Optimization-based VINS: Consistency, Marginalization, and FEJ

Chuchu Chen, Patrick Geneva, Yuxiang Peng, Woosik Lee, and Guoquan Huang

Abstract— **In this work, we present a comprehensive analysis of the application of the First-estimates Jacobian (FEJ) design methodology in nonlinear optimization-based Visual-Inertial Navigation Systems (VINS). The FEJ approach fixes system linearization points to preserve proper observability properties of VINS and has been shown to significantly improve the estimation performance of state-of-the-art filtering-based methods. However, its direct application to optimization-based estimators holds challenges and pitfalls, which we addressed in this paper. Specifically, we carefully examine the observability and its relation to inconsistency and FEJ, based on this, we explain how to properly apply and implement FEJ within four marginalization archetypes commonly used in non-linear optimization-based frameworks. FEJ's effectiveness and applications to VINS are investigated and demonstrate significant performance improvements. Additionally, we offer a detailed discussion of results and guidelines on how to properly implement FEJ in optimization-based estimators.**

I. Introduction and Related Work

Extensive research has been devoted to the visual-inertial 3D motion tracking problem due to its prevalence in enabling consumer experiences (e.g. AR/VR) and robotic autonomy [1], [2]. Visual-Inertial Navigation Systems (VINS) address this problem by taking advantage of a high-frequency inertial measurement unit (IMU) and information-dense camera to estimate the 6 degree-of-freedom (DoF) pose and its corresponding uncertainty [3]. There are two families of VINS estimators generally: 1) efficient filter-based methods which only perform linearization once, and 2) optimization-based methods which formulate a nonlinear least-squares (NLS) problem with all available measurements. The latter performs relinearization with a higher computational burden.

A common visualization technique in the optimization-based method is the "factor graph" [see Figure 1]. Factor graphs represent states as nodes and measurements as edges that connect to their involved states [4], [5]. Significant research efforts have been concentrated on reducing the computational complexity through methods such as inertial preintegration [6]–[8], incremental smoothing [9], [10], sliding-window or keyframing marginalization schemes [11]–[15], or information sparsification methods [16], [17]. Optimization-based VINS continue to grow in popularity given their capability to decrease errors through relinearization and the aforementioned methods aimed at boosting efficiency.

An inherent challenge to VINS is preventing information gain in the unobservable state space corresponding to the 4 DoF global yaw and position of the platform [18], [19]. However, due to the *nonlinear* nature of VINS, which requires *linearization* of both the system dynamics and measurement functions, discrepancies between sequential linearization points causes information to be mistakenly gained along unobservable directions, leading to over-confident uncertainty estimates, reduced accuracy, and inconsistent estimation [18].

To address this issue within the filter-based context, "observability-aware" estimators have been proposed, such as robocentric state representation [20], [21], invariant error state formulation [22], [23], observability-constrained (OC) [18], [24], and First-estimates Jacobian (FEJ) methodology [19], [25], [26]. The FEJ technique has become popular for its simplicity and significant performance improvement compared to others. However, in the literature, the consistency and application of FEJ to optimization-based VINS estimators have not been extensively explored. This is mainly because identifying the right state variables to be fixed is challenging as it closely relates to VINS observability and the leveraged state marginalization procedure.

Several research efforts have attempted to integrate observability-aware methodologies into optimization-based VINS. For example, the Prior-Linearization (PL) fixed-lag smoother was presented by Dong-Si and Mourikis [27] leveraged the "prior" estimates (i.e., the "first-estimates" at marginalization) when evaluating Jacobians involved with the marginal prior factor to address the linearization discrepancy. Huang et al. [28] extended this work to leverage the observability-constrained (OC) method and reduced potential linearization errors caused by FEJ, while enforcing that the 4 DoF nullspace remains valid. Other recent works, such as OKVIS [12], [29] and Basalt [17] have applied FEJ to the marginalized prior factor to avoid the erroneous accumulation of information, but do not modify the linearization points of measurements involved with remaining states (i.e. the IMU preintegration and camera observations connected to the states linked to marginal prior factor) nor evaluate system consistency.

The use of the invariant state $\mathbb{SE}_2(3)$ representation [30], has attracted attention due to its ability to satisfy observability properties without special modification. It has been argued that using invariant state representations with anchored features ensures consistency and enables consistent estimation [31]–[34]. However, due to the specialized nature of the invariant state space, it is difficult to leverage arbitrary state representations and maintain consistency (e.g. *global* feature representation [23] require special care).

The VI-DSO [35] and DM-VIO [36] have also paid particular attention to the use of FEJ during marginalization. DM-VIO introduced the concept of "delayed marginalization" which improves traditional marginalization by allowing for

This work was partially supported by the University of Delaware (UD) College of Engineering, the NSF (IIS-1924897, SCH-2014264), and Google ARCore. Geneva is also partially supported by the NASA DE Space Grant Graduate Fellowship.

The authors are with the Robot Perception and Navigation Group (RPNG), University of Delaware, Newark, DE 19716, USA. Email: {ccchu,pgeneva,yxpeng,woosik,ghuang}@udel.edu

the relinearization of typically marginalized states which have not converged. While these works have shown promising directions for addressing the consistency of optimization-based VINS, a detailed numerical study of how FEJ relates to observability, evaluation of the consistency of the state estimate, how different marginalization strategies impact consistency, and how to apply FEJ within the factor-graph context is missing from the literature.

In this study, we conduct a thorough analysis to shed light on the impacts of integrating the FEJ methodology into optimization-based VINS. In particular, we explore the application of FEJ with different marginalization strategies and under different sensor noise levels. We also offer a practical and detailed guide for researchers and engineers to effectively apply FEJ methodology to their systems. The primary contributions of our work include:

- A comprehensive analysis of the significance and effectiveness of incorporating FEJ methodology into optimization-based VINS algorithms.
- We investigate and evaluate the performance of four state-of-the-art marginalization archetypes most commonly used in optimization-based VINS.
- Our numerical studies include an in-depth examination to determine the optimal strategy for applying FEJ to different scenarios and provide a detailed guide.

II. Optimization-Based VINS

We formulate the NLS problem over the entire trajectory up to the current time t_k. The system state consists of the current navigation states, $\mathbf{x}_k$, and 3D features, $\mathbf{x}_f$:

$$\mathbf{x}_{0:k} = \begin{bmatrix}\mathbf{x}_0^\top & \dots & \mathbf{x}_k^\top & \mathbf{x}_f^\top\end{bmatrix}^\top \tag{1}$$

$$\mathbf{x}_k = \begin{bmatrix}{}^{I_k}_{G}\bar{q}^\top & {}^{G}\mathbf{p}_{I_k}^\top & {}^{G}\mathbf{v}_{I_k}^\top & \mathbf{b}_g^\top & \mathbf{b}_a^\top\end{bmatrix}^\top \tag{2}$$

$$\mathbf{x}_f = \begin{bmatrix}{}^{G}\mathbf{f}_1{}^\top & \dots & {}^{G}\mathbf{f}_g^\top\end{bmatrix}^\top \tag{3}$$

where ${}^{I}_{G}\bar{q}$ is the unit quaternion[1] that represents the rotation ${}^{I}_{G}\mathbf{R}$ from global frame $\{G\}$ to the IMU frame $\{I\}$; ${}^{G}\mathbf{p}_I$ and ${}^{G}\mathbf{v}_I$ are the IMU position and velocity in $\{G\}$, respectively; $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases; and the feature state, $\mathbf{x}_f$, comprises the global position of g landmarks.

A. Batch-MAP Formulation

At timestep t_k, the *batch maximum a posteriori* (MAP) seeks to solve for the history of the state estimate $\mathbf{x}_{0:k}$ by maximizing the posterior PDF leveraging: 1) prior information $\mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$, 2) IMU motion constraints $\mathbf{u}$, and 3) camera observation measurements $\mathbf{z}$:

$$p \propto p(\mathbf{x}_0) \prod_{i=0}^{k-1} p(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{u}_k) \prod_{\mathbf{z}_{i,j} \in \mathcal{Z}_{0:k}} p(\mathbf{z}_{i,j}|\mathbf{x}_i, \mathbf{f}_j) \tag{4}$$

where the set $\mathcal{Z}_{0:k}$ denotes all measurements between $[t_0, t_k]$. Under the Gaussian distribution assumption, maximizing the above PDF is equivalent to minimizing:

$$\mathcal{C}(\mathbf{x}_{0:k}) = \mathcal{C}_{p_0} + \sum_{i=0}^{k-1} \mathcal{C}_{I_i} + \sum_{\mathbf{z}_{i,j} \in \mathcal{Z}_{0:k}} \mathcal{C}_{f_{ij}} \tag{5}$$

where we define the following costs (we refer the reader to the technical report for detailed equations [38]):

$$\text{Prior:} \quad \mathcal{C}_{p_0} = \frac{1}{2}||\mathbf{x}_0 \boxminus \hat{\mathbf{x}}_0||^2_{\mathbf{P}_0} \tag{6}$$

$$\text{Inertial: [7]} \quad \mathcal{C}_{I_i} = \frac{1}{2}||\mathbf{x}_{i+1} \boxminus \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)||^2_{\mathbf{Q}_i} \tag{7}$$

$$\text{Camera: [39]} \quad \mathcal{C}_{f_{ij}} = \frac{1}{2}||\mathbf{z}_{ij} \boxminus \mathbf{h}(\mathbf{x}_i, \mathbf{f}_j)||^2_{\mathbf{R}_{ij}} \tag{8}$$

where $||\mathbf{a}||^2_{\mathbf{W}} := \mathbf{a}^\top \mathbf{W}^{-1} \mathbf{a}$ and can be solved iteratively given an initial linearization point. The second-order Taylor series of the l-th iteration with linearization point $\hat{\mathbf{x}}^l_{0:k}$ is:

$$\mathcal{C}(\hat{\mathbf{x}}^l_{0:k} \boxplus \delta\mathbf{x}^l_{0:k}) \simeq \mathcal{C}(\hat{\mathbf{x}}^l_{0:k}) + \mathbf{b}^{l^\top} \delta\mathbf{x}^l_{0:k} + \frac{1}{2}\delta\mathbf{x}^{l^\top}_{0:k} \mathbf{A}^l \delta\mathbf{x}^l_{0:k}$$

where $\delta\mathbf{x}^l_{0:k}$ is the state correction, $\mathbf{A}^l$ the Hessian, and $\mathbf{b}^l$ the gradient. We thus have the updated state $\hat{\mathbf{x}}^{l+1}_{0:k}$ via:

$$\mathbf{A}^l \delta\mathbf{x}^l_{0:k} = -\mathbf{b}^l \quad \Rightarrow \quad \hat{\mathbf{x}}^{l+1}_{0:k} = \hat{\mathbf{x}}^l_{0:k} \boxplus \delta\mathbf{x}^l_{0:k} \tag{9}$$

Note that Gauss-Newton approximates the Hessian $\mathbf{A}^l$ using the first derivative [40].

III. Marginalization and Consistency

Ideally, as the robot moves through the environment and observes new features, one could reformulate the batch-MAP problem and solve the objective function, Eq. (5), using all measurements. However, as the state size grows larger, this quickly becomes prohibitively expensive (complexity is $O(n^3)$ in terms of state size) for real-time estimation. It thus becomes necessary to apply state marginalization techniques to bound computational complexity.

This marginalization directly leads to problematic inconsistencies as states become permanently approximated and fixed (i.e. the marginalized states) as the observability properties deviate from the batch-MAP estimator [27]. The estimator erroneously believes it has gained information along directions that it can not measure, resulting in overconfident estimates. This inconsistency can be a significant problem, degrading accuracy and making the estimator unreliable. In the following sections, we will explain the inconsistency problem arising from state marginalization.

A. State Marginalization

The graph is partitioned into three sets: the to-be-marginalized states $\mathbf{x}_M$, the remaining states connected to $\mathbf{x}_M$, which are referred to as $\mathbf{x}_R$ (also known as the Markov blanket), and the new states not involved, denoted as $\mathbf{x}_N$. Before marginalization, the MAP cost $\mathcal{C}(\mathbf{x}_{0:k})$ at t_k is:

$$\mathcal{C}(\mathbf{x}_M, \mathbf{x}_R, \mathbf{x}_N) = \mathcal{C}_{mr}(\mathbf{x}_M, \mathbf{x}_R) + \mathcal{C}_{rn}(\mathbf{x}_R, \mathbf{x}_N) \tag{10}$$

where $\mathcal{C}_{mr}$ contains all costs involving $\mathbf{x}_M$ and $\mathbf{x}_R$, and $\mathcal{C}_{rn}$ contains all costs involving $\mathbf{x}_R$ and $\mathbf{x}_N$. As $\mathbf{x}_M$ and $\mathbf{x}_N$ have no joint costs, the minimization of the total cost is:

[1] Throughout the paper, $\hat{\mathbf{x}}$ is used to denote the *current* best estimate of a random variable $\mathbf{x}$ with $\delta\mathbf{x} = \mathbf{x} \boxminus \hat{\mathbf{x}}$ denotes the error state. For the quaternion error state, we employ JPL multiplicative error [37] and use $\delta\boldsymbol{\theta} \in \mathbb{R}^3$ defined by the error quaternion i.e., $\delta\bar{q} = \bar{q} \otimes \hat{\bar{q}}^{-1} \simeq [\frac{1}{2}\delta\boldsymbol{\theta}^\top \ 1]^\top$. The "$\boxplus$" and "$\boxminus$" operations map elements to and from a given manifold and equate to simple "+" and "-" for vector variables.

$$\min_{\mathbf{x}_M,\mathbf{x}_R,\mathbf{x}_N} \mathcal{C}(\mathbf{x}_M,\mathbf{x}_R,\mathbf{x}_N)$$
$$= \min_{\mathbf{x}_R,\mathbf{x}_N}\left(\min_{\mathbf{x}_M}\mathcal{C}_{mr}(\mathbf{x}_M,\mathbf{x}_R)+\mathcal{C}_{rn}\left(\mathbf{x}_R,\mathbf{x}_N\right)\right)$$

To solve the above problem, we minimize $\mathcal{C}_{mr}$ with respect to $\mathbf{x}_M$ as:

$$\mathcal{C}_{mr} \simeq \mathcal{C}(\hat{\mathbf{x}}_M(k),\hat{\mathbf{x}}_R(k)) + \mathbf{b}_M(k)^\top \begin{bmatrix}\delta\mathbf{x}_M(k)\\ \delta\mathbf{x}_R(k)\end{bmatrix}$$
$$+\frac{1}{2}\begin{bmatrix}\delta\mathbf{x}_M(k)\\ \delta\mathbf{x}_R(k)\end{bmatrix}^\top \mathbf{A}_M(k)\begin{bmatrix}\delta\mathbf{x}_M(k)\\ \delta\mathbf{x}_R(k)\end{bmatrix} \quad (11)$$
$$\mathbf{b}_M(k)=\begin{bmatrix}\mathbf{b}_{mm}(k)\\ \mathbf{b}_{rm}(k)\end{bmatrix},\ \mathbf{A}_M(k)=\begin{bmatrix}\mathbf{A}_{mm}(k) & \mathbf{A}_{mr}(k)\\ \mathbf{A}_{rm}(k) & \mathbf{A}_{rr}(k)\end{bmatrix}$$

where $\hat{\mathbf{x}}(k)$ denotes the best state estimate at time t_k, $\mathbf{b}_M(k)$ and $\mathbf{A}_M(k)$ are the gradient and Hessian matrix computed using $\hat{\mathbf{x}}(k)$, respectively. The optimal value of $\min_{\mathbf{x}_M}\mathcal{C}_{mr}(\mathbf{x}_M,\mathbf{x}_R)$ is:

$$\mathcal{C}_{mr}\simeq \alpha + \mathbf{b}_p(k)^\top\delta\mathbf{x}_R(k)+\frac{1}{2}\delta\mathbf{x}_R(k)^\top\mathbf{A}_p(k)\delta\mathbf{x}_R(k)$$
$$\mathbf{b}_p(k)=\mathbf{b}_{rm}(k)-\mathbf{A}_{rm}(k)\mathbf{A}_{mm}^{-1}(k)\mathbf{b}_{mm}(k) \quad (12)$$
$$\mathbf{A}_p(k)=\mathbf{A}_{rr}(k)-\mathbf{A}_{rm}(k)\mathbf{A}_{mm}^{-1}(k)\mathbf{A}_{mr}(k) \quad (13)$$

where α is independent from state $\mathbf{x}_R$ and $\mathbf{x}_M$. This cost is now independent of $\mathbf{x}_M$ resulting in computational savings while approximating the original non-linear costs with a second order Taylor-series and *permanently* fixes its linearization at t_k (see Eq. (12) and (13)).

B. System Observability

We now investigate the system observability properties before (t_k) and after marginalization ($t_{k'}$). This analysis follows the work of [27], [41] and [28] with the further extension to include inertial biases (we refer the reader to the companion technical report [38]). We first consider the full batch-MAP state Hessian $\mathbf{A}_{k'}$ at time $t_{k'}$:

$$\mathbf{A}_{k'}^{\text{full}}=\begin{bmatrix}\mathbf{A}_{mm}(k') & \mathbf{A}_{mr}(k') & \mathbf{0}\\ \mathbf{A}_{rm}(k') & \mathbf{A}_{rr}(k') & \mathbf{A}_{rn}(k')\\ \mathbf{0} & \mathbf{A}_{nr}(k') & \mathbf{A}_{nn}(k')\end{bmatrix} \quad (14)$$

One can see that *all* the states have been evaluated at the latest timestamp $t_{k'}$. Considering the case when marginalization has been performed at time t_k, and the state has been optimized with measurements at the following time $t_{k'}$:

$$\mathbf{A}_{k'}^{\text{marg}}=\begin{bmatrix}\mathbf{A}_{mm}(k) & \mathbf{A}_{mr}(k) & \mathbf{0}\\ \mathbf{A}_{rm}(k) & \mathbf{A}_{rr}(k)+\mathbf{A}_{rr}(k') & \mathbf{A}_{rn}(k')\\ \mathbf{0} & \mathbf{A}_{nr}(k') & \mathbf{A}_{nn}(k')\end{bmatrix} \quad (15)$$

It can be proved that (see [38], [41]):

$$\begin{aligned}\texttt{rank}(\mathbf{A}_{k'}^{\text{full}}) &< \texttt{rank}(\mathbf{A}_{k'}^{\text{marg}})\\ \texttt{dim}(\mathbb{N}(\mathbf{A}_{k'}^{\text{full}})) &> \texttt{dim}(\mathbb{N}(\mathbf{A}_{k'}^{\text{marg}}))\end{aligned} \quad (16)$$

where $\mathbb{N}(\mathbf{A})$ is the nullspace of $\mathbf{A}$.

This implies that by performing marginalization, which fixes the information related to the marginal and remaining states at time t_k, spurious information has been gained to cause a loss of nullspace dimension and leading to *inconsistencies*. We can inspect the nullspace of $\mathbf{A}_{k'}^{\text{full}}$ to see:

$$\mathbb{N}(\mathbf{A}_{k'}^{\text{full}})=\begin{bmatrix}\mathbb{N}_{x_0}^\top & \cdots & \mathbb{N}_{x_{k'}}^\top & \mathbb{N}_{f_0}^\top & \cdots & \mathbb{N}_{f_g}^\top\end{bmatrix}^\top \quad (17)$$

$$\mathbb{N}_{x_i}^\top=\begin{bmatrix}({}^{I_i}_{G}\mathbf{R}\,\mathbf{g})^\top & (\lfloor{}^{G}\mathbf{p}_{I_i}\rfloor\mathbf{g})^\top & (\lfloor{}^{G}\mathbf{v}_{I_i}\rfloor\mathbf{g})^\top & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3}\\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}$$
$$\mathbb{N}_{f_j}=\begin{bmatrix}\lfloor{}^{G}\mathbf{f}_j\rfloor\mathbf{g} & \mathbf{I}_3\end{bmatrix} \quad (18)$$

where $\mathbf{g}$ denotes the global gravity.

Remark: The 4 DoF unobservable directions can be interpreted as the last three columns corresponding to global translation, while the first column corresponds to the global rotation about the gravity vector (yaw). The marginalization of states at t_k make $\mathbf{A}_{k'}^{\text{marg}}$ gain rank and $\mathbb{N}(\mathbf{A}_{k'}^{\text{marg}})$ lose rank about the global yaw. This analytical nullspace also shows that only orientation, position, velocity, and features affect the dimensionality, and therefore, there is no need to FEJ the biases or auxiliary variables.

IV. Marginalization and FEJ

In this section, we discuss how the First-estimates Jacobian (FEJ) methodology is leveraged to prevent erroneous information gain due to marginalization. The key idea is to evaluate the Hessian using the first estimate $\hat{\mathbf{x}}_R(k)$ instead of the current estimate $\hat{\mathbf{x}}_R(k')$ for all states $\mathbf{x}_R$ involved with the marginal (the second cost in Eq. (10)):

$$\mathcal{C}(\mathbf{x}(k))\simeq\mathcal{C}(\hat{\mathbf{x}}_R(k'),\hat{\mathbf{x}}_N(k'))+\mathbf{b}(\hat{\mathbf{x}}_R(k),\hat{\mathbf{x}}_N(k'))^\top\delta\mathbf{x}(k')$$
$$+\frac{1}{2}\delta\mathbf{x}(k')^\top\mathbf{A}(\hat{\mathbf{x}}_R(k),\hat{\mathbf{x}}_N(k'))\delta\mathbf{x}(k') \quad (19)$$

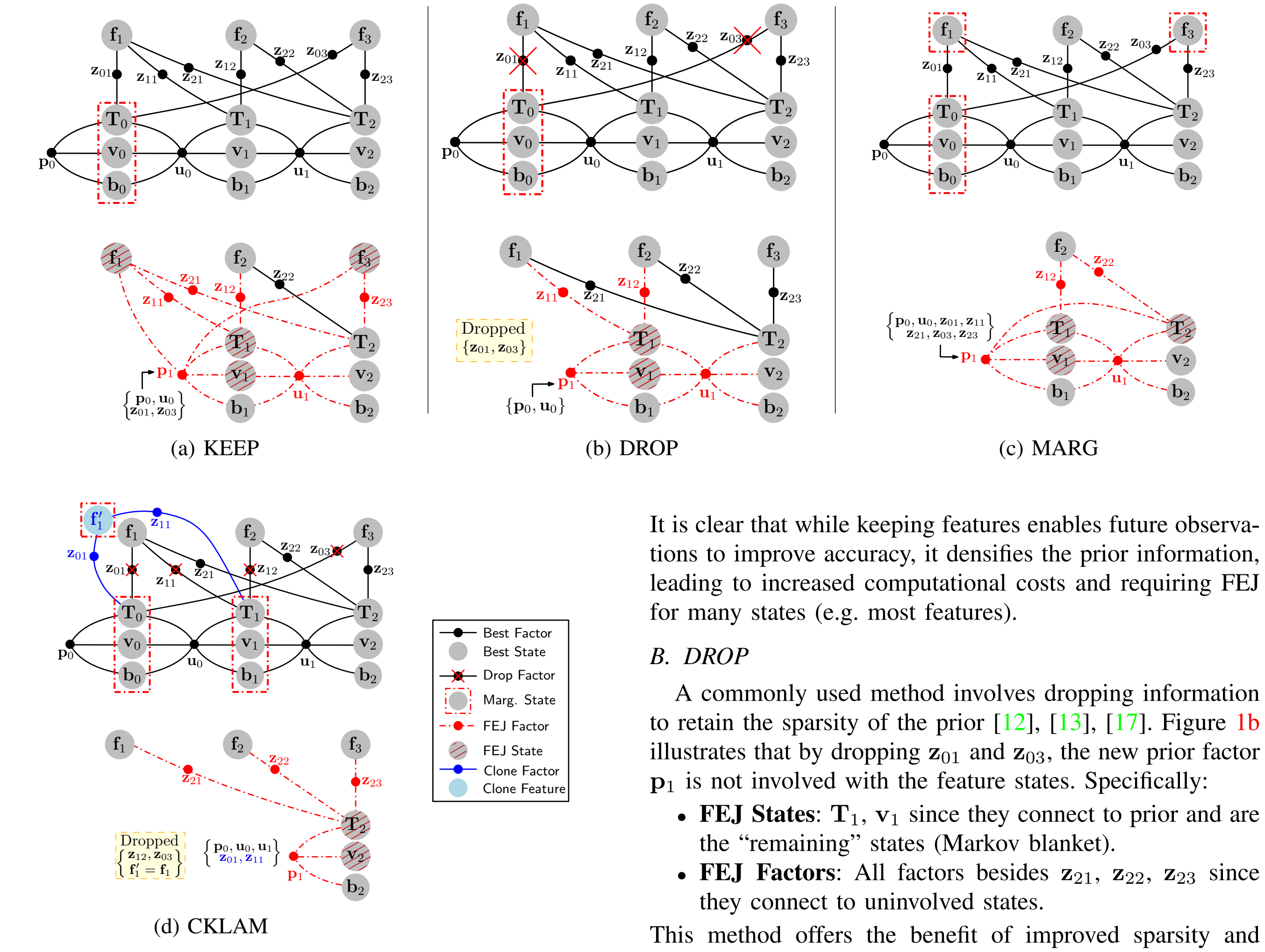
where $\delta\mathbf{x}(k')=[\delta\mathbf{x}_R(k')^\top\ \ \delta\mathbf{x}_N(k')^\top]^\top$. Note that the linearization point of $\mathbf{x}_R$ only needs to be changed for the Hessian $\mathbf{A}(\hat{\mathbf{x}}_R(k),\hat{\mathbf{x}}_A(k'))$ and gradient $\mathbf{b}(\hat{\mathbf{x}}_R(k),\hat{\mathbf{x}}_N(k'))$ computation, while the residual and states which do not affect the observability properties can use the best estimates (e.g., biases, see Eq. (17)). Algorithm 1 outlines this process.

Implementation Guide: We perform simple "bookkeeping" of the $\mathbf{x}_R$ states that need to use their first estimates. During marginalization, for states that have not been FEJ'ed yet, we record the current estimate as the FEJ for the states in $\mathbf{x}_R$ which appeared in the nullspace, see Eq. (17), as they impact consistency. Subsequently, factors connected to $\mathbf{x}_R$ should use FEJ for the states which have been FEJ'ed, $\hat{\mathbf{x}}_R(k)$, during Jacobian evaluation, while the others leverage the best estimate, $\hat{\mathbf{x}}_R(k')$ or $\hat{\mathbf{x}}_N(k')$.

Algorithm 1 Sliding-window optimization VINS with FEJ

```
Build factor graph and perform iterative optimization:
  • Construct optimization problem using all measurements at
    timestamp $t(k')$ [see Eq. (5)] and linearize the cost function
    using the following linearization points:
      – If state has FEJ: Use its first estimate $\hat{\mathbf{x}}(k)$
      – Else: Use its best updated estimate $\hat{\mathbf{x}}(k')$
  • Solve and correct the state [see Eq. (9)].
State marginalization:
  • Select the states in $\mathbf{x}_M$ to be marginalize.
  • For each state in $\mathbf{x}_R$ connected to $\mathbf{x}_M$:
      – If state in nullspace [Eq. (17)] and not FEJ'ed: Record
        the current estimate as its FEJ value.
  • Perform marginalization and calculate new prior information
    and gradient [see Eq. (12), (13)].
```

For example, a camera observation measurement factor which relates a feature not connected to the marginal prior but is observed from a pose connected to such prior should have its Jacobians evaluated using the *best* feature estimate



(a) KEEP

(b) DROP

(c) MARG

(d) CKLAM

Fig. 1: Four marginalization cases with the to-marginalized nodes in a red box and edges (measurements) that are "dropped" being crossed out. Nodes with fixed linearization points are shaded in red, while dashed red edges denote those evaluated and linearized with FEJ. The factor graph includes bearing observations $\mathbf{z}$, inertial measurements $\mathbf{u}$, and the prior $\mathbf{p}_0$. State variables (nodes) are represented as grey circles, and measurements (edges) connect related states. $\mathbf{f}_j$ is the jth feature, $\mathbf{T}_i$ is robot pose at t_i, $\mathbf{v}_i$ and $\mathbf{b}_i$ are the robot velocity and biases.

and the *first* pose estimate recorded when marginalization was performed. In what follows, we detail how marginalization strategies impact the connectivity of states to the marginal prior and thus FEJ.

A. KEEP

The simplest case is marginalizing only the inertial states while keeping a map of environmental features. As shown in Figure 1a, after marginalization, both features and the oldest inertial states become involved with the marginalized prior, thus requiring the fixing of many states. Specifically:

- **FEJ States**: $\mathbf{T}_1$, $\mathbf{v}_1$, $\mathbf{f}_1$, $\mathbf{f}_3$ since they connect to prior and are the "remaining" states (Markov blanket).
- **FEJ Factors**: All factors besides $\mathbf{z}_{22}$ since it connects to uninvolved states.

It is clear that while keeping features enables future observations to improve accuracy, it densifies the prior information, leading to increased computational costs and requiring FEJ for many states (e.g. most features).

B. DROP

A commonly used method involves dropping information to retain the sparsity of the prior [12], [13], [17]. Figure 1b illustrates that by dropping $\mathbf{z}_{01}$ and $\mathbf{z}_{03}$, the new prior factor $\mathbf{p}_1$ is not involved with the feature states. Specifically:

- **FEJ States**: $\mathbf{T}_1$, $\mathbf{v}_1$ since they connect to prior and are the "remaining" states (Markov blanket).
- **FEJ Factors**: All factors besides $\mathbf{z}_{21}$, $\mathbf{z}_{22}$, $\mathbf{z}_{23}$ since they connect to uninvolved states.

This method offers the benefit of improved sparsity and avoids the need to perform FEJ on the feature states. However, it comes at the cost of a new sub-optimal problem containing less constraint information about the features.

C. MARG

Another common case is to completely marginalize features observed by the to-be-marginalized inertial state. As shown in Figure 1c, this marginal prior now relates to all poses from which the features have been observed. Specifically:

- **FEJ States**: $\mathbf{T}_1$, $\mathbf{v}_1$, and $\mathbf{T}_2$ since they connect to prior and are the "remaining" states (Markov blanket).
- **FEJ Factors**: All factors have some portion of their Jacobians FEJ'ed

In this case, the marginal prior density has increased, but the state size has decreased significantly, likely providing significant computational benefits. However, a key downside is that future feature observations cannot be leveraged (i.e. they are treated as new features), and all poses need to be FEJ'ed, while the velocity $\mathbf{v}_2$, the remaining feature $\mathbf{f}_2$, and biases are still not required to be.

D. CKLAM

Another promising approach is the CKLAM marginalization technique [11] (which has been recently adopted in [14]). As shown in Figure 1d, to preserve the sparse

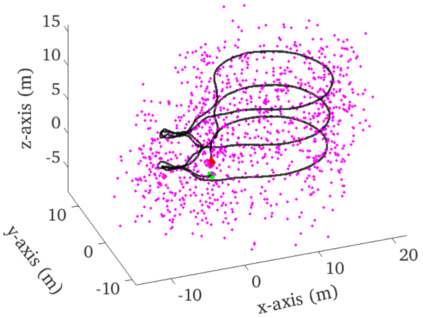


Fig. 2: Simulated `Gore` trajectory (221m), with green and red circles marking the starting and ending points and magenta simulated point features.

TABLE I: Simulation parameters and prior standard deviations for measurement perturbations.

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	10	Tracked Feat.	100
Min. Track Length	5	Max. SLAM Feat.	25

structure of the optimization problem, to-be-marginalized feature $\mathbf{f}_1$ is duplicated along with its measurements $\mathbf{z}_{01}$ and $\mathbf{z}_{11}$ connecting to the to-be-marginalized inertial states (note that $\mathbf{f}_1$ is observed at t_3 as it would be otherwise marginalized), and then this new feature $\mathbf{f}'_1$ is marginalized alongside the inertial states. All other feature measurements (e.g. $\mathbf{z}_{12}$ and $\mathbf{z}_{03}$) related to the to-be-marginalized states, which do not sufficiently constrain their feature, are dropped, introducing a loss of information. Specifically:

- **FEJ States**: $\mathbf{T}_2$, $\mathbf{v}_2$ since they connect to prior and are the "remaining" states (Markov blanket).
- **FEJ Factors**: All factors have some portion of their Jacobians FEJ'ed

CKLAM reduces to the DROP case, see Section IV-B, when only a single inertial state is marginalized (e.g., a sliding window). However, for multiple states, it enables the inclusion of feature observation information into the prior factor without increasing computational complexity while minimizing information loss. This makes CKLAM appealing for its computational efficiency and accuracy gains in "shifting" window and keyframe-based VINS.

V. Numerical Study

We simulate a realistic indoor 221 meter (m) three-floor dataset, see Figure 2. The system leverages the OpenVINS simulator [39] to generate realistic visual bearings and inertial measurements using the parameters listed in Table I. We leverage CPI preintegration [7] to create the inertial factor and use Ceres Solver [42] for optimization. The metrics used are Absolute Trajectory Error (ATE) [43] and Normalized Estimation Error Squared (NEES) [44], which should match the 3 DoF state size for both orientation and position if the estimator is consistent. The reported ATE and NEES are averaged over 20 Monte Carlo runs.

TABLE II: Average ATE over 20 `Gore` dataset runs with varying image noises and number of marginalized inertial states. Time is for optimization and marginalization only (no covariance recovery). Complete table is reported in [38].

σ	N		Algo.	ATE (deg/m)	NEES (3)	Time (ms)
1 pixel image noise	marg. 1 clone	FEJ	KEEP	0.354 / 0.118	2.713 / 2.451	21.7 ± 8.1
			DROP	0.966 / 0.259	3.039 / 2.667	12.5 ± 4.6
			MARG	0.930 / 0.250	3.173 / 2.661	10.9 ± 4.5
			CKLAM	0.966 / 0.259	3.039 / 2.667	12.4 ± 4.5
		No-FEJ	KEEP	2.328 / 0.407	181.7 / 26.702	18.8 ± 6.6
			DROP	0.931 / 0.259	2.946 / 2.781	10.6 ± 3.3
			MARG	1.007 / 0.256	3.039 / 2.502	9.2 ± 3.8
			CKLAM	0.931 / 0.259	2.946 / 2.781	10.2 ± 3.1
	marg. 3 clones	FEJ	KEEP	0.380 / 0.132	3.057 / 2.733	8.8 ± 10.5
			DROP	0.919 / 0.289	2.661 / 3.103	5.2 ± 7.0
			MARG	0.816 / 0.235	2.979 / 2.619	4.3 ± 5.9
			CKLAM	0.907 / 0.280	2.664 / 3.079	5.1 ± 6.8
		No-FEJ	KEEP	2.123 / 0.376	155.3 / 22.074	8.2 ± 9.3
			DROP	1.104 / 0.299	3.027 / 3.010	4.7 ± 6.0
			MARG	0.879 / 0.233	3.037 / 2.476	3.9 ± 5.2
			CKLAM	1.002 / 0.287	2.952 / 2.926	4.5 ± 5.8
	marg. 6 clones	FEJ	KEEP	0.362 / 0.129	2.733 / 2.898	4.5 ± 7.6
			DROP	1.186 / 0.331	2.900 / 2.777	3.1 ± 6.2
			MARG	0.844 / 0.214	3.268 / 2.196	2.4 ± 5.1
			CKLAM	0.854 / 0.261	2.548 / 2.639	3.0 ± 6.0
		No-FEJ	KEEP	1.556 / 0.295	93.6 / 14.878	4.4 ± 7.1
			DROP	1.403 / 0.350	3.415 / 2.951	2.6 ± 5.1
			MARG	0.812 / 0.206	3.300 / 2.216	2.2 ± 4.5
			CKLAM	1.048 / 0.277	2.997 / 2.740	2.6 ± 5.1
3 pixel image noise	marg. 1 clone	FEJ	KEEP	0.868 / 0.294	3.021 / 3.776	26.4 ± 9.2
			DROP	1.827 / 0.542	2.909 / 2.895	14.8 ± 4.4
			MARG	2.088 / 0.618	4.160 / 4.097	13.2 ± 4.7
			CKLAM	1.827 / 0.542	2.909 / 2.895	14.5 ± 4.5
		No-FEJ	KEEP	9.648 / 1.682	2226.5 / 115.1	21.6 ± 6.6
			DROP	1.940 / 0.510	2.701 / 2.670	12.1 ± 3.0
			MARG	3.143 / 0.657	4.504 / 3.572	10.0 ± 2.9
			CKLAM	1.940 / 0.510	2.701 / 2.670	12.1 ± 2.9

We present a sliding/shifting-window optimization-based VINS, where either the oldest or N-oldest IMU states are marginalized at each timestep. Optimization is only performed over a complete window to ensure sufficient feature constraints (e.g. a shifting window of 3 will optimize 3 times less than a sliding window). Features are triangulated after they are above a minimal track length and are handled via the four cases, see Section IV, or marginalized when their tracking is lost.

A. Experiment 1: Validation of FEJ's Impact

We first investigate how the different marginalization techniques are impacted by the use of FEJ and different shifting window sizes. The results are presented in Table II and Figure 3. In general, it can be seen that the KEEP method achieves significant gains in accuracy and consistency when leveraging FEJ, while the other marginalization methods are less sensitive to the use of FEJ. Overall, FEJ guarantees consistency and improves performance in most cases. Thus, we highly recommend its use.

1) KEEP: It is clear that FEJ has significant accuracy and consistency gains over the No-FEJ method, which performed with triple the ATE and a significantly overconfident covariance. This improvement is attributed to the marginal prior

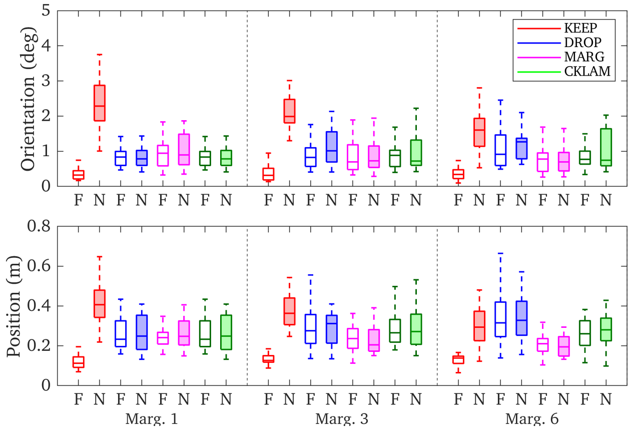


Fig. 3: Average ATE for 20 `Gore` dataset runs with different algorithms (see Table II, 1 pixel noise). FEJ is denoted as "F" (empty) and No-FEJ as "N" (shaded). Colors indices different marginalization methods.

becoming connected to the feature states and these features remaining tracked for long periods, leading to erroneous information accumulation. An additional 100 run Monte Carlo, see Figure 4, shows the inconsistent yaw quickly diverging and directly impacting the x-y positional estimates. Compared to other methods, it becomes evident that FEJ'ed KEEP has the most accurate state estimation performance, as expected since no information is dropped, at the cost of higher computational complexity. Conversely, No-FEJ performs even worse than sub-optimal approaches such as MARG, DROP or CKLAM. This demonstrates the crucial nature of consistency in optimization-based VINS.

2) DROP: It is not surprising to see that the FEJ and No-FEJ algorithms remain consistent in general. The use of FEJ here does show some improvement, particularly the orientation, over No-FEJ, but since the marginal prior only connects to the oldest state, there are only small gains. It is interesting that as more inertial states are marginalized in the shifting window, the performance difference between FEJ and No-FEJ is more apparent. The DROP method has the worst accuracy due to the significant amount of information loss but remains extremely efficient.

3) MARG: Here again, both FEJ and No-FEJ algorithms remain in general consistent, with good computational efficiency due to the reduced state size after feature marginalization. As investigated in the later ablation study, Section V-B, the consistency is due to the features being marginalized and the inertial states in the current window having a smaller impact. It can also be seen that FEJ has slightly worse accuracy, which can be equated to requiring all clones to use their FEJ instead of the best estimate and thus can introduce linearization errors. In the larger noise case, FEJ is able to outperform the No-FEJ method.

4) CKLAM: It is clear that as more inertial states are marginalized, the CKLAM method outperforms the DROP method due to the more information incorporated into the marginal prior. For the sliding window case, CKLAM has the same performance as DROP since a single feature observation cannot be leveraged. The use of FEJ here does show some improvement, particularly in the orientation, over No-FEJ. However, the gains are small since the marginal prior only connects to the oldest state.

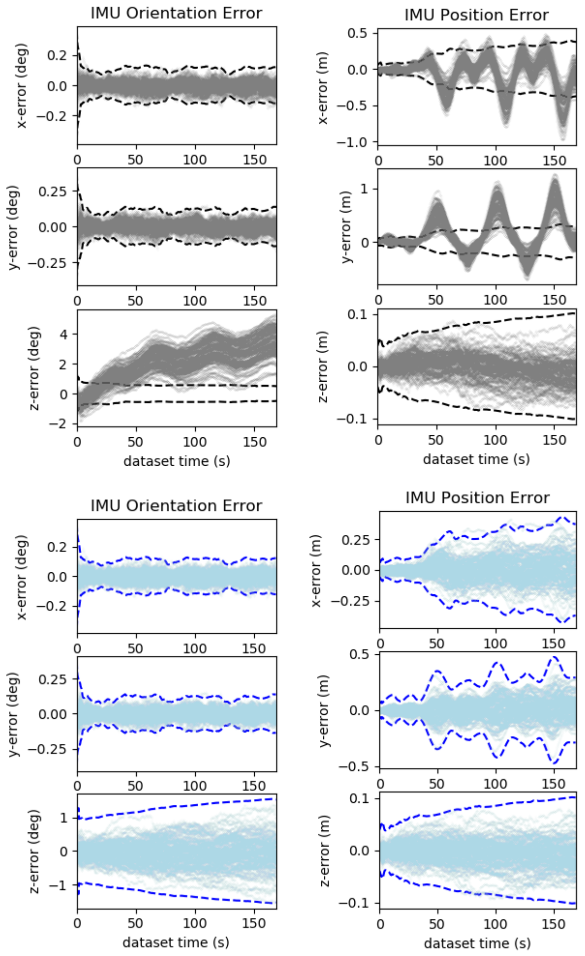


Fig. 4: IMU errors and $\pm 3\sigma$ bounds (dashed lines) for 100 Monte Carlo runs (1-pixel noise and KEEP marginalization). No-FEJ (top, black) and FEJ (bottom, blue).

B. Experiment 2: Ablation Study

Next we investigate the impact of FEJ'ing different states and factors have on the performance and consistency of the sliding window configuration (e.g. one state is marginalized and thus CKLAM is not reported). Shown in Table III, the FEJ'ing of feature states within the image factors, OF, has a significant impact when performing KEEP marginalization, clearly motivating the use of FEJ. As expected, both DROP and MARG methods are not impacted by the FEJ'ing of the features since no feature is connected to the prior and thus are independent to FEJ. However, it is clear that FEJ'ing of the inertial states in *just* the preintegration or image factors causes erroneous information to be gained due to the two factors having conflicting linearization points. This observation further provides insight into why FEJ is required to remedy the discrepancy between the marginal prior factor and the active states.

TABLE III: Average ATE over 20 Gore dataset runs for different combinations of FEJ'ed Jacobians. "I" denotes IMU preintegration, $\mathcal{C}_{I_i}$, cost have been changed, while "O" means changes in the feature observations cost, $\mathcal{C}_{f_{ij}}$. "C" indicates the poses are FEJ'ed, while "F" means features are FEJ'ed.

Algo.	Config.	ATE (deg/m)	NEES (3)
KEEP	IC	2.309 / 0.404	181.571 / 26.491
	OC	2.190 / 0.390	165.368 / 25.283
	OF	0.406 / 0.122	3.098 / 2.536
	OF+OC	0.384 / 0.122	3.013 / 2.655
	IC+OF+OC	0.353 / 0.118	2.705 / 2.443
	No-FEJ	2.304 / 0.403	178.556 / 26.335
DROP	IC	1.119 / 0.279	5.608 / 3.350
	OC	1.500 / 0.346	12.668 / 5.100
	OF	0.994 / 0.270	3.316 / 2.952
	OF+OC	1.500 / 0.346	12.668 / 5.100
	IC+OF+OC	0.967 / 0.256	3.043 / 2.624
	No-FEJ	0.994 / 0.270	3.316 / 2.952
MARG	IC	1.038 / 0.258	6.610 / 3.292
	OC	1.186 / 0.282	7.531 / 3.659
	OF	0.947 / 0.248	3.320 / 2.590
	OF+OC	1.186 / 0.282	7.534 / 3.661
	IC+OF+OC	0.922 / 0.248	3.164 / 2.670
	No-FEJ	0.947 / 0.248	3.320 / 2.590

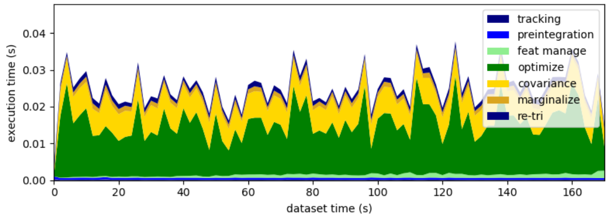


Fig. 5: Timing of the different components for the sliding window with KEEP marginalization.

C. Experiment 3: Noise Sensitivity

We now inspect the impact image measurement noise has on FEJ. Shown in the bottom of Table II, an additional set of sliding window Monte Carlo simulations have been performed for 3 pixel image noise. It can be observed that as the noise increases, the accuracy of both the No-FEJ KEEP and DROP decreases and additional inconsistencies arise. This degradation can be equated to higher noises causing larger information gain in the incorrect unobservable directions, see Eq. (16), in the No-FEJ method (this has been seen in filter-based VINS, see Chen et al. [26]). While FEJ can have additional linearization errors caused by higher noises, it is clear that the impact of inconsistencies caused by observability outweighs this by many orders of magnitude even at high image noise.

D. Findings and Discussions

It is evident from the preceding numerical study that FEJ exhibits impressive performance due to its ability to ensure correct observability properties of optimization-based VINS. Due to the partially observable nature of the visual-inertial systems (i.e. 4 DoF unobservable), different state marginalization methods have different impacts on the estimator consistency, necessitating the proper implementation of FEJ. Incorrect or partial implementations may introduce linearization discrepancies, causing erroneous information gain and degrading performance. We hope the detailed analysis and examples provided in this paper have clarified why and how FEJ should be realized in applicable practical systems.

When the measurement noise is high, inconsistencies caused by the observability mismatch become more apparent, and FEJ's benefit becomes more pronounced. Among the four primitive strategies of marginalization presented, Section IV, we recommend leveraging a hybrid of the four to take advantage of each. For example, KEEP to improve accuracy with long-lived features, DROP and CKLAM to improve efficiency, and MARG for lost features. We additionally stress that even estimators that *do not* recover covariance for computational savings (see Figure 5) will still reap benefits from applying the FEJ methodology through reduced estimation errors.

Based on our analysis and numerical study, the following key takeaways should be highlighted:

- Ensuring consistency is *essential* to improve the accuracy and robustness of optimization-based VINS.
- Some marginalization methods typically used to increase sparsity (e.g. DROP) happen to be able to maintain certain levels of consistent performance due to removing feature correlations with the marginal prior and thus linearization discrepancies, which, however, was not well understood in the literature.
- FEJ improves accuracy and consistency when properly implemented by taking into account the system observability and the applied marginalization methods. It is especially beneficial when dealing with long-lived features or a prior map.

VI. Conclusion and Future Work

This paper presents a comprehensive analysis of the application of the First-estimates Jacobian (FEJ) methodology within the optimization-based VINS framework. Drawing on the fundamental observability analysis of VINS, we demonstrate how marginalization's fixing of state linearization points causes erroneous gains of information, which can be addressed by FEJ. Four different marginalization methods commonly used in state-of-the-art optimization-based VINS are then presented and a detailed discussion on FEJ's application is performed. Through a series of numerical simulations, we investigated the estimator performance and verified the use of FEJ to guarantee estimation consistency and in turn, improve accuracy. We explore the impact of sensor noise and shifting window size, and we showed that FEJ has the largest impact when feature states become correlated with the marginal prior (e.g. long-lived environmental features or prior map). We close with a series of observations and recommendations, emphasizing that while FEJ requires precision during implementation, it greatly enhances optimization-based VINS. We strongly advocate for its use.

In the future, we will investigate the application of FEJ2 [26] methodology to reduce errors caused by fixing of linearization points, as well as explore methods to improve the efficiency of covariance recovery.

References

[1] C. Chen, P. Geneva, Y. Peng, W. Lee, and G. Huang, "Monocular visual-inertial odometry with planar regularities," in *Proc. of the IEEE International Conference on Robotics and Automation*, London, UK., 2023.

[2] C. Chen, Y. Yang, P. Geneva, W. Lee, and G. Huang, "Visual-inertial-aided online mav system identification," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyoto, Japan., 2022.

[3] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019.

[4] F. Dellaert, M. Kaess, *et al.*, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[5] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige, "g2o: A general framework for (hyper) graph optimization," in *International conference on robotics and automation (ICRA)*, 2011, pp. 9–13.

[6] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration theory for fast and accurate visual-inertial navigation," *IEEE Transactions on Robotics*, pp. 1–18, 2015.

[7] K. Eckenhoff, P. Geneva, and G. Huang, "Closed-form preintegration methods for graph-based visual-inertial navigation," *International Journal of Robotics Research*, vol. 38, no. 5, pp. 563–586, 2019. [Online]. Available: https://github.com/rpng/cpi

[8] Y. Yang, B. P. W. Babu, C. Chen, G. Huang, and L. Ren, "Analytic combined imu integrator for visual-inertial navigation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020.

[9] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[11] E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis, "C-KLAM: Constrained keyframe-based localization and mapping," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 3638–3643.

[12] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[13] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[14] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "ICE-BA: Incremental, consistent and efficient bundle adjustment for visual-inertial slam," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1974–1982.

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

[16] J. Hsiung, M. Hsiao, E. Westman, R. Valencia, and M. Kaess, "Information sparsification in visual-inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1146–1153.

[17] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 422–429, 2019.

[18] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 158–176, 2013.

[19] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[20] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 298–304.

[21] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," *International Journal of Robotics Research*, Apr. 2019.

[22] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016.

[23] Y. Yang, C. Chen, W. Lee, and G. P. Huang, "Decoupled right invariant error states for consistent visual-inertial navigation," *IEEE Robotics and Automation Letters*, 2022.

[24] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent EKF SLAM estimators," *International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, Apr. 2010.

[25] ——, "A first-estimates Jacobian EKF for improving SLAM consistency," in *Proc. of the 11th International Symposium on Experimental Robotics*, Athens, Greece, July 2008.

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

[27] T.-C. Dong-Si and A. I. Mourikis, "Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 5655–5662.

[28] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "An observability constrained sliding window filter for SLAM," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, Sept. 2011, pp. 65–72.

[29] S. Leutenegger, "OKVIS2: Realtime scalable visual-inertial slam with loop closure," *arXiv preprint arXiv:2202.09199*, 2022.

[30] M. Brossard, A. Barrau, P. Chauchat, and S. Bonnabel, "Associating uncertainty to extended poses for on lie group imu preintegration with rotating earth," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 998–1015, 2021.

[31] P. Chauchat, A. Barrau, and S. Bonnabel, "Invariant smoothing on lie groups," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1703–1710.

[32] J. Huai, Y. Lin, Y. Zhuang, and M. Shi, "Consistent right-invariant fixed-lag smoother with application to visual inertial slam," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, 2021, pp. 6084–6092.

[33] D. Lisus, "Dissecting visual-inertial SLAM: Errors, landmarks, and consistency," Master's thesis, McGill University, 2022.

[34] D. Lisus, M. Cohen, and J. R. Forbes, "Know what you don't know: Consistency in sliding window filtering with unobservable states applied to visual-inertial slam," *IEEE Robotics and Automation Letters*, 2023.

[35] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2510–2517.

[36] L. von Stumberg and D. Cremers, "DM-VIO: Delayed marginalization visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1408–1415, 2022.

[37] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., Mar. 2005. [Online]. Available: http://mars.cs.umn.edu/tr/reports/Trawny05b.pdf

[38] C. Chen, P. Geneva, Y. Peng, W. Lee, and G. Huang, "Technical report: Optimization-based VINS: Consistency, marginalization, and FEJ," University of Delaware, Tech. Rep. RPNG-2023-GRAPH, 2023. [Online]. Available: https://udel.edu/~ghuang/papers/tr_graph.pdf

[39] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: a research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020. [Online]. Available: https://github.com/rpng/open_vins

[40] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms*. Springer, 2000, pp. 298–372.

[41] T.-C. Dong-Si and A. I. Mourikis, "Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis," Dept. of Electrical Engineering, University of California, Riverside, Tech. Rep., 2011. [Online]. Available: https://intra.ece.ucr.edu/~mourikis/tech_reports/fixed_lag.pdf

[42] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," https://github.com/ceres-solver/ceres-solver, 2022.

[43] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7244–7251.

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