: In the first step, the wind produced by our experimental setup is measured. The local wind speed is recorded at 50 points in the wind cone in front of the fans, where samples are denser in regions where the wind speed changes quickly. The data is recorded using a hand-held

[3]https://www.intelrealsense.com/wp-content/uploads/2019/09/Intel_RealSense_Tracking_Camera_Datasheet_Rev004_release.pdf

TABLE III: Trajectories estimates in our real-world experiments. We use (d) to indicate that a drag board was attached to the drone.

	ATE Position [m]			ATE Rotation [deg]		
	VIO	VIMO	HDVIO	VIO	VIMO	HDVIO
Circle (d)	**0.07**	0.10	**0.07**	2.02	**1.80**	2.06
Circle	**0.06**	0.08	**0.06**	1.21	1.19	**1.17**
Lemniscate (d)	0.38	0.34	**0.30**	**2.39**	2.93	2.81
Lemniscate	0.27	0.32	**0.20**	2.44	1.93	**1.84**

anemometer (Basetech BS-10AN) whose position is tracked using the motion-capture system. To obtain the ground-truth wind speed map shown in Fig. 1, a smoothing spline is fitted to the data.

2) Lift and Drag Coefficients: Since HDVIO estimates the disturbance force acting on the vehicle, we calculate the wind force based on the wind speed map. The aerodynamic forces acting on a quadrotor are primarily determined by the body/fuselage drag, f_d^{fus}, the induced drag from the propellers f_d^{ind}, and the lift and drag incurred by the flat-plate drag board attached to the top of the quadrotor, f_l^{brd} and f_d^{brd}. The magnitudes of the forces can be approximated as [4, 6, 36]:

$$\begin{aligned} f_d^{\text{fus}} &= 0.5\,\rho\,A^{\text{fus}}\,c_d^{\text{fus}}\,v_{\text{rel}}^2 \\ f_d^{\text{ind}} &= k\,v_{\text{rel}} \\ f_{l|d}^{\text{brd}} &= 0.5\,\rho\,A^{\text{brd}}\,c_{l|d}^{\text{brd}}(\alpha)\,v_{\text{rel}}^2\ , \end{aligned} \tag{8}$$

where ρ is the air density, A is a surface area, v_{rel} the relative air speed, α is the angle of attack of the dragboard, k is the propeller drag coefficient, and $c_{l|d}$ are the lift and drag coefficients of the fuselage and dragboard. The relative air speed is given as the norm of the relative velocity, i.e., the sum of the ego motion and the wind.

In this model, the fuselage is represented by a square prism with an angle-of-attack independent drag coefficient of $c_d^{\text{fus}} = 2.0$ [37]. For the flat-plate wing, we use a simple model for high angles-of-attack that has found widespread application in propeller modeling [10, 36] and fits the experimental data for flat-plate wings [38]:

$$c_l^{\text{brd}}(\alpha) = \sin(2\alpha)\ , \qquad c_d^{\text{brd}}(\alpha) = 2\sin^2(\alpha).$$

To validate the predicted forces for the fuselage and drag board in Eq. (8), the quadrotor has been mounted on a load cell[4]. At 7 m/s wind speed, the lift and drag measurements are within 10 % of the calculated values. Furthermore, the linear propeller drag coefficient is found to be $k = 0.145$ Ns/m.

3) Wind Forces: Our method separates aerodynamic effects, i.e., body drag and induced drag, from external forces. Therefore, to obtain the ground truth for the disturbance caused by the wind, we calculate the forces acting on the quadrotor when the fans are active and subtract the force calculated when the fans are turned off.

B. Dataset Collection

In this set of experiments, we fly the quadrotor in a wind field with wind gusts up to 25 km/h. The training data consists of approximately 10 min of random trajectories flown without wind. We use 80% of this data to train the neural network and the remaining 20% for validation. We exclusively use *random* trajectories which are generated by sampling position data using a Gaussian Process. This approach ensures diverse training data and prevents overfitting to specific trajectories. The test trajectories consist of a circle and a lemniscate trajectory with a max speed of 2 m/s. We also recorded a second data set which features the same training, validation, and test trajectories, but the quadrotor is equipped with a drag board. In this case, we want to increase the magnitude of the drag and the external force due to the wind gusts to highlight the advantage of HDVIO compared to VIMO.

C. Evaluation

We present the estimate of the external force due to the wind gusts in Fig. 8. Since the wind gusts hit the quadrotor along the y axes of the world reference frame, we show the y component of the estimated force as well as the force norm. The external forces are estimated in the body frame of the quadrotor, cf. Eq. 1. We align them to the world frame, which corresponds to the motion-capture reference frame, using the ground-truth orientations. Using the ground-truth orientations allows us to directly compare the estimates of our method against the ones of VIMO. From Fig. 8, we see that our method achieves more accurate force estimates than VIMO. In particular, HDVIO outperforms VIMO in the prediction of the wind gusts when the quadrotor enters the wind field. This is visible from the fact that our method accurately predicts the peaks of the wind force.

As stated by the authors [1], the measurement model in VIMO, in case of continuous external disturbances, introduces an inconsistency in the estimates of the accelerometer bias resulting in decreased accuracy of the motion estimate. We show in Fig. 9 the accelerometer bias estimated by the VIO algorithm, by VIMO, and by our method. This is done for a sequence where the quadrotor, equipped with the drag board, flies a circle trajectory. We do not have access to the ground-truth accelerometer bias. However, we consider the one estimated by the VIO algorithm as a good approximation of the ground-truth value since the VIO achieves very high performance in this sequence, cf. Table III, thanks to the high number of visual features being tracked. The bias estimate of HDVIO closely follows the one of the VIO system, while the estimate of VIMO converges to a wrong value along the x-axis. We include the position and orientation absolute trajectory errors in Table III. In all 4 sequences, the rich texture environment simplifies the pose estimation problem. For this reason, the three algorithms achieve similar accuracy.

VI. Discussion

The proposed hybrid dynamics model, combining a point-mass quadrotor model with a learning-based residual force term, overcomes the limitations of the state-of-the-art visual-inertial-model-based odometry system, VIMO, in the case of large model mismatch (high speeds, systematic noise) and

[4]https://www.ati-ia.com/products/ft/ft_models.aspx?id=Mini40

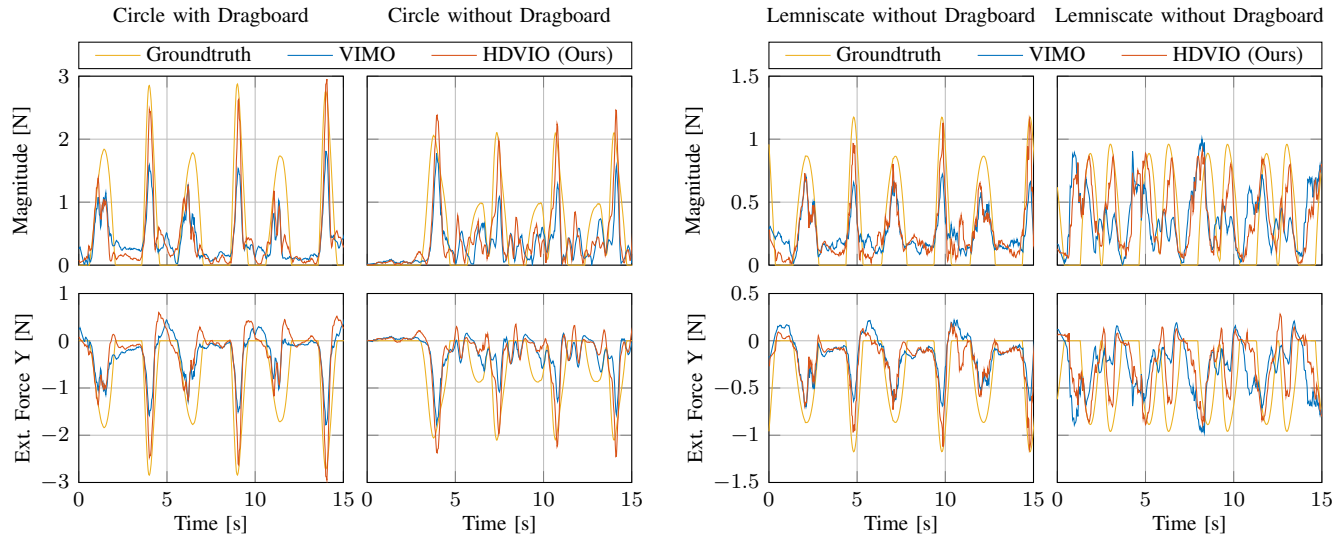


Fig. 8: Wind disturbance estimates in our real-world experiments. The magnitude and the y-axis component of the wind force estimated by HDVIO and VIMO. Left: circle trajectory. Right: Lemniscate trajectory. In all the plots, it is visible that HDVIO achieves more accurate force estimates than VIMO.

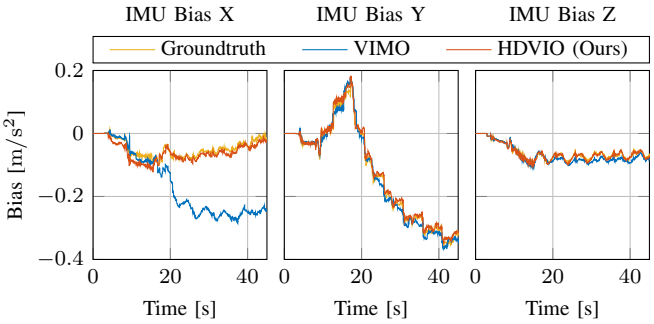


Fig. 9: Accelerometer bias estimates from our real-world experiments. The ground-truth bias is obtained from the VIO system. The estimates of our HDVIO match the ground-truth values, while, VIMO estimates diverge along the x-axis.

continuous external disturbances (continuous wind). Indeed, our HDVIO increases the accuracy of both motion and external force estimation.

Our learning-based module outperforms first-principle state-of-the-art quadrotor models, which have access to the full state of the quadrotor, in predicting the aerodynamic drag force. In contrast to these methods, our HDVIO only relies on thrust and gyroscope measurements, which are commonly available on any commercial drone platform.

Another advantage of our approach is that our network does not rely on ground-truth thrust forces to supervise training. In fact, our training strategy consists of minimizing the difference between the relative position and velocity changes predicted by propagating our hybrid drone dynamics with respect to the supervision quantities. Obtaining the supervision signal does not require access to expensive motion-capture systems but can be obtained from SLAM solutions [34, 39]. These SLAM solutions rely only on camera and IMU data, simplifying the training data collection process. In Fig. 7, we show such an experiment, where we trained the learning-based component purely with position and velocity signal obtained via SLAM. Furthermore, even a human pilot can be used to control the drone, as expert pilots typically control a drone by sending a collective thrust command along with the desired bodyrates [40]. This simple training data collection alleviates a limitation of our hybrid drone model: while a single dynamics model works only for a specific drone, it is easy to record data for training a new model.

Our hybrid drone model exhibits strong generalization capabilities to velocities and trajectories unseen during training. To show generalization to unseen velocities, we train HDVIO* in Sec. IV-B, on a dataset containing speeds only up to 2 m/s (HDVIO is trained with speeds up to 9 m/s). When tested on the full range of speeds (up to 8 m/s), we observe that HDVIO* achieves a decrease in performance of only 4% in ATE_T and still outperforms the baselines (see Table II). We show the generalization performance w.r.t. the kind of trajectory in different examples. In the NeuroBEM dataset (see Table I), the test dataset contains 30% of trajectories completely unseen during training, and 70% is at least different in speed and size. Our system outperforms BEM and PolyFit by 50% and 20%, respectively, and is only inferior to NeuroBEM which has access to the full vehicle state. Furthermore, our method can predict the external force acting on the drone flying an unseen random trajectory 40% more accurately than VIMO (see Fig. 6). In Sec. V-B, we train our network exclusively on random trajectories and observe more accurate estimate of the wind force (see Fig. 8) and accelerometer bias (see Fig. 9).

Furthermore, HDVIO achieves high robustness w.r.t. VIO failures and continuous external disturbances. In Sec. IV-B, HDVIO achieves the largest improvements, equal to 41% and 33%, on the fastest trajectory, Egg 8 m/s (see Table II and Fig. 5). Due to motion blur and fast yaw changes tracking features is difficult here, resulting in the VIO system accumulating large drift. Moreover, neglecting drag effects in the drone model, as in [1, 2], is not a proper assumption at this speed. We evaluate the ability to estimate external forces in

the presence of continuous perturbations: pulling rope (see Sec. IV-C) and wind (see Sec. V). In all these challenging scenarios, HDVIO outperforms the baselines, by up to 40% in force prediction (see Fig. 6), highlighting its robustness.

In this work, similarly to VIMO, the model is assumed to be fixed during a flight. Changes in the mass (external payload) or in the actuation inputs (hardware degradation) are seen as an external force. An interesting venue for future work is to train the neural network to estimate these model changes as residual forces.

We decided to use in HDVIO as well as in VIMO and in the VIO system without the dynamics model, the visual frontend proposed in [12]. Our choice is based on the high robustness achieved by [12] thanks to its semi-direct approach to visual feature tracking and low computational requirements. These characteristics are very appealing for VIO applications onboard flying vehicles.

VII. Conclusions

This work proposes a novel method to model the quadrotor dynamics in visual-inertial odometry systems. Our dynamics model combines a first principles quadrotor model with a learning-based component that captures unmodeled effects, such as aerodynamic drag. The proposed method overcomes the limitations of the state-of-the-art visual-inertial-model-based odometry system, VIMO, by increasing the accuracy of motion and external force estimation up to 33% and 40%, respectively. Our learning-based component shows strong generalization capabilities beyond the type and speed of the trajectories seen in the training dataset. An evaluation of the accuracy of the residual force estimates shows that our learning-based component outperforms sophisticated first-principle models that have access to the full state of the quadrotor. Experiments in controlled wind conditions show that our hybrid dynamics model achieves accurate predictions of the force affecting the quadrotor due to continuous wind.

Our HDVIO method can increase the safety of autonomous flights in hazardous scenarios, such as fast flights and operations in windy conditions. In view of the increasing drone usage in our everyday life, such aspects are becoming more and more relevant and we believe that our work makes a valuable contribution towards this goal.

References

[1] Barza Nisar, Philipp Foehn, Davide Falanga, and Davide Scaramuzza. VIMO: Simultaneous visual inertial model-based odometry and force estimation. *Robotics: Science and Systems (RSS)*, 2019.

[2] Ziming Ding, Tiankai Yang, Kunyi Zhang, Chao Xu, and Fei Gao. Vid-fusion: Robust visual-inertial-dynamics odometry for accurate external force estimation. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021.

[3] Chuchu Chen, Yulin Yang, Patrick Geneva, Woosik Lee, and Guoquan Huang. Visual-inertial-aided online mav system identification. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2022.

[4] Leonard Bauersfeld, Elia Kaufmann, Philipp Foehn, Sihao Sun, and Davide Scaramuzza. NeuroBEM: Hybrid aerodynamic quadrotor model. *Robotics: Science and Systems (RSS)*, 2021.

[5] Sihao Sun, Coen C de Visser, and Qiping Chu. Quadrotor gray-box model identification from high-speed flight data. *Journal of Aircraft*, 2019.

[6] Leonard Bauersfeld and Davide Scaramuzza. Range, endurance, and optimal speed estimates for multicopters. *IEEE Robot. Autom. Lett.*, 2022.

[7] Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 2022.

[8] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.*, 2018.

[9] Fadri Furrer, Michael Burri, Markus Achtelik, and Roland Siegwart. Rotors—a modular gazebo mav simulator framework. In *Robot operating system (ROS)*. 2016.

[10] Rajan Gill and Raffaello D'Andrea. Propeller thrust and drag in forward flight. In *2017 IEEE Conf. on Control Tech. and Applications (CCTA)*, 2017.

[11] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. Accessed: 2023-19-05.

[12] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robot.*, 2017.

[13] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *Int. J. Robot. Research*, 2015.

[14] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Trans. Robot.*, 2016.

[15] Amado Antonini, Winter Guerra, Varun Murali, Thomas Sayre-McCord, and Sertac Karaman. The blackbird uav dataset. *Int. J. Robot. Research*, 2020.

[16] Kunyi Zhang, Tiankai Yang, Ziming Ding, Sheng Yang, Teng Ma, Mingyang Li, Chao Xu, and Fei Gao. The visual-inertial-dynamical multirotor dataset. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2022.

[17] Teodor Tomić and Sami Haddadin. A unified framework for external wrench estimation, interaction control and collision reflexes for flying robots. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2014.

[18] Burak Yüksel, Cristian Secchi, Heinrich H Bülthoff, and Antonio Franchi. A nonlinear force observer for quadrotors and application to physical interactive tasks. In *2014 IEEE/ASME Int. Conf. on Advanced Intel. Mechatronics*, 2014.

[19] Fabio Ruggiero, Jonathan Cacace, Hamid Sadeghian, and Vincenzo Lippiello. Impedance control of vtol uavs with a momentum-based external generalized forces estimator. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014.

[20] Christopher D McKinnon and Angela P Schoellig. Unscented external force and torque estimation for quadrotors. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2016.

[21] Federico Augugliaro and Raffaello D'Andrea. Admittance control for physical human-quadrocopter interaction. In *IEEE Eur. Control Conf. (ECC)*, 2013.

[22] Andrea Tagliabue, Mina Kamel, Sebastian Verling, Roland Siegwart, and Juan Nieto. Collaborative transportation using mavs via passive force control. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017.

[23] Dinuka Abeywardena, Zhan Wang, Gamini Dissanayake, Steven L Waslander, and Sarath Kodagoda. Model-aided state estimation for quadrotor micro air vehicles amidst wind disturbances. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2014.

[24] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2007.

[25] Andrea Tagliabue, Aleix Paris, Suhan Kim, Regan Kubicek, Sarah Bergbreiter, and Jonathan P How. Touch the wind: Simultaneous airflow, drag and interaction sensing on a multirotor. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2020.

[26] Mederic Fourmy, Thomas Flayols, Pierre-Alexandre Léziart, Nicolas Mansard, and Joan Solà. Contact forces preintegration for estimation in legged robotics using factor graphs. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021.

[27] Yunlong Song, Selim Naji, Elia Kaufmann, Antonio Loquercio, and Davide Scaramuzza. Flightmare: A flexible quadrotor simulator. 2020.

[28] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, 2018.

[29] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar Von Stryk. Comprehensive simulation of quadrotor uavs using ros and gazebo. In *International conference on simulation, modeling, and programming for autonomous robots*, 2012.

[30] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *AIAA guidance, navigation and control conference and exhibit*, 2007.

[31] Matko Orsag and Stjepan Bogdan. Influence of forward and descent flight on quadrotor dynamics. *Recent Advances in Aircraft Technology*, 2012.

[32] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. Accessed 2023-03-02.

[33] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[34] Giovanni Cioffi, Titus Cieslewski, and Davide Scaramuzza. Continuous-time vs. discrete-time vision-based slam: A comparative study. *IEEE Robot. Autom. Lett.*, 2022.

[35] Philipp Foehn, Elia Kaufmann, Angel Romero, Robert Penicka, Sihao Sun, Leonard Bauersfeld, Thomas Laengle, Giovanni Cioffi, Yunlong Song, Antonio Loquercio, and Davide Scaramuzza. Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight. *Science Robotics*, 2022.

[36] Guillaume Ducard and Minh-Duc Hua. Modeling of an unmanned hybrid aerial vehicle. In *2014 IEEE Conf. on Control Applications (CCA)*, 2014.

[37] Nils Paul van Hinsberg. Aerodynamics of smooth and rough square-section prisms at incidence in very high reynolds-number cross-flows. *Experiments in Fluids*, 2021.

[38] Robert E. Sheldahl and Paul C. Klimas. Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines. In *Sandia National Labs., Albuquerque, NM (USA)*, 1981.

[39] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[40] Christian Pfeiffer, Simon Wengeler, Antonio Loquercio, and Davide Scaramuzza. Visual attention prediction improves performance of autonomous drone racing agents. *Plos one*, 2022.

VIII. Appendix

A. Comparison against VID-Fusion

VID-Fusion is a visual-inertial-model-based odometry method very similar to VIMO. The main difference between the two methods is the external force model. While in VIMO, the external force is modeled as a Gaussian random variable with zero mean, in VID-Fusion, the mean is obtained by integrating the difference between accelerometer and thrust measurements. According to the authors, this prior helps the estimation of continuous external forces. In the main paper, we did not include VID-Fusion [2] among the baselines for conciseness. Since the drone model is the same as the one used in VIMO, VID-Fusion has the same limitations. Consequently, including VID-Fusion in the baselines does not affect the conclusions drawn in the paper.

In this appendix, for the sake of completeness, we present the same experiments as in the main paper including VID-Fusion among the baselines. We refer the reader to the main paper for the detailed description of the experimental setup and focus here only on the results.

1) Blackbird Dataset: We present the ATE_T and ATE_R on the evaluation sequences of the Blackbird dataset in Table IV. Since VIMO and VID-Fusion use the same drone dynamics model and there are no external perturbances acting on the drone in this dataset, their trajectory estimation accuracy is similar. Our method outperforms both baselines as well as the VIO solution. Large improvements are in the fast sequences, where our learned aerodynamics model brings additional information to the VIO backend. However, the performance difference is small in slow sequences, where including aerodynamic effects in the drone model is less effective.

TABLE IV: Evaluation of the trajectory estimates in the Blackbird dataset. In bold are the best values, and the second-best values are underlined.

Trajectory Name	$v_{\max}$ [m/s]	Evaluation Metric: ATE_T [m] / ATE_R [deg]			
		VIO	VID	VIMO	HDVIO (ours)
Bent Dice	3	**0.20** / 1.78	0.25 / **1.18**	0.31 / 1.53	0.21 / 1.53
Clover	5	0.90 / 3.52	0.83 / 2.48	0.88 / 3.66	**0.60** / **2.08**
Egg	5	1.07 / 1.54	0.81 / 1.61	0.75 / 1.34	**0.59** / **1.21**
Egg	6	1.40 / 2.35	1.10 / 2.42	0.98 / 4.89	**0.83** / **1.62**
Egg	8	1.79 / 4.55	1.47 / 4.84	1.57 / 3.69	**1.06** / **2.89**
Mouse	5	1.10 / 4.54	0.54 / 2.10	0.76 / 2.14	**0.36** / **1.40**
Star	1	0.17 / 0.78	0.18 / **0.54**	0.18 / 1.05	**0.16** / 0.58
Star	3	0.62 / 3.50	0.50 / 2.93	0.43 / **1.38**	**0.38** / 1.40
Winter	4	0.97 / 2.92	0.66 / 2.05	0.69 / 2.46	**0.57** / **1.54**

2) VID Dataset: We evaluate the estimates of the external force in the *sequence 17* of the VID dataset in Fig 10. In this sequence, the quadrotor is attached to a rope. Ground-truth forces are available from a force sensor attached to the other end of the rope. The force estimates are aligned to the motion-capture reference frame using the *posyaw* alignment method [33]. Our hybrid drone model has learned to compensate for a systematic residual error affecting the thrust inputs.

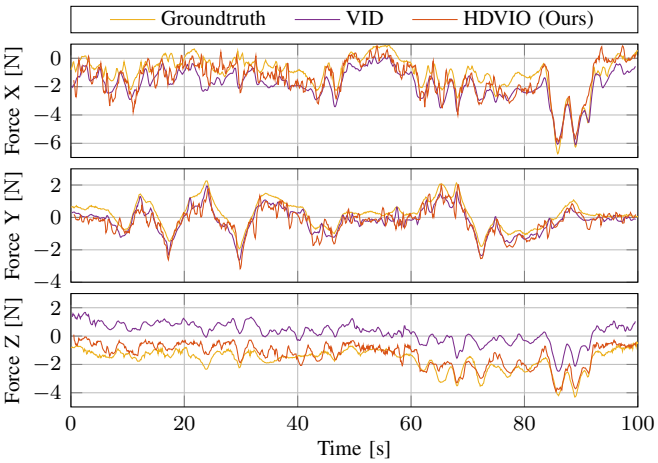
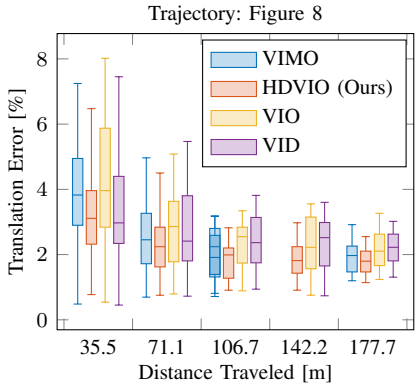


Fig. 10: Comparison of the external force estimate in *the sequence 17* of the VID dataset. Comparison of the external force estimate. HDVIO drastically improves the force estimation along the z-axis resulting in a 42% reduction of the RMSE compared to VID-Fusion.



	Method	Figure 8	Rectangle
ATE pos [m]	VIO	1.89	2.09
	VID	1.84	1.99
	VIMO	1.67	1.89
	HDVIO	**1.48**	**1.72**
ATE rot [deg]	VIO	4.02	2.01
	VID	**3.50**	2.49
	VIMO	3.70	1.92
	HDVIO	3.70	**1.86**

Fig. 11: The plot and accompanying table show how our HDVIO performs in a setting where the training data for the learning-based component is gathered from a vision-based SLAM system in the outdoor sequences of the VID dataset. The flown trajectories are at low speeds below 3 m/s, which is why all four methods show good performance, with HDVIO being the more accurate.

We believe that the cause of this error is inaccuracy in the rotor/thrust coefficients used to compute the collective thrust inputs from the rotor speed measurements. It is visible from the force estimates along the z-axis, that, VID-Fusion, similar to VIMO (see Fig. 6), is not able to compensate for this systematic error. The RMSE achieved by VID-Fusion along the z-axis is 1.95 N. The overall RMSE achieved by VID-Fusion is 1.12 N. The RMSE achieved by HDVIO along the z-axis is 0.55 N. The overall RMSE achieve by HDVIO is 0.65 N. We do not include the estimates of VIMO in Fig. 10 for the sake of readability. The ATE_T achieved by VID-Fusion is the same as the one achieved by VIO, VIMO, and, HDVIO namely 0.02 m.

For completeness, we evaluate VID-Fusion also on the outdoor sequences. We present the relative errors and the ATE_T and ATE_R in Fig. 11. In these experiments, our hybrid drone model has been trained without using an external motion-capture system. The flown trajectories are at low speeds below 3 m/s, which is why all four methods show good performance, with HDVIO being the more accurate.

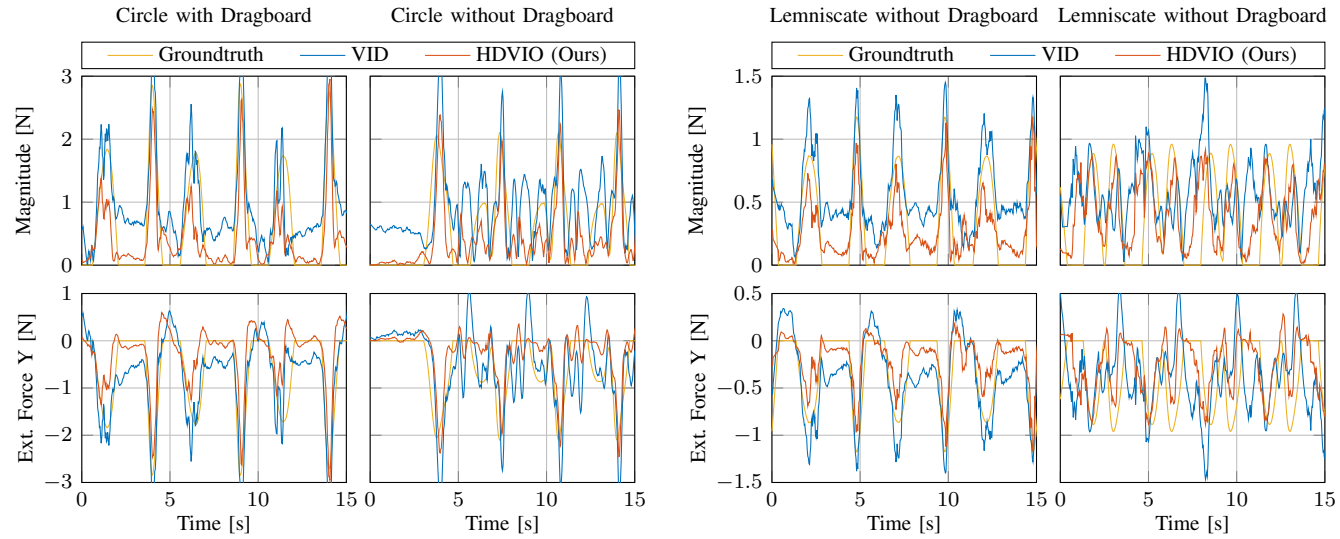


Fig. 12: Wind disturbance estimates in our real-world experiments. The magnitude and the y-axis component of the wind force estimated by HDVIO and VID-Fusion. Left: circle trajectory. Right: Lemniscate trajectory. In all the plots, it is visible that HDVIO achieves more accurate force estimates than VID-Fusion.

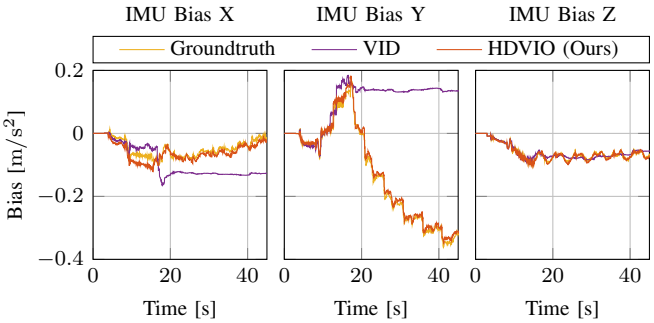


Fig. 13: Accelerometer bias estimates from our real-world experiments. The ground-truth bias is obtained from the VIO system. The estimates of our HDVIO match the ground-truth values, while, VID-Fusion estimates diverge along the x-axis.

3) Real-world Experiments: The estimates of the external force of VID-Fusion due to the wind gusts are in Fig. 12. We show the force along the y-axis, which is the direction of the wind gusts, after alignment to the world frame using the ground-truth orientations. From Fig. 12, it is clear that HDVIO outperforms VID-Fusion. Notably, VID-Fusion overestimates the wind force when the quadrotor enters the wind field. The reason for this behavior is that VID-Fusion jointly estimates the drag force with the external force.

Similar to VIMO, the measurement model in VID-Fusion introduces inconsistency in the estimates of the accelerometer bias resulting in decreased accuracy of the motion estimate. We show in Fig. 13 the accelerometer bias estimated by the VIO algorithm, by VID-Fusion, and by our method. This is done for a sequence where the quadrotor, equipped with the dragboard, flies a circle trajectory. Here, we consider the bias estimated by the VIO algorithm as a good approximation of the ground-truth values since the VIO achieves very high performance in this sequence thanks to the large number of visual features being tracked. The bias estimate of HDVIO closely matches the one of the VIO system, while the estimate of VID-Fusion converges to wrong values. We include the position and orientation absolute trajectory errors in Tab. V. In all 4 sequences, the rich texture environment simplifies the pose estimation problem. For this reason, the four algorithms achieve similar accuracy.

TABLE V: Experimental results from our real-world experiments. We use (d) to indicate that a dragboard was attached to the drone.

	ATE Position [m]				ATE Rotation [deg]			
	VIO	VID	VIMO	HDVIO	VIO	VID	VIMO	HDVIO
Circle (d)	**0.07**	0.10	0.1	**0.07**	2.02	2.31	**1.80**	2.06
Circle	**0.06**	**0.06**	0.08	**0.06**	1.21	1.37	1.19	**1.17**
Lemniscate (d)	0.38	0.53	0.34	**0.30**	**2.39**	**2.39**	2.93	2.81
Lemniscate	0.27	0.28	0.32	**0.20**	2.44	2.05	1.93	**1.84**

B. Clarification on Training with Vision-based SLAM Supervision

We show in Fig. 7, that our hybrid drone model can be trained using pose supervision from a vision-based SLAM system [34] on the VID dataset. However, the improvement in pose estimates is small compared to the baselines because these trajectories contain slow flights.

An alternative would have been to train with supervision from the vision-based SLAM system on the Blackbird dataset which contains faster flights. However, this was not possible because most of the train sequences did not include images at the time this work was carried out.