Attitude Estimation Using Inertial Measurement Unit and Magnetometer readings

Mohammad Hussein Yoosefian Nooshabadi

1 Introduction

Attitude, or *orientation*, refers to the angular position of an object—together with its attached frame, called the *body frame*—relative to a reference frame, called the *global frame* (see Figure 1). Estimating attitude is a fundamental task across many domains, from robotics to sports analytics [1]. For instance, in the autonomous navigation of a quadrotor, the first requirement is to stabilize the vehicle, which in turn demands precise knowledge of its attitude [2]. Other applications include human motion analysis [3] and virtual reality [4].

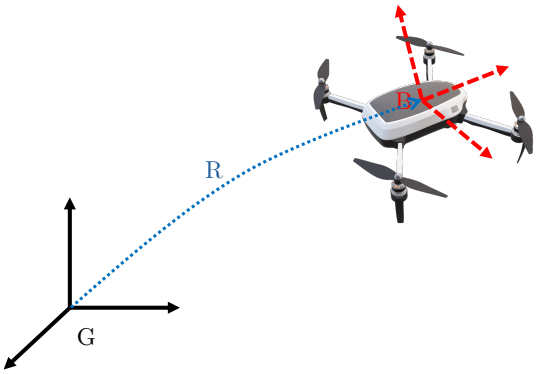


Figure 1: Attitude representation for a quadrotor. Attitude is the angular position of a body frame B relative to a global (or reference) frame G. The goal in this project is to determine the attitude using readings from an IMU and a magnetometer.

Among the various sensing modalities available for attitude estimation, the most widely used is the *Inertial Measurement Unit* (IMU). The core components of an IMU are gyroscopes, which measure the three-dimensional angular velocity of the body, and accelerometers, which measure the three-dimensional specific force acting on the body. In many configurations, IMUs are also equipped with magnetometers, which measure the direction of the local magnetic field vector and serve as an additional reference for heading estimation.

The problem of attitude estimation using IMU and magnetometer sensors has been studied for decades. Early approaches include Davenport's q-method [5] and the QUEST algorithm [6], which used quaternion algebra to compute the optimal rotation efficiently. The work in [7] introduced a quaternion-based Kalman filter for spacecraft attitude, laying the foundation for modern attitude filtering using inertial sensors.

By the early 2000s, various Kalman-based attitude estimation methods have been developed, including extended Kalman filter (EKF) [8] and unscented Kalman filter (UKF) [9]. In practice, EKF/UKF performance depends on proper tuning of process and measurement noise covariances, as a result, adaptive schemes have been proposed to adjust these gains online [10]. A comparison between EKF and UKF and an implementation guide for them can be found in [11] and [12], respectively.

In the last decade, significant attention has been paid to filter consistency and exploiting the geometric structure of the attitude estimation problem. One notable development is the Invariant Extended Kalman Filter (IEKF) [13], which leverages the Lie Group structure of the attitude state (SO(3)). In essence, it performs the EKF prediction and update in a manner independent of the current estimate, by formulating errors on the Lie algebra. This yields more reliable linearization because the filter's Jacobians do not change with the estimated orientation.

In this project, we design and implement four Bayesian filters for attitude estimation

- EKF
- UKF
- Error State EKF (ESEKF)
- Enhanced EKF (EEKF).

We thoroughly analyze these algorithms for two different scenarios, in which magnetometer has different noise distributions

- Gaussian noise
- Student's-t noise.

We also show the effectiveness of the algorithms by testing them on a real-world dataset, *EuRoC mav dataset* [14]. For more information please see Section 4.

2 Problem Formulation

In this project, we use two different problem formulations, which are briefly discussed in the following.

2.1 First formulation: quaternions

Using quaternions in the state vector is the most common method for attitude estimation. We will implement EKF, UKF, and EEKF using this problem formulation.

State vector. In this project, we focus on attitude estimation using IMU and magnetometer measurements. The state to be estimated is

$$x = \begin{bmatrix} q^\top & b_g^\top & b_a^\top \end{bmatrix}^\top \in \mathbb{R}^{10}, \tag{1}$$

where $q = \begin{bmatrix} q_w & q_v^\top \end{bmatrix}^\top \in \mathbb{R}^4$ is the unit quaternion representing the attitude, $b_g, b_a \in \mathbb{R}^3$ are gyroscope and accelerometer bias vectors respectively.

State transition function. We use the following equations

$$\begin{aligned} q_k &= q_{k-1} \otimes \exp_q\left(\frac{\Delta t}{2}(\omega_{m,k-1} - b_{g,k-1} - n_g)\right) \\ b_{g,k} &= b_{g,k-1} + n_{b_g} \\ b_{a,k} &= b_{a,k-1} + n_{b_a}, \end{aligned} \tag{2}$$

where $n_g \sim \mathcal{N}(0, \sigma_g^2 I_3)$ is the gyroscope noise, $n_{b_g} \sim \mathcal{N}(0, \sigma_{b_g}^2 I_3)$ and $n_{b_a} \sim \mathcal{N}(0, \sigma_{b_a}^2 I_3)$ are gyroscope and accelerometer random walk (bias noise) terms. So, the final noise vector is $n_f = \begin{bmatrix} n_g^\top & n_{b_g}^\top & n_{b_a}^\top \end{bmatrix}^\top \in \mathbb{R}^9$, which is a zero mean Gaussian with covariance $Q = \text{blkdiag}(\sigma_g^2 I_3\,,\, \sigma_{b_g}^2 I_3\,,\, \sigma_{b_a}^2 I_3)$. Moreover, $\omega_{m,k-1} \in \mathbb{R}^3$ is the gyroscope reading at time step $k-1$ and Δt is the gyroscope sampling time. Further, we define

$$\begin{aligned} \exp_q(\eta) &= \begin{bmatrix} \cos(\|\eta\|) \\ \frac{\eta}{\|\eta\|}\sin(\|\eta\|) \end{bmatrix} \\ p \otimes q &= \begin{bmatrix} p_w q_w - p_v \cdot q_v \\ p_w q_v + q_w p_v + p_v \times q_v \end{bmatrix}. \end{aligned} \tag{3}$$

Measurement function. For the measurement, we use the accelerometer and magnetometer readings

$$y_k = \begin{bmatrix} -R_k^\top g^G + b_{a_k} \\ R_k^\top m^G \end{bmatrix} + \begin{bmatrix} n_a \\ n_t \end{bmatrix}, \tag{4}$$

where $n_a \sim \mathcal{N}(0, \sigma_a^2 I_3)$ and $n_t \sim \mathcal{N}(0, \sigma_t^2 I_3)$ are accelerometer and magnetometer noise. So, the final noise vector is $n_h = \begin{bmatrix} n_a^\top & n_t^\top \end{bmatrix}^\top \in \mathbb{R}^6$, which is a zero mean Gaussian with covariance $R = \text{blkdiag}(\sigma_a^2 I_3 \,,\, \sigma_t^2 I_3)$. Also, $g^G \in \mathbb{R}^3$ and $m^G \in \mathbb{R}^3$ are the local gravity vector and the Earth's magnetic field, respectively, both expressed in the global frame. Further, $R_k := R(q_k)$ is the rotation matrix corresponding to the quaternion, which is defined as

$$R(q) = \begin{bmatrix} 2q_w^2 + 2q_x^2 - 1 & 2q_x q_y - 2q_w q_z & 2q_x q_z + 2q_w q_y \\ 2q_x q_y + 2q_w q_z & 2q_w^2 + 2q_y^2 - 1 & 2q_y q_z - 2q_w q_x \\ 2q_x q_z - 2q_w q_y & 2q_y q_z + 2q_w q_x & 2q_w^2 + 2q_z^2 - 1 \end{bmatrix}, \tag{5}$$

with $q = \begin{bmatrix} q_w & q_v^\top \end{bmatrix}^\top$ and $q_v = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^\top$. For more information and derivations please see [15].

2.2 Second formulation: error state

It turns out that computing the Posterior Cramer Rao Bound (PCRB) for the formulation given in Section 2.1 is not possible. The reason is that the first component of the process noise, i.e., the gyroscope noise, is non-additive. Since $n_g \in \mathbb{R}^3$ and $q_k \in \mathbb{R}^4$ (i.e., the noise lies on a lower-dimensional manifold than the state), the conditional probability $p(q_{k+1} \mid q_k)$ is singular. Nevertheless, researchers proposed to linearize the state transition function, so that we have an additive noise [15]. For a general system $x_{k+1} = f(x_k, n_k)$ with $n \sim \mathcal{N}(0, Q)$ we can use first-order Taylor approximation to write

$$x_{k+1} = f(x_k, n_k) \approx f(x_k) + \tilde{n}_k, \qquad \tilde{n} \sim \mathcal{N}(0, \tilde{Q}), \tag{6}$$

where

$$\tilde{Q} = G(x_k) Q G(x_k)^\top, \qquad G(x_k) = \left.\frac{\partial f}{\partial n_k}\right|_{n_k = 0}.$$

As a result, for the state transition function (2) we have

$$\begin{aligned} q_{k+1} &= q_k \otimes \exp_q\left(\frac{\Delta t}{2}(\omega_{m,k} - b_{g,k})\right) + \tilde{n}_g \\ b_{g,k+1} &= b_{g,k} + n_{b_g} \\ b_{a,k+1} &= b_{a,k} + n_{b_a}. \end{aligned} \tag{7}$$

So, $G(x_k)$ in this case will be a block matrix of the form

$$G(x_k) = \text{blkdiag}(G_{11}(x_k) \,,\, I_3 \,,\, I_3),$$

where $G_{11}(x_k)$ can be computed using (2). Using (7) we have

$$\ln p(x_{k+1} \mid x_k) = -\frac{1}{2} \left(x_{k+1} - f(x_k)\right)^\top \left(G Q G^\top\right)^{-1} \left(x_{k+1} - f(x_k)\right). \tag{8}$$

Since $\text{rank}(Q) = 9$ but $G \in \mathbb{R}^{10 \times 9}$, we see that $GQG^\top$ is not full-rank, hence not invertible. For this reason, researchers proposed to define and use *error state*. The idea is to write the state variables as 'nominal plus error'. In other words,

$$x = \bar{x} \oplus \delta x, \tag{9}$$

where $\oplus$ is the general addition, $\bar{x}$ is the nominal state and δx is the error. In our case, we have

$$\begin{aligned} q &= \bar{q} \otimes \delta q \\ b_g &= \bar{b}_g + \delta b_g \\ b_a &= \bar{b}_a + \delta b_a. \end{aligned} \tag{10}$$

Now, to have a minimal representation of the attitude, we define $\delta\theta = \log_q(\delta q)$, where $\delta\theta$ is the error from the nominal attitude and the mapping $\log_q(\cdot)$ is the inverse of $\exp_q(\cdot)$ and is given by

$$\log_q(q) = 2\arccos q_w \frac{q_v}{\|q_v\|}. \tag{11}$$

With these new variables, we define the error state vector as

$$\delta x = \begin{bmatrix} \delta\theta \\ \delta b_g \\ \delta b_a \end{bmatrix} \in \mathbb{R}^9. \tag{12}$$

Differentiating (10) and discretizing, after some manipulations we obtain the linearized model for the state (12)

$$\delta x_{k+1} = F_k \delta x_k + G n_k, \tag{13}$$

with

$$F_k = \begin{bmatrix} I_3 - [\bar{\omega}_k]_\times \Delta t & -I_3\Delta t & 0_3 \\ 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & I_3 \end{bmatrix}, \quad G = \text{blkdiag}(-I_3, I_3, I_3), \tag{14}$$

and $\bar{\omega} = \omega_{m,k} - b_{g,k}$ and $n_k \sim \mathcal{N}(0, \Delta t^2 Q)$. We see that the G matrix in this linearized model not only does not depend on the state, but it does not make the effective covariance, i.e., $GQG^\top$ singular.

For the measurement equation, we use the following approximation

$$R(q) \approx R(\bar{q})(I_3 - [\delta\theta]_\times). \tag{15}$$

Substituting (15) into (4) we can compute the Jacobian matrix H. After some manipulation, we get

$$H_{k+1} = \begin{bmatrix} \left[-R(\bar{q}_{k+1|k})^\top g^G\right]_\times & 0_3 & I_3 \\ \left[R(\bar{q}_{k+1|k})^\top m^G\right]_\times & 0_3 & 0_3 \end{bmatrix}. \tag{16}$$

For more information, please refer to [8].

3 Filter Design and Implementation

As mentioned, in this project we make use of four filters. Since EKF, UKF, and ESEKF are well-known, we do not discuss their derivation and formulation here. For a derivation and comparison of EKF and UKF please see [11]. For a comprehensive review on ESEKF see [8]. In this section, we explain the idea behind the Enhanced EKF (EEKF). An online performance metric for filters is the average normalized innovation squared (ANIS) which should be normally distributed

$$\bar{\epsilon}_k^2 = \frac{1}{k}\sum_{i=1}^{k} \epsilon_i^2 \sim \mathcal{N}(n_y, \frac{2n_y}{k}) \tag{17}$$

with n_y being the measurement dimension and

$$\epsilon_i^2 = z_i^\top (H_i P_{i|i-1} H_i^\top + R)^{-1} z_i \tag{18}$$

where z_i is the innovation of the filter. The idea is to enforce this distribution by adaptively changing the measurement covariance matrix R. Assuming 95% confidence interval, we can impose the constraint

$$n_y - 2\sqrt{\frac{2n_y}{k}} \leq \bar{\epsilon}_k^2 \leq n_y + 2\sqrt{\frac{2n_y}{k}} \tag{19}$$

Using the following recursion for ANIS

$$\bar{\epsilon}_k^2 = \frac{1}{k}\left((k-1)\bar{\epsilon}_{k-1}^2 + \epsilon_k\right) \tag{20}$$

we get

$$\underbrace{k\left(n_y - 2\sqrt{\frac{2n_y}{k}}\right) - (k-1)\bar{\epsilon}_{k-1}^2}_{\bar{\tau}_l} \leq \epsilon_k^2 \leq \underbrace{k\left(n_y + 2\sqrt{\frac{2n_y}{k}}\right) - (k-1)\bar{\epsilon}_{k-1}^2}_{\bar{\tau}_u} \tag{21}$$

Defining X as the new measurement noise covariance and using (18) we get

$$\bar{\tau}_l \leq z_k^\top (H_k P_{k|k-1} H_k^\top + X)^{-1} z_k \leq \bar{\tau}_u. \tag{22}$$

It is easy to see that the lower bound constraint is non-convex. For the upper bound we can use the Schur-complement to write

$$\begin{bmatrix} \Sigma & z_k \\ z_k^\top & \bar{\tau}_u \end{bmatrix} \succeq 0 \quad \Leftrightarrow \quad z_k^\top (H_k P_{k|k-1} H_k^\top + X)^{-1} z_k \leq \bar{\tau}_u \tag{23}$$

with $\Sigma = H_k P_{k|k-1} H_k^\top + X$. So, using off-the-shelf semidefinite program solvers like MOSEK, we can solve the following feasibility problem

$$\begin{aligned} &\begin{bmatrix} \Sigma & z_k \\ z_k^\top & \bar{\tau}_u \end{bmatrix} \succeq 0 \\ &X \succ 0 \\ &\Sigma = H_k P_{k|k-1} H_k^\top + X \end{aligned} \tag{24}$$

for X at each iteration (in case the bounds (22) are violated) and use X in place of R in the EKF pipeline. A pseudo code for the algorithm is given in Algorithm 1.

There are several limitations to our approach in EEKF. First, due to its non-convexity, we simply discarded the lower bound in (22). This means that this method only works if the upper-bound is violated. Second, instead of solving a feasibility problem, we can introduce an objective function that among all possible choices of X chooses the one that minimizes e.g., the distance between the mean of the ANIS and n_y. Lastly, solving Semidefinite programs can be computationally intensive, so the method might be not applicable in resource-constrained applications.

4 Performance Analysis

In this section we present and analyze the results of implementing the filters on synthetic data and real-world datasets. Since the state vector is large, we do not have enough space to present all of the results, so we only show one plot per axis.

4.1 Synthetic data

In this section, we consider two scenarios.

4.1.1 Gaussian measurement noise

This is the nominal scenario. In this case, both process and measurement noise are assumed to be Gaussian. Simulation results are shown in Figure 2. In Figure 2a we only have the RMSE plot for ESEKF, since as we mentioned, PCRB can only be computed for the error state. We observe that the RMSE converges to the std, meaning that the estimator has a very small bias. In Figure 2b we observe that the EEKF has a similar performance to UKF and they both have smaller RMSE than EKF and ESEKF. Figure 2c shows the filters efficiency, and again, we see that EEKF is the most efficient filter. Figures 2d to 2f show the RMSE of the attitude, yaw, pitch, and roll, respectively. Again, one can observe that EEKF outperforms the other three filters. Lastly, in Figures 2g to 2j the ANIS is plotted. In these figures, the black solid line is the ideal mean of ANIS and the black dashed curves represent the 95% confidence interval for the ideal normal distribution. The colored lines and shaded areas respectively represent the mean and 95% confidence interval

Algorithm 1 Enhanced Extended Kalman Filter

Initialization:
$\hat{x}_0 \leftarrow$ initial estimate of the state
$P_{0|0} \leftarrow$ initial covariance of the filter
The state and measurement equations:

$$x_{k+1} = f(x_k, u_{k+1}, w_k)$$
$$y_{k+1} = h(x_{k+1}, u_{k+1}) + v_{k+1}$$

procedure PREDICTIONSTEP($\hat{x}_k$, P_k)
    1. State prediction
    $\hat{x}_{k+1|k} \leftarrow f(\hat{x}_k, u_{k+1}, 0)$
    2. Covariance prediction
    $P_{k+1|k} \leftarrow F_k P_k F_k^\top + GQG^\top$ ▷ with $F = \frac{\partial f}{\partial x_k}\,|_{\hat{x}_k}$, $G = \frac{\partial f}{\partial w_k}\,|_{\hat{x}_k}$
end procedure

procedure UPDATESTEP($\hat{x}_{k+1|k}$, $P_{k+1|k}$, y_{k+1})
    1. Consistency check
    $S \leftarrow H_{k+1} P_{k+1|k} H_{k+1}^\top + R$ ▷ with $H = \frac{\partial h}{\partial x_k}\,|_{\hat{x}_{k|k+1}}$
    $z_k \leftarrow y_{k+1} - h(\hat{x}_{k+1|k}, u_{k+1})$
    if $z_k^\top S^{-1} z_k > \bar{\tau}_u$ **then** ▷ With $\bar{\tau}_u$ given in (22)
        Solve (24) and then $R \leftarrow X$
        $S \leftarrow H_{k+1} P_{k+1|k} H_{k+1}^\top + R$
    end if
    $K \leftarrow P_{k+1|k} H_{k+1}^\top S^{-1}$
    2. State update
    $\hat{x}_{k+1} = \hat{x}_{k+1|k} + K z_k$
    3. Covariance update
    $P_{k+1} \leftarrow (I - K H_{k+1}) P_{k+1|k}$
end procedure

Table 1: Average computation time per simulation iteration for various filters

Filtering method	EKF	ESEKF	UKF	EEKF
CPU time (ms)	0.024	0.043	0.35	0.11

of the empirical distribution of ANIS. As we can see, EEKF and UKF have the best performance. Comparing EKF and EEKF one can easily see the improvement. Computation time per iteration for various filters is reported in Table 1. We see that EKF has the least computation time and UKF is the most computationally expensive algorithm.

4.1.2 non-Gaussian measurement noise

Recently, it was observed that magnetometer noise tends to be heavy-tailed and its characteristic are captured more accurately via Student-t distribution [16]. As a result, for the second scenario, we modeled $n_t \sim \mathcal{S}(\nu_t = 3)$, where $\mathcal{S}(\nu)$ denotes a student-t distribution with parameter ν. To implement the filters, the student-t noise is approximated by a Gaussian via moment matching.

The results are shown in Figure 3. We see that, as expected, generally the filters' performance has degraded. Specifically, there is a gap between RMSE and std in Figure 3a, meaning that the filter has a large bias. The trend for other figures is similar to those of the Gaussian scenario. It is worth noting that the mean of the ANIS for EEKF shown in Figure 3i is smaller than the ideal mean n_y. This is because in designing EEKF we discarded the lower bound in (22). In other words, the EEKF does not correct the ANIS if it is less than its lower bound.

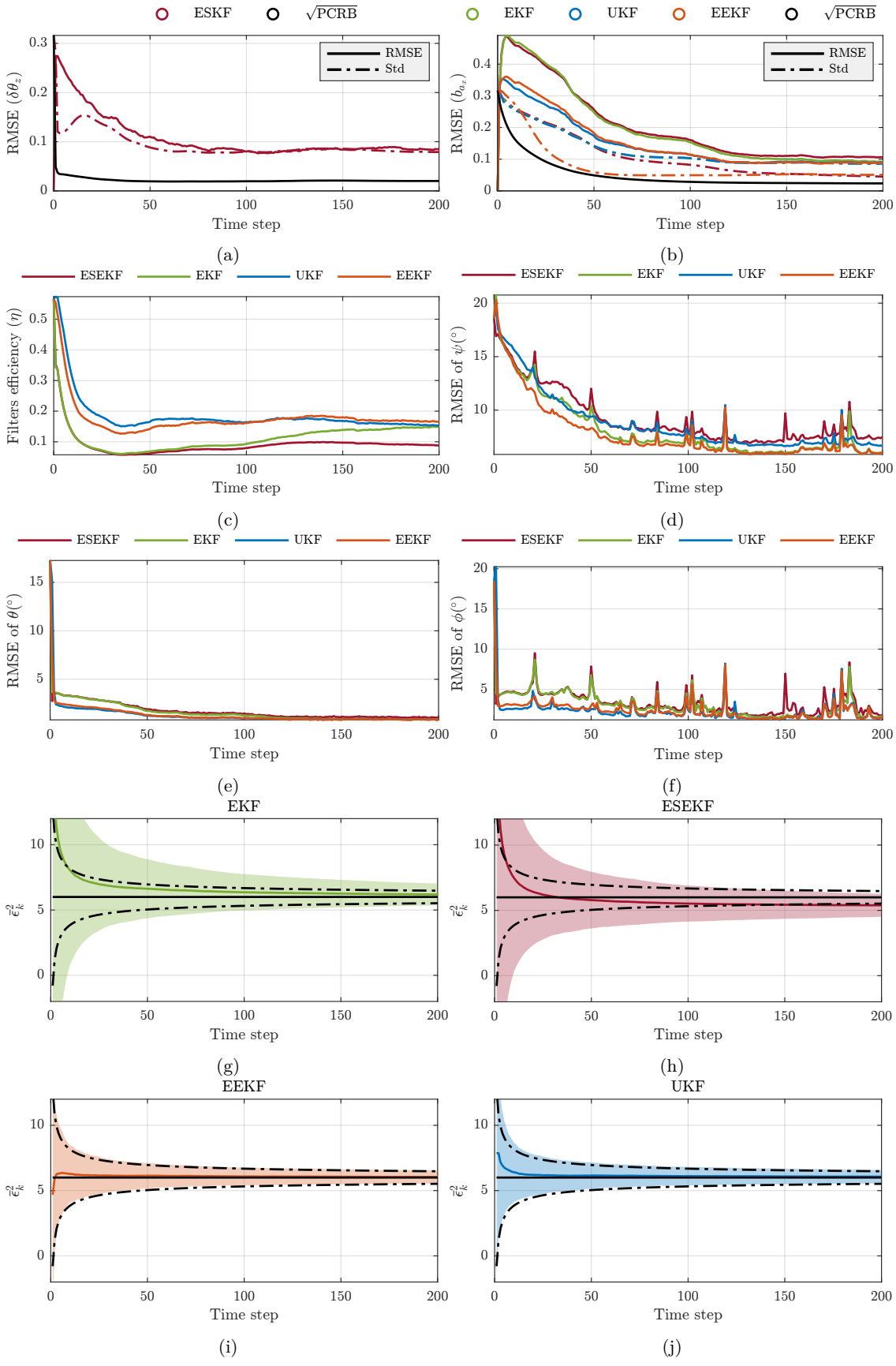


Figure 2: Results for the case with Gaussian measurement noise.

Table 2: Average computation time per simulation iteration for various filters implemented on EuRoC dataset

Filtering method	EKF	UKF	EEKF
CPU time (ms)	0.060	0.46	94.9

Another interesting observation is that in the non-Gaussian case, the RMSE of the yaw angle is more degraded compared to the other two angles. The reason is that the magnetometer is the main sensor that estimates the yaw angle. In other words, using just accelerometer and a gyroscope, one cannot estimate the yaw angle. Further discussion is provided in Section 4.2.

4.2 EuRoC mav dataset

The EuRoC MAV dataset [14] provides synchronized IMU, stereo camera, and ground-truth motion-capture data collected from a micro aerial vehicle flying in indoor environments. However, the drawback is that it lacks magnetometer readings. It can be proved that when using just an IMU the yaw angle becomes unobservable. Nevertheless, the algorithms are implemented on the dataset and the results are shown in Figure 4. We see from Figure 4a that the filters perform poorly in estimating ψ. As we discussed, this is due to the fact that this angle is not observable when we only use IMU. We also see a chattering behavior in the RMSE of θ and ϕ. The reason for this behavior is the violation of our assumption. When modeling the accelerometer, we ignored the acceleration of the body, as we assumed it is negligible compared to the gravity. However, since the drone has abrupt movements in some parts of its trajectory, this assumption is violated. Also, this violation of assumption caused inconsistency for the two filters EKF and UKF. However, EEKF, as expected, is consistent.

To prove that the acceleration assumption is violated, we did an experiment in which we removed the body acceleration from the data. The results are shown in Figure 5. We no longer see the chattering in RMSE plots. Also, the filters are now consistent. Moreover, the results of the EEKF and EKF are identical, since the EKF is consistent.

The computation time for filters is given in Table 2. We see that EEKF is more expensive due to the fact that it needs to correct the highly inconsistent EKF at each iteration by solving a semidefinite program.

5 Conclusions and Outlook

In this project, we considered the problem of attitude estimation using readings from accelerometer, gyroscope, and magnetometer. We used three common filtering approaches, namely, EKF, UKF, and ESEKF. Also, using consistency constraints we modifed the EKF and named the new filter EEKF. We tested these four algorithms on synthetic data and real-world datasets. One key observation was the superior performance of the EEKF compared to other methods. Also, we observed that the assumption that acceleration of the body frame is negligible compared to the gravity is critical for the performance of the algorithms.

There are several directions for future work. One immediate extension to EEKF is to use a better convex relaxation for the lower bound in (22) than simply discarding it. One can also solve an optimization problem, instead of a feasibility problem, to minimize an objective function. This might result in a better performance of EEKF.

Lastly, from a practical point of view, when attitude estimation is needed, it is better to perform a full pose estimation, position and orientation. In this case, we do consider the acceleration of the body frame, hence the attitude estimation is more accurate.

References

[1] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of guidance, control, and dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

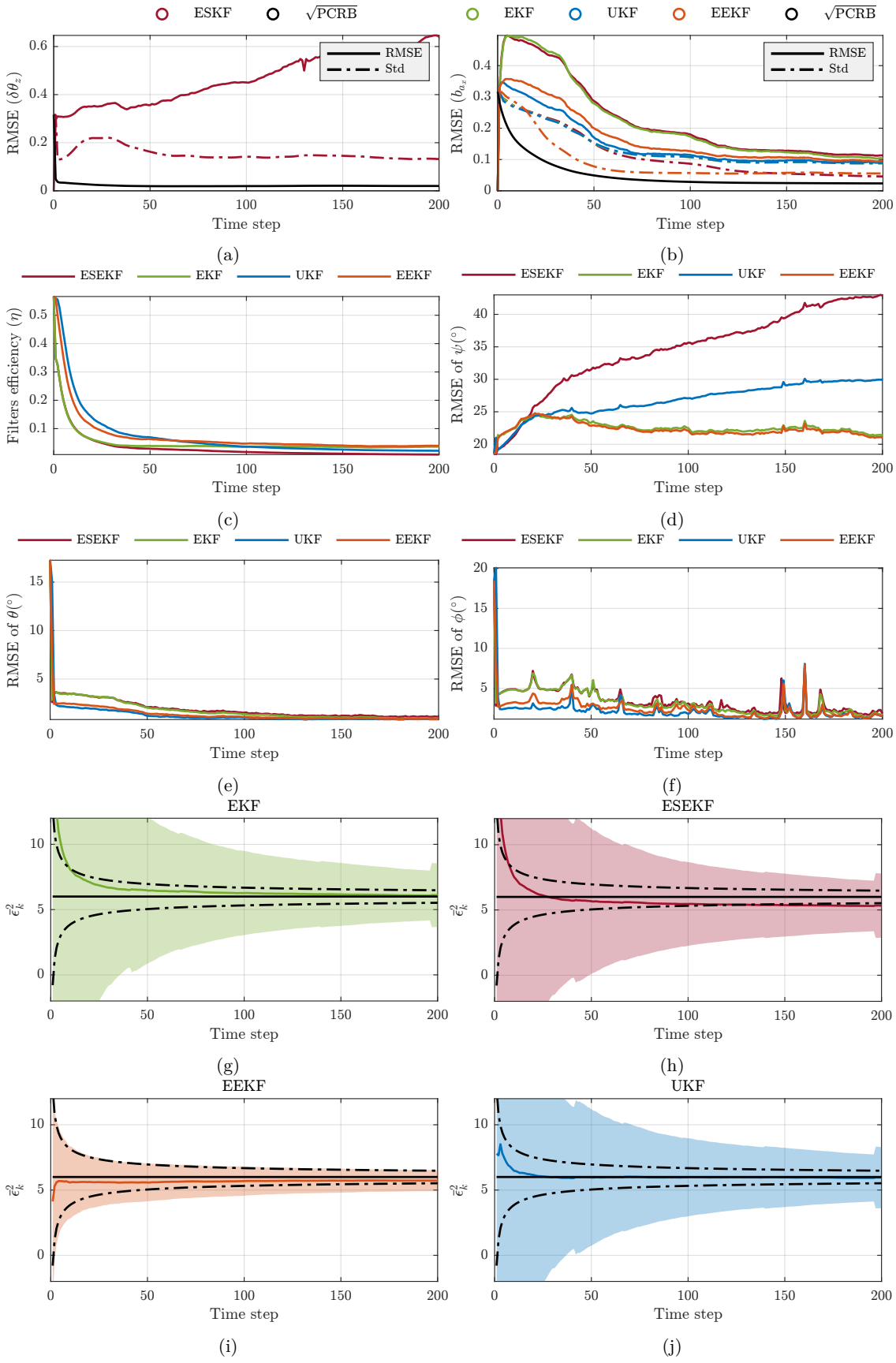


Figure 3: Results for the case with Student's-t magnetometer noise.

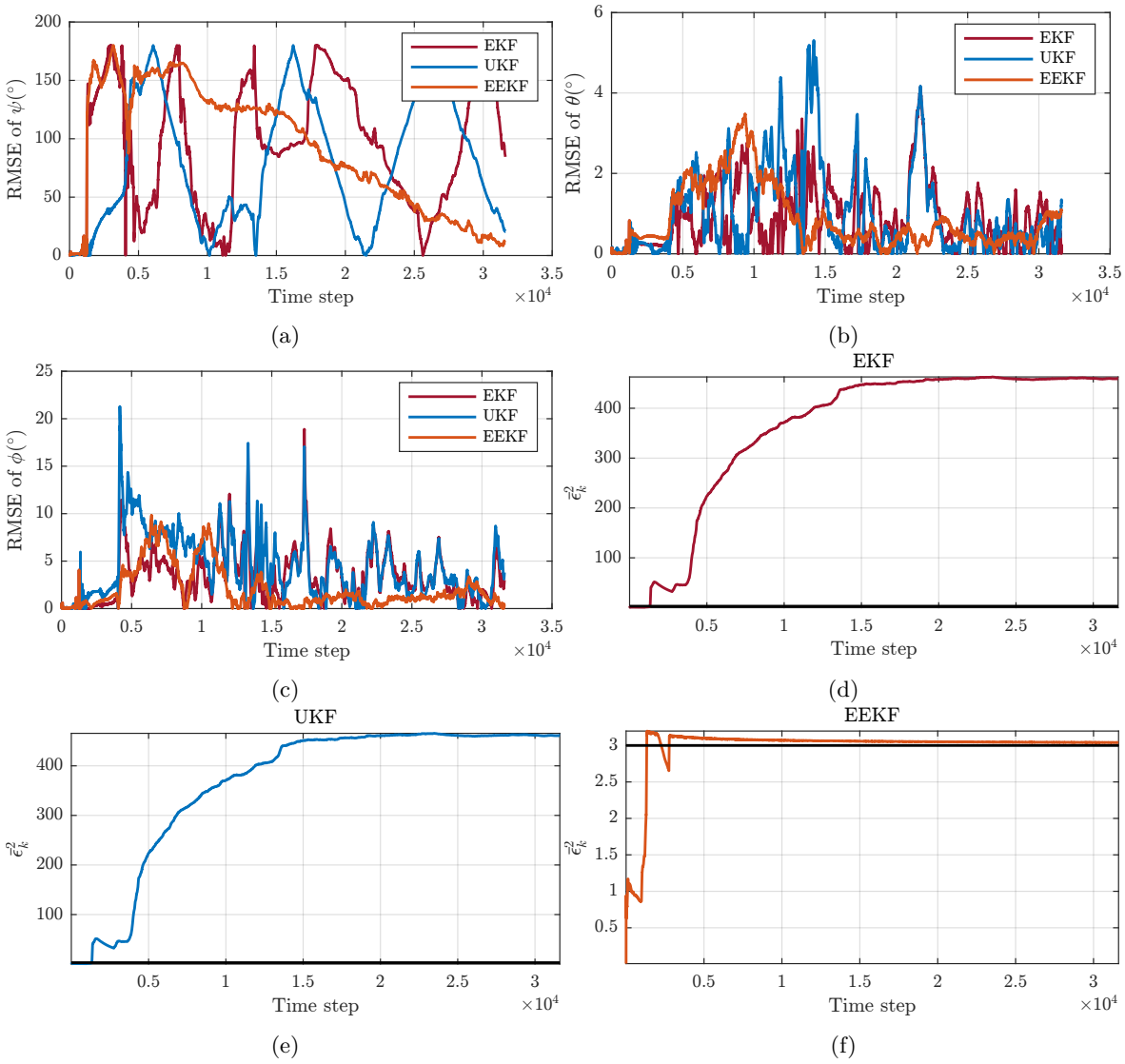


Figure 4: Implementation results for EuRoC mav dataset.

[2] F. Hoffmann, N. Goddemeier, and T. Bertram, "Attitude estimation and control of a quadrocopter," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 1072–1077.

[3] J. Li, K. Liu, and J. Wu, "Ego-body pose estimation via ego-head pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 142–17 151.

[4] K. Wu, H. Zhang, and Y. Mao, "Vision-inertia virtual glasses relative attitude estimate within a rotation free cabin," 2018.

[5] P. B. Davenport, *A vector approach to the algebra of rotations with applications*. National Aeronautics and Space Administration, 1968, vol. 4696.

[6] M. D. Shuster and S. D. Oh, "Three-axis attitude determination from vector observations," *Journal of guidance and Control*, vol. 4, no. 1, pp. 70–77, 1981.

[7] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance, control, and Dynamics*, vol. 5, no. 5, pp. 417–429, 1982.

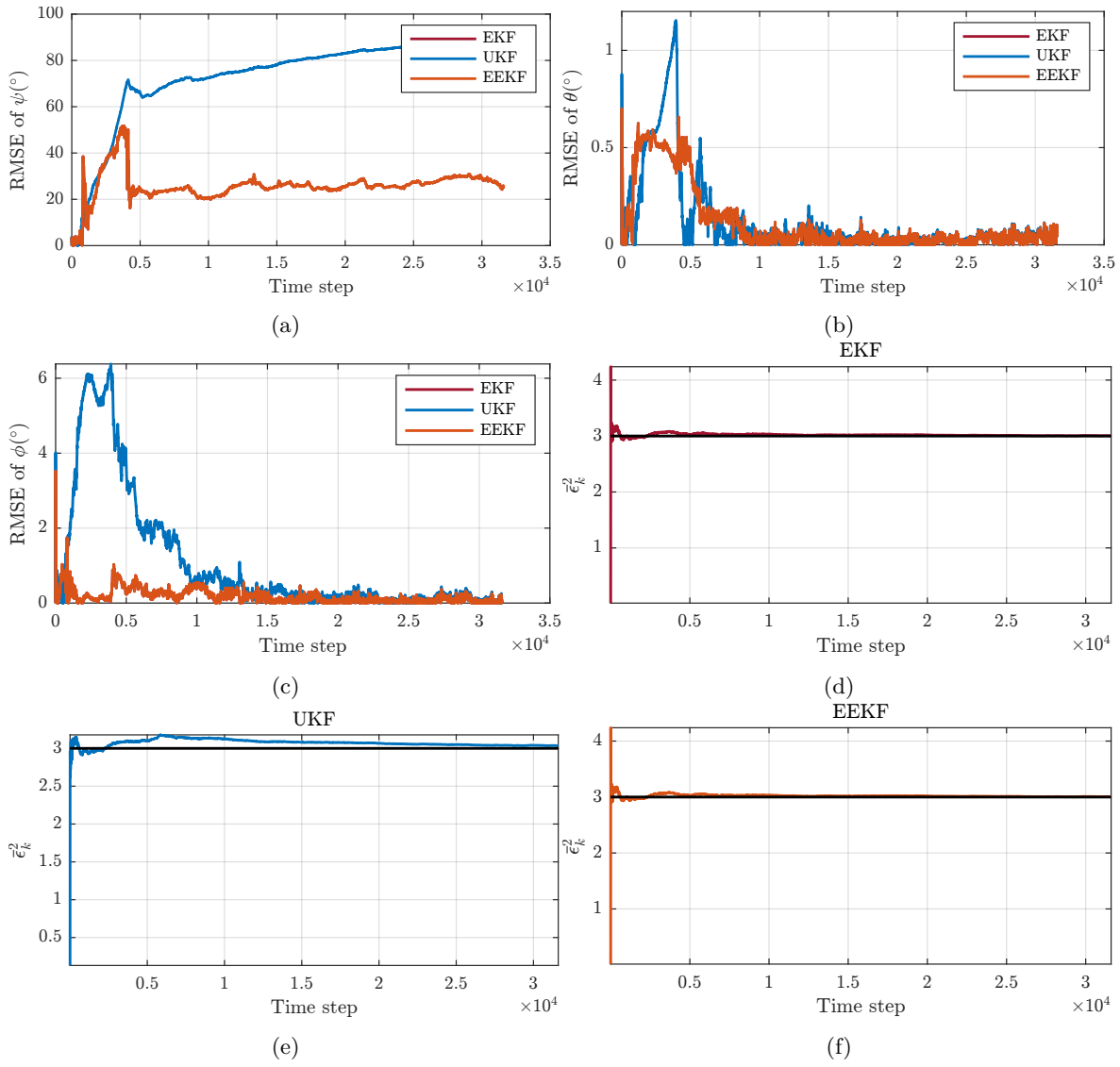


Figure 5: Implementation results for EuRoC mav dataset with corrected acceleration.

[8] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[9] H. G. De Marina, F. J. Pereda, J. M. Giron-Sierra, and F. Espinosa, "Uav attitude estimation using unscented kalman filter and triad," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4465–4474, 2011.

[10] A. C. Chiella, B. O. Teixeira, and G. A. Pereira, "Robust attitude estimation using an adaptive unscented kalman filter," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7748–7754.

[11] M. Rhudy and Y. Gu, "Understanding nonlinear kalman filters part 1: Selection of ekf or ukf," *Interactive Robotics Letters*, vol. 208, 2013.

[12] ——, "Understanding nonlinear kalman filters, part ii: an implementation guide," *Interactive Robotics Letters*, vol. 1, no. 1, pp. 1–18, 2013.

[13] 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.

[14] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[15] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *arXiv preprint arXiv:1704.06053*, 2017.

[16] D. Cilden-Guler, M. Raitoharju, R. Piche, and C. Hajiyev, "Nanosatellite attitude estimation using kalman-type filters with non-gaussian noise," *Aerospace Science and Technology*, vol. 92, pp. 66–76, 2019.