# Online Self-Calibration for Visual-Inertial Navigation Systems: Models, Analysis and Degeneracy

Yulin Yang - yuyang@udel.edu
Patrick Geneva - pgeneva@udel.edu
Xingxing Zuo - xingxing.zuo@tum.de
Guoquan Huang - ghuang@udel.edu

Department of Mechanical Engineering
University of Delaware, Delaware, USA

**RPNG**

# Contents

**Abstract**

In this paper, we study in-depth the problem of online self-calibration for robust and accurate visual-inertial state estimation. In particular, we first perform a complete observability analysis for visual-inertial navigation systems (VINS) with full calibration of sensing parameters, including IMU/camera intrinsics and IMU-camera spatial-temporal extrinsic calibration, along with readout time of rolling shutter (RS) cameras (if used). We investigate different inertial model variants containing IMU intrinsic parameters that encompass most commonly used models for low-cost inertial sensors. With these models, the state transition matrix and visual measurement Jacobians are analytically derived and the observability analysis of linearized VINS with full sensor calibration is performed. The analysis results prove that, as intuitively assumed in the literature, VINS with full sensor calibration has four unobservable directions, corresponding to the system's global yaw and translation, while all sensor calibration parameters are observable given fully-excited 6-axis motion. Moreover, we, for the first time, identify primitive degenerate motions for IMU/camera intrinsic calibration, which, when combined, may produce complex degenerate motions. This result holds true for the different inertial model variants investigated in this work and has significant impacts on practical applications of online self-calibration to many robotic platforms. Extensive Monte-Carlo simulations and real-world experiments are performed to validate both the observability analysis and identified degenerate motions, showing that online self-calibration improves system accuracy. We compare the proposed *online* self-calibration on commonly-used IMUs against the state-of-art *offline* calibration toolbox Kalibr, and show that the proposed system achieves better consistency and repeatability. As sensor calibration plays an important role in making real VINS work well in practice, based on our analysis and experimental evaluations, we also provide practical guidelines for how to perform online IMU-camera sensor self-calibration.

# 1 Introduction

Due to the decreasing cost of integrated inertial/visual sensor rigs, visual-inertial navigation system (VINS), which fuses high-rate inertial readings from an IMU and images of the surrounding environment from a camera, has gained great popularity in 6 degree-of-freedom (DoF) motion tracking for mobile devices and autonomous robots – such as micro aerial vehicles (MAV) [1], self-driving cars [2], unmanned ground vehicles (UGV) [3, 4] and smart phones [5, 6] – during the past decades [7]. Many efficient and robust VINS algorithms based on filtering [8, 5, 9, 10, 11] or batch least squares solvers [12, 13, 14, 15, 16] have been developed in recent years to address this pose estimation problem.

There are many factors which attribute to VINS performance, such as visual feature tracking/triangulation, velocity/biases initialization and sensor calibration. Among them, robust and accurate sensor calibration – including the rigid transformation between sensors (spatial calibration), time offset between IMU-camera (temporal calibration), image line readout time for rolling shutter (RS) cameras, and IMU/camera intrinsics – is crucial, especially when plug-and-play visual-inertial sensor rigs with widely available off-the-shelf low-cost IMUs and rolling shutter cameras are deployed. In addition, sensor calibration itself can be changed due to extended usage, sensor replacement and environmental effects such as varying temperature, humidity, vibrations, non-rigid mounting, and among others. For example, IMU biases and intrinsics suffer from the temperature and humidity changes [17], and rigid transformation between IMU and camera can vary if the sensor is replaced or subjected to vibration. As such, online sensor self-calibration in VINS has attracted significant attentions and research efforts [5, 6, 18, 11, 19] in recent years, due to its potential to handle poor prior calibration or calibration changes, which can degrade the state estimate accuracy in the case that these calibrations are treated to be true.

System observability analysis for VINS with online IMU-camera [20, 21, 22] or IMU/camera intrinsic [23, 19] calibration has also been carried out to show that these calibration parameters can be identified given fully excited motions. However, complete analysis for VINS with full calibration parameters – including IMU/camera intrinsics, IMU-camera rigid transformation, temporal time offset, and camera RS readout time – is still absent from the existing literature.

Blindly performing online calibration is risky, as in most cases domain knowledge on specific motions and prior distribution choices are needed to ensure calibration can converge consistently [24]. In the meantime, existing research efforts [25, 18, 19] have also identified several basic motion profiles, termed degenerate motions, that cause online sensor self-calibration failures. In this work, we investigate degenerate motions which impact the deployment of VINS on mobile robots, which typically have constrained motions, when jointly estimating IMU/camera intrinsics, IMU-camera spatial-temporal calibration, and RS readout time. For example, aerial and ground vehicles can only perform a few motion profiles due to their under-actuation, and can easily "fall" into degenerate conditions for calibration during typical deployment. As compared to our investigation into degenerate motions when performing full-parameter self-calibration, and their impacts especially for under-actuated autonomous robots, most approaches on VINS sensor self-calibration are limited to either handheld or trajectory segments involving rich motion information [17, 26].

In this paper, we build an accurate and robust monocular VINS estimator with full self-calibration. We also investigate in-depth the observability analysis for visual-inertial self-calibration and perform degenerate motion analysis for *all* calibration parameters. In particular, the main contributions of this work include:

- An efficient filter-based visual-inertial estimator capable of performing self-calibration for all spatial-temporal extrinsic and intrinsic parameters.

- We perform a complete observability and degeneracy analysis for the proposed visual-inertial models and, for the first time, identify the degenerate motions that cause IMU and camera intrinsic parameters to become unobservable.

- Extensive simulations and real-world experiments are performed to verify the parameter convergence of the estimator with online self-calibration under fully-excited 6DoF motion and a series of identified degenerate motions of practical significance.

- We show that under general motion, self-calibration is necessary to achieve accurate pose estimation in a robust manner for consumer grade sensors, which continue to become more prevalent, with only minimal computational impact. Additionally, we show that degenerate motions can and do have a significant negative impact on the performance of the estimator, leading to a series of guideline recommendations.

The rest of the paper is organized as follows: after reviewing the related work in Section 2 and estimation preliminaries in Section 3, we present the sensing models including inertial and camera models in Section 4. In Section 5, we derive the lienarized system dynamics and measurement model of the VINS with full sensor calibration. Based on that, we perform the observability analysis and degenerate motion identification in Sections 6 and 7, while the proposed estimator is presented in Section 8. In Sections 9, 10, 11 and 12, we extensively validate our analysis and estimator through both simulations and real-world experiments. Finally, we offer discussions and final remarks in Sections 13 and 14.

# 2 Related Works

Extensive works have studied online or offline IMU and camera calibration for VINS. However, the joint self-calibration of all the calibration parameters for visual and inertial sensors (including IMU/camera intrinsics, IMU-camera spatial-temporal calibration and RS readout time) was not investigated sufficiently, and the observability analysis and degenerate motion identification for the complete VINS with calibration are still missing from the literature. In terms of the complete parameter calibration for VINS, the related works can be divided into the following four categories:

## 2.1 Camera Calibration

Visual-only offline calibration of camera intrinsic parameters is well studied [27]. For example, [28] demonstrated camera self-calibration without a pattern, based on which [29] and [30] proposed to use sum of Gaussian filters. Recently, [31] extended the above works to use RGB video with objects of non-rigid shape for camera self-calibration.

Many works have also focused on RS camera calibration, especially for the image line readout time. For instance, after formulating the geometric models of RS cameras and investigating how RS affects the image generation, [32] leveraged flashing LED lights to calibrate the RS readout time. Similarly, [33] used continuous time trajectory representation to model RS effects and performed calibration with an April tag pattern [34]. However, these methods assume known camera intrinsics (including distortion parameters) when calibrating the RS readout time.

[35] investigated the self-calibration of multiple omnidirectional RS cameras. They first initialized the rigid transformation and time offset between cameras based on the structure-from-motion (SFM) trajectories generated from each monocular camera with global shutter (GS) assumption. Then, all the camera related parameters (i.e., the rigid transformation, time offset, camera intrinsics, and RS readout time for each camera) are refined by a bundle adjustment (BA) with all the camera measurements. [36] presented the first minimal solution to absolute pose estimation of a RS camera with unknown focal length and unknown radial distortion, from seven point correspondences. With the proposed solvers they can achieve accurate solutions for camera poses, RS readout time, focal length and radial distortion. Note that both [36] and [35] are offline calibration algorithms and rely only on visual sensors.

[37] proposed to combine a gyroscope for RS camera image correction with natural scenes. However, it requires the pre-calibration of rotation between gyroscope and RS camera and the motion of the camera is restricted to rotation only. [23] investigated online camera intrinsic calibration (only focal length and principal points) within a VINS framework.

In contrast to the above works only focusing on camera calibration, our proposed method fuses the measurements of an IMU and a monocular camera, and provides online calibration of all the camera parameters including camera intrinsics and RS readout time, Both radial-tangential (*radtan*) and equivalent-distant (*equidist*) distortion [38] are supported.

## 2.2 IMU Intrinsic Calibration

Generally, the gyroscope and acceleration biases are needed for accurate inertial modeling. They are both modeled as random walks and estimated as part of IMU states. It is a common practice to estimate biases online in VINS such as [39], [21], [12] and [8].

Besides these biases, the IMU intrinsic parameters – including the scale correction and axis misalignment for gyroscope and accelerometer, the rotation from gyroscope or accelerometer frame to IMU frame, and the g-sensitivity – also need to be calibrated offline or online, especially for

low-cost inertial sensors. [40] calibrated the IMU's scale correction and axis misalignment for aerial vehicles by solving a least squares problem with known special sensor motions. [41] improved the IMU pre-integration [13] to incorporate the IMU intrinsic parameters in a keyframe based VINS algorithm for online self-calibration. [42] studied IMU intrinsic calibration within multi-state constrained Kalman filter (MSCKF) by using a stereo camera and an IMU sensor, where they also examined the inertial calibration results under planar and random motions.

Building upon our prior work [19], in which we have investigated online IMU intrinsic calibration with the minimal sensor configuration of a single IMU and a monocular camera and compared the performance of four different IMU intrinsic model variants in VINS, in this work, we study 18 different IMU intrinsic model variants which can encompass or be equivalent to most published IMU models for inertial navigation and perform online self-calibration. Comprehensive degenerate motion analysis, which can cause online self-calibration to fail, is also provided.

## 2.3   Joint IMU-Camera Self Calibration

Since VINS fuses IMU measurements and camera images, the joint calibration of IMU-camera parameters is preferred to improve system accuracy and robustness. Extensive works have studied joint sensor calibration in VINS. For instance, [20] proposed to use an extended Kalman filter (EKF) for the spatial calibration (i.e. the rigid transformation between the camera and IMU) of VINS and performed an observability analysis. They showed that the rigid transformation is not fully observable under one-axis rotation. [43] proposed to use the recursive Sigma-Point Kalman filter to estimate IMU intrinsics and IMU-camera spatial parameters with measurements from an IMU and a monocular camera. However, [20] and [43] did not calibrate the camera intrinsics or IMU-camera time offset and both relied on calibration chessboards.

[38] developed the well-known calibration toolbox: Kalibr, a continuous-time spline-based batch estimator, for IMU-camera extrinsics, time offset and camera intrinsics calibration. [44] extended Kalibr to incorporate and estimate IMU intrinsics (including scaling parameters, axis misalignments, and g-sensitivity). [45] further extended the above work to calibrate readout time for RS cameras. [46] formulated a maximum likelihood estimation problem based on discrete IMU poses to calibrate the IMU-camera spatial-temporal and IMU intrinsic parameters. The above mentioned works are all offline methods and need calibration targets. In addition, they do not support full-parameter joint optimization of camera intrinsics with other calibration parameters.

[26] reduced IMU-camera calibration optimization complexity by selecting the most informative trajectory segments for calibration. The selection is based on the information matrix of the measurements from the trajectory segments. Although this work does not need a calibration board, the temporal calibration between IMU and camera is not included.

Many recent VINS algorithms perform online IMU-camera joint calibration. [14] and [47] is able to perform online IMU-camera extrinsic and time offset calibration with natural scene to improve the system robustness and accuracy. [6] proposed to use linear pose interpolation to model RS effects and calibrate readout time. [48] proposed a multi-camera aided VINS with online IMU-camera spatial-temporal and camera intrinsic calibration. [11] further proposed a generalized polynomial based pose interpolation for readout time calibration of RS cameras. However, the IMU intrinsics were not considered in the above systems.

The closest works to ours are by Li et al. [17] and Huai et al. [49] which included IMU-camera extrinsics, time offset, rolling-shutter readout time, camera and IMU intrinsics into the state vector of VINS. The former is built with MSCKF [8] based visual-inertial odometry while the latter is based on a key-frame sliding-window filter VINS. Both systems calibrated all these parameters. However, no system observability was present in [17] and degenerate motion analysis was still

missing from [49]. Instead, system observability and degenerate motion analysis are the focus of our work along with more extensive multi-run statistical validations of the calibration results. In addition, we also evaluate different IMU model variants which have appeared in literature.

## 2.4 Observability and Degeneracy

Observability analysis plays an important role in state estimation [50, 51, 52], especially when the system incorporates calibration parameters [53, 25, 18, 19]. We wish to identify whether these calibration parameters can be calibrated with visual and inertial measurements, and also identify degenerate motions, which might cause calibration to fail. In addition, observability properties can be leveraged for consistent estimator design [54, 52, 55]. [20] performed the observability analysis for VINS with IMU-camera spatial calibration and showed that the spatial calibration is observable given fully excited motion of the IMU. They also found that that one-axis rotation is degenerate for the calibration of IMU-camera translation. [21] studied the IMU-camera self-calibration and performed nonlinear observability analysis using Lie derivative to show that the rigid transformation between IMU-camera is observable given random motions. [22] simplified the proof and analytically showed that the spatial calibration between the IMU and RGBD camera is observable. [25] analyzed the identifiability for IMU-camera temporal calibration given the measurements of a single IMU and a monocular camera and identified a degenerate motion that can cause the IMU-camera time offset to become unobservable. [42] studied the observability of stereo VINS with IMU intrinsics also based on Lie derivative to build the observability matrix and showed that the IMU intrinsics (including scale correction and axis misalignment for gyroscope and accelerometer, respectively) is observable given fully exited motions. [23] built the observability matrix for VINS using linearized system model and showed that the camera intrinsics (only including focal length and principal points in their work) is observable. However, none of the above mentioned works ever performed and verified the observability analysis with full-parameter calibration for VINS.

In our previous work [18], we built the observability matrix for VINS using the linearized system with IMU-camera spatial-temporal calibration and showed that given fully excited motions all these calibration parameters are observable. We have also, for first time, identified four degenerate motions that can cause these calibration to become unobservable. In our recent work [19], we performed observability analysis for monocular VINS with IMU intrinsic calibration (including scale and axis-misalignment for gyroscope and accelerometer, the rotation from gyroscope or accelerometer to IMU frame), and identified the degenerate motions for the IMU intrinsics. Building upon these prior works, in this work, we perform full-parameter calibration – including IMU intrinsics with g-sensitivity, camera intrinsics and the IMU-camera spatial-temporal calibration with RS readout time – for VINS with a single IMU and a monocular RS camera. Comprehensive observability analysis and degenerate motion identification are performed for these calibration parameters. Both simulations and real world experiments are also leveraged to verify our analysis.

## 3 Estimation Preliminaries

The state $\mathbf{x}$ is propagated forward from timestep $k-1$ to $k$ using incoming system control inputs $\mathbf{u}_{k-1}$ based on the following generic nonlinear function:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \tag{1}$$

where $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ denotes the white Gaussian noise of the control input. The state estimate at $k$ can be predicted from the state estimate at $k-1$ with the nonlinear system:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}, \mathbf{0}) \tag{2}$$

where the subscript $i|j$ denotes the estimate at time $i$ given the measurements up to time $j$; $\hat{\mathbf{x}}$ denotes the estimated value for state $\mathbf{x}$ and $\tilde{\mathbf{x}} = \mathbf{x} \boxminus \hat{\mathbf{x}}$ represents the error states, where "$\boxminus$" can be defined within a manifold [56]. The inverse operation of "$\boxminus$" can be defined as $\mathbf{x} = \hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}}$, accordingly. The state covariance can be defined on the error states as $\tilde{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$.

After linearizing the nonlinear function [see Eq. (1)] at current state estimate $\hat{\mathbf{x}}_{k-1|k-1}$, the propagated state covariance $\mathbf{P}_{k|k-1}$ for state estimate $\hat{\mathbf{x}}_{k|k-1}$ can be computed as:

$$\tilde{\mathbf{x}}_{k|k-1} \simeq \boldsymbol{\Phi}_{k-1|k-1}\tilde{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1} \tag{3}$$

$$\mathbf{P}_{k|k-1} = \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1|k-1}\boldsymbol{\Phi}_{k-1}^{\top} + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^{\top} \tag{4}$$

where $\boldsymbol{\Phi}_{k-1}$ and $\mathbf{G}_{k-1}$ are the state transition matrix and noise Jacobians, respectively. A nonlinear measurement function can be described as:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k \tag{5}$$

where $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is white Gaussian noise. For the EKF update, we need to first linearize the above equation at the current state estimate $\hat{\mathbf{x}}_{k|k-1}$ as:

$$\mathbf{z}_k = \mathbf{h}(\hat{\mathbf{x}}_{k|k-1} \boxplus \tilde{\mathbf{x}}_{k|k-1}) + \mathbf{n}_k \simeq \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{H}_k\tilde{\mathbf{x}}_{k|k-1} + \mathbf{n}_k \tag{6}$$

$$\Rightarrow \tilde{\mathbf{z}}_k \triangleq \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) \simeq \mathbf{H}_k\tilde{\mathbf{x}}_{k|k-1} + \mathbf{n}_k \tag{7}$$

where $\mathbf{H}_k$ is the measurement Jacobian and $\tilde{\mathbf{z}}_k$ is the measurement residual. With these, we can now perform an EKF update to refine state estimates and covariance at time step $k$ [57]:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} \boxplus \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})) \tag{8}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} \tag{9}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^{\top}(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\top} + \mathbf{R}_k)^{-1} \tag{10}$$

## 4  Sensing Models

### 4.1  IMU Intrinsic Model

We define an IMU as containing two separate frames of reference (see Fig. 1): gyroscope frame $\{w\}$, accelerometer frame $\{a\}$. The base "inertial" frame $\{I\}$ should be determined to coincide with either $\{w\}$ or $\{a\}$. Different from the model in [26], we define the raw angular velocity reading ${}^w\boldsymbol{\omega}_m$ from the gyroscope and linear acceleration readings ${}^a\mathbf{a}_m$ from the accelerometer as:

$$^w\boldsymbol{\omega}_m = \mathbf{T}_w{}_I^w\mathbf{R}^I\boldsymbol{\omega} + \mathbf{T}_g{}^I\mathbf{a} + \mathbf{b}_g + \mathbf{n}_g \tag{11}$$

$$^a\mathbf{a}_m = \mathbf{T}_a{}_I^a\mathbf{R}^I\mathbf{a} + \mathbf{b}_a + \mathbf{n}_a \tag{12}$$

where $\mathbf{T}_w$ and $\mathbf{T}_a$ are invertible $3 \times 3$ matrices which represent the scale imperfection and axis misalignment for $\{w\}$ and $\{a\}$, respectively. ${}_I^w\mathbf{R}$ and ${}_I^a\mathbf{R}$ denote the rotation from the gyroscope frame and acceleration frame to base "inertial" frame $\{I\}$, respectively. Note that, if we choose $\{I\}$ coincides with $\{w\}$, then ${}_I^w\mathbf{R} = \mathbf{I}_3$. Otherwise, ${}_I^a\mathbf{R} = \mathbf{I}_3$. $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyroscope and
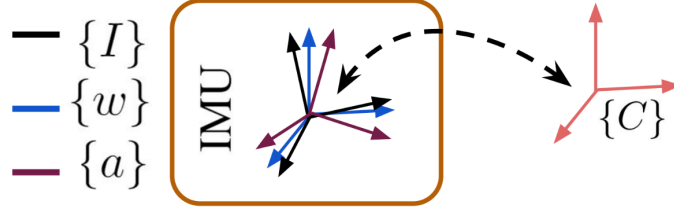
Figure 1: An IMU sensor composed of accelerometer and gyroscope. The base "inertial" frame can be determined to coincide with either accelerometer frame $\{a\}$ or gyroscope frame $\{w\}$. There is a rigid 6D transformation between camera frame $\{C\}$ and inertial frame $\{I\}$.

accelerometer biases, which are modeled as random walks; $\mathbf{n}_g$ and $\mathbf{n}_a$ are the zero-mean Gaussian noises contaminating the measurements. $\mathbf{T}_g$ denotes the g-sensitivity, which represents the effects of acceleration to the gyroscope readings. Similar to the works by [17] and [26], we do not take into account the translation between the gyroscope and accelerometer, since it is negligible for most IMUs. We can write the true (or corrected) angular velocity $^I\boldsymbol{\omega}$ and linear acceleration $^I\mathbf{a}$ as:

$$^I\boldsymbol{\omega} = {}^I_w\mathbf{R}\mathbf{D}_w \left( {}^w\boldsymbol{\omega}_m - \mathbf{T}_g {}^I\mathbf{a} - \mathbf{b}_g - \mathbf{n}_g \right) \tag{13}$$

$$^I\mathbf{a} = {}^I_a\mathbf{R}\mathbf{D}_a \left( {}^a\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a \right) \tag{14}$$

where $\mathbf{D}_w = \mathbf{T}_w^{-1}$ and $\mathbf{D}_a = \mathbf{T}_a^{-1}$. In practice we calibrate $\mathbf{D}_a$, $\mathbf{D}_w$, $^I_a\mathbf{R}$ (or $^I_w\mathbf{R}$) and $\mathbf{T}_g$ to prevent the need to have the unnecessary matrix inversions in the above measurement equations. We only calibrate either $^I_w\mathbf{R}$ or $^I_a\mathbf{R}$ in Eq. (13) and Eq. (14) since the base "inertial" frame coincides with one of sensor frames. If both $^I_w\mathbf{R}$ and $^I_a\mathbf{R}$ were calibrated, it would make the rotation between the IMU and camera unobservable due to over parameterization which will be validated in Section 9.5.

### 4.1.1 IMU intrinsic model variants.

Given the above general model [see Eq. (13) and Eq. (14)], different choices of these intrinsic parameters can be made [17, 46, 58, 26, 41, 42]. In the following, we present a range of commonly-used IMU intrinsic model variants, and will later compare against each other within an online filter-based VINS. Specifically, each model is defined as follows:

- *imu1*: includes the rotation $^I_w\mathbf{R}$, 6 parameters for $\mathbf{D}_w$ (and thus denoted by $\mathbf{D}_{w6}$) and 6 parameters for $\mathbf{D}_a$ (and thus denoted by $\mathbf{D}_{a6}$), as they assume the upper-triangular structure:

$$\mathbf{D}_{*6} = \begin{bmatrix} d_{*1} & d_{*2} & d_{*4} \\ 0 & d_{*3} & d_{*5} \\ 0 & 0 & d_{*6} \end{bmatrix} \tag{15}$$

- *imu2*: includes the rotation $^I_a\mathbf{R}$ instead, $\mathbf{D}_{a6}$ and $\mathbf{D}_{w6}$, which is the model used by [26].

- *imu3*: combines *imu1*'s $\mathbf{D}_{w6}$ and $^I_w\mathbf{R}$ into a general $3 \times 3$ matrix containing 9 parameters in total. Thus, in this variant we estimate the upper-triangle $\mathbf{D}_{a6}$ and a full matrix $\mathbf{D}_{w9}$ as:

$$\mathbf{D}_{*9} = \begin{bmatrix} d_{*1} & d_{*4} & d_{*7} \\ d_{*2} & d_{*5} & d_{*8} \\ d_{*3} & d_{*6} & d_{*9} \end{bmatrix} \tag{16}$$

- *imu4*: is an extension of *imu2* with a combination of the $\mathbf{D}_{a6}$ and $^I_a\mathbf{R}$. Similarly, in this variant we estimate the upper-triangle $\mathbf{D}_{w6}$ and a full matrix $\mathbf{D}_{a9}$.

Table 1: IMU model variants and their estimated parameters.

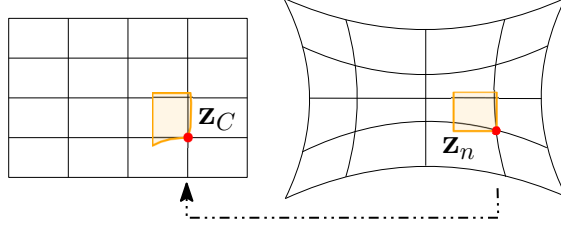| Model | Dim. | $\mathbf{D}_w$ | $\mathbf{D}_a$ | $^I_w\mathbf{R}$ | $^I_a\mathbf{R}$ | $\mathbf{T}_g$ |
|---|---|---|---|---|---|---|
| *imu0* | 0 | - | - | - | - | - |
| *imu1* | 15 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a6}$ | $^I_w\mathbf{R}$ | - | - |
| *imu2* | 15 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a6}$ | - | $^I_a\mathbf{R}$ | - |
| *imu3* | 15 | $\mathbf{D}_{w9}$ | $\mathbf{D}_{a6}$ | - | - | - |
| *imu4* | 15 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a9}$ | - | - | - |
| *imu5* | 18 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a6}$ | $^I_w\mathbf{R}$ | $^I_a\mathbf{R}$ | - |
| *imu6* | 24 | $\mathbf{D}'_{w6}$ | $\mathbf{D}'_{a6}$ | $^I_w\mathbf{R}$ | - | $\mathbf{T}_{g9}$ |
| *imu11* | 21 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a6}$ | $^I_w\mathbf{R}$ | - | $\mathbf{T}_{g6}$ |
| *imu12* | 21 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a6}$ | - | $^I_a\mathbf{R}$ | $\mathbf{T}_{g6}$ |
| *imu13* | 21 | $\mathbf{D}_{w9}$ | $\mathbf{D}_{a6}$ | - | - | $\mathbf{T}_{g6}$ |
| *imu14* | 21 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a9}$ | - | - | $\mathbf{T}_{g6}$ |
| *imu21* | 24 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a6}$ | $^I_w\mathbf{R}$ | - | $\mathbf{T}_{g9}$ |
| *imu22* | 24 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a6}$ | - | $^I_a\mathbf{R}$ | $\mathbf{T}_{g9}$ |
| *imu23* | 24 | $\mathbf{D}_{w9}$ | $\mathbf{D}_{a6}$ | - | - | $\mathbf{T}_{g9}$ |
| *imu24* | 24 | $\mathbf{D}_{w6}$ | $\mathbf{D}_{a9}$ | - | - | $\mathbf{T}_{g9}$ |
| *imu31* | 9 | - | $\mathbf{D}_{a9}$ | - | - | - |
| *imu32* | 9 | $\mathbf{D}_{w9}$ | - | - | - | - |
| *imu33* | 6 | - | - | - | - | $\mathbf{T}_{g6}$ |
| *imu34* | 9 | - | - | - | - | $\mathbf{T}_{g9}$ |

Figure 2: Distorting from normalized to a raw image pixel.

- $imu1A$ $(A = 1, \cdots, 4)$: combines $imuA$ with a 6-parameter g-sensitivity $\mathbf{T}_{g6}$ as:

$$\mathbf{T}_{g6} = \begin{bmatrix} t_{g1} & t_{g2} & t_{g4} \\ 0 & t_{g3} & t_{g5} \\ 0 & 0 & t_{g6} \end{bmatrix} \tag{17}$$

- $imu2A$ $(A = 1, \cdots, 4)$: combines $imuA$ a the 9-parameter g-sensitivity $\mathbf{T}_{g9}$ as:

$$\mathbf{T}_{g9} = \begin{bmatrix} t_{g1} & t_{g4} & t_{g7} \\ t_{g2} & t_{g5} & t_{g8} \\ t_{g3} & t_{g6} & t_{g9} \end{bmatrix} \tag{18}$$

- $imu5$: contains $\mathbf{D}_{w6}$, $\mathbf{D}_{a6}$, $_w^I\mathbf{R}$ and $_a^I\mathbf{R}$. This is a redundant over-parameterized model which will be used to verify that $_w^I\mathbf{R}$ and $_a^I\mathbf{R}$ should not be calibrated simultaneously.

- $imu6$: contains $\mathbf{D}'_{w6}$, $\mathbf{D}'_{a6}$, $_w^I\mathbf{R}$ and $\mathbf{T}_{g9}$. This is equivalent to the *scale-misalignment* IMU intrinsic model [44] used in the calibration toolbox [38]. $\mathbf{D}'_{*6}$ assumes the lower triangular structure:

$$\mathbf{D}'_{*6} = \begin{bmatrix} d_{*1} & 0 & 0 \\ d_{*2} & d_{*4} & 0 \\ d_{*3} & d_{*5} & d_{*6} \end{bmatrix} \tag{19}$$

- $imu3A$ $(A = 1, \cdots, 4)$: models a subset of the parameters of the general model while assuming the others known; that is, only calibrates $\mathbf{D}_{a9}$ in $imu31$, $\mathbf{D}_{w9}$ in $imu32$, $\mathbf{T}_{g6}$ in $imu33$, and $\mathbf{T}_{g9}$ in $imu34$.

These different models are summarized in Table 1. Note that for presentation clarity, $imu22$ is used in the ensuing system derivations and analysis.

## 4.2 Camera Model

Consider a 3D point feature, $^G\mathbf{p}_f$, that is captured by a camera with visual measurement function written as:

$$\mathbf{z}_C = \begin{bmatrix} u \\ v \end{bmatrix} + \mathbf{n}_C \tag{20}$$

where $\mathbf{n}_C$ denotes the measurement noise; $u$ and $v$ are the distorted image pixel coordinates:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{h}_d \left( \mathbf{z}_n, \mathbf{x}_{Cin} \right) \tag{21}$$

where $\mathbf{z}_n = [u_n \ v_n]^\top$ represents the normalized image pixel and $\mathbf{h}_d(\cdot)$ maps the normalized image pixel onto the image plane based on the camera intrinsic parameters $\mathbf{x}_{Cin}$ and camera model. Specifically, a pinhole model with radial-tangential (*radtan*) or equivalent-distant (*equidist*) distortion can be used, and the *radtan* model is used in the following derivations and analysis [see [38, 59]]. Fig. 2 visualizes this image distortion operation. $\mathbf{x}_{Cin}$ and $\mathbf{h}_d(\cdot)$ are given by:

$$\mathbf{x}_{Cin} = \begin{bmatrix} f_u & f_v & c_u & c_v & k_1 & k_2 & p_1 & p_2 \end{bmatrix}^\top \tag{22}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_u & 0 \\ 0 & f_v \end{bmatrix} \begin{bmatrix} u_d \\ v_d \end{bmatrix} + \begin{bmatrix} c_u \\ c_v \end{bmatrix} \tag{23}$$

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = \begin{bmatrix} du_n + 2p_1 u_n v_n + p_2(r^2 + 2u_n^2) \\ dv_n + p_1(r^2 + 2v_n^2) + 2p_2 u_n v_n \end{bmatrix} \tag{24}$$

where $r^2 = u_n^2 + v_n^2$; $d = 1 + k_1 r^2 + k_2 r^4$; $f_u$ and $f_v$ are the camera focal length; $c_u$ and $c_v$ denotes the image principal point; $k_1$ and $k_2$ represent the radial distortion coefficients while $p_1$ and $p_2$ are tangential distortion coefficients.

Normalized image pixel $u$ and $v$ can be acquired by projecting 3D feature $^C\mathbf{p}_f = [^C x_f \ ^C y_f \ ^C z_f]^\top$ in camera frame into 2D plane as:

$$\mathbf{z}_n = \mathbf{h}_p\left(^C\mathbf{p}_f\right) \triangleq \frac{1}{^C z_f} \begin{bmatrix} ^C x_f \\ ^C y_f \end{bmatrix} \tag{25}$$

$$^C\mathbf{p}_f = \mathbf{h}_t(^I_G\mathbf{R}, {}^G\mathbf{p}_I, {}^C_I\mathbf{R}, {}^C\mathbf{p}_I, {}^G\mathbf{p}_f) \triangleq {}^C_I\mathbf{R}^I_G\mathbf{R}\left(^G\mathbf{p}_f - {}^G\mathbf{p}_I\right) + {}^C\mathbf{p}_I \tag{26}$$

where $\{^C_I\mathbf{R}, {}^C\mathbf{p}_I\}$ represents the rigid transformation between the IMU and camera frames.

### 4.2.1 Temporal calibration.

Two common variants of camera sensing modes are global shutter (GS) and rolling shutter (RS). GS cameras expose all pixels at a single time instance, while, typically lower-cost, RS cameras expose each row sequentially. As shown by [6], it may lead to large estimation errors if this RS effect is not taken into account when using RS cameras. Additionally, the camera and IMU measurement timestamps can be incorrect due to processing or communication delays, or different clock references. To address this, we model both the time offset and camera readout time to ensure all measurements are processed in a common clock frame of reference and at the correct corresponding poses. Specially, $t_d$ denotes the time offset between IMU and camera timeline and $t_r$ denotes the RS readout time for the whole image. If $t$ denotes the time when the pixel is captured, the RS measurement function for a pixel captured in the $m$-th row (out of total $M$ rows) is given by:

$$^C\mathbf{p}_f = \mathbf{h}_t(^{I(t)}_G\mathbf{R}, {}^G\mathbf{p}_{I(t)}, {}^C_I\mathbf{R}, {}^C\mathbf{p}_I, {}^G\mathbf{p}_f) \triangleq {}^C_I\mathbf{R}^{I(t)}_G\mathbf{R}\left(^G\mathbf{p}_f - {}^G\mathbf{p}_{I(t)}\right) + {}^C\mathbf{p}_I \tag{27}$$

$$t_I = t_C + t_d \tag{28}$$

$$t = t_I + \frac{m}{M} t_r \tag{29}$$

where $t_I$ is the IMU state time corresponding to the captured image time $t_C$ when the first row of the image is collected. If the readout time $t_r = 0$, then the camera is actually a GS camera and all rows are a function of the same pose. As usual, $\{^G_{I(t)}\mathbf{R}, {}^G\mathbf{p}_{I(t)}\}$ is the IMU global pose corresponding to the camera measurement time $t$.

# 5 System Models

The state vector $\mathbf{x}$ of the visual-inertial system under consideration includes the inertial navigation state $\mathbf{x}_I$, IMU intrinsic parameter $\mathbf{x}_{in}$, IMU-camera spatial-temporal extrinsic calibration $\mathbf{x}_{IC}$, camera intrinsic calibration $\mathbf{x}_{Cin}$ and feature positions $\mathbf{x}_f$, which is given by:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^\top & \mathbf{x}_{IC}^\top & \mathbf{x}_{Cin}^\top & \mathbf{x}_f^\top \end{bmatrix}^\top \tag{30}$$

$$\mathbf{x}_I = \begin{bmatrix} \mathbf{x}_n^\top & | & \mathbf{x}_b^\top & | & \mathbf{x}_{in}^\top \end{bmatrix}^\top = \begin{bmatrix} {}_G^I\bar{q}^\top & {}^G\mathbf{p}_I^\top & {}^G\mathbf{v}_I^\top & | & \mathbf{b}_g^\top & \mathbf{b}_a^\top & | & \mathbf{x}_{in}^\top \end{bmatrix}^\top$$

$$\mathbf{x}_{in} = \begin{bmatrix} \mathbf{x}_{Dw}^\top & \mathbf{x}_{Da}^\top & {}_a^I\bar{q}^\top & \mathbf{x}_{Tg}^\top \end{bmatrix}^\top \tag{31}$$

$$\mathbf{x}_{IC} = \begin{bmatrix} {}_I^C\bar{q}^\top & {}^C\mathbf{p}_I^\top & t_d & t_r \end{bmatrix}^\top \tag{32}$$

where ${}_G^I\bar{q}$ denotes quaternion with JPL convention [60] and corresponds to the rotation matrix ${}_G^I\mathbf{R}$, which represents the rotation from $\{G\}$ to $\{I\}$. ${}^G\mathbf{p}_I$ and ${}^G\mathbf{v}_I$ denote the IMU position and velocity in $\{G\}$. $\mathbf{x}_n$ denotes the IMU navigation states containing the ${}_G^I\bar{q}$, ${}^G\mathbf{p}_I$ and ${}^G\mathbf{v}_I$. $\mathbf{x}_b$ denotes the IMU bias states containing $\mathbf{b}_g$ and $\mathbf{b}_a$. ${}_I^C\bar{q}$ and ${}^C\mathbf{p}_I$ denotes the rigid transformation between $\{C\}$ and $\{I\}$. $t_d$ and $t_r$ represent the IMU-camera time offset and camera readout time. IMU intrinsics, $\mathbf{x}_{in}$, contains $\mathbf{x}_{Dw}$, $\mathbf{x}_{Da}$, $\mathbf{x}_{Tg}$ and ${}_a^I\bar{q}$, where $\mathbf{x}_{Dw}$, $\mathbf{x}_{Da}$ and $\mathbf{x}_{Tg}$ are non-zero elements stored column-wise in $\mathbf{D}_w$, $\mathbf{D}_a$ and $\mathbf{T}_g$. Specifically, they are defined as:

$$\mathbf{x}_{D*} = \begin{bmatrix} d_{*1} & d_{*2} & d_{*3} & d_{*4} & d_{*5} & d_{*6} \end{bmatrix}^\top \tag{33}$$

$$\mathbf{x}_{Tg} = \begin{bmatrix} t_{g1} & t_{g2} & t_{g3} & t_{g4} & t_{g5} & t_{g6} & t_{g7} & t_{g8} & t_{g9} \end{bmatrix}^\top \tag{34}$$

It is important to note we use the quaternion left multiplicative error defined by $\bar{q} \approx [\frac{1}{2}\delta\boldsymbol{\theta}^\top \ 1]^\top \otimes \hat{\bar{q}}$, where $\otimes$ denotes quaternion multiplication and error state is equivalent to the $SO(3)$ error (i.e. ${}_G^I\mathbf{R} \approx (\mathbf{I}_3 - \lfloor\delta\boldsymbol{\theta}\rfloor){}_G^I\hat{\mathbf{R}})$ [60]. The dynamics of the inertial navigation state $\mathbf{x}_I$ is given by [61]:

$$
{}_G^I\dot{\bar{q}} = \frac{1}{2}\boldsymbol{\Omega}({}^I\boldsymbol{\omega}){}_G^I\bar{q} \ , \quad {}^G\dot{\mathbf{p}}_I = {}^G\mathbf{v}_I \tag{35}
$$

$$
{}^G\dot{\mathbf{v}}_I = {}_G^I\mathbf{R}^{\top I}\mathbf{a} - {}^G\mathbf{g} \ , \quad \dot{\mathbf{b}}_g = \mathbf{n}_{wg} \ , \quad \dot{\mathbf{b}}_a = \mathbf{n}_{wa}
$$

where $\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}$, $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$ are zero-mean white Gaussian noises driving $\mathbf{b}_g$ and $\mathbf{b}_a$, respectively, and the known global gravity assumes ${}^G\mathbf{g} = [0 \ 0 \ 9.81]^\top$, while the rest of the states have zero dynamics.

## 5.1 Analytic Inertial Integration

In the following, we present the analytic IMU integration and corresponding error state transition matrix, which was originally presented in our prior work [62] and later extended to include IMU intrinsics [19]. Specifically, we compute the integration of IMU dynamics based on Eq. (35) from

time step $t_k$ to $t_{k+1}$:

$$
{}^{I_{k+1}}_{G}\mathbf{R} = \Delta\mathbf{R}_k^\top {}^{I_k}_{G}\mathbf{R} \tag{36}
$$

$$
{}^{G}\mathbf{p}_{I_{k+1}} = {}^{G}\mathbf{p}_{I_k} + {}^{G}\mathbf{v}_{I_k}\delta t_k + {}^{I_k}_{G}\mathbf{R}^\top \Delta\mathbf{p}_k - \frac{1}{2}{}^{G}\mathbf{g}\delta t_k^2 \tag{37}
$$

$$
{}^{G}\hat{\mathbf{v}}_{I_{k+1}} = {}^{G}\hat{\mathbf{v}}_{I_k} + {}^{I_k}_{G}\mathbf{R}^\top \Delta\mathbf{v}_k - {}^{G}\mathbf{g}\delta t_k \tag{38}
$$

$$
\mathbf{b}_{g_{k+1}} = \mathbf{b}_{g_k} + \int_{t_k}^{t_{k+1}} \mathbf{n}_{wg} d\tau \tag{39}
$$

$$
\mathbf{b}_{a_{k+1}} = \mathbf{b}_{a_k} + \int_{t_k}^{t_{k+1}} \mathbf{n}_{wa} d\tau \tag{40}
$$

where $\delta t_k = t_{k+1} - t_k$, and the three IMU integration quantities are given by:

$$
\Delta\mathbf{R}_k \triangleq {}^{I_k}_{I_{k+1}}\mathbf{R} = \exp\left(\int_{t_k}^{t_{k+1}} {}^{I_\tau}\boldsymbol{\omega} d\tau\right) \tag{41}
$$

$$
\Delta\mathbf{p}_k \triangleq \int_{t_k}^{t_{k+1}} \int_{t_k}^{s} {}^{I_k}_{I_\tau}\mathbf{R}^{I_\tau}\mathbf{a} d\tau ds \tag{42}
$$

$$
\Delta\mathbf{v}_k \triangleq \int_{t_k}^{t_{k+1}} {}^{I_k}_{I_\tau}\mathbf{R}^{I_\tau}\mathbf{a} d\tau \tag{43}
$$

where $\exp(\cdot)$ is the $SO(3)$ matrix exponential [63]. The current best estimate of the true angular velocity, ${}^{I_k}\boldsymbol{\omega}$, and linear acceleration, ${}^{I_k}\mathbf{a}$, within this time interval $[t_k, t_{k+1}]$ is the expectation of Eq. (13) and Eq. (14). Assuming constant ${}^{I_k}\hat{\boldsymbol{\omega}}$ and ${}^{I_k}\hat{\mathbf{a}}$ within the time interval, we approximate $\Delta\hat{\mathbf{R}}_k$, $\Delta\hat{\mathbf{p}}_k$ and $\Delta\hat{\mathbf{v}}_k$ as:

$$
\Delta\hat{\mathbf{R}}_k \simeq \exp\left({}^{I_k}\hat{\boldsymbol{\omega}}\delta t_k\right) \tag{44}
$$

$$
\Delta\hat{\mathbf{p}}_k \simeq \left(\int_{t_k}^{t_{k+1}} \int_{t_k}^{s} {}^{I_k}_{I_\tau}\hat{\mathbf{R}} d\tau ds\right) {}^{I_k}\hat{\mathbf{a}} \triangleq \boldsymbol{\Xi}_2 {}^{I_k}\hat{\mathbf{a}} \tag{45}
$$

$$
\boldsymbol{\Xi}_2 \triangleq \int_{t_k}^{t_{k+1}} \int_{t_k}^{s} \exp\left({}^{I_k}\hat{\boldsymbol{\omega}}\delta\tau\right) d\tau ds \tag{46}
$$

$$
\Delta\hat{\mathbf{v}}_k \simeq \left(\int_{t_k}^{t_{k+1}} {}^{I_k}_{I_\tau}\hat{\mathbf{R}} d\tau\right) {}^{I_k}\hat{\mathbf{a}} \triangleq \boldsymbol{\Xi}_1 {}^{I_k}\hat{\mathbf{a}} \tag{47}
$$

$$
\boldsymbol{\Xi}_1 \triangleq \int_{t_k}^{t_{k+1}} \exp\left({}^{I_k}\hat{\boldsymbol{\omega}}\delta\tau\right) d\tau \tag{48}
$$

where $\delta\tau = t_\tau - t_k$, $\boldsymbol{\Xi}_1$ and $\boldsymbol{\Xi}_2$ are defined as integration components which can be evaluated either analytically [19] or numerically using the Runge–Kutta fourth-order (RK4) method. ${}^{I_k}\hat{\boldsymbol{\omega}}$ and ${}^{I_k}\hat{\mathbf{a}}$ are computed as (note that we drop the timestamp $k$ for simplicity):

$$
{}^{I}\hat{\boldsymbol{\omega}} = {}^{I}_{w}\hat{\mathbf{R}}\hat{\mathbf{D}}_w{}^{w}\hat{\boldsymbol{\omega}} \tag{49}
$$

$$
{}^{w}\hat{\boldsymbol{\omega}} = {}^{w}\boldsymbol{\omega}_m - \hat{\mathbf{T}}_g{}^{I}\hat{\mathbf{a}} - \hat{\mathbf{b}}_g =: \begin{bmatrix} {}^{w}\hat{w}_1 & {}^{w}\hat{w}_2 & {}^{w}\hat{w}_3 \end{bmatrix}^\top \tag{50}
$$

$$
{}^{I}\hat{\mathbf{a}} = {}^{I}_{a}\hat{\mathbf{R}}\hat{\mathbf{D}}_a{}^{a}\hat{\mathbf{a}} \tag{51}
$$

$$
{}^{a}\hat{\mathbf{a}} = {}^{a}\mathbf{a}_m - \hat{\mathbf{b}}_a =: \begin{bmatrix} {}^{a}\hat{a}_1 & {}^{a}\hat{a}_2 & {}^{a}\hat{a}_3 \end{bmatrix}^\top \tag{52}
$$

where $_w^I\hat{\mathbf{R}} = \mathbf{I}_3$ for *imu22*. As $^{I_k}\hat{\boldsymbol{\omega}}$ and $^{I_k}\hat{\mathbf{a}}$ are assumed to be constant, the state estimate at $t_{k+1}$ is propagated as follows [see Eq. (36)-(40)]:

$$_G^{I_{k+1}}\hat{\mathbf{R}} \simeq \Delta\mathbf{R}_k^{\top} {}_G^{I_k}\hat{\mathbf{R}} \tag{53}$$

$$^G\hat{\mathbf{p}}_{I_{k+1}} \simeq {}^G\hat{\mathbf{p}}_{I_k} + {}^G\hat{\mathbf{v}}_{I_k}\delta t_k + {}_G^{I_k}\hat{\mathbf{R}}^{\top}\Delta\hat{\mathbf{p}}_k - \frac{1}{2}{}^G\mathbf{g}\delta t_k^2 \tag{54}$$

$$^G\hat{\mathbf{v}}_{I_{k+1}} \simeq {}^G\hat{\mathbf{v}}_{I_k} + {}_G^{I_k}\hat{\mathbf{R}}^{\top}\Delta\hat{\mathbf{v}}_k - {}^G\mathbf{g}\delta t_k \tag{55}$$

$$\hat{\mathbf{b}}_{g_{k+1}} = \hat{\mathbf{b}}_{g_k} \tag{56}$$

$$\hat{\mathbf{b}}_{a_{k+1}} = \hat{\mathbf{b}}_{a_k} \tag{57}$$

## 5.2 Linearized System Model

We first linearize the three IMU pre-integration components [see Eq. (41)-(43)]:

$$\Delta\mathbf{R}_k = \Delta\hat{\mathbf{R}}_k\Delta\tilde{\mathbf{R}}_k \triangleq \Delta\hat{\mathbf{R}}_k \exp\left(\mathbf{J}_r(\Delta\hat{\boldsymbol{\theta}}_k)^{I_k}\tilde{\boldsymbol{\omega}}\delta t_k\right) \tag{58}$$

$$\Delta\mathbf{p}_k = \Delta\hat{\mathbf{p}}_k + \Delta\tilde{\mathbf{p}}_k \triangleq \Delta\hat{\mathbf{p}}_k - \boldsymbol{\Xi}_4{}^{I_k}\tilde{\boldsymbol{\omega}} + \boldsymbol{\Xi}_2{}^{I_k}\tilde{\mathbf{a}} \tag{59}$$

$$\Delta\mathbf{v}_k = \Delta\hat{\mathbf{v}}_k + \Delta\tilde{\mathbf{v}}_k \triangleq \Delta\hat{\mathbf{v}}_k - \boldsymbol{\Xi}_3{}^{I_k}\tilde{\boldsymbol{\omega}} + \boldsymbol{\Xi}_1{}^{I_k}\tilde{\mathbf{a}} \tag{60}$$

where $\mathbf{J}_r(\Delta\hat{\boldsymbol{\theta}}_k) \triangleq \mathbf{J}_r\left(^{I_k}\hat{\boldsymbol{\omega}}\delta t_k\right)$ denotes the right Jacobian of $SO(3)$ [63]. The derivation and the definitions of $^{I_k}\tilde{\boldsymbol{\omega}}$ and $^{I_k}\tilde{\mathbf{a}}$ can be found in Appendix A. The integrated components $\boldsymbol{\Xi}_3$ and $\boldsymbol{\Xi}_4$ are defined as:

$$\boldsymbol{\Xi}_3 \triangleq \int_{t_k}^{t_{k+1}} {}_{I_\tau}^{I_k}\mathbf{R}\lfloor^{I_\tau}\mathbf{a}\rfloor\mathbf{J}_r\left(^{I_k}\boldsymbol{\omega}\delta\tau\right)\delta\tau d\tau \tag{61}$$

$$\boldsymbol{\Xi}_4 \triangleq \int_{t_k}^{t_{k+1}}\int_{t_k}^{s} {}_{I_\tau}^{I_k}\mathbf{R}\lfloor^{I_\tau}\mathbf{a}\rfloor\mathbf{J}_r\left(^{I_k}\boldsymbol{\omega}\delta\tau\right)\delta\tau d\tau ds \tag{62}$$

With the IMU preintegration, the linearized inertial navigation system of error state is given by:

$$\delta\boldsymbol{\theta}_{k+1} \simeq \Delta\hat{\mathbf{R}}_k^{\top}\delta\boldsymbol{\theta}_k + \mathbf{J}_r\left(\Delta\hat{\boldsymbol{\theta}}_k\right)\delta t_k{}^{I_k}\tilde{\boldsymbol{\omega}}$$

$$^G\tilde{\mathbf{p}}_{I_{k+1}} \simeq {}^G\tilde{\mathbf{p}}_{I_k} + {}^G\tilde{\mathbf{v}}_k\delta t_k - {}_G^{I_k}\hat{\mathbf{R}}^{\top}\lfloor\Delta\hat{\mathbf{p}}_k\rfloor\delta\boldsymbol{\theta}_k + {}_G^{I_k}\hat{\mathbf{R}}^{\top}\Delta\tilde{\mathbf{p}}_k$$

$$^G\tilde{\mathbf{v}}_{I_{k+1}} \simeq {}^G\tilde{\mathbf{v}}_{I_k} - {}_G^{I_k}\hat{\mathbf{R}}^{\top}\lfloor\Delta\hat{\mathbf{v}}_k\rfloor\delta\boldsymbol{\theta}_k + {}_G^{I_k}\hat{\mathbf{R}}^{\top}\Delta\tilde{\mathbf{v}}_k$$

As such, the full linearized error-state system for *imu22* is:

$$\tilde{\mathbf{x}}_{I_{k+1}} \simeq \boldsymbol{\Phi}_{I(k+1,k)}\tilde{\mathbf{x}}_{I_k} + \mathbf{G}_{Ik}\mathbf{n}_{dk} \tag{63}$$

$$\boldsymbol{\Phi}_{I(k+1,k)} = \begin{bmatrix} \boldsymbol{\Phi}_{nn} & \boldsymbol{\Phi}_{wa}\mathbf{H}_b & \boldsymbol{\Phi}_{wa}\mathbf{H}_{in} \\ \mathbf{0}_{6\times9} & \mathbf{I}_6 & \mathbf{0}_{6\times24} \\ \mathbf{0}_{24\times9} & \mathbf{0}_{24\times6} & \mathbf{I}_{24} \end{bmatrix} \tag{64}$$

$$\mathbf{G}_{Ik} = \begin{bmatrix} \boldsymbol{\Phi}_{wa}\mathbf{H}_n & \mathbf{0}_{9\times6} \\ \mathbf{0}_6 & \mathbf{I}_6\delta t_k \\ \mathbf{0}_{24\times6} & \mathbf{0}_{24\times6} \end{bmatrix} \tag{65}$$

where $\boldsymbol{\Phi}_{I(k+1,k)}$ and $\mathbf{G}_{Ik}$ are the state transition matrix and noise Jacobians for the inertial state $\mathbf{x}_I$ dynamics, $\mathbf{H}_b$, $\mathbf{H}_{in}$ and $\mathbf{H}_n$ are Jacobians related to bias, IMU intrinsics and noises, which can

be found in Appendix A, $\mathbf{n}_{dk} = [\mathbf{n}_{dg}^\top \ \mathbf{n}_{da}^\top \ \mathbf{n}_{dwg}^\top \ \mathbf{n}_{dwa}^\top]^\top$ is the discrete-time IMU noises, while $\boldsymbol{\Phi}_{nn}$ and $\boldsymbol{\Phi}_{wa}$ can be computed as:

$$\boldsymbol{\Phi}_{nn} = \begin{bmatrix} \Delta\hat{\mathbf{R}}_k^\top & \mathbf{0}_3 & \mathbf{0}_3 \\ -_G^{I_k}\hat{\mathbf{R}}^\top\lfloor\Delta\hat{\mathbf{p}}_k\rfloor & \mathbf{I}_3 & \mathbf{I}_3\delta t_k \\ -_G^{I_k}\hat{\mathbf{R}}^\top\lfloor\Delta\hat{\mathbf{v}}_k\rfloor & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{66}$$

$$\boldsymbol{\Phi}_{wa} = \begin{bmatrix} \mathbf{J}_r(\delta\boldsymbol{\theta}_k)\delta t_k & \mathbf{0}_3 \\ -_G^{I_k}\hat{\mathbf{R}}^\top\boldsymbol{\Xi}_4 & _G^{I_k}\hat{\mathbf{R}}^\top\boldsymbol{\Xi}_2 \\ -_G^{I_k}\hat{\mathbf{R}}^\top\boldsymbol{\Xi}_3 & _G^{I_k}\hat{\mathbf{R}}^\top\boldsymbol{\Xi}_1 \end{bmatrix} \tag{67}$$

Without loss of generality, we consider a single 3D feature $^G\mathbf{p}_f$ in the state vector $\mathbf{x}_f$. Since there is zero dynamics for $\mathbf{x}_{IC}$, $\mathbf{x}_{Cin}$ and $\mathbf{x}_f$, we can write the state transition matrix for the whole state vector $\mathbf{x}$ as [see Eq. (30)]:

$$\boldsymbol{\Phi}_{k+1,k} = \begin{bmatrix} \boldsymbol{\Phi}_{I(k+1,k)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_{IC} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Phi}_{Cin} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Phi}_f \end{bmatrix} \tag{68}$$

where $\boldsymbol{\Phi}_{IC} = \mathbf{I}_8$, $\boldsymbol{\Phi}_{Cin} = \mathbf{I}_8$, and $\boldsymbol{\Phi}_f = \mathbf{I}_3$.

## 5.3 Linearized Measurement Model

We first build the overall camera measurements function $\mathbf{h}_C(\cdot)$ by incorporating the distortion function $\mathbf{h}_d(\cdot)$ [see Eq. (21)], the projection function $\mathbf{h}_p(\cdot)$ [see Eq. (25)] and the transformation function $\mathbf{h}_t(\cdot)$ [see Eq. (27)]:

$$\mathbf{z}_C = \mathbf{h}_C(\mathbf{x}) + \mathbf{n}_C \tag{69}$$

$$= \mathbf{h}_d(\mathbf{z}_n, \mathbf{x}_{Cin}) + \mathbf{n}_C \tag{70}$$

$$= \mathbf{h}_d(\mathbf{h}_p(^{C_k}\mathbf{p}_f), \mathbf{x}_{Cin}) + \mathbf{n}_C \tag{71}$$

$$= \mathbf{h}_d(\mathbf{h}_p(\mathbf{h}_t(^{C(t)}_G\mathbf{R}, {}^G\mathbf{p}_{C(t)}, {}^G\mathbf{p}_f)), \mathbf{x}_{Cin}) + \mathbf{n}_C \tag{72}$$

We need to linearize the overall visual model for the update, which is given by:

$$\tilde{\mathbf{z}}_C \simeq \mathbf{H}_C\tilde{\mathbf{x}} + \mathbf{n}_C \tag{73}$$

where $\tilde{\mathbf{z}}_C \triangleq \mathbf{z}_C - \mathbf{h}_C(\hat{\mathbf{x}})$ and $\mathbf{H}_C \triangleq \frac{\partial\tilde{\mathbf{z}}_C}{\partial\tilde{\mathbf{x}}}$. Using the chainrule we get the following Jacobian matrix:

$$\mathbf{H}_C = \begin{bmatrix} \frac{\partial\tilde{\mathbf{z}}_C}{\partial\tilde{\mathbf{x}}_I} & \frac{\partial\tilde{\mathbf{z}}_C}{\partial\tilde{\mathbf{x}}_{IC}} & \frac{\partial\tilde{\mathbf{z}}_C}{\partial\tilde{\mathbf{x}}_{Cin}} & \frac{\partial\tilde{\mathbf{z}}_C}{\partial\tilde{\mathbf{x}}_f} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{\mathbf{p}_f}\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{\mathbf{x}}_I} & \mathbf{H}_{\mathbf{p}_f}\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{\mathbf{x}}_{IC}} & \frac{\partial\tilde{\mathbf{z}}_C}{\partial\tilde{\mathbf{x}}_{Cin}} & \mathbf{H}_{\mathbf{p}_f}\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{\mathbf{x}}_f} \end{bmatrix} \tag{74}$$

where $\mathbf{H}_{\mathbf{p}_f} = \frac{\partial\tilde{\mathbf{z}}_C}{\partial\tilde{\mathbf{z}}_n}\frac{\partial\tilde{\mathbf{z}}_n}{\partial^C\tilde{\mathbf{p}}_f}$. All the pertinent matrices $\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{\mathbf{x}}_I}$, $\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{\mathbf{x}}_{IC}}$, $\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{\mathbf{x}}_f}$ and $\mathbf{H}_{\mathbf{p}_f}$ can be computed as shown in Appendix B.

# 6 Observability Analysis

Observability analysis plays an important role in determining whether or not the states are estimable for given measurements and can also be leveraged to identify degenerate motions that can negatively

affect estimation performance [54, 64]. While the observability analysis of VINS has been well studied [65], the observability properties and degenerate motions of VINS with full self-calibration (in particular, IMU and camera intrinsic calibration) have not been sufficiently investigated. To this end, following [65], we construct the observability matrix as follows:

$$
\mathcal{O} = \begin{bmatrix} \mathcal{O}_1 \\ \mathcal{O}_2 \\ \vdots \\ \mathcal{O}_k \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{C1}\boldsymbol{\Phi}_{1,1} \\ \mathbf{H}_{C2}\boldsymbol{\Phi}_{2,1} \\ \vdots \\ \mathbf{H}_{Ck}\boldsymbol{\Phi}_{k,1} \end{bmatrix} \tag{75}
$$

We write the $k$-th row of $\mathcal{O}$ as:

$$
\mathcal{O}_k = \begin{bmatrix} \mathbf{M}_n & \mathbf{M}_b & \mathbf{M}_{in} & \mathbf{M}_{IC} & \mathbf{M}_{Cin} & \mathbf{M}_f \end{bmatrix} \tag{76}
$$

where $\mathbf{M}_n$, $\mathbf{M}_b$, $\mathbf{M}_{in}$, $\mathbf{M}_{IC}$, $\mathbf{M}_{Cin}$ and $\mathbf{M}_f$ represent the matrix block relating to the state [see Eq. (30)] with detailed derivations in Appendix C. We now look to find the unobservable subspace $\mathbf{N}$ such that $\mathcal{O}\mathbf{N} = \mathbf{0}$. The following can be found:

**Lemma 1.** *Given fully excited motions, monocular VINS system with online calibration of IMU intrinsics* $\mathbf{x}_{in}$, *camera intrinsics* $\mathbf{x}_{Cin}$ *and IMU-camera spatial-temporal parameters* $\mathbf{x}_{IC}$ *(including RS readout time) has 4 unobservable directions, which relate to the global yaw and global translation.*

$$
\mathbf{N} = \begin{bmatrix} {}^{I_1}_G\hat{\mathbf{R}}{}^G\mathbf{g} & \mathbf{0}_3 \\ -\lfloor {}^G\hat{\mathbf{p}}_{I_1} \rfloor {}^G\mathbf{g} & \mathbf{I}_3 \\ -\lfloor {}^G\hat{\mathbf{v}}_{I_1} \rfloor {}^G\mathbf{g} & \mathbf{0}_3 \\ \mathbf{0}_{46\times 1} & \mathbf{0}_{46\times 3} \\ -\lfloor {}^G\hat{\mathbf{p}}_f \rfloor {}^G\mathbf{g} & \mathbf{I}_3 \end{bmatrix} \tag{77}
$$

*Proof.* See Appendix D. □

We notice that the terms $\mathbf{M}_{in}$ and $\mathbf{M}_{IC}$ (shown in Appendix C) contain ${}^w\hat{\boldsymbol{\omega}}$, ${}^a\hat{\mathbf{a}}$, ${}^I\hat{\boldsymbol{\omega}}$ and ${}^G\hat{\mathbf{v}}_I$, corresponding to the sensor platform motion. This implies that, $\mathbf{M}_{in}$ and $\mathbf{M}_{IC}$, corresponding to IMU intrinsics $\mathbf{x}_{in}$ and IMU-camera spatial-temporal parameters $\mathbf{x}_{IC}$ (including RS effects), are motion-dependent and time-varying. Specifically, we have the following properties for the IMU intrinsics and camera spatial-temporal parameters:

**Proposition 1.** *For monocular VINS, the IMU intrinsic calibration and IMU-camera spatial-temporal calibration (including RS readout time) are sensitive to sensor motions. Given fully excited motions,* $\mathbf{x}_{in}$ *and* $\mathbf{x}_{IC}$ *are observable.*

Given this observation and numerical simulations of VINS based on a monocular camera and IMU, shown in Fig. 4, we can confirm that all these calibration parameters are observable and can be estimated given fully-excited motions. While we omit the derivations and simulation results here, the other IMU intrinsic model variants besides the *imu22* presented are also fully observable in the case of fully-excited motions.

Similarly, the camera intrinsics, $\mathbf{M}_{Cin}$, are mainly affected by the environmental structure (the $u$ and $v$ measurements of the 3D point features) with no motion terms are involved. Hence, we have:

**Proposition 2.** *For monocular VINS, the camera intrinsic calibration* $\mathbf{x}_{Cin}$ *is affected by the structure of the environment features and less sensitive to sensor motions.*

The camera intrinsic parameters are observable for most motion cases, even for under-actuated motions (i.e. planar motion), which can also be verified by our simulation results shown in Fig. 4-8. While we omit the results here, the *equidist* camera distortion model also satisfies the above proposition.

# 7 Degenerate Motion Identification

While the observability properties found in the preceding section hold with *general* motions, this may not always be the case in reality and thus the identification of *degenerate* motion profiles that cause extra unobservable directions in the state space, becomes important. Based on the observability analysis, we can further identify the degenerate motions corresponding to the state of the system, such as the inertial state and features, along with the calibration parameters introduced during online self-calibration. As the degenerate motion analysis of VINS has been studied in the prior work [51, 52, 66, 67], we here focus only on motions that cause the calibration parameters to become unobservable.

## 7.1 Inertial IMU Intrinsic Parameters

As mentioned in Section 6, $\mathbf{M}_{in}$ is heavily motion affected, and thus, the IMU intrinsics are extremely susceptible to be unobservable under certain motions. Because the bias terms and IMU intrinsics are tightly-coupled, by carefully inspecting the observability matrix $\mathcal{O}$, we find a selection of basic motion types which can cause the IMU intrinsics to become unobservable for *imu22*. Note that similar results are applicable to other IMU model variants.

### 7.1.1 Degenerate motions for $\mathbf{D}_w$

As the gyroscope related IMU intrinsics $\mathbf{D}_w$ are coupled with gyroscope bias $\mathbf{b}_g$ and the angular velocity readings $^w\boldsymbol{\omega}$ from the IMU, we have the following results:

**Lemma 2.** *If any component of $^w\boldsymbol{\omega}$ (including $^w\omega_1$, $^w\omega_2$, $^w\omega_3$) is constant, then $\mathbf{D}_w$ will become unobservable.*

*Proof.* If $^ww_1$ is constant, $d_{w1}$ will be unobservable with unobservable directions as:

$$\mathbf{N}_{w1} = \begin{bmatrix} \mathbf{0}_{1\times9} & (\hat{\mathbf{D}}_w^{-1}{}_w^I\hat{\mathbf{R}}^\top\mathbf{e}_1)^\top{}^ww_1 & \mathbf{0}_{1\times3} & 1 & \mathbf{0}_{1\times42} \end{bmatrix}^\top \tag{78}$$

If $^ww_2$ is constant, $d_{w2}$ and $d_{w3}$ will be unobservable with unobservable directions as:

$$\mathbf{N}_{w2} = \begin{bmatrix} \mathbf{0}_{1\times9} & (\hat{\mathbf{D}}_w^{-1}{}_w^I\hat{\mathbf{R}}^\top\mathbf{e}_1)^\top{}^ww_2 & \mathbf{0}_{1\times4} & 1 & \mathbf{0}_{1\times41} \\ \mathbf{0}_{1\times9} & (\hat{\mathbf{D}}_w^{-1}{}_w^I\hat{\mathbf{R}}^\top\mathbf{e}_2)^\top{}^ww_2 & \mathbf{0}_{1\times5} & 1 & \mathbf{0}_{1\times40} \end{bmatrix}^\top \tag{79}$$

If $^ww_3$ is constant, $d_{w4}$, $d_{w5}$ and $d_{w6}$ are unobservable with unobservable directions as:

$$\mathbf{N}_{w3} = \begin{bmatrix} \mathbf{0}_{1\times9} & (\hat{\mathbf{D}}_w^{-1}{}_w^I\hat{\mathbf{R}}^\top\mathbf{e}_1)^\top{}^ww_3 & \mathbf{0}_{1\times6} & 1 & \mathbf{0}_{1\times39} \\ \mathbf{0}_{1\times9} & (\hat{\mathbf{D}}_w^{-1}{}_w^I\hat{\mathbf{R}}^\top\mathbf{e}_2)^\top{}^ww_3 & \mathbf{0}_{1\times7} & 1 & \mathbf{0}_{1\times38} \\ \mathbf{0}_{1\times9} & (\hat{\mathbf{D}}_w^{-1}{}_w^I\hat{\mathbf{R}}^\top\mathbf{e}_3)^\top{}^ww_3 & \mathbf{0}_{1\times8} & 1 & \mathbf{0}_{1\times37} \end{bmatrix}^\top \tag{80}$$

See Appendix E for the verification of these unobservable directions. □

### 7.1.2 Degenerate motions for $\mathbf{D}_a$

Similarly, as $^a\mathbf{a}$ can affect the observability property for the accelerometer related IMU intrinsics $\mathbf{D}_a$, we have:

**Lemma 3.** *If any component of $^a\mathbf{a}$ (including $^a a_1$, $^a a_2$ and $^a a_3$) is constant, then $\mathbf{D}_a$ will become unobservable.*

*Proof.* If $^a a_1$ is constant, $d_{a1}$, pitch and yaw of $^I_a\mathbf{R}$ are unobservable with unobservable directions as:

$$
\mathbf{N}_{a1} = \begin{bmatrix}
\mathbf{0}_{12\times1} & \mathbf{0}_{12\times1} & \mathbf{0}_{12\times1} \\
\hat{\mathbf{D}}_a^{-1}\mathbf{e}_1{}^a a_1 & \hat{\mathbf{D}}_a^{-1}\mathbf{e}_2\hat{d}_{a1}{}^a a_1 & \hat{\mathbf{D}}_a^{-1}\mathbf{e}_3\hat{d}_{a1}\hat{d}_{a3}{}^a a_1 \\
\mathbf{0}_{6\times1} & \mathbf{0}_{6\times1} & \mathbf{0}_{6\times1} \\
1 & 0 & 0 \\
0 & \hat{d}_{a3} & 0 \\
0 & -\hat{d}_{a2} & 0 \\
0 & \hat{d}_{a5} & \hat{d}_{a6}\hat{d}_{a3} \\
0 & -\hat{d}_{a4} & -\hat{d}_{a2}\hat{d}_{a6} \\
0 & 0 & \hat{d}_{a2}\hat{d}_{a5} - \hat{d}_{a4}\hat{d}_{a3} \\
\mathbf{0}_{3\times1} & -{}^I_a\hat{\mathbf{R}}\mathbf{e}_3 & {}^I_a\hat{\mathbf{R}}(\mathbf{e}_1\hat{d}_{a2} + \mathbf{e}_2\hat{d}_{a3}) \\
\mathbf{0}_{28\times1} & \mathbf{0}_{28\times1} & \mathbf{0}_{28\times1}
\end{bmatrix}
\tag{81}
$$

If $^a a_2$ is constant, $d_{a2}$, $d_{a3}$ and roll of $^I_a\mathbf{R}$ are unobservable with unobservable directions as:

$$
\mathbf{N}_{a2} = \begin{bmatrix}
\mathbf{0}_{12\times1} & \mathbf{0}_{12\times1} & \mathbf{0}_{12\times1} \\
\hat{\mathbf{D}}_a^{-1}\mathbf{e}_1{}^a a_2 & \hat{\mathbf{D}}_a^{-1}\mathbf{e}_2{}^a a_2 & \hat{\mathbf{D}}_a^{-1}\mathbf{e}_3\hat{d}_{a3}{}^a a_2 \\
\mathbf{0}_{6\times1} & \mathbf{0}_{6\times1} & \mathbf{0}_{6\times1} \\
0 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & \hat{d}_{a6} \\
0 & 0 & -\hat{d}_{a5} \\
\mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & -{}^I_a\hat{\mathbf{R}}\mathbf{e}_1 \\
\mathbf{0}_{28\times1} & \mathbf{0}_{28\times1} & \mathbf{0}_{28\times1}
\end{bmatrix}
\tag{82}
$$

If $^a a_3$ is constant, $d_{a4}$, $d_{a5}$ and $d_{a6}$ are unobservable with unobservable directions as:

$$
\mathbf{N}_{a3} = \begin{bmatrix}
\mathbf{0}_{1\times12} & (\hat{\mathbf{D}}_a^{-1}\mathbf{e}_1)^\top{}^a a_3 & \mathbf{0}_{1\times9} & 1 & \mathbf{0}_{1\times33} \\
\mathbf{0}_{1\times12} & (\hat{\mathbf{D}}_a^{-1}\mathbf{e}_2)^\top{}^a a_3 & \mathbf{0}_{1\times10} & 1 & \mathbf{0}_{1\times32} \\
\mathbf{0}_{1\times12} & (\hat{\mathbf{D}}_a^{-1}\mathbf{e}_3)^\top{}^a a_3 & \mathbf{0}_{1\times11} & 1 & \mathbf{0}_{1\times31}
\end{bmatrix}^\top
\tag{83}
$$

See Appendix E for the verification of these unobservable directions. □

### 7.1.3 Degenerate motions for $\mathbf{T}_g$haha

As $^I\mathbf{a}$ (the acceleration in IMU frame) can affect the observability property for the g-sensitivity $\mathbf{T}_g$, by close inspection of special configurations for $^I\mathbf{a}$, we have:

**Lemma 4.** *If any component of $^I\mathbf{a}$ (including $^I a_1$, $^I a_2$ and $^I a_3$) is constant, then $\mathbf{T}_g$ will become unobservable.*

Table 2: Summary of basic degenerate motions for online IMU intrinsics calibration (*imu22*).

| Motion Types | Dim. | Unobservable Parameters |
|---|---|---|
| constant $^w\omega_1$ | 1 | $d_{w1}$ |
| constant $^w\omega_2$ | 2 | $d_{w2}, d_{w3}$ |
| constant $^w\omega_3$ | 3 | $d_{w4}, d_{w5}, d_{w6}$ |
| constant $^aa_1$ | 3 | $d_{a1}$, pitch and yaw of $^I_a\mathbf{R}$ |
| constant $^aa_2$ | 3 | $d_{a2}, d_{a3}$, roll of $^I_a\mathbf{R}$ |
| constant $^aa_3$ | 3 | $d_{a4}, d_{a5}, d_{a6}$ |
| constant $^Ia_1$ | 3 | $t_{g1}, t_{g2}, t_{g3}$ |
| constant $^Ia_2$ | 3 | $t_{g4}, t_{g5}, t_{g6}$ |
| constant $^Ia_3$ | 3 | $t_{g7}, t_{g8}, t_{g9}$ |

*Proof.* If $^Ia_1$ is constant, $t_{g1}$, $t_{g2}$ and $t_{g3}$ are unobservable with unobservable directions as:

$$\mathbf{N}_{g1} = \begin{bmatrix} \mathbf{0}_{3\times9} & \mathbf{I}_3{}^Ia_1 & \mathbf{0}_{3\times18} & -\mathbf{I}_3 & \mathbf{0}_{3\times25} \end{bmatrix}^\top$$

If $^Ia_2$ is constant, $t_{g4}$, $t_{g5}$ and $t_{g6}$ are unobservable with unobservable directions as:

$$\mathbf{N}_{g2} = \begin{bmatrix} \mathbf{0}_{3\times9} & \mathbf{I}_3{}^Ia_2 & \mathbf{0}_{3\times21} & -\mathbf{I}_3 & \mathbf{0}_{3\times22} \end{bmatrix}^\top$$

If $^Ia_3$ is constant, $t_{g7}$, $t_{g8}$ and $t_{g9}$ are unobservable with unobservable directions as:

$$\mathbf{N}_{g3} = \begin{bmatrix} \mathbf{0}_{3\times9} & \mathbf{I}_3{}^Ia_3 & \mathbf{0}_{3\times24} & -\mathbf{I}_3 & \mathbf{0}_{3\times19} \end{bmatrix}^\top$$

See Appendix E for the verification of these unobservable directions. □

### 7.1.4 Discussion on IMU intrinsic degeneracy

It is evident from the above analysis that the IMU intrinsic calibration is sensitive to sensor motion and thus all 6 axes need to be excited to ensure all of them can be calibrated. These findings are summarized in Table 2. It should be noted that any combination of these primitive motions is still degenerate and causes all related parameters to become unobservable (e.g., planar motion with constant acceleration). It is also important to mention that it is common that $^I_a\mathbf{R} \simeq \mathbf{I}_3$ and $\mathbf{D}_a \simeq \mathbf{I}_3$ for most IMUs, and thus, $^a\hat{\mathbf{a}} \simeq {}^I\mathbf{a}$. As such, the degenerate motions for $\mathbf{D}_a$ will also lead to the unobservability of $\mathbf{T}_g$, and vice-versa. Again, this degenerate motion analysis can be extended to other model variants, which is omitted here for brevity.

## 7.2 IMU-Camera Spatial-Temporal Parameters

Leveraging our previous work [18] where we have studied four commonly-seen degenerate motions of VINS with only IMU-camera spatial-temporal calibration, we here show these degenerate motions hold true for VINS with full-parameter calibration:

**Lemma 5.** *The IMU-camera spatial-temporal calibration will become unobservable, if the sensor platform undergoes the following degenerate motions:*

- *Pure translation*

- *One-axis rotation*

Table 3: Summary of basic degenerate motions for online IMU-camera spatial-temporal calibration.

| Motion Types | Unobservable Parameters | Observable |
|---|---|---|
| pure translation | $^C\mathbf{p}_I$ | $^C_I\mathbf{R}, t_d, t_r$ |
| one-axis rotation | $^C\mathbf{p}_I$ along rotation axis | $^C_I\mathbf{R}, t_d, t_r$ |
| constant $^I\boldsymbol{\omega}$ | $t_d$ and | $^C_I\mathbf{R}, t_r$ |
| constant $^I\mathbf{v}$ | $^C\mathbf{p}_I$ along rotation axis | |
| constant $^I\boldsymbol{\omega}$ | $t_d$ and | $^C_I\mathbf{R}, t_r$ |
| constant $^G\mathbf{a}$ | $^C\mathbf{p}_I$ along rotation axis | |

- *Constant local angular and linear velocity*

- *Constant local angular velocity and global linear acceleration*

*Proof.* If the system undergoes pure translation (no rotation), the translation part $^C\mathbf{p}_I$ of the spatial calibration will be unobservable, residing along the following unobservable directions:

$$\mathbf{N}_{pt} = \begin{bmatrix} \mathbf{0}_{3\times 45} & \mathbf{I}_3 & \mathbf{0}_{3\times 10} & -({}^G_{I_1}\hat{\mathbf{R}}{}^I_C\hat{\mathbf{R}})^\top \end{bmatrix}^\top \tag{84}$$

If the system undergoes random (general) translation but with only one-axis rotation, the translation calibration $^C\mathbf{p}_I$ along the rotation axis will be unobservable, with the following unobservable direction:

$$\mathbf{N}_{oa} = \begin{bmatrix} \mathbf{0}_{1\times 45} & ({}^C_I\hat{\mathbf{R}}{}^I\hat{\mathbf{k}})^\top & 0_{1\times 10} & -({}^G_{I_1}\hat{\mathbf{R}}{}^I\hat{\mathbf{k}})^\top \end{bmatrix}^\top \tag{85}$$

where $^I\mathbf{k}$ is the constant rotation axis in the IMU frame $\{I\}$.

If the VINS undergoes constant local angular velocity $^I\boldsymbol{\omega}$ and linear velocity $^I\mathbf{v}$, the time offset $t_d$ will be unobservable with the following unobservable direction:

$$\mathbf{N}_{t1} = \begin{bmatrix} \mathbf{0}_{1\times 42} & ({}^C_I\hat{\mathbf{R}}{}^I\hat{\boldsymbol{\omega}})^\top & -({}^C_I\hat{\mathbf{R}}{}^I\hat{\mathbf{v}})^\top & -1 & \mathbf{0}_{1\times 12} \end{bmatrix}^\top \tag{86}$$

If the VINS undergoes constant local angular velocity $^I\boldsymbol{\omega}$ and global acceleration $^G\mathbf{a}$, the time offset $t_d$ will be unobservable with the following unobservable direction:

$$\mathbf{N}_{t2} = \begin{bmatrix} \mathbf{0}_{1\times 6} & {}^G\hat{\mathbf{a}} & \mathbf{0}_{1\times 30} & ({}^C_I\hat{\mathbf{R}}{}^I\hat{\boldsymbol{\omega}})^\top & \mathbf{0}_{1\times 3} & -1 & \mathbf{0}_{1\times 9} & -({}^G\hat{\mathbf{v}}_{I_1})^\top \end{bmatrix}^\top \tag{87}$$

$\square$

Table 3 summarizes these degenerate motions for completeness. It is important to note that unlike $t_d$ (whose Jacobian is mainly affected by the sensor motion), the Jacobian for RS readout time, $t_r$, is also affected by the feature observations due to the term $\frac{m}{M}$ [see Eq. (133)], and is observable, as hundreds of features can be observed from different image rows during exploration.

## 7.3 Camera Intrinsic Parameters

As mentioned before, the camera intrinsics are mainly affected by the observed features. By investigating special feature configurations, we find the following degenerate case for camera calibration when using a *radtan* distortion model:

**Lemma 6.** *The camera intrinsics will become unobservable if the following conditions are satisfied:*

- *The features keep the same depth relative to the camera (e.g., $^C z_f$ is constant in value).*

- *The camera moves with one-axis rotation and the rotation axis is defined as $^C \mathbf{k} = \mathbf{e}_3$.*

*Proof.* The camera focal length $f_u$, $f_v$, the camera distortion model $k_1$, $k_2$, $p_1$ and $p_2$ will become unobservable with unobservable direction:

$$\mathbf{N}_{Cin} = \begin{bmatrix} \mathbf{0}_{1\times 47} & f_u & f_v & \mathbf{0}_{1\times 2} & 2k_1 & 4k_2 & p_1 & p_2 & {}^G\mathbf{k}^\top \end{bmatrix}^\top$$

with $^G\mathbf{k} = {}_{I_0}^G\hat{\mathbf{R}}{}_C^I\hat{\mathbf{R}}{}^C\mathbf{k}{}^C z_f$ .

See Appendix E for the verification of these unobservable directions. □

As an example, if a ground vehicle is performing planar motion with a upward facing camera only observing features from the ceilings, the above two conditions will hold and thus the camera intrinsics with *radtan* distortion model will be unobservable. Nevertheless, since it is common to observe hundreds of features, it might be rarely the case that every feature maintains the same relative depth, $^C z_f$, to the camera, and thus, this degeneracy may not happen in practice if features are tracked uniformly throughout images. It is interesting to note that this degenerate case does not work for camera models with *equidist* distortion.

# 8   State Estimator

Leveraging our MSCKF-based VINS [10], the proposed estimator extends the state vector $\mathbf{x}_k$ at time step $k$ to include the current IMU state $\mathbf{x}_{I_k}$, a sliding window of cloned IMU poses $\mathbf{x}_c$, the calibration parameters ($\mathbf{x}_{IC}$ and $\mathbf{x}_{Cin}$) and feature state $\mathbf{x}_f$.

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{I_k}^\top & \mathbf{x}_c^\top & \mathbf{x}_{IC}^\top & \mathbf{x}_{Cin}^\top & \mathbf{x}_f^\top \end{bmatrix}^\top \tag{88}$$

$$\mathbf{x}_c = \begin{bmatrix} {}_G^{I_{ck-1}}\bar{q}^\top & {}^G\mathbf{p}_{I_{ck-1}}^\top & \cdots & {}_G^{I_{ck-n}}\bar{q}^\top & {}^G\mathbf{p}_{I_{ck-n}}^\top \end{bmatrix}^\top \tag{89}$$

where $\mathbf{x}_I$, $\mathbf{x}_{IC}$, $\mathbf{x}_{Cin}$ and $\mathbf{x}_f$ are the same as Eq. (30), $\mathbf{x}_c$ denotes the sliding window containing $n$ cloned IMU poses with index from $ck - n$ to $ck - 1$. Note that the IMU intrinsics $\mathbf{x}_{in}$ are contained in the current IMU state $\mathbf{x}_{I_k}$.

As $\mathbf{x}_c$, $\mathbf{x}_{IC}$, $\mathbf{x}_{Cin}$ and $\mathbf{x}_f$ have zero dynamics, we only propagate the estimate and covariance of the next IMU state based on Eq. (53)-(57) and Eq. (63), which all incorporate the IMU intrinsics $\mathbf{x}_{in}$.

As in [5], we handle the IMU-camera time offset $t_d$ when we clone the "true" IMU pose corresponding to image measurements. For example, if we clone the current IMU pose $\{_G^{I_k}\bar{q}, {}^G\mathbf{p}_{I_k}\}$ into the sliding window as $\{_G^{I_{ck}}\bar{q}, {}^G\mathbf{p}_{I_{ck}}\}$ using:

$$_{I_{ck}}^G\mathbf{R} \simeq {}_{I_k}^G\mathbf{R}\exp({}^{I_k}\hat{\boldsymbol{\omega}}\tilde{t}_d) \tag{90}$$

$$^G\mathbf{p}_{I_{ck}} \simeq {}^G\mathbf{p}_{I_k} + {}^G\mathbf{v}_{I_k}\tilde{t}_d \tag{91}$$

with the linearized clone Jacobians as:

$$\begin{bmatrix} \delta\boldsymbol{\theta}_{I_{ck}} \\ {}^G\tilde{\mathbf{p}}_{I_{ck}} \end{bmatrix} \simeq \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & {}^{I_k}\hat{\boldsymbol{\omega}} \\ \mathbf{0}_3 & \mathbf{I}_3 & {}^G\hat{\mathbf{v}}_{I_k} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\theta}_{I_k} \\ {}^G\tilde{\mathbf{p}}_{I_k} \\ \tilde{t}_d \end{bmatrix} \tag{92}$$

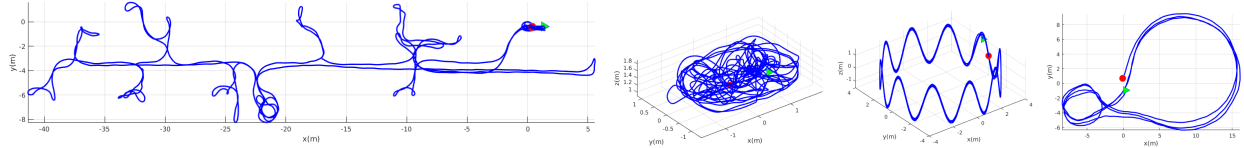Both $\mathbf{x}_{in}$ and $t_d$ will be updated through correlations when visual feature measurements are present.

Figure 3: Simulated trajectories for Monte-Carlo simulations. Left: *tum_corridor* with fully excited 3D motion; Middle left: *tum_room* with 1 axis rotation and 3D translation; Middle right: *sine_3d* with constant acceleration along $x$ direction; Right: *udel_gore* planar motion with constant $z$ and only yaw rotation. The green triangle and red circle denote the beginning and ending of these trajectories, respectively.

Features are processed in two different ways: short features update the state through the MSCKF nullspace operation [8, 68], and long-tracked features are initialized into the state vector and refined over time for improved accuracy [69]. We utilize first-estimates Jacobians (FEJ) [50, 5] to preserve the system unobservable subspace and improve the estimator consistency. We directly model the camera intrinsic and IMU-camera spatial calibration through the visual measurement functions [see Eq. (69)] and update them in the filter with Jacobians in Eq. (74).

For the RS cameras, the feature measurements from different image rows are captured at different timestamps. This indicates that we cannot directly find a cloned pose in the sliding window for $\{^G_{I(t)}\mathbf{R}, {}^G\mathbf{p}_{I(t)}\}$ shown in Eq. (27). Therefore, for the readout time calibration, we model the feature measurement affected by RS effects through pose interpolation [6, 11]. For example, if the feature measurement is in the $m$-th row with total $M$ rows in an image, we can find two bounding clones $ci - 1$ and $ci$ based on the measurement time $t$. Hence, the corresponding time $t$ is between two clones within the sliding window, that is: $t_{ck-n} \le t_{ci-1} \le t \le t_{ci} \le t_{ck}$. We can then find the *virtual* IMU pose $\{^G_{I(t)}\mathbf{R}, {}^G\mathbf{p}_{I(t)}\}$ between clones $ci - 1$ and $ci$ with:

$$\lambda = (t_I + \frac{m}{M}t_r - t_{ci-1})/(t_{ci} - t_{ci-1}) \tag{93}$$

$$^G_{I(t)}\mathbf{R} = {}^G_{I_{ci-1}}\mathbf{R}\exp\left(\lambda\log\left(^G_{I_{ci-1}}\mathbf{R}^\top{}^G_{I_{ci}}\mathbf{R}\right)\right) \tag{94}$$

$$^G\mathbf{p}_{I(t)} = (1-\lambda)^G\mathbf{p}_{I_{ci-1}} + \lambda^G\mathbf{p}_{I_{ci}} \tag{95}$$

To summarize, feature measurements which occur at different rows of the image can be related to the state vector defined in Eq. (88) through the above linear pose interpolation. This measurement function can then be linearized for use in the EKF update (see Appendix F). Note that a higher-order polynomial pose interpolation as used by [11] can be utilized if necessary. In the above derivations, although we assume the image timestamp refers to the timestamp of the first image row, the interpolation scalar in Eq. (93) can be easily modified to account for situations when the image timestamp of the RS camera refers to the middle or last image row.

## 9 Simulation Validations

The proposed estimator is implemented within the OpenVINS framework [10], which contains a real-time modular sliding window EKF-based filter and simulator. The project can already handle IMU-camera spatial-temporal and camera intrinsic calibrations, excluding RS readout time. In this work we extend the estimator to address the calibration of all the parameters presented. The new VINS estimator maintains the original faster than real-time performance in both simulated and real-world datasets.

The core of the simulator has a continuous-time $SE(3)$ B-spline trajectory representation which allows for the calculation of pose, velocity, and accelerations at any given timestamp along the tra-

Table 4: Simulation parameters and prior standard deviations that perturbations of measurements and initial states were drawn from.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| IMU Scale | 0.003 | IMU Skew | 0.003 |
| Rot. atoI (rad) | 0.003 | Rot. wtoI (rad) | 0.003 |
| Gyro. White Noise | 1.6968e-04 | Gyro. Rand. Walk | 1.9393e-05 |
| Accel. White Noise | 2.0000e-3 | Accel. Rand. Walk | 3.0000e-3 |
| Focal Len. (px/m) | 0.50 | Cam. Center (px) | 0.60 |
| d1 and d2 | 0.008 | d3 and d4 | 0.002 |
| Rot. CtoI (rad) | 0.004 | Pos. IinC (m) | 0.010 |
| Readout Time (ms) | 0.5 | Timeoff (s) | 0.005 |
| Cam Freq. (hz) | 20 | IMU Freq. (hz) | 400 |
| Avg. Feats | 100 | Num. SLAM | 50 |
| Num. Clones | 20 | Feat. Rep. | GLOBAL |

Table 5: Average absolute trajectory error (ATE) and normalized estimation errror squared (NEES) over 20 runs of the proposed system evaluated on *tum_corridor* with true or perturbed calibration parameters, with and without online calibration. *radtan* camera distortion model and different IMU intrinsic models are used. The notation "true" means the groundtruth calibration, while "perturbed" means the perturbed calibration states. Failures are denoted with "-".

| IMU Model | ATE (deg) | ATE (m) | Ori. NEES | Pos. NEES | IMU Model | ATE (deg) | ATE (m) | Ori. NEES | Pos. NEES |
|---|---|---|---|---|---|---|---|---|---|
| true w/ calib *imu1* | 0.462 | 0.164 | 1.910 | 1.423 | perturbed w/ calib *imu1* | 0.454 | 0.163 | 2.173 | 1.473 |
| true w/ calib *imu2* | 0.460 | 0.164 | 2.103 | 1.422 | perturbed w/ calib *imu2* | 0.446 | 0.162 | 2.150 | 1.465 |
| true w/ calib *imu3* | 0.461 | 0.163 | 1.883 | 1.422 | perturbed w/ calib *imu3* | 0.459 | 0.163 | 2.125 | 1.454 |
| true w/ calib *imu4* | 0.458 | 0.163 | 2.102 | 1.424 | perturbed w/ calib *imu4* | 0.450 | 0.162 | 2.150 | 1.456 |
| true w/ calib *imu11* | 0.544 | 0.177 | 1.947 | 1.472 | perturbed w/ calib *imu11* | 0.550 | 0.178 | 2.243 | 1.498 |
| true w/ calib *imu12* | 0.540 | 0.177 | 2.123 | 1.476 | perturbed w/ calib *imu12* | 0.544 | 0.178 | 2.175 | 1.493 |
| true w/ calib *imu13* | 0.544 | 0.176 | 1.914 | 1.472 | perturbed w/ calib *imu13* | 0.546 | 0.177 | 2.180 | 1.482 |
| true w/ calib *imu14* | 0.544 | 0.179 | 2.124 | 1.483 | perturbed w/ calib *imu14* | 0.538 | 0.177 | 2.169 | 1.504 |
| true w/ calib *imu21* | 0.572 | 0.183 | 1.990 | 1.514 | perturbed w/ calib *imu21* | 0.576 | 0.182 | 2.250 | 1.508 |
| true w/ calib *imu22* | 0.567 | 0.184 | 2.145 | 1.513 | perturbed w/ calib *imu22* | 0.590 | 0.187 | 2.194 | 1.561 |
| true w/ calib *imu23* | 0.571 | 0.183 | 1.962 | 1.514 | perturbed w/ calib *imu23* | 0.593 | 0.185 | 2.200 | 1.550 |
| true w/ calib *imu24* | 0.566 | 0.183 | 2.141 | 1.512 | perturbed w/ calib *imu24* | 0.585 | 0.186 | 2.189 | 1.552 |
| true w/ calib *imu31* | 0.447 | 0.161 | 2.076 | 1.378 | perturbed w/ calib *imu31* | 0.451 | 0.162 | 2.110 | 1.428 |
| true w/ calib *imu32* | 0.444 | 0.161 | 1.879 | 1.396 | perturbed w/ calib *imu32* | 0.447 | 0.162 | 1.968 | 1.430 |
| true w/ calib *imu33* | 0.529 | 0.175 | 2.096 | 1.378 | perturbed w/ calib *imu33* | 0.527 | 0.177 | 2.101 | 1.411 |
| true w/ calib *imu34* | 0.548 | 0.180 | 2.103 | 1.430 | perturbed w/ calib *imu34* | 0.549 | 0.179 | 2.113 | 1.422 |
| true w/ calib *imu6* | 0.572 | 0.183 | 1.734 | 1.517 | perturbed w/ calib *imu6* | 0.567 | 0.179 | 1.910 | 1.491 |
| true w/o calib | 0.433 | 0.159 | 2.069 | 1.332 | perturbed w/o calib | - | - | - | - |
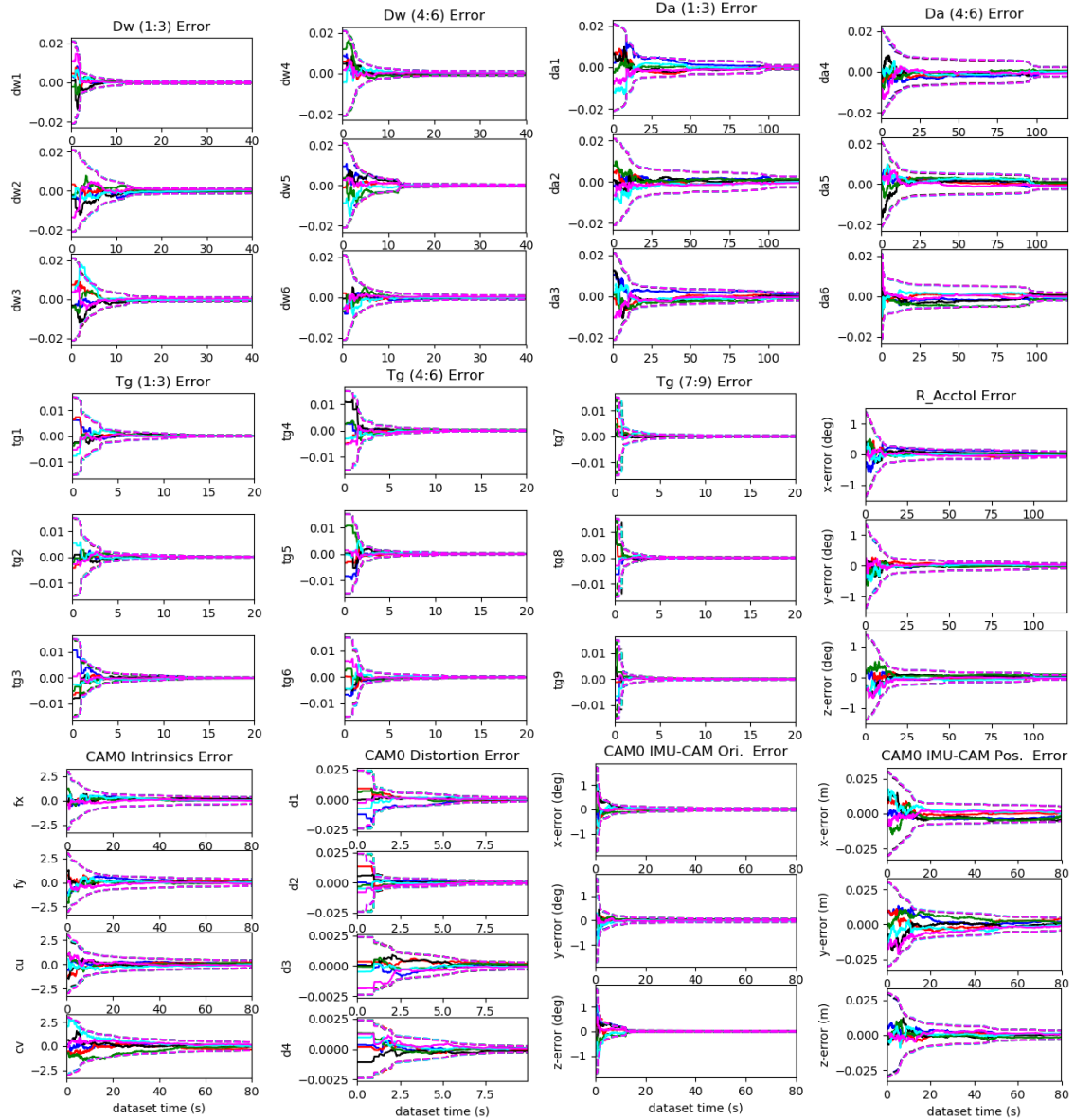
Figure 4: Calibration results for proposed system evaluated on *tum_corridor* with fully excited motion (using *imu22* and *radtan*). $3\sigma$ bounds (dotted lines) and estimation errors (solid lines) for six different runs (different colors) with different realization of the measurement noise and initial perturbations. All the calibration parameters converge nicely.
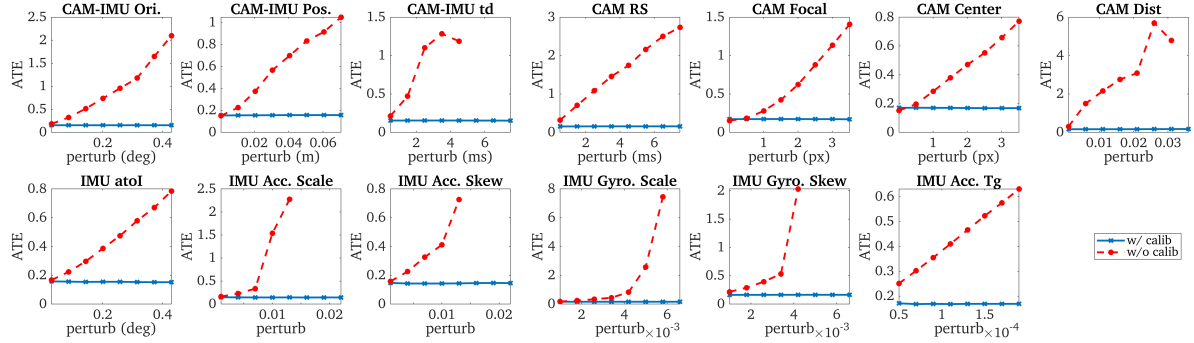
Figure 5: Absolute trajectory errors (ATE) using *imu22* and *radtan* on the *tum_corridor* with full 3D motion given different levels of perturbation. Each parameter calibrated was only perturbed while other parameters were initialized to their true values and not estimated. ATE above eight meters were not reported in the figures and can be considered diverged.

jectory. The true angular velocity and linear accelerations can be directly found and corrupted using the random walk biases and white noises. The basic configurations for our simulator are listed in Table 4. All simulation convergence figures show the $3\sigma$ bounds (dotted lines) and estimation errors (solid lines) for six different runs (different colors) with different realization of the measurement noises and initial calibration state perturbations.

To simulate RS visual bearing measurements, we follow the logic presented by [5] and [11]. Specifically, static environmental features are first generated along the length of the trajectory at random depths and bearings. Then, for a given imaging time of features in view, we project each into the current image frame using the true camera intrinsic and distortion model to find the corresponding observation row. Given this projected row and image time, we can find the pose at which this row should have been (i.e., the pose at which that RS row should have been exposed). We can then re-project this feature into the new pose and iterate until the projected row does not change (which typically requires 2-3 iterations). We now have a feature measurement which occurred at the correct pose for its given RS row. This measurement is then corrupted with white noise. The imaging timestamp corresponding to the starting row is then shifted by the true IMU-camera time offset $t_d$ to simulate cross-sensor delay.

## 9.1 Simulation with Fully-Excited Motion

We first perform a general trajectory simulation, for which we perform full calibration of the IMU-camera extrinsics, time offset, RS readout time, camera intrinsics with *radtan* model and IMU intrinsics with *imu22*. The trajectory, shown in the left of Figure 3, is designed based on *tum_corridor* sequence of TUM visual-inertial dataset with full excitation of all 6 axes and provides a realistic 3D hand-held motion [70]. From the results shown in Figure 4 and 9, the estimation errors and $3\sigma$ bounds for all the calibration parameters (including *imu22* and *radtan*) can converge quite nicely, verifying that the analysis for general motions holds true. We plot results from six different realizations of the initial calibration guesses based on the specified priors, and it is clear that the estimates for all these calibration parameters are able to converge from different initial guesses to near the ground truth. Each parameter is able to "gain" information since their $3\sigma$ bounds shrink. These results verify our Lemma 1 that all these online calibration parameters are observable given a fully-excited motion.
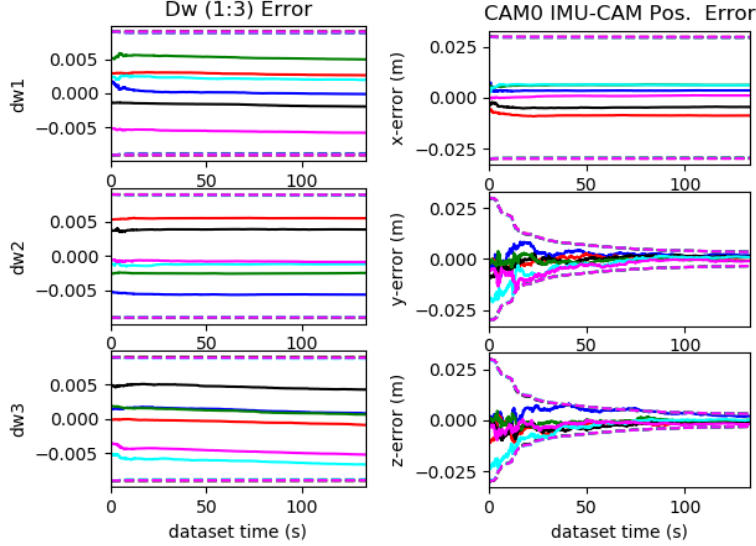
Figure 6: Calibration results for the proposed system evaluated on *tum_room* with one-axis rotation using *imu22* and *radtan*. Note that the estimation errors and $3\sigma$ bounds for $d_{w1}$, $d_{w2}$, $d_{w3}$ and the IMU-camera position calibration along the rotation axis can not converge.

## 9.2   Sensitivities to Perturbations

The next natural question is how robust the system is to the initial perturbations and whether the use of online sensor calibration enables improvements in robustness and accuracy. Shown in Figure 5, for each of the different calibration parameters we perturb it with different levels of noise on the *tum_corridor* trajectory (note that we also change the initial prior provided to the filter as the initial prior changes). We can see that the proposed estimator is relatively invariant to the initial inaccuracies of the parameters and is, in general, able to output a near constant trajectory error. A filter, which does not perform this online estimation, has its trajectory estimation error quickly increase to non-usable levels. It is interesting to see that even small levels of perturbations can cause huge trajectory errors which further verifies the motivations to perform online calibration.

## 9.3   Comparison of Inertial Model Variants

We next compare the different proposed inertial model variants. We estimate all calibration parameters and perturb them based on Table 4. Shown in Table 5, it is clear that the choice between the variants has little impact on estimation accuracy which indicates they provide almost the same amount of correction to the inertial readings.

   The accuracy of the standard VIO system which does not calibrate any parameters online and uses the groundtruth calibration values, denoted *true w/o calib*, has the best accuracy due to the use of the true parameters. If we do perturb the initial calibration and do not estimate it, denoted *perturbed w/o calib*, the system quickly becomes unstable and diverges unless smaller levels of perturbations are used. Note that the results presented in Section 9.2 are for each parameter perturbed individually, while here all calibration parameters are perturbed at once thus resulting in divergence. With full online calibration, the system can still output stable and consistent trajectories with only a small loss in estimation accuracy given perturbed calibration values.
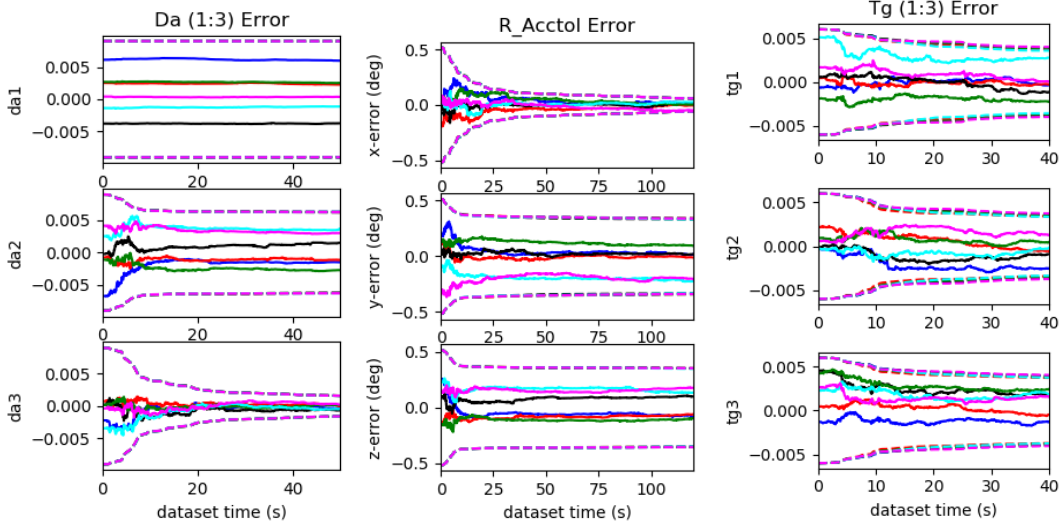
Figure 7: Calibration results for the proposed system evaluated the *sine_3d* with constant acceleration along x direction using *imu22* and *radtan*. The estimation errors and $3\sigma$ bounds for $d_{a1}$, pitch and yaw of $_a^I\mathbf{R}$ cannot converge. Note that $t_{g1}$, $t_{g2}$ and $t_{g3}$ are also unobservable.

## 9.4 Degenerate Motion Verification

We now verify the identified degenerate motions and present results for three special motions. In all simulations, we perform full-parameter calibration to fully test our system and present the complete results in Appendix G. The trajectories shown in Figure 3 are created as follows:

- One-axis rotation with a modified *tum_room* trajectory, see middle left, which has its roll and pitch orientation changes removed to create a yaw and 3D translation only dataset.

- Constant local $a_x$ with modified *sine_3d*, see middle right, for which we have a constant pitch and make the current yaw angle tangent to the trajectory in the x-y plane (gives constant local acceleration along local x-axis).

- Planar motion with modified *udel_gore*, see right, which has its roll and pitch orientation removed and all poses are projected to x-y plane (planar motion in the global x-y plane).

### 9.4.1 One-axis rotation motion.

Shown in Figure 6, the first 3 parameters ($d_{w1}$, $d_{w2}$ and $d_{w3}$) for $\mathbf{D}_w$ do not converge at all (the $3\sigma$ bounds are almost straight lines), which matches our analysis, see Table 2. These parameters should be unobservable in the case of one-axis rotation with $^ww_x$ (roll) and $^ww_y$ (pitch) are constant. Additionally, the translation between IMU and camera does not converge either. The x-error of the IMU-camera translation even diverges reinforcing the undesirability of degenerate motions and verifies the analysis presented in Table 3.

### 9.4.2 Constant local acceleration motion.

The results shown in Figure 7, where we have enforced that the local acceleration along the x-axis, $a_x$, is constant. The $d_{a1}$, and pitch and yaw of $_a^I\mathbf{R}$ does not converge, thus validating our analysis shown in Table 2. Note that in the simulation, we have set $_a^I\mathbf{R} \simeq \mathbf{I}_3$ and $\mathbf{D}_a \simeq \mathbf{I}_3$. Hence, $^a\hat{\mathbf{a}} \simeq {}^I\mathbf{a}$
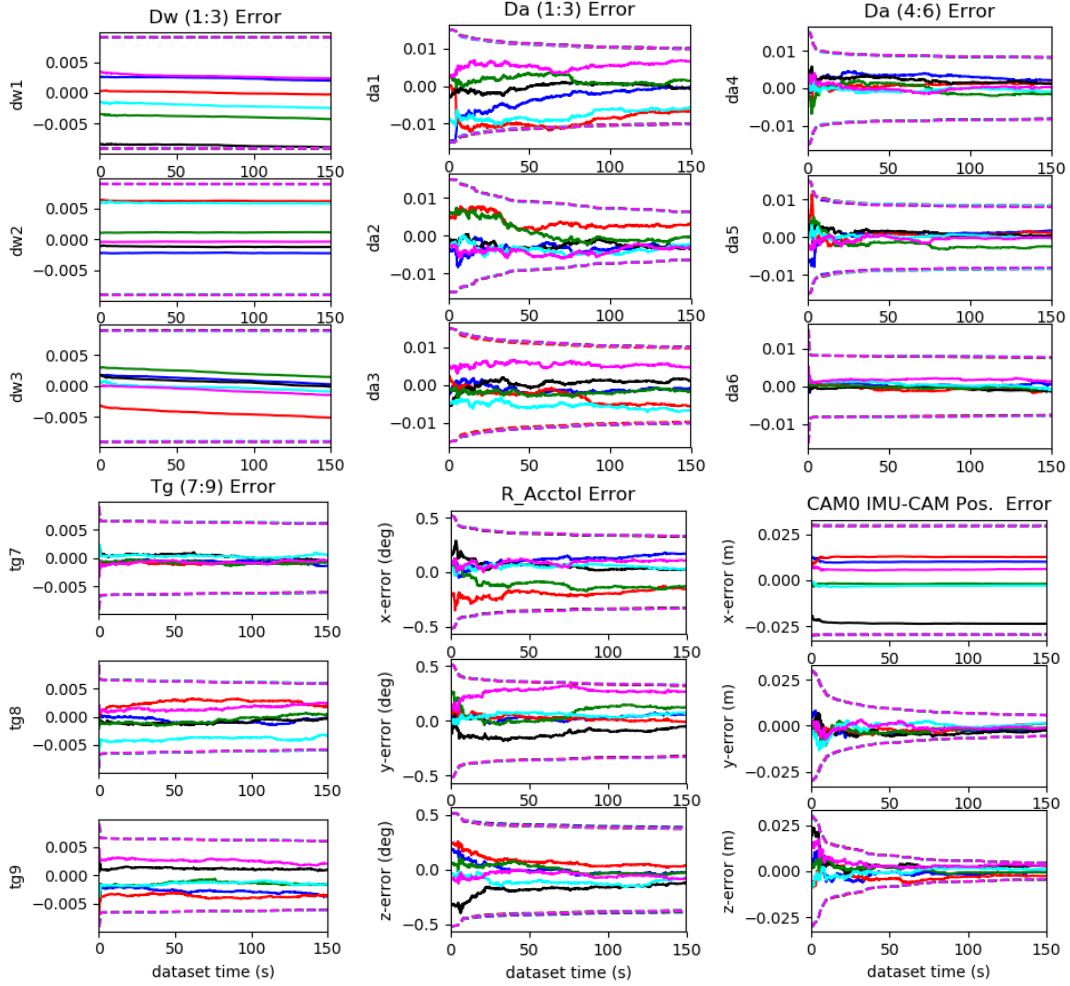
Figure 8: Calibration results of the proposed system evaluated on *udel_gore* with planar motion using *imu22* and *radtan*. With planar motion, the estimation errors and 3 $\sigma$ bounds of $d_{w1}$, $d_{w2}$, $d_{w3}$, $t_{g7}$, $t_{g8}$, $t_{g9}$ and the IMU-camera position cannot converge. Due to lack of motion excitation, the parameters of $\mathbf{D}_a$ and $_a^I\mathbf{R}$ converge much slower than the other motion cases.
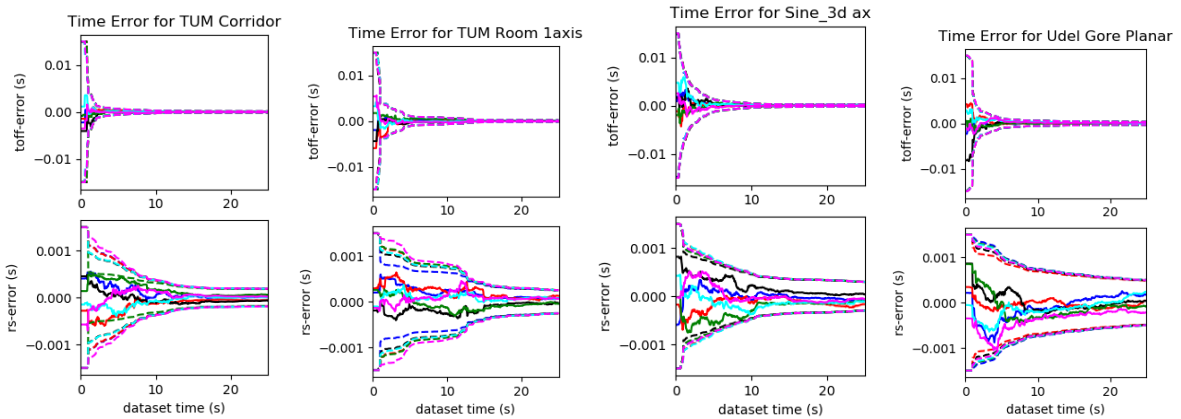


Figure 9: Camera temporal and read out time calibration results of the proposed system (using *imu22* and *radtan*) for different trajectories. The temporal parameters can finally converge in all the 4 motion types. Note that the readout time converges slower in the planar motion case probably due to the lack of motion in the beginning.
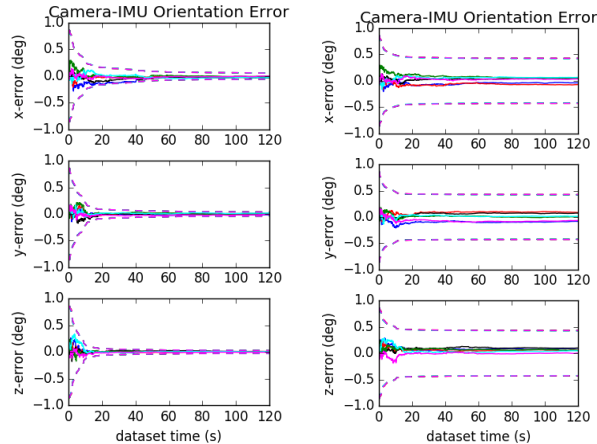
Figure 10: Camera to IMU orientation errors when using IMU *imu2* (left) and the over paramterized *imu5* (right). Note that only the IMU intrinsics and relative pose between IMU and camera were online calibrated.

and $^I a_x$ is also near constant. Therefore, three terms of g-sensitivity ($t_{g1}$, $t_{g2}$ and $t_{g2}$) are also unobservable and converge much slower than other terms.

### 9.4.3 Planar motion.

Shown in Figure 8, with one-axis rotation (yaw axis) for planar motion the $d_{w1}$, $d_{w2}$ and $d_{w3}$ for $\mathbf{D}_w$ and the IMU-camera translation are unobservable and does not converge. Since the $^I \mathbf{a}_z$ is constant, the last three terms of gravity sensitivity ($t_{g7}$, $t_{g8}$ and $t_{g9}$) become unobservable and cannot converge. Both these results verify our analysis shown in Tables 2 and 3. Additionally, this trajectory is quite smooth with small excitation of linear acceleration, hence, the terms of $\mathbf{D}_a$ and $^I_a \mathbf{R}$ in general converge much slower than the fully excited motion case.

### 9.5 Simulated Over Parametrization

We now look to investigate the impact of poor choice of calibration parameters which *over parameterizes* the IMU intrinsics. The *imu5* model, see Table 1, is an over parametrization since we calibrate both 9 parameters for gyroscope and accelerometer, which causes the IMU-camera orientation to be affected since the intermediate inertial frame $\{I\}$ is not constrained. If we change the relative rotation from $\{I\}$ to $\{C\}$, then this perturbed rotation can be absorbed into the $\{a\}$ to $\{I\}$ and $\{w\}$ to $\{I\}$ terms. Thus, it means we have an extra 3 degrees of freedom (DoF) for rotation not constrained by our measurements. We compare this *imu5* model to its close equivalent *imu2* model in Figure 10. We can see that even though the trajectory fully excites the sensor platform, the convergence of $^C_I \mathbf{R}$ becomes much worse if we calibrate IMU-camera extrinsics and all 18 parameters for the IMU model *imu5* even when the *same* priors and measurements are used. This further motivates the use of the minimal calibration parameters to ensure fast and robust convergence of all state parameters.

## 10 Real-World TUM RS VIO Datasets

We first evaluate our proposed algorithms on the TUM RS VIO Dataset which contains a time-synchronized stereo pair of two uEye UI-3241LE-M-GL cameras (left: global-shutter and right:

Figure 11: Results on TUM Rolling Shutter VIO Dataset *without* rolling shutter readout time calibration, with different IMU intrinsic models. The averaged absolute trajectory errors (ATE) of 5 runs in degree (top) and meters (bottom) are provided. Note that the camera intrinsics, and IMU-camera spatial-temporal calibration.

Table 6: Averaged absolute trajectory errors (ATE) of 5 runs over all 8 sequences of the TUM Rolling Shutter VIO Dataset with rolling shutter, camera intrinsics, and IMU-CAM spatial-temporal calibration.

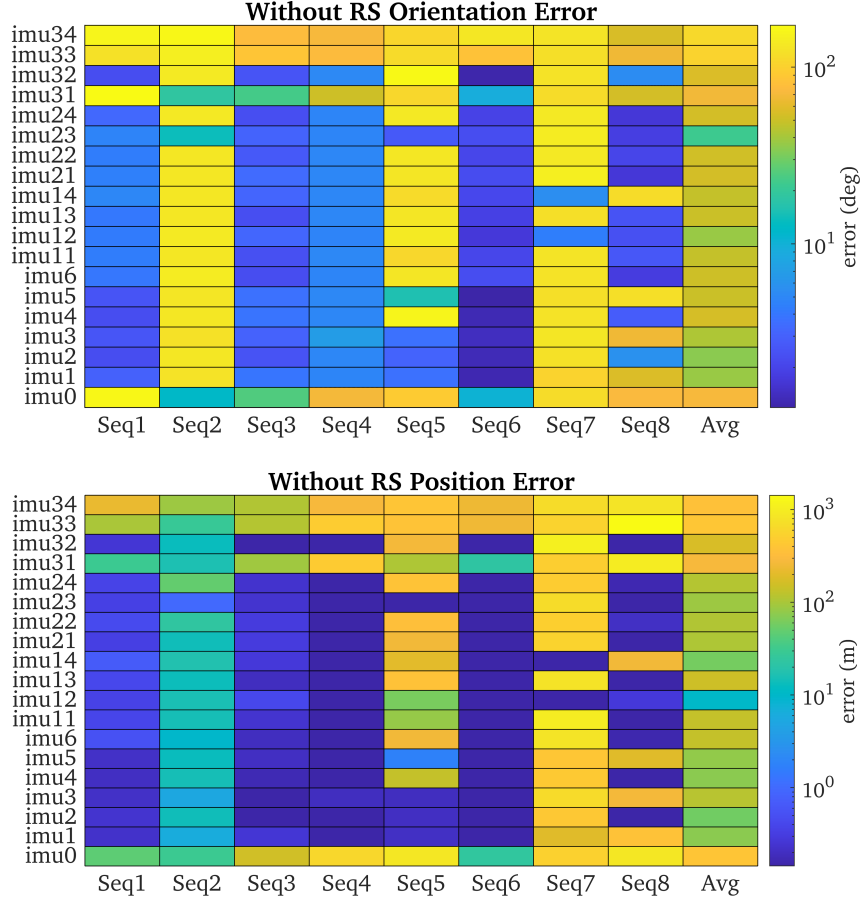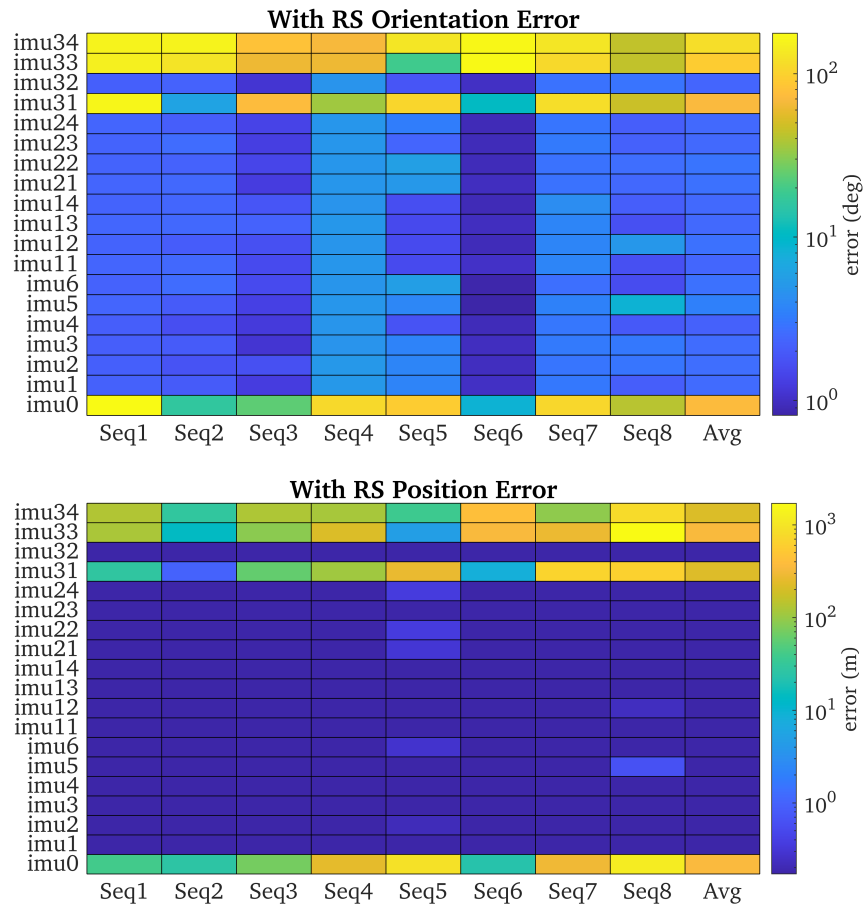| IMU Model | ATE (deg) | ATE (m) |
|:---:|:---:|:---:|
| *imu0* | 72.994 | 363.610 |
| *imu1* | 2.574 | 0.092 |
| *imu2* | 2.679 | 0.094 |
| *imu3* | 2.590 | 0.093 |
| *imu4* | 2.205 | 0.076 |
| *imu5* | 3.418 | 0.149 |
| *imu32* | 2.368 | 0.079 |

Figure 12: Results on TUM Rolling Shutter VIO Dataset *with* rolling shutter readout time calibration, with different IMU intrinsic models. The averaged absolute trajectory errors (ATE) of 5 runs in degree (top) and meters (bottom) are provided. Note that the camera intrinsics, and IMU-camera spatial-temporal calibration.
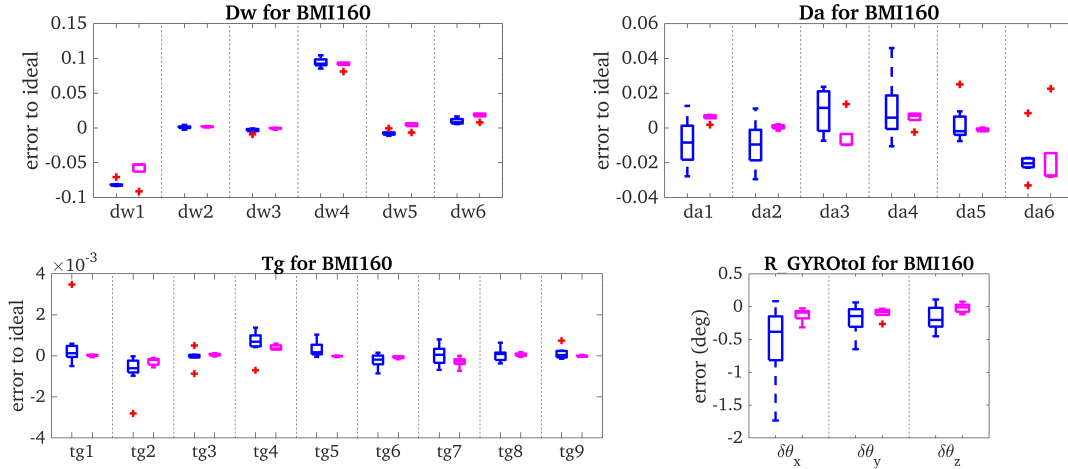
Figure 13: IMU intrinsic evaluation of Bosch BMI160 IMU used in TUM Rolling Shutter VIO datasets using the proposed method with *imu6* and Kalibr relative to the "ideal" sensor intrinsics. The boxplots show the final converged value of both methods. Kalibr (magenta, right in each group) was run with two global shutter cameras and a Bosch BMI160 IMU available over 5 calibration datasets using April tag board, while the proposed system (blue, left) was run with only one rolling shutter camera and the same IMU on the 8 data sequences without any tags. Note that *imu6* is equivalent to the *scale-misalignment* IMU model of Kalibr.

rolling-shutter) and a Bosch BMI160 IMU [71]. When collecting data, the cameras were operated at 20Hz while the IMU operated at 200Hz and an OptiTrack system captured the ground-truth motion. The dataset is provided in both "raw" and "calibrated" formats. The "calibrate" dataset has had IMU intrinsics calibrated from Kalibr [38] pre-applied to the "raw" dataset along with some re-sampling. We evaluate our proposed system by using the right (RS) camera directly with the raw datasets, which have very noisy measurements with varying sensing rates. Hence, the raw datasets are more challenging compared to the calibrated datasets. We re-calibrated the camera intrinsics and IMU-camera spacial-temporal parameters using the raw calibration datasets as the provided calibration parameters were only for the calibrated datasets. We directly use Eq. (29) since the dataset timestamps correspond to the first row of the image. Note that we set the initial values for $\mathbf{D}_a$, $\mathbf{D}_w$, ${}_a^I\mathbf{R}$ and ${}_w^I\mathbf{R}$ as identity and $\mathbf{T}_g$ as zeros, while the initial readout time for the whole RS image is set to 20ms as prior calibration. All IMU intrinsic models listed in Table 1 were run with and without RS calibration. The results are presented in the following sections.

## 10.1   RS Self-Calibration

The results are shown in Figure 11 and 12, with and without RS readout calibration, respectively. It is clear that the systems without RS readout time calibration and without IMU intrinsic calibration (*imu0* and *imu31-imu34*) are unstable and diverges to large orientation and positional errors. With IMU intrinsic calibration, the system still fails for certain datasets (*imu1-imu24*) and thus online readout time calibration will greatly improve the system robustness for RS cameras. The finally estimated RS readout time for each image is around 30ms, which means given the image resolution of $1280 \times 1024$, the row readout time should be around 29us.

## 10.2 IMU Intrinsic Self-Calibration

We focus on the results in Figure 12 which has RS enabled. It is clear from the performance of *imu0* that without IMU intrinsic calibration the BMI160 IMU will cause large trajectory errors, with the models which do perform intrinsic calibration being an order of magnitude more accurate. Table 6 shows the average error over all sequences for the first 6 IMU models. It can be seen that the *imu5* model which over parameterizes the intrinsics has worst accuracy in both orientation and position trajectory estimates, while the accuracy of the other *imu1* - *imu4* models is comparable to each other (similar accuracy level). We further do an ablation study with models *imu31* - *imu34* to find the individual impact of each of the IMU intrinsic parameters. We can see that the *imu32* model which estimates $\mathbf{D}_{w9}$ has large accuracy gains over the other four. This indicates that the readings from gyroscope of BMI160 are very noisy. The calibration of $\mathbf{D}_{w9}$ dominates the performance of this VINS system, and just the calibration of it can achieve similar results as full IMU model calibration (see bottom of Table 6). Through all these we show that online IMU intrinsic calibration can enhance both the system robustness and accuracy.

## 10.3 Comparison to Kalibr Calibration

We run Kalibr's offline calibration with *scale-misalignment* [1] IMU model on 5 calibration datasets provided for the Bosch BMI160 IMU and treat these results as reference values to compare to the proposed *online* calibration results. The Kalibr calibration datasets were collected with the stereo camera pair both operating with a global shutter mode along with an April tag board [38]. For the results evaluation, we directly report $\mathbf{T}'_w = (\mathbf{D}'_w)^{-1}$, $\mathbf{T}'_a = (\mathbf{D}'_a)^{-1}$ and $^w_I\mathbf{R} = ^I_w\mathbf{R}^\top$. By contrast, the proposed system is run on one RS camera with only temporal environmental feature tracks on the 8 data sequences using *imu6*, which is equivalent to the *scale-misalignment* IMU model of Kalibr.

As shown in the boxplots in Figure 13, even though we run on more challenging datasets in real-time, our proposed system can still achieve reasonable calibration results for $\mathbf{D}'_w$, $\mathbf{D}'_a$, $\mathbf{T}_g$ and $^I_w\mathbf{R}$, which are close to the values from baseline Kalibr. Additionally we can see that the values of g-sensitivity $\mathbf{T}_g$ of the BMI160 IMU are generally one or two orders smaller than the other IMU intrinsics. This matches the results presented in Figure 12, for which the estimation errors of *imu1* - *imu4* (without g-sensitivity) are similarly to those of *imu11* - *imu14* (with g-sensitivity of 6 parameters) and *imu21* - *imu24* (with g-sensitivity of 9 parameters). This means that the proposed system performance is less sensitive to g-sensitivity, no matter 0, 6 or 9 parameters are used.

We can also see that the terms in scale-misalignment for gyroscope $\mathbf{D}'_w$ for the BMI160 IMU are much larger than $\mathbf{D}'_a$ and $\mathbf{T}_g$. This indicates that the readings from gyroscope of BMI160 are very noisy and the calibration of $\mathbf{D}'_w$ dominates the performance of this VINS system. This fact is again confirmed by model *imu32*, which only calibrates the $\mathbf{D}'_w$ and can achieve similar results as full IMU model calibration (see bottom of Table 6).

## 11 More Validation and Investigation of Camera and IMU Calibration

We further evaluate the proposed self-calibration system on a custom made visual-inertial sensor rig (VI-Rig, shown in Figure 14) which contains multiple IMU and camera sensors to facilitate

---

[1]https://github.com/ethz-asl/kalibr/wiki/Multi-IMU-and-IMU-intrinsic-calibration

Figure 14: Visual-Inertial Sensor Rig contains a MicroStrain GX5-25 IMU, MicroStrain GX5-35, Xsens MTi 100, FLIR Blackfly camera and RealSense T265 tracking camera. The RealSense T265 tracking camera contains an integrated IMU and a fisheye stereo camera.

Table 7: Average processing time for each image (including propagation and update) for the proposed system with (w/) and without (w/o) online calibration (unit: second) on the 10 datasets collected with VI-Rig. The time increase (0.0036s in average) for online calibration is negligible compared to no calibration.

| Algorithm | Data-1 | Data-2 | Data-3 | Data-4 | Data-5 | Data-6 | Data-7 | Data-8 | Data-9 | Data-10 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| w/ online calib | 0.0227 | 0.0231 | 0.0228 | 0.0223 | 0.0229 | 0.0220 | 0.0214 | 0.0228 | 0.0223 | 0.0215 | 0.0224 |
| w/o online calib | 0.0187 | 0.0191 | 0.0189 | 0.0190 | 0.0190 | 0.0185 | 0.0186 | 0.0186 | 0.0189 | 0.0190 | 0.0188 |

an investigation into how they individually impact overall performance. Specifically, it contains a MicroStrain GX5-25, MicroStrain GX5-35, Xsens MTi 100, FLIR blackfly camera and RealSense T265 tracking camera which contains an integrated BMI055 IMU along with a fisheye stereo camera. Here we note that all cameras used are not rolling shutter to ensure fair comparison against the baseline Kalibr [38] which only supports IMU-camera calibration with global shutter cameras. In total 10 datasets were collected of an April tag board on which both the proposed system and the Kalibr calibration toolbox were run to report repeatbility statistics and expected real-world performance of both systems. During data collection, all 6-axis motion of VI-Rig were excited to avoid degenerate motions for calibration parameters.

## 11.1 Visual Front-End and Initial Conditions

To provide a fair comparison, we modified the front-end of the proposed system to directly use the same April tag detection as Kalibr and to only use such tags during estimation. Additionally, while the proposed system was only run with one of the four IMUs and either the Blackfly or left T265 Realsense camera, Kalibr used all the available sensors to ensure the highest and most consistent performance (4 IMUs and 3 cameras). The *imu6* model is used during evaluation, which is equivalent to the *scale-misalignment* IMU model of Kalibr. We define the "ideal" IMU sensor intrinsics as $\mathbf{D}'_w = \mathbf{D}'_a = \mathbf{I}_3$, ${}^I_w\mathbf{R} = {}^I_a\mathbf{R} = \mathbf{I}_3$ and $\mathbf{T}_g = \mathbf{0}_3$ if factory or offline calibration has been pre-applied. Generally, these values are what the users expect for a ready-to-use IMU, and are the initial values that the proposed estimator starts from. The quality of each IMU can be evaluated by how close the converged calibrated values are to these "ideal" values.
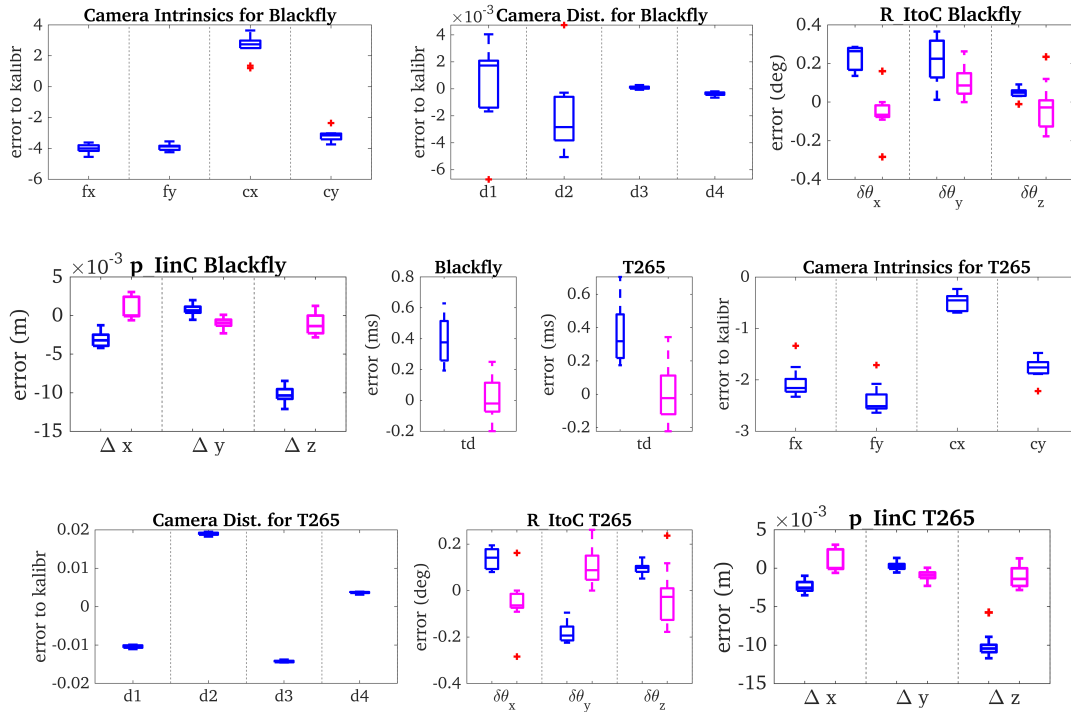
Figure 15: Comparison of the proposed method and Kalibr relative to a baseline Kalibr value. The boxplots show the final converged value of both methods, while for camera intrinsic only the proposed is reported since Kalibr fixes this during optimization. Kalibr (magenta, right in each group) was run with all cameras and IMUs available over 10 datasets, while the proposed system (blue, left) was run with either the Blackfly camera or left T265 fisheye and the MicroStrain GX-25 IMU resulting in 10 runs for each.



Figure 16: Comparison of the proposed method with *imu6* and Kalibr relative to the "ideal" sensor intrinsics. The boxplots show the final converged value of both methods. Kalibr (magenta, right in each group) was run with all cameras and IMUs available over 10 datasets, while the proposed system (blue, left) was run with either the Blackfly camera or left T265 fisheye resulting in 20 runs.
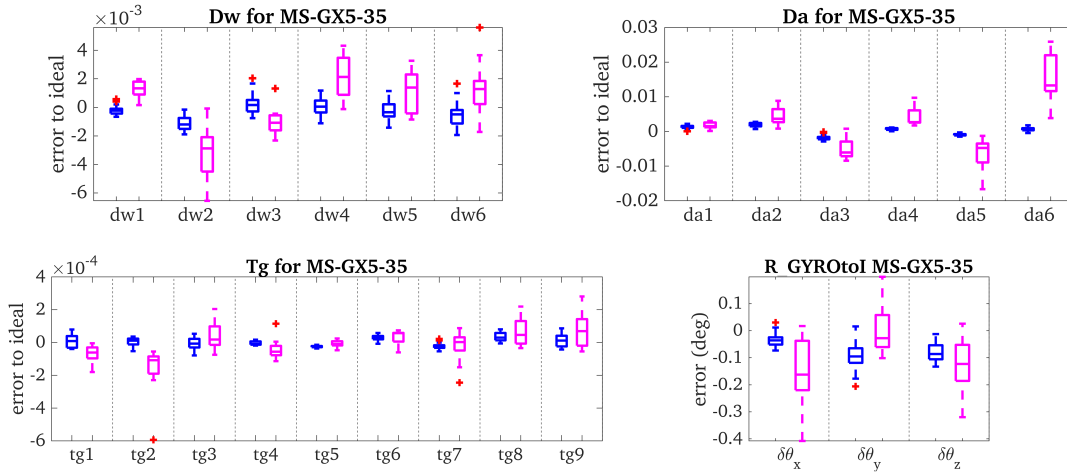
Figure 17: Comparison of the proposed method with *imu6* and Kalibr relative to the "ideal" sensor intrinsics. The boxplots show the final converged value of both methods. Kalibr (magenta, right in each group) was run with all cameras and IMUs available over 10 datasets, while the proposed system (blue, left) was run with either the Blackfly camera or left T265 fisheye resulting in 20 runs.
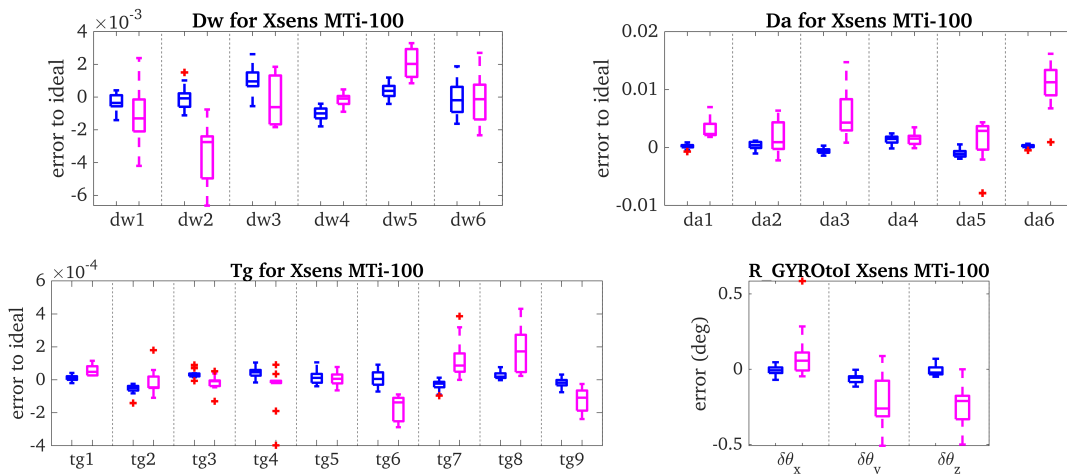


Figure 18: Comparison of the proposed method with *imu6* and Kalibr relative to the "ideal" sensor intrinsics. The boxplots show the final converged value of both methods. Kalibr (magenta, right in each group) was run with all cameras and IMUs available over 10 datasets, while the proposed system (blue, left) was run with either the Blackfly camera or left T265 fisheye resulting in 20 runs.
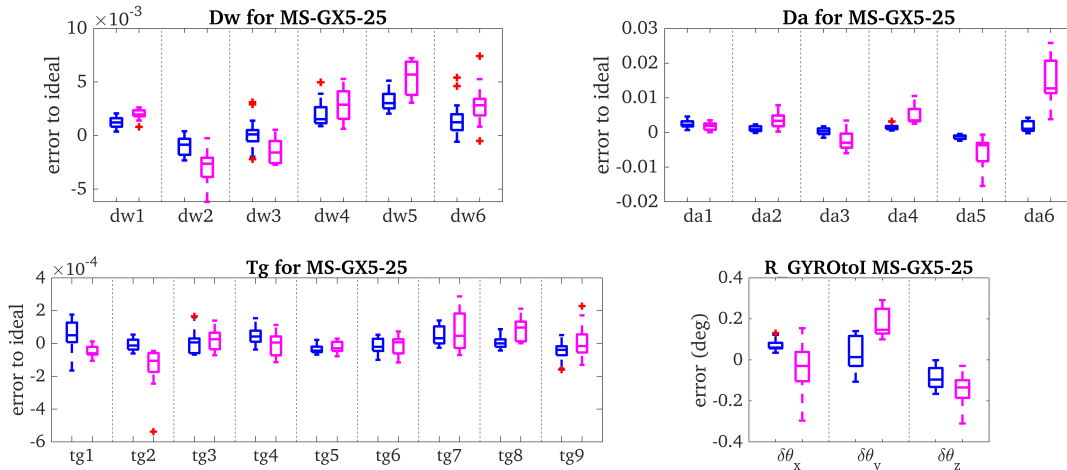
Figure 19: Comparison of the proposed method with *imu6* and Kalibr relative to the "identity" sensor intrinsics. The boxplots show the final converged value of both methods. Kalibr (magenta, right in each group) was run with all cameras and IMUs available over 10 datasets, while the proposed system (blue, left) was run with either the Blackfly camera or left T265 fisheye resulting in 20 runs.
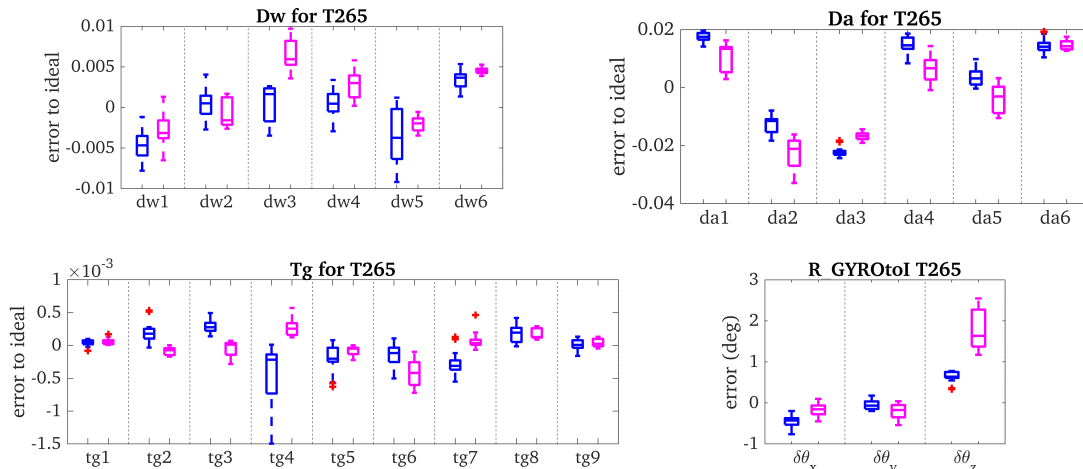
## 11.2    IMU-Camera Spatiotemporal Extrinsics and Intrinsics

We first investigate the convergence of the IMU-camera extrinsics and temporal parameters along with the camera intrinsics of the proposed system. The results shown in Figure 15 demonstrate that the proposed system is able to calibrate the spatial and temporal parameters with both high repeatability and accuracy relative to the offline Kalibr calibration baseline. Additionally shown is the convergence of camera intrinsics estimated by the proposed algorithm relative to the Kalibr static calibration results which are fixed during their IMU-camera calibration. Although the camera intrinsic estimates of blackfly and T265 camera have a few deviations compared to the reference values from Kalibr, the proposed system has very good convergence and high repeatability (the groundtruth is not known here).

## 11.3    IMU Intrinsic Parameters

The calibration results are summarized as boxplots shown in Figure 16 - 19 for the MicroStrain MS-GX5-25, MS-GX5-35, Xsens MTi-100 and T265 IMU, respectively. As shown, the calibration errors of the proposed system are quite close to the results of Kalibr, and the estimate differences of $\mathbf{D}'_w$, $\mathbf{D}'_a$ and $\mathbf{T}_g$ are around 1e-3, 1e-1 and 1e-4, respectively, for MS-GX5-25, MX-GX5-35 and Xsens MTi-100. Additionally, our proposed algorithm demonstrates better repeatability as the calibration errors have smaller variances and less outliers. In general, we can discuss the following results concerning the IMUs presented throughout the paper (see Figure 13 and 16 - 19):

- The MicroStrain MS-GX5-25, MS-GX5-35 and Xsens MTi-100 IMU are more close to "ideal" IMU than T265 IMU and BMI160 IMU. This is reasonable since both the MicroStrain and Xsens IMU are more expensive high-end IMUs with likely more sophisticate out-of-factory calibration than T265 IMU and BMI 160 IMU.

- For each IMU, the g-sensitivity terms are, in general, one or two orders smaller than the other terms of the IMU intrinsic model. This suggests that the g-sensitivity should not have

Table 8: Absolute Trajectory Error (ATE) on EuRoC MAV Vicon room sequences (with units degrees/meters).

| IMU Model | V1_01_easy | V1_02_medium | V1_03_difficult | V2_01_easy | V2_02_medium | V2_03_difficult | Average |
|---|---|---|---|---|---|---|---|
| imu0 | 0.657 / 0.043 | 1.805 / 0.060 | 2.437 / 0.069 | 0.869 / 0.109 | 1.373 / 0.080 | 1.277 / 0.180 | 1.403 / 0.090 |
| imu1 | 0.601 / 0.055 | 1.924 / 0.065 | 2.334 / 0.073 | 1.201 / 0.115 | 1.342 / 0.086 | 1.710 / 0.168 | 1.519 / 0.094 |
| imu2 | 0.552 / 0.054 | 1.990 / 0.062 | 2.197 / 0.083 | 0.960 / 0.107 | 1.453 / 0.085 | 1.666 / 0.216 | 1.470 / 0.101 |
| imu3 | 0.606 / 0.055 | 1.905 / 0.065 | 2.359 / 0.073 | 1.180 / 0.114 | 1.335 / 0.088 | 1.640 / 0.167 | 1.504 / 0.094 |
| imu4 | 0.569 / 0.056 | 1.969 / 0.069 | 2.165 / 0.076 | 0.846 / 0.127 | 1.636 / 0.094 | 1.577 / 0.195 | 1.461 / 0.103 |

    significant effects on system performance. This is likely due to the handheld motion of the platform and levels of achievable acceleration magnitudes.

- The BMI160 IMU (Figure 13), has a much more significant gyroscope calibration, $\mathbf{D}'_w$, compared to its accelerometer calibration and other IMUs. Thus the BMI160 can see large accuracy gains from only calibrating $\mathbf{D}'_w$, while for other IMUs, the calibration of $\mathbf{D}'_a$ should be more impactful.

Hence, these results validate the accuracy and consistency of the IMU intrinsic calibration of the proposed *online real-time* algorithm, which outputs comparable calibration results to Kalibr's offline calibration procedures.

## 11.4 Timing Evaluation

We also evaluate the running time for the proposed system with and without online sensor calibration shown in Table 7. We use the 10 datasets recorded with the VI-Rig and only use the measurements from MicroStrain GX5-25 IMU and the left camera of T265 for evaluation. In order to get more realistic timing evacuation, no April tags are detected and only the natural features tracked from images are used. We track 200 features from each image and keep at most 30 SLAM point features in the state vector with a sliding window of 20 clones. The averaged execution time for processing each coming image (including propagation and update) is recorded (shown in Table 7). The average execution time of the proposed system with online calibration is 0.0224s, which shows negligible increases than 0.0188s, which is the average running time without online calibration.

## 12 Real-World Degenerate Motion Demonstration and Analysis

Next, we evaluate the proposed system on a collection of real-world datasets which exhibit varying degrees of degenerate motions. Specifically, we evaluate on the EuRoc MAV dataset [72] which has under-actuated MAV with weakly excited acceleration and approximately 1-axis rotation. We also evaluate on the KAIST complex urban dataset [73] which is a planar autonomous vehicle dataset with relatively constant velocity throughout. We note that we do not estimate the g-sensitivity since it has been demonstrated in the preceding sections that it is not significant for VINS performance, and test only the *imu1 - imu4* models.

### 12.1 EuRoC MAV: Under-Actuated Motion

The EuRoC MAV dataset [72] contains a series of trajectories from a MAV and provides 20 Hz grayscale stereo images, 200 Hz inertial readings, and an external groundtruth pose from a motion

Figure 20: Simulation results for $\mathbf{D}_a$ of the proposed system evaluated with groundtruth *tum_room1* (left) and *EuRoc V1_02* (right) trajectories using *imu2* and *radtan*. 3 sigma bounds (dotted lines) and estimation errors (solid lines) for four different runs (different colors) with different realization of the measurement noise and initial perturbations are drawn. The *tum_room1* estimation errors and $3\sigma$ bounds converge nicely, while due to lack of motion excitation, the convergence of $\mathbf{D}_a$, especially $d_{a4}$, $d_{a5}$ and $d_{a6}$, for the *EuRoc V1_02* is poor.
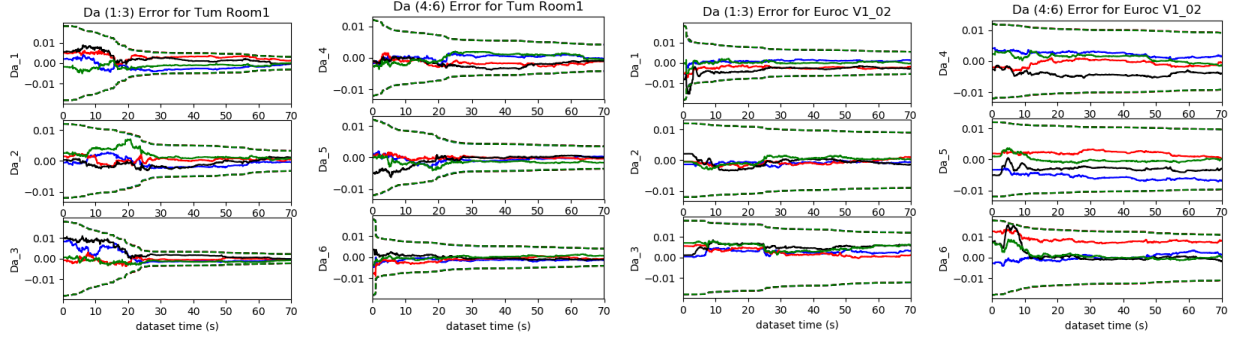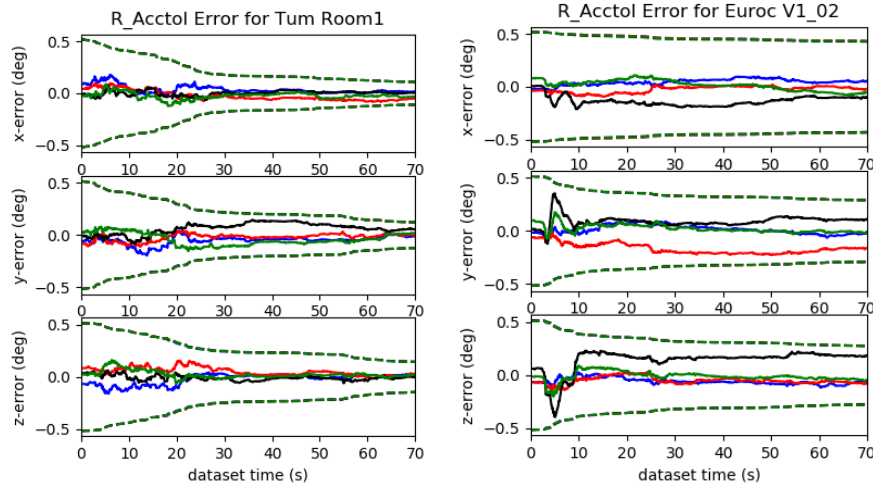


Figure 21: Simulation results for $_a^I\mathbf{R}$ of the proposed system evaluated with groundtruth *tum_room1* and *EuRoc V1_02* trajectories using *imu2* and *radtan*. With *tum_room1* trajectory, the estimation errors and $3\sigma$ bounds of $_a^I\mathbf{R}$ converge nicely, while the convergence is poor on the *EuRoc V1_02*.
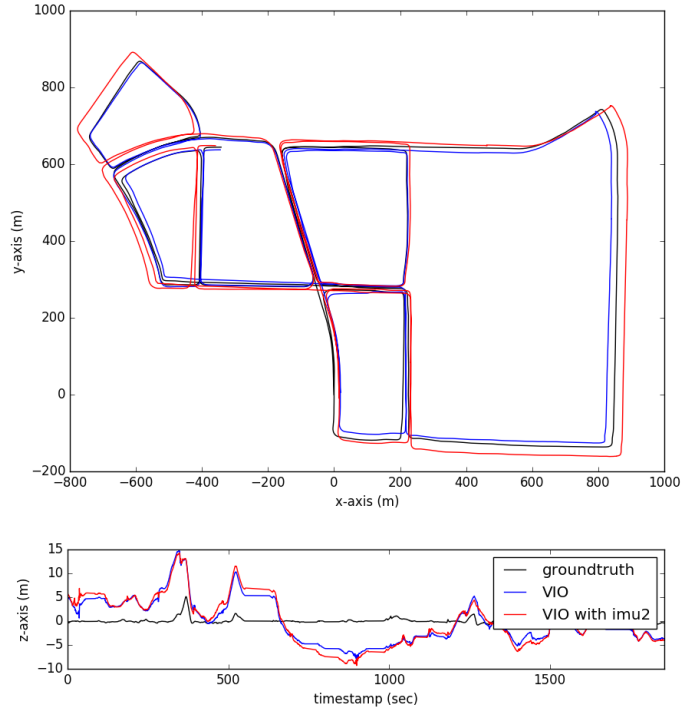
Figure 22: Trajectory plots for the KAIST Urban 39 dataset 10km in total length. Figure is best seen in color.

capture system. The proposed estimator is run with just the left camera on each of the Vicon room datasets and report the results in Table 8. Here we can see that not performing IMU intrinsic calibration, *imu0* model, outperforms the methods which additionally estimate the IMU intrinsics. This makes sense since the IMU intrinsics suffer from a large number of degenerate motions caused with constant local angular velocity and linear acceleration which can be expected for the MAV platform. Additionally, we believe that this is specifically caused by the MAV being unable to fully excite its 6DoF motion for a given small time interval and thus undergoes (nearly) degenerate motions locally throughout the whole trajectory, hurting the sliding-window filter. We can see that for more dynamic datasets, such as V2_03_difficult, there are still some partial improvements in accuracy possibly due to the more dynamic motion exhibited.

In order to verify our reasoning that the accuracy loss is caused by degenerate motions, we use the groundtruth trajectories of *tum_room1* with full 6DoF motion and *EuRoc V1_02* to simulate synthetic inertial and feature bearing measurements (see Section 9 on how we perform simulation) and evaluate our system with these simulated data. Figure 20 and 21 shows four different run with estimation errors and $3\sigma$ bounds for $\mathbf{D}_a$ and $_a^I\mathbf{R}$. It is clear that the motion of sensor on the *EuRoc V1_02* trajectory (right) is mildly excited and the acceleration readings are varying very slowly within the local window, causing poor convergence of the $\mathbf{D}_a$ and $_a^I\mathbf{R}$ with relatively flat $3\sigma$ bounds as compared to the *tum_room1* (left). This verifies that the online IMU intrinsic calibration will benefit VINS with fully-excited motion (e.g., the *tum_room1* trajectory) and might not be a good option for under-actuated motions such as the *EuRoc V1_02* trajectory.
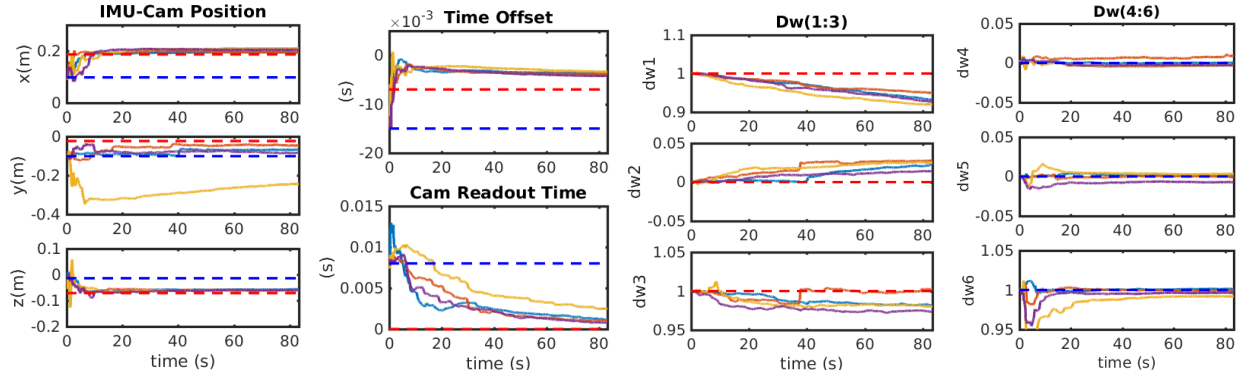
Figure 23: Calibration results (colored solid lines) for $^{C}\mathbf{p}_I$, $t_d$, $t_r$ and $\mathbf{D}_w$ of the proposed system evaluated with four VI-Rig planar motion datasets using *imu2* and *equi-dist*. Red and blue dotted lines denote the reference value from Kalibr and initial (perturbed) values, respectively. Colored solid line represents the estimated calibration parameters during online calibration for each dataset. All the temporal calibration, $dw_4$, $dw_5$ and $dw_6$ can converge well to the reference values, while the y component of $^{C}\mathbf{p}_I$, $dw_1$, $dw_2$ and $dw_3$ diverges.
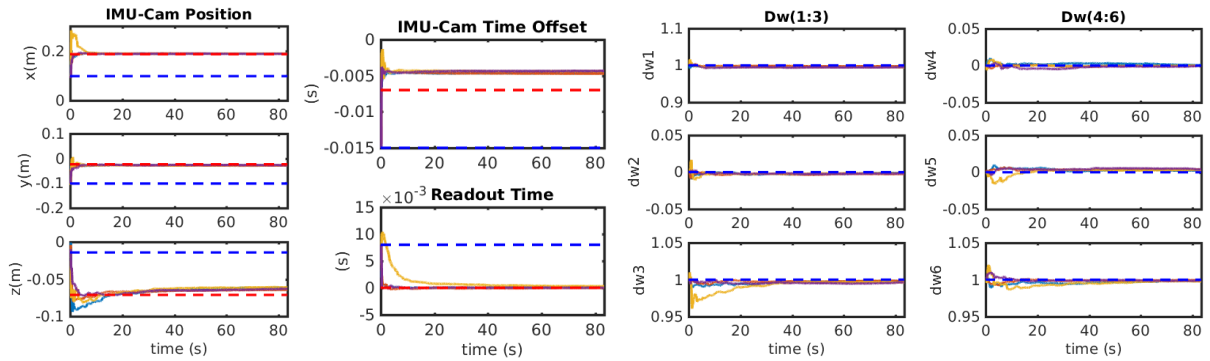


Figure 24: Calibration results (colored solid lines) for $^{C}\mathbf{p}_I$, $t_d$, $t_r$ and $\mathbf{D}_w$ of the proposed system evaluated with four 3D-motion datasets from Section 11 using *imu2* and *equidist*. Red and blue dotted lines denote the reference value from Kalibr and initial (perturbed) values, respectively. Colored solid line represents the estimated calibration parameters during online calibration for each dataset. All the temporal calibration, $^{C}\mathbf{p}_I$, and $\mathbf{D_w}$ can converge well to the reference values.

## 12.2  KAIST Complex Urban: Planar Motion

Next we evaluate on the KAIST Complex Urban dataset [73], which provides stereo grayscale images at 10 Hz, 200 Hz inertial readings, and a pseudo-groundtruth from an offline optimization method with the RTK GPS. As in the Table 2 and 3, this dataset contains planar motion and, in the case of *imu2*, 6 parameters are unobservable. We use the stereo camera pair along with the IMU to remove the scale ambiguity for monocular VINS caused by constant local acceleration [66, 67].

We plot Urban 39 trajectory estimates of the proposed system with and without IMU intrinsics in Figure 22. It can be seen that the system with IMU intrinsic calibration has larger drift compared to the one without. When looking at the absolute trajectory error (ATE) in respect to the dataset's groundtruth, the error of the standard VIO, is 1.58 degrees with 13.03 meters (0.12%), while with *imu2* online IMU intrinsic calibration it is 1.41 degrees with 23.13 meters (0.22%). We propose that this larger error is due to the introduced unobservable directions in online IMU intrinsic calibration when experiencing planar motion.

## 12.3  VI-Rig Planar Motion Datasets

Lastly, we evaluate on 4 datasets collected with VI-Rig (shown in Figure 14) under planar motion. In this evaluation, Microstrain GX5-25 and the left camera of T265 are used. When collecting data, we put the VI-Vig on a chair and only move the chairs in the ground plane to make sure VI-Vig is performing planar motion with global yaw as rotation axis, which is also the y-axis (pointing downward) of the camera. We calibrate all calibration parameters using *imu2* and *equidist* when running the system. Since T265 is a global shutter camera, the readout time is 0.00s.

The calibration results for the translation parameters ${}^C\mathbf{p}_I$, time offset $t_d$, readout time $t_r$ and the $\mathbf{D}w$ are shown in Figure 23. All the temporal calibration can converge well to the reference values based on offline calibration results of Kalibr. Note that $t_r$ converges to almost 0s as expected and $t_d$ converges from 0.015s to 0.005s with reference values as 0.007s. The final estimation errors are around 0.002s, which is pretty small. While the x and z components of ${}^C\mathbf{p}_I$ can also converge well to the reference values with small standard deviations (smaller than 0.4cm), the y component diverges with estimation errors more than 5cm and the standard deviation reach 3cm since it is along the rotation axis of the camera and hence, unobservable. Since the system has only yaw rotation for the IMU sensor, the $dw_1$, $dw_2$ and $dw_3$ are also unobservable (see Table 2), and their calibration results diverge a lot compared to those of $dw_4$, $dw_5$ and $dw_6$. This result verifies our degenerate motion analysis for IMU-camera and IMU intrinsic calibration. As comparison, we also plot the online calibration results of the proposed system running on another four datasets from Section 11 with fully excited motions in Figure 24. We use the same scale to plot the results for both Figure 23 and 24. It is clear that all these calibration parameters ($t_r$, $t_d$, ${}^C\mathbf{p}_I$ and $\mathbf{D}_w$) can converge better in fully excited motions than planar motion.

## 13  Discussion: Online Self-Calibration?

As learnt from the preceding extensive Monte-Carlo simulations and real world experiments, we highly recommend online self-calibration for VINS especially in the following scenarios:

- Poor calibration priors are provided.

- Low-end IMUs or cameras are used.

- RS cameras are used.

- The system undergoes fully-excited motions.

Specifically, as shown in Table 5 (simulation) and Figures 11 and 12 (TUM RS VIO Datasets), if the system starts with imperfect calibration, the system without online self-calibration is highly likely to fail, clearly demonstrated by Figure 5. But online calibration can greatly improve the system robustness and accuracy. From Figure 12, we see that online calibration for the low-end IMU (Bosch BMI160) and RS readout time is necessary, which can improve the system performance greatly. Based on the results from the EuRoC MAV dataset (Table 8) and KAIST datasets (Figure 22), online calibration, especially IMU intrinsic calibration, can hurt the system performance when the system undergoes underactuated motions.

Based on our analysis on the degenerate motions for these calibration parameters, we can recommend what calibration parameters should be calibrated with different motion types:

- Fully excited 3D motion: all calibration parameters.

- Mildly excited 3D motion: IMU-camera spatial-temporal calibration, readout time, and camera intrinsics.

- Under-actuated motions: readout time and camera intrinsics.

As shown in our degenerate motion analysis, there are a large number of motion types that prohibit accurate calibration of the IMU intrinsics and IMU-camera spatial calibration, while the camera intrinsics and IMU-camera temporal calibration are more robust to different motions. More importantly, in the most commonly-seen motion cases of aerial and ground vehicles, there is usually at least one unobservable direction due to these robots traveling with either underactuated 3D or planar motion. The impact on performance was shown with the EuRoC MAV and KAIST Urban datasets (Table 8 and Figure 22) where the use of online IMU calibration may hurt the estimator.

Due to the high likelihood of experiencing degenerate motions for some periods of time, solely based on our analysis and results, we do *not* recommend performing online IMU intrinsic and IMU-camera spatial calibration during real-time operations for most underactuated motions (e.g., planar motion and one-axis rotation for most ground vehicles). The exception to this is the handheld cases (e.g., mobile AR/VR), which often exhibit full 6DoF motions and thus is recommended to perform online calibration to improve estimation accuracy, especially when low-end IMUs or RS cameras are used. For both of these applications, we do recommend using an offline batch optimization to obtain an accurate initial calibration guess for the filter and/or treat the calibration parameters (especially intrinsics) as "true" if one knows they are going to experience degenerate motions. For online IMU intrinsic calibration, it is not necessary to calibrate the full IMU model and instead one may calibrate only the dominating parameters in the inertial models (e.g., $\mathbf{D}_w$ for BMI160 IMU or $\mathbf{D}_a$ for MicroStrain, Xsens and T265 IMUs).

# 14    Conclusions and Future Work

In this paper, we address the problem of online full-parameter self-calibration for visual-inertial navigation to achieve accurate and robust performance. We first investigate different IMU intrinsic model variants which contain scale correction, axis misalignment and g-sensitivity. These model variants can cover most used inertial models in practice. We also introduce the full visual measurement model which contains IMU-camera spatial-temporal parameters including rolling shutter readout time. After computing the state transition matrix and measurements Jacobian regarding

the state containing full calibration parameters, we perform observability analysis based on the linearized VINS system and show that with full-parameter calibration it still has only 4 unobservable directions, which relate to global yaw and global translation. All the calibration parameters of VINS are observable given fully excited motions.

Based on the observability analysis, we, for the first time, have identified basic degenerate motion patterns for IMU/camera intrinsics, and any combination of these degenerate motions will still be unobservable directions. Extensive validation on simulated and real-world datasets are performed to verify both the observability and degenerate motion analysis. We also show that online self-calibration can improve the robustness and accuracy of VINS. As shown through our experiments, online IMU intrinsic calibration is risky due to its dependence on the motion profile to ensure observability. In the case of autonomous (ground) vehicles, most trajectories have degenerate motions, thus resulting in *not* recommending online calibration of IMU intrinsics for robots with underactuated motions. In the case of handheld motion, however we found that the estimation of calibration parameters improved performance as expected.

In the future, we will investigate a complete degenerate motion analysis for multi-visual-inertial system. In addition, robust algorithms to perform online calibration under degenerate motions will also be studied.

# Appendix A: Appendix A: IMU Intrinsic Jacobians

In the following derivations, we will compute the Jacobians for all the variables that might appear in the IMU models, including scale/axis correction for gyroscope $\mathbf{D}_w$ (6 parameters), scale/axis correction for accelerometer $\mathbf{D}_a$ (6 parameters), rotation from gyroscope to IMU frame $_w^I\mathbf{R}$, rotation from accelerometer to IMU frame $_a^I\mathbf{R}$ and g-sensitivity $\mathbf{T}_g$ (9 parameters). We repeat the corrected IMU readings for easier derivation:

$$^I\boldsymbol{\omega} = {}_w^I\mathbf{R}\mathbf{D}_w \left( {}^w\boldsymbol{\omega}_m - \mathbf{T}_g{}^I\mathbf{a} - \mathbf{b}_g - \mathbf{n}_g \right) \tag{96}$$

$$^I\mathbf{a} = {}_a^I\mathbf{R}\mathbf{D}_a \left( {}^a\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a \right) \tag{97}$$

To simplify the derivations, we define $^I\hat{\mathbf{a}}$ and $^I\tilde{\mathbf{a}}$ as:

$$^I\mathbf{a} \simeq {}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a \left( {}^a\mathbf{a}_m - \hat{\mathbf{b}}_a \right) + {}_a^I\hat{\mathbf{R}}\mathbf{H}_{Da}\tilde{\mathbf{x}}_{Da} + \lfloor {}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a \left( {}^a\mathbf{a}_m - \hat{\mathbf{b}}_a \right) \rfloor \delta\boldsymbol{\theta}_{Ia} - {}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a\tilde{\mathbf{b}}_a - {}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a\mathbf{n}_a$$

$$^I\hat{\mathbf{a}} = {}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a \left( {}^a\mathbf{a}_m - \hat{\mathbf{b}}_a \right)$$

$$^I\tilde{\mathbf{a}} = {}_a^I\hat{\mathbf{R}}\mathbf{H}_{Da}\tilde{\mathbf{x}}_{Da} + \lfloor {}^I\hat{\mathbf{a}} \rfloor \delta\boldsymbol{\theta}_{Ia} - {}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a\tilde{\mathbf{b}}_a - {}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a\mathbf{n}_a$$

We define $^I\hat{\boldsymbol{\omega}}$ and $^I\tilde{\boldsymbol{\omega}}$ as:

$$^I\boldsymbol{\omega} = {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w \left( {}^w\boldsymbol{\omega}_m - \hat{\mathbf{T}}_g{}^I\hat{\mathbf{a}} - \hat{\mathbf{b}}_g \right) + \lfloor {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w \left( {}^w\boldsymbol{\omega}_m - \hat{\mathbf{T}}_g{}^I\hat{\mathbf{a}} - \hat{\mathbf{b}}_g \right) \rfloor \delta\boldsymbol{\theta}_{Iw}$$

$$\qquad + {}_w^I\hat{\mathbf{R}}\mathbf{H}_{Dw}\tilde{\mathbf{x}}_{Dw} - {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w \left( \hat{\mathbf{T}}_g{}^I\tilde{\mathbf{a}} + \mathbf{H}_{Tg}\tilde{\mathbf{x}}_{Tg} \right) - {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w \left( \tilde{\mathbf{b}}_g + \mathbf{n}_g \right)$$

$$^I\hat{\boldsymbol{\omega}} = {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w \left( {}^w\boldsymbol{\omega}_m - \hat{\mathbf{T}}_g{}^I\hat{\mathbf{a}} - \hat{\mathbf{b}}_g \right)$$

$$^I\tilde{\boldsymbol{\omega}} = -{}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w\tilde{\mathbf{b}}_g + {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g{}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a\tilde{\mathbf{b}}_a + {}_w^I\hat{\mathbf{R}}\mathbf{H}_{Dw}\tilde{\mathbf{x}}_{Dw} - {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g{}_a^I\hat{\mathbf{R}}\mathbf{H}_{Da}\tilde{\mathbf{x}}_{Da}$$

$$\qquad + \lfloor {}^I\hat{\boldsymbol{\omega}} \rfloor \delta\boldsymbol{\theta}_{Iw} - {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g\lfloor {}^I\hat{\mathbf{a}} \rfloor \delta\boldsymbol{\theta}_{Ia} - {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w\mathbf{H}_{Tg}\tilde{\mathbf{x}}_{Tg} - {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w\mathbf{n}_g + {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g{}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a\mathbf{n}_a$$

where we have:

$$\mathbf{H}_{Dw} = \begin{bmatrix} {}^w\hat{w}_1\mathbf{e}_1 & {}^w\hat{w}_2\mathbf{e}_1 & {}^w\hat{w}_2\mathbf{e}_2 & {}^w\hat{w}_3\mathbf{I}_3 \end{bmatrix} \tag{98}$$

$$\mathbf{H}_{Da} = \begin{bmatrix} {}^a\hat{a}_1\mathbf{e}_1 & {}^a\hat{a}_2\mathbf{e}_1 & {}^a\hat{a}_2\mathbf{e}_2 & {}^a\hat{a}_3\mathbf{I}_3 \end{bmatrix} \tag{99}$$

$$\mathbf{H}_{Tg} = \begin{bmatrix} {}^I\hat{a}_1\mathbf{I}_3 & {}^I\hat{a}_2\mathbf{I}_3 & {}^I\hat{a}_3\mathbf{I}_3 \end{bmatrix} \tag{100}$$

By summarizing the above equations, we have:

$$\begin{bmatrix} {}^{I_k}\tilde{\boldsymbol{\omega}} \\ {}^{I_k}\tilde{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_b & \mathbf{H}_{in} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_b \\ \tilde{\mathbf{x}}_{in} \end{bmatrix} + \mathbf{H}_n \begin{bmatrix} \mathbf{n}_g \\ \mathbf{n}_a \end{bmatrix} \tag{101}$$

where we have defined:

$$\mathbf{H}_b = \mathbf{H}_n = \begin{bmatrix} -{}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w & {}_w^I\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g{}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a \\ \mathbf{0}_3 & -{}_a^I\hat{\mathbf{R}}\hat{\mathbf{D}}_a \end{bmatrix} \tag{102}$$

$$\mathbf{H}_{in} = \begin{bmatrix} \mathbf{H}_w & \mathbf{H}_a & \mathbf{H}_{Iw} & \mathbf{H}_{Ia} & \mathbf{H}_g \end{bmatrix} \tag{103}$$

with:

$$\mathbf{H}_w = \begin{bmatrix} {}^I_w\hat{\mathbf{R}}\mathbf{H}_{Dw} \\ \mathbf{0}_3 \end{bmatrix} \tag{104}$$

$$\mathbf{H}_a = \begin{bmatrix} -{}^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_{ga}{}^I_a\hat{\mathbf{R}}\mathbf{H}_{Da} \\ {}^I_a\hat{\mathbf{R}}\mathbf{H}_{Da} \end{bmatrix} \tag{105}$$

$$\mathbf{H}_{Iw} = \begin{bmatrix} \lfloor {}^I\hat{\boldsymbol{\omega}} \rfloor \\ \mathbf{0}_3 \end{bmatrix} \tag{106}$$

$$\mathbf{H}_{Ia} = \begin{bmatrix} -{}^I_a\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}\lfloor {}^I\hat{\mathbf{a}} \rfloor \\ \lfloor {}^I\hat{\mathbf{a}} \rfloor \end{bmatrix} \tag{107}$$

$$\mathbf{H}_g = \begin{bmatrix} -{}^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w\mathbf{H}_{Tg} \\ \mathbf{0}_3 \end{bmatrix} \tag{108}$$

Hence, $\Delta\mathbf{R}_k$ from Eq. (41), can be written as:

$$\Delta\mathbf{R}_k \simeq \exp\left({}^{I_k}\boldsymbol{\omega}\delta t_k\right) = \exp\left(({}^{I_k}\hat{\boldsymbol{\omega}} + {}^{I_k}\tilde{\boldsymbol{\omega}})\delta t_k\right) \exp\left(\Delta\hat{\boldsymbol{\theta}}_k\right) \exp\left(\mathbf{J}_r(\Delta\hat{\boldsymbol{\theta}}_k)^{I_k}\tilde{\boldsymbol{\omega}}\delta t_k\right) \tag{109}$$

where $\Delta\hat{\boldsymbol{\theta}}_k = {}^{I_k}\hat{\boldsymbol{\omega}}\delta t_k$.

$\Delta\mathbf{p}_k$ from Eq. (42) can be written as:

$$\begin{aligned}
\Delta\mathbf{p}_k &= \int_{t_k}^{t_{k+1}} \int_{t_k}^{s} {}^{I_k}_{I_\tau}\mathbf{R}^{I_\tau}\mathbf{a}\,d\tau ds \\
&\simeq \int_{t_k}^{t_{k+1}} \int_{t_k}^{s} \exp\left({}^{I_k}\boldsymbol{\omega}\delta\tau\right) {}^{I_k}\mathbf{a}\,d\tau ds \\
&= \int_{t_k}^{t_{k+1}} \int_{t_k}^{s} \exp\left(\left({}^{I_k}\hat{\boldsymbol{\omega}} + {}^{I_k}\tilde{\boldsymbol{\omega}}\right)\delta\tau\right) \left({}^{I_k}\hat{\mathbf{a}} + {}^{I_k}\tilde{\mathbf{a}}\right) d\tau ds \\
&\simeq \underbrace{\int_{t_k}^{t_{k+1}} \int_{t_k}^{s} \exp\left({}^{I_k}\hat{\boldsymbol{\omega}}\delta\tau\right) {}^{I_k}\hat{\mathbf{a}}\,d\tau ds}_{\Delta\hat{\mathbf{p}}_k} - \underbrace{\int_{t_k}^{t_{k+1}} \int_{t_k}^{s} \exp\left({}^{I_k}\hat{\boldsymbol{\omega}}\delta\tau\right) \lfloor {}^{I_k}\hat{\mathbf{a}} \rfloor \mathbf{J}_r({}^{I_k}\boldsymbol{\omega}\delta\tau)\delta\tau d\tau ds\, {}^{I_k}\tilde{\boldsymbol{\omega}}}_{\boldsymbol{\Xi}_4} \\
&\quad + \underbrace{\int_{t_k}^{t_{k+1}} \int_{t_k}^{s} \exp\left({}^{I_k}\hat{\boldsymbol{\omega}}\delta\tau\right) d\tau ds}_{\boldsymbol{\Xi}_2}\, {}^{I_k}\tilde{\mathbf{a}}
\end{aligned}$$

$\Delta \mathbf{v}_k$ from Eq. (43) can be written as:

$$\Delta \mathbf{v}_k = \int_{t_k}^{t_{k+1}} {}^{I_k}_{I_\tau}\mathbf{R}^{I_\tau}\mathbf{a}d\tau$$

$$\simeq \int_{t_k}^{t_{k+1}} \exp\left({}^{I_k}\boldsymbol{\omega}\delta\tau\right){}^{I_k}\mathbf{a}d\tau$$

$$= \int_{t_k}^{t_{k+1}} \exp\left(\left({}^{I_k}\hat{\boldsymbol{\omega}} + {}^{I_k}\tilde{\boldsymbol{\omega}}\right)\delta\tau\right)\left({}^{I_k}\hat{\mathbf{a}} + {}^{I_k}\tilde{\mathbf{a}}\right)d\tau$$

$$\simeq \underbrace{\int_{t_k}^{t_{k+1}} \exp\left({}^{I_k}\hat{\boldsymbol{\omega}}\delta\tau\right){}^{I_k}\hat{\mathbf{a}}d\tau}_{\Delta\hat{\mathbf{v}}_k} - \underbrace{\int_{t_k}^{t_{k+1}} \exp\left({}^{I_k}\hat{\boldsymbol{\omega}}\delta\tau\right)\lfloor{}^{I_k}\hat{\mathbf{a}}\rfloor\mathbf{J}_r({}^{I_k}\boldsymbol{\omega}\delta\tau)\delta\tau d\tau\,{}^{I_k}\tilde{\boldsymbol{\omega}}}_{\boldsymbol{\Xi}_3}$$

$$+ \underbrace{\int_{t_k}^{t_{k+1}} \exp\left({}^{I_k}\hat{\boldsymbol{\omega}}\delta\tau\right)d\tau\,{}^{I_k}\tilde{\mathbf{a}}}_{\boldsymbol{\Xi}_1}$$

By summarizing the above derivations, we have:

$$\Delta\mathbf{R}_k = \Delta\hat{\mathbf{R}}_k \exp\left(\mathbf{J}_r(\Delta\hat{\boldsymbol{\theta}}_k){}^{I_k}\tilde{\boldsymbol{\omega}}\delta t_k\right) \tag{110}$$

$$\Delta\mathbf{p}_k = \Delta\hat{\mathbf{p}}_k - \boldsymbol{\Xi}_4{}^{I_k}\tilde{\boldsymbol{\omega}} + \boldsymbol{\Xi}_2{}^{I_k}\tilde{\mathbf{a}} \tag{111}$$

$$\Delta\mathbf{v}_k = \Delta\hat{\mathbf{v}}_k - \boldsymbol{\Xi}_3{}^{I_k}\tilde{\boldsymbol{\omega}} + \boldsymbol{\Xi}_1{}^{I_k}\tilde{\mathbf{a}} \tag{112}$$

The linearized model for IMU dynamics can be written as:

$$\tilde{\mathbf{x}}_{n_{k+1}} = \boldsymbol{\Phi}_{nn}\tilde{\mathbf{x}}_{n_k} + \boldsymbol{\Phi}_{wa}\begin{bmatrix}{}^{I_k}\tilde{\boldsymbol{\omega}}\\{}^{I_k}\tilde{\mathbf{a}}\end{bmatrix} \tag{113}$$

where:

$$\boldsymbol{\Phi}_{nn} = \begin{bmatrix}\Delta\hat{\mathbf{R}}_k^\top & \mathbf{0}_3 & \mathbf{0}_3\\ -{}^G_{I_k}\hat{\mathbf{R}}\lfloor\Delta\hat{\mathbf{p}}_k\rfloor & \mathbf{I}_3 & \mathbf{I}_3\delta t_k\\ -{}^G_{I_k}\hat{\mathbf{R}}\lfloor\Delta\hat{\mathbf{v}}_k\rfloor & \mathbf{0}_3 & \mathbf{I}_3\end{bmatrix} \tag{114}$$

$$\boldsymbol{\Phi}_{wa} = \begin{bmatrix}\mathbf{J}_r(\Delta\hat{\boldsymbol{\theta}}_k)\delta t_k & \mathbf{0}_3\\ -{}^G_{I_k}\hat{\mathbf{R}}\boldsymbol{\Xi}_4 & {}^G_{I_k}\hat{\mathbf{R}}\boldsymbol{\Xi}_2\\ -{}^G_{I_k}\hat{\mathbf{R}}\boldsymbol{\Xi}_3 & {}^G_{I_k}\hat{\mathbf{R}}\boldsymbol{\Xi}_1\end{bmatrix} \tag{115}$$

By plugging Eq. (101) into Eq. (113) and adding biases, the overall linearized system for IMU state can be written as:

$$\tilde{\mathbf{x}}_{I_{k+1}} = \boldsymbol{\Phi}_{I(k+1,k)}\tilde{\mathbf{x}}_{I_k} + \mathbf{G}_k\mathbf{n}_{dI} \tag{116}$$

where $\mathbf{n}_{dI} = \begin{bmatrix}\mathbf{n}_{dg}^\top & \mathbf{n}_{da}^\top & \mathbf{n}_{dwg}^\top & \mathbf{n}_{dwa}^\top\end{bmatrix}^\top$ denotes the discretized IMU noises; $\boldsymbol{\Phi}_{I(k+1,k)}$ and $\mathbf{G}_k$ are computed as:

$$\boldsymbol{\Phi}_{I(k+1,k)} = \begin{bmatrix}\boldsymbol{\Phi}_{nn} & \boldsymbol{\Phi}_{wa}\mathbf{H}_b & \boldsymbol{\Phi}_{wa}\mathbf{H}_{in}\\ \mathbf{0}_{6\times9} & \mathbf{I}_6 & \mathbf{0}_{6\times m}\\ \mathbf{0}_{m\times9} & \mathbf{0}_{m\times6} & \mathbf{I}_m\end{bmatrix} \tag{117}$$

$$\mathbf{G}_k = \begin{bmatrix}\boldsymbol{\Phi}_{wa}\mathbf{H}_n & \mathbf{0}_{9\times6}\\ \mathbf{0}_6 & \mathbf{I}_6\delta t_k\\ \mathbf{0}_{m\times6} & \mathbf{0}_{m\times6}\end{bmatrix} \tag{118}$$

Note that $\mathbf{n}_{d*} \sim \mathcal{N}(\mathbf{0}, \frac{\sigma_*^2 \mathbf{I}_3}{\delta t_k})$ and hence the covariance for $\mathbf{n}_{dI}$ can be written as:

$$\mathbf{Q}_{dI} = \begin{bmatrix} \frac{\sigma_g^2}{\delta t_k} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \frac{\sigma_a^2}{\delta t_k} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \frac{\sigma_{wg}^2}{\delta t_k} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{\sigma_{wa}^2}{\delta t_k} \mathbf{I}_3 \end{bmatrix} \tag{119}$$

## Appendix B: Appendix B: Camera Jacobians

We will show the detailed derivations for the Jacobians shown in Eq. (74). The camera intrinsic Jacobians $\mathbf{H}_{Cin}$ can be written as:

$$\mathbf{H}_{Cin} = \begin{bmatrix} \frac{\partial \tilde{\mathbf{z}}_C}{\partial [\tilde{f}_u\ \tilde{f}_v\ \tilde{c}_u\ \tilde{c}_v]^\top} & \frac{\partial \tilde{\mathbf{z}}_C}{\partial [\tilde{k}_1\ \tilde{k}_2\ \tilde{p}_1\ \tilde{p}_2]^\top} \end{bmatrix} \tag{120}$$

$$\frac{\partial \tilde{\mathbf{z}}_C}{\partial [\tilde{f}_u\ \tilde{f}_v\ \tilde{c}_u\ \tilde{c}_v]^\top} = \begin{bmatrix} u_d & 0 & 1 & 0 \\ 0 & v_d & 0 & 1 \end{bmatrix} \tag{121}$$

$$\frac{\partial \tilde{\mathbf{z}}_C}{\partial [\tilde{k}_1\ \tilde{k}_2]^\top} = \begin{bmatrix} f_u u_n r^2 & f_u u_n r^4 \\ f_v v_n r^2 & f_v v_n r^4 \end{bmatrix} \tag{122}$$

$$\frac{\partial \tilde{\mathbf{z}}_C}{\partial [\tilde{p}_1\ \tilde{p}_2]^\top} = \begin{bmatrix} 2 f_u u_n v_n & f_u(r^2 + 2u_n^2) \\ f_v(r^2 + 2v_n^2) & 2 f_v u_n v_n \end{bmatrix} \tag{123}$$

We continue to compute $\frac{\partial \tilde{\mathbf{z}}_C}{\partial \tilde{\mathbf{z}}_n}$ and $\frac{\partial \tilde{\mathbf{z}}_n}{\partial^C \tilde{\mathbf{p}}_f}$ for $\mathbf{H}_{\mathbf{p}_f}$ within Eq. (74) as:

$$\frac{\partial \tilde{\mathbf{z}}_C}{\partial \tilde{\mathbf{z}}_n} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \tag{124}$$

$$\frac{\partial \tilde{\mathbf{z}}_C}{\partial^C \tilde{\mathbf{p}}_f} = \frac{1}{^C z_f^2} \begin{bmatrix} ^C z_f & 0 & -^C x_f \\ 0 & ^C z_f & -^C y_f \end{bmatrix} \tag{125}$$

with $\mathbf{h}_{11}$, $\mathbf{h}_{12}$, $\mathbf{h}_{21}$ and $\mathbf{h}_{22}$ defined as:

$$\mathbf{h}_{11} = f_u(d + 2k_1 u_n^2 + 4k_2 u_n^2 r^2 + 2p_1 v_n + 6p_2 u_n)$$
$$\mathbf{h}_{12} = f_u(2k_1 u_n v_n + 4k_2 u_n v_n r^2 + 2p_1 u_n + 2p_2 v_n)$$
$$\mathbf{h}_{21} = f_v(2k_1 u_n v_n + 4k_2 u_n v_n r^2 + 2p_1 u_n + 2p_2 v_n)$$
$$\mathbf{h}_{22} = f_v(d + 2k_1 v_n^2 + 4k_2 v_n^2 r^2 + 6p_1 v_n + 2p_2 u_n)$$

The Jacobians of $^C \mathbf{p}_f$ regarding to the IMU state $\mathbf{x}_I$ are written as:

$$\frac{\partial^C \tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}_I} = \begin{bmatrix} \frac{\partial^C \tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}_n} & \frac{\partial^C \tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}_b} & \frac{\partial^C \tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}_{in}} \end{bmatrix} \tag{126}$$

$$\frac{\partial^C \tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}_n} = {}_I^C\hat{\mathbf{R}} {}_G^I\hat{\mathbf{R}} \begin{bmatrix} \lfloor ^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_I \rfloor {}_I^G\hat{\mathbf{R}} & -\mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} \tag{127}$$

$$\frac{\partial^C \tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}_b} = \mathbf{0}_{3\times6}, \quad \frac{\partial^C \tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}_{in}} = \mathbf{0}_{3\times24} \tag{128}$$

The Jacobians of $^C\mathbf{p}_f$ regarding to the IMU-camera spatial-temporal calibration state $\mathbf{x}_{IC}$ are written as:

$$\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{\mathbf{x}}_{IC}} = \begin{bmatrix} \frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\delta\boldsymbol{\theta}_{IC}} & \frac{\partial^C\tilde{\mathbf{p}}_f}{\partial^C\tilde{\mathbf{p}}_I} & \frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{t}_d} & \frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{t}_r} \end{bmatrix} \tag{129}$$

$$\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\delta\boldsymbol{\theta}_{IC}} = \lfloor^C_I\hat{\mathbf{R}}^I_G\hat{\mathbf{R}}\left(^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_I\right)\rfloor \tag{130}$$

$$\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial^C\tilde{\mathbf{p}}_I} = \mathbf{I}_3 \tag{131}$$

$$\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{t}_d} = {}^C_I\hat{\mathbf{R}}^I_G\hat{\mathbf{R}}\left(\lfloor(^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_I)\rfloor^G_I\hat{\mathbf{R}}^I\hat{\boldsymbol{\omega}} - {}^G\hat{\mathbf{v}}_I\right) \tag{132}$$

$$\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{t}_r} = \frac{m}{M}\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{t}_d} \tag{133}$$

Note that when computing the Jacobians for $t_d$ and $t_r$, we are using the following linearization:

$$^G_{I(t)}\mathbf{R} \simeq {}^G_{I(\hat{t})}\hat{\mathbf{R}}\exp(\delta\boldsymbol{\theta}_I)\exp(^I\hat{\boldsymbol{\omega}}\tilde{t}_d + \frac{m}{M}{}^I\hat{\boldsymbol{\omega}}\tilde{t}_r) \tag{134}$$

$$^G\mathbf{p}_{I(t)} \simeq {}^G\hat{\mathbf{p}}_{I(\hat{t})} + {}^G\tilde{\mathbf{p}}_I + {}^G\hat{\mathbf{v}}_I\tilde{t}_d + \frac{m}{M}{}^G\hat{\mathbf{v}}_I\tilde{t}_r \tag{135}$$

The Jacobians of $^C\mathbf{p}_f$ regarding to the feature state $\mathbf{x}_f$ is written as:

$$\frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\tilde{\mathbf{x}}_f} = \frac{\partial^C\tilde{\mathbf{p}}_f}{\partial\delta^G\tilde{\mathbf{p}}_f} = {}^C_I\hat{\mathbf{R}}^I_G\hat{\mathbf{R}} \tag{136}$$

## Appendix C: Appendix C: Observability Matrix

We show the detailed derivations for $\mathbf{M}_n$, $\mathbf{M}_b$, $\mathbf{M}_{in}$, $\mathbf{M}_{IC}$, $\mathbf{M}_{Cin}$ and $\mathbf{M}_f$. The $\mathbf{M}_n$ is computed as:

$$\mathbf{M}_n = \mathbf{H}_{\mathbf{p}_f}{}^C_I\hat{\mathbf{R}}^{I_k}_G\hat{\mathbf{R}}\begin{bmatrix}\boldsymbol{\Gamma}_1 & \boldsymbol{\Gamma}_2 & \boldsymbol{\Gamma}_3\end{bmatrix} \tag{137}$$

with:

$$\boldsymbol{\Gamma}_1 = \lfloor^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_1} - {}^G\hat{\mathbf{v}}_{I_1}\delta t_k + \frac{1}{2}{}^G\mathbf{g}\delta t_k^2\rfloor^G_{I_1}\hat{\mathbf{R}}$$

$$\boldsymbol{\Gamma}_2 = -\mathbf{I}_3$$

$$\boldsymbol{\Gamma}_3 = -\mathbf{I}_3\delta t_k$$

The $\mathbf{M}_b$ is computed as:

$$\mathbf{M}_b = \mathbf{H}_{\mathbf{p}_f}{}^C_I\hat{\mathbf{R}}^{I_k}_G\hat{\mathbf{R}}\begin{bmatrix}\boldsymbol{\Gamma}_4 & \boldsymbol{\Gamma}_5\end{bmatrix} \tag{138}$$

with:

$$\boldsymbol{\Gamma}_4 = -\left(\lfloor^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k}\rfloor^G_{I_k}\hat{\mathbf{R}}\mathbf{J}_r\left(\Delta\hat{\boldsymbol{\theta}}_k\right)\delta t_k + {}^G_{I_k}\hat{\mathbf{R}}\boldsymbol{\Xi}_4\right)^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w$$

$$\boldsymbol{\Gamma}_5 = \left(\lfloor^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k}\rfloor^G_{I_k}\hat{\mathbf{R}}\mathbf{J}_r\left(\Delta\hat{\boldsymbol{\theta}}_k\right)^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g\delta t_k + {}^G_{I_k}\hat{\mathbf{R}}\left(\boldsymbol{\Xi}_4{}^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g + \boldsymbol{\Xi}_2\right)\right)^I_a\hat{\mathbf{R}}\hat{\mathbf{D}}_a$$

The $\mathbf{M}_{in}$ can be computed as:

$$\mathbf{M}_{in} = \mathbf{H}_{\mathbf{p}_f I}{}^C_I \hat{\mathbf{R}}{}^{I_k}_G \hat{\mathbf{R}} \begin{bmatrix} \boldsymbol{\Gamma}_6 & \boldsymbol{\Gamma}_7 & \boldsymbol{\Gamma}_8 & \boldsymbol{\Gamma}_9 \end{bmatrix} \tag{139}$$

with:

$$\boldsymbol{\Gamma}_6 = \left( \lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}^G_{I_k}\hat{\mathbf{R}} \mathbf{J}_r \left( \Delta\hat{\boldsymbol{\theta}}_k \right) \delta t_k + {}^G_{I_k}\hat{\mathbf{R}}\boldsymbol{\Xi}_4 \right) \mathbf{H}_{Dw}$$

$$\boldsymbol{\Gamma}_7 = -\left( \lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}^G_{I_k}\hat{\mathbf{R}} \mathbf{J}_r \left( \Delta\hat{\boldsymbol{\theta}}_k \right) {}^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g \delta t_k + {}^G_{I_k}\hat{\mathbf{R}} \left( \boldsymbol{\Xi}_4 {}^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g + \boldsymbol{\Xi}_2 \right) \right) {}^I_a\hat{\mathbf{R}}\mathbf{H}_{Da}$$

$$\boldsymbol{\Gamma}_8 = -\left( \lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}^G_{I_k}\hat{\mathbf{R}} \mathbf{J}_r \left( \Delta\hat{\boldsymbol{\theta}}_k \right) {}^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g \delta t_k + {}^G_{I_k}\hat{\mathbf{R}}(\boldsymbol{\Xi}_4 {}^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w\hat{\mathbf{T}}_g + \boldsymbol{\Xi}_2) \right) \lfloor {}^{I_k}\hat{\mathbf{a}} \rfloor$$

$$\boldsymbol{\Gamma}_9 = -\left( \lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}^G_{I_k}\hat{\mathbf{R}} \mathbf{J}_r \left( \Delta\hat{\boldsymbol{\theta}}_k \right) \delta t_k + {}^G_{I_k}\hat{\mathbf{R}}\boldsymbol{\Xi}_4 \right) {}^I_w\hat{\mathbf{R}}\hat{\mathbf{D}}_w\mathbf{H}_{Tg}$$

The $\mathbf{M}_{IC}$ can be computed as:

$$\mathbf{M}_{IC} = \mathbf{H}_{\mathbf{p}_f I}{}^C_I \hat{\mathbf{R}}{}^{I_k}_G \hat{\mathbf{R}} \begin{bmatrix} \boldsymbol{\Gamma}_{10} & \boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12} & \boldsymbol{\Gamma}_{13} \end{bmatrix} \tag{140}$$

$$\boldsymbol{\Gamma}_{10} = \lfloor \left( {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \right) \rfloor {}^G_{I_k}\hat{\mathbf{R}}{}^I_C\hat{\mathbf{R}} \tag{141}$$

$$\boldsymbol{\Gamma}_{11} = {}^G_{I_k}\hat{\mathbf{R}}{}^I_C\hat{\mathbf{R}} \tag{142}$$

$$\boldsymbol{\Gamma}_{12} = \lfloor \left( {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \right) \rfloor {}^G_{I_k}\hat{\mathbf{R}}{}^{I_k}\hat{\boldsymbol{\omega}} - {}^G\hat{\mathbf{v}}_{I_k} \tag{143}$$

$$\boldsymbol{\Gamma}_{13} = \frac{m}{M}\boldsymbol{\Gamma}_{12} \tag{144}$$

The $\mathbf{M}_{Cin}$ and $\mathbf{M}_f$ can be written as:

$$\mathbf{M}_{Cin} = \mathbf{H}_{Cin} \tag{145}$$

$$\mathbf{M}_f = \mathbf{H}_{\mathbf{p}_f I}{}^C_I \hat{\mathbf{R}}{}^{I_k}_G \hat{\mathbf{R}} \tag{146}$$

## Appendix D: Appendix D: Proof of Lemma 1

For Eq. (77), we first verify $\mathcal{O}\mathbf{N} = \mathbf{0}$ as:

$$\Leftrightarrow \mathcal{O}_k\mathbf{N} = \mathbf{0}$$

$$\Leftrightarrow \left( \boldsymbol{\Gamma}_1 {}^{I_1}_G\hat{\mathbf{R}} - \boldsymbol{\Gamma}_2 \lfloor {}^G\hat{\mathbf{p}}_{I_1} \rfloor - \boldsymbol{\Gamma}_3 \lfloor {}^G\hat{\mathbf{v}}_{I_1} \rfloor - \lfloor {}^G\hat{\mathbf{p}}_f \rfloor \right) {}^G\mathbf{g} = 0$$

$$\Leftrightarrow \left( \lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_1} - {}^G\hat{\mathbf{v}}_{I_1}\delta t_k + \frac{1}{2}{}^G\mathbf{g}\delta t_k^2 \rfloor \right) {}^G\mathbf{g} + \left( \lfloor {}^G\hat{\mathbf{p}}_{I_1} \rfloor + \lfloor {}^G\hat{\mathbf{v}}_{I_1} \rfloor \delta t_k - \lfloor {}^G\hat{\mathbf{p}}_f \rfloor \right) {}^G\mathbf{g} = \mathbf{0}$$

Hence, we can conclude that the observability matrix $\mathcal{O}$ has at least 4 unobservable directions.

In the following, we will try to show that there are only 4 unobservable directions under general situations. With abusing of notion, we can rewrite the observability matrix as:

$$\mathcal{O} = \begin{bmatrix} \mathcal{O}_1^\top & \dots & \mathcal{O}_k^\top \end{bmatrix}^\top$$

$$= \begin{bmatrix} \mathbf{M}_{n,1} & \mathbf{M}_{b,1} & \mathbf{M}_{in,1} & \mathbf{M}_{IC,1} & \mathbf{M}_{Cin,1} & \mathbf{M}_{f,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{M}_{n,k} & \mathbf{M}_{b,k} & \mathbf{M}_{in,k} & \mathbf{M}_{IC,k} & \mathbf{M}_{Cin,k} & \mathbf{M}_{f,k} \end{bmatrix}$$

By adjusting the column of $\mathbf{M}_f$ in $\mathcal{O}$, we can get $\mathcal{O}'$:

$$\mathcal{O}' \triangleq \begin{bmatrix} \mathcal{O}_I \mid \mathcal{O}_{in} \mid \mathcal{O}_{IC} \mid \mathcal{O}_f \end{bmatrix}$$
$$\triangleq \left[ \begin{array}{ccc|c|c|c} \mathbf{M}_{n,1} & \mathbf{M}_{b,1} & \mathbf{M}_{f,1} & \mathbf{M}_{in,1} & \mathbf{M}_{IC,1} & \mathbf{M}_{Cin,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{M}_{n,k} & \mathbf{M}_{b,k} & \mathbf{M}_{f,k} & \mathbf{M}_{in,k} & \mathbf{M}_{IC,k} & \mathbf{M}_{Cin,k} \end{array} \right]$$

It is clear that the column rank of $\mathcal{O}'$ is the same as $\mathcal{O}$.

$\mathcal{O}_I$ corresponds to the IMU navigation state, IMU bias state and feature state. $\mathcal{O}_I$ is equivalent to the standard VINS observability matrix in [52] and it has null space of 4DoF.

$\mathcal{O}_{in}$ corresponds to the IMU intrinsic parameters. By checking the Eq. (139), it is clearly that $\mathcal{O}_{in}$ will be affected by time-varying ${}^w\boldsymbol{\omega}(t)$ (in $\mathbf{H}_{Dw}$), ${}^a\mathbf{a}(t)$ (in $\mathbf{H}_{Da}$) and ${}^I\mathbf{a}(t)$ (in $\lfloor{}^I\mathbf{a}\rfloor$ and $\mathbf{H}_{Tg}$). Under generate motions, $\mathcal{O}_{in}$ can be of full column rank.

$\mathcal{O}_{IC}$ corresponds to the IMU-camera spatial and temporal calibration parameters. By checking the Eq. (140), we can see that the $\mathcal{O}_{IC}$ is affected by the time-varying IMU pose $\{{}^I_G\mathbf{R}(t), {}^G\mathbf{p}_I(t)\}$ and the IMU kinematics $\{{}^I\boldsymbol{\omega}(t), {}^I\mathbf{v}(t)\}$. In addition, $\Gamma_{13}$ in $\mathbf{M}_{IC}$ are also affected by the point feature measurement through $\frac{m}{M}$, of which $m$ will change under general measurement assumptions. Hence, $\mathcal{O}_{IC}$ can be of full column rank with random motions.

$\mathcal{O}_{Cin}$ corresponds to the camera intrinsic parameters. It is clear that $\mathcal{O}_{Cin}$ is only affected the environmental structure and is of full column rank as long as $\{u_n, v_n\}$ varies in different image tracks.

Since $\mathcal{O}_{in}$, $\mathcal{O}_{IC}$ and $\mathcal{O}_{Cin}$ are affected by different system parameters, and under general motion conditions, $[\mathcal{O}_{in} \ \mathcal{O}_{IC} \ \mathcal{O}_{Cin}]$ is also of full column rank. Therefore, the column rank of $\mathcal{O}'$ is determined by $\mathcal{O}_I$. Since $\mathcal{O}_I$ has 4 DoF null space, the $\mathcal{O}'$ also has 4 DoF. Hence, we can conclude that $\mathcal{O}$ only has 4 DoF null space. We also verify this conclusion through simulation results shown in Fig 4.

## Appendix E: Appendix E: Null Space Proofs

In this section, we provide the verification for the null spaces listed in this paper.

### E.1: Verification of Lemma 2

For $\mathbf{N}_{w1}$, we have:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{w1} = \mathbf{0}$$
$$\Leftrightarrow \boldsymbol{\Gamma}_4 \hat{\mathbf{D}}_w^{-1}{}^I_w \hat{\mathbf{R}}^\top \mathbf{e}_1{}^w\omega_1 + \boldsymbol{\Gamma}_6 \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top = \mathbf{0}$$
$$\Leftrightarrow (\lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k}\rfloor {}^G_{I_k}\hat{\mathbf{R}}\mathbf{J}_r\left(\Delta\boldsymbol{\theta}_k\right)\delta t_k + {}^G_{I_k}\hat{\mathbf{R}}\boldsymbol{\Xi}_4)\left(-\mathbf{e}_1{}^w\omega_1 + \mathbf{e}_1{}^w\omega_1\right) = \mathbf{0}$$

For $\mathbf{N}_{w2}$, we have:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{w2} = \mathbf{0}$$
$$\Leftrightarrow \boldsymbol{\Gamma}_4 \hat{\mathbf{D}}_w^{-1}{}^I_w \hat{\mathbf{R}}^\top \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}{}^w\omega_2 + \boldsymbol{\Gamma}_6 \times \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^\top = \mathbf{0}$$
$$\Leftrightarrow (\lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k}\rfloor {}^G_{I_k}\hat{\mathbf{R}}\mathbf{J}_r\left(\Delta\boldsymbol{\theta}_k\right)\delta t_k + {}^G_{I_k}\hat{\mathbf{R}}\boldsymbol{\Xi}_4)\left(-\begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}{}^w\omega_2 + \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}{}^w\omega_2\right) = \mathbf{0}$$

For $\mathbf{N}_{w3}$, we have:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{w3} = \mathbf{0}$$

$$\Leftrightarrow \mathbf{\Gamma}_4 \hat{\mathbf{D}}_w^{-1}{}_w^I \hat{\mathbf{R}}^\top \mathbf{I}_3{}^w \omega_3 + \mathbf{\Gamma}_6 \times \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}^\top = \mathbf{0}$$

$$\Leftrightarrow \left( \lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}_{I_k}^G \hat{\mathbf{R}} \mathbf{J}_r\left(\Delta\boldsymbol{\theta}_k\right) \delta t_k + {}_{I_k}^G \hat{\mathbf{R}} \boldsymbol{\Xi}_4 \right) \left( -\mathbf{I}_3{}^w \omega_3 + \mathbf{I}_3{}^w \omega_3 \right) = \mathbf{0}$$

### E.2: Verification of Lemma <span style="color:red">3</span>

We first verify the first column of $\mathbf{N}_{a1}$:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{a1} \mathbf{e}_1 = \mathbf{0}$$

$$\Leftrightarrow \mathbf{\Gamma}_5 \hat{\mathbf{D}}_a^{-1} \mathbf{e}_1{}^a a_1 + \mathbf{\Gamma}_7 \times [1\ 0\ 0\ 0\ 0\ 0]^\top = \mathbf{0}$$

$$\Leftrightarrow \left( \lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}_{I_k}^G \hat{\mathbf{R}} \mathbf{J}_r\left(\Delta\boldsymbol{\theta}_k\right) {}_w^I \hat{\mathbf{R}} \hat{\mathbf{D}}_w \hat{\mathbf{T}}_g \delta t_k + {}_{I_k}^G \hat{\mathbf{R}} \left( \boldsymbol{\Xi}_4 {}_w^I \hat{\mathbf{R}} \hat{\mathbf{D}}_w \hat{\mathbf{T}}_g + \boldsymbol{\Xi}_2 \right) \right) {}_a^I \hat{\mathbf{R}} \left( \mathbf{e}_1{}^a a_1 - \mathbf{e}_1{}^a a_1 \right) = \mathbf{0}$$

We then verify the second column of $\mathbf{N}_{a1}$:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{a1} \mathbf{e}_2 = \mathbf{0}$$

$$\Leftrightarrow \mathbf{\Gamma}_5 \hat{\mathbf{D}}_a^{-1} \mathbf{e}_2 d_{a1}{}^a a_1 + \mathbf{\Gamma}_7 \times [0\ d_{a3}\ -d_{a2}\ d_{a5}\ -d_{a4}\ 0]^\top - \mathbf{\Gamma}_8 {}_a^I \hat{\mathbf{R}} \mathbf{e}_3 = \mathbf{0}$$

$$\Leftrightarrow \begin{bmatrix} 0 \\ d_{a1}{}^a a_1 \\ 0 \end{bmatrix} - \begin{bmatrix} d_{a3}{}^a a_2 + d_{a5}{}^a a_3 \\ -d_{a2}{}^a a_2 - d_{a4}{}^a a_3 \\ 0 \end{bmatrix} + \begin{bmatrix} d_{a3}{}^a a_2 + d_{a5}{}^a a_3 \\ -(d_{a1}{}^a a_1 + d_{a2}{}^a a_2 + d_{a4}{}^a a_3) \\ 0 \end{bmatrix} = \mathbf{0}$$

The third column of $\mathbf{N}_{a1}$ can be verified as:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{a1} \mathbf{e}_3 = \mathbf{0}$$

$$\Leftrightarrow \mathbf{\Gamma}_5 \hat{\mathbf{D}}_a^{-1} \mathbf{e}_3 d_{a1} d_{a3}{}^a a_1 + \mathbf{\Gamma}_8 {}_a^I \mathbf{R}(\mathbf{e}_1 d_{a2} + \mathbf{e}_2 d_{a3}) + \mathbf{\Gamma}_7 \times [0\ 0\ 0\ d_{a6} d_{a3}\ -d_{a2} d_{a6}\ d_{a2} d_{a5} - d_{a4} d_{a3}] = \mathbf{0}$$

$$\Leftrightarrow \begin{bmatrix} 0 \\ 0 \\ d_{a1} d_{a3}{}^a a_1 \end{bmatrix} - \begin{bmatrix} d_{a6} d_{a3}{}^a a_3 \\ -d_{a2} d_{a6}{}^a a_3 \\ d_{a2} d_{a5}{}^a a_3 - d_{a4} d_{a3}{}^a a_3 \end{bmatrix} - \begin{bmatrix} -d_{a6} d_{a3}{}^a a_3 \\ d_{a2} d_{a6}{}^a a_3 \\ d_{a1} d_{a3}{}^a a_1 - d_{a5} d_{a2}{}^a a_3 + d_{a4} d_{a3}{}^a a_3 \end{bmatrix} = \mathbf{0}$$

The first two columns of $\mathbf{N}_{a2}$ can be verified as:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{a2} [\mathbf{e}_1\ \mathbf{e}_2] = \mathbf{0}$$

$$\Leftrightarrow \mathbf{\Gamma}_5 \hat{\mathbf{D}}_a^{-1} [\mathbf{e}_1\ \mathbf{e}_2]^a a_2 + \mathbf{\Gamma}_7 \times \begin{bmatrix} 0\ 1\ 0\ 0\ 0\ 0 \\ 0\ 0\ 1\ 0\ 0\ 0 \end{bmatrix}^\top = \mathbf{0}$$

$$\Leftrightarrow \left( \lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}_{I_k}^G \hat{\mathbf{R}} \mathbf{J}_r\left(\Delta\boldsymbol{\theta}_k\right) {}_w^I \hat{\mathbf{R}} \hat{\mathbf{D}}_w \hat{\mathbf{T}}_g \delta t_k + {}_{I_k}^G \hat{\mathbf{R}} \left( \boldsymbol{\Xi}_4 {}_w^I \hat{\mathbf{R}} \hat{\mathbf{D}}_w \hat{\mathbf{T}}_g + \boldsymbol{\Xi}_2 \right) \right) {}_a^I \hat{\mathbf{R}} \left( [\mathbf{e}_1\ \mathbf{e}_2]^a a_2 - [\mathbf{e}_1\ \mathbf{e}_2]^a a_2 \right) = \mathbf{0}$$

The third column of $\mathbf{N}_{a3}$ can be verified as:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{a2} \mathbf{e}_1 d_{a3}{}^a a_2 = \mathbf{0}$$

$$\Leftrightarrow \mathbf{\Gamma}_5 \hat{\mathbf{D}}_a^{-1} \mathbf{e}_1 d_{a3}{}^a a_2 + \mathbf{\Gamma}_7 \times [0\ 0\ 0\ 0\ d_{a6}\ -d_{a5}]^\top - \mathbf{\Gamma}_8 {}_a^I \mathbf{R} \mathbf{e}_1 = \mathbf{0}$$

$$\Leftrightarrow \begin{bmatrix} 0 \\ 0 \\ d_{a3}{}^a a_2 \end{bmatrix} - \begin{bmatrix} 0 \\ d_{a6}{}^a a_3 \\ -d_{a5}{}^a a_3 \end{bmatrix} + \begin{bmatrix} 0 \\ d_{a6}{}^a a_3 \\ -(d_{a3}{}^a a_2 + d_{a5}{}^a a_3) \end{bmatrix} = \mathbf{0}$$

The verification of $\mathbf{N}_{a3}$ can be described as:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{a3} = \mathbf{0}$$
$$\Leftrightarrow \boldsymbol{\Gamma}_5 \hat{\mathbf{D}}_a^{-1} \mathbf{I}_3{}^a a_3 + \boldsymbol{\Gamma}_7 \times [\mathbf{0}_3 \ \ \mathbf{I}_3]^\top = \mathbf{0}$$
$$\Leftrightarrow \left( \lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}^G_{I_k}\hat{\mathbf{R}} \mathbf{J}_r \left( \Delta \boldsymbol{\theta}_k \right) {}^I_w \hat{\mathbf{R}} \hat{\mathbf{D}}_w \hat{\mathbf{T}}_g \delta t_k + {}^G_{I_k}\hat{\mathbf{R}} \left( \boldsymbol{\Xi}_4 {}^I_w \hat{\mathbf{R}} \hat{\mathbf{D}}_w \hat{\mathbf{T}}_g + \boldsymbol{\Xi}_2 \right) \right) {}^I_a \hat{\mathbf{R}} \left( \mathbf{I}_3{}^a a_3 - \mathbf{I}_3{}^a a_3 \right) = \mathbf{0}$$

## E.3: Verification of Lemma 4

The $\mathbf{N}_{g1}$ can be verified as:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{g1} = \mathbf{0}$$
$$\Leftrightarrow \boldsymbol{\Gamma}_4 \mathbf{I}_3{}^I a_1 - \boldsymbol{\Gamma}_9 \times [\mathbf{I}_3 \ \mathbf{0}_3 \ \mathbf{0}_3]^\top = \mathbf{0}$$
$$\Leftrightarrow - (\lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}^G_{I_k}\hat{\mathbf{R}} \mathbf{J}_r \left( \Delta \boldsymbol{\theta}_k \right) \delta t_k + {}^G_{I_k}\hat{\mathbf{R}} \boldsymbol{\Xi}_4 ) {}^I_w \hat{\mathbf{R}} \hat{\mathbf{D}}_w \left( \mathbf{I}_3{}^I a_1 - \mathbf{I}_3{}^I a_1 \right) = \mathbf{0}$$

The $\mathbf{N}_{g2}$ can be verified as:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{g2} = \mathbf{0}$$
$$\Leftrightarrow \boldsymbol{\Gamma}_4 \mathbf{I}_3{}^I a_2 - \boldsymbol{\Gamma}_9 \times [\mathbf{0}_3 \ \mathbf{I}_3 \ \mathbf{0}_3]^\top = \mathbf{0}$$
$$\Leftrightarrow - (\lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}^G_{I_k}\hat{\mathbf{R}} \mathbf{J}_r \left( \Delta \boldsymbol{\theta}_k \right) \delta t_k + {}^G_{I_k}\hat{\mathbf{R}} \boldsymbol{\Xi}_4 ) {}^I_w \hat{\mathbf{R}} \hat{\mathbf{D}}_w \left( \mathbf{I}_3{}^I a_2 - \mathbf{I}_3{}^I a_2 \right) = \mathbf{0}$$

The $\mathbf{N}_{g3}$ can be verified as:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{g3} = \mathbf{0}$$
$$\Leftrightarrow \boldsymbol{\Gamma}_4 \mathbf{I}_3{}^I a_3 - \boldsymbol{\Gamma}_9 \times [\mathbf{0}_3 \ \mathbf{0}_3 \ \mathbf{I}_3]^\top = \mathbf{0}$$
$$\Leftrightarrow - (\lfloor {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k} \rfloor {}^G_{I_k}\hat{\mathbf{R}} \mathbf{J}_r \left( \Delta \boldsymbol{\theta}_k \right) \delta t_k + {}^G_{I_k}\hat{\mathbf{R}} \boldsymbol{\Xi}_4 ) {}^I_w \hat{\mathbf{R}} \hat{\mathbf{D}}_w \left( \mathbf{I}_3{}^I a_3 - \mathbf{I}_3{}^I a_3 \right) = \mathbf{0}$$

## E.4: Verification of Lemma 6

Note that:

$$\mathbf{M}_{Cin} \times [f_u \ f_v \ 0 \ 0 \ 2k_1 \ 4k_2 \ p_1 \ p_2]^\top = \begin{bmatrix} f_u \left( u_d + 2k_1 u_n r^2 + 4k_2 u_n r^4 + 2p_1 u_n v_n + p_2(r^2 + 2u_n^2) \right) \\ f_v \left( v_d + 2k_1 v_n r^2 + 4k_2 v_n r^4 + p_1(r^2 + 2u_n^2) + 2p_2 u_n v_n \right) \end{bmatrix}$$
$$= \begin{bmatrix} f_u \left( u_n + 3k_1 u_n r^2 + 5k_2 u_n r^4 + 4p_1 u_n v_n + 2p_2(r^2 + 2u_n^2) \right) \\ f_v \left( v_n + 3k_1 v_n r^2 + 5k_2 v_n r^4 + 2p_1(r^2 + 2u_n^2) + 4p_2 u_n v_n \right) \end{bmatrix}$$

At the same time, with one-axis rotation assumption, we have:

$$\mathbf{M}_f{}^G\mathbf{k} = \mathbf{H}_{\mathbf{p}_f} {}^C_I \hat{\mathbf{R}} {}^{I_k}_G \hat{\mathbf{R}} \cdot {}^G_{I_0} \hat{\mathbf{R}} {}^I_C \hat{\mathbf{R}} {}^C\mathbf{k}^C z_f$$
$$= \mathbf{H}_{\mathbf{p}_f} \mathbf{e}_3 {}^C z_f$$
$$= \frac{\partial \tilde{\mathbf{z}}_C}{\partial \tilde{\mathbf{z}}_n} \begin{bmatrix} 1 & 0 & -u_n \\ 0 & 1 & -v_n \end{bmatrix} \mathbf{e}_3$$
$$= -\frac{\partial \tilde{\mathbf{z}}_C}{\partial \tilde{\mathbf{z}}_n} \begin{bmatrix} u_n \\ v_n \end{bmatrix}$$

Note that ${}^{I_k}_G \hat{\mathbf{R}} = {}^{I_k}_{I_1} \hat{\mathbf{R}} {}^{I_1}_G \hat{\mathbf{R}}$ and ${}^{I_k}_{I_1} \hat{\mathbf{R}}{}^I\mathbf{k} = {}^I\mathbf{k}$ (due to one-axis rotation). We can easily verify that:

$$\frac{\partial \tilde{\mathbf{z}}_C}{\partial \tilde{\mathbf{z}}_n} \begin{bmatrix} u_n \\ v_n \end{bmatrix} = \begin{bmatrix} f_u \left( u_n + 3k_1 u_n r^2 + 5k_2 u_n r^4 + 4p_1 u_n v_n + 2p_2(r^2 + 2u_n^2) \right) \\ f_v \left( v_n + 3k_1 v_n r^2 + 5k_2 v_n r^4 + 2p_1(r^2 + 2u_n^2) + 4p_2 u_n v_n \right) \end{bmatrix}$$

Therefore, the verification of $\mathbf{N}_{Cin}$ can be written as:

$$\Leftrightarrow \mathcal{O}_k \mathbf{N}_{Cin} = \mathbf{0}$$

$$\Leftrightarrow \mathbf{M}_{Cin} \times [f_u \ f_v \ 0 \ 0 \ 2k_1 \ 4k_2 \ p_1 \ p_2]^\top + \mathbf{M}_f{}^G \mathbf{k} = \mathbf{0}$$

## Appendix F: Appendix F: Interpolation Jacobians

We perturb ${}_{I_{ci-1}}^{G}\mathbf{R}$ and ${}_{I_{ci}}^{G}\mathbf{R}$ as:

$$
{}_{I_{ci-1}}^{G}\mathbf{R} = {}_{I_{ci-1}}^{G}\hat{\mathbf{R}} \exp\left(\delta\boldsymbol{\theta}_{I_{ci-1}}\right) \tag{147}
$$

$$
{}_{I_{ci}}^{G}\mathbf{R} = {}_{I_{ci}}^{G}\hat{\mathbf{R}} \exp\left(\delta\boldsymbol{\theta}_{I_{ci}}\right) \tag{148}
$$

By representing $\boldsymbol{\theta}_{i-1,i} = \log\left({}_{I_{ci-1}}^{G}\mathbf{R}^\top {}_{I_{ci}}^{G}\mathbf{R}\right)$, we have the linearization for the interpolation as:

$$
\delta\boldsymbol{\theta}_{I(t)} \simeq \left(\exp(-\hat{\lambda}\hat{\boldsymbol{\theta}}_{i-1,i}) - \mathbf{J}_r(\hat{\lambda}\hat{\boldsymbol{\theta}}_{i-1,i})\mathbf{J}_l^{-1}(\hat{\boldsymbol{\theta}}_{i-1,i})\right)\delta\boldsymbol{\theta}_{I_{ci-1}} + \mathbf{J}_r(\hat{\lambda}\hat{\boldsymbol{\theta}}_{i-1,i})\mathbf{J}_r^{-1}(\hat{\boldsymbol{\theta}}_{i-1,i})\delta\boldsymbol{\theta}_{I_{ci}} + \frac{m}{M}\boldsymbol{\omega}_{i-1,i}\tilde{t}_r
$$

$$
{}^{G}\tilde{\mathbf{p}}_{I(t)} \simeq (1-\hat{\lambda}){}^{G}\tilde{\mathbf{p}}_{I_{ci-1}} + \hat{\lambda}{}^{G}\tilde{\mathbf{p}}_{I_{ci}} + \frac{m}{M}\mathbf{v}_{i-1,i}\tilde{t}_r
$$

where:

$$
\hat{\boldsymbol{\omega}}_{i-1,i} = \frac{\hat{\boldsymbol{\theta}}_{i-1,i}}{t_{ci} - t_{ci-1}} \tag{149}
$$

$$
\hat{\mathbf{v}}_{i-1,i} = \frac{{}^{G}\hat{\mathbf{p}}_{I_{ci}} - {}^{G}\hat{\mathbf{p}}_{I_{ci-1}}}{t_{ci} - t_{ci-1}} \tag{150}
$$

## Appendix G: Appendix G: Degenerate Motion Simulation Results

The complete calibration plots for IMU/camera intrinsic and IMU-camera spatial calibration under one-axis rotation, constant $a_x$ acceleration and planar motion are shown in Figure 25, 26 and 27. All the temporal calibration results for IMU-camera time offset and rolling shutter readout time on the four simulated trajectories are shown in Figure 9.

## References

[1] Shaojie Shen et al. "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV". In: *Proc. of the IEEE International Conference on Robotics and Automation*. Hong Kong, China, 2014, pp. 4974–4981.

[2] Lionel Heng et al. "Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 4695–4702.

[3] Mingming Zhang et al. "Pose Estimation for Ground Robots: On Manifold Representation, Integration, Reparameterization, and Optimization". In: *IEEE Transactions on Robotics* 37.4 (2021), pp. 1081–1099. DOI: 10.1109/TRO.2020.3043970.

[4] Woosik Lee et al. "Visual-Inertial-Wheel Odometry with Online Calibration". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020.
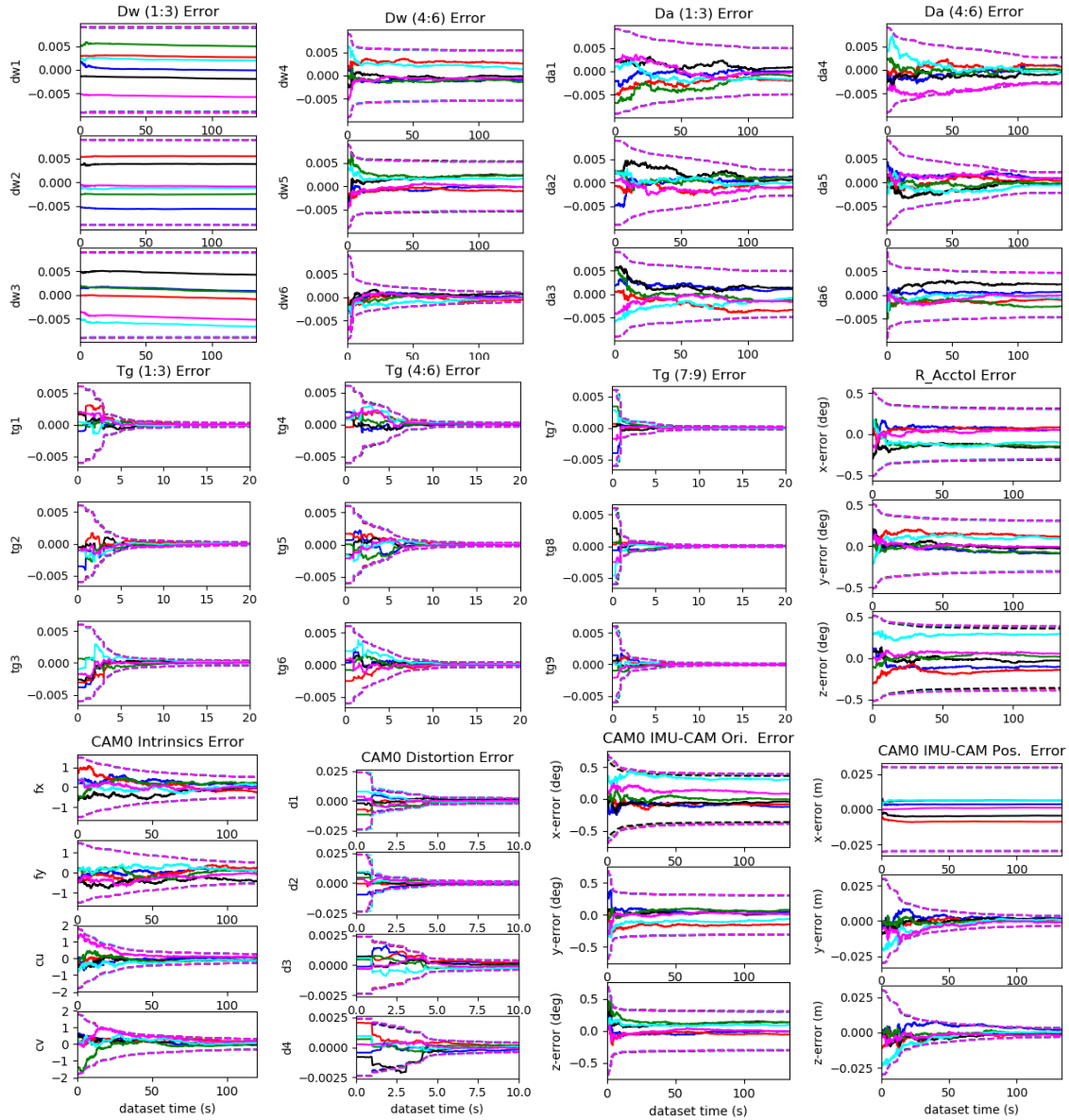
Figure 25: Calibration results for the proposed system evaluated on *tum_room* with one-axis rotation using *imu22* and *radtan*. 3 sigma bounds (dotted lines) and estimation errors (solid lines) for six different runs (different colors) with different realization of the measurement noise and initial perturbations. Note that the estimation errors and 3 $\sigma$ bounds for $d_{w1}$, $d_{w2}$, $d_{w3}$ and the IMU-CAM position calibration along the rotation axis can not converge.
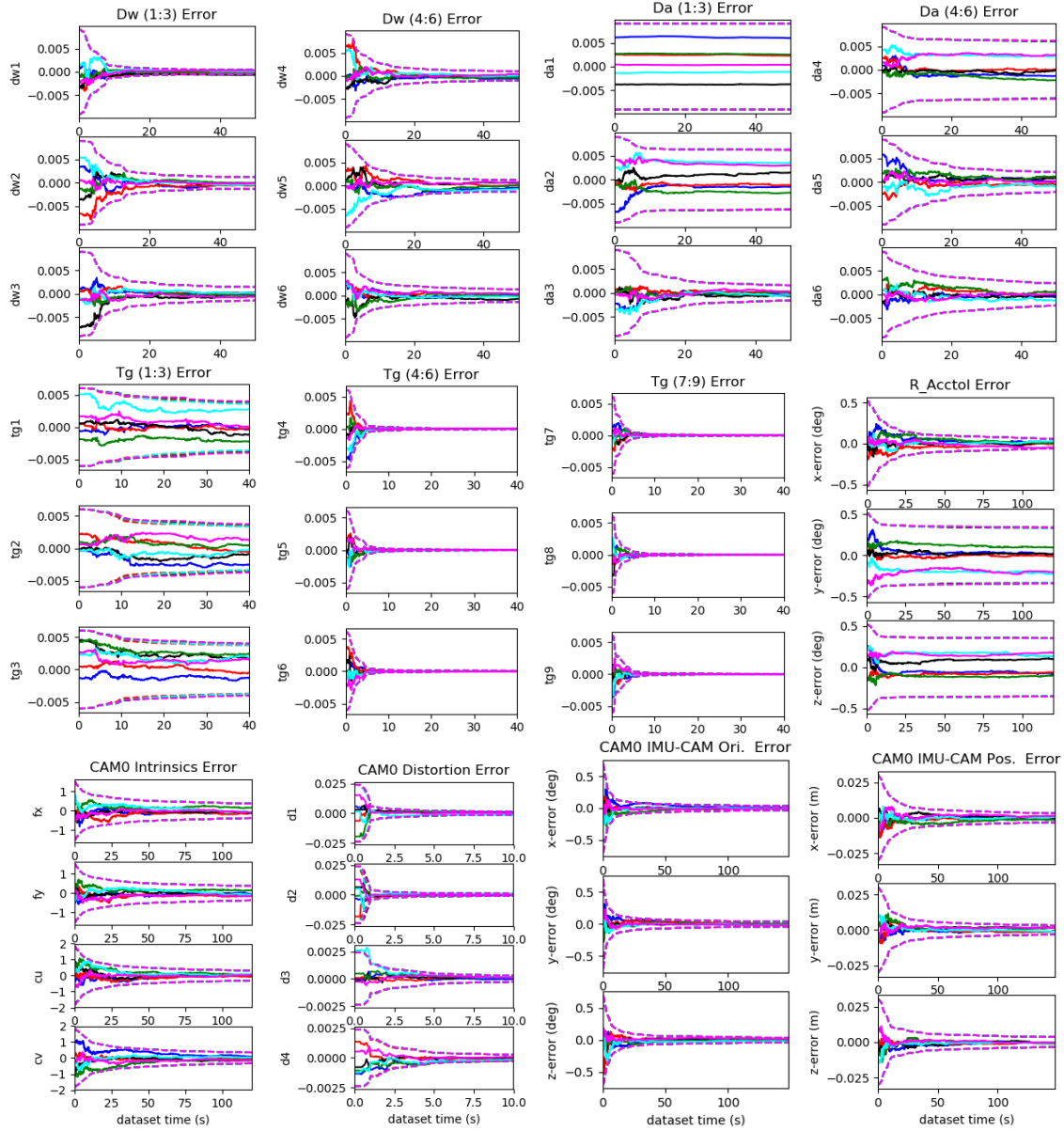
Figure 26: Calibration results for the proposed system evaluated the *sine_3d* with constant acceleration along x direction using *imu22* and *radtan*. 3 sigma bounds (dotted lines) and estimation errors (solid lines) for six different runs (different colors) with different realization of the measurement noise and initial perturbations. The estimation errors and $3\sigma$ bounds for $d_{a1}$, pitch and yaw of $_a^I\mathbf{R}$ cannot converge. Note that $t_{g1}$, $t_{g2}$ and $t_{g3}$ are also unobservable.
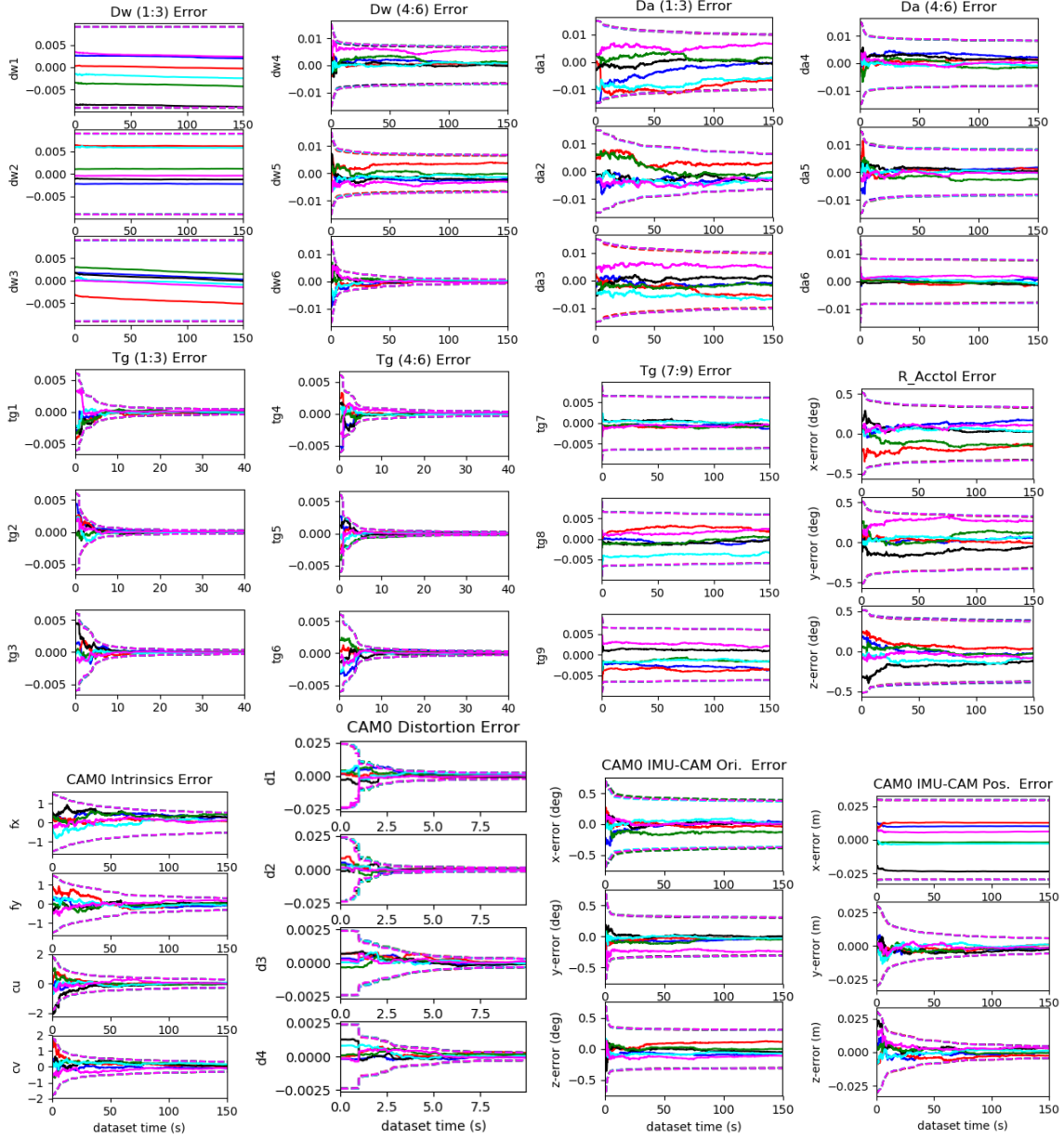
Figure 27: Calibration results of the proposed system evaluated on *udel_gore* with planar motion using *imu22* and *radtan*. 3 sigma bounds (dotted lines) and estimation errors (solid lines) for six different runs (different colors) with different realization of the measurement noise and initial perturbations. With planar motion, the estimation errors and 3 $\sigma$ bounds of $d_{w1}$, $d_{w2}$, $d_{w3}$, $t_{g7}$, $t_{g8}$, $t_{g9}$ and the IMU-CAM position cannot converge. Due to lack of motion excitation, the parameters of $\mathbf{D}_a$ and $_a^I\mathbf{R}$ converge much slower than the other motion cases.

[5]     M. Li and A. Mourikis. "High-Precision, Consistent EKF-based Visual-Inertial Odometry". In: *International Journal of Robotics Research* 32.6 (2013), pp. 690–711.

[6]     Chao Guo et al. "Efficient Visual-Inertial Navigation using a Rolling-Shutter Camera with Inaccurate Timestamps". In: *Proc. of the Robotics: Science and Systems Conference.* Berkeley, CA, 2014.

[7]     Guoquan Huang. "Visual-Inertial Navigation: A Concise Review". In: *Proc. International Conference on Robotics and Automation.* Montreal, Canada, May 2019.

[8]     A. I. Mourikis and S. I. Roumeliotis. "A multi-state constraint Kalman filter for vision-aided inertial navigation". In: *Proceedings of the IEEE International Conference on Robotics and Automation.* Rome, Italy, 2007, pp. 3565–3572.

[9]     Kejian J Wu et al. "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices". In: *Robotics: Science and Systems Conference (RSS).* 2015.

[10]    Patrick Geneva et al. "OpenVINS: A Research Platform for Visual-Inertial Estimation". In: *Proc. of the IEEE International Conference on Robotics and Automation.* Paris, France, 2020. URL: https://github.com/rpng/open_vins.

[11]    Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang. "MIMC-VINS: A Versatile and Resilient Multi-IMU Multi-Camera Visual-Inertial Navigation System". In: *IEEE Transactions on Robotics* 37.5 (2021), pp. 1360–1380. DOI: 10.1109/TRO.2021.3049445.

[12]    Stefan Leutenegger et al. "Keyframe-based visual-inertial odometry using nonlinear optimization". In: *International Journal of Robotics Research* 34.3 (2015), pp. 314–334.

[13]    C. Forster et al. "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry". In: *IEEE Transactions on Robotics* 33.1 (Feb. 2017), pp. 1–21.

[14]    T. Qin, P. Li, and S. Shen. "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator". In: *IEEE Transactions on Robotics* 34.4 (2018), pp. 1004–1020.

[15]    Vladyslav C. Usenko et al. "Visual-Inertial Mapping With Non-Linear Factor Recovery". In: *IEEE Robotics and Automation Letters* 5 (2020), pp. 422–429.

[16]    Carlos Campos et al. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM". In: *IEEE Transactions on Robotics* (2021), pp. 1–17. DOI: 10.1109/TRO.2021.3075644.

[17]    M. Li et al. "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation". In: *IEEE International Conference on Robotics and Automation (ICRA).* 2014, pp. 409–416.

[18]    Yulin Yang et al. "Degenerate Motion Analysis for Aided INS with Online Spatial and Temporal Calibration". In: *IEEE Robotics and Automation Letters (RA-L)* 4.2 (2019), pp. 2070–2077.

[19]    Yulin Yang et al. "Online IMU Intrinsic Calibration: Is It Necessary?" In: *Proc. of Robotics: Science and Systems (RSS).* Corvallis, Or, 2020.

[20]    F. M. Mirzaei and S. I. Roumeliotis. "A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation". In: *IEEE Transactions on Robotics* 24.5 (Oct. 2008), pp. 1143–1156.

[21]    Jonathan Kelly and Gaurav S. Sukhatme. "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-Calibration". In: *International Journal of Robotics Research* 30.1 (Jan. 2011), pp. 56–79.

[22]   Chao Guo and Stergios Roumeliotis. "IMU-RGBD Camera 3D Pose Estimation and Extrinsic Calibration: Observability Analysis and Consistency Improvement". In: *Proc. of the IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany, 2013.

[23]   Shu-Hua Tsao and Shau-Shiun Jan. "Observability Analysis and Performance Evaluation of EKF-Based Visual-Inertial Odometry With Online Intrinsic Camera Parameter Calibration". In: *IEEE Sensors Journal* 19.7 (2019), pp. 2695–2703. DOI: 10.1109/JSEN.2018.2886764.

[24]   Thomas Schneider et al. "Observability-aware self-calibration of visual and inertial sensors for ego-motion estimation". In: *IEEE Sensors Journal* 19.10 (2019), pp. 3846–3860.

[25]   M. Li and A. I. Mourikis. "Online Temporal Calibration for Camera-IMU Systems: Theory and Algorithms". In: 33.7 (June 2014), pp. 947–964.

[26]   T. Schneider et al. "Observability-Aware Self-Calibration of Visual and Inertial Sensors for Ego-Motion Estimation". In: *IEEE Sensors Journal* 19.10 (2019), pp. 3846–3860. ISSN: 2379-9153. DOI: 10.1109/JSEN.2019.2893809.

[27]   Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[28]   O. D. Faugeras, Q. T. Luong, and S. J. Maybank. "Camera self-calibration: Theory and experiments". In: *Computer Vision — ECCV'92*. Ed. by G. Sandini. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 321–334. ISBN: 978-3-540-47069-4.

[29]   Gang Qian and Rama Chellappa. "Bayesian Self-Calibration of a Moving Camera". In: *Computer Vision Image Understanding* 95.3 (Sept. 2004), 287–316. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2004.03.009.

[30]   Javier Civera et al. "Camera self-calibration for sequential Bayesian structure from motion". In: *2009 IEEE International Conference on Robotics and Automation*. 2009, pp. 403–408. DOI: 10.1109/ROBOT.2009.5152719.

[31]   Antonio Agudo. "Total Estimation from RGB Video: On-line Camera Self-Calibration, Non-Rigid Shape and Motion". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. 2021, pp. 8140–8147. DOI: 10.1109/ICPR48806.2021.9412923.

[32]   Marci Meingast, Christopher Geyer, and S. Shankar Sastry. "Geometric Models of Rolling-Shutter Cameras". In: *The 6th Workshop on Omnidirectional Vision* abs/cs/0503076 (2005), p. 4.

[33]   Luc Oth et al. "Rolling Shutter Camera Calibration". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1360–1367. DOI: 10.1109/CVPR.2013.179.

[34]   E. Olson. "AprilTag: A Robust and Flexible Visual Fiducial System". In: *Proc. of the IEEE International Conference on Robotics and Automation*. Shanghai, China, 2011, pp. 3400–3407.

[35]   Thanh-Tin Nguyen and Maxime Lhuillier. "Self-calibration of omnidirectional multi-cameras including synchronization and rolling shutter". In: *Computer Vision and Image Understanding* 162 (2017), pp. 166–184. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2017.08.010.

[36]   Zuzana Kukelova et al. "Minimal Rolling Shutter Absolute Pose with Unknown Focal Length and Radial Distortion". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 698–714. ISBN: 978-3-030-58558-7.

[37]   Alexandre Karpenko et al. "Digital video stabilization and rolling shutter correction using gyroscopes". In: *Stanford University Computer Science Tech Report CSTR 2011-03* 1.2 (2011), p. 13.

[38]  P. Furgale, J. Rehder, and R. Siegwart. "Unified temporal and spatial calibration for multi-sensor systems". In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 1280–1286.

[39]  Eagle S. Jones and Stefano Soatto. "Visual-inertial Navigation, Mapping and Localization: A Scalable Real-time Causal Approach". In: *International Journal of Robotics Research* 30.4 (Apr. 2011), pp. 407–430.

[40]  Pierre-Jean Bristeau et al. "The Navigation and Control technology inside the AR.Drone micro UAV". In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, pp. 1477–1484. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20110828-6-IT-1002.02327.

[41]  Yao Xiao et al. "Online IMU Self-Calibration for Visual-Inertial Systems". In: *Sensors* 19.7 (2019). ISSN: 1424-8220. DOI: 10.3390/s19071624.

[42]  Jae Hyung Jung, Sejong Heo, and Chan Gook Park. "Observability Analysis of IMU Intrinsic Parameters in Stereo Visual–Inertial Odometry". In: *IEEE Transactions on Instrumentation and Measurement* 69.10 (2020), pp. 7530–7541. DOI: 10.1109/TIM.2020.2985174.

[43]  Dave Zachariah and Magnus Jansson. "Joint calibration of an inertial measurement unit and coordinate transformation parameters using a monocular camera". In: *2010 International Conference on Indoor Positioning and Indoor Navigation*. 2010, pp. 1–7. DOI: 10.1109/IPIN.2010.5646840.

[44]  J. Rehder et al. "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 4304–4311. DOI: 10.1109/ICRA.2016.7487628.

[45]  Jianzhu Huai et al. "Continuous-Time Spatiotemporal Calibration of a Rolling Shutter Camera-IMU System". In: *CoRR* abs/2108.07200 (2021). arXiv: 2108.07200.

[46]  Janosch Nikolic et al. "Non-Parametric Extrinsic and Intrinsic Calibration of Visual-Inertial Sensor Systems". In: *IEEE Sensors Journal* 16.13 (2016), pp. 5433–5443. DOI: 10.1109/JSEN.2016.2556662.

[47]  Tong Qin and Shaojie Shen. "Online Temporal Calibration for Monocular Visual-Inertial Systems". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 3662–3669. DOI: 10.1109/IROS.2018.8593603.

[48]  Kevin Eckenhoff et al. "Multi-Camera Visual-Inertial Navigation with Online Intrinsic and Extrinsic Calibration". In: *Proc. International Conference on Robotics and Automation*. Montreal, Canada, May 2019.

[49]  Jianzhu Huai et al. "Observability Analysis and Keyframe-Based Filtering for Visual Inertial Odometry With Full Self-Calibration". In: *IEEE Transactions on Robotics* 38.5 (2022), pp. 3219–3237. DOI: 10.1109/TRO.2022.3172474.

[50]  Guoquan Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. "Observability-based Rules for Designing Consistent EKF SLAM Estimators". In: *International Journal of Robotics Research* 29.5 (Apr. 2010), pp. 502–528. DOI: 10.1177/0278364909353640.

[51]  A. Martinelli. "Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination". In: *IEEE Transactions on Robotics* 28.1 (2012), pp. 44–60.

[52]  J.A. Hesch et al. "Consistency Analysis and Improvement of Vision-aided Inertial Navigation". In: *IEEE Transactions on Robotics* 30.1 (2013), pp. 158–176.

[53] Agostino Martinelli. "State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration". In: *IEEE Transactions on Robotics* 27.2 (2011), pp. 239–255.

[54] Guoquan Huang. "Improving the Consistency of Nonlinear Estimators: Analysis, Algorithms, and Applications". PhD thesis. Department of Computer Science and Engineering, University of Minnesota, 2012. URL: https://conservancy.umn.edu/handle/11299/146717.

[55] K. Wu et al. "An invariant-EKF VINS algorithm for improving consistency". In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2017, pp. 1578–1585.

[56] Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2017.

[57] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*. Vol. 1. London: Academic Press, 1979.

[58] J. Rehder and R. Siegwart. "Camera/IMU Calibration Revisited". In: *IEEE Sensors Journal* 17.11 (2017), pp. 3257–3268. ISSN: 2379-9153. DOI: 10.1109/JSEN.2017.2674307.

[59] OpenCV Developers Team. *Open Source Computer Vision (OpenCV) Library*. Available: http://opencv.org. 2021.

[60] Nikolas Trawny and Stergios I. Roumeliotis. *Indirect Kalman Filter for 3D Attitude Estimation*. Tech. rep. University of Minnesota, Dept. of Comp. Sci. & Eng., Mar. 2005. URL: http://mars.cs.umn.edu/tr/reports/Trawny05b.pdf.

[61] Averil B. Chatfield. *Fundamentals of High Accuracy Inertial Navigation*. American Institute of Aeronautics and Astronautics, 1997.

[62] Yulin Yang et al. "Analytic Combined IMU Integration for Visual-Inertial Navigation". In: *Proc. of the IEEE International Conference on Robotics and Automation*. Paris, France, 2020.

[63] Gregory Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Vol. 2. Springer Science & Business Media, 2011.

[64] Agostino Martinelli. "Observability Properties and Deterministic Algorithms in Visual-Inertial Structure from Motion". In: *Found. Trends Robot* 3.3 (Dec. 2014), 139–209. ISSN: 1935-8253. DOI: 10.1561/2300000030.

[65] J. A. Hesch et al. "Consistency Analysis and Improvement of Vision-aided Inertial Navigation". In: *IEEE Transactions on Robotics* 30.1 (2014), pp. 158–176. ISSN: 1941-0468. DOI: 10.1109/TRO.2013.2277549.

[66] K. J. Wu et al. "VINS on wheels". In: *Proc. of the IEEE International Conference on Robotics and Automation*. 2017, pp. 5155–5162.

[67] Yulin Yang and Guoquan Huang. "Observability Analysis of Aided INS with Heterogeneous Features of Points, Lines and Planes". In: *IEEE Transactions on Robotics* 35.6 (Dec. 2019), pp. 399–1418.

[68] Yulin Yang, James Maley, and Guoquan Huang. "Null-Space-based Marginalization: Analysis and Algorithm". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vancouver, Canada, 2017, pp. 6749–6755.

[69] Mingyang Li and Anastasios I Mourikis. "Optimization-based estimator design for vision-aided inertial navigation". In: *Robotics: Science and Systems*. Berlin, Germany, 2013, pp. 241–248.

[70] David Schubert et al. "The TUM VI benchmark for evaluating visual-inertial odometry". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1680–1687.

[71] D. Schubert et al. "Rolling-Shutter Modelling for Visual-Inertial Odometry". In: *International Conference on Intelligent Robots and Systems (IROS)*. 2019.

[72] Michael Burri et al. "The EuRoC micro aerial vehicle datasets". In: *The International Journal of Robotics Research* 35.10 (2016), pp. 1157–1163.

[73] Jinyong Jeong et al. "Complex urban dataset with multi-level sensors from highly diverse urban environments". In: *The International Journal of Robotics Research* 38.6 (2019), pp. 642–657.