Q-Learning-Based Noise Covariance Adaptation in Kalman Filter for MARG Sensors Attitude Estimation

Xiang Dai, Vahid Nateghi, Hassen Fourati, Christophe Prieur

Q-Learning-Based Noise Covariance Adaptation in Kalman Filter for MARG Sensors Attitude Estimation

Xiang Dai
GIPSA-Lab, Inria, CNRS
Univ. Grenoble Alpes, Grenoble INP
Grenoble, France
xiang.dai@gipsa-lab.grenoble-inp.fr

Vahid Nateghi
Aerospace Science and Technology Dept.
Politecnico di Milano
Milan, Italy
vahid.nateghi@mail.polimi.it

Hassen Fourati
GIPSA-Lab, Inria, CNRS
Univ. Grenoble Alpes, Grenoble INP
Grenoble, France
hassen.fourati@gipsa-lab.grenoble-inp.fr

Christophe Prieur
GIPSA-Lab, CNRS
Univ. Grenoble Alpes, Grenoble INP
Grenoble, France
christophe.prieur@gipsa-lab.grenoble-inp.fr

***Abstract*—The attitude estimation of a rigid body by magnetic, angular rate, and gravity (MARG) sensors is a research subject for a large variety of engineering applications. A standard solution for building up the observer is usually based on the Kalman filter and its different extensions for versatility and practical implementation. However, the performance of these observers has long suffered from the inaccurate process and measurement noise covariance matrices, which in turn entails tedious parameter turning procedures. To overcome the laborious noise covariance matrices regulation, we propose in this paper a Q-learning-based approach to autonomously adapt the values of process and measurement noise covariance matrices. The Q-learning method establishes a reinforcement learning mechanism that forces the noise covariance matrices pair with the least difference between predictions and measurements of output to be found in a predetermined candidate set of noise covariance matrices. The effectiveness of the Q-learning approach, applied to Extended Kalman filter-based attitude estimation, is validated through the Monte Carlo method that uses real flight data on an unmanned aerial vehicle.**

***Index Terms*—Extended Kalman filter, Q-learning, Attitude estimation, Reinforcement learning**

I. Introduction

The determination of attitude and position are involved in navigation applications of any moving rigid bodies. The Kalman filter (KF) and especially its Extended version (EKF) for nonlinear dynamic models have been widely used for several years for this purpose [1]–[4]. A linearization step should be achieved for the nonlinear models of navigation at the current state estimate to approximate the KF equations. In such algorithms, it is well known that the model and measurement noise covariance matrices play essential roles, while it is challenging to design an optimal estimator in the absence of exact statistical knowledge about these matrices when using inertial and magnetic sensors, for example. The improper values of the noise covariance matrices in the filtering algorithm cause underestimation or overestimation of the uncertainty in model or sensor measurements, leading to degradation in attitude estimation performance, for example, usually represented by quaternions, rotation matrices, or Euler angles. Several approaches have been presented in the literature to overcome this problem of covariance matrix by using different strategies of adaptation of the measurement noise covariance matrix [5]–[7] or process noise one only [8]. An improvement is targeted if the adaptation is applied to the two noise covariance matrices. The main endeavor of this paper is to adapt the two noise covariance matrices simultaneously and in an autonomous way.

Recent advancements in Reinforcement Learning (RL) have made it appealing to be implemented to cope with uncertain environments. Specifically, this work is motivated by the strength of the Q-learning method [9]–[12] in which an intelligent agent learns how to take action in an environment with uncertain parameters. To the best of our knowledge, the process and measurement noise covariance matrices adaptation for the EKF based on Q-learning has not been solved yet within the scope of attitude and related states estimation by MARG sensors.

In this paper, we first formulate the attitude estimation problem of a rigid body, in which the rotation is represented by quaternion and the state vector to be estimated is composed of quaternion and gyroscope bias. In the following, the dynamic and observation models are illustrated, and the traditional EKF for this purpose is also formulated. Next, we propose a Q-learning-based extended Kalman filter (QLEKF) approach, which consists of three parallel filters. The first filter, a traditional EKF with initial values of noise covariance matrices, is run for filtering performance comparison. The

second filter, a learning EKF, is implemented to evaluate the estimation performance and search for appropriate noise covariance matrices. A reward is defined by comparing a predescribed index for the performance of the traditional EKF and the learning EKF, and the best couple of the process and measurement noise covariance matrices are selected. The third filter, the learned EKF, uses the noise covariance matrices returned by the learning EKF filter to provide the final estimated states. The estimator gradually adopts the most suitable noise covariance matrices selected by the Q-learning algorithm according to the cumulative reward.

The main contributions of the paper are twofold. First, the proposed algorithm successfully adapts the process and measurement noise covariance matrices for attitude and bias estimation using MARG sensors through the combination of Q-learning and EKF. Second, the superiority of the proposed algorithm over the traditional EKF in estimating attitude and gyroscope bias is validated through multiple Monte Carlo simulations based on real flight data.

The rest of the paper is organized as follows. Section II formulates the attitude representation and sensor models. Section III details the dynamic process and observation models, the traditional EKF, and the proposed QLEKF to estimate the rigid body attitude and gyroscope bias. Random Monte Carlo simulations that use real flight data are performed in Section IV. Conclusion and future research directions are given in Section V.

II. Attitude and Gyro Bias Models Formulation for Navigation

In this section, we first introduce quaternion to represent the attitude of rigid body and then formulate the sensor models. These materials are used later to establish the process/observation models and the estimation approaches.

A. Attitude representation with quaternion

In this paper, the attitude of the rigid body is represented by a quaternion-based formulation for its simplified algebra and universal application in 3D orientation. We first introduce the quaternion definition [13]:

$$\boldsymbol{q} = q_1 i + q_2 j + q_3 k + q_4, \tag{1}$$

where q_4 is the scalar part, $q_1 i + q_2 j + q_3 k$ is the vector part and i, j, k are hyperimaginary numbers.

The quaternion can also be written in a column vector, without loss of generality and for a concise formulation, the quaternions that appear in the rest of the paper are assumed with unit norm, that is:

$$\boldsymbol{q} = [q_1 \ q_2 \ q_3 \ q_4]^T, \ \|\boldsymbol{q}\| = 1. \tag{2}$$

Given a rigid body, let $\mathcal{B}$ denote its body frame and $\mathcal{N}$ denote the navigation (inertial) frame, then its position expressed in $\mathcal{B}$ (e.g. $\boldsymbol{p}^b \in \mathbf{R}^3$) and in $\mathcal{N}$ (e.g. $\boldsymbol{p}^n \in \mathbf{R}^3$) can be related by a rotation matrix as

$$\boldsymbol{p}^b = \boldsymbol{C}_n^b(\boldsymbol{q}_n^b)\boldsymbol{p}^n, \tag{3}$$

where $\boldsymbol{C}_n^b(\boldsymbol{q}_n^b) \in \mathbf{R}^{3\times 3}$ is the rotation matrix from $\mathcal{N}$ to $\mathcal{B}$ expressed in $\mathcal{B}$ using quaternion elements [14],

$$\boldsymbol{C}_n^b\left(\boldsymbol{q}_n^b\right) = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2\left(q_1q_2+q_3q_4\right) & 2\left(q_1q_3-q_2q_4\right) \\ 2\left(q_1q_2-q_3q_4\right) & 1-2q_1^2-2q_3^2 & 2\left(q_2q_3+q_1q_4\right) \\ 2\left(q_1q_3+q_2q_4\right) & 2\left(q_2q_3-q_1q_4\right) & 1-2q_1^2-2q_2^2 \end{bmatrix}.$$

Without arising ambiguity, we denote $\boldsymbol{q} = \boldsymbol{q}_n^b$ hereinafter in the paper. As the rigid body moves in 3D space, its associated quaternion time derivative can be formulated as [13]:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\boldsymbol{q}, \tag{4}$$

where $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T \in \mathbf{R}^3$ is the true angular velocity of $\mathcal{B}$ w.r.t. $\mathcal{N}$ expressed in $\mathcal{B}$, and

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \ \lfloor\boldsymbol{\omega}\times\rfloor = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}.$$

Let $T_e = t_{k+1} - t_k$ denote the sampling interval. To render (4) to be computationally efficient, we consider a discrete model under the assumption that the evolution of $\boldsymbol{\omega}$ during T_e is linear. In correspondence, the first order derivative of $\boldsymbol{\omega}$ can be expressed as a constant:

$$\dot{\boldsymbol{\omega}} = (\boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k)/T_e,$$

and higher order derivatives are accordingly zero.

Applying Taylor series expansion to $\boldsymbol{q}_{k+1}$ around time instant t_k by substituting the first order derivative (4) and neglecting items of higher order derivatives yield the first order quaternion integration [15] as

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \frac{T_e}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_k)\boldsymbol{q}_k. \tag{5}$$

B. Sensor models

The MARG sensors considered in this paper are composed of a 3-axis gyroscope, a 3-axis magnetometer and a 3-axis accelerometer. These sensors measure respectively real angular velocity $\boldsymbol{\omega}^r \in \mathbf{R}^3$, magnetic field $\boldsymbol{m} \in \mathbf{R}^3$ and acceleration $\boldsymbol{a} \in \mathbf{R}^3$ of the rigid body in $\mathcal{B}$ w.r.t. $\mathcal{N}$. The simplified measurement equations are formulated as follows [16]:

$$\begin{cases} \boldsymbol{\omega}^r = \boldsymbol{\omega} + \boldsymbol{b}^g + \boldsymbol{v}^g, \\ \boldsymbol{a} = \boldsymbol{C}_n^b(\boldsymbol{q})\boldsymbol{g} + \boldsymbol{v}^a, \\ \boldsymbol{m} = \boldsymbol{C}_n^b(\boldsymbol{q})\boldsymbol{h} + \boldsymbol{v}^m, \end{cases} \tag{6}$$

where $\boldsymbol{g} \in \mathbf{R}^3$ is the gravity vector, $\boldsymbol{h} \in \mathbf{R}^3$ is the earth magnetic field, $\boldsymbol{b}^g \in \mathbf{R}^3$ is the bias of gyroscope, and $\boldsymbol{v}^g \in \mathbf{R}^3$, $\boldsymbol{v}^a \in \mathbf{R}^3$, $\boldsymbol{v}^m \in \mathbf{R}^3$ are assumed to be uncorrelated Gaussian noises with zero mean and covariance matrices $\boldsymbol{\Sigma}_g = \sigma_g^2\boldsymbol{I}_3$, $\boldsymbol{\Sigma}_a = \sigma_a^2\boldsymbol{I}_3$ and $\boldsymbol{\Sigma}_m = \sigma_m^2\boldsymbol{I}_3$.

In this paper, we consider that the body linear acceleration expressed in $\mathcal{N}$ is zero since the rigid body is just rotating on itself and its center of mass is not moving. In addition, to be more focused on the attitude estimation, namely attitude of rigid body and gyroscope biases, accelerometer and magnetometer biases are omitted in the sensor models.

III. Dynamic process and observation models and learning-based Kalman filtering for navigation

In this section, we first build up the navigation dynamic model, based on which the traditional EKF is introduced to perform the attitude and bias estimation. In the continuation, the Q-learning method is proposed to recursively adapt the process and measurement noise covariance matrices in the EKF using a feedback from the uncertain environment.

A. Model design

The state vector consists of rotation quaternion and bias of gyroscope, and the state transition equation is formulated as

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \begin{bmatrix}\boldsymbol{q}_{k+1}\\ \boldsymbol{b}^g_{k+1}\end{bmatrix} = f(\boldsymbol{x}_k) + \boldsymbol{w}_k \\ &= \begin{bmatrix}\boldsymbol{I}_3 + \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^r_k - \boldsymbol{b}^g_k)T_e & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}_3\end{bmatrix}\begin{bmatrix}\boldsymbol{q}_k\\ \boldsymbol{b}^g_k\end{bmatrix} + \begin{bmatrix}\boldsymbol{w}^q_k\\ \boldsymbol{w}^g_k\end{bmatrix}, \end{aligned} \quad (7)$$

where $\boldsymbol{b}^g_k$ is the gyroscope bias at time step k and modeled as standard random walk, and $\boldsymbol{w}^q_k$ and $\boldsymbol{w}^g_k$ form the process noise, assumed to be uncorrelated Gaussian with zero mean and covariance matrices $\boldsymbol{\Sigma}^q_w = \sigma^2_{w,q}\boldsymbol{I}_3$ and $\boldsymbol{\Sigma}^g_w = \sigma^2_{w,g}\boldsymbol{I}_3$. As such, the process noise covariance matrix $\boldsymbol{Q}_k$ can be expressed as

$$\boldsymbol{Q}_k = \begin{bmatrix}\boldsymbol{\Sigma}^q_w & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Sigma}^g_w\end{bmatrix}. \quad (8)$$

In what follows, we construct the observation model by grouping those of accelerometer and magnetometer as

$$\begin{aligned}\boldsymbol{y}_k &= \begin{bmatrix}\boldsymbol{a}_k\\ \boldsymbol{m}_k\end{bmatrix} = g(\boldsymbol{x}_k) + \boldsymbol{v}_k \\ &= \begin{bmatrix}\boldsymbol{C}^b_n(\boldsymbol{q}_k) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{C}^b_n(\boldsymbol{q}_k)\end{bmatrix}\begin{bmatrix}\boldsymbol{g}\\ \boldsymbol{h}\end{bmatrix} + \begin{bmatrix}\boldsymbol{v}^a_k\\ \boldsymbol{v}^m_k\end{bmatrix}, \end{aligned} \quad (9)$$

where $\boldsymbol{v}^a_k$ and $\boldsymbol{v}^m_k$ are the measurement noises of accelerometer and magnetometer, and are assumed uncorrelated with a zero mean covariance matrices $\boldsymbol{\Sigma}^v_a = \sigma^2_{v,a}\boldsymbol{I}_3$ and $\boldsymbol{\Sigma}^v_m = \sigma^2_{v,m}\boldsymbol{I}_3$.

Correspondingly, the measurement noise covariance matrix $\boldsymbol{R}_k$ is expressed as

$$\boldsymbol{R}_k = \begin{bmatrix}\boldsymbol{\Sigma}^v_a & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Sigma}^v_m\end{bmatrix}. \quad (10)$$

B. The traditional extended Kalman filter

In this section, based on dynamic process and observation models presented before, first we introduce the EKF for state estimation, summarized in Alg. 1. We express the EKF in a higher level of abstraction as function format since it will be used to construct the QLEKF in the following subsection.

In Alg. 1, $\hat{\boldsymbol{x}}_{k|k-1}$ is the *a priori* state estimate at step k given the knowledge of the process model prior to step k, $\boldsymbol{A}$ is the Jacobian matrix of partial derivatives of f w.r.t. $\boldsymbol{x}$, $\boldsymbol{P}_{k|k-1} = E[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})^T]$ is the *a priori* error covariance matrix estimate at step k given knowledge of the process model prior to step k, $\boldsymbol{C}$ is the Jacobian matrix of partial derivatives of g w.r.t. $\boldsymbol{x}$, $\boldsymbol{K}_k$ is the Kalman gain at step k (derived from conditional probability density function given that $\boldsymbol{x}$ and $\boldsymbol{y}$ are jointly Gaussian distributed), $\tilde{\boldsymbol{y}}_k$ is the measurement innovation term at step k, $\hat{\boldsymbol{x}}_{k|k}$ is the *a posteriori* state estimate at step k given measurement $\boldsymbol{y}_k$, $\boldsymbol{P}_{k|k} = E[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k})(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k})^T]$ is the *a posteriori* error covariance matrix at step k given measurement $\boldsymbol{y}_k$.

Algorithm 1 Traditional Extended Kalman Filter

$\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \tilde{\boldsymbol{y}}_k = EKF(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{P}_{k-1|k-1}, \boldsymbol{y}_k, \boldsymbol{Q}_k, \boldsymbol{R}_k)$
Input $\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{P}_{k-1|k-1}, \boldsymbol{y}_k, \boldsymbol{Q}_k, \boldsymbol{R}_k$
$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1})$ {time update of state estimate}
$\boldsymbol{A} = \frac{\partial f}{\partial \boldsymbol{x}}|\hat{\boldsymbol{x}}_{k-1|k-1}$
$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}\boldsymbol{P}_{k-1|k-1}\boldsymbol{A}^T + \boldsymbol{Q}_k$ {time update of error covariance matrix}
$\boldsymbol{C} = \frac{\partial g}{\partial \boldsymbol{x}}|\hat{\boldsymbol{x}}_{k|k-1}$
$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{P}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{R}_k)^{-1}$ {compute the Kalman gain}
$\tilde{\boldsymbol{y}}_k = \boldsymbol{y}_k - g(\hat{\boldsymbol{x}}_{k|k-1})$ {compute the innovation term}
$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\tilde{\boldsymbol{y}}_k$ {measurement update of state estimate}
$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{C}\boldsymbol{P}_{k|k-1}$ {measurement update of error covariance matrix}
Output $\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \tilde{\boldsymbol{y}}_k$

C. Preliminaries on Q-learning approach

Q-learning is a method in reinforcement learning that maximizes the long-term reward in a multistate environment. That environment typically consists of discrete state-action pairs, each of which is assigned with a scalar, called Q-value.

In Q-learning, the agent learns to achieve a larger Q-value under uncertainty through a trial-and-error process by shifting among these state-action pairs. Suppose that the agent is at state s, and it needs to determine an action a to reach the next state s'. Commonly, there are various methods to determine how to choose the action [17] in Q-learning. In this paper, we adopt the ϵ-greedy algorithm [18]: a standard algorithm in the RL domain where the agent chooses a random action with a probability of ϵ or pick the action that maximizes the Q-value with the probability of (1-ϵ). The value of ϵ defines how much the agent should rely on the exploration or exploitation; the greater the ϵ, the higher the probability to explore.

Each time after executing an action a, the agent receives a response from the environment, which is translated to a reward (R) showing how good the action is. Significantly, Q-learning at its core seeks to maximize the cumulative reward by performing the best action at each state [12]. The cumulative reward is stored as Q-value through the Q-learning update rule as

$$Q(s,a) = Q(s,a) + \alpha[R + \gamma \max_a Q(s',a) - Q(s,a)], \quad (11)$$

where $Q(s,a) \in \mathbf{R}$ is the Q-value for the action a in state s, $R \in \mathbf{R}$ is the reward gained by exeuting action a in state s, α is the learning rate, and γ is the discount factor.

D. Q-learning-based EKF for covariance matrices adaptation

In the QLEKF approach, the agent attempts to find appropriate values of process and measurement noise covariance matrices to improve the performance of the traditional EKF.

To implement the QLEKF, a states set S, composed of M candidates of process noise covariance matrices $\{\boldsymbol{Q}_k^{(1)}, \boldsymbol{Q}_k^{(2)}, ..., \boldsymbol{Q}_k^{(M)}\}$ and N candidates of measurement noise covariance matrices $\{\boldsymbol{R}_k^{(1)}, \boldsymbol{R}_k^{(2)}, ..., \boldsymbol{R}_k^{(N)}\}$, is first conceived to form a $M \times N$ grid. As such, element (i, j) in the grid, namely state (i, j), represents the couple of noise covariance matrices $(\boldsymbol{Q}_k^{(i)}, \boldsymbol{R}_k^{(j)})$. In the continuation, an action set $A_{i,j}$ is formed with valid actions that can transit (i, j) to its adjacent states in the grid or keep it at (i, j). It is worth mentioning that the subset of valid actions at each element of the grid depends on the location of the agent comprising the corner area, the boundary area, and the central area. The last thing that needs to be defined is the reward. One can assume the reward is the difference of the innovation sequences between the EKF with the nominal (initial) process and measurement noise covariance matrices and the EKF with $\boldsymbol{Q}_k^{(i)}$ and $\boldsymbol{R}_k^{(j)}$ corresponding to the current state of the agent.

The QLEKF comprises three EKFs as shown in Figure 1: the traditional EKF, which uses an initial value of covariance matrices; the learning EKF, which searches appropriate covariance matrices by the Q-learning algorithm; and the learned EKF, which outputs the result of estimation according to the covariance matrices found by the learning EKF.

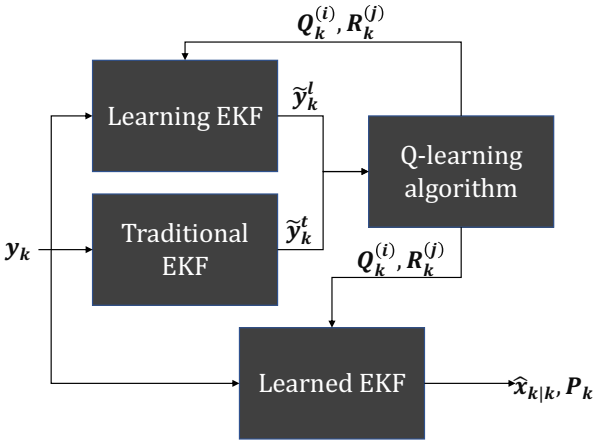

Fig. 1: Schematic diagram of QLEKF

Based on the description of EKF in Subsection III-B, the QLEKF is presented in Alg. 2. Steps 12, 13, and 15 indicate the traditional EKF, the learning EKF, and the learned EKF respectively. Step 20 ensures the identical initial conditions at each iteration for the traditional EKF and the learning EKF, which brings a fair and reliable reward accumulation in Step 14. In Alg. 2, superscripts t and l stand for the traditional and learning EKF, respectively.

Algorithm 2 Q-learning Extended Kalman Filter

1: Initialize $\hat{\boldsymbol{x}}_{0|0}^t$, $\hat{\boldsymbol{x}}_{0|0}^l$, $\boldsymbol{P}_{0|0}^t$, $\boldsymbol{P}_{0|0}^l$, s and ϵ
2: $k \leftarrow 1$, $Q(s, a) \leftarrow \mathbf{0}$, $R \leftarrow 0$
3: **for** each iteration **do**
4: Generate N from uniform distribution $N \sim U(0, 1)$
5: **if** $N < \epsilon$ **then**
6: select action a randomly
7: **else**
8: $a = \arg\max_a Q(s, a)$
9: **end if**
10: Execute action a and obtain state s'
11: **for** each time step in one iteration **do**
12: $[\hat{\boldsymbol{x}}_{k|k}^t, \boldsymbol{P}_{k|k}^t, \tilde{\boldsymbol{y}}_k^t]$
$= EKF(\hat{\boldsymbol{x}}_{k-1|k-1}^t, \boldsymbol{P}_{k-1|k-1}^t, \boldsymbol{y}_k^t, \boldsymbol{Q}_k, \boldsymbol{R}_k)$
13: $[\hat{\boldsymbol{x}}_{k|k}^l, \boldsymbol{P}_{k|k}^l, \tilde{\boldsymbol{y}}_k^l]$
$= EKF(\hat{\boldsymbol{x}}_{k-1|k-1}^l, \boldsymbol{P}_{k-1|k-1}^l, \boldsymbol{y}_k^l, \boldsymbol{Q}_k^{(i)}, \boldsymbol{R}_k^{(j)})$
14: $R \leftarrow R + \{[(\tilde{\boldsymbol{y}}_k^t)^T \tilde{\boldsymbol{y}}_k^t]^{0.5} - [(\tilde{\boldsymbol{y}}_k^l)^T \tilde{\boldsymbol{y}}_k^l]^{0.5}\}$
15: $[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \tilde{\boldsymbol{y}}_k]$
$= EKF(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{P}_{k-1|k-1}, \boldsymbol{y}_k, \boldsymbol{Q}_k^{(i)}, \boldsymbol{R}_k^{(j)})$
16: $k \leftarrow k + 1$
17: **end for**
18: $Q(s, a) = Q(s, a) + \alpha[R + \gamma \max_a Q(s', a) - Q(s, a)]$
19: $s \leftarrow s'$, $R \leftarrow 0$
20: $\hat{\boldsymbol{x}}_{k|k}^l \leftarrow \hat{\boldsymbol{x}}_{k|k}^t$, $\boldsymbol{P}_{k|k}^l \leftarrow \boldsymbol{P}_{k|k}^t$
21: **end for**
22: **return** $\{\hat{\boldsymbol{x}}_{k|k}\}$ and $\{\boldsymbol{P}_{k|k}\}$

IV. Numerical simulations with real data

In this section, the effectiveness of the learned EKF in Alg. 2 is validated and compared with the traditional EKF using data from a real flight trajectory of an unmanned aerial vehicle.

A. Simulation setup

We use the ground truth data from the real flight trajectory of an unmanned aerial vehicle extracted from the experiments Euroc [19] at http://robotics.ethz.ch/~asl-datasets/ijrr_euroc_mav_dataset/machine_hall/MH_01_easy. From the ground truth data, we select the bias of gyroscope $\boldsymbol{b}^g$ and quaternion. The latter is first transformed into unit quaternion and then substituted into (5) to solve out the true $\boldsymbol{\omega}$ using matrix pseudo inverse.

For the numerical simulations, the sampling rate of the MARG sensors is set to 100Hz, the gravity and the earth magnetic field vectors are set to $\boldsymbol{g} = [0, 0, 9.81]$ m/s^2, $\boldsymbol{h} = [0.23, 0.01, 0.41]$ Gauss. For the initialization of each Monte Carlo simulation, the quaternion is set as the first quaternion of the ground truth data[1], and the gyroscope bias and the error covariance matrix are set to $\boldsymbol{b}_0^g = [2.2, 2, 2]$ mrad/s, $\boldsymbol{P}_0 = 10\boldsymbol{I}_7$.

In terms of the traditional EKF, we set $\sigma_{w,q} = 1 \times 10^{-3}$, $\sigma_{w,g} = 0.01$ rad/s, $\sigma_{v,a} = 0.01$ m/s^2, and $\sigma_{v,m} = 1 \times$

[1]It needs to be normalized as unit quaternion to keep all the quaternion along the simulation being unit quaternion.

TABLE I: Average statistics of mean of quaternion error norm and Root Mean Square Error (RMSE) of gyroscope bias after convergence in 50 Monte Carlo simulations

	Mean of quaternion error* ($\times 10^{-3}$)	RMSE of gyroscope bias (mrad/s)		
		x-axis	y-axis	z-axis
Traditional EKF	1.419	9.944	4.449	6.078
Learned EKF	0.866	4.893	2.846	3.912

* For a ground truth unit quaternion $\boldsymbol{q}_{true}$, the quaternion error of its unit estimate $\hat{\boldsymbol{q}}$ is computed as $\|\boldsymbol{q}_{true}^{-1} \otimes \hat{\boldsymbol{q}} - \boldsymbol{q}_I\|$, where $\boldsymbol{q}_I = [0\ 0\ 0\ 1]^T$ is the identity quaternion.

10^{-3} m/s^2. The size of Q-learning grid is set to $M = N = 5$, of which $\{\boldsymbol{Q}_k^{(i)}\}$ and $\{\boldsymbol{R}_k^{(j)}\}$ are set as a geometric progression with ratio of 10 and $\boldsymbol{Q}_k^{(3)} = \boldsymbol{Q}_k$, $\boldsymbol{R}_k^{(3)} = \boldsymbol{R}_k$. The Q-learning search starts at the grid $(\boldsymbol{Q}_k^{(1)}, \boldsymbol{R}_k^{(1)})$. For each Monte Carlo simulation, we run 200 iterations and 100 time steps for each iteration. The learning rate, discount factor and random action selection probability are fixed to $\alpha = 0.1$, $\gamma = 0.9$ and $\epsilon = 0.1$.

B. Results and analysis

Table I shows that after convergence of 50 Monte Carlo simulations, the learned EKF evidently outperforms the traditional EKF in estimating the rigid body attitude and the gyroscope bias, where the learned EKF averagely improves 39.0% of the mean of quaternion error norm and 50.8%, 36.0% and 35.6% of RMSE of gyroscope bias of x, y, and z axes compared to the traditional EKF.

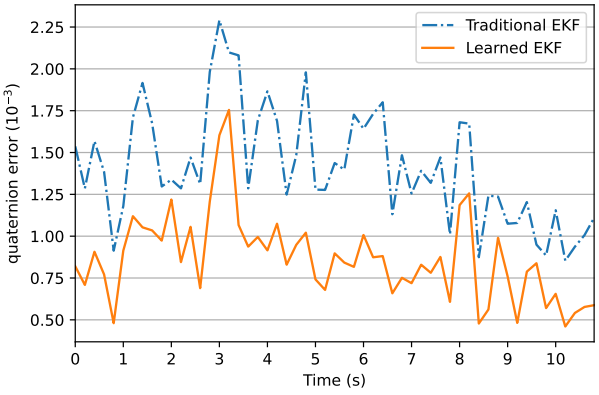

Fig. 2: Trajectory of average quaternion error between traditional EKF and learned EKF after convergence in 50 Monte Carlo simulations

In Fig. 2-5, we plot the evolution of quaternion error norm and gyroscope bias error, in which each point on the curve stands for the mean of 20 consecutive samplings of 50 Monte Carlo simulations after convergence, making 5 points collected during the period of 1 sec. As shown in Fig. 2, the 2 curves share similar tendency and fluctuations after convergence, while the quaternion error of the learned EKF is always distinctly lower than that of the traditional EKF in

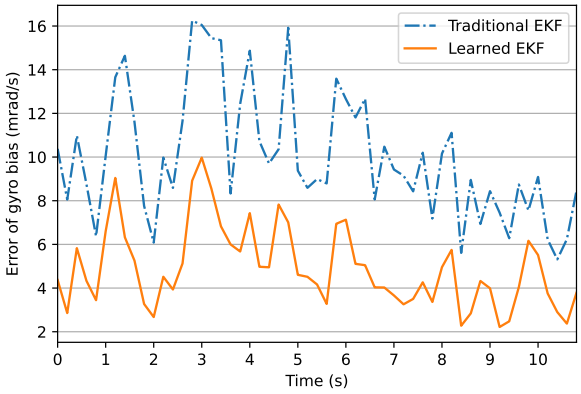

Fig. 3: Trajectory of average error of gyroscope bias on x axis between traditional EKF and learned EKF after convergence in 50 Monte Carlo simulations

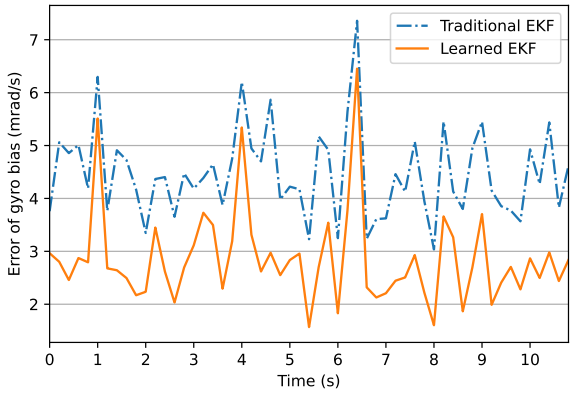

Fig. 4: Trajectory of average error of gyroscope bias on y axis between traditional EKF and learned EKF after convergence in 50 Monte Carlo simulations

the whole convergence period, which indicates that the reward update (Step 14 in Alg. 2) and Q-value update (Step 18 in Alg. 2) force $\{\boldsymbol{Q}_k^{(i)}\}$ and $\{\boldsymbol{R}_k^{(j)}\}$ to search for values with better innovation term, thus leading to a better attitude estimate after convergence. Similar behaviors of error of gyroscope bias can be observed in Fig. 3-5 as well, which again demonstrates the advantage of the learned EKF in the tuning process and measurement noise covariance matrices over the traditional EKF.

V. Conclusions

In this paper, we have first introduced a traditional EKF to estimate the attitude and gyroscope bias of a rigid body in rotation movement. To address the often-cumbersome tuning of process and measurement noise covariance matrices in the EKF, we have then introduced a Q-learning algorithm to enforce the filter to search for more accurate covariance matrices along the estimation course. This is realized by

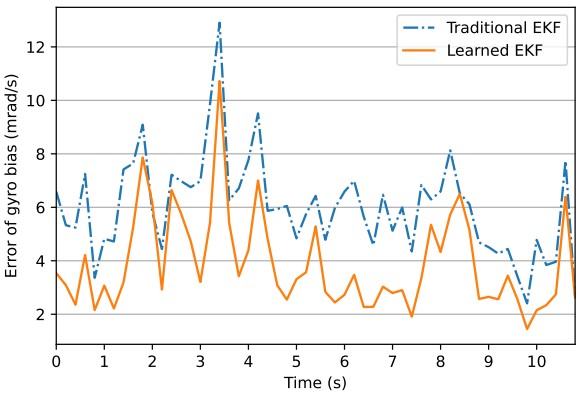

Fig. 5: Trajectory of average error of gyroscope bias on z axis between traditional EKF and learned EKF after convergence in 50 Monte Carlo simulations

assigning a reward to covariance matrices that can produce better innovation terms than the traditional EKF, which encourages covariance matrices associated with better measurement estimation to be more likely selected by the reinforcement learning process.

Through Monte Carlo numerical simulations based on real flight data of an unmanned aerial vehicle, the proposed learned EKF on average has revealed an improvement of 39.0% in attitude estimation and at least 35.6% in gyroscope bias estimation compared to the traditional EKF after convergence.

Future work on the QLEKF can be undertaken by incorporating the position and linear velocity in the estimation, and combining vision data with an inertial measurement unit (IMU) for better navigation.

REFERENCES

[1] M. Zmitri, H. Fourati, and C. Prieur, "BiLSTM network-based extended Kalman filter for magnetic field gradient aided indoor navigation," *IEEE Sensors Journal*, 2022.

[2] M. Zmitri, H. Fourati, and C. Prieur, "Magnetic field gradient-based EKF for velocity estimation in indoor navigation," *Sensors*, vol. 20, no. 20, p. 5726, 2020.

[3] A. M. Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.

[4] S. Bonnabel, "Left-invariant extended Kalman filter and attitude estimation," in *2007 46th IEEE Conference on Decision and Control*, pp. 1027–1032, IEEE, 2007.

[5] Á. Odry, I. Kecskes, P. Sarcevic, Z. Vizvari, A. Toth, and P. Odry, "A novel fuzzy-adaptive extended kalman filter for real-time attitude estimation of mobile robots," *Sensors*, vol. 20, no. 3, p. 803, 2020.

[6] A. Assad, W. Khalaf, and I. Chouaib, "Novel adaptive fuzzy extended Kalman filter for attitude estimation in GPS-denied environment," *Gyroscopy and Navigation*, vol. 10, no. 3, pp. 131–146, 2019.

[7] K. Xiong and C. Wei, "Adaptive iterated extended Kalman filter for relative spacecraft attitude and position estimation," *Asian Journal of Control*, vol. 20, no. 4, pp. 1595–1610, 2018.

[8] H. Wang, Z. Deng, B. Feng, H. Ma, and Y. Xia, "An adaptive Kalman filter estimating process noise covariance," *Neurocomputing*, vol. 223, pp. 12–17, 2017.

[9] D. Kim, T. Lee, S. Kim, B. Lee, and H. Y. Youn, "Adaptive packet scheduling in IoT environment based on Q-learning," *Procedia Computer Science*, vol. 141, pp. 247–254, 2018.

[10] S. A. A. Rizvi and Z. Lin, "Output feedback Q-learning for discrete-time linear zero-sum games with application to the H-infinity control," *Automatica*, vol. 95, pp. 213–221, Sept. 2018.

[11] A. Maoudj and A. L. Christensen, "Q-learning-based navigation for mobile robots in continuous and dynamic environments," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 1338–1345, IEEE, 2021.

[12] K. Xiong, C. Wei, and H. Zhang, "Q-learning for noise covariance adaptation in extended Kalman filter," *Asian Journal of Control*, vol. 23, no. 4, pp. 1803–1816, 2021.

[13] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.

[14] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton university press, 1999.

[15] J. R. Wertz, *Spacecraft Attitude Determination and Control*, vol. 73. Springer Science & Business Media, 2012.

[16] H. Fourati, N. Manamanni, L. Afilal, and Y. Handrich, "A Nonlinear Filtering Approach for the Attitude and Dynamic Body Acceleration Estimation Based on Inertial and Magnetic Sensors: Bio-Logging Application," *IEEE Sensors Journal*, vol. 11, pp. 233–244, Jan. 2011.

[17] J. Jiang and J. Xin, "Path planning of a mobile robot in a free-space environment using Q-learning," *Progress in artificial intelligence*, vol. 8, no. 1, pp. 133–142, 2019.

[18] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *MIT press*, 2018.

[19] 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.