

# RBE549 Project4 Visual Inertial Odometry

Haoying Zhou

Department of Robotics Engineering  
Worcester Polytechnic Institute  
Worcester, MA, 01609  
Email: hzhou6@wpi.edu

Zhentian Qian

Department of Robotics Engineering  
Worcester Polytechnic Institute  
Worcester, MA, 01609  
Email: zqian@wpi.edu

## I. METHODOLOGY FOR IMPLEMENTATION

All following functions are based on the references given in the problem descriptions [1, 2, 3, 4]

### A. Initialize Gravity and Bias

The biases of the measured angular velocity  $\mathbf{b}_g \in \mathbb{R}^3$  are calculated by averaging the first few hundreds of IMU messages:

$$\mathbf{b}_g = \frac{1}{N} \sum_{k=1}^N \boldsymbol{\omega}_{m,k} \quad (1)$$

where  $\boldsymbol{\omega}_{m,k} \in \mathbb{R}^3$  is measured angular velocity at time instant  $k$ .

The gravitational acceleration,  ${}^I\mathbf{g} \in \mathbb{R}^3$ , expressed in the local frame, is also calculated in a similar fashion:

$${}^I\mathbf{g} = -\frac{1}{N} \sum_{k=1}^N \mathbf{a}_{m,k} \quad (2)$$

The gravitational acceleration,  ${}^I\mathbf{g} \in \mathbb{R}^3$ , expressed in the IMU frame, can then be calculated as:

$${}^G\mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ -\|{}^I\mathbf{g}\| \end{bmatrix} \quad (3)$$

The initial orientation between the inertial frame and the IMU frame can be established as:

$${}^I\mathbf{g} = \mathbf{C}({}^I\bar{q}) {}^G\mathbf{g} \quad (4)$$

where  $\mathbf{C}(\cdot)$  denotes a rotational matrix. Let  $v = {}^G\mathbf{g} \times {}^I\mathbf{g}$ . Let  $s = \|v\|$  (sine of angle). Let  $c = a \cdot b$  (cosine of angle). Then the rotation matrix  $\mathbf{C}({}^I\bar{q})$  is given by:

$$\mathbf{C}({}^I\bar{q}) = I + [v]_{\times} + [v]_{\times}^2 \frac{1-c}{s^2} \quad (5)$$

### B. Batch IMU Processing

In this module, the imu messages in the imu\_msg\_buffer would be processed. The process model is executed for each imu message. The state information is subsequently updated. This process is repeated until the time bound is reached. The time bound is set by the message time of the current image message, so that all imu messages received before the image message would be processed. The processed imu messages are subsequently removed from the buffer.

## C. Process Model

1) Remove the bias from the measured gyro and acceleration:

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\mathbf{b}}_g \quad (6)$$

$$\hat{\mathbf{a}} = \mathbf{a}_m - \hat{\mathbf{b}}_a \quad (7)$$

2) Compute discrete transition and noise covariance matrix:

$$\mathbf{F} = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}]_{\times} & -\mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ -\mathbf{C}({}^I\bar{q}) & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{C}({}^I\bar{q}) & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (8)$$

Note the  $\mathbf{F}$  is made square as to enable matrix operations.

$$\mathbf{G} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{C}({}^I\bar{q}) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (9)$$

3) Approximate matrix exponential: The discrete state time state transition matrix at time instance  $k$  can be calculated as:

$$\begin{aligned} \Phi_k &= \Phi(t_{k+1}, t_k) = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}(\tau) d\tau\right) \\ &\approx \exp(\mathbf{F}(t_{k+1} - t_k)) = \exp(\mathbf{F}\Delta t) \end{aligned} \quad (10)$$

For matrix exponential  $\exp(\mathbf{X})$ , we have its Taylor expansion as:

$$\exp(\mathbf{X}) = \sum_{k=0}^{\infty} \frac{1}{k!} \mathbf{X}^k \quad (11)$$

where  $X^0$  is defined to identity matrix  $I$  with the same dimensions as  $X$ .

The matrix exponential is then approximated to the 3rd order, which can be considered to be accurate enough assuming  $\Delta t < 0.01s$ :

$$\Phi_k \approx \sum_{k=0}^3 \frac{1}{k!} \mathbf{F}^k \Delta t^k = I + \mathbf{F}\Delta t + \frac{1}{2} \mathbf{F}^2 \Delta t^2 + \frac{1}{6} \mathbf{F}^3 \Delta t^3 \quad (12)$$

4) *Modify the transition matrix:* The transition matrix is modified in the following fashion [2, 5]:

$$\begin{aligned}\Phi_k[0:3, 0:3] &= \mathbf{C}(\mathring{G}\bar{q}_k)\mathbf{C}(\mathring{I}\bar{q}_{k-1}) \\ u &= \mathbf{C}(\mathring{G}\bar{q}_{k-1})^G \mathbf{g} \\ s &= \frac{u^T}{\|u\|}\end{aligned}\quad (13)$$

Then we can update the transition matrix in discrete time as follows:

$$\begin{aligned}\Phi_{k+1}[6:9, 0:3] &= \Phi_k[6:9, 0:3] \\ &- (\Phi_k[6:9, 0:3]u - [\mathring{G}\dot{\hat{\mathbf{v}}}_{k-1} - \mathring{G}\dot{\hat{\mathbf{v}}}_k] \times \mathring{G}\mathbf{g})s\end{aligned}\quad (14)$$

$$\begin{aligned}\Phi_k[12:15, 0:3] &= \Phi_k[12:15, 0:3] \\ &- (\Phi_k[12:15, 0:3]u \\ &- [\Delta t \mathring{G}\dot{\hat{\mathbf{v}}}_{k-1} + \mathring{G}\dot{\hat{\mathbf{p}}}_{I,k-1} - \mathring{G}\dot{\hat{\mathbf{p}}}_{I,k}] \times \mathring{G}\mathbf{g})s\end{aligned}\quad (15)$$

5) *Propagate the state covariance matrix:* We employ the following partitioning for the covariance:

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{II_{k|k}} & \mathbf{P}_{IC_{k|k}} \\ \mathbf{P}_{IC_{k|k}}^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix}\quad (16)$$

where  $\mathbf{P}_{II_{k|k}}$  is the  $21 \times 21$  covariance matrix of the evolving IMU state,  $\mathbf{P}_{CC_{k|k}}$  is the  $6N \times 6N$  covariance matrix of the camera pose estimation, and  $\mathbf{P}_{IC_{k|k}}$  is the correlation between the errors in the IMU state and the camera pose estimates.

The covariance matrix of the propagated state can then be given by:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P}_{II_{k+1|k}} & \Phi_k \mathbf{P}_{IC_{k|k}} \\ \mathbf{P}_{IC_{k+1|k}}^T & \Phi_k^T \mathbf{P}_{CC_{k|k}} \end{bmatrix}\quad (17)$$

where the propagated covariance of the IMU state is:

$$\mathbf{P}_{II_{k+1|k}} = \Phi_k \mathbf{P}_{II_{k|k}} \Phi_k^T + \mathbf{Q}_k\quad (18)$$

The discrete-time noise covariance matrix  $\mathbf{Q}_k$  can be computed as:

$$\begin{aligned}\mathbf{Q}_k &= \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) \mathbf{G} \mathbf{Q} \mathbf{G} \Phi(t_{k+1}, t_k)^T d\tau \\ &\approx \Phi_k \mathbf{G} \mathbf{Q} \mathbf{G} \Phi_k^T \Delta t\end{aligned}\quad (19)$$

6) *Fix the covariance to be symmetric:* The covariance is then fixed to be symmetric:

$$\mathbf{P}_{k+1|k} = \frac{\mathbf{P}_{k+1|k} + \mathbf{P}_{k+1|k}^T}{2}\quad (20)$$

#### D. Predict New State

Refer to Eq. (6) in [2], we have:

$$\mathring{G}\dot{\hat{q}}_t = \frac{1}{2} \Omega(\omega(t)) \mathring{G}\bar{q}_t.\quad (21)$$

According to Section 1.6.1 in [5]:

$$\mathring{G}\bar{q}_{t+\Delta t} = \exp\left(\frac{1}{2} \Omega(\omega) \Delta t\right) \mathring{G}\bar{q}_t\quad (22)$$

$$\begin{aligned}\exp\left(\frac{1}{2} \Omega(\omega) \Delta t\right) &= \left( \cos\left(\frac{\|\omega\|}{2} \Delta t\right) \cdot \mathbf{I}_{4 \times 4} \right. \\ &\quad \left. + \frac{1}{\|\omega\|} \sin\left(\frac{\|\omega\|}{2} \Delta t\right) \right) \cdot \Omega(\omega)\end{aligned}\quad (23)$$

The above expression will cause numerical instability for very small  $\omega$ , due to  $\|\omega\|$  appearing in the denominator. The limit is therefore computed of the above equation as  $\|\omega\|$  goes towards zero, using L'Hôpital's rule.

$$\lim_{\|\omega\| \rightarrow 0} \exp\left(\frac{1}{2} \Omega(\omega) \Delta t\right) = (\mathbf{I} + \frac{\Delta t}{2} \Omega(\omega))\quad (24)$$

The quaternions  $\mathring{G}\bar{q}_{t+\Delta t/2}$  at  $t + \frac{\Delta t}{2}$  and  $\mathring{G}\bar{q}_{t+\Delta t}$  at  $t + \Delta t$  can then be calculated accordingly.

We apply the 4th-order Runge-Kutta method for numerical integration.

$$\begin{aligned}k_{1,v} &= \mathbf{C}(\mathring{G}\bar{q}_{t+\Delta t}) \hat{\mathbf{a}} + \mathring{G}\mathbf{g} \\ k_{1,p} &= \mathring{G}\dot{\hat{\mathbf{v}}}_t \\ k_{2,v} &= \mathbf{C}(\mathring{G}\bar{q}_{t+\frac{\Delta t}{2}}) \hat{\mathbf{a}} + \mathring{G}\mathbf{g} \\ k_{2,p} &= \mathring{G}\dot{\hat{\mathbf{v}}}_t + k_{1,v} \frac{\Delta t}{2} \\ k_{3,v} &= \mathbf{C}(\mathring{G}\bar{q}_{t+\frac{\Delta t}{2}}) \hat{\mathbf{a}} + \mathring{G}\mathbf{g} \\ k_{3,p} &= \mathring{G}\dot{\hat{\mathbf{v}}}_t + k_{2,v} \frac{\Delta t}{2} \\ k_{4,v} &= \mathbf{C}(\mathring{G}\bar{q}_{t+\Delta t}) \hat{\mathbf{a}} + \mathring{G}\mathbf{g} \\ k_{4,p} &= \mathring{G}\dot{\hat{\mathbf{v}}}_t + k_{3,v} \frac{\Delta t}{2}\end{aligned}\quad (25)$$

Putting the slopes together, we have:

$$\mathring{G}\dot{\hat{\mathbf{p}}}_{I,t+\Delta t} \approx \mathring{G}\mathring{G}\dot{\hat{\mathbf{p}}}_{I,t} + \frac{\Delta t}{6} (k_{1,v} + 2k_{2,v} + 2k_{3,v} + k_{4,v})\quad (26)$$

And

$$\mathring{G}\dot{\hat{\mathbf{v}}}_{t+\Delta t} \approx \mathring{G}\dot{\hat{\mathbf{v}}}_t + \frac{\Delta t}{6} (k_{1,p} + 2k_{2,p} + 2k_{3,p} + k_{4,p})\quad (27)$$

#### E. State Augmentation

Based on Equation 16 and Equation 17, when new images are received, the state should be augmented with the new camera state. The pose of the new camera state can be computed from the latest IMU state:

$$\begin{aligned}\mathring{C}\hat{q} &= \mathring{C}\hat{q} \otimes \mathring{I}\hat{q} \\ \mathring{C}\hat{p}_c &= \mathring{G}\hat{p}_c + C(\mathring{I}\hat{q})^T \mathring{I}\hat{p}_c\end{aligned}\quad (28)$$

where  $\mathring{C}\hat{q}$  is the quaternion of the new camera with respect to the world frame,  $\mathring{I}\hat{q}$  is the quaternion of the new camera with respect to the IMU frame,  $\mathring{I}\hat{q}$  is the quaternion of the IMU with respect to the world frame,  $\mathring{C}\hat{p}_c$  is the position of the new camera with respect to the world frame,  $\mathring{G}\hat{p}_c$  is the position of the IMU with respect to the world frame and  $\mathring{I}\hat{p}_c$  is the position of the new camera with respect to the IMU frame.

Therefore, as shown in the reference [3], the augmented covariance matrix is:

## II. RESULTS

The estimated and ground truth trajectories are both plotted in x-y plane and 3d space, as shown in Figure 1. We can see that they tightly match each other, indicating an accurate estimation.

This observation is further backed by qualitative analysis, which is summarized in Table I. We can see that our estimated trajectory has small RMSE, and mean and median errors. The std is also small, which suggests consistent tracking performance.

TABLE I: Qualitative analysis of the estimated trajectory with respect to Vicon.

RMSE	Mean	Median	Std	Min	Max
9.5cm	8.6cm	8.1cm	4.2cm	0.4cm	27cm

$$\begin{aligned} \mathbf{P}_{I_{k|k}} &= \begin{bmatrix} \mathbf{I}_{21+6N} \\ J \end{bmatrix} \mathbf{P}_{k|k} \begin{bmatrix} \mathbf{I}_{21+6N} \\ J \end{bmatrix}^T \\ J &= \begin{bmatrix} \mathbf{J}_I & \mathbf{0}_{6 \times 6N} \end{bmatrix} \\ \mathbf{J}_I &= \begin{bmatrix} C(\hat{g}) & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \\ -C(\hat{g})^T [\hat{p}_c]_{\times} & \mathbf{0}_{3 \times 9} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix} \end{aligned} \quad (29)$$

where  $N$  is the number of the camera states.

Eventually, the covariance is fixed to be symmetric:

$$\mathbf{P}_{k+1|k} = \frac{\mathbf{P}_{I_{k+1|k}} + \mathbf{P}_{I_{k+1|k}}^T}{2} \quad (30)$$

### F. Add Feature Observations

We would like to add all features in `feature_msg` to `self.map_server`.

If the feature does exist in the `self.map_server` before, then, we will directly store it into the `self.map_server`. Otherwise, we will store the feature into the observation set of `self.map_server`. The features in the observation set mean the features are tracked, therefore, we can calculate out the tracking rate correspondingly.

### G. Measurement Update

As described in the papaer [3], we can obtain the residual of the measurement is defined as :

$$\mathbf{r}_o^j = \mathbf{H}_{X,o}^j \tilde{\mathbf{x}} + \mathbf{n}^j \quad (31)$$

Therefore, we can update the extend Kalman filter[2] as following:

Firstly, we will employ the QR decomposition of the matrix  $H_x$ :

$$\begin{aligned} r_o &= [Q_1 \quad Q_2] \begin{bmatrix} T_H \\ 0 \end{bmatrix} \tilde{\mathbf{x}} + n_o \\ \begin{bmatrix} Q_1^T r_o \\ Q_2^T r_o \end{bmatrix} &= \begin{bmatrix} T_H \\ 0 \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} Q_1^T n_o \\ Q_2^T n_o \end{bmatrix} \end{aligned} \quad (32)$$

Then we can derive that:

$$\begin{aligned} r_n &= Q_1^T r_o \\ &= T_H \tilde{\mathbf{x}} + n_n \\ &= T_H \tilde{\mathbf{x}} + Q_1^T n_o \end{aligned} \quad (33)$$

Furthermore, we can compute Kalman gain:

$$K = P T_H^T (T_H P T_H^T + R_n)^{-1} \quad (34)$$

where  $P$  is the transition matrix,  $R_n$  is the observation noise. then, the state error can be calculated via:

$$\delta x = K r_n \quad (35)$$

Eventually, the state covariance matrix can be updated through:

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K} \mathbf{T}_H) \mathbf{P}_{k+1|k} (\mathbf{I} - \mathbf{K} \mathbf{T}_H)^T + \mathbf{K} \mathbf{R}_n \mathbf{K}^T \quad (36)$$

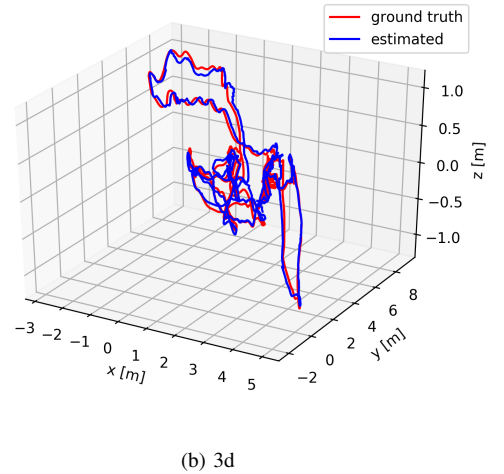
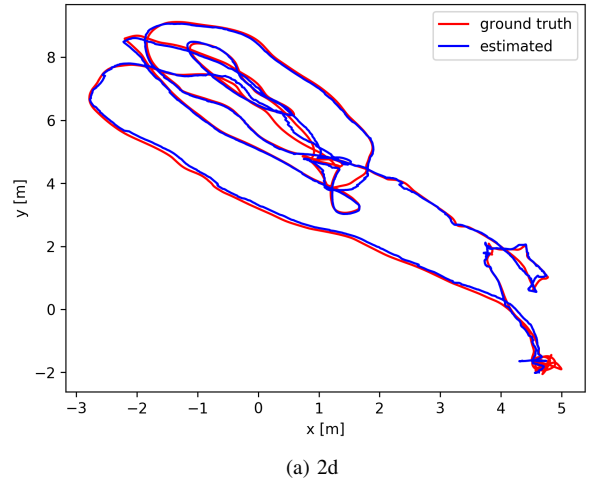


Fig. 1: Estimated(in blue) and ground truth(in red) trajectories in x-y plane (left) and 3d space(right).

## REFERENCES

- [1] Nitin J. Sanket and Lening Li. *P4 Guidance*. URL: <https://rbe549.github.io/fall2022/proj/p4/>.
- [2] Anastasios I Mourikis, Stergios I Roumeliotis, et al. "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation." In: *ICRA*. Vol. 2. 2007, p. 6.
- [3] Ke Sun et al. "Robust stereo visual inertial odometry for fast autonomous flight". In: *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 965–972.
- [4] Kartik Mohta et al. *VIO GitHub Repository*. URL: [https://github.com/KumarRobotics/msckf\\_vio](https://github.com/KumarRobotics/msckf_vio).
- [5] Nikolas Trawny and Stergios I Roumeliotis. "Indirect Kalman filter for 3D attitude estimation". In: *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep 2* (2005), p. 2005.