Visual-Inertial State Estimation with Decoupled Error and State Representations

Chuchu Chen*, Yuxiang Peng*, and Guoquan Huang

University of Delaware, Newark, DE 19706, USA ‡ §

Abstract. In this paper, we advocate the Decoupled Error and State (DES) methodology for state estimation, which uses distinct representations for error and state estimates and updates the state through tailored functions based on the selected representations. Focusing on Visual-Inertial Navigation Systems (VINS), for the first time, we analytically discover the connections between the prominent VINS estimators, offering a unified view and insightful understanding of the SOTA algorithms. Building upon this discovery along with the proposed DES idea, we further develop the DES-VINS. The proposed estimator adopts a global-centric state to naturally represent the physical quantities concerned by the underlying navigation system, while designing a new error representation by lifting orientation to mitigate the issues caused by linearization and ensure proper observability properties. Interestingly, despite not being constrained by the Lie-group affine properties (which are often challenging to ensure in practice), the DES-VINS estimator is shown to share the identical properties of the linearized error-state system as the invariant EKF. However, the DES-VINS algorithm allows efficient and consistent integration of long-tracked SLAM features (which are almost always needed in practice), being 3× faster than the invariant VINS. Extensive numerical studies and real-world experiments are presented to compare with SOTA VINS estimators, providing valuable insights into their performance and applicability.

Keywords: Localization and Mapping, Perception

1 Introduction and Related Work

Visual-inertial navigation systems (VINS) fuse the high-rate inertial measurements with the visual information to estimate 6 degrees-of-freedom (d.o.f) poses, leading to their widespread applications across diverse fields [10, 15, 29, 39, 40]. VINS is centered on a state estimation algorithm that aims to optimally fuses sensor data. VINS estimators fall into two main categories: the filter-based method, which linearizes the system once, and the optimization-based methods, which solve a nonlinear least-squares (NLS) problem with relinearization.

‡This work was partially supported by the University of Delaware (UD) College of Engineering, the Delaware NASA/EPSCoR Seed Grant, the NSF (MRI-2018905, SCH-2014264), Google ARCore, and Meta Reality Labs.

§* equal contribution

Beyond solving the estimation problem, problem design and formulation is also crucial and widely studied in the literature, forming the key focus of our work.

Most VINS estimators use a *global-centric* formulation, directly estimating states relative to a fixed global (world) frame [7, 32, 33, 38, 41, 42]. It typically estimates the orientation (represented by $\mathcal{SO}(3)$ or unit quaternions), position, velocity of the sensing platform, and environmental features in the global frame. The standard linearized estimators based on this formulation may encounter the inconsistency issue related to the system observability. Discrepancies between sequential linearization points in the estimator can result in spurious information gains along unobservable directions, hurting the performance [11, 14, 17, 24, 31, 34]. Alternatively, in a *robocentric* formulation, the moving body frame of the sensor platform serves as the navigation frame of reference [8, 9], estimating the relative pose between consecutive locations and the current pose with respect to the initial (body) frame can be reconstructed by incrementally combining new relative pose estimates [26, 27]. Recently, *invariant*-EKF-based VINS estimators have seen a growing interest [3, 5, 22, 23, 25, 43], which are grounded on the Lie group observer design theory [4] and assume the group affinity of the underlying systems (which is often hard to hold in practice). This method models the state on the manifold using a Matrix Lie group [2, 16] and improves performance, but computation increases with more state variables (e.g., features). Interestingly, both the invariant and robocentric formulations, despite requiring linearization and being designed from different perspectives, have an unobservable subspace independent of linearization points.

All the aforementioned standard VINS estimators, despite their unique formulations and specific estimator characteristics, employ the same representation of the error and state estimates – representing the same physical or mathematical quantities – so that they can be naturally aligned and corrected. However, this is *not* the only choice. As shown in [47], one can represent the state vector as keyframe poses and employ a B-spline function to model the error state, thus effectively reducing the error state size and improving efficiency of pose estimation. We thus formalize this idea that the error and state estimates do *not* necessarily share the same representation; that is, the state correction operation ($\mathbf{x} = \hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}}$) can be a distinct *nonlinear* function, whose form is determined by the specific state and error state representations selected. Surprisingly, this perspective, though used implicitly in past approaches, has never been thoroughly studied. Meanwhile, there is a notable lack of in-depth investigation and analysis of the various design choices for estimators. The extensive literature and historical development of VINS algorithms, each designed with unique perspectives, can challenge practitioners navigating this complex field.

Therefore, for the first time, we examine the prominent VINS algorithms through a unified vision, analytically establishing equivalent connections between them and showcasing the capability to transit flexibly between design choices. Reflecting on our discoveries, we introduce DES-VINS, which combines the global-centric state representation with a novel error-state design. Interestingly, DES-VINS achieves equality with the invariant formulation in accuracy and consistency despite not being strictly bound by Lie group structures and

system group affinity. However, it achieves speeds 3× faster and enables straightforward initialization, all thanks to the DES concept. In summary, the primary contributions of our work include:

- We introduce the DES design methodology, which allows the state and error state to have different representations, broadening the potential design spaces for estimators.
- For the first time, we analytically identify the transformations between the SOTA VINS estimators. This unified vision brings together various aspects into a cohesive understanding, providing deep and clear insight. Furthermore, it allows leveraging the unique strengths of each formulation to enhance estimation performance.
- We develop the DES-VINS algorithm, which employs a global-centric state alongside a novel error state formulation. Interestingly, without relying on the group-affine property or introducing extra constraints, the DES-VINS error state is proven to be equivalent to the invariant formulation, which ensures accuracy and consistency while achieving a threefold speed improvement in integrating long-tracked SLAM features and enabling a simple system initialization method. With extensive experiments to compare different estimator designs, we provide valuable insights and discussions.

2 Decoupling Error and State Representations

In this section, we describe the key idea of the proposed decoupled-error-and-state (DES) representation for linearized state estimator design. Given measurements $\mathbf{z}$, a least-squares formulation is often used, equivalent to maximum likelihood estimation under mild assumptions [36]:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} ||\mathbf{z} - \mathbf{h}(\mathbf{x})||^2 \tag{1}$$

Here, $\mathbf{x}$ is the state to be estimated, and $\mathbf{h}(\cdot)$ is the typically nonlinear measurement function. It is often transformed into a linear least-squares problem in terms of the error state $\tilde{\mathbf{x}}$:

$$\tilde{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}}} ||\mathbf{r} - \mathbf{H}\tilde{\mathbf{x}}||^2 \tag{2}$$

where $\mathbf{r} = \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})$ is the measurement residual, and $\mathbf{H}$ is the measurement Jacobian, typically linearized around the current state estimate $\hat{\mathbf{x}}$. The state estimate can thereafter be corrected iteratively as:

$$\hat{\mathbf{x}}^{\oplus} = \hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}} \tag{3}$$

In a standard estimator, the correction or update operation $\boxplus$ maps the state estimate and its corresponding error of the *same* representation (or parameterization). For example, in the case of position $\mathbf{p}$ in the vector space, a simple addition can be used: $\hat{\mathbf{p}}^{\oplus} = \hat{\mathbf{p}} + \tilde{\mathbf{p}}$. For orientation, when the state is represented in the $\mathcal{SO}(3)$ group, the state correction is typically expressed as $\hat{\mathbf{R}}^{\oplus} = \exp(-\delta\boldsymbol{\theta})\hat{\mathbf{R}}$,

where $\delta\boldsymbol{\theta}$ is the corresponding error angle. In the invariant formulation, where states are modeled within a special Lie group, $\mathbf{x} \in SE_2(3)$, the state correction is $\mathbf{x}^{\oplus} = \exp(\tilde{\mathbf{x}})\hat{\mathbf{x}}$, with $\exp(\cdot)$ as the matrix exponential map [2, 16].

However, we advocate a general DES-estimator design methodology that the state and error state are *not* restricted to share the same representations and can be chosen with different representations of different mathematical quantities. Consequently, the state correction $\boxplus$ becomes a *nonlinear* mapping. This offers flexibility in state estimator design and opens up new avenues for exploration. A carefully crafted error-state formulation can result in the superior performance of the corresponding linearized estimator, in terms of observability, efficiency, and convergence. In the following, we will apply this DES methodology to VINS, closely inspect different VINS formulations and identify their connections.

3 Unifying VINS Estimators

In VINS, the state vector $\mathbf{x}$ and error state $\tilde{\mathbf{x}}$ can be summarized as:

$$\mathbf{x} = (\mathbf{x}_I, \mathbf{x}_b, \mathbf{x}_f), \quad \tilde{\mathbf{x}} = (\tilde{\mathbf{x}}_I, \tilde{\mathbf{x}}_b, \tilde{\mathbf{x}}_f) \tag{4}$$

where $\mathbf{x}_I$ is the IMU state, typically including IMU orientation, position, and velocity; $\mathbf{x}_f$ denotes the features, which in general consists of 3D environmental feature positions; Let $\mathbf{x}_b$ represent the bias, with $\tilde{\mathbf{x}}_b$ denoting the corresponding error state. The navigation state is then defined as $\mathbf{x}_n := (\mathbf{x}_I, \mathbf{x}_f)$. In the following, we will carefully detail the modeling of the state and error states for each estimator formulation. The global frame of reference is denoted by $\{G\}$, the local IMU frame by $\{I\}$, and the camera frame by $\{C\}$. For simplicity, we assume there is no spatial-temporal offset between the camera and IMU. In this section, we explain the key concept in filtering-based VINS, while the proposed DES methodology is versatile and suitable for any linearized estimators, including those based on optimization.

3.1 System Models

IMU Model and Propagation A canonical six-axis IMU provides linear acceleration and angular velocity measurements, $\mathbf{a}_m$ and $\boldsymbol{\omega}_m$, modeled as:

$$\mathbf{a}_m = \mathbf{a} + \mathbf{b}_a + \mathbf{n}_a, \quad \boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g \tag{5}$$

where $\mathbf{n}_g$ and $\mathbf{n}_a$ are zero-mean white Gaussian noise, $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases, driven by noises (i.e., $\dot{\mathbf{b}}_g = \mathbf{n}_{wg}, \dot{\mathbf{b}}_a = \mathbf{n}_{wa}$). Integrating inertial readings $\mathbf{u}_{k:k+1}$ within the time interval $[t_k, t_{k+1}]$, we get:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_{k:k+1}, \mathbf{w}_k), \quad \tilde{\mathbf{x}}_{k+1} \simeq \boldsymbol{\Phi}_{k+1,k}\tilde{\mathbf{x}}_k + \mathbf{G}_k\mathbf{w}_k \tag{6}$$

where $\mathbf{w}_k$ is discretized noise vector, $\boldsymbol{\Phi}_{k+1,k}$ is the state transition matrix and $\mathbf{G}_k$ is the noise Jacobian matrix:

$$\boldsymbol{\Phi}_{k+1,k} = \begin{bmatrix} \boldsymbol{\Phi}_{nn} & \boldsymbol{\Phi}_{nb} & \mathbf{0}_{9\times 3N} \\ \mathbf{0}_{6\times 9} & \mathbf{I}_6 & \mathbf{0}_{6\times 3N} \\ \mathbf{0}_{3N\times 9} & \boldsymbol{\Phi}_{fb} & \mathbf{I}_{3N} \end{bmatrix}, \mathbf{G}_k = \begin{bmatrix} \mathbf{G}_{nn} & \mathbf{0}_{9\times 6} \\ \mathbf{0}_6 & \mathbf{I}_6\Delta t \\ \mathbf{G}_{fn} & \mathbf{0}_{3N\times 6} \end{bmatrix} \tag{7}$$

N denotes the number of features in the state and $\Delta t = t_{k+1} - t_k$ is the time difference. The subscript denotes the Jacobians corresponding to different states (i.e., $\boldsymbol{\Phi}_{nb} = \partial \tilde{\mathbf{x}}_n / \partial \tilde{\mathbf{x}}_b$). Considering the variety in error state formulations, the matrices $\boldsymbol{\Phi}_{k+1,k}$ and $\mathbf{G}_k$ can exhibit distinct derivations. The standard EKF propagation equation can then be leveraged to propagate the error covariance [36].

Visual Measurement and Update The camera observes environmental features from its poses. A point feature measurement at t_k is expressed as:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k := \boldsymbol{\Lambda}({}^{C_k}\mathbf{p}_{f_i}) + \mathbf{n}_k \tag{8}$$

where $\mathbf{z}_k$ is the point feature measurement in pixel coordinates, ${}^{C_k}\mathbf{p}_{f_i}$ is the i'th feature position expressed in the camera frame. $\boldsymbol{\Lambda}(\cdot)$ is the camera projection model; $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the white Gaussian noise; Linearizing Eq. (8) gives:

$$\mathbf{r}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k) \simeq \mathbf{H}_k \tilde{\mathbf{x}}_k + \mathbf{n}_k \tag{9}$$

where $\mathbf{r}_k$ is the residual, $\mathbf{H}_k$ is the Jacobian. Once it passes the Mahalanobis gating test, we can apply the EKF update equations to update the filter [36].

3.2 VINS Estimators

We next examine different estimator designs. For simplicity, we assume a single feature and omit the subscript k in the derivations.

Global-Centric Formulation The global-centric formulation estimates the robot state in the global frame of reference $\{G\}$. Following the MSCKF 2.0 [34], the state and error state can be defined as:

$$\mathbf{x}_n := \left({}^G_I\mathbf{R}, {}^G\mathbf{p}_I, {}^G\mathbf{v}_I, {}^G\mathbf{p}_f\right), \qquad \tilde{\mathbf{x}}_n = \left[\tilde{\boldsymbol{\theta}}^\top \ {}^G\tilde{\mathbf{p}}_I^\top \ {}^G\tilde{\mathbf{v}}_I^\top \ {}^G\tilde{\mathbf{p}}_f^\top\right]^\top \in \mathbb{R}^{12} \tag{10}$$

For orientation, we utilize the $\mathcal{SO}(3)$ perturbation model: ${}^G_I\mathbf{R} = \exp(-\tilde{\boldsymbol{\theta}}){}^G_I\hat{\mathbf{R}} \simeq (\mathbf{I} - \lfloor\tilde{\boldsymbol{\theta}}\rfloor){}^G_I\hat{\mathbf{R}}$, where $\lfloor\cdot\rfloor$ denotes the skew-symmetric matrix. Other states are represented in vector space, allowing for additive error, i.e., ${}^G\mathbf{p}_I = {}^G\hat{\mathbf{p}}_I + {}^G\tilde{\mathbf{p}}_I$.

Invariant Formulation The navigation state $\mathbf{X}_n$ is modeled in a special Lie group. The biases, are still within the vector space (i.e., $\mathbf{b}_g = \hat{\mathbf{b}}_g + \delta\mathbf{b}_g \in \mathbb{R}^3$). This approach is now widely recognized as the imperfect invariant EKF (see Table 2 in [19] for a summary). We will now focus on the derivation for the navigation state $\mathbf{X}_n$ and its error state $\delta\mathbf{x}_n$:

$$\mathbf{X}_n := \begin{bmatrix} {}^G_I\mathbf{R} & {}^G\mathbf{p}_I & {}^G\mathbf{v}_I & {}^G\mathbf{p}_f \\ \mathbf{0}_3 & & \mathbf{I}_3 & \end{bmatrix} \in SE_3(3), \quad \delta\mathbf{x}_n := \left[\delta\boldsymbol{\theta}^\top \ \delta\mathbf{p}_I^\top \ \delta\mathbf{v}_I^\top \ \delta\mathbf{p}_f^\top\right]^\top \in \mathbb{R}^{12} \tag{11}$$

Defining $\hat{\mathbf{X}}_n$ as the state estimate we adopt the right invariant perturbation as:

$$\mathbf{X}_n = \exp(\delta\mathbf{x}_n)\hat{\mathbf{X}}_n \tag{12}$$

$$= \begin{bmatrix} \exp(-\delta\boldsymbol{\theta}) & \mathbf{J}_l(-\delta\boldsymbol{\theta})\delta\mathbf{p}_I & \mathbf{J}_l(-\delta\boldsymbol{\theta})\delta\mathbf{v} & \mathbf{J}_l(-\delta\boldsymbol{\theta})\delta\mathbf{p}_f \\ \mathbf{0}_3 & & \mathbf{I}_3 & \end{bmatrix} \begin{bmatrix} {}^G_I\hat{\mathbf{R}} & {}^G\hat{\mathbf{p}}_I & {}^G\hat{\mathbf{v}}_I & {}^G\hat{\mathbf{p}}_f \\ \mathbf{0}_3 & & \mathbf{I}_3 & \end{bmatrix} \quad (13)$$

where $\mathbf{J}_l(\cdot)$ denotes the left Jacobian. By expanding the matrix, we derive perturbations for each state variable. This allows us to establish the relationship between the error state $\tilde{\mathbf{x}}$ for the global-centric formulation and $\delta\mathbf{x}$ for the invariant formulation. For the orientation and position, we have:

$$\begin{aligned} {}^G_I\mathbf{R} &= \exp(-\tilde{\boldsymbol{\theta}}){}^G_I\hat{\mathbf{R}} = \exp(-\delta\boldsymbol{\theta}){}^G_I\hat{\mathbf{R}} && \Rightarrow \tilde{\boldsymbol{\theta}} = \delta\boldsymbol{\theta} \quad (14)\\ {}^G\mathbf{p}_I &= {}^G\hat{\mathbf{p}}_I + {}^G\tilde{\mathbf{p}}_I = \exp(-\delta\boldsymbol{\theta}){}^G\hat{\mathbf{p}}_I + \mathbf{J}_l(-\delta\boldsymbol{\theta})\delta\mathbf{p}_I && \Rightarrow {}^G\tilde{\mathbf{p}}_I \simeq \lfloor {}^G\hat{\mathbf{p}}_I \rfloor \delta\boldsymbol{\theta} + \delta\mathbf{p}_I \end{aligned}$$

where we have assumed $\mathbf{J}_l(-\delta\boldsymbol{\theta}) \simeq \mathbf{I}_3$ due to the generally small magnitude of the rotation error $\delta\boldsymbol{\theta}$. Similarly, we can derive the velocity and feature errors:

$${}^G\tilde{\mathbf{v}}_I \simeq \lfloor {}^G\hat{\mathbf{v}}_I \rfloor \delta\boldsymbol{\theta} + \delta\mathbf{v}_I \ , \ {}^G\tilde{\mathbf{p}}_f \simeq \lfloor {}^G\hat{\mathbf{p}}_f \rfloor \delta\boldsymbol{\theta} + \delta\mathbf{p}_f \quad (15)$$

Clearly, we can draw the connection between the global-centric ($\tilde{\mathbf{x}}_n$) and invariant ($\delta\mathbf{x}_n$) error states:

$$\begin{bmatrix} \tilde{\boldsymbol{\theta}} \\ {}^G\tilde{\mathbf{p}}_I \\ {}^G\tilde{\mathbf{v}}_I \\ {}^G\tilde{\mathbf{p}}_f \end{bmatrix} \simeq \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \lfloor {}^G\hat{\mathbf{p}}_I \rfloor & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \lfloor {}^G\hat{\mathbf{v}}_I \rfloor & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \lfloor {}^G\hat{\mathbf{p}}_f \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\mathbf{p}_I \\ \delta\mathbf{v}_I \\ \delta\mathbf{p}_f \end{bmatrix} \Leftrightarrow \tilde{\mathbf{x}}_n \simeq \mathbf{A}\delta\mathbf{x}_n \quad (16)$$

Robocentric Formulation The robocentric VINS is reformulated with respect to the local IMU frame $\{I\}$ [8, 9, 26]. It requires some special designs such as incorporating gravity into the state vector and a composition step for shifting the robocentric frames [26], which are beyond this paper's scope. Instead, we focus on the minimal state variables:

$$\mathbf{x}'_n := \left({}^I_G\mathbf{R}, {}^I\mathbf{p}_G, {}^I\mathbf{v}_G, {}^I\mathbf{p}_f\right), \qquad \tilde{\mathbf{x}}_n = \left[\tilde{\boldsymbol{\theta}}'^\top \ {}^I\tilde{\mathbf{p}}_G^\top \ {}^I\tilde{\mathbf{v}}_G^\top \ {}^I\tilde{\mathbf{p}}_f^\top\right]^\top \in \mathbb{R}^{12} \quad (17)$$

Similarly, for the orientation error state we have: ${}^I_G\mathbf{R} = \exp(-\tilde{\boldsymbol{\theta}}'){}^I_G\hat{\mathbf{R}} \simeq (\mathbf{I} - \lfloor\tilde{\boldsymbol{\theta}}'\rfloor){}^I_G\hat{\mathbf{R}}$. The remaining states allow for simple additive error. We then look into the error state between the global-centric ($\tilde{\mathbf{x}}_n$) and robocentric ($\tilde{\mathbf{x}}'_n$):

$$\begin{aligned} {}^I_G\mathbf{R} &= {}^G_I\mathbf{R}^\top && \Rightarrow \ \tilde{\boldsymbol{\theta}} = -{}^I_G\mathbf{R}^\top\tilde{\boldsymbol{\theta}}' \quad (18)\\ {}^G\mathbf{p}_I &= -{}^I_G\mathbf{R}^\top{}^I\mathbf{p}_G && \Rightarrow {}^G\tilde{\mathbf{p}}_I \simeq -\lfloor {}^G\hat{\mathbf{p}}_I \rfloor {}^I_G\hat{\mathbf{R}}^\top\tilde{\boldsymbol{\theta}}' - {}^I_G\hat{\mathbf{R}}^\top{}^I\tilde{\mathbf{p}}_G \\ {}^G\mathbf{v}_I &= -{}^I_G\mathbf{R}^\top{}^I\mathbf{v}_G && \Rightarrow {}^G\tilde{\mathbf{v}}_I \simeq -\lfloor {}^G\hat{\mathbf{v}}_I \rfloor {}^I_G\hat{\mathbf{R}}^\top\tilde{\boldsymbol{\theta}}' - {}^I_G\hat{\mathbf{R}}^\top{}^I\tilde{\mathbf{v}}_G \\ {}^G\mathbf{p}_f &= {}^I_G\mathbf{R}^\top\left({}^I\mathbf{p}_f - {}^I\mathbf{p}_G\right) && \Rightarrow {}^G\tilde{\mathbf{p}}_f \simeq -\lfloor {}^G\hat{\mathbf{p}}_f \rfloor {}^I_G\hat{\mathbf{R}}^\top\tilde{\boldsymbol{\theta}}' + {}^I_G\hat{\mathbf{R}}^\top{}^I\tilde{\mathbf{p}}_f - {}^I_G\hat{\mathbf{R}}^\top{}^I\tilde{\mathbf{p}}_G \end{aligned}$$

We have left out the detailed derivations in this paper. For those, please see our supplementary material [13]. To put in the compact matrix form, we have:

$$\begin{bmatrix} \tilde{\boldsymbol{\theta}} \\ {}^G\tilde{\mathbf{p}}_I \\ {}^G\tilde{\mathbf{v}}_I \\ {}^G\tilde{\mathbf{p}}_f \end{bmatrix} \simeq \begin{bmatrix} -{}^I_G\hat{\mathbf{R}}^\top & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\lfloor {}^G\hat{\mathbf{p}}_I \rfloor {}^I_G\hat{\mathbf{R}}^\top & -{}^I_G\hat{\mathbf{R}}^\top & \mathbf{0}_3 & \mathbf{0}_3 \\ -\lfloor {}^G\hat{\mathbf{v}}_I \rfloor {}^I_G\hat{\mathbf{R}}^\top & \mathbf{0}_3 & -{}^I_G\hat{\mathbf{R}}^\top & \mathbf{0}_3 \\ -\lfloor {}^G\hat{\mathbf{p}}_f \rfloor {}^I_G\hat{\mathbf{R}}^\top & -{}^I_G\hat{\mathbf{R}}^\top & \mathbf{0}_3 & {}^I_G\hat{\mathbf{R}}^\top \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\theta}}' \\ {}^I\tilde{\mathbf{p}}_G \\ {}^I\tilde{\mathbf{v}}_G \\ {}^I\tilde{\mathbf{p}}_f \end{bmatrix} \Leftrightarrow \tilde{\mathbf{x}}_n \simeq \mathbf{B}\tilde{\mathbf{x}}'_n \quad (19)$$

3.3 Remarks: The Unified Persective

For the first time ever, we have analytically shown the transformations between the error states of global-centric ($\tilde{\mathbf{x}}_n$), invariant ($\delta\mathbf{x}_n$) and robocentric ($\tilde{\mathbf{x}}'_n$), demystifying the unknown relationship between the popular VINS estimators:

$$\tilde{\mathbf{x}}_n = \mathbf{A}\delta\mathbf{x}_n = \mathbf{B}\tilde{\mathbf{x}}'_n \ , \quad \delta\mathbf{x}_n = \mathbf{D}\tilde{\mathbf{x}}'_n \tag{20}$$

where $\mathbf{A}$ are shown in Eq. (16) and $\mathbf{B}$ can be found in Eq. (19), while $\mathbf{D}$ involves only rotation matrix blocks:

$$\mathbf{D} = \mathbf{A}^{-1}\mathbf{B} = -{}^{I}_{G}\hat{\mathbf{R}}^{\top} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & -\mathbf{I}_3 \end{bmatrix} \tag{21}$$

Building upon these findings, we can also formulate novel equations for linearized IMU propagation and visual updates, tailored to each formulation. Assume the propagation and update equations with global-centric formulation are expressed as Eq. (6) and (9), we can reframe the linearized propagation and update equations accordingly. To illustrate this process, we present the invariant formulation as a uniform example. As the error state for invariant and global-centric has the relationship $\tilde{\mathbf{x}} = \mathbf{A}\delta\mathbf{x}$, substituting into Eq.(6) and Eq.(9) yields:

$$\delta\mathbf{x}_{k+1} \simeq \mathbf{A}_{k+1}^{-1}\boldsymbol{\Phi}_{k+1,k}\mathbf{A}_k\delta\mathbf{x}_k + \mathbf{A}_{k+1}^{-1}\mathbf{G}_k\mathbf{w}_k \tag{22}$$

$$= \delta\boldsymbol{\Phi}_{k+1,k}\delta\mathbf{x}_k + \delta\mathbf{G}_k\mathbf{w}_k \tag{23}$$

$$\delta\mathbf{r}_k \simeq \mathbf{H}_k\mathbf{A}_k\delta\mathbf{x}_k + \mathbf{n}_k = \delta\mathbf{H}_k\delta\mathbf{x}_k + \mathbf{n}_k \tag{24}$$

The above derivations are applicable to any error-state formulation, offer a unified perspective across different design choices, providing a deep and clear understanding of VINS. This sheds light on a novel estimator design choice and unlocks the potential to flexibly "transfer" between different formulations for specific uses. We will demonstrate their practical use in the following.

4 DES-VINS Estimator Design

We now introduce an alternative consistent estimator design, termed DES-VINS. In our design choice, we adopt the global-centric state, Eq. (10), as it directly represents the physical quantities of the navigation system and offers a straightforward interpretation. Thanks to the DES concept, our selection of the error state is guided by two key considerations: (i) Addressing the primary source of nonlinearity in the system, which comes from orientation. By lifting orientation into the error state, we aim to mitigate issues related to linearization and ensure the observability of the linearized error-state system. (ii) Given the state estimate, we aim to design the error state to facilitate linear update operations with respect to itself to ensure consistent observability between the state and

error state. In line with Eq. (4), the complete state and error state are defined as:

$$\mathbf{x} := (\mathbf{x}_n, \mathbf{x}_b) \ , \qquad \tilde{\mathbf{x}} := (\tilde{\mathbf{x}}_n^*, \tilde{\mathbf{x}}_b^*) \tag{25}$$

$$\mathbf{x}_n := \left({}^G_I\mathbf{R}, {}^G\mathbf{p}_I, {}^G\mathbf{v}_I, {}^G\mathbf{p}_f\right) \ , \qquad \mathbf{x}_b := (\mathbf{b}_g, \mathbf{b}_a) \tag{26}$$

$$\tilde{\mathbf{x}}_n^* := \left[\tilde{\boldsymbol{\theta}}^{*\top} \ \tilde{\mathbf{p}}_I^{*\top} \ \tilde{\mathbf{v}}^{*\top} \ \tilde{\mathbf{p}}_f^{*\top}\right]^\top \in \mathbb{R}^{12} \ , \qquad \tilde{\mathbf{x}}_b^* := \left[\tilde{\mathbf{b}}_g^{*\top} \ \tilde{\mathbf{b}}_a^{*\top}\right]^\top \in \mathbb{R}^6 \tag{27}$$

where the error state $\tilde{\mathbf{x}}_n^*$ is carefully designed as:

$$\tilde{\boldsymbol{\theta}}^* = -{}^I_G\hat{\mathbf{R}}^\top \delta\tilde{\boldsymbol{\theta}}' \ , \ \tilde{\mathbf{p}}_I^* = -{}^I_G\hat{\mathbf{R}}^\top {}^I\tilde{\mathbf{p}}_G \ , \ \tilde{\mathbf{v}}^* = -{}^I_G\hat{\mathbf{R}}^\top {}^I\tilde{\mathbf{v}}_G \tag{28}$$

$$\tilde{\mathbf{p}}_f^* = {}^I_G\hat{\mathbf{R}}^\top {}^I\tilde{\mathbf{p}}_{G\to f} = {}^I_G\hat{\mathbf{R}}^\top ({}^I\tilde{\mathbf{p}}_f - {}^I\tilde{\mathbf{p}}_G) \tag{29}$$

Note that the error state formulation for the IMU presented here can be viewed as multiplying the robocentric error state, $\tilde{\mathbf{x}}'$, with a rotation matrix, $-{}^I_G\hat{\mathbf{R}}^\top$. For the feature, this can be viewed as the error state of the global feature represented in the IMU frame, multiplied by ${}^I_G\hat{\mathbf{R}}^\top$. The correction of the state estimate, utilizing our proposed specialized error state and state formulation, clearly would become a nonlinear operation [see Eq. (3)]. Specifically, the orientation and position are corrected as:

$${}^G_I\hat{\mathbf{R}}^\oplus = \exp(-\tilde{\boldsymbol{\theta}}^*){}^G_I\hat{\mathbf{R}} \ , \quad {}^G\hat{\mathbf{p}}_I^\oplus \simeq {}^G\hat{\mathbf{p}}_I + \lfloor {}^G\hat{\mathbf{p}}_I \rfloor \tilde{\boldsymbol{\theta}}^* + \tilde{\mathbf{p}}_I^* \tag{30}$$

For biases, simple addition is employed (i.e., $\mathbf{b}_g := \hat{\mathbf{b}}_g + \tilde{\mathbf{b}}_g^*$). Velocity and feature state corrections can be similarly derived, as detailed in the supplementary material [13]. For the simplicity and consistency of the presentation, in what follows, we shift our attention to the navigation state. Substituting the above error state, we also identify the following transformation between the global-centric and the DES-VINS formulations:

$$\begin{aligned}
&\tilde{\boldsymbol{\theta}} = -{}^I_G\hat{\mathbf{R}}^\top \delta\tilde{\boldsymbol{\theta}}' && \Rightarrow \ \tilde{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}^* \\
&{}^G\tilde{\mathbf{p}}_I \simeq -\lfloor {}^G\hat{\mathbf{p}}_I \rfloor {}^I_G\hat{\mathbf{R}}^\top \tilde{\boldsymbol{\theta}}' - {}^I_G\hat{\mathbf{R}}^\top {}^I\tilde{\mathbf{p}}_G && \Rightarrow {}^G\tilde{\mathbf{p}}_I \simeq \lfloor {}^G\hat{\mathbf{p}}_I \rfloor \tilde{\boldsymbol{\theta}}^* + \tilde{\mathbf{p}}_I^* \\
&{}^G\tilde{\mathbf{v}}_I \simeq -\lfloor {}^G\hat{\mathbf{v}}_I \rfloor {}^I_G\hat{\mathbf{R}}^\top \tilde{\boldsymbol{\theta}}' - {}^I_G\hat{\mathbf{R}}^\top {}^I\tilde{\mathbf{v}}_G && \Rightarrow {}^G\tilde{\mathbf{v}}_I \simeq \lfloor {}^G\hat{\mathbf{v}}_I \rfloor \tilde{\boldsymbol{\theta}}^* + \tilde{\mathbf{v}}^* \\
&{}^G\tilde{\mathbf{p}}_f \simeq -\lfloor {}^G\hat{\mathbf{p}}_f \rfloor {}^I_G\hat{\mathbf{R}}^\top \tilde{\boldsymbol{\theta}}' + {}^I_G\hat{\mathbf{R}}^\top {}^I\tilde{\mathbf{p}}_f - {}^I_G\hat{\mathbf{R}}^\top {}^I\tilde{\mathbf{p}}_G && \Rightarrow \ {}^G\tilde{\mathbf{p}}_f \simeq \lfloor {}^G\hat{\mathbf{p}}_f \rfloor \tilde{\boldsymbol{\theta}}^* + \tilde{\mathbf{p}}_f^*
\end{aligned} \tag{31}$$

The compact matrix form is given by:

$$\begin{bmatrix} \tilde{\boldsymbol{\theta}} \\ {}^G\tilde{\mathbf{p}}_I \\ {}^G\tilde{\mathbf{v}}_I \\ {}^G\tilde{\mathbf{p}}_f \end{bmatrix} \simeq \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \lfloor {}^G\hat{\mathbf{p}}_I \rfloor & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \lfloor {}^G\hat{\mathbf{v}}_I \rfloor & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \lfloor {}^G\hat{\mathbf{p}}_f \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\theta}}^* \\ \tilde{\mathbf{p}}_I^* \\ \tilde{\mathbf{v}}^* \\ \tilde{\mathbf{p}}_f^* \end{bmatrix} \Leftrightarrow \tilde{\mathbf{x}}_n \simeq \mathbf{C}\tilde{\mathbf{x}}_n^* \tag{32}$$

Remarkably, the matrix $\mathbf{C}$ in (32) is identical to matrix $\mathbf{A}$ in (16), resulting in:

$$\tilde{\mathbf{x}}_n^* \simeq \delta\mathbf{x}_n \tag{33}$$

This result is much intriguing, by realizing that the proposed error state $\tilde{\mathbf{x}}^*$ is not particularly designed in $SE_2(3)$, without concerning the group affinity constraints, but reaches an equivalent invariant formulation [see Eq. (11)], sharing the same propagation and update Jacobian and thus preserving the same linearized system properties (up to same linearization).

4.1 Observability and Consistency

System observability is fundamental to state estimation [28, 35]. VINS is partially observable arising from the nature of the measurements – both the IMU and camera provide only relative information about the sensing platform [24, 31]. It has been discussed that the global-centric formulation, makes the global orientation appear to be observable and thus reduces the nullspace to only *three* d.o.f dimension. This causes the filter to gain extra information, leading to inconsistency, and inaccuracy in estimation results[1].

As such, the observability-constrained (OC)-estimator design has emerged, with the First-Estimates Jacobian (FEJ) methodology [11, 12, 14, 24, 30, 31] being a notable approach, which ensures observability by using the first state estimate for evaluating Jacobians across all time periods and significantly improving VINS accuracy and consistency. In contrast, the invariant and robocentric have the unobservable subspace independent of the state vector. Consequently, these estimators inherently avoid inconsistencies arising from erroneous information affecting the system's unobservable directions due to linearization. Likewise, DES-VINS shares the same state translation and measurement Jacobian matrix as the invariant formulation, thereby enjoying the same observability properties. For more information and derivations, see our supplementary materials [13].

This opens a compelling discussion and prompts thoughtful considerations in estimator design. While the underlying nonlinear system remains the same, the choice of error state representation can significantly influence the properties of the linearized estimator. These variations can have a substantial impact on estimation performance and should be carefully considered in estimator design.

4.2 Efficient Feature Integration

There are two methods to balance efficiency and accuracy in VINS with many features. The first method marginalizes features from the state vector using the MSCKF nullspace projection [38], which is efficient but loses information by treating long-track features as multiple short ones. The second method selectively incorporates long-track features into the state vector as SLAM features, improving accuracy by maintaining correlation across sliding windows.

For the global-centric formulation, a feature exhibiting zero dynamics (i.e., ${}^G\dot{\mathbf{p}}_f = 0$) can be trivially incorporated into the covariance, as it remains uncorrelated with other state variables (i.e., $\boldsymbol{\Phi}_{fb} = \mathbf{0}$ in Eq. (7)). However, in the invariant formulation, these features are correlated with other state variables

[1] An estimator is consistent when its errors are zero-mean (unbiased) and the covariance matrix is equal to that reported by the estimator (see [1], Section 5.4).

during propagation (i.e., $\boldsymbol{\Phi}_{fb} \neq \mathbf{0}$). As discussed in [44], involving l features in global-centric propagation requires $\mathcal{O}(l)$ computation, but $\mathcal{O}(l^2)$ in the invariant formulation, considerably slow down the system in multi-step propagation with high IMU frequency. A straightforward approach is to decouple the feature from Lie group structure, removing the correlation but requiring FEJ the features [44].

Thanks to the unified perspective we have established and the connections derived [see Section 3], we propose a novel method leverages the benefits of the global-centric formulation for efficient propagation while preserving the consistency property of the DES (invariant) formulation without the need to FEJ the features. Given the covariance ${}^{inv}\mathbf{P}_k$ of the DES (invariant) formulation, we first "transfer (propagate)" the covariance to the global-centric formulation, ${}^{gc}\mathbf{P}_k$ [see Eq. (16)]:

$$ {}^{gc}\mathbf{P}_k = \mathbf{A}_k {}^{inv}\mathbf{P}_k \mathbf{A}_k^\top \tag{34} $$

Since the features are uncorrelated with other states for the global-centric formulation, covariance propagation can be computed efficiently:

$$ {}^{gc}\mathbf{P}_{k+1} = \boldsymbol{\Phi}_{k+1,k} {}^{gc}\mathbf{P}_k \boldsymbol{\Phi}_{k+1,k}^\top + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^\top \tag{35} $$

We then "transfer (recover)" the desired covariance through:

$$ {}^{inv}\mathbf{P}_{k+1} = \mathbf{A}_{k+1}^{-1} {}^{gc}\mathbf{P}_{k+1} \mathbf{A}_{k+1}^{-\top} \tag{36} $$

This method, flexible "transition" between different formulations, allows for efficient and consistent incorporation of features by leveraging the benefits of both global-centric and DES (invariant) formulations, will be validated in Section 5.1.

Note that the proposed method still requires $\mathcal{O}(l^2)$ complexity but avoids extra computation in multi-step propagation, maintaining sparsity in feature propagation to minimize overhead. Incorporating features into the state requires $\mathcal{O}(l^3)$ computations during updates, making the $\mathcal{O}(l^2)$ overhead in propagation a worthwhile trade-off. This eliminates the need for OC design, which might introduce unmodeled errors [14].

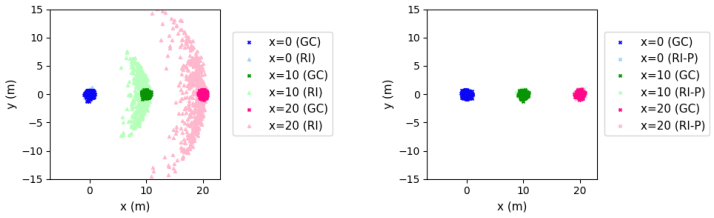

Fig. 1: Visualization of initial 2D position distributions in Euclidean space using invariant (RI, lighter color) and global-centric (GC, darker color) filters. Different colors represent different x positions. The left figure shows RI without covariance propagation, and the right shows RI with propagation (RI-P).

4.3 Estimator Initialization

Successful VINS initialization requires accurate initial state and covariance, via static initialization [20], dynamic initialization [18, 37], or using a pre-built map [21, 45]. Inappropriate initial covariance can cause system drift, a detail that might be overlooked in the literature. In the ensuing discussion, we assume initialization in a map with the initial state estimate $\hat{\mathbf{x}}_0$ and covariance, ${}^{gc}\mathbf{P}_0$, for the global-centric formulation as:

$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, {}^{gc}\mathbf{P}_0), \quad \tilde{\mathbf{x}}_0 = \left[\tilde{\boldsymbol{\theta}}_0^\top \ {}^G\tilde{\mathbf{p}}_{I_0}^\top \ {}^G\tilde{\mathbf{v}}_{I_0}^\top\right]^\top \tag{37}$$

$${}^{gc}\mathbf{P}_0 = E[\tilde{\mathbf{x}}_0\tilde{\mathbf{x}}_0^\top] = \mathtt{diag}({}^{gc}\mathbf{P}_{\boldsymbol{\theta}_0}, {}^{gc}\mathbf{P}_{\mathbf{p}_0}, {}^{gc}\mathbf{P}_{\mathbf{v}_0}) \tag{38}$$

where $E[\cdot]$ is the expected value, ${}^{gc}\mathbf{P}_{\boldsymbol{\theta}_0}$, ${}^{gc}\mathbf{P}_{\mathbf{p}_0}$ and ${}^{gc}\mathbf{P}_{\mathbf{v}_0}$ denote the initial covariance for orientation, position and velocity, respectively. In the global-centric formulation, the initial state and covariance are easily interpreted. For instance, setting ${}^{gc}\mathbf{P}_{\mathbf{p}_0}$ to $0.2 \times 0.2 \times \mathbf{I}_3$ indicates that the initial position error in each direction follows a Gaussian distribution with a 0.2-meter standard deviation. In the invariant formulation, the initial state and covariance are specified as:

$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{X}}_0, {}^{inv}\mathbf{P}_0) \ , \delta\mathbf{x}_0 = \left[\delta\boldsymbol{\theta}_0^\top \ \delta\mathbf{p}_{I_0}^\top \ \delta\mathbf{v}_{I_0}^\top\right]^\top$$
$${}^{inv}\mathbf{P}_0 = E[\delta\mathbf{x}_0\delta\mathbf{x}_0^\top] = \mathtt{diag}({}^{inv}\mathbf{P}_{\boldsymbol{\theta}_0}, {}^{inv}\mathbf{P}_{\mathbf{p}_0}, {}^{inv}\mathbf{P}_{\mathbf{v}_0})$$

However, in the Lie group representation, the same initial covariance does not equate to the same physical uncertainty in Euclidean space. This issue also exists for the DES-VINS. To visualize, in Figure 1, we first set the same initial prior covariance for both global-centric and invariant (i.e., ${}^{gc}\mathbf{P}_0 = {}^{inv}\mathbf{P}_0 = 0.1 \times \mathbf{I}_6$). The initial orientation is set to be identity, while we vary the initial position on the x-axis starting from origin $[0, 0, 0]^\top$ to $[20, 0, 0]^\top$. We then plot the 2D position sample distributions corresponding to different error states, $\tilde{\mathbf{x}}_0$ and $\delta\mathbf{x}_0$. Clearly, when the position is not at the origin, the DES (invariant) will yield a larger sample distribution compared with the global-centric as the distance to the origin increases, which does not correctly depict the initial uncertainty. To address this issue, we establish the relationship between the initial error state covariances as:

$$\begin{aligned} {}^{gc}\mathbf{P}_0 = E[\tilde{\mathbf{x}}_0\tilde{\mathbf{x}}_0^\top] = E[(\mathbf{A}_0\delta\mathbf{x}_0)(\mathbf{A}_0\delta\mathbf{x}_0)^\top] = \mathbf{A}_0 {}^{inv}\mathbf{P}_0\mathbf{A}_0^\top \\ \Rightarrow \quad {}^{inv}\mathbf{P}_0 = \mathbf{A}_0^{-1} {}^{gc}\mathbf{P}_0\mathbf{A}_0^{-\top} \end{aligned} \tag{39}$$

Following this propagation, the initial distributions will align closely with the global-centric representation (Figure 1, right). Drawing on these insights, to easily initialize the system with an invariant formulation on a pre-built map, start by setting the covariance, ${}^{gc}\mathbf{P}_0$, using the global-centric formulation for easy quantification of initial state estimate uncertainty. Then, apply covariance propagation [Eq. (39)] to derive the initial covariance in the invariant formulation. Further discussions are in Section 5.1 .

5 Performance Evaluation

5.1 Numerical Studies

In the numerical study, we evaluate different estimators:

- **Global-centric (GC)**: We leverage OpenVINS [20], a SOTA MSCKF VINS, with and without the FEJ method, denoted as GC-fej and GC-std.
- **Decoupled right-invariant (DRI)**: In line with [44], we implement the right invariant (RI) error state formulation for VINS on top of OpenVINS with decoupled features, denoted as DRI. DRI-fej applies FEJ to SLAM features, while DRI-std does not.
- **DES-VINS**: Our method with decoupled state and error state formulations, which has been proven equivalent to the invariant error state formulations. We consistently incorporate SLAM features as discussed in Section 4.2.

To clarify, the term "decouple" holds a dual meaning in our context. Following [44], in the decoupled-right invariant (DRI), decouple denotes the separation of SLAM features from the Lie group structure. Meanwhile, in DES-VINS, it means the independent and separate handling of error and state representations. Although DES-VINS is mathematically equivalent to the invariant formulation, we aim to conduct a comprehensive numerical study of different estimators under extreme conditions, such as 8-pixel noise, to fully assess their strengths and weaknesses. These studies spark interesting discussions beyond the results themselves, as detailed in Section 6 . We also evaluate efficiency and estimator initialization to highlight the importance and utility of the proposed DES design concept. We didn't consider the robocentric formulation due to its complexity and scope limitations, but it is an interesting possibility for future investigation.

We generate realistic visual-bearing and inertial measurements, with simulation parameters detailed in Table 1. The simulated trajectory can be found in the supplementary material [13]. Our reported metrics include Absolute Trajectory Error (ATE), Normalized Estimation Error Squared (NEES), and Average NEES (ANEES) [1, 46], where the ANEES is calculated as: $\text{ANEES} := \frac{1}{6M}\sum_{i=1}^{M}[(\mathbf{x}_{gt}^{i} \boxminus \hat{\mathbf{x}}^{i})^{\top}\mathbf{P}^{-1}(\mathbf{x}_{gt}^{i} \boxminus \hat{\mathbf{x}}^{i})]$, $\mathbf{x}_{gt}$ denotes the ground truth. For the RI, the estimation error is computed as: $\mathbf{x}_{gt} \boxminus \hat{\mathbf{x}} := \log\left(\mathbf{X}_{gt}\hat{\mathbf{X}}^{-1}\right)$, where $\mathbf{X} \in SE(3)$ includes orientation and position. For the GC formulation, the estimation error is defined as: $\mathbf{x}_{gt} \boxminus \hat{\mathbf{x}} = [\log\left(\mathbf{R}_{gt}\hat{\mathbf{R}}^{\top}\right)^{\top} \quad (\mathbf{p}_{gt} - \hat{\mathbf{p}})^{\top}]^{\top}$. For an estimator to be considered consistent, the magnitude of NEES should align with the 3 d.o.f. for orientation and position and its ANEES should be 1.

Sensitivity to Noise We challenge the estimators with increased inertial and camera noise, but due to space limits, only results for camera noise are presented, shown in Figure 2; additional results are in our supplementary material [13]. Clearly, when the measurement noise is small, all formulations exhibit comparable performance. However, as noise levels increase, leading to a decline in

estimation accuracy due to accumulative errors, the distinctions among the formulations become more apparent. Overall, formulations with consistency properties (GC-fej, DRI-fej, and DES-VINS) outperform those lacking consistency (GC-std and DRI-std), as demonstrated by lower ATE values and more ideal ANEES. Remarkably, FEJ-based estimation methods maintain superior performance over non-consistent approaches, even with 8-pixel measurement noise and potential inaccuracies in initial state estimates, such as triangulated feature positions. Among all, DES-VINS demonstrates the best performance, thanks to its capacity to maintain system observability without modifying the Jacobian.

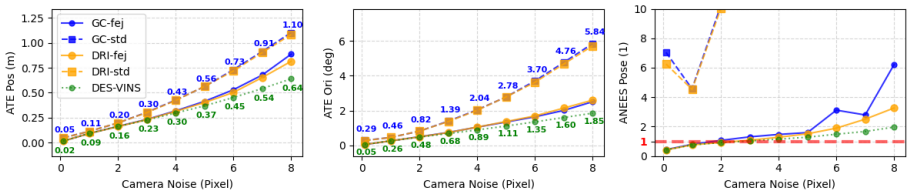

Fig. 2: Estimation ATE (left, middle) and ANEES (right) under varying camera measurement noise with different estimator formulations averaged from 200 runs. An ideal ANEES for a consistent estimator is 1. The ANEES value over 10 are truncated for clarity. The ATE figures only display values for the worst (blue: GC-std) and the best performance (green: DES-VINS).

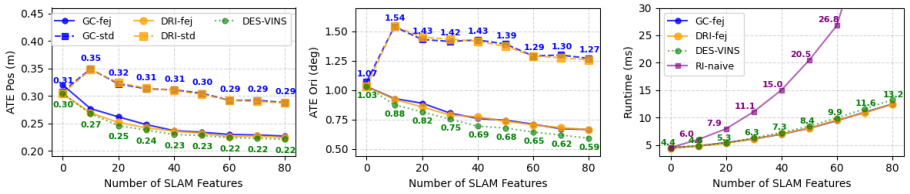

Fig. 3: Comparative estimation ATE (left, middle) and runtime (right) for different estimator formulations with different numbers of SLAM features based on 200 runs. ATE results omit RI-naive due to the same performance as DES-VINS; runtime shows only FEJ for each formulation, as they are similar to standard versions. The ATE figures only display values for the worst (blue: GC-std) and the best performance (green: DES-VINS), while the runtime plot shows values for the slowest (purple: RI-naive) and the proposed one (green: DES-VINS).

Efficiency We next examine the performance impact of incorporating SLAM features into the state with particular attention to efficiency, as detailed in Figure 3. By default, our simulations include 200 features (see Table 1). We report system performance using a range of 0 to 80 SLAM features, while processing all remaining features as MSCKF features. In these experiments, we set the camera measurement noise to 3 pixels to accentuate the performance differences. Generally, estimators that maintain consistency yield better results. Furthermore, the increase in SLAM features enhances estimation accuracy, showing their benefit

to the system. More importantly, the timing is reported in the right subfigure of Figure 3. As discussed in Section 4.2, naively incorporating SLAM features into the RI formulation significantly increases computational load. We include 'RI-naive', which propagate with SLAM features directly and 'DES-VINS' uses our proposed method for efficient propagation (Section 4.2). As expected, Naive incorporation results in a substantial increase in computation time, becoming approximately 2.7× slower with 60 SLAM features, and 4× slower with 80 SLAM features, compared to DES-VINS. On the other hand, in comparison to the SOTA invariant method (i.e., DRI-fej), the proposed DES-VINS is theoretically more rigorous without relying on the FEJ approximation, all while maintaining minimal efficiency costs, achieving 12% accuracy improvement with only a 6% in computational overhead [see supplementary material for complete results [13]].

Table 1: Simulation parameters

Parameter	Value	Parameter	Value
Gyro. White Noise	1.6968e-4	Gyro. Rand. Walk	1.9393e-5
Accel. White Noise	2.0000e-3	Accel. Rand. Walk	3.0000e-3
Cam Freq. (Hz)	10	IMU Freq. (Hz)	400
Num. Clones	11	Tracked Feat.	200

Table 2: Estimation performance

Pos. (m)	Estimator	Cov Prop?	ATE(deg/m)	NEES(3)
0	GC	✗	0.474 / 0.163	2.412 / 1.501
	RI	✗	0.471 / 0.156	2.393 / 1.344
1000	GC	✗	0.474 / 0.163	2.410 / 1.499
	RI	✗	**0.941 / 2.665**	**5.143 / 12.525**
	RI	✓	0.472 / 0.155	2.392 / 1.343

Initialization We examined the impact of initial covariances in two scenarios: starting at the origin, common in static initialization, and with the initial position shifted by 1 km in each axis, typical for systems initializing in a pre-built map. The state estimates are initialized using ground truth data. The results are reported in Table 2, where "Cov Prop" denotes if the extra covariance propagation is applied [See Eq.(39)]. As expected, the estimation performance of GC will not be affected by the initial position since its error state can be correctly represented by the initial covariance. For invariant formulation (RI), initializing with a large position and naively setting the same small initial covariance - due to ground truth system initialization - is shown to be inaccurate and inconsistent with large ATE and NEES values. This issue can be resolved by applying covariance propagation, Eq.(39), to establish an appropriate initial covariance.

5.2 Real-World Experiments

Table 3: Absolute trajectory error (ATE) for each formulation in degrees/cm.

	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
GC-std [20]	1.92 / 10.0	1.32 / 15.1	1.80 / 22.1	0.99 / 19.8	0.92 / 34.3	0.79 / 6.3	1.87 / 6.8	2.47 / 8.7	1.00 / 10.9	1.88 / 8.1	1.11 / 23.3
DRI-std [44]	1.73 / 10.0	1.22 / 14.3	1.69 / 21.7	1.11 / 20.3	1.00 / 32.5	0.81 / 5.5	1.91 / 6.7	2.40 / 8.4	0.86 / 11.1	1.79 / 8.2	1.18 / 21.6
GC-fej [20]	1.43 / 9.6	0.89 / 14.0	1.61 / 20.0	1.04 / 15.0	0.96 / 26.8	0.53 / 4.8	1.83 / 6.5	1.91 / 6.6	0.75 / 14.0	1.65 / 7.6	1.41 / 22.7
DRI-fej [44]	1.50 / 9.8	0.73 / 12.1	1.68 / 20.5	1.02 / 16.6	0.76 / 31.2	0.59 / 4.9	1.81 / 6.5	1.99 / 6.0	0.74 / 10.8	1.60 / 8.0	1.45 / 22.4
DES-VINS	1.56 / 9.2	0.75 / 12.8	1.74 / 20.9	1.06 / 15.5	0.71 / 27.7	0.62 / 5.4	1.76 / 6.4	1.99 / 6.2	0.75 / 10.7	1.62 / 7.6	1.41 / 23.3

We further evaluate and compare different estimator formulations with real-world Euroc Mav dataset [6], presented in Table 3. In our experiments, we maintained 11 clones and a maximum of 50 SLAM features in the state. For other tracked features, we conducted MSCKF updates, where the max number of MSCKF updates is 40. The results show that estimators designed with consistency property outperform their inconsistent ones as evidenced by lower ATE values. Along with the simulation results, this underscores the critical importance of maintaining system consistency. However, we also observed that it

occurs to be more sensitive to outliers, which is currently rejected by the Mahalanobis gating test, where differences in linearization points can make an impact.

6 Beyond the Results: A Dissuasion

6.1 Ensuring Correct Observability Properties

First of all, both the simulation and experimental results underscore the significance of assuring estimator consistency (in terms of NEES), as only an estimator that is consistent and can provide trustworthy results in practice. Thanks to the DES concept and established connections between estimators, DES-VINS efficiently and consistently integrates SLAM features with theoretical rigor. Numerical studies indicate that DES-VINS offers superior accuracy, particularly in high-noise environments, due to its ability to maintain observability without modifying the measurement Jacobian. However, in real-world experiments, all the consistent estimator design choices demonstrated comparable and similar performance. This similar performance likely arises from negligible differences between the first and current estimates in environments with low measurement noise, where, without iterative updates, the first estimate is not inherently worse.

6.2 Understanding Error States

For real systems, understanding and thus interpreting error states of an estimator is of practical importance but could be challenging in some formulations (e.g., invariant). For example, when representing error states and thus uncertainty for position and velocity, it is more natural to understand them in $\mathbb{R}^3$. We have shown that careful initialization with proper initial covariance is crucial and should be considered when evaluating estimators. In our numerical studies, we calculate NEES for the invariant formulation within the $SE(3)$ group. However, for practical applications like real-time obstacle avoidance or data association, it is often more practical to use the covariance from the global-centric formulation. This can be easily achieved through covariance propagation, as explained in Section 4.3 , thanks to the flexible transformation between formulations.

6.3 Dirty Laundry

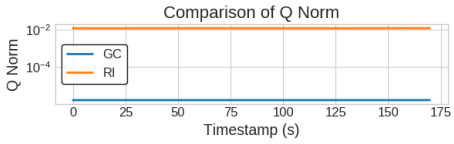

Fig. 4: Comparison of $||\mathbf{Q}||_F$ between global-centric (GC) and invariant (RI) formulations over time.

No free lunch. While the invariant formulation does not have an issue of system observability related to linearization, we did find some potential caveat

about the noise covariance (which is not revealed yet in our numerical studies though). Specifically, a notable distinction arises in the noise Jacobian for the invariant formulation [i.e., $\delta\mathbf{G}$ in Eq. (23)] and for the global-centric formulation [i.e., $\mathbf{G}$ in Eq. (6)]. In the $\delta\mathbf{G}$, the global state estimates (${}^{G}\mathbf{p}_{I}$) are included, whereas in $\mathbf{G}$, all states are relative states (i.e., $\Delta\mathbf{p} = {}^{G}\mathbf{p}_{I_{k+1}} - {}^{G}\mathbf{p}_{I_k}$). Please refer to our supplementary material [13] for the analytical noise Jacobians. This results in significant differences in the magnitude of the noise Jacobian and thus the propagated noise covariance, $\mathbf{Q}_{k+1} = \delta\mathbf{G}_k\mathbf{Q}_k\delta\mathbf{G}_k^{\top}$, especially for large-scale outdoor scenarios where the positions may grow unbound. As depicted in Figure 4, where we utilized the same trajectory as the one used for testing initialization with a shifted initial position, it is evident that the Frobenius norm $||\mathbf{Q}||_F$ for the invariant is significantly larger than that for the global-centric. This result however suggests that the IMU measurements may have much greater uncertainty in the invariant formulation than the global-centric. Also, the global state in the invariant noise Jacobian (i.e., ${}^{G}\mathbf{p}_{I}$) might have larger errors than the relative state (i.e., $\Delta\mathbf{p}$) for the global-centric. Although this potential caveat has not shown any symptom yet in our finite studies, we will investigate it further.

It is important to note that linear observability analysis concerns the corresponding deterministic linearized system only, without considering the stochastic noise. If choosing a different representation of errors from the states of the underlying physical systems, it remains unclear that proper observability of the linearized error-state system can ensure proper estimability of the states, which we will study further in the future.

7 Conclusion and Future Work

In this paper, we have formalized a general DES estimator design methodology, which does not require the state and error state to share the same representation and instead can be decoupled from each other. With this methodology, we have unified different popular VINS estimators including the global/robo-centric and invariant formulations by analytically deriving transformations between them and showing the capability to flexibly transfer different formulations. This offers a fresh perspective in estimator design, allowing to harness distinct advantages from various formulations tailored to specific purposes, thereby enhancing overall performance. With the application to VINS, we have developed a new DES-VINS estimator, utilizing global-centric state vectors and a novel error state formulation. Although fundamentally different, it has been proven equivalent to the invariant formulation, even without adhering to the design constraints of group-affine properties. Thanks to the unified perspective, DES-VINS efficiently integrates long-tracked SLAM features — achieving speeds $3\times$ faster compared to the invariant formulation — while ensuring consistency. Additionally, we show an easy method for proper system initialization. Extensive numerical studies and real-world experiments are presented to evaluate the estimator performances with different formulations sparking fruitful discussions. Looking forward, we see great promise in systematically developing formulations tailored to specific requirements, a strategy applicable to all nonlinear estimators.

Bibliography

[1] Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with applications to tracking and navigation: theory algorithms and software. John Wiley & Sons (2001)

[2] Barfoot, T.D.: State Estimation for Robotics: Second Edition. Cambridge University Press, 2 edn. (2024)

[3] Barrau, A., Bonnabel, S.: An EKF-SLAM algorithm with consistency properties. CoRR **abs/1510.06263** (2015), http://arxiv.org/abs/1510.06263

[4] Barrau, A., Bonnabel, S.: The invariant extended kalman filter as a stable observer. IEEE Transactions on Automatic Control **62**(4), 1797–1812 (2016)

[5] Bonnabel, S.: Symmetries in observer design: review of some recent results and applications to ekf-based slam. CoRR **abs/1105.2254** (2011), http://arxiv.org/abs/1105.2254

[6] Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R.: The euroc micro aerial vehicle datasets. The International Journal of Robotics Research (2016)

[7] Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Transactions on Robotics **37**(6), 1874–1890 (2021)

[8] Castellanos, J.A., Martinez-Cantin, R., Tardós, J.D., Neira, J.: Robocentric map joining: Improving the consistency of ekf-slam. Robotics and autonomous systems **55**(1), 21–29 (2007)

[9] Castellanos, J.A., Neira, J., Tardós, J.D.: Limits to the consistency of ekf-based slam. IFAC Proceedings Volumes **37**(8), 716–721 (2004)

[10] Chen, C., Geneva, P., Peng, Y., Lee, W., Huang, G.: Monocular visual-inertial odometry with planar regularitiee. In: Proc. of the IEEE International Conference on Robotics and Automation. London, UK. (2023)

[11] Chen, C., Geneva, P., Peng, Y., Lee, W., Huang, G.: Optimization-based vins: Consistency, marginalization, and fej. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2023)

[12] Chen, C., Peng, Y., Huang, G.: Fast and consistent covariance recovery for sliding-window optimization-based vins. In: Proc. International Conference on Robotics and Automation. Yokohama, Japan (May 2024)

[13] Chen, C., Peng, Y., Huang, G.: Supplementary materials: Visual-inertial state estimation with decoupled error and state representations. Tech. rep., University of Delaware (2024), https://udel.edu/~ghuang/papers/tr_des.pdf

[14] Chen, C., Yang, Y., Geneva, P., Huang, G.: FEJ2: A consistent visual-inertial state estimator design. In: International Conference on Robotics and Automation (ICRA). Philadelphia, USA (2022)

[15] Chen, C., Yang, Y., Geneva, P., Lee, W., Huang, G.: Visual-inertial-aided online mav system identification. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2022)
[16] Chirikjian, G.S.: Stochastic models, information theory, and Lie groups, volume 2: Analytic methods and modern applications, vol. 2. Springer Science & Business Media (2011)
[17] Dong-Si, T.C., Mourikis, A.I.: Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In: 2011 IEEE International Conference on Robotics and Automation. pp. 5655–5662. IEEE (2011)
[18] Dong-Si, T.C., Mourikis, A.I.: Estimator initialization in vision-aided inertial navigation with unknown camera-imu calibration. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1064–1071. IEEE (2012)
[19] Fornasier, A., Ge, Y., van Goor, P., Mahony, R., Weiss, S.: Equivariant symmetries for inertial navigation systems. arXiv preprint arXiv:2309.03765 (2023)
[20] Geneva, P., Eckenhoff, K., Lee, W., Yang, Y., Huang, G.: OpenVINS: a research platform for visual-inertial estimation. In: Proc. of the IEEE International Conference on Robotics and Automation. Paris, France (2020), https://github.com/rpng/open_vins
[21] Geneva, P., Huang, G.: Map-based visual-inertial localization: A numerical study. In: International Conference on Robotics and Automation (ICRA). Philadelphia, USA (2022)
[22] Goor, P.v., Mahony, R.: An equivariant filter for visual inertial odometry. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 14432–14438 (2021). https://doi.org/10.1109/ICRA48506.2021.9561769
[23] Heo, S., Park, C.G.: Consistent ekf-based visual-inertial odometry on matrix lie group. IEEE Sensors Journal **18**(9), 3780–3788 (2018). https://doi.org/10.1109/JSEN.2018.2808330
[24] Hesch, J.A., Kottas, D.G., Bowman, S.L., Roumeliotis, S.I.: Consistency analysis and improvement of vision-aided inertial navigation. IEEE Transactions on Robotics **30**(1), 158–176 (2013)
[25] Huai, J., Lin, Y., Zhuang, Y., Shi, M.: Consistent right-invariant fixed-lag smoother with application to visual inertial slam. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 6084–6092 (May 2021), https://ojs.aaai.org/index.php/AAAI/article/view/16758
[26] Huai, Z., Huang, G.: Robocentric visual-inertial odometry. International Journal of Robotics Research (Apr 2019)
[27] Huai, Z., Huang, G.: Square-root robocentric visual-inertial odometry with online spatiotemporal calibration. IEEE Robotics and Automation Letters **7**(4), 9961–9968 (2022)
[28] Huang, G.: Improving the Consistency of Nonlinear Estimators: Analysis, Algorithms, and Applications. Ph.D. thesis, Department of Computer Science and Engineering, University of Minnesota (2012), https://conservancy.umn.edu/handle/11299/146717

[29] Huang, G.: Visual-inertial navigation: A concise review. In: Proc. International Conference on Robotics and Automation. Montreal, Canada (May 2019)
[30] Huang, G., Mourikis, A.I., Roumeliotis, S.I.: A first-estimates Jacobian EKF for improving SLAM consistency. In: Proc. of the 11th International Symposium on Experimental Robotics. Athens, Greece (Jul 2008)
[31] Huang, G., Mourikis, A.I., Roumeliotis, S.I.: Observability-based rules for designing consistent EKF SLAM estimators. International Journal of Robotics Research **29**(5), 502–528 (Apr 2010). https://doi.org/10.1177/0278364909353640
[32] Leutenegger, S.: Okvis2: Realtime scalable visual-inertial slam with loop closure. arXiv preprint arXiv:2202.09199 (2022)
[33] Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual–inertial odometry using nonlinear optimization. The International Journal of Robotics Research **34**(3), 314–334 (2015)
[34] Li, M., Mourikis, A.I.: High-precision, consistent ekf-based visual-inertial odometry. The International Journal of Robotics Research **32**(6), 690–711 (2013)
[35] Martinelli, A.: Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. IEEE Transactions on Robotics **28**(1), 44–60 (2011)
[36] Maybeck, P.S.: Stochastic Models, Estimation, and Control, vol. 1. Academic Press, London (1979)
[37] Merrill, N., Geneva, P., Katragadda, S., Chen, C., Huang, G.: Fast monocular visual-inertial initialization leveraging learned single-view depth. In: Proc. Robotics: Science and Systems (RSS). Daegu, Republic of Korea (Jul 2023)
[38] Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided inertial navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 3565–3572. Rome, Italy (Apr 10–14, 2007)
[39] Peng, Y., Chen, C., Huang, G.: Quantized visual-inertial odometry. In: Proc. International Conference on Robotics and Automation. Yokohama, Japan (May 2024)
[40] Peng, Y., Chen, C., Huang, G.: Ultrafast square-root filter-based VINS. In: Proc. International Conference on Robotics and Automation. Yokohama, Japan (May 2024)
[41] Qin, T., Li, P., Shen, S.: VINS-Mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics **34**(4), 1004–1020 (2018)
[42] Usenko, V., Demmel, N., Schubert, D., Stückler, J., Cremers, D.: Visual-inertial mapping with non-linear factor recovery. IEEE Robotics and Automation Letters **5**(2), 422–429 (2019)
[43] Wu, K., Zhang, T., Su, D., Huang, S., Dissanayake, G.: An invariant-ekf vins algorithm for improving consistency. pp. 1578–1585 (Sept 2017)
[44] Yang, Y., Chen, C., Lee, W., Huang, G.P.: Decoupled right invariant error states for consistent visual-inertial navigation. IEEE Robotics and Automation Letters (2022)

[45] Zhang, Z., Jiao, Y., Huang, S., Xiong, R., Wang, Y.: Map-based visual-inertial localization: Consistency and complexity. IEEE Robotics and Automation Letters **8**(3), 1407–1414 (2023)
[46] Zhang, Z., Scaramuzza, D.: A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7244–7251. IEEE (2018)
[47] Zheng, X., Li, M., Mourikis, A.I.: Decoupled representation of the error and trajectory estimates for efficient pose estimation. In: Robotics: Science and Systems (2015)