Technical Report
FEJ2: A Consistent Visual-Inertial State Estimator Design

Chuchu Chen - ccchu@udel.edu
Yulin Yang - yuyang@udel.edu
Patrick Geneva - pgeneva@udel.edu
Guoquan Huang - ghuang@udel.edu

Department of Mechanical Engineering
University of Delaware, Delaware, USA

Robot Perception and Navigation Group (RPNG)
Tech Report - RPNG-2021-FEJ2
Last Updated - September 27, 2021

Contents

1 Visual-Inertial Systems

In this section, we briefly overview the EKF-based visual-inertial system which fuses IMU readings and environmental feature measurements.

1.1 Propagation

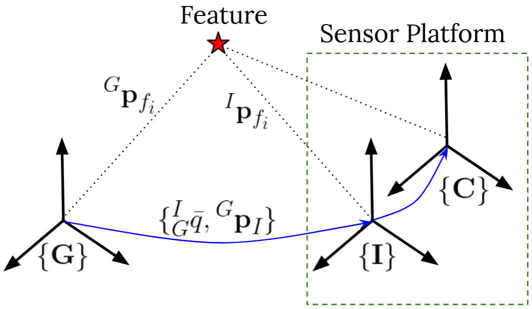

Figure 1: Sensor platform comprising an IMU and a camera. $\{{}^I_G\bar{q}, {}^G\mathbf{p}_I\}$ represents the pose (orientation and position) of the IMU frame $\{I\}$ with respect to the global frame $\{G\}$. ${}^G\mathbf{p}_{f_i}$ and ${}^I\mathbf{p}_{f_i}$ denote the position of the ith feature in Global frame and Local IMU frame, respectively; $\{C\}$ is the camera frame.

Specifically, the state vector $\mathbf{x}$ consists of the IMU state $\mathbf{x}_I$ and the feature state $\mathbf{x}_f$.

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_I^\top & \mathbf{x}_f^\top\end{bmatrix}^\top \tag{1}$$

$$= \begin{bmatrix}{}^I_G\bar{q}^\top & {}^G\mathbf{p}_I^\top & {}^G\mathbf{v}_I^\top & \mathbf{b}_g^\top & \mathbf{b}_a^\top & | & {}^G\mathbf{p}_{f_1}^\top & \dots & {}^G\mathbf{p}_{f_M}^\top\end{bmatrix}^\top \tag{2}$$

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

The IMU kinematics are used to evolve the state from time t_k to t_{k+1} [1]:

$${}^I_G\dot{\bar{q}}(t) = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}(t)){}^I_G\bar{q}(t) \tag{3}$$

$${}^G\dot{\mathbf{p}}_I(t) = {}^G\mathbf{v}_I(t) \tag{4}$$

$${}^G\dot{\mathbf{v}}_I(t) = {}^{I(t)}_G\mathbf{R}^\top\mathbf{a}(t) \tag{5}$$

$$\dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t) \tag{6}$$

$$\dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) \tag{7}$$

$${}^G\dot{\mathbf{p}}_{f_i}(t) = \mathbf{0} \qquad i = 1, \dots, M \tag{8}$$

where $\boldsymbol{\omega}(t) = [\omega_1 \ \omega_2 \ \omega_3]^\top$ and $\mathbf{a}(t)$ are the angular velocity and acceleration in the IMU local frame $\{I\}$; $\boldsymbol{\Omega}(\boldsymbol{\omega}(t)) = \begin{bmatrix}-\lfloor\boldsymbol{\omega}\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\top & 0\end{bmatrix}$ where $\lfloor\cdot\rfloor$ is the skew-symmetric matrix. $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$ are white

Gaussian noise that drive the IMU biases. A canonical three-axis IMU provides linear acceleration and angular velocity measurements, ${}^{I}\mathbf{a}_m$ and ${}^{I}\boldsymbol{\omega}_m$, expressed in the local IMU frame $\{I\}$ modeled as:

$$\begin{aligned} \mathbf{a}_m(t) &= \mathbf{a}(t) + {}^{I}_{G}\mathbf{R}(t){}^{G}\mathbf{g} + \mathbf{b}_a(t) + \mathbf{n}_a(t) && (9) \\ \boldsymbol{\omega}_m(t) &= \boldsymbol{\omega}(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) && (10) \end{aligned}$$

where ${}^{G}\mathbf{g} \simeq [0, 0, -9.8]^\top$ is the gravitational acceleration expressed in $\{G\}$, $\mathbf{n}_g$ and $\mathbf{n}_a$ are zero-mean white Gaussian noise. ${}^{I}_{G}\mathbf{R}$ denotes the rotation matrix from global frame to local IMU frame. Linearizing Eq.(3)-(7) about the current state estimate yields the state estimate propagation model:

$$\begin{aligned} {}^{I}_{G}\dot{\hat{\bar{q}}}(t) &= \frac{1}{2}\boldsymbol{\Omega}\left(\hat{\boldsymbol{\omega}}(t)\right){}^{I}_{G}\hat{\bar{q}}(t) && (11) \\ {}^{G}\dot{\hat{\mathbf{p}}}_I &= {}^{G}\hat{\mathbf{v}}_I(t) && (12) \\ {}^{G}\dot{\hat{\mathbf{v}}}_I(t) &= {}^{I(t)}_{G}\hat{\mathbf{R}}^\top\hat{\mathbf{a}}(t) && (13) \\ \dot{\hat{\mathbf{b}}}_g(t) &= \mathbf{0} && (14) \\ \dot{\hat{\mathbf{b}}}_a(t) &= \mathbf{0} && (15) \\ {}^{G}\dot{\hat{\mathbf{p}}}_{f_i}(t) &= \mathbf{0} && (16) \end{aligned}$$

where $\hat{\mathbf{a}}(t) = \mathbf{a}_m(t) - \hat{\mathbf{b}}_a(t)$ and $\hat{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}_m(t) - \hat{\mathbf{b}}_g(t)$. For the sake of EKF covariance propogation, we define our error state as:

$$\begin{aligned} \tilde{\mathbf{x}} &= \begin{bmatrix} \tilde{\mathbf{x}}_I^\top & \tilde{\mathbf{x}}_f^\top \end{bmatrix}^\top && (17) \\ &= \begin{bmatrix} {}^{I}_{G}\delta\boldsymbol{\theta}^\top & {}^{G}\tilde{\mathbf{p}}_I^\top & {}^{G}\tilde{\mathbf{v}}_I^\top & {}^{G}\tilde{\mathbf{b}}_g^\top & {}^{G}\tilde{\mathbf{b}}_a^\top & | & {}^{G}\tilde{\mathbf{p}}_{f_i}^\top & \dots & {}^{G}\tilde{\mathbf{p}}_{f_M}^\top \end{bmatrix} && (18) \end{aligned}$$

Now we can conclude the continuous-time error state propagation as:

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}(t) &= \begin{bmatrix} \dot{\tilde{\mathbf{x}}}_I(t) \\ \dot{\tilde{\mathbf{x}}}_f(t) \end{bmatrix} && (19) \\ &= \begin{bmatrix} \mathbf{F}_I(t) & \mathbf{0}_{15\times 3M} \\ \mathbf{0}_{3M\times 15} & \mathbf{0}_{3M} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_I(t) \\ \tilde{\mathbf{x}}_f(t) \end{bmatrix} + \begin{bmatrix} \mathbf{G}_I(t) \\ \mathbf{0}_{3M\times 12} \end{bmatrix}\mathbf{n}(t) && (20) \\ &= \mathbf{F}(t)\tilde{\mathbf{x}}(t) + \mathbf{G}(t)\mathbf{n}(t) && (21) \end{aligned}$$

where $\mathbf{n}(t) = \begin{bmatrix} \mathbf{n}_g^\top & \mathbf{n}_{wg}^\top & \mathbf{n}_a^\top & \mathbf{n}_{wa}^\top \end{bmatrix}^\top$ represents the system noise. The continuous-time error-state transition matrix $\mathbf{F}_I(t)$ and the input noise matrix $\mathbf{G}_I(t)$ are

$$\mathbf{F}_I(t) = \begin{bmatrix} -\lfloor\hat{\boldsymbol{\omega}}\rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -{}^{I}_{G}\hat{\mathbf{R}}^\top\lfloor\hat{\mathbf{a}}\rfloor & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -{}^{I}_{G}\hat{\mathbf{R}}^\top \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \qquad (22)$$

$$\mathbf{G}_I(t) = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -{}^{I}_{G}\hat{\mathbf{R}}^\top & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \qquad (23)$$

We model the system noise as zero-mean Gaussian process with auto-correlation.

$$\mathbb{E}\left[\mathbf{n}(t)\mathbf{n}^{\top}(t)\right] = \mathbf{Q}\delta(t-\tau) \tag{24}$$

In piratical EKF implementation, we can numerically propagate the state estimate when new IMU measurements Eq. (9)(10) are received. In order to propagate the system error covariance we typically derive the discreate-time state translation matrix from time t_k to t_{k+1} as $\hat{\mathbf{\Phi}}(t_{k+1}, t_k)$. It can be found either numerically or analytically [3] as the solution to the following equations with initial condition $\hat{\mathbf{\Phi}}(k,k) = \mathbf{I}_{15+3M}$.

$$\dot{\hat{\mathbf{\Phi}}}(k+1,k) = \mathbf{F}(t)\hat{\mathbf{\Phi}}(k+1,k) \tag{25}$$

which can be calculated analytically as:

$$\hat{\mathbf{\Phi}}(k+1,k) = \begin{bmatrix} \hat{\mathbf{\Phi}}_{11} & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\mathbf{\Phi}}_{14} & \mathbf{0}_3 & \mathbf{0}_3 \\ \hat{\mathbf{\Phi}}_{21} & \mathbf{I}_3 & \mathbf{I}_3\Delta t & \hat{\mathbf{\Phi}}_{24} & \hat{\mathbf{\Phi}}_{25} & \mathbf{0}_3 \\ \hat{\mathbf{\Phi}}_{31} & \mathbf{0}_3 & \mathbf{I}_3 & \hat{\mathbf{\Phi}}_{34} & \hat{\mathbf{\Phi}}_{35} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_{3\times 3M} & \mathbf{0}_{3\times 3M} & \mathbf{0}_{3\times 3M} & \mathbf{0}_{3\times 3M} & \mathbf{0}_{3\times 3M} & \mathbf{I}_{3M} \end{bmatrix} \tag{26}$$

where $\Delta t = t_{k+1} - t_k$ and:

$$\hat{\mathbf{\Phi}}_{11} = \exp\left(-\int_{t_k}^{t_{k+1}} \lfloor \hat{\boldsymbol{\omega}}(\tau) \rfloor \mathrm{d}\tau\right) \tag{27}$$

$$\hat{\mathbf{\Phi}}_{14} = -\int_{t_k}^{t_{k+1}} {}^{I_{k+1|k}}_{I_\tau}\hat{\mathbf{R}}\mathrm{d}\tau \tag{28}$$

$$\hat{\mathbf{\Phi}}_{21} = -\lfloor {}^{G}\hat{\mathbf{p}}_{I_{k+1|k}} - {}^{G}\hat{\mathbf{p}}_{I_{k|k}} - {}^{G}\hat{\mathbf{v}}_{I_{k|k}}\Delta t + \frac{1}{2}{}^{G}\mathbf{g}\Delta t^2 \rfloor {}^{I_{k|k}}_{G}\hat{\mathbf{R}}^{\top} \tag{29}$$

$$\hat{\mathbf{\Phi}}_{24} = \int_{t_k}^{t_{k+1}} \int_{t_1}^{\theta} {}^{G}_{I_s}\hat{\mathbf{R}} \lfloor {}^{I_s}\hat{\mathbf{a}} \rfloor \int_{t_k}^{s} {}^{I_s}_{I_\tau}\hat{\mathbf{R}}\mathrm{d}\tau\mathrm{d}s\mathrm{d}\theta \tag{30}$$

$$\hat{\mathbf{\Phi}}_{25} = -\int_{t_k}^{t_{k+1}} \int_{t_k}^{s} {}^{G}_{I_\tau}\hat{\mathbf{R}}\mathrm{d}\tau\mathrm{d}s \tag{31}$$

$$\hat{\mathbf{\Phi}}_{31} = -\lfloor {}^{G}\hat{\mathbf{v}}_{I_{k+1|k}} - {}^{G}\hat{\mathbf{v}}_{I_{k|k}} + {}^{G}\mathbf{g}\Delta t \rfloor {}^{I_{k|k}}_{G}\hat{\mathbf{R}}^{\top} \tag{32}$$

$$\hat{\mathbf{\Phi}}_{34} = \int_{t_k}^{t_{k+1}} {}^{I_s}_{G}\hat{\mathbf{R}}^{\top} \lfloor {}^{I_s}\hat{\mathbf{a}} \rfloor \int_{t_k}^{s} {}^{I_\tau}_{I_s}\hat{\mathbf{R}}^{\top}\mathrm{d}\tau\mathrm{d}s \tag{33}$$

$$\hat{\mathbf{\Phi}}_{35} = -\int_{t_k}^{t_{k+1}} {}^{I_\tau}_{G}\hat{\mathbf{R}}^{\top}\mathrm{d}\tau \tag{34}$$

The standard EKF propagates the error covariance as:

$$\mathbf{P}_{k+1|k} = \hat{\mathbf{\Phi}}(k+1,k)\mathbf{P}_{k|k}\hat{\mathbf{\Phi}}(k+1,k)^{\top} + \mathbf{Q}_k \tag{35}$$

We use $\hat{\mathbf{x}}_{j|i}$ and $\mathbf{P}_{j|i}$ to represent the state estimate and covariance at time-step j computed using measurements up to time-step i. The discrete-time system noise covariance $\mathbf{Q}_k$ can be computed as:

$$\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \hat{\mathbf{\Phi}}(k+1,\tau)\mathbf{G}(\tau)\mathbf{Q}\mathbf{G}(\tau)^{\top}\hat{\mathbf{\Phi}}(k+1,\tau)^{\top}\mathrm{d}\tau \tag{36}$$

1.2 Measurement Update

Assuming a calibrated perspective camera, the bearing measurement of the ith feature at timestep t_{k+1} can be related to the state by the following:

$$\begin{align} \mathbf{z}_{k+1} &= \mathbf{h}(\mathbf{x}_{k+1}) + \mathbf{n}_{k+1} \tag{37} \\ &:= \boldsymbol{\Lambda}({}^{C_{k+1}}\mathbf{p}_{f_i}) + \mathbf{n}_{k+1} \tag{38} \end{align}$$

where $\mathbf{n}_{k+1} \sim N(\mathbf{0}, \mathbf{R}_{k+1})$ is the measurement noise; $\boldsymbol{\Lambda}$ is the camera perspective projection function that project the 3D feature point ${}^{C_{k+1}}\mathbf{p}_{f_i}$ expressed in the current IMU frame $\{C_{k+1}\}$ onto the image plane:

$$\begin{align} &\boldsymbol{\Lambda}(\begin{bmatrix} x & y & z \end{bmatrix}^\top) = \frac{1}{z}\begin{bmatrix} x \\ y \end{bmatrix} \tag{39} \\ &\text{where} \quad {}^{C_{k+1}}\mathbf{p}_{f_i} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = {}^C_I\mathbf{R}\,{}^{I_{k+1}}_G\mathbf{R}\left({}^G\mathbf{p}_{f_{i,k+1}} - {}^G\mathbf{p}_{I_{k+1}}\right) + {}^C\mathbf{p}_I \tag{40} \end{align}$$

where $\{{}^C_I\mathbf{R}, {}^C\mathbf{p}_I\}$ represent the transformation between camera and IMU (see Figure 1). Linearizing (37) with respect to the current state $\hat{\mathbf{x}}_{k+1|k}$ we get the following:

$$\begin{align} \mathbf{z}_{k+1} &\simeq \mathbf{h}(\hat{\mathbf{x}}_{I_{k+1|k}}, \hat{\mathbf{x}}_{f_{k+1|k}}) + \hat{\mathbf{H}}_{I_{k+1}}(\mathbf{x}_{I_{k+1}} \boxminus \hat{\mathbf{x}}_{I_{k+1|k}}) + \hat{\mathbf{H}}_{f_{k+1}}(\mathbf{x}_{f_{k+1}} \boxminus \hat{\mathbf{x}}_{f_{k+1|k}}) + \mathbf{n}_{k+1} \tag{41} \\ &= \mathbf{h}(\hat{\mathbf{x}}_{I_{k+1|k}}, \hat{\mathbf{x}}_{f_{k+1|k}}) + \begin{bmatrix} \hat{\mathbf{H}}_{I_{k+1}} & \hat{\mathbf{H}}_{f_{k+1}} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{I_{k+1|k}} \\ \tilde{\mathbf{x}}_{f_{k+1|k}} \end{bmatrix} + \mathbf{n}_{k+1} \tag{42} \\ &= \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) + \hat{\mathbf{H}}_{k+1}\tilde{\mathbf{x}}_{k+1|k} + \mathbf{n}_{k+1} \tag{43} \\ \Rightarrow \mathbf{r}_{k+1} &\triangleq \mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) \tag{44} \end{align}$$

where $\hat{\mathbf{H}}_{k+1}$ denotes the Jacobian evaluated at $\hat{\mathbf{x}}_{k+1|k}$. $\hat{\mathbf{H}}_{k+1}$ only contains non-zero blocks for the pose and the ith feature and, thus is computed as ($k+1|k$ subscripts are dropped for brevity):

$$\begin{align} &\hat{\mathbf{H}}_{k+1} \tag{45} \\ &= \begin{bmatrix} \hat{\mathbf{H}}_{I_{k+1}} & \hat{\mathbf{H}}_{f_{k+1}} \end{bmatrix} \tag{46} \\ &= \frac{\partial \mathbf{r}_{k+1}}{\partial {}^{C_{k+1|k}}\tilde{\mathbf{p}}_{f_i}} \left[\begin{array}{ccccc|ccc} \frac{\partial {}^{C_{k+1|k}}\tilde{\mathbf{p}}_{f_i}}{\partial \delta\boldsymbol{\theta}_{k+1}} & \frac{\partial {}^{C_{k+1|k}}\tilde{\mathbf{p}}_{f_i}}{\partial {}^G\tilde{\mathbf{p}}_{I_{k+1}}} & \frac{\partial {}^{C_{k+1|k}}\tilde{\mathbf{p}}_{f_i}}{\partial {}^G\tilde{\mathbf{v}}_{I_{k+1}}} & \frac{\partial {}^{C_{k+1|k}}\tilde{\mathbf{p}}_{f_i}}{\partial {}^G\tilde{\mathbf{b}}_g} & \frac{\partial {}^{C_{k+1|k}}\tilde{\mathbf{p}}_{f_i}}{\partial {}^G\tilde{\mathbf{b}}_a} & \cdots & \frac{\partial {}^{C_{k+1|k}}\tilde{\mathbf{p}}_{f_i}}{\partial {}^G\tilde{\mathbf{p}}_{f_{i,k+1|k}}} & \cdots \end{array} \right] \notag \\ &= \nabla\hat{\mathbf{h}}_i \left[\begin{array}{ccccc|ccc} \hat{\mathbf{H}}_{\theta_{k+1}} & \hat{\mathbf{H}}_{p_{k+1}} & \hat{\mathbf{H}}_{v_{k+1}} & \hat{\mathbf{H}}_{b_{g,k+1}} & \hat{\mathbf{H}}_{b_{a,k+1}} & \cdots & \hat{\mathbf{H}}_{f_{i,k+1}} & \cdots \end{array} \right] \tag{47} \\ &= \nabla\hat{\mathbf{h}}_i \left[\begin{array}{ccccc|ccc} \hat{\mathbf{H}}_{\theta_{k+1}} & \hat{\mathbf{H}}_{p_{k+1}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \cdots & \hat{\mathbf{H}}_{f_{i,k+1}} & \cdots \end{array} \right] \tag{48} \end{align}$$

where

$$\begin{align} \hat{\mathbf{H}}_{\theta_{k+1}} &= \lfloor {}^{I_{k+1|k}}_G\hat{\mathbf{R}}\left({}^G\hat{\mathbf{p}}_{f_{i,k+1|k}} - {}^G\hat{\mathbf{p}}_{I_{k+1|k}}\right)\rfloor \tag{49} \\ \hat{\mathbf{H}}_{p_{k+1}} &= -\hat{\mathbf{H}}_{f_{i,k+1}} = -{}^{I_{k+1|k}}_G\hat{\mathbf{R}} \tag{50} \\ \nabla\hat{\mathbf{h}}_i &= \frac{1}{\hat{z}_i{}^2}\begin{bmatrix} \hat{z}_i & 0 & -\hat{x}_i \\ 0 & \hat{z}_i & -\hat{y}_i \end{bmatrix} {}^C_I\mathbf{R} \tag{51} \end{align}$$

Once the measurement residual $\mathbf{r}_{k+1}$ and Jacobian $\mathbf{H}_{k+1}$ with respect to states are computed and successfully pass the Mahalanobis gating test, we can apply the EKF update equations [4] to update the filter as:

$$\mathbf{S}_{k+1} = \mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^{\top} + \mathbf{R}_{k+1} \tag{52}$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^{\top}\mathbf{S}_{k+1}^{-1} \tag{53}$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} \boxplus \mathbf{K}_{k+1}\mathbf{r}_{k+1} \tag{54}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^{\top} \tag{55}$$

1.3 Observability Analysis

System observability plays a crucial role in state estimation [5, 6]. Understanding system observability provides a deep insight about the system's geometrical properties and determines the minimal measurement modalities needed. With the state transition matrix, Eq. (26), and measurement Jacobian, Eq. (44), we construct the observability matrix [7]:

$$\mathcal{O} \triangleq \begin{bmatrix} \mathbf{H}_0\mathbf{\Phi}(0,0) \\ \mathbf{H}_1\mathbf{\Phi}(1,0) \\ \vdots \\ \mathbf{H}_{k+1}\mathbf{\Phi}(k+1,0) \end{bmatrix} \tag{56}$$

If $\mathcal{O}$ is of full column rank, the system is fully observable. However, VINS is partial observable with a nullspace $\mathbf{N}$ satisfying $\mathcal{O}\mathbf{N} = \mathbf{0}$ (see [8, 7]). The nullspace $\mathbf{N}$ describes the state unobservable subspace cannot be recovered with measurements.

The nullspace $\mathbf{N}$ for VINS should be of *four* d.o.f and relates to the global yaw and translation [7]. At timestep t_k it can be computed as:

$$\mathbf{N}_k = \begin{bmatrix} \mathbf{0}_3 & {}^{I_k}_{G}\mathbf{R}\,{}^{G}\mathbf{g} \\ \mathbf{I}_3 & -\lfloor {}^{G}\mathbf{p}_{I_k} \rfloor {}^{G}\mathbf{g} \\ \mathbf{0}_3 & -\lfloor {}^{G}\mathbf{v}_{I_k} \rfloor {}^{G}\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \vdots & \vdots \\ \mathbf{I}_3 & -\lfloor {}^{G}\mathbf{p}_{f_{i,k}} \rfloor {}^{G}\mathbf{g} \\ \vdots & \vdots \end{bmatrix} \tag{57}$$

it is easy to verify that the "ideal" observability matrix $\check{\mathcal{O}}_{j,k} = \check{\mathbf{H}}_j\check{\mathbf{\Phi}}(j,k)$which evaluate at *true* state satisfies $\check{\mathcal{O}}_{j,k}\mathbf{N}_k = \mathbf{0}$ for any j and k . The standard EKF, which always computes the state transition $\hat{\mathbf{\Phi}}(k,0)$ and measurement Jacobian $\hat{\mathbf{H}}_k$ using the current state estimates, makes the global orientation appears to be observable and thus reducing the nullspace to only *three* d.o.f [9]. This causes the filter gain *extra* information and become overconfident to its estimation results. Moreover, a valid state transition matrix should have the semi-group property [10]:

$$\mathbf{\Phi}(k+1,k-1) = \mathbf{\Phi}(k+1,k)\mathbf{\Phi}(k,k-1) \tag{58}$$

The state translation matrix $\hat{\mathbf{\Phi}}$ [see Eq. (26)] evaluated at *current* state estimate for standard EKF violates this property s.t. (see [9],[10, Lemma 4.1]).

$$\hat{\mathbf{\Phi}}(k+1,k-1) \neq \hat{\mathbf{\Phi}}(k+1,k)\hat{\mathbf{\Phi}}(k,k-1) \tag{59}$$

The above issues cause inconsistency and degrade accuracy.

2 Consistent Visual-Inertial Estimator Design

Leveraging the prior observability-based consistent estimator designs [8] and motivated by the great success of such estimators in visual-inertial systems [11, 9], we aim to improve this observability-based design by addressing its significant caveat of (potentially) large linearization errors. To this end, in what follows, after reviewing the observability-based methodology, we present in detail the proposed FEJ2 consistent estimator design methodology.

2.1 Observability-based Methodology

2.1.1 OC-VINS

OC [11, 7] maintains the proper system observability properties by enforcing the initial (and thus the current) unobservable nullspace $\mathbf{N}_0$ to span *correct* system unobservable directions. Note that we assume there is one single feature in the state vector and can be detected and initialized into state vector as ${}^G\hat{\mathbf{p}}_{f_{1|0}}$ at the first timestep t_0. Note that due to the feature is zero dynamics, we have $\hat{\mathbf{p}}_{f_{k|0}} = \hat{\mathbf{p}}_{f_{1|0}}$.

$$\mathbf{H}_k \mathbf{\Phi}(k,0)\mathbf{N}_0 = \mathbf{0}, \quad \forall k \geq 0 \tag{60}$$

$$\text{where} \quad \mathbf{N}_0 = \mathbf{N}(\hat{\mathbf{x}}_{1|0}) = \begin{bmatrix} \mathbf{0}_3 & {}^{I_{1|0}}_{G}\hat{\mathbf{R}}{}^G\mathbf{g} \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{I_{1|0}} \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & -\lfloor {}^G\hat{\mathbf{v}}_{I_{1|0}} \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{f_{1|0}} \rfloor {}^G\mathbf{g} \end{bmatrix} \tag{61}$$

This property is maintained by OC through finding the best $\mathbf{\Phi'}(k+1,k)$ and $\mathbf{H}_k^{'}$ such that:

$$\mathbf{N}_{k+1} = \mathbf{\Phi}^{'}(k+1,k)\mathbf{N}_k \tag{62}$$

$$\mathbf{H}_k^{'}\mathbf{N}_k = \mathbf{0}, \quad \forall k \geq 0 \tag{63}$$

where $\mathbf{N}_k$ denotes the system unobservable subspace at timestep k *propagated* from the initial chosen $\mathbf{N}_0$. $\hat{\mathbf{H}}_{k+1}$ denotes the measurement Jacobian evaluating with current state estimates at timestamp t_{k+1} and t_k. $\hat{\mathbf{\Phi}}(k+1,k)$ is the standard state translation matrix (26) that propogate the state from timestamp t_k to t_{k+1}. Modificition rules are proposed in OC [7] to maintain the correction nullspace $\hat{\mathbf{N}}_k$ at each timestamp and enforce the unobservable directions. Bacically, the first block column of $\hat{\mathbf{N}}_k$ in the first constraint Eq. (62) is automatically satisfied because of the structure of $\hat{\mathbf{\Phi}}$ Eq. (26). We need to satisfy the following equation:

$$\begin{bmatrix} {}^{I_{k+1|k}}_{G}\hat{\mathbf{R}}{}^G\mathbf{g} \\ -\lfloor {}^G\hat{\mathbf{p}}_{I_{k+1|k}} \rfloor {}^G\mathbf{g} \\ -\lfloor {}^G\hat{\mathbf{v}}_{I_{k+1|k}} \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 \\ \mathbf{0}_3 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{\Phi}}_{11} & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\mathbf{\Phi}}_{14} & \mathbf{0}_3 \\ \hat{\mathbf{\Phi}}_{21} & \mathbf{I}_3 & \mathbf{I}_3\Delta t & \hat{\mathbf{\Phi}}_{24} & \hat{\mathbf{\Phi}}_{25} \\ \hat{\mathbf{\Phi}}_{31} & \mathbf{0}_3 & \mathbf{I}_3 & \hat{\mathbf{\Phi}}_{34} & \hat{\mathbf{\Phi}}_{35} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} {}^{I_{k|k-1}}_{G}\hat{\mathbf{R}}{}^G\mathbf{g} \\ -\lfloor {}^G\hat{\mathbf{p}}_{I_{k|k-1}} \rfloor {}^G\mathbf{g} \\ -\lfloor {}^G\hat{\mathbf{v}}_{I_{k|k-1}} \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 \\ \mathbf{0}_3 \end{bmatrix} \tag{64}$$

The following conditions for the first three block rows need to be satisfied. We refer to [7] for more detailed derivations. From the first block row we get

$${}^{I_{k+1|k}}_{G}\hat{\mathbf{R}}{}^G\mathbf{g}\hat{\mathbf{\Phi}}_{11} = {}^{I_{k|k-1}}_{G}\hat{\mathbf{R}}{}^G\mathbf{g} \tag{65}$$

$$\Rightarrow \; \hat{\mathbf{\Phi}}'_{11} = {}^{I_{k+1|k}}_{I_{k|k-1}}\hat{\mathbf{R}} \tag{66}$$

We use $\mathbf{\Phi}'_{ij}$ to denote the target OC discrete-time state translation matrix. The second and third block rows need to satisfy the following equations. Note that these are obtained from the discrete-time state translation matrix [3].

$$\hat{\mathbf{\Phi}}_{21}{}^{I_{k|k-1}}_{G}\hat{\mathbf{R}}^G\mathbf{g} = \lfloor \Delta t^G\hat{\mathbf{v}}_{I_{k|k-1}} + {}^G\hat{\mathbf{p}}_{I_{k|k-1}} - {}^G\hat{\mathbf{p}}_{I_{k+1|k}} \rfloor^G\mathbf{g} \tag{67}$$

$$\hat{\mathbf{\Phi}}_{31}{}^{I_{k|k-1}}_{G}\hat{\mathbf{R}}^G\mathbf{g} = \lfloor {}^G\hat{\mathbf{v}}_{I_{k|k-1}} - {}^G\hat{\mathbf{v}}_{I_{k+1|k}} \rfloor^G\mathbf{g} \tag{68}$$

We seek to find $\mathbf{\Phi}^{'}_{21}$ and $\mathbf{\Phi}^{'}_{31}$ to ensure the above two constraints are satisfied. Note that both of them have the form $\mathbf{\Phi}^{'}_{ij}\mathbf{a}_j = \mathbf{b}_j$ To find the minimal perturbations, we can solve the following minimization problem:

$$\min_{\mathbf{\Phi}^{'}_{ij}} \left\| \mathbf{\Phi}^{'}_{ij} - \hat{\mathbf{\Phi}}_{ij} \right\|^2_{\mathcal{F}} \tag{69}$$

$$\text{subject to} \quad \mathbf{\Phi}^{'}_{ij}\mathbf{a}_j = \mathbf{b}_j \tag{70}$$

where $\|\cdot\|_{\mathcal{F}}$ is the Frobenious norm. The constrained minimization problem will give solution to construct the optimal OC state translation matrix $\mathbf{\Phi}(k+1,k)'$. After employing the method of Lagrange multipliers, and solving the corresponding KKT optimality conditions, the optimal $\mathbf{\Phi}^{'}_{ij}$ can be found as:

$$\mathbf{\Phi}^{'}_{ij} = \hat{\mathbf{\Phi}}_{ij} - \left(\hat{\mathbf{\Phi}}_{ij}\mathbf{a}_j - \mathbf{b}_j \right) (\mathbf{a}_j^\top\mathbf{a}_j)^{-1}\mathbf{a}_j^\top \tag{71}$$

The modified $\mathbf{\Phi}'(k+1,k)$ becomes a valid state translation matrix with the semi-group property [e.g., it satisfies Eq. (58)]. Then we seek to modify the measurement Jacobian $\hat{\mathbf{H}}_{k+1}$ and find the one satisfy the second constraint Eq. (63). Given the linearized measurement residual equation Eq. (44) at timestamp t_{k+1} we preserve Eq. (63) through follows:

$$\mathbf{0} = \hat{\mathbf{H}}_{k+1}\hat{\mathbf{N}}_{k+1} \tag{72}$$

$$= \nabla\hat{\mathbf{h}} \begin{bmatrix} \lfloor {}^{I_{k+1|k}}_{G}\hat{\mathbf{R}}({}^G\hat{\mathbf{p}}_{f_{k+1|k}} - {}^G\hat{\mathbf{p}}_{I_{k+1|k}}) \rfloor & -{}^{I_{k+1|k}}_{G}\hat{\mathbf{R}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & {}^{I_{k+1|k}}_{G}\hat{\mathbf{R}} \end{bmatrix} \begin{bmatrix} \mathbf{0}_3 & {}^{I_{k+1|k}}_{G}\hat{\mathbf{R}}^G\mathbf{g} \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{I_{k+1|k}} \rfloor^G\mathbf{g} \\ \mathbf{0}_3 & -\lfloor {}^G\hat{\mathbf{v}}_{I_{k+1|k}} \rfloor^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{f_{1|0}} \rfloor^G\mathbf{g} \end{bmatrix} \tag{73}$$

Since $\hat{\mathbf{H}}_{\mathbf{p}_{k+1}} = -\hat{\mathbf{H}}_{f_{k+1}} = -{}^{I_{k+1|k}}_{G}\hat{\mathbf{R}}$, the first column of $\mathbf{N}_{k+1}$ is satisfied. Therefore, we then need to satisfy the second column as:

$$\mathbf{0} = \begin{bmatrix} \hat{\mathbf{H}}_{\boldsymbol{\theta}_{k+1}} & \hat{\mathbf{H}}_{\mathbf{p}_{k+1}} \end{bmatrix} \begin{bmatrix} {}^{I_{k+1|k}}_{G}\hat{\mathbf{R}}^G\mathbf{g} \\ -\lfloor {}^G\hat{\mathbf{p}}_{I_{k+1|k}} \rfloor^G\mathbf{g} + \lfloor {}^G\hat{\mathbf{p}}_{f_{1|0}} \rfloor^G\mathbf{g} \end{bmatrix} \tag{74}$$

$$= \hat{\mathbf{A}}_{k+1}\mathbf{h}_{k+1} \tag{75}$$

Note that Eq. (74) is of form $\hat{\mathbf{H}}_{k+1}\mathbf{h}_{k+1} = \mathbf{0}$. We compute the optimal $\mathbf{H}'_{k+1}$ with the similar minimization problem:

$$\min_{\mathbf{A}'_{k+1}} \left\| \mathbf{A}'_{k+1} - \hat{\mathbf{A}}_{k+1} \right\|^2_{\mathcal{F}} \tag{76}$$

$$\text{subject to} \quad \mathbf{A}'_{k+1}\mathbf{h}_{k+1} = \mathbf{0} \tag{77}$$

The optimal Jacobian $\mathbf{A}'_{k+1}$ can be computed as:

$$\mathbf{A}'_{k+1} = \hat{\mathbf{A}}_{k+1} - \hat{\mathbf{A}}_{k+1}\mathbf{h}_{k+1}(\mathbf{h}_{k+1}^\top \mathbf{h}_{k+1})^{-1}\mathbf{h}_{k+1}^\top \tag{78}$$

The Jacobian $\mathbf{H}'_{k+1}$ can be recovered by $\mathbf{A}'_{k+1}$ as

$$\mathbf{A}'_{k+1} = \begin{bmatrix} \hat{\mathbf{H}}'_{\boldsymbol{\theta}_{k+1}} & \hat{\mathbf{H}}'_{\mathbf{p}_{k+1}} \end{bmatrix} \tag{79}$$

$$\Rightarrow \quad \hat{\mathbf{H}}'_{k+1} = \begin{bmatrix} \hat{\mathbf{H}}'_{\boldsymbol{\theta}_{k+1}} & \hat{\mathbf{H}}'_{\mathbf{p}_{k+1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \hat{\mathbf{H}}'_{\mathbf{p}_{f_{k+1}}} \end{bmatrix} \tag{80}$$

where $\hat{\mathbf{H}}'_{\mathbf{p}_{f_{k+1}}} = -\hat{\mathbf{H}}'_{\mathbf{p}_{k+1}}$. With the modified Jacobians, we process the filter updated introduced in Section 1.2.

2.1.2 FEJ-VINS

FEJ-based VINS [9, 12] directly selects the first ever available estimate, denoted by $\bar{\mathbf{x}}$, for each state variable as linearization points at all future timesteps. During propagation, FEJ evaluates the state translation matrix $\bar{\boldsymbol{\Phi}}(k+1,k)$ with *propagated* state estimates instead of the *updated* ones to ensure $\bar{\boldsymbol{\Phi}}(k+1,k)$ has a valid semi-group property (e.g., satisfies Eq. (58) [10, Lemma 4.2]).

$$\bar{\boldsymbol{\Phi}}(k+1,k-1) = \bar{\boldsymbol{\Phi}}(k+1,k)\bar{\boldsymbol{\Phi}}(k,k-1) \tag{81}$$

The analytical state translation matrix can be derived as:

$$\bar{\boldsymbol{\Phi}}(k+1,k) = \begin{bmatrix} \bar{\boldsymbol{\Phi}}_{11} & \mathbf{0}_3 & \mathbf{0}_3 & \bar{\boldsymbol{\Phi}}_{14} & \mathbf{0}_3 & \mathbf{0}_3 \\ \bar{\boldsymbol{\Phi}}_{21} & \mathbf{I}_3 & \mathbf{I}_3\Delta t & \bar{\boldsymbol{\Phi}}_{24} & \bar{\boldsymbol{\Phi}}_{25} & \mathbf{0}_3 \\ \bar{\boldsymbol{\Phi}}_{31} & \mathbf{0}_3 & \mathbf{I}_3 & \bar{\boldsymbol{\Phi}}_{34} & \bar{\boldsymbol{\Phi}}_{35} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_{3\times 3M} & \mathbf{0}_{3\times 3M} & \mathbf{0}_{3\times 3M} & \mathbf{0}_{3\times 3M} & \mathbf{0}_{3\times 3M} & \mathbf{I}_{3M} \end{bmatrix} \tag{82}$$

where $\Delta t = t_{k+1} - t_k$ and:

$$\bar{\boldsymbol{\Phi}}_{11} = \exp\left(-\int_{t_k}^{t_{k+1}} \lfloor \hat{\boldsymbol{\omega}}(\tau) \rfloor \mathrm{d}\tau\right) \tag{83}$$

$$\bar{\boldsymbol{\Phi}}_{14} = -\int_{t_k}^{t_{k+1}} {}^{I_{k+1|k}}_{I_\tau}\hat{\mathbf{R}}\mathrm{d}\tau \tag{84}$$

$$\bar{\boldsymbol{\Phi}}_{21} = -\lfloor {}^G\hat{\mathbf{p}}_{I_{k+1|k}} - {}^G\hat{\mathbf{p}}_{I_{k|k-1}} - {}^G\hat{\mathbf{v}}_{I_{k|k-1}}\Delta t + \frac{1}{2}{}^G\mathbf{g}\Delta t^2 \rfloor {}^{I_{k|k-1}}_{G}\hat{\mathbf{R}}^\top \tag{85}$$

$$\bar{\boldsymbol{\Phi}}_{24} = \int_{t_k}^{t_{k+1}} \int_{t_1}^{\theta} {}^G_{I_s}\hat{\mathbf{R}} \lfloor {}^{I_s}\hat{\mathbf{a}} \rfloor \int_{t_k}^{s} {}^{I_s}_{I_\tau}\hat{\mathbf{R}}\mathrm{d}\tau\mathrm{d}s\mathrm{d}\theta \tag{86}$$

$$\bar{\boldsymbol{\Phi}}_{25} = -\int_{t_k}^{t_{k+1}} \int_{t_k}^{s} {}^G_{I_\tau}\hat{\mathbf{R}}\mathrm{d}\tau\mathrm{d}s \tag{87}$$

$$\bar{\boldsymbol{\Phi}}_{31} = -\lfloor {}^G\hat{\mathbf{v}}_{I_{k+1|k}} - {}^G\hat{\mathbf{v}}_{I_{k|k-1}} + {}^G\mathbf{g}\Delta t \rfloor {}^{I_{k|k-1}}_{G}\hat{\mathbf{R}}^\top \tag{88}$$

$$\bar{\boldsymbol{\Phi}}_{34} = \int_{t_k}^{t_{k+1}} {}^{I_s}_{G}\hat{\mathbf{R}}^\top \lfloor {}^{I_s}\hat{\mathbf{a}} \rfloor \int_{t_k}^{s} {}^{I_\tau}_{I_s}\hat{\mathbf{R}}^\top \mathrm{d}\tau\mathrm{d}s \tag{89}$$

$$\bar{\boldsymbol{\Phi}}_{35} = -\int_{t_k}^{t_{k+1}} {}^{I_\tau}_{G}\hat{\mathbf{R}}^\top \mathrm{d}\tau \tag{90}$$

FEJ also selects the first state estimates $\bar{\mathbf{x}}$ to construct measurement Jacobian $\bar{\mathbf{H}}_{k+1}$ and its linearized residual as:

$$
\begin{aligned}
\bar{\mathbf{r}}_{k+1} &\simeq \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) && (91)\\
&\simeq \begin{bmatrix} \bar{\mathbf{H}}_{I_{k+1}} & \bar{\mathbf{H}}_{f_{k+1}} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{I_{k+1|k}} \\ \tilde{\mathbf{x}}_{f_{k+1|k}} \end{bmatrix} + \mathbf{n}_{k+1} && (92)\\
&= \bar{\mathbf{H}}_{k+1}\tilde{\mathbf{x}}_{k+1|k} + \mathbf{n}_{k+1} && (93)
\end{aligned}
$$

where $\bar{\mathbf{H}}_{k+1}$ denotes the Jacobian evaluated at $\bar{\mathbf{x}}_{k+1|k}$. Note that IMU pose is only used one-time for system linearization, thus $\hat{\mathbf{x}}_{I_{k+1|k}} = \bar{\mathbf{x}}_{I_{k+1|k}}$. Assume the feature is first detected at timestep t_0, as it has zero dynamics, we have ${}^G\bar{\mathbf{p}}_{f_i} = {}^G\hat{\mathbf{p}}_{f_{i,1|0}}$. Therefore, $\bar{\mathbf{H}}_{k+1}$ can be computed as:

$$
\begin{aligned}
\bar{\mathbf{H}}_{k+1} &= \begin{bmatrix} \bar{\mathbf{H}}_{I_{k+1}} & \bar{\mathbf{H}}_{f_{k+1}} \end{bmatrix} && (94)\\
&= \nabla\bar{\mathbf{h}}_i \left[\begin{array}{ccccc|ccc} \bar{\mathbf{H}}_{\theta_{k+1}} & \bar{\mathbf{H}}_{\mathbf{p}_{k+1}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \bar{\mathbf{H}}_{f_{i,k+1}} & \dots \end{array}\right] && (95)
\end{aligned}
$$

where

$$
\begin{aligned}
\bar{\mathbf{H}}_{\theta_{k+1}} &= \lfloor {}^{I_{k+1|k}}_{G}\hat{\mathbf{R}} \left({}^G\hat{\mathbf{p}}_{f_{i,1|0}} - {}^G\hat{\mathbf{p}}_{I_{k+1|k}} \right) \rfloor && (96)\\
\hat{\mathbf{H}}_{p_{k+1}} &= -\bar{\mathbf{H}}_{f_{k+1}} = -{}^{I_{k+1|k}}_{G}\hat{\mathbf{R}} && (97)\\
\nabla\bar{\mathbf{h}}_i &= \frac{1}{\bar{z}_i{}^2} \begin{bmatrix} \bar{z}_i & 0 & -\bar{x}_i \\ 0 & \bar{z}_i & -\bar{y}_i \end{bmatrix} {}^C_I\mathbf{R} && (98)
\end{aligned}
$$

Since there are no linearization point changes,the unobservable subspace dimensions are automatically preserved, which can be verified as follows.

$$
\begin{aligned}
\mathbf{N}_{k+1} &= \bar{\boldsymbol{\Phi}}(k+1,k)\mathbf{N}_k && (99)\\
\bar{\mathbf{H}}_k\mathbf{N}_k &= \mathbf{0} && (100)\\
\Rightarrow \quad \bar{\mathbf{H}}_{k+1}\bar{\boldsymbol{\Phi}}(k+1,0)\mathbf{N}_0 &= \mathbf{0}, \quad \forall k \geq 0 && (101)
\end{aligned}
$$

We now show that the underlying FEJ assumption can possibly incur large linearization errors by re-deriving the linearized measurement with a first order Taylor series expansion at $\bar{\mathbf{x}}$ (subscript $k+1$ dropped for clarity):

$$
\begin{aligned}
\mathbf{z} &\simeq \mathbf{h}(\bar{\mathbf{x}}) + \bar{\mathbf{H}}(\mathbf{x} \boxminus \bar{\mathbf{x}}) + \mathbf{n} && (102)\\
&= \mathbf{h}(\bar{\mathbf{x}}) + \bar{\mathbf{H}}(\hat{\mathbf{x}} \boxminus \bar{\mathbf{x}}) + \bar{\mathbf{H}}(\mathbf{x} \boxminus \hat{\mathbf{x}}) + \mathbf{n} && (103)\\
&\simeq \mathbf{h}(\hat{\mathbf{x}}) + \bar{\mathbf{H}}\tilde{\mathbf{x}} + \mathbf{n} && (104)
\end{aligned}
$$

FEJ estimator uses Eq. (104) during update (both residual and error state are at the current estimates). Although $\bar{\mathbf{H}}$ ensures the system unobservable subspace, it might lead the system to suffer from un-optimal updates due to poor $\bar{\mathbf{x}}$. Specifically, FEJ assumes the following:

$$
\boxed{\mathbf{h}(\hat{\mathbf{x}}) \simeq \mathbf{h}(\bar{\mathbf{x}}) + \bar{\mathbf{H}}(\hat{\mathbf{x}} \boxminus \bar{\mathbf{x}})} \qquad (105)
$$

This approximation can introduce extra linearization errors caused by the estimates change ($\hat{\mathbf{x}} \boxminus \bar{\mathbf{x}}$), especially when *poor* initial estimates $\bar{\mathbf{x}}$ are used.

2.2 FEJ2 Methodology

The above estimator designs from observability perspective avoid spurious information erroneously flooding into the unobservable directions of the state space, thus improving consistency. As they rely on initial estimates to construct Jacobians, the estimator may suffer from the poor state initialization. To address this issue, the proposed FEJ2 consistently propagates state forward leveraging either FEJ or OC but improves the update step. We seek to linearize the system at the *current* state estimates $\hat{\mathbf{x}}$ for the smallest linearization errors and use the *first-estimates Jacobian* $\bar{\mathbf{H}}$ to avoid extra information gain along unobservable directions.

To achieve this, FEJ2 derives a more accurate linear model to approximate the nonlinear system as:

$$\begin{aligned}
\mathbf{z}_{k+1} &\simeq \mathbf{h}(\hat{\mathbf{x}}_{k+1}) + \hat{\mathbf{H}}_{k+1}\tilde{\mathbf{x}}_{k+1} + \mathbf{n}_{k+1} && (106)\\
&= \mathbf{h}(\hat{\mathbf{x}}_{k+1}) + (\bar{\mathbf{H}}_{k+1} + \hat{\mathbf{H}}_{k+1} - \bar{\mathbf{H}}_{k+1})\tilde{\mathbf{x}}_{k+1} + \mathbf{n}_{k+1} && (107)\\
\hat{\mathbf{r}}_{k+1} &= \mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1}) && (108)\\
&\simeq \bar{\mathbf{H}}_{k+1}\tilde{\mathbf{x}}_{k+1} + \Delta\mathbf{H}_{k+1}\tilde{\mathbf{x}}_{k+1} + \mathbf{n}_{k+1} && (109)
\end{aligned}$$

where $\Delta\mathbf{H}_{k+1} = \hat{\mathbf{H}}_{k+1} - \bar{\mathbf{H}}_{k+1}$ indirectly captures linearization points change between *first* and *current* state estimates. Therefore, we project Eq. (109) onto the left nullspace of $\Delta\mathbf{H}$ using the QR decomposition, i.e., $\Delta\mathbf{U}_{k+1}^{\top}\Delta\mathbf{H}_{k+1} = \mathbf{0}$.

$$\begin{aligned}
\Delta\mathbf{H}_{k+1} &= \begin{bmatrix}\Delta\mathbf{Q}_{k+1} & \Delta\mathbf{U}_{k+1}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{T}_{k+1}\\ \mathbf{0}\end{bmatrix} && (110)\\
&= \Delta\mathbf{Q}_{k+1}\Delta\mathbf{T}_{k+1} && (111)
\end{aligned}$$

where $\begin{bmatrix}\Delta\mathbf{Q}_{k+1} & \Delta\mathbf{U}_{k+1}\end{bmatrix}$ is a unitary matrix, columns of $\Delta\mathbf{Q}_{k+1}$ and $\Delta\mathbf{U}_{k+1}$ form bases for the range and nullspace of $\Delta\mathbf{H}_{k+1}$, respectively. $\Delta\mathbf{T}_{k+1}$ is an upper triangular matrix. Multiplying $\Delta\mathbf{U}_{k+1}^{\top}$ to Eq. (109) resulting the measurement residual $\mathbf{r}_{k+1}^{*}$ and Jacobian $\mathbf{H}_{k+1}^{*}$ as:

$$\begin{aligned}
\Delta\mathbf{U}_{k+1}^{\top}\hat{\mathbf{r}}_{k+1} &= \Delta\mathbf{U}_{k+1}^{\top}\bar{\mathbf{H}}_{k+1}\tilde{\mathbf{x}}_{k+1} + \Delta\mathbf{U}_{k+1}^{\top}\Delta\mathbf{H}_{k+1}\tilde{\mathbf{x}}_{k+1} + \Delta\mathbf{U}_{k+1}^{\top}\mathbf{n}_{k+1}\\
&= \Delta\mathbf{U}_{k+1}^{\top}\bar{\mathbf{H}}_{k+1}\tilde{\mathbf{x}}_{k+1} + \Delta\mathbf{U}_{k+1}^{\top}\mathbf{n}_{k+1} && (112)\\
\Rightarrow \quad \mathbf{r}_{k+1}^{*} &= \mathbf{H}_{k+1}^{*}\tilde{\mathbf{x}}_{k+1} + \mathbf{n}_{k+1}^{*} && (113)
\end{aligned}$$

where $\mathbf{n}_{k+1}$ is zero mean independent and identically distributed (i.i.d) Gaussian noise with covariance $\mathbf{R}_{k+1}$ [13] and $\mathbf{n}^{*} \sim \mathcal{N}(\mathbf{0}, \Delta\mathbf{U}_{k+1}^{\top}\mathbf{R}_{k+1}\Delta\mathbf{U}_{k+1})$. We then update the filter with the modified $\mathbf{r}_{k+1}^{*}$ and $\mathbf{H}_{k+1}^{*}$. An overview of different estimators is given in Algorithm 1.

As evident from the above, the QR factorization is leveraged to find the left nullspace $\Delta\mathbf{U}$ and get rid of $\Delta\mathbf{H}$ via $\Delta\mathbf{U}^{\top}\Delta\mathbf{H} = \mathbf{0}$ (subscript $k+1$ dropped for clarity). In order to find such nullspace, $\Delta\mathbf{H}_{m\times n}$ is required to be full column rank, where m and n are the size of measurements and states, respectively. Assuming $\texttt{rank}(\Delta\mathbf{H}) = n$, by the rank-nullity theorem, we find $\texttt{nullity}(\Delta\mathbf{U}) = m - n$, that is, the dimension of the left nullspace $\dim(\Delta\mathbf{U})$ and also the *compressed* measurement size [$\mathbf{r}^{*}$ in Eq. (109)] after the nullspace operation. While this is the case for many multi-view scenarios, some systems may have less visual observations than the dimensions of the state (i.e., $m \not> n$). For example, assuming M environmental features are observed by the monocular camera Eq. (37), we can find the size of measurements $m = 2M$, while the state consists the 6 d.o.f IMU pose and the 3D feature positions, thus $n = 6 + 3M$. We notice that $2M \not> 6 + 3M, \forall M \geq 0$. If this occurs, it is not difficult to address in an engineering sound way as discussed in Section 2.4.2.

2.3 FEJ2 Properties

We now show that FEJ2 not only preserves the proper observability properties but compensates for large linearization errors as in the FEJ-based estimators. Note that, in what follows, we drop off the subscripts to make the ensuring derivations more concise.

Lemma 2.1. *FEJ2 employs a linearized system model that has an unobservable subspace of correct dimensions and structure, and shares the same (initial) nullspace of the observability matrix as the FEJ and OC.*

Proof. The observability matrix of the FEJ ($\bar{\mathcal{O}}$) and OC ($\mathcal{O}'$) assume the following conditions:

$$\text{OC:} \quad \mathcal{O}'\mathbf{N}_0 = \mathbf{0} \tag{114}$$

$$\text{FEJ:} \quad \bar{\mathcal{O}}\mathbf{N}_0 = \bar{\mathbf{H}}\bar{\mathbf{\Phi}}\mathbf{N}_0 = \mathbf{0} \tag{115}$$

where $\mathbf{N}_0$ is the selected initial system unobservable subspace Eq.(61). It is not difficult to show that FEJ2 shares the same structure of observability matrix with FEJ and OC:

$$\mathcal{O}^*\mathbf{N}_0 = \mathbf{H}^*\bar{\mathbf{\Phi}}\mathbf{N}_0 = \Delta\mathbf{U}^\top\bar{\mathbf{H}}\bar{\mathbf{\Phi}}\mathbf{N}_0 = \Delta\mathbf{U}^\top\bar{\mathcal{O}}\mathbf{N}_0 = \mathbf{0}$$

This proves that FEJ2 keeps the *correct* unobservable subspace. □

Lemma 2.2. *FEJ2 has larger covariance estimates than FEJ.*

Proof. Without losing generality, we assume the noise covariance is isotropic and identical (i.e., $\mathbf{R} = \mathbf{I}$). After the QR factorization for $\Delta\mathbf{H}$ as in Eq. (110), due to the unitary property we have $\Delta\mathbf{Q}\Delta\mathbf{Q}^\top + \Delta\mathbf{U}\Delta\mathbf{U}^\top = \mathbf{I}$. The information matrix of FEJ2 can then be derived as:

$$\mathbf{\Sigma}^* = \mathbf{H}^{*\top}\mathbf{H}^* \tag{116}$$

$$= (\Delta\mathbf{U}^\top\bar{\mathbf{H}})^\top(\Delta\mathbf{U}^\top\bar{\mathbf{H}}) \tag{117}$$

$$= \bar{\mathbf{H}}^\top\bar{\mathbf{H}} - \bar{\mathbf{H}}^\top\Delta\mathbf{Q}\Delta\mathbf{Q}^\top\bar{\mathbf{H}} \tag{118}$$

$$\triangleq \bar{\mathbf{\Sigma}} - \Delta\mathbf{\Sigma} \tag{119}$$

Notice that $\bar{\mathbf{\Sigma}} = \bar{\mathbf{H}}^\top\bar{\mathbf{H}}$ is the FEJ information matrix, $\Delta\mathbf{\Sigma}$ is a positive semidefinite matrix. Eq. (119) shows FEJ2 deflates the information matrix of FEJ, which, in turn, inflates the covariance matrix $\mathbf{P}$ of system since $\mathbf{P} = \mathbf{\Sigma}^{-1}$. □

Basically, FEJ2 projects the measurement residual function onto the left nullspace of $\Delta\mathbf{H}$. As $\Delta\mathbf{H}$ is the difference between Jacobians evaluate with $\bar{\mathbf{x}}$ and $\hat{\mathbf{x}}$, a general consideration is it represents the linearization errors. In addition, since $\hat{\mathbf{H}} = \bar{\mathbf{H}} + \Delta\mathbf{H}$, where $\bar{\mathbf{H}}$ ensures that the linearized system keeps the same observability property as the underlying nonlinear system, $\Delta\mathbf{H}$ causes *extra* information gain along unobservable directions. At this point, we see that FEJ2 is a more consistent and accurate estimator which guards the system observability properties and better models the system uncertainty compared to FEJ.

2.4 FEJ2 Applications

FEJ2 can be implemented to different VINS frameworks as a base estimator design. In this section we continue to address the FEJ2 nullpace projection operation and explain how to utilize FEJ2 idea into SLAM and multi-state constraint Kalman filter (MSCKF) based visual-inertial odometry (VIO) [14].

Algorithm 1 VINS Algorithm Overview

```
Propagation
    Input : IMU measurement, syetem state estimate $\hat{\mathbf{x}}_{k|k}$ and covariance $\hat{\mathbf{P}}_{k|k}$
    Output: system propagated state estimate $\hat{\mathbf{x}}_{k+1|k}$ and covariance $\hat{\mathbf{P}}_{k+1|k}$
        •Propagate system state estimate via Eq.(3)-(8)
        •Compute the state translation matrix $\mathbf{\Phi}(k+1,k)$
            If do Standard
                $\mathbf{\Phi}(k+1,k) = \hat{\mathbf{\Phi}}(k+1,k)$ via Eq.(26)
            else
                $\mathbf{\Phi}(k+1,k) = \bar{\mathbf{\Phi}}(k+1,k)$ via Eq. (82)
        •Propagate the state covariance matrix via Eq.(35)
                $\hat{\mathbf{P}}_{k+1|k} = \mathbf{\Phi}(k+1,k)\hat{\mathbf{P}}_{k|k}\ \mathbf{\Phi}_I(k+1,k)^\top + \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^\top$
Update
    Input : Sensor observations, propogated state estimate $\hat{\mathbf{x}}_{k+1|k}$ and covariance $\hat{\mathbf{P}}_{k+1|k}$
    Output: System updated state estimate $\hat{\mathbf{x}}_{k+1|k+1}$ and covariance $\hat{\mathbf{P}}_{k+1|k+1}$
        •Receive feature measurements $\mathbf{z}_{k+1}$
            Note that we assume all features have been observed and initialized before
        •Compute the measurement residual $\mathbf{r}_{k+1}$ and the Jacobian $\mathbf{H}_{k+1}$
            if do Standard
                - Linearize the measurement function at the *best* state estimate
                  $\mathbf{r}_{k+1} = \hat{\mathbf{r}}_{k+1}$ , $\mathbf{H}_{k+1} = \hat{\mathbf{H}}_{k+1}$ via Eq. (44)
            else if do FEJ
                - Linearize the measurement function at the *first* state estimate
                  $\mathbf{r}_{k+1} \simeq \hat{\mathbf{r}}_{k+1}$ , $\mathbf{H}_{k+1} = \bar{\mathbf{H}}_{k+1}$ via Eq. (93)
            else if do OC
                - Linearize the measurement function at the *best* state estimate
                  Get $\mathbf{r}_{k+1} = \hat{\mathbf{r}}_{k+1}$ and $\hat{\mathbf{H}}_{k+1}$
                - Modify $\hat{\mathbf{H}}_{k+1}$ via Eq. (62) (63). Get $\mathbf{H}_{k+1} = \mathbf{H}'_{k+1}$.
            else if do FEJ2
                - Using linearized standard and FEJ measurement residual functions, Eq. (44)(93)
                  to find the linearized FEJ2 measurement residual equation Eq. (109).
                - Project the measurements to the left nullspace of $\Delta\mathbf{H}$.
                  Get $\mathbf{r}_{k+1} = \mathbf{r}^*_{k+1}, \mathbf{H}_{k+1} = \mathbf{H}^*_{k+1}$
        •Update the filter via Eq. (52)-(54)
         $\mathbf{K}_{k+1} = \mathbf{P}_{k+1}\mathbf{H}^\top_{k+1}\left(\mathbf{H}_{k+1}\mathbf{P}_{k+1}\mathbf{H}^\top_{k+1} + \mathbf{R}_{k+1}\right)^{-1}$
         $\mathbf{x}_{k+1|k+1} = \mathbf{x}_{k+1|k} \boxplus \mathbf{K}_{k+1}\mathbf{r}_{k+1}$
         $\mathbf{P}_{k+1|k+1} = \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{P}_{k+1|k}$
```

2.4.1 FEJ2-EKF SLAM

In the SLAM system, the state vector includes the current IMU state and the features Eq. (2). The standard measurement Jacobians are showed in Eq. (96). To simplify the discussion we assume the robot directly receives relative position measurements of features as:

$$\mathbf{z}_i = {}^{C_{k+1|k}}\mathbf{p}_{f_i} = {}^{C}_{I}\mathbf{R}^{I_{k+1|k}}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{f_{i,k+1|k}} - {}^{G}\mathbf{p}_{I_{k+1|k}}\right) + {}^{C}\mathbf{p}_I + \mathbf{n}_{i,k+1} \tag{120}$$

Assume there are M features in total that can be observed by the robot at every timestep. Stacking all features measurements at timestep t_{k+1} we will get:

$$\mathbf{z}_{k+1} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_M \end{bmatrix} = \begin{bmatrix} {}^{C_{k+1|k}}\mathbf{p}_{f_1} \\ \vdots \\ {}^{C_{k+1|k}}\mathbf{p}_{f_M} \end{bmatrix} + \mathbf{n}_{k+1} \tag{121}$$

Note that given this measurement model we have $\nabla\mathbf{h}_i = \mathbf{I}$; Given the standard Jacobian $\hat{\mathbf{H}}_{k+1}$ Eq. (48) and first-estimates Jacobian $\bar{\mathbf{H}}_{k+1}$ Eq. (95), we can get the structure of $\Delta\mathbf{H}$ as:

$$\Delta\mathbf{H} = \begin{bmatrix} \Delta\mathbf{H}_\theta & \mathbf{0} & \ddots & & \\ \vdots & \vdots & & \mathbf{0} & \\ \Delta\mathbf{H}_\theta & \mathbf{0} & & & \ddots \end{bmatrix}_{m\times n} \tag{122}$$

As $\Delta\mathbf{H}_\theta$ is the only non-zero matrix block, we apply the FEJ2 nullspace operation [see Eq. (113)] to $\Delta\mathbf{H}_\theta$ instead of the full state. It relaxes the matrix rank requirement and only needs $m > 3$, which is common in practice.

2.4.2 FEJ2-MSCKF VIO

In the famous MSCKF VIO framework [15], the state vector includes the latest IMU state $\mathbf{x}_I$ and a sliding window of cloned poses [15]. For those MSCKF features [16] which will be marginalized from state vector immediately, we simply perform FEJ updates.

More attentions are paid to the SLAM features which can be reliably tracked longer than the current sliding window. Those features are initialized into the active state vector and used for update until got lost. The state vector at timestep t_{k+1} thus include:

$$\mathbf{x}_{k+1} = \left[\begin{array}{cc|c} \mathbf{x}^\top_{I_{k+1}} & \mathbf{x}^\top_{\text{clone}} & \mathbf{x}^\top_{f_{k+1}} \end{array}\right] \tag{123}$$

$$= \left[\begin{array}{cc|ccc} \mathbf{x}^\top_{I_{k+1}} & \mathbf{x}^\top_{\text{clone}} & \cdots & {}^{G}\mathbf{p}^\top_{f_{i,k+1}} & \cdots \end{array}\right] \tag{124}$$

where $\mathbf{x}_{\text{clone}}$ refers to all the active cloned poses. As shown in [9], assume ith feature is processed at timestep $\alpha_i + i$ and it has been initialized into state vector with the triangulated value ${}^{G}\bar{\mathbf{p}}_{f_i}$. The $\hat{\mathbf{H}}_{k+1}$ and $\bar{\mathbf{H}}_{k+1}$ with respect to the IMU current IMU state and feature can be computed as:

$$\nabla\hat{\mathbf{h}}_i = \frac{1}{\hat{z}_i{}^2}\begin{bmatrix} \hat{z}_i & 0 & -\hat{x}_i \\ 0 & \hat{z}_i & -\hat{y}_i \end{bmatrix} {}^{C}_{I}\mathbf{R} \qquad \nabla\bar{\mathbf{h}}_i = \frac{1}{\bar{z}_i{}^2}\begin{bmatrix} \bar{z}_i & 0 & -\bar{x}_i \\ 0 & \bar{z}_i & -\bar{y}_i \end{bmatrix} {}^{C}_{I}\mathbf{R}$$

$$\hat{\mathbf{H}}_{\theta_{k+1}} = \lfloor {}^{I_{k+1|\alpha_i}}_{G}\hat{\mathbf{R}}\left({}^{G}\hat{\mathbf{p}}_{f_{k+1|\alpha_i}} - {}^{G}\hat{\mathbf{p}}_{I_{k+1|\alpha_i}}\right)\rfloor \qquad \bar{\mathbf{H}}_{\theta_{k+1}} = \lfloor {}^{I_{k+1|k}}_{G}\hat{\mathbf{R}}\left({}^{G}\bar{\mathbf{p}}_{f_i} - {}^{G}\hat{\mathbf{p}}_{I_{k+1|k}}\right)\rfloor$$

$$\hat{\mathbf{H}}_p = -\hat{\mathbf{H}}_f = -{}^{I_{k+1|\alpha_i}}_{G}\hat{\mathbf{R}} \qquad \bar{\mathbf{H}}_p = -\bar{\mathbf{H}}_f = -{}^{I_{k+1|k}}_{G}\hat{\mathbf{R}}$$

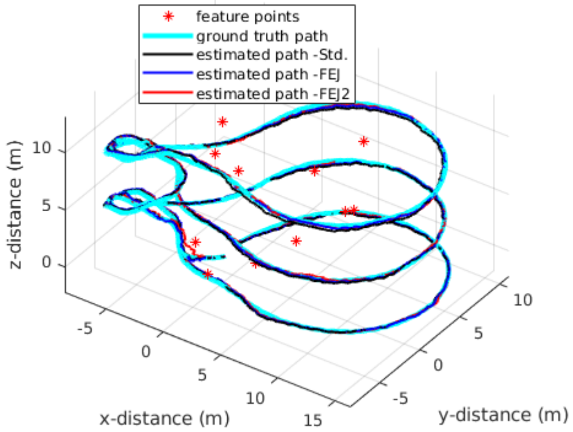

Figure 2: A typical run for proposed estimators on udel_gore dataset.

As normally the measurements from latest image just contain the Jacobians for the current IMU pose and the feature states, the structure of $\Delta\mathbf{H}$ with respect to $\mathbf{x}_{I_{k+1}}$ and SLAM features $\mathbf{x}_f$ can be simplified as:

$$\Delta\mathbf{H}_{k+1} = \begin{bmatrix} \Delta\mathbf{H}_{\theta_1} & \Delta\mathbf{H}_{p_1} & \Delta\mathbf{H}_{f_1} & & \\ \vdots & \vdots & & \ddots & \\ \Delta\mathbf{H}_{\theta_M} & \Delta\mathbf{H}_{p_M} & & & \Delta\mathbf{H}_{f_M} \end{bmatrix} \tag{125}$$

$$= \begin{bmatrix} \Delta\mathbf{H}_{\theta} & \Delta\mathbf{H}_{p} & \Delta\mathbf{H}_{f} \end{bmatrix}_{m\times n} \tag{126}$$

$$= \begin{bmatrix} \Delta\mathbf{H}_{I} & \Delta\mathbf{H}_{f} \end{bmatrix} \tag{127}$$

However, as mentioned before, $\Delta\mathbf{H}$ can not be of full column rank with monocular camera measurements. Even though the left nullspace for $\Delta\mathbf{H}$ exists with stereo measurements, numerical issue happens occasionally from our experiences. Especially, $\Delta\mathbf{H}_f$ is a *small* and *sparse* matrix [see Eq. (96)], each block row only contains one non-zero block to the i-th feature, it could lead to numeral instability when employing QR factorization. Another potential challenge of applying FEJ2 is that, the size of measurements reduces from m to $m-n$ after the nullspace operation Eq. (113), which might cause some geometrical information loss. We will further investigate in the future.

To this end, we propose a flexible and stable FEJ2 design, select to project on the left nullspace of $\Delta\mathbf{H}_I$, which is normally dense and tall, instead of the full error Jacobian.

3 Monte-Carlo Simulations

A series of Monte-Carlo simulations for both SLAM and MSCKF-VIO were conducted under various conditions to validate the preceding theoretical analyses and demonstrate the capability of FEJ2. Note that in the following sections, we use STD to represent standard estimator in the tables and figures.

3.1 EKF SLAM

We first ran a SLAM simulation (as introduced in Section 2.4.1) where a robot following a 3D trajectory as shown in Figure 2 and continuously observes fixed number of landmarks at every timestep. Accordingly, landmarks will be initialized into state vector at the first time. The OpenVINS simulator [16] is leveraged to simulate IMU readings and 20 environmental features. The IMU readings are used to propagate the robot state forward. In this simulation, robot continuously records a pair of relative position measurements of 20 landmarks with noise standard deviation from 1% to 22% of the robot-to-landmark distance. Table 1 reports the averaged values of Root Mean Square Error (RMSE) [17] and Normalized Estimation Error Squared (NEES) [18] over 50 Monte-Carlo runs. The 8 subplots in Figure 3 shows the average RMSE for IMU orientation and position.

Table 1: The average RMSE and NEES over 50 Monte-Carlo simulations for the relative position measurement model with different estimators.

Noise Standard Derivation	**Estimator**	**RMSE Ori.** (deg)	**RMSE Pos.** (m)	**NEES Ori.**	**NEES Pos.**
1%	STD	1.315	0.266	4.055	4.582
	OC	0.266	0.184	2.404	3.435
	FEJ	0.264	0.181	2.423	3.330
	FEJ2	**0.259**	**0.179**	**2.342**	**3.207**
14%	STD	1.370	0.320	3.930	4.740
	OC	0.365	0.255	2.368	3.844
	FEJ	0.366	0.247	2.411	3.628
	FEJ2	**0.352**	**0.244**	**2.282**	**3.502**
18%	STD	1.458	0.376	4.086	4.896
	OC	0.472	0.323	2.397	4.119
	FEJ	0.491	0.311	2.493	3.887
	FEJ2	**0.446**	**0.305**	**2.287**	**3.692**
22%	STD	1.512	0.430	4.156	4.993
	OC	0.582	0.390	2.459	4.367
	FEJ	0.670	0.377	2.666	4.156
	FEJ2	**0.538**	**0.363**	**2.317**	**3.486**

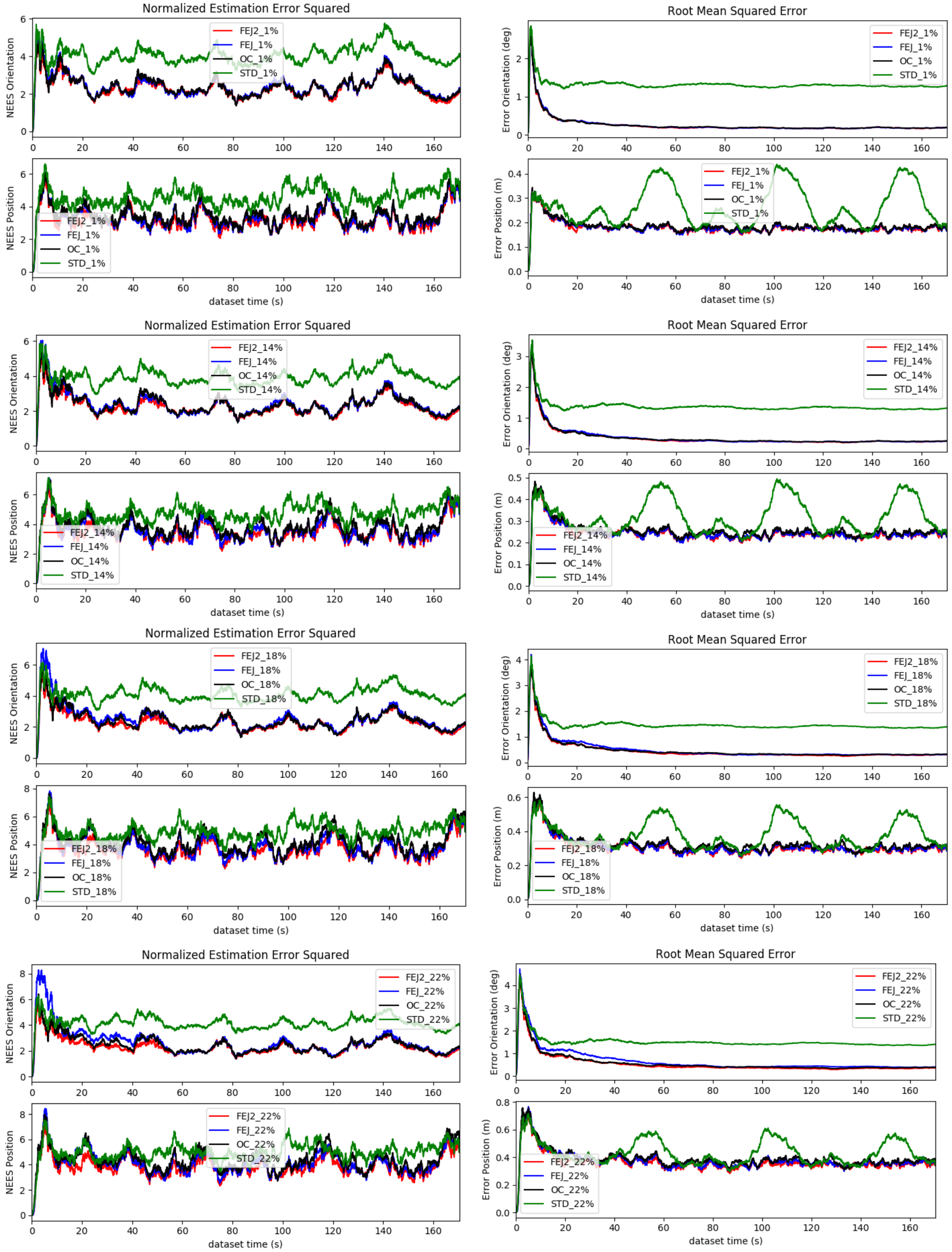

Figure 3: The average RMSE and NEES over 50 Monte-Carlo simulations under different measurement noises

Table 2: Key simulation / estimator parameters for monocular simulation (left) and stereo simulation (right).

Parameter	Value	Parameter	Value
IMU Freq. (hz)	400	Camera Freq. (hz)	10
No. SLAM Feat.	50	Camera Clones	11
Gyro. White Noise	1.6968e-04	Gyro. Rand. Walk	1.9393e-05
Accel. White Noise	2.0000e-03	Accel. Rand. Walk	3.0000e-03

Parameter	Value	Parameter	Value
IMU Freq. (hz)	400	Camera Freq. (hz)	10
No. SLAM Feat.	50	Camera Clones	11
Gyro. White Noise	1.6968e-04	Gyro. Rand. Walk	1.9393e-05
Accel. White Noise	2.0000e-03	Accel. Rand. Walk	3.0000e-03

It is clear from the results in Table 1 that FEJ2 achieves the smallest RMSE compared to other estimators, thus is the most accurate estimator design. Ideally, a consistent estimator should have position and orientation NEES values smaller or close to 3. Table 1 shows the position and orientation NEES values of FEJ2, similar to FEJ and OC, are around 3, which verify our analysis that FEJ2 is a consistent estimator. With a further observation of the results, we notice that both the orientation and position NEES values of FEJ2 are smaller than those of OC and FEJ, which further verify our analysis that FEJ2 has larger covariance than FEJ. Hence, FEJ2 should be more conservative and makes the system less *sensitive* to noises.

3.2 FEJ2 MSCKF VINS

In this section, we present the estimation performance within a hybrid MSCKF-based VIO system [19, 16] using different simulation setups (shown in Table 2) and measurement noises. Specifically, we first simulate monocular camera measurements with relative low IMU noises, and then a stereo camera with increased IMU noises to further challenge the system.

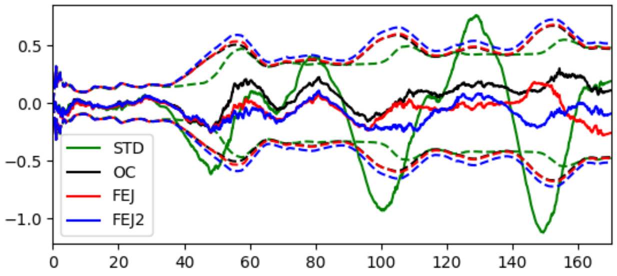

Figure 4: IMU pitch angle errors (solid lines) and $\pm 3\sigma$ bounds (dashed lines) for udel_gore simulations using stereo measurements and 2 pixel measurement noise with different estimators. Note that the $\pm 3\sigma$ bounds of the OC (black) and the FEJ (red) are almost identical.

Figure 4 shows the IMU pitch angle estimation error and the corresponding 3σ bounds of uncertainty with stereo camera using different VINS estimators for a single run. As evident, the standard (STD) estimator is inconsistent with estimation errors frequently out of the 3σ bounds. While the errors for OC, FEJ and FEJ2 are well within their 3σ bounds, indicating their consistency, the 3σ bounds of FEJ2 are slightly broader than the others, showing that FEJ2 is more conservative. These results strengthen our claim that FEJ2 has larger system covariance estimation (Lemma 2.2).

50 runs Monte-Carlo simulations are also conducted. Figure 5 and Figure 6 shows the averaged NEES and RMS errors of robot pose with monocular and stereo measurements with different noises. While Table 3 presents the corresponding average values of RMSE and NEES. Note that in stereo and monocular simulations, different IMU noises are used as showed in Table 2.

From the above results we notice that when measurement noise is small, the average RMSE of OC, FEJ and FEJ2 are very similar to each other since the linearization errors might be small. In the meanwhile, standard (STD) VINS, which is inconsistent, performs the worst with largest

Table 3: The average RMSE and NEES over 50 Monte-Carlo simulations under different measurement noises with monocular measurements (top) and stereo measurements (bottom).

(a) Monocular measurement simulation

Noise (pix.)	Estimator	RMSE Ori. (deg.)	RMSE Pos. (m)	NEES Ori.	NEES Pos.
1	STD	0.412	0.130	23.874	4.911
	OC	0.242	0.119	3.290	3.540
	FEJ	0.242	0.120	3.284	3.617
	FEJ2	**0.238**	**0.118**	**3.150**	**3.443**
2	STD	1.078	0.242	114.724	7.578
	OC	0.453	0.206	3.479	4.074
	FEJ	0.490	0.198	3.759	3.902
	FEJ2	**0.446**	**0.192**	**3.176**	**3.459**
3	STD	2.139	0.402	407.221	13.212
	OC	0.716	0.301	3.964	5.051
	FEJ	0.861	0.289	4.965	4.763
	FEJ2	**0.650**	**0.264**	**3.198**	**3.581**

(b) Stereo measurement simulation

Noise (pix.)	Estimator	RMSE Ori. (deg.)	RMSE Pos. (m)	NEES Ori.	NEES Pos.
1	STD	0.344	0.109	15.447	4.874
	OC	0.257	0.100	3.599	3.416
	FEJ	0.256	0.100	3.438	3.322
	FEJ2	**0.238**	**0.095**	**3.324**	**2.965**
2	STD	1.352	0.207	24.395	5.795
	OC	0.472	0.194	3.851	3.920
	FEJ	0.471	0.193	3.668	3.782
	FEJ2	**0.429**	**0.182**	**3.420**	**3.227**
3	STD	0.344	0.109	15.447	4.874
	OC	0.257	0.100	3.599	3.416
	FEJ	0.256	0.100	3.438	3.322
	FEJ2	**0.238**	**0.095**	**3.324**	**2.965**

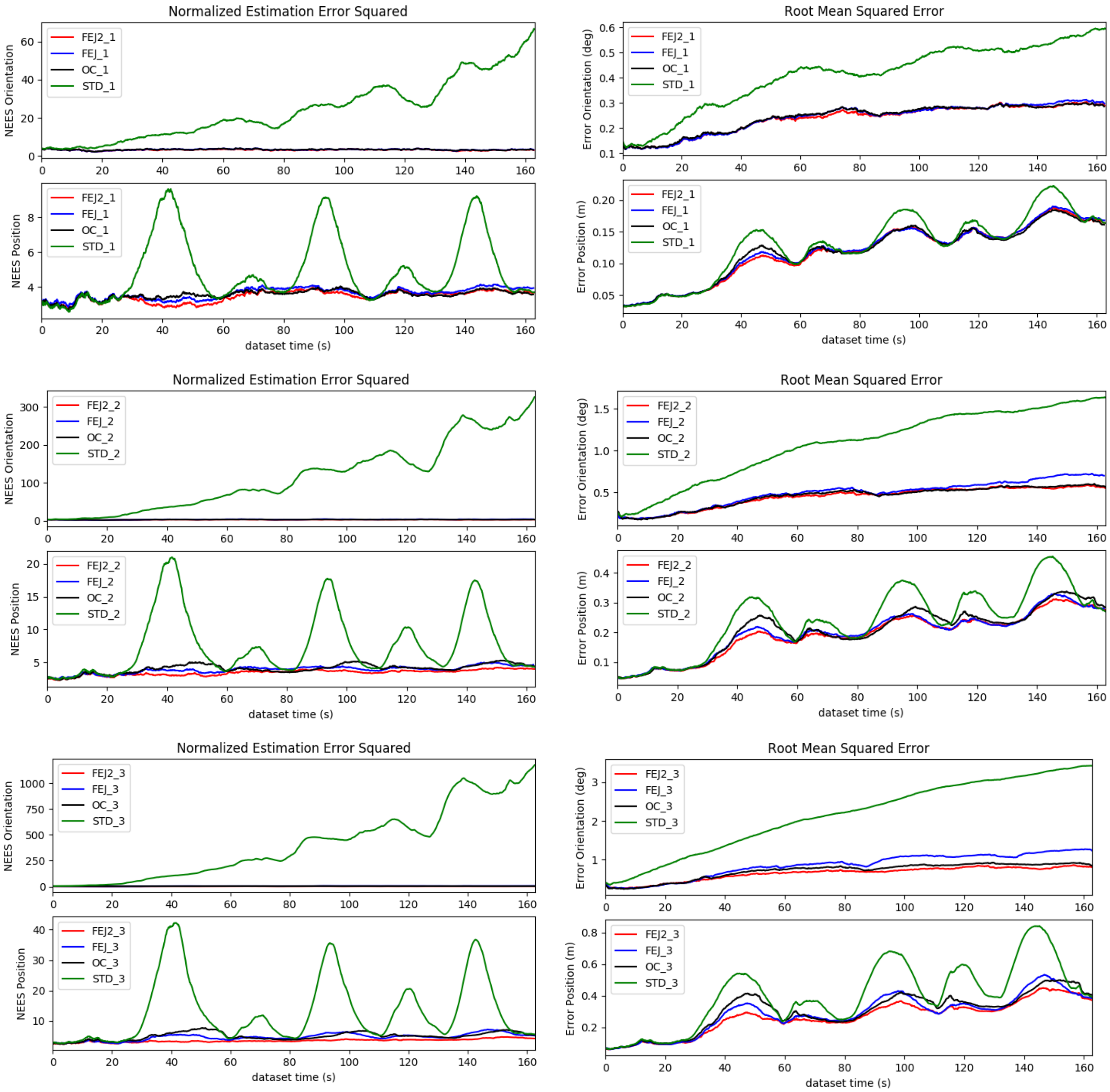

Figure 5: The average RMSE and NEES over 50 Monte-Carlo simulations under different measurement noises (monocular simulation)

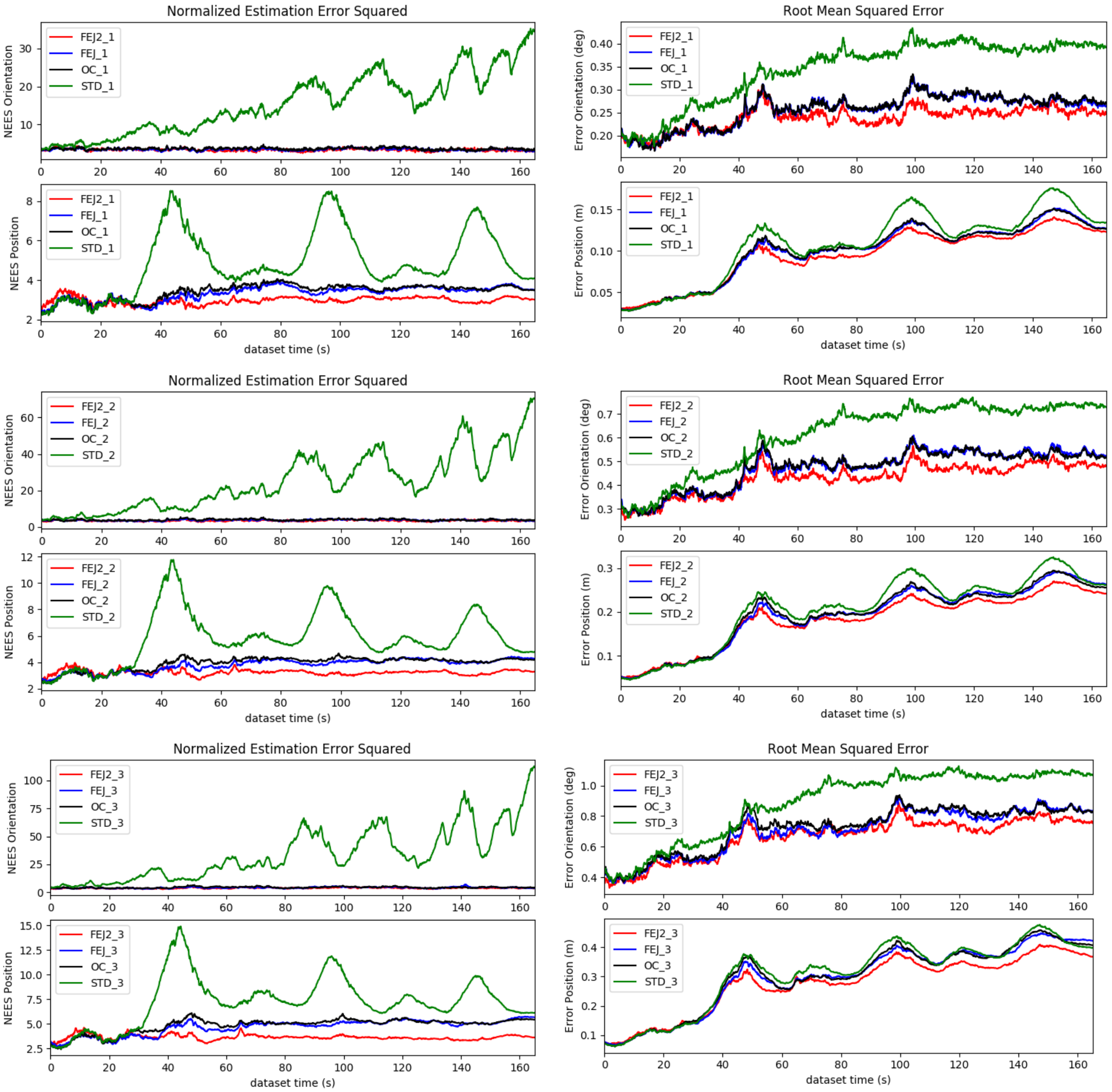

Figure 6: The average RMSE and NEES over 50 Monte-Carlo simulations under different measurement noises (stereo simulation)

Table 4: The average RMSE and NEES over 50 Monte-Carlo simulations for 8 pixel measurement noise with monocular measurements.

Noise (pix.)	**Estimator**	**RMSE Ori.** (deg.)	**RMSE Pos.** (m)	**NEES Ori.**	**NEES Pos.**
8	STD.	-	-	-	-
	OC	3.195	1.104	12.993	25.887
	FEJ	4.711	1.608	32.096	55.259
	FEJ2	**2.258**	**0.677**	**5.980**	**6.387**

estimation errors. When we increase measurement noises, the linearization errors will become more influential and we can observe more apparent accuracy improvements of the FEJ2 compared to other VINS estimators.

We also test with unrealistic noise (i.e., 8 pixel), FEJ and OC diverge quickly due to the large linearization errors, while FEJ2 can still work and outperform the others.

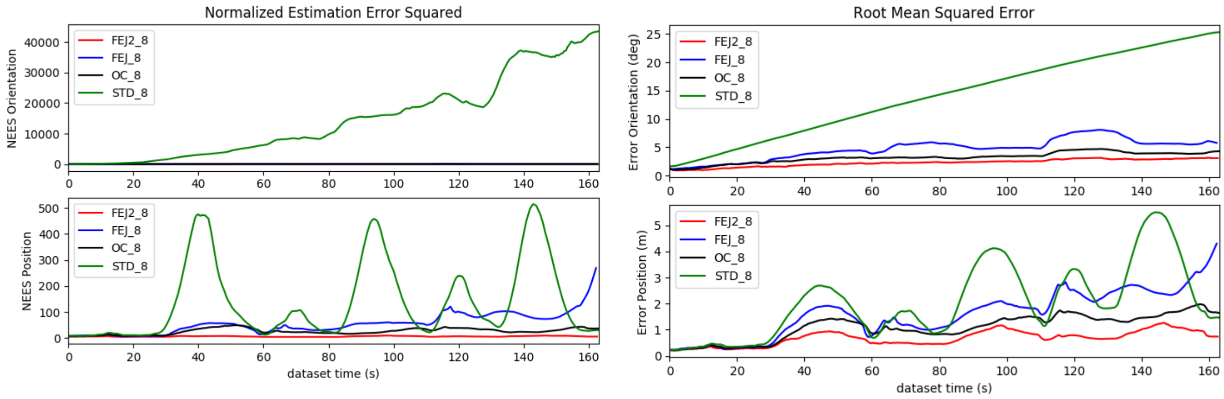

Figure 7: The average RMSE and NEES over 50 Monte-Carlo simulations for 8 pixel measurement noise with monocular measurements.

4 Real-World Experiments

Building upon OpenVINS [16], we further test our FEJ2 based VINS estimators on the Euroc Mav dataset [20] and TUM-VI dataset [21] with both monocular and stereo configurations. In the experiments, we keep 11 clones and at most 50 SLAM features in the state vector. For other tracked features we perform MSCKF update and marginalize them out of state vectors. All estimators were run ten times on each dataset to compensate for randomness and the averaged absolute trajectory error (ATE) [17] values are reported . The results are shown in Table 5, it is clear that OC, FEJ and FEJ2 produce smaller ATE than the standard EKF. Although these three consistent estimators are performing very close to each other, FEJ2 still achieves better accuracy on average, especially in monocular camera scenarios. Compared with stereo, FEJ and OC are more likely to suffer from bad feature triangulation due to less visual constraints in monocular case. This would lead to relatively larger linearization errors, which FEJ2 has focused on, hence, this verifies that FEJ2 can achieve better accuracy.

These results, along with those from Monte-Carlo simulations presented in the previous section, support that our proposed FEJ2 improves the VINS estimator in both consistency and accuracy.

Table 5: Ten runs absolute trajectory error (ATE) for each algorithm in units of degree/meters.

Dataset		**V1_01_easy**	**V1_02_medium**	**V1_03_difficult**	**V2_01_easy**	**V2_02_medium**	**V2_03_difficult**
Mono	STD	0.956/0.076	1.783/0.080	2.638/0.074	0.951/0.098	1.856/0.085	**1.415**/0.154
	OC	0.554/0.077	0.615/0.071	2.933/0.068	0.880/0.093	1.595/0.077	1.835/0.188
	FEJ	0.872/0.056	0.574/**0.052**	**2.079**/0.096	0.928/**0.092**	1.599/0.074	1.874/0.168
	FEJ2	**0.679**/**0.053**	**0.564**/0.059	2.346/**0.061**	**0.791**/0.101	**1.233**/**0.047**	1.808/**0.146**
Stereo	STD	0.792/0.061	1.958/0.059	2.551/0.053	1.078/0.055	1.693/0.064	2.337/0.077
	OC	0.615/0.071	1.772/0.046	**2.468**/**0.045**	1.098/0.059	1.231/0.051	**1.052**/0.061
	FEJ	**0.547**/**0.052**	**1.702**/0.079	2.498/**0.045**	1.172/0.058	1.268/0.049	1.118/**0.058**
	FEJ2	0.564/0.059	1.770/**0.045**	2.503/0.047	**0.975**/**0.053**	**1.202**/**0.047**	1.101/0.062

Dataset		**MH_01_easy**	**MH_02_easy**	**MH_03_medium**	**MH_04_difficult**	**MH_05_difficult**
Mono	STD	1.958/0.246	2.337/0.275	1.355/0.155	1.640/0.306	1.872/0.438
	OC	1.346/0.190	1.121/0.161	1.580/0.162	1.114/0.277	1.124/0.421
	FEJ	1.556/0.107	0.920/0.156	1.204/**0.142**	1.109/**0.261**	1.212/**0.347**
	FEJ2	**1.503**/**0.098**	**0.740**/**0.105**	**1.137**/**0.142**	**0.783**/0.282	**1.145**/0.391
Stereo	STD	1.656/0.183	2.185/0.215	1.641/0.115	1.331/0.223	2.042/0.352
	OC	1.606/0.123	1.136/0.161	1.007/0.183	1.151/0.289	**0.899**/0.288
	FEJ	**1.102**/0.117	**0.968**/**0.142**	**1.157**/0.157	1.090/**0.214**	1.288/0.208
	FEJ2	1.193/**0.070**	1.366/0.174	1.824/**0.138**	**1.038**/0.273	1.124/**0.163**

Dataset		**room1**	**room2**	**room3**	**room4**	**room5**	**room6**
Mono	STD	2.088/0.084	-/-	5.731/0.133	1.449/0.056	**1.032**/0.091	-/-
	OC	**0.811**/0.075	1.205/**0.091**	1.659/0.101	**0.789**/0.050	1.161/0.092	2.642/0.087
	FEJ	0.906/0.074	**1.016**/0.097	**1.287**/**0.085**	0.857/0.048	1.401/0.093	1.301/**0.046**
	FEJ2	0.825/**0.072**	1.331/0.096	1.316/0.086	0.845/**0.045**	1.182/**0.084**	1.277/0.128
Stereo	STD	1.536/0.066	-/-	6.303/0.127	3.220/0.066	**0.895**/0.103	6.019/0.101
	OC	0.788/**0.054**	0.904/**0.060**	1.396/**0.064**	0.880/0.041	0.928/0.099	1.635/0.048
	FEJ	**0.784**/0.056	**0.865**/0.061	1.336/0.065	0.890/0.040	1.014/0.098	1.492/0.051
	FEJ2	0.864/0.058	1.397/0.063	**1.261**/0.068	**0.855**/**0.038**	1.021/**0.097**	**1.225**/**0.039**

References

[1] Nikolas Trawny and Stergios I. Roumeliotis. *Indirect Kalman Filter for 3D Attitude Estimation.* Tech. rep. University of Minnesota, Dept. of Comp. Sci. & Eng., Mar. 2005.

[2] Christoph Hertzberg et al. "Integrating Generic Sensor Fusion Algorithms with Sound State Representations Through Encapsulation of Manifolds". In: *Information Fusion* 14.1 (Jan. 2013), pp. 57–77. ISSN: 1566-2535.

[3] Joel A Hesch et al. *Observability-constrained vision-aided inertial navigation.* Tech. rep. UMN Multiple Autonomous Robotic Systems (MARS) Laboratory, 2005. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.721.6118&rep=rep1&type=pdf.

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

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

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

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

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

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

[10] Guoquan Huang, Michael Kaess, and John Leonard. "Towards Consistent Visual-Inertial Navigation". In: *Proc. of the IEEE International Conference on Robotics and Automation.* Hong Kong, China, May 2014, pp. 4926–4933. DOI: 10.1109/ICRA.2014.6907581.

[11] J.A. Hesch et al. "Camera-IMU-based localization: Observability analysis and consistency improvement". In: *International Journal of Robotics Research* 33 (2014), pp. 182–201.

[12] Guoquan Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. "A First-Estimates Jacobian EKF for Improving SLAM Consistency". In: *Proc. of the 11th International Symposium on Experimental Robotics.* Athens, Greece, July 2008.

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

[14] M. Li and A. I. Mourikis. "Improving the Accuracy of EKF-based Visual-Inertial Odometry". In: *Proc. of the IEEE International Conference on Robotics and Automation.* St. Paul, MN, 2012, pp. 828–835.

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

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

[17] Zichao Zhang and Davide 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.

[18] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. New York: John Willey and Sons, 2001.

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

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

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