

# The Visual-Inertial-Dynamical Multirotor Dataset

Kunyi Zhang\*, Tiankai Yang\*, Ziming Ding, Chao Xu and Fei Gao

**Abstract**— Recently, the community has witnessed numerous datasets built for developing and testing state estimators. However, for some applications such as aerial transportation or search-and-rescue, the contact force or other disturbance must be perceived for robust planning and control, which is beyond the capacity of these datasets. This paper introduces a Visual-Inertial-Dynamical (VID) dataset, not only focusing on traditional six degrees of freedom (6-DOF) pose estimation but also providing dynamical characteristics of the flight platform for external force perception or dynamics-aided estimation. The VID dataset contains hardware synchronized imagery and inertial measurements, with accurate ground truth trajectories for evaluating common visual-inertial estimators. Moreover, the proposed dataset highlights rotor speed and motor current measurements, control inputs, and ground truth 6-axis force data to evaluate external force estimation. To the best of our knowledge, the proposed VID dataset is the first public dataset containing visual-inertial and complete dynamical information in the real world for pose and external force evaluation. The dataset<sup>1</sup> and related files<sup>2</sup> are open-sourced.

## I. INTRODUCTION

As the demand of GPS-denied navigation increases, state estimator plays an indispensable role in the real-time estimation of accurate robot poses. However, in many applications, there is an pressing requirement for the estimation of contact forces or other disturbances. For instance, in aerial transportation and delivery, drones are required to operate under a heavy payload, thus forming a multi-rigid body [1], whose mathematical model needs to be recognized online. In aerial manipulation, a drone equipped with a manipulator [2, 3] also requires precise force feedback to stabilize itself. Moreover, in autonomous flight with severe winds [4]–[6], the absence of an accurate external disturbances estimation significantly harms the effectiveness of planning and control. In response to the above, we propose a dataset containing multirotor dynamics as well as visual and inertial data.

In the decade, there are several datasets [7]–[11] for state estimation of aerial robots have been proposed. These datasets typically provide exteroceptive sensing, including imagery and range measurements, and proprioceptive data such as inertial measurements, for developing onboard pose estimation algorithms. The EUROC dataset [7] pioneers the visual-inertial benchmarking for drones, where a collection of sequences with accurate ground truth is pre-

All authors are with the State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China, and also with the Huzhou Institute of Zhejiang University, Huzhou 313000, China. E-mail: {kunyizhang, tkyang, zm ding and fgaoaa}@zju.edu.cn

\* Equal contributors.

<sup>1</sup><https://github.com/ZJU-FAST-Lab/VID-Dataset>

<sup>2</sup><https://github.com/ZJU-FAST-Lab/VID-Flight-Platform>



Fig. 1: Multirotor platform built for the dataset

sented. Although the EUROC dataset significantly pushes the boundary of visual SLAM, it only contains motions with moderate speed, thus not applicable to aggressive flights. The Zurich Urban MAV [8] dataset contains long-distance data recorded from a tethered multirotor flying in an urban environment without highly accurate ground truth. The Upenn Fast Flight [9] provides challenging sequences of fast speed and changing illumination but lacks ground truth as well. Recently, UZH proposes a first-person-view (FPV) dataset [10] to challenge the capability of existing state estimation algorithms in extreme conditions. This dataset consists of visual, event, and inertial streams collected in aggressive flights. Besides, the Blackbird Dataset [11] also targets high-speed applications. It provides fruitful scenes with photorealistic image rendering and also includes motor speed sensors.

Thanks to the above-mentioned public datasets, state estimation algorithms have been significantly improved in the last decade. The early visual odometry systems, such as SVO [12], ORB-SLAM [13] and DSO [14] estimate 6-DoF poses with merely images. Modern IMU-aided visual odometry systems, including MSCKF [15], Vins-mono [16], Open-vins [17] and ORB-SLAM3 [18], opt to utilize accelerometer and gyroscope as kinematic inputs to get more accurate state estimation. Besides, model-based visual-inertial odometry VIMO [19] and visual-inertial-dynamics state estimation VID-Fusion [20], which simultaneously estimate the pose and the external force, are a trending research topic emergent in the recent SLAM community. Since a multirotor’s dynamical characteristics can be determined entirely from the rotor model [21], dynamics is considered as a new information source that can further help state estimation, especially for highly aggressive motions. These estimators also obtain the external force, torque, or disturbance applied to the vehicle in real-time as a byproduct.

	EuRoC MAV [7]	Zurich Urban MAV [8]	UPenn Fast Flight [9]	UZH FPV dataset [10]	Blackbird dataset [11]	Ours
Camera <sup>a</sup>	20Hz	20Hz	40Hz	30/50Hz	120/60Hz <sup>e</sup>	60Hz <sup>Synced</sup>
IMU	200Hz	10Hz	200Hz	500/1000Hz	100Hz	400Hz <sup>Synced</sup>
Segmentation	n/a	n/a	n/a	n/a	60Hz	n/a
Ground truth	100Hz	n/a	n/a	20Hz	360Hz	100/5Hz
RTK	n/a	n/a	n/a	n/a	n/a	5Hz
Control inputs <sup>b</sup>	n/a	n/a	n/a	n/a	190Hz	100Hz <sup>Synced</sup>
Tachometer <sup>c</sup>	n/a	n/a	n/a	n/a	190Hz	1000Hz <sup>Synced</sup>
Ammeter <sup>c</sup>	n/a	n/a	n/a	n/a	n/a	1000Hz <sup>Synced</sup>
6-axis force sensor <sup>d</sup>	n/a	n/a	n/a	n/a	n/a	100Hz

TABLE I: Multirotor datasets comparison. <sup>a</sup> Frequencies are expressed as (general image)/(range image). <sup>b</sup> Control inputs mean the veritable inputs to motors, including target rotor speeds and motor currents. <sup>c</sup> Both are measurements of rotor speed and motor current. <sup>d</sup> Sensor data from a 6-axis force sensor. <sup>e</sup> Visual environments are synthesized in photorealistic simulation. <sup>Synced</sup> Sensor data are time-synchronized in the VID dataset.

Nevertheless, few datasets focus on the dynamical characteristics of multirotor platforms to support the aforementioned researches. The Blackbird dataset [11] takes one step towards this by providing a model of multirotor dynamics, which is simply described as the sum of all motor speeds and their corresponding thrusts. There are four additional motor speed sensors (as tachometers) embedded in the multirotor with customized optical motor encoders, and the thrust coefficient measured from experiments conducted in a wind tunnel. However, there are some drawbacks existing in the Blackbird dataset. Firstly, measurement by optical encoder cannot reflect motor speed accurately. Especially when speed is low or changing rapidly, it differs significantly from the control input. Secondly, only one propeller thrust coefficient is provided to represent the thrust coefficient of all motors. Actually for a multirotor, the thrust coefficients usually vary. Even in the Blackbird dataset, there is a significant difference in the motor speeds when hovering. Thirdly, the Blackbird dataset is insufficient for interactive perception of the environment due to the absence of ground truth of external force.

To bridge this gap and lay the foundation for developing a dynamics-aided state estimator for multirotors, we herein present the VID dataset with complete dynamical characteristics. The distinctive features of the VID dataset compared with others are demonstrated in TABLE I. Besides, there are several unique contributions of this work:

- 1) An available and completely public dataset for pose and external force estimation, which includes visual, inertial and dynamical data as well as the ground truth of external forces and poses.
- 2) An open-sourced flight platform equipped with versatile sensors for more advanced applications, along with fully identified dynamical and inertial parameters.
- 3) A general and exhaustive hardware synchronization solution for time alignment of images, inertial measurements, control inputs, rotor speeds, and motor currents, which leads to better sensor fusion.

This paper is structured as follows. In Section II, we introduce the multirotor platform, which is customized for

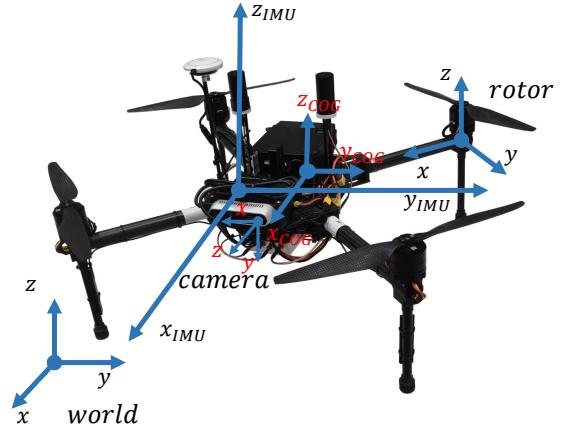


Fig. 2: Flight platform with coordinates.

data collection of versatile sensors. In Section III, we offer a single propulsion unit dynamics model and some essential calibration information. In Section IV, we evaluate our dataset by conducting several experiments with state-of-the-art algorithms. In Section V, we conclude with known limitations and future extensions of this work.

## II. DATASETS

### A. Flight platform

The platform (Fig. 1) is modified from a 650mm wheelbase multirotor<sup>3</sup>. To get high-precision measurements of rotor speed and motor current, a motor with a pair of orthogonal Hall position sensors and a current sampling unit<sup>4</sup> is selected as the propulsion unit. There are a depth camera<sup>5</sup>, a IMU integrated with the flight controller<sup>6</sup>, a customized circuit board<sup>2</sup> installed with 2 microcontrollers, separately

<sup>3</sup><https://www.dji.com/cn/matrice100>

<sup>4</sup><https://www.roboMaster.com/zh-CN/products/components/general/M3508>

<sup>5</sup><https://www.intelrealsense.com/depth-camera-d435i>

<sup>6</sup><https://www.dji.com/cn/n3>

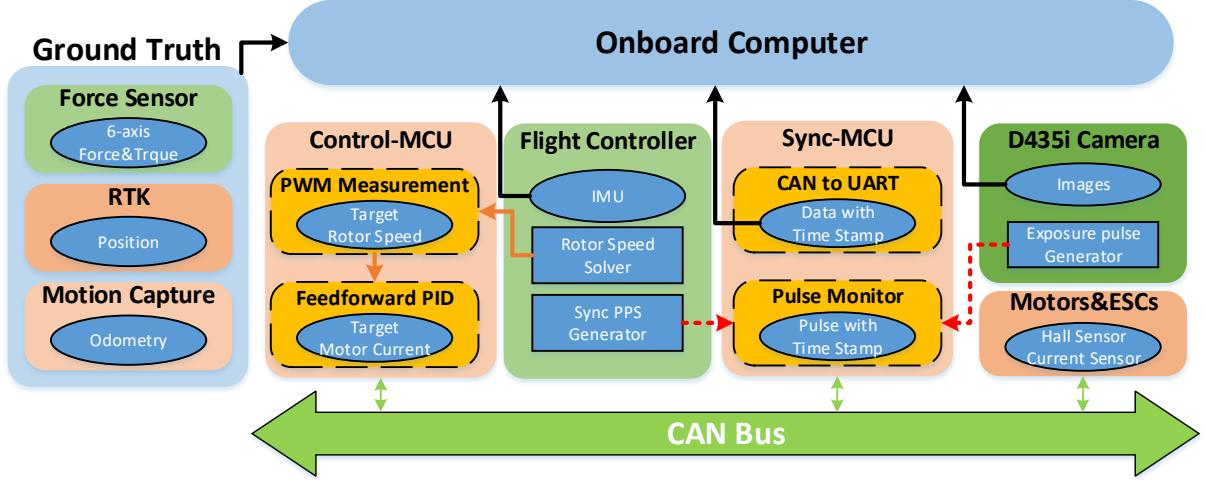


Fig. 3: Hardware system framework (Red lines: sync pluses)



Fig. 4: Experiment place. (1) Outdoor scene in a parking lot.  
(2) Indoor scene when recording data.

for motor control (Control-MCU) and hardware synchronization (Sync-MCU), and a computer<sup>7</sup> for data collection equipped on the multirotor. Here, we list common parameters of the multirotor and onboard sensors in TABLE III and show the relevant coordinates in Fig. 2. The hardware system framework is shown in Fig. 3.

### B. Data sources

1) *Rotor speed and motor current*: The motor feedbacks rotor speed and current messages with the Controller Area Network (CAN) at 1000Hz. The Sync-MCU monitors the above messages and sends them to the onboard computer by a Universal Asynchronous Receiver Transmitter (UART).

The units of rotor speed and current are revolutions-per-minute ( $rpm$ ) and  $\frac{20}{2^{16}} A$ .

2) *Control input*: The flight controller continuously generates control commands and sends them out by Pulse Width Modulation (PWM) waves. The Control-MCU measures the waves to get 100Hz target rotor speeds and uses a PID regulator with feedforward compensation to calculate target motor currents at 100Hz. Then, the Control-MCU sends the control inputs (target rotor speeds and motor currents) to the CAN bus, and the Sync-MCU converts the above CAN messages to UART packs and sends them to the computer.

3) *IMU*: The IMU measures acceleration, angular velocity and magnetometer data at 400Hz.

4) *Camera*: The camera offers stereo imagery and range measurements at 60Hz.

### C. Hardware time synchronization

To improve the quality of sensor fusion, the above onboard sensors are hardware synchronized by electronic pulses. Fig. 3 illustrates the relevant devices, pulses and dataflow.

	Symbol	Description
Time Reference System	<i>ros</i>	ROS time
	<i>fc</i>	flight controller
	<i>sm</i>	Sync-MCU
	<i>w</i>	Real world
Data	<i>pps</i>	PPS from flight controller
	<i>ppssm</i>	PPS detected by Sync-MCU
	<i>imu</i>	IMU data from flight controller
	<i>img</i>	Images
	<i>imgp</i>	Image pulses (60Hz) detected by Sync-MCU
	<i>md</i>	Motor data

TABLE II: Various time reference systems and data types.

We assume that the time of a certain reference system is  $t^{sys}$ , the time stamp sequence of data in a certain time reference system is  $T_i^{sys}$ , and the time difference between two time reference systems is expressed as

$$\Delta_{sys1}^{sys2}(t^w) = t^{sys2}(t^w) - t^{sys1}(t^w).$$

<sup>7</sup><https://www.dji.com/cn/manifold-2>

Prop.	Value	Description	Prop.	Value	Description
$l$	650 mm	Distance wheelbase	propeller	13 × 4.4 inch	Propeller model
$\eta_n$	1 rpm	Resolution of tachometer <sup>a</sup>	$\sigma_n$	1.459 rpm/ $\sqrt{Hz}$	Noise density of tachometer <sup>a</sup>
$\eta_I$	1.2207e <sup>-3</sup> A	Resolution of ammeter <sup>a</sup>	$\sigma_I$	4.950e <sup>-2</sup> A/ $\sqrt{Hz}$	Noise density of ammeter <sup>a</sup>
$\sigma_g$	0.001 rad/s/ $\sqrt{Hz}$	Noise density of gyr. <sup>b</sup>	$\sigma_{b_g}$	3.500e-5 rad/s <sup>2</sup> / $\sqrt{Hz}$	Gyr. random walk bias <sup>b</sup>
$\sigma_a$	0.100 m/s <sup>2</sup> / $\sqrt{Hz}$	Noise density of acc. <sup>b</sup>	$\sigma_{a_g}$	4.000e-3 m/s <sup>3</sup> / $\sqrt{Hz}$	Acc. random walk bias <sup>b</sup>
$\eta_f$	[0.1, 0.1, 0.1] N	Noise of force sensor <sup>c</sup>	$\eta_m$	[5e-3, 5e-3, 3e-3] N · m	Noise of torque sensor <sup>c</sup>
$T_I^C$	[1.8875e-1, 1.75e-3, 5.55e-2] m	Trs. IMU → camera <sup>d,e</sup>	$T_I^G$	[0.0, 0.0, 9.25e-3] m	Trs. IMU → geometric <sup>d,f</sup>
Image size	640px×480px	Image width and height	baseline	50mm	Distance between stereo pair

TABLE III: Multirotor characteristics. <sup>a</sup> Characteristics of the rotor tachometers and the motor ammeters are statistically calibrated. <sup>b</sup> IMU calibration result by Allan covariance. <sup>c</sup> Noise of force sensor parameters are tested by static experiments. <sup>d</sup>  $T_A^B$  represents translation from A frame to B frame in A frame. <sup>e</sup> Camera extrinsic translation parameter from CAD model. <sup>f</sup> We define the center of the multirotor as the geometric center with the same direction as the IMU frame, which is obtained from the CAD model.

### Algorithm 1 Pulse detection and validity check

```

1:  $valid_{imgp} \leftarrow false$ 
2: function PPS_DET(  $T_{ipps}^{fc}$ ,  $T_{ipps}^{ros}$ ,  $T_{ipppsm}^{sm}$ ,  $T_{ipppsm}^{ros}$  )
3:   if  $fabs(T_{ipps}^{ros} - T_{ipppsm}^{ros}) < 0.01$  sec then
4:      $\Delta_{sm}^{fc} \leftarrow T_{ipps}^{fc} - T_{ipppsm}^{sm}$ 
5:      $\hat{\Delta}_{fc}^{ros} \leftarrow T_{ipps}^{ros} - T_{ipps}^{fc}$ 
6:   end if
7: end function
8: function IMGPULSE_DETECT( $T_{iimgp}^{sm}, T_{iimgp}^{ros}$ )
9:   if  $valid_{\hat{\Delta}_{sm}^{fc}}$  then
10:     $\hat{T}_{ipppsm}^{ros} \leftarrow T_{iimgp}^{sm} + \Delta_{sm}^{fc} + \hat{\Delta}_{fc}^{ros}$ 
11:    if  $fabs(\hat{T}_{ipppsm}^{ros} - T_{ipppsm}^{ros}) < 0.01$  sec then
12:      Output:  $valid_{imgp} \leftarrow true$ 
13:      Output:  $T_{recimgp}^{fc} \leftarrow T_{iimgp}^{sm} + \Delta_{sm}^{fc}$ 
14:      Output:  $T_{recimgp}^{ros} \leftarrow T_{iimgp}^{ros}$ 
15:    end if
16:   end if
17: end function

```

We define the symbols of time reference systems and data, which are listed in TABLE II.

When the time synchronization system starts to work, the flight controller generates a pulse-per-sec (PPS) and marks the IMU data corresponding to the pulse (one pulse every 400 IMU data). The timestamps of IMU and PPS from the flight controller are respectively  $T_{iimu}^{fc}$  and  $T_{ipps}^{fc}$ . The camera generates a 60Hz pulse, which corresponds to the time when the camera starts to expose. The Sync-MCU monitors both pulses and stamps them with its internal time ( $T_{ipppsm}^{sm}$  and  $T_{iimgp}^{sm}$ ). When receiving the rotor speeds and motor currents from the CAN bus, the Sync-MCU stamps them with  $T_{imd}^{sm}$ . Besides, Robot Operation System (ROS) also stamps the data with

$$\{T_{ipps}^{ros}, T_{ipppsm}^{ros}, T_{iimu}^{ros}, T_{iimgp}^{ros}, T_{iimgp}^{sm}, T_{imd}^{sm}\}.$$

Afterwards, the computer uses Algorithm 1 to check and process the pulse messages and Algorithm 2 to align the data from different time reference system to system  $fc$  (regarded as system  $w$ ).

### Algorithm 2 Time stamp alignment

```

1: Notation:  $T_{recimg}^{fc}, T_{recimg}^{ros}, valid_{img} \leftarrow false$ 
2: function MOTOR_DATA_ALIGN( $T_{imd}^{sm}$ )
3:    $T_{imd}^{fc} \leftarrow T_{imd}^{sm} + \Delta_{sm}^{fc}$ 
4:   Output:  $T_{imd}^w \leftarrow T_{imd}^{fc}$ 
5: end function
6: function IMU_ALIGN( $T_{iimu}^{fc}$ )
7:   Output:  $T_{iimu}^w \leftarrow T_{iimu}^{fc}$ 
8: end function
9: function IMG_ALIGN( $T_{iimg}^{ros}$ )
10:  if  $valid_{imgp}$  then
11:    if  $T_{iimg}^{ros} - T_{recimgp}^{ros} \in [0.007, 0.016]$  sec then
12:       $valid_{img} \leftarrow true$ 
13:       $T_{recimg}^{fc} \leftarrow T_{recimgp}^{ros}, T_{recimg}^{ros} \leftarrow T_{iimg}^{ros}$ 
14:      Output:  $T_{imd}^w \leftarrow T_{recimg}^{fc}$ 
15:    else
16:      if  $valid_{img}$  then
17:         $\hat{T}_{iimg}^{fc} \leftarrow T_{recimg}^{fc} + (T_{iimg}^{ros} - T_{recimg}^{ros})$ 
18:        Output:  $T_{imd}^w \leftarrow \hat{T}_{iimg}^{fc}$ 
19:      end if
20:    end if
21:  end if
22: end function

```

#### D. Ground truth

1) *Indoor pose:* In an 18m × 9m × 3.5m indoor environment, we use a motion capture system<sup>8</sup> composed of 15 cameras to provide the multirotor’s 6-DOF pose at 100Hz.

2) *Outdoor pose:* In a 100m × 35m outdoor parking environment (Fig. 4), the multirotor is equipped with a Global-Navigation-Satellite-System (GNSS) receiver<sup>9</sup>, which subscribes to a Real-Time Kinematic (RTK) location service<sup>10</sup> and receives the multirotor’s pose at a frequency of 5Hz.

3) *External force:* As shown in Fig. 4, we fix a 6-axis force sensor<sup>11</sup> on the ground. Indoors, we use an elastic

<sup>8</sup><https://www.vicon.com/hardware/cameras/vantage>

<sup>9</sup><https://www.zhdgps.com>

<sup>10</sup><https://mall.qxwz.com/market/services/FindCM>

<sup>11</sup><https://robotiq.com/products/ft-300-force-torque-sensor>

	Environment	Feature	Trajectory	Weight (gram)	Duration (second)	Imu Imagery	Dynamical data	RTK	Ground truth	Force sensor
1	Outdoor	Fast, with yaw	Rectangle	3547.2	113.21					
2		Slow, with yaw	Rectangle	3547.2	175.38					
3		with yaw	Round	3541.7	147.34	✓	✓	✓		
4		without yaw	Round	3541.7	105.85					
5		with yaw	8-character	3541.7	184.29					
6		without yaw	8-character	3541.4	243.73					
7	Night	Fast, with yaw	Rectangle	3460.0	133.58					
8		Slow, with yaw	Rectangle	3541.6	179.48	✓	✓	✓		
9		without yaw	Round	3541.6	82.07					
10		without yaw	8-character	3547.7	121.87					
11	Indoor	Loadless	Hovor	3096.1	79.04					
12			Round	3096.1	117.89	✓	✓		✓	
13			8-character	3096.1	109.17					
14	Indoor	Loaded	Hovor	3372.2	80.11					
15			Round	3372.2	107.15	✓	✓		✓	
16			8-character	3372.2	138.41					
17	Indoor	Rope pulled	Random	3101.5	126.53	✓	✓		✓	✓
18			Random	3101.5	155.38					
19	Motor test	Channel 1 Channel 2 Channel 3 Channel 4			45.04					
20					45.08					
21					43.00					
22					43.84					

TABLE IV: Dataset information

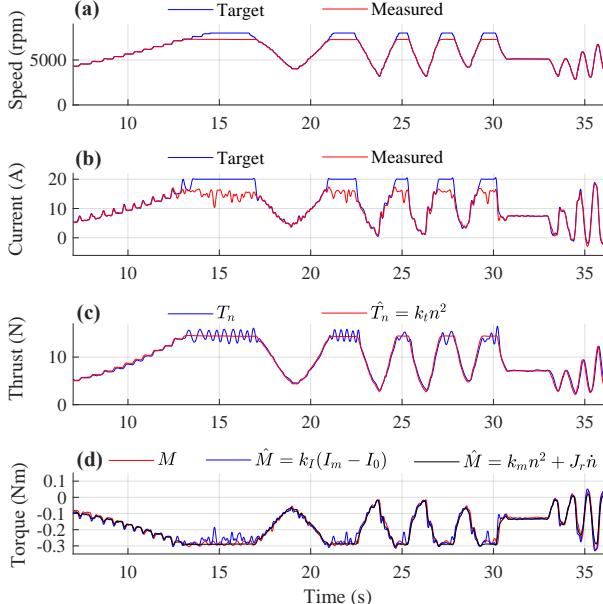


Fig. 5: Propulsion system unit test (Seq. 19). (1) Target rotor speed and measured rotor speed; (2) Target motor current and measure motor current; (3) The thrust measured by the force sensor and the thrust fitted from rotor speed; (4) The torque measured by the force sensor and the torque fitted by rotor speed and motor current.

rope with one end fixed on the force sensor and the other end pulling the multirotor to obtain the ground truth of the external forces acting on the multirotor during flight.

### III. DATA VALIDATION

#### A. Dynamical system modelling

In order to establish the complete dynamics of the multirotor, a basic aerodynamic modelling of a single propeller propulsion system is required. Here, the Rankine-Froude momentum theorem [22] models the steady-state thrust and counter torque generated by a hovering rotor, as shown below:

$$T := C_T \rho A R^2 \omega^2, \quad (1)$$

$$M := C_M \rho A R^3 \omega^2, \quad (2)$$

where, for any rotor,  $C_T$  is the thrust coefficient,  $C_M$  is the torque coefficient.  $\rho$  is the density of air,  $A$  is the rotor disk area,  $R$  is the radius of rotor and  $\omega$  is the angular velocity.  $C_T$  and  $C_M$  depend on rotor geometry and profile.

We replace angular velocity  $\omega$  (unit:  $rad/s$ ) with the rotor speed  $n$  (unit:  $rpm$ ), and briefly express Eq. (1) and (2) as the following lumped parameters:

$$T = k_t n^2, \quad (3)$$

$$M = k_m n^2 + J_r \dot{n}, \quad (4)$$

where,  $T$  and  $M$  represent thrust (unit:  $N$ ) and counter torque (unit:  $N \cdot m$ ), which are produced by a single rotating propeller, and  $k_t$ ,  $k_m$  and  $J_r$  mean thrust coefficient, counter torque coefficients and rotor's moment of inertia of z-axis, respectively. It should be noted that, compared with Eq. (2), Eq. (4) adds the additional torque generated when the rotor accelerates and decelerates [23].

In order to identify the above models, we mount a motor fixed with a propeller to the 6-axis force sensor, and obtain the model parameters by analyzing its tachometer and thrust

data. Fig. 5 shows the aerodynamics of a propulsion unit, the complete identification parameters are listed in the TABLE V. The thrust coefficients of the four channels are similar, but the torque coefficients differ significantly, especially between the clockwise and counterclockwise propellers.

Apart from rotor speed  $n$ , motor armature current  $I_m$  is also provided for redundant input resource. The counter torque provided by the rotor is equivalent to the output torque of the motor, and according to electric machinery [24], the latter is proportional to the armature current. We can build another torque model as follows:

$$M = M_e - M_0 \\ = k_I(I_m - I_0), \quad (5)$$

where  $k_I$  is the current torque coefficient.  $I_0$  and  $M_0 = k_I I_0$  represent actual no-load current and no-load torque, respectively. In other words, we can only drive the rotor to rotate if armature current  $I_m$  is greater than  $I_0$ . The forth chart of Fig. 5 also adds a comparison with Eq. 5. The corresponding current torque coefficients of the multirotor are also shown in TABLE V.

Besides, the relationship of rotor speed and motor current can be derived from Eq. 4 and 5 and formulated as follow:

$$I_m = \frac{k_m n^2 + J_r \dot{n}}{k_I} + I_0. \quad (6)$$

Rotor speed can be converted into motor current according to the above relationship by Control-MCU, which is also shown in Fig. 6. The above is the complete dynamical model of a single propulsion unit.

### B. Rotor speed and motor current

For more rapid motor response, we use corresponding motor current calculated by Eq. 6 to output to the ESC, rather than controlling the motor directly by PWM on our platform. In order to verify the accuracy of the rotor speed and motor current and the validity of dynamical model of the propulsion unit, we compare target rotor speed and its corresponding motor current with their measured values in the same sequence demonstrated in Fig. 5. As shown, the actual (measured) values can closely track the target values by the aforementioned method. In fact, however, neither of them can reach the theoretical maximum due to the insufficient driving forces of the ESCs. By the way, the measured motor current data is drawn after filtering due to the high noise level.

### C. Calibration

In order to make better use of the sensor data, it is of great importance to get the characteristics of the sensors themselves, the extrinsic parameters between different sensors and the inertial parameters of the flight platform.

The extrinsic parameters between the IMU and the camera are calibrated by Kalibr [25] or measured from CAD

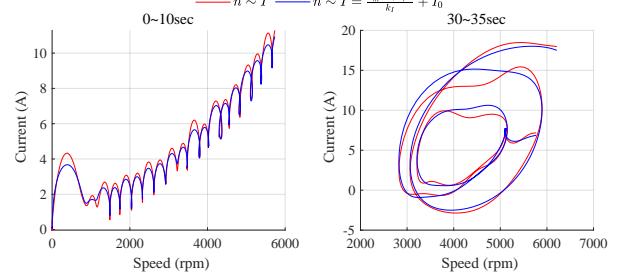


Fig. 6: Comparison of the relationship between rotor speed and current with measured value and fitting model (Seq. 19).

	$k_t$ (N/rpm <sup>2</sup> )	$k_m$ (N · m/rpm <sup>2</sup> )	$J_r$ (kg · m <sup>2</sup> )	$k_I$ (N · m/A)
1	2.6890e-7	-5.6343e-9	-9.8316e-6	-2.7506e-5
2	2.8190e-7	4.7180e-9	8.5648e-6	2.2204e-5
3	2.7263e-7	-5.7012e-9	-9.7166e-6	-2.7694e-5
4	2.7741e-7	4.8260e-9	9.8131e-6	2.1910e-5

TABLE V: Propeller dynamics model. The number represents the designed multirotor's rotor or motor channel, counting from the upper right corner counterclockwise. A negative sign means the opposite direction.

drawings. IMU intrinsic parameters are calibrated by Kalibr-allan<sup>12</sup>. Besides, we also quantitatively analyze the noise characteristics of the rotor tachometers and current ammeters. More detailed information can be found on the website<sup>1</sup>.

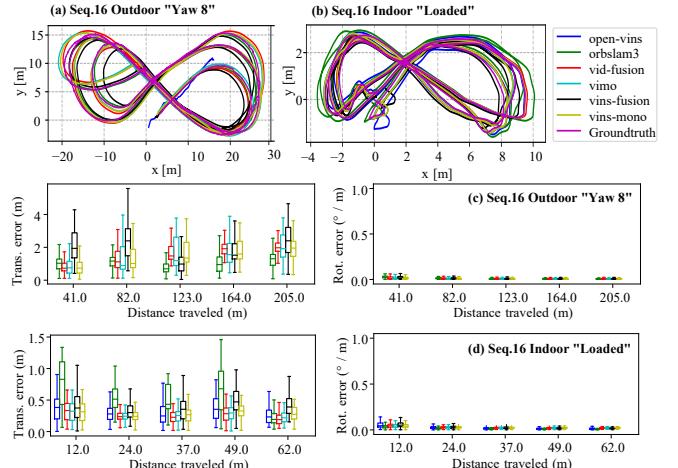


Fig. 7: Top view and translation&rotation error of the sequence outdoor-yaw-8 and indoor-loaded-8

## IV. EXPERIMENTS

### A. Pose estimation

Robust and accurate pose estimation is beneficial for external force estimation, so we select two scenarios to show the accuracy of some state-of-the-art algorithms before external force estimation. Openvins, ORB-SLAM3 and

<sup>12</sup><https://github.com/ethz-asl/kalibr/wiki/IMU-Noise-Model>

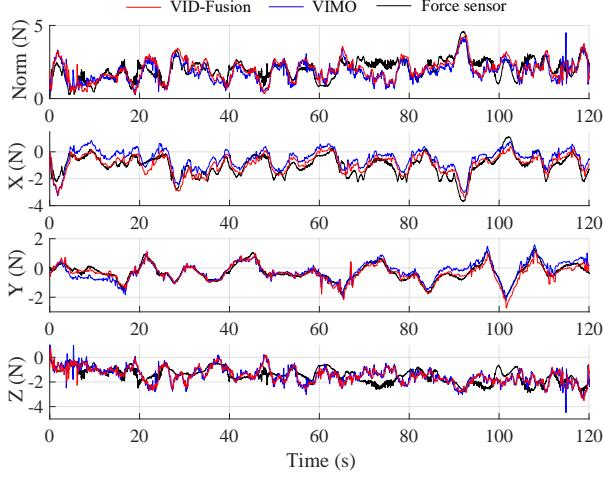


Fig. 8: External force while being pulled by a rope.

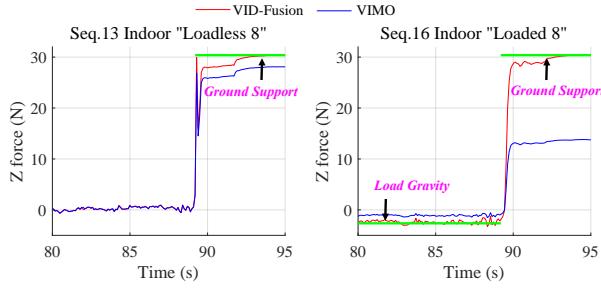


Fig. 9: External force estimation with&without load.

VINS-Fusion work with binocular images and inertial measurements, VINS-mono takes monocular images and inertial measurements, VIMO and VID-Fusion add some dynamical model and rotor speed compared to VINS-mono. The last two regard the multirotor as a mass point model, so we input the algebraic sum of the 4 propeller thrusts (as described in III-A) as the driving force to the algorithms.

It is worth mentioning that, all algorithms take the same camera intrinsic, extrinsic parameters, imu parameters, and dynamics parameters with loop-closure disabled.

As shown in Fig. 7, Openvins, VINS-mono, VIMO and VID-Fusion outperform over the other two algorithms, yet VIMO and VID-Fusion with extra dynamical information do not show a significant improvement over VINS-mono in indoor sequences. ORB-SLAM3 works best outdoors, while Openvins hangs due to the presence of a moving vehicle in the initial field of view. Evaluation results are provided by [26].

The VID dataset is competent for the evaluation of vision-based pose estimation algorithms, and we choose VINS-mono, which works relatively well indoors, to do a comparative test in external force estimation.

### B. External force estimation

The external force estimation performance of VID-Fusion and VIMO is evaluated in three scenarios including flights without load (seq. 13), with load (seq. 16) and with a elastic

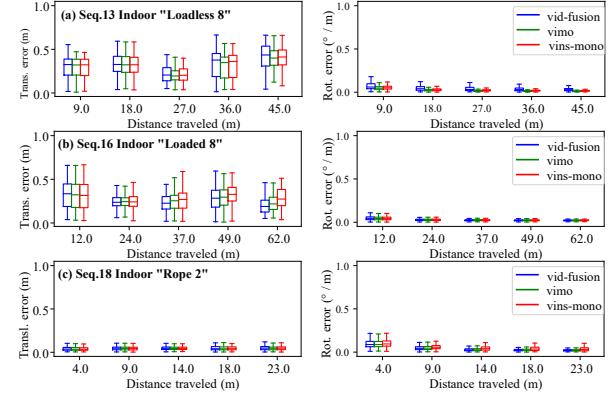


Fig. 10: VIMO, VID-Fusion and VINS-mono pose estimation.

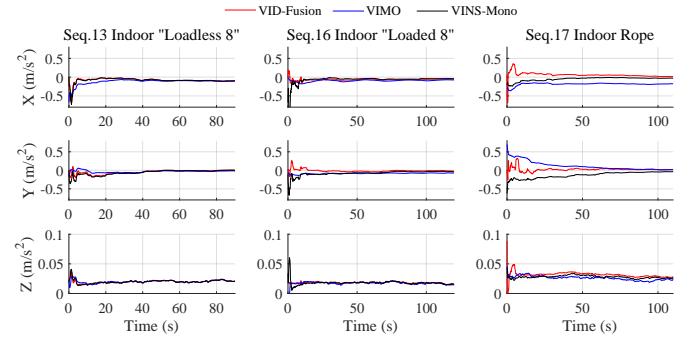


Fig. 11: IMU acceleration bias estimation.

rope (seq. 17).

Fig. 9 illustrates that both algorithms can sense and estimate the weight of the load during flight and the ground support after landing, which corresponds to the multirotor weight without load. But VIMO can not estimate the force correctly after the multirotor is loaded, mainly because it models the external force as a zero-mean noise. Fig. 8 demonstrates the external force estimation from the rope pulling experiment (seq. 17). Both algorithms can barely track the change in the contact force, but also have errors in the z-axis.

To explore the two algorithms in more detail, VINS-mono is chosen to compare their pose and acceleration bias estimation in the same scenarios.

1) *without load*: The three algorithms have similar performance in pose estimation (Fig. 10-a) and all the estimated acceleration biases converge to the same value (Fig. 11-a).

2) *with a fixed load*: Although VIMO and VID-Fusion perform better in pose estimation (Fig. 10-b), the former could not estimate external force correctly (Fig. 9) with a different convergence of bias estimation (Fig. 11-b).

3) *external force changes*: In the scenario where a rope is attached between the multirotor and the force sensor, there is little difference among the three algorithms in pose estimation (Fig. 10-c). VIMO and VID-Fusion can't track the external force ideally (Fig. 8), which may lead to biased estimation of acceleration bias (Fig. 11-c).

## V. DISCUSSION AND FUTURE WORK

### A. Conclusion

In this paper, we develop a multirotor platform carrying various time-synchronized sensors. Based on this, we present the VID multirotor dataset for state and external force estimation. By presenting several experiments, we validate the utilities of our proposed dataset and identify some problems with existing algorithms for external force estimation.

- 1) VIMO and VID-Fusion estimate a joint external force through the known dynamical model. Once the mass and the thrust coefficients are provided inaccurately, they would be counted in the external force.
- 2) In both algorithms, the multirotor dynamics do not contribute significantly to pose estimation either in the absence of external forces or in the presence of a constant load. And if the external force varies, it affects the bias estimation, or worse, can cause the estimator to crash when the external force changes too much, for example, during take-off or landing.
- 3) It is not reasonable to treat all these unmodelled forces as a combined external force, since the multirotor is a rigid body and there are drag force, gyroscopic torque, etc. during flight.

### B. Limitations

Our dataset still has several known limitations which need to be addressed in the near future.

- 1) Due to insufficient motor power, the multirotor has limited flexibility and load capacity, which results in unaggressive flight data.
- 2) Since there is no feasible recording environment currently, ground truth of the external torque is absent.
- 3) The dynamical model of the propulsion unit is individually identified, while the drag and other forces and torques of the multirotor are difficult to calibrate in various scenes.

### C. Future work

It is our sincere hope that the VID dataset will facilitate scientific research and industrial applications in related fields.

In the future, we plan to improve the capabilities of drones by equipping more perceptual sensors (such as event camera or lidar), offering more detailed parameters, recording more aggressive and challenging sequences, providing ground truth of external torques which are lacking currently.

## REFERENCES

- [1] C. Y. Son, H. Seo, D. Jang, and H. J. Kim, “Real-time optimal trajectory generation and control of a multi-rotor with a suspended load for obstacle avoidance,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1915–1922, 2020.
- [2] S. Kim, S. Choi, H. Kim, J. Shin, H. Shim, and H. J. Kim, “Robust control of an equipment-added multirotor using disturbance observer,” *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1524–1531, 2018.
- [3] D. Lee, D. Jang, H. Seo, and H. Jin Kim, “Model predictive control for an aerial manipulator opening a hinged door,” in *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, 2019, pp. 986–991.
- [4] H. Seo, D. Lee, C. Y. Son, C. J. Tomlin, and H. J. Kim, “Robust trajectory planning for a multirotor against disturbance based on hamilton-jacobi reachability analysis,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3150–3157.
- [5] J. Ji, X. Zhou, C. Xu, and F. Gao, “Cmpcc: Corridor-based model predictive contouring control for aggressive drone flight,” *arXiv preprint arXiv:2007.03271*, 2020.
- [6] Y. Wu, Z. Ding, C. Xu, and F. Gao, “External forces resilient safe motion planning for quadrotor,” *arXiv preprint arXiv:2103.11178*, 2021.
- [7] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achterlik, and R. Siegwart, “The euroc micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
- [8] A. L. Majdik, C. Till, and D. Scaramuzza, “The zurich urban micro aerial vehicle dataset,” *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 269–273, 2017.
- [9] K. Sun, K. Mohta, B. Pfommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, “Robust stereo visual inertial odometry for fast autonomous flight,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.
- [10] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, “Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6713–6719.
- [11] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, “The blackbird uav dataset,” *The International Journal of Robotics Research*, p. 0278364920908331, 2020.
- [12] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 15–22.
- [13] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [14] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.
- [15] M. Li and A. I. Mourikis, “High-precision, consistent ekf-based visual-inertial odometry,” *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.
- [16] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [17] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, “Openvins: A research platform for visual-inertial estimation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.
- [18] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam,” *arXiv preprint arXiv:2007.11898*, 2020.
- [19] B. Nisar, P. Foehn, D. Falanga, and D. Scaramuzza, “Vimo: Simultaneous visual inertial model-based odometry and force estimation,” *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2785–2792, 2019.
- [20] Z. Ding, T. Yang, K. Zhang, C. Xu, and F. Gao, “Vid-fusion: Robust visual-inertial-dynamics odometry for accurate external force estimation,” *arXiv preprint arXiv:2011.03993*, 2020.
- [21] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *IEEE Robotics and Automation magazine*, vol. 19, no. 3, pp. 20–32, 2012.
- [22] J. Leishman, *Principles of Helicopter Aerodynamics*, ser. Cambridge Aerospace Series. Cambridge University Press, 2016. [Online]. Available: <https://books.google.co.jp/books?id=uscAMQAACAAJ>
- [23] 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*, vol. 5, no. 3, pp. 3861–3867, 2020.
- [24] S. Chapman, *Electric Machinery Fundamentals*. McGraw-Hill, 2012.
- [25] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1280–1286.
- [26] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.