2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Reinforcement Learning for Orientation Estimation Using Inertial Sensors with Performance Guarantee

Liang Hu[1*], Yujie Tang[2*], Zhipeng Zhou[2] and Wei Pan[2]

***Abstract*— This paper presents a deep reinforcement learning (DRL) algorithm for orientation estimation using inertial sensors combined with a magnetometer. Lyapunov's method in control theory is employed to prove the convergence of orientation estimation errors. The estimator gains and a Lyapunov function are parametrised by deep neural networks and learned from samples based on the theoretical results. The DRL estimator is compared with three well-known orientation estimation methods on both numerical simulations and real dataset collected from commercially available sensors. The results show that the proposed algorithm is superior for arbitrary estimation initialisation and can adapt to a drastic angular velocity profile for which other algorithms can be hardly applicable. To the best of our knowledge, this is the first DRL-based orientation estimation method with an estimation error boundedness guarantee.**

I. Introduction

Orientation estimation is essential in robotics, navigation, control, and human motion analysis [1], [2], [3]. Recently, orientation estimation has been dramatically advanced by the development of accurate sensors. Multiple sensors are usually combined to estimate the orientation, i.e., sensor fusion. Depending on the availability of sensors and applications, various sensor fusion techniques have been proposed, e.g., the inertial measurement units (IMU) and magnetometer [4], [5], [6], the magnetometer and camera [7], and the IMU and visual sensor [8], [9], etc. In this paper, we focus on orientation estimate using inertial sensors and a magnetometer.

The estimation algorithms can be summarised into three categories: (1) Bayesian estimation, (2) optimisation and (3) deep learning. In Bayesian estimation, the well-known extended Kalman filter (EKF) and the unscented Kalman filter (UKF) were used to estimating the orientation [4], [5], [10]. The key idea is to approximate the orientation states by a Gaussian distribution based on the linearisation technique and the deterministic sampling technique. Furthermore, the complementary filter was developed based on the EKF, which exploits the complementary characteristics of gyroscopes and that of accelerometer and magnetometer at different time scales [6]. In optimisation, the orientation estimation is obtained based on gradient-based optimisation algorithms [11], [12]. Until recently, deep learning was introduced to estimate the orientation [13], in which a deep neural network is trained to mimic the noise distribution of gyroscopes such that accurate orientation estimates can be obtained by open-loop integration of the noise-free gyro measurements. These algorithms showed superior estimation performance empirically. However, the performance can not be theoretically guaranteed, i.e., the orientation estimate error never diverges. This paper will employ Lyapunov's method in control theory to prove the estimation error boundedness guarantee using samples. Based on the theoretical result, we will develop a reinforcement learning (RL) based algorithm to learn the estimator from samples.

RL was first applied for state estimation in [14]. Motivated by this work, we plan to develop an RL algorithm to learn the estimator gain using samples while the orientation estimator remains the structure of conventional EKF. The key idea is, the estimator gain will be approximated by a deep neural network (DNN) as a function of the sequence of estimate errors. Unlike other popular RL algorithms [15], [16], [17], the value function will be treated as a Lyapunov function used to guarantee the estimation performance. Lyapunov's method has been widely used as a basic tool for stability analysis in control theory [18]. To analyse the stability, the key is to find a scalar "energy-like" Lyapunov function for the considered system such that the derivative/difference of Lyapunov function along the state trajectory is semi-negative definite. Nonetheless, the construction/learning of the Lyapunov function is not trivial. In [19], a straightforward approach is proposed to construct the Lyapunov function for nonlinear systems using DNNs. Recently, the asymptotic stability in model-free RL is given for robotic control tasks in [20]. Inspired by the works [19], [20], we will also parametrise the Lyapunov function as a DNN and learn the parameters from samples. After that, a soft actor-critic (SAC) like algorithm [17] that incorporates the Lyapunov boundedness condition in the objective function to be optimised is proposed. Using the learned estimator gain, the estimate error of the orientation estimator is guaranteed to be bounded all the time.

In summary, we combine Lyapunov's method and DRL to design a state estimator with estimation error boundedness guarantee for orientation estimation. The main contribution of this paper has threefold:

1) To the best of our knowledge, this is the first DRL-based orientation estimation method using inertial sensors combined with a magnetometer;
2) The boundedness guarantee for estimation error is proved using Lyapunov's method in control theory;
3) The proposed algorithm is superior for arbitrary estimation initialisation and can adapt to enormous angular velocities for which other algorithms, such as the EKF,

*Equal Contribution.

[1]L. Hu is with the School of Computer Science and Electronic Engineering, University of Essex, UK.

[2]Y. Tang, Z. Zhou and W. Pan are with the Department of Cognitive Robotics, Delft University of Technology, Netherlands. For Correspondence: wei.pan@tudelft.nl.

UKF and complementary filter algorithms, can be hardly applicable.

The rest of the paper is organised as follows. In Section II, the orientation estimation problem is formulated. Section III, the theoretical result on estimation error boundedness guarantee is proved. Section IV, a DRL algorithm based on Soft Actor-Critic (SAC) combined with theoretical results, is proposed to learn the estimator gain. In Section V, our method is compared with the EKF, UKF and complementary filter algorithms on simulated and real datasets. The conclusion is given in Section VI.

II. Problem Formulation

This paper uses the inertial sensors (3D accelerometers and 3D gyroscopes) combined with the magnetometer to estimate the orientation. As in [3], the system dynamics is the standard orientation dynamics in (1). Moreover, our goal is to design the estimator gain like the classic Kalman filter. Unlike other nonlinear filtering techniques based on linearisation, we will show that the estimator gain's computation can be solved as an RL problem.

A. System dynamics and state estimator

The orientation dynamics is standard as given in [3]:

$$q_{t+1}^{\mathrm{nb}} = q_t^{\mathrm{nb}} \odot \exp_q\left(\frac{T}{2}(y_{\omega,t} - e_{\omega,t})\right), \tag{1}$$

where $q_t^{\mathrm{nb}} \in \mathbb{R}^4$ is the unit quaternion for the orientation of the body frame with respect to the navigation frame at time instant $t \in [0, T]$, $\exp_q(\cdot)$ corresponds to the exponential function of the quaternion, and $y_{\omega,t}$ is the gyroscope measurement. The distribution of the gyroscope noise is assumed to be Gaussian, i.e., $e_{\omega,t} \sim \mathcal{N}(0, \Sigma_\omega)$ where Σ_ω is the covariance matrix.

Assuming that the linear acceleration is approximately zero, the measurement equations are given as follows:

$$y_{\mathrm{a},t} = -R_t^{\mathrm{bn}} g^{\mathrm{n}} + e_{\mathrm{a},t}, \tag{2a}$$

$$y_{\mathrm{m},t} = R_t^{\mathrm{bn}} m^{\mathrm{n}} + e_{\mathrm{m},t}, \tag{2b}$$

where $y_{\mathrm{a},t}, y_{\mathrm{m},t} \in \mathbb{R}^3$ are accelerometer and magnetometer measurements at time instant t respectively, R_t^{bn} is the rotation matrix from the navigation frame to the body frame at time instant t, $g^{\mathrm{n}}, m^{\mathrm{n}}$ denote the local earth gravity vector and the local earth magnetic field vector, respectively. The noises $e_{\mathrm{a},t} \sim \mathcal{N}(0, \Sigma_{\mathrm{a}})$, and $e_{\mathrm{m},t} \sim \mathcal{N}(0, \Sigma_{\mathrm{m}})$ with $\Sigma_m = \sigma_m^2 \mathcal{I}_3$ and $\Sigma_{\mathrm{a}} = \sigma_{\mathrm{a}}^2 \mathcal{I}_3$.

To estimate q_{t+1}^{nb}, the following estimator in terms of the orientation deviation is often proposed [3], [21]:

$$\hat{q}_{t+1|t}^{\mathrm{nb}} = \hat{q}_{t|t}^{\mathrm{nb}} \odot \exp_q\left(\tfrac{T}{2} y_{\omega,t}\right), \tag{3a}$$

$$\hat{\eta}_{t+1} = K_{t+1}(y_{t+1} - \hat{y}_{t+1|t}), \tag{3b}$$

$$\hat{q}_{t+1|t+1}^{\mathrm{nb}} = \exp_q(\hat{\eta}_{t+1}) \odot \hat{q}_{t+1|t}^{\mathrm{nb}} \tag{3c}$$

with

$$y_t = \begin{pmatrix} y_{\mathrm{a},t} \\ y_{\mathrm{m},t} \end{pmatrix}, \quad \hat{y}_{t+1|t} = \begin{pmatrix} -R\left\{\hat{q}_{t|t}^{\mathrm{nb}} \odot \exp_q(\frac{T}{2} y_{\omega,t})\right\}^\top g^n \\ R\left\{\hat{q}_{t|t}^{\mathrm{nb}} \odot \exp_q(\frac{T}{2} y_{\omega,t})\right\}^\top m^n \end{pmatrix},$$

where $\hat{q}_{t+1|t}^{\mathrm{nb}}$ is the linearisation point parametrised in terms of quaternions, $\hat{\eta}_{t+1}^{\mathrm{n}}$ is the state estimate of the orientation deviation, and $R\{\cdot\}$ denotes the matrix formula of translation from quaternion to rotation. The goal is to obtain K_{t+1}, i.e., the estimator gain at time instant $t+1$, which will be explained later in Section II-B.

Define the orientation error

$$\tilde{q}_t \triangleq q_t^{\mathrm{nb}} \odot \left(\hat{q}_{t|t}^{\mathrm{nb}}\right)^{\mathrm{c}}, \tag{4}$$

or equivalently $q_t^{\mathrm{nb}} = \tilde{q}_t \odot \hat{q}_{t|t}^{\mathrm{nb}}$. From (1), (2) and (3), we have

$$\begin{aligned} \tilde{q}_{t+1} &= q_{t+1}^{\mathrm{nb}} \odot \left(\hat{q}_{t+1|t+1}^{\mathrm{nb}}\right)^{\mathrm{c}} \\ &= \Big(\left(\tilde{q}_t \odot \hat{q}_{t|t}^{\mathrm{nb}}\right) \odot \exp_q\left(\tfrac{T}{2}(y_{\omega,t} - e_{\omega,t})\right)\Big) \odot \\ &\quad \Big(\Big(\exp_q\left(\tfrac{1}{2} K_{t+1}\left(y_{t+1} - \hat{y}_{t+1|t}\right)\right) \odot \hat{q}_{t|t}^{\mathrm{nb}} \odot \exp_q(\tfrac{T}{2} y_{\omega,t})\Big)\Big)^{\mathrm{c}} \end{aligned} \tag{5}$$

where, $(\cdot)^c$ denotes the conjugate of quaternion.

Furthermore, to escape the unit determinant condition of the quaternion representation of rotation, the logarithm map of the quaternion is used [22]:

$$[\eta_{t+1}]_\times = \log(\tilde{q}_{t+1}) \tag{6}$$

where η_{t+1} is the orientation deviation and the skew operator $[\cdot]_\times$ produces the cross-product matrix.

B. Estimate Error Dynamics as Markov Decision Process

By combining (5) and (6), the estimate error dynamics can actually be modelled as a Markov decision process (MDP) which is defined as a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C}, \gamma >$:

$$\tilde{q}_{t+1} \sim \mathcal{P}\left(\tilde{q}_{t+1} | \tilde{q}_t, K_{t+1}\right), \forall t \in \mathbb{Z}_+, \tag{7}$$

where the estimate error $\tilde{q}_t \in \mathcal{S}$ is the state, the estimator gain $K_{t+1} \in \mathcal{A}$ is the action sampled from a stochastic policy. Considering that the ground truth q_t is known during training phase, the mapping between $\tilde{q}_t$ and $\hat{\eta}_t$ is bijective to some extent according to (3c) and (4). For convenience, in our implementation of the algorithm, we treat $\pi(K_{t+1}|\hat{\eta}_t)$ and $\pi(K_{t+1}|\tilde{q}_t)$ equivalently.

The state dynamics can be characterised by the transition probability function $\mathcal{P}(\tilde{q}_{t+1}|\tilde{q}_t, K_{t+1})$. RL algorithms can be used to find the policy π, given a cost function[1] $C(\tilde{q}_t, K_{t+1}) \in \mathcal{C}$ that measures the goodness of a state-action pair. In state estimation, it is often desired that the estimate error $\tilde{q}_t$ converges exponentially to a finite bound in mean square. As such, the cost function is selected as $C(\tilde{q}_t, K_{t+1}) = \mathbb{E}_{P(\cdot|\tilde{q}_t, K_{t+1})}[\|\tilde{q}_{t+1}\|^2]$, and the return is the sum of discounted cost $\sum_{\tau=t}^{\infty} \gamma^{\tau-t} C(\tilde{q}_t, K_{t+1})$ with the discount factor $\gamma \in [0,\ 1)$, where $\mathbb{E}[\cdot]$ denotes the expected value.

Definition 1: [23] The estimate error $\tilde{q}_t$ in the MDP (7) is said to be exponentially bounded in mean square if $\exists\ \eta > 0$ and $0 < \varphi < 1$, such that

$$\mathbb{E}[\|\tilde{q}_t\|^2] \leq \eta \mathbb{E}[\|\tilde{q}_0\|^2]\varphi^t + p, \tag{8}$$

[1]We will use cost, which is often used in control literature, instead of reward.

holds at all the time instants $t \geq 0$, where p is a positive constant number.

In this paper, our goal is to learn the estimator gain $K_{t+1} = \pi(\hat{\eta}_t)$ in (3) which can be seen as a policy obtained using an RL algorithm, such that the mean square of the estimate error of $\tilde{q}_t$ in (7) is guaranteed to converge exponentially to a positive bound. Different from the EKF where K_{t+1} is computed using the linearisation approximation, in this paper K_{t+1} is approximated by a DNN $\pi(\cdot)$.

III. Estimation Error Boundedness Guarantee

In this section, we propose the main theorem to guarantee the boundedness of the estimate error. Before proceeding, some notations need be clarified. $\rho(\tilde{q}_0)$ denotes the distribution of the starting state estimate error $\tilde{q}_0$. The state distribution of state estimate error at a certain instant t as $P(\tilde{q}_t|\rho,\pi,t)$ is defined in an iterative way: $P(\tilde{q}_{t+1} = s'|\rho,\pi,t+1) = \int_S P(\tilde{q}_t = s|\rho,\pi,t)P_\pi(s'|s)ds$. The following assumption, which is often used in RL literature, is needed:

Assumption 1: The Markov chain in (7) induced by a policy π is ergodic with a unique distribution probability. That is, $\exists\ p_\pi(s)$, such that

$$p_\pi(s) = \lim_{t\to\infty} P(\tilde{q}_t = s|\rho,\pi,t) \tag{9}$$

Theorem 1: The error dynamics (7) is exponentially bounded in mean square if there exists a Lyapunov function $L(\tilde{q}_t) : S \to R^+$ and positive constants α_1, α_2 and δ such that

$$\alpha_1 \mathbb{E}_\pi[\|\tilde{q}_t\|^2] - \delta \leq L(\tilde{q}_t) \leq \alpha_1 \mathbb{E}_\pi[\|\tilde{q}_t\|^2] \tag{10}$$

and

$$\begin{aligned}\lim_{N\to+\infty}[\ln(\mathbb{E}_{\tilde{q}_t\sim\mu_N}(\mathbb{E}_{\tilde{q}_{t+1}\sim P_\pi} L(\tilde{q}_{t+1})))\\ -\mathbb{E}_{\tilde{q}_t\sim\mu_N}\ln(L(\tilde{q}_t))] \leq -\alpha_2\end{aligned} \tag{11}$$

where

$$\mu_N(s) \triangleq \frac{1}{N}\sum_{t=0}^{N-1} P(\tilde{q}_t = s|\rho,\pi,t) \tag{12}$$

Proof: We have

$$\begin{aligned}&\ln(\mathbb{E}_{\tilde{q}_t\sim\mu_N}(\mathbb{E}_{\tilde{q}_{t+1}\sim P_\pi} L(\tilde{q}_{t+1}))\\ =&\ln(\int_S \frac{1}{N}\sum_{t=0}^{N-1} P(\tilde{q}_t = s|\rho,\pi,t)\int_S P_\pi(s'|s)L(s')\,ds'\,ds)\\ =&\ln(\int_S(\int_S \frac{1}{N}\sum_{t=0}^{N-1} P(\tilde{q}_t = s|\rho,\pi,t)P_\pi(s'|s)\,ds)L(s')\,ds')\\ =&\ln(\int_S(\frac{1}{N}\sum_{t=0}^{N-1} P(\tilde{q}_{t+1} = s'|\rho,\pi,t+1))L(s')\,ds')\\ =&\ln((\frac{1}{N}\sum_{t=0}^{N-1}\int_S P(\tilde{q}_{t+1} = s'|\rho,\pi,t+1))L(s')\,ds')\\ \geq&\frac{1}{N}\sum_{t=0}^{N-1}\ln((\int_S P(\tilde{q}_{t+1} = s'|\rho,\pi,t+1))L(s')\,ds')\end{aligned} \tag{13}$$

where the last inequality follows from the fact that $\ln(x)$ is a concave function on R^+. Similarly, noting that $-\ln(x)$ is

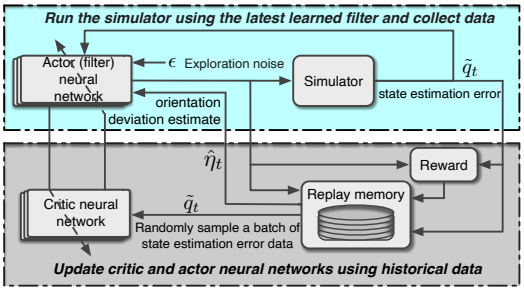


Fig. 1. Offline RL training process of orientation state estimator

a convex function we have

$$\begin{aligned}&-\mathbb{E}_{\tilde{q}_t\sim\mu_N}\ln L(\tilde{q}_t)\\ =&-\int_S \frac{1}{N}\sum_{t=0}^{N-1} P(\tilde{q}_t = s|\rho,\pi,t)\ln(L(s))\,ds\\ =&\frac{1}{N}\sum_{t=0}^{N-1}\int_S P(\tilde{q}_t = s|\rho,\pi,t)(-\ln L(s))\,ds\\ \geq&\frac{1}{N}\sum_{t=0}^{N-1} -\ln(\int_S P(\tilde{q}_t = s|\rho,\pi,t)L(s)\,ds)\end{aligned} \tag{14}$$

It follows from the above two inequalities that

$$\begin{aligned}&\ln(\mathbb{E}_{\tilde{q}_t\sim\mu_N}(\mathbb{E}_{\tilde{q}_{t+1}\sim P_\pi} L(\tilde{q}_{t+1})) - \mathbb{E}_{\tilde{q}_t\sim\mu_N}\ln L(\tilde{q}_t))\\ \geq&\frac{1}{N}\sum_{t=0}^{N-1}\ln\frac{\int_S P(\tilde{q}_{t+1} = s'|\rho,\pi,t+1)L(s')\,ds'}{\int_S P(\tilde{q}_t = s|\rho,\pi,t)L(s)\,ds}\\ \geq&\frac{1}{N}\sum_{t=0}^{N-1}\ln\frac{\mathbb{E}_{\tilde{q}_{t+1}} L(\tilde{q}_{t+1})}{\mathbb{E}_{\tilde{q}_t} L(\tilde{q}_t)}\end{aligned} \tag{15}$$

Substituting the above into (11), we obtain

$$\lim_{N\to+\infty}\frac{1}{N}\sum_{t=0}^{N-1}\ln\frac{\mathbb{E}_{\tilde{q}_{t+1}} L(\tilde{q}_{t+1})}{\mathbb{E}_{\tilde{q}_t} L(\tilde{q}_t)} \leq -\alpha_2 \tag{16}$$

then

$$\lim_{N\to+\infty}\frac{1}{N}\ln\frac{\mathbb{E}_{\tilde{q}_N} L(\tilde{q}_N)}{\mathbb{E}_{\tilde{q}_0} L(\tilde{q}_0)} \leq -\alpha_2 \tag{17}$$

It means that $\forall\epsilon > 0, \exists N_\epsilon$, $\frac{1}{N}\ln\frac{\mathbb{E}_{\tilde{q}_N} L(\tilde{q}_N)}{\mathbb{E}_{\tilde{q}_0} L(\tilde{q}_0)} < -\alpha_2 + \epsilon < 0$ holds when $N > N_\epsilon$, namely

$$\frac{\mathbb{E}_{\tilde{q}_N} L(\tilde{q}_N)}{\mathbb{E}_{\tilde{q}_0} L(\tilde{q}_0)} \leq \mathrm{e}^{N(-\alpha_2+\epsilon)}, \forall N > N_\epsilon \tag{18}$$

So we get for sufficiently large $N > N_\epsilon$,

$$\mathbb{E}_{\tilde{q}_N\sim P(\tilde{q}_N|\rho,\pi,N)} L(\tilde{q}_N) \leq \mathrm{e}^{N(-\alpha_2+\epsilon)}\mathbb{E}_{\tilde{q}_0\sim\rho(\tilde{q}_0)} L(\tilde{q}_0) \tag{19}$$

By Equation (10) we have the following result

$$\begin{aligned}&\mathbb{E}_{\tilde{q}_N\sim P(\tilde{q}_N|\rho,\pi,N)}\mathbb{E}_\pi\|\tilde{q}_N\|^2\\ \leq&\mathrm{e}^{N(-\alpha_2+\epsilon)}\mathbb{E}_{\tilde{q}_0\sim\rho(\tilde{q}_0)}\mathbb{E}_\pi\|\tilde{q}_0\|^2 + \frac{\delta}{\alpha_1}\end{aligned} \tag{20}$$

So far, it has been proved that the estimate error $\tilde{q}_t$ in (7) is exponentially bounded according to Definition 1.

IV. Lyapunov-based Reinforcement Learning Orientation Estimation Algorithm

In this section, we will combine SAC algorithm [17], one of the state-of-the-art RL algorithms, with the theoretical result in Section III to learn the gain/policy K_{t+1} for the state estimator (3).

Considering MDP in (7), the orientation estimation problem can be viewed as a RL problem in which the policy is sought after by minimising the expected accumulated cost. Here a stochastic policy is chosen as $\pi(K_{t+1} \mid \tilde{q}_t) \sim \mathcal{N}(K_{t+1}(\tilde{q}_t), \sigma)$ from which the gain K_{t+1} for a given state $\tilde{q}_t$ is sampled [24]. The corresponding Q-function (a.k.a, state-action value function) is given as:

$$Q_\pi(\tilde{q}_t, K_{t+1}) = C_t(\tilde{q}_t, K_{t+1}) + \gamma \mathbb{E}_{\tilde{q}_{t+1}}[V_\pi(\tilde{q}_{t+1})] \quad (21)$$

To this end, K_{t+1} can be learned by many existing RL algorithms.

Motivated by the works in [20], [25], [26], [27], we propose to incorporate the theoretical result in Theorem 1 to policy optimisation in SAC as a constrained optimisation problem. First of all, a Lyapunov candidate needs to be selected at the first instance. In the context of RL, a Lyapunov candidate will be parametrised/selected as the Q-function [19], [28]. In this paper, we choose $L(\tilde{q}_t)$ in (10) as:

$$L_(\tilde{q}_t) = \mathbb{E}_{K_{t+1}\sim\pi}[L_c(\tilde{q}_t, K_{t+1})] \quad (22)$$

where $L_c(\tilde{q}_t, K_{t+1}) = Q(\tilde{q}_t, K_{t+1})$. Then the constrained optimisation problem is:

$$\begin{aligned} \min_{\pi} \quad & Q_\pi(\tilde{q}_t, K_{t+1}) \\ \text{s.t.} \quad & (10) \text{ and } (11) \\ & -\ln(\pi(K_{t+1} \mid \tilde{q}_t)) \geq \mathcal{H}_t \end{aligned} \quad (23)$$

where $Q_\pi(\tilde{q}_t, K_{t+1})$ is defined in (21), the second constraint is the minimum entropy constraint used in the SAC to improve the exploration in the action space [29] where $\mathcal{H}_t$ is the desired bound.

Denote the parametrised actor and critic as $\pi_\theta(K_{t+1}|\tilde{q}_t)$ and $Q_\phi(\tilde{q}_t, K_{t+1})$ respectively, where θ and ϕ are the parameters of the DNNs. To ensure the positiveness of a Lyapunov function, $L_\phi(\tilde{q}_t, K_{t+1})$ is selected as the square of a DNN as $L_\phi(\tilde{q}_t, K_{t+1}) = f_\phi^\top(\tilde{q}_t, K_{t+1}) f_\phi(\tilde{q}_t, K_{t+1})$, where f is the vector output of a DNN parameterised by ϕ. On the other hand, the stochastic policy $\pi_\theta(K_{t+1}|\tilde{q}_t)$ is parametrised by a DNN f_θ that depends on the state $\tilde{q}_t$ and a Gaussian noise ϵ.

Solving the above constrained optimisation problem is equivalent to minimising the following objective function:

$$\begin{aligned} J(\theta) = \mathbb{E}_{\tilde{q}_t, a_t, \tilde{q}_{t+1}, c_t \sim D}[\alpha(\ln(\pi_\theta(f_\theta(\tilde{q}_t, \epsilon)|\tilde{q}_t)) + \mathcal{H}_t) \\ + \lambda\big(\ln L_\phi(\tilde{q}_{t+1}, f_\theta(\tilde{q}_{t+1}, \epsilon)) - \ln L(\tilde{q}_t, a_t) + \alpha_2\big)] \end{aligned} \quad (24)$$

where $\mathcal{D}$ is the replay memory of the training samples, α and λ are Lagrange multipliers which control the relative importance of constraints in (23).

In the actor-critic framework, the policy network parameters are updated through stochastic gradient descent of (24). The training process can be seen in Fig. 1. It can be proved that the policy can converge to an optimal one that ensures the orientation estimate error $\mathbb{E}[\|\tilde{q}_t\|^2]$ converges to a constant as $t \to \infty, \forall \tilde{q}_t \in S$. The proof is standard and omitted due to page limits. The readers can refer to Section IV-D in [25] for more details. The pseudocode of the proposed Lyapunov-based reinforcement learning orientation estimation (LRLOE) algorithm is showed in Algorithm 1.

TABLE I
Hyperparameters of the proposed estimator

Hyperparameters	Value
Time horizon	1000
SGD batch size	256
Actor learning rate	1e-4
Critic learning rate	3e-4
Lyapunov learning rate	3e-4
Soft replacement (τ)	5e-3
Discount (γ)	0.999
Structure of a_ϕ	(128,64,32)
Structure of L_θ	(128,64,32)

Algorithm 1 LRLOE algorithm

1: Set the initial parameters ϕ for the Lyapunov function $\mathcal{L}_\phi$, θ for the estimator gain policy $\pi_\theta(K_1|\tilde{q}_0)$, λ for the Lagrangian multiplier, α for the temperature parameter, and the replay memory $\mathcal{D}$
2: Set the target parameter $\bar{\theta}$ as $\bar{\theta} \leftarrow \theta$
3: **while** Training **do**
4: **for** each data collection step **do**
5: Choose estimator gain K_{k+1} using $\pi_\theta(K_{k+1}|\tilde{q}_k)$
6: Simulate (1) and (2) with the orientation estimator (3) to collect samples $\tilde{q}_k$
7: $\mathcal{D} \leftarrow \mathcal{D} \cup \tilde{q}_k$
8: **end for**
9: update L_ϕ, π_θ, λ, α by optimising (24)
10: **end while**
11: Output θ^*, ϕ^*, λ^*, and α^*

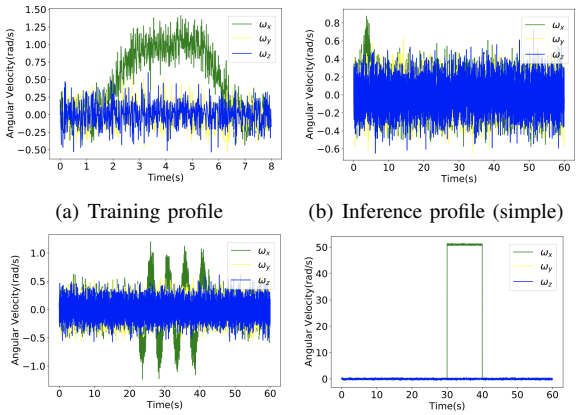
V. Experimental Results

In this Section, we train and infer on both simulated and real datasets. The estimation results has exhibited good performance, compared with three well-known orientation estimation algorithms: EKF [5], UKF [10], and complementary filter [6].

A. Results for simulated dataset

The RL policy is trained on a relatively trivial profile (see figure. 2(a)), then tested on three other independent profiles (see figure. 2(b), 2(c) and 2(d)). The sampling rate is set to 100 Hz (consistent with real data in Section V-B). The sensor noise are sampled with the following distribution [3]:

$$\begin{aligned} e_{\omega,t} &\sim \mathcal{N}(0, \Sigma_\omega), & \Sigma_\omega &= 0.0003, \\ e_{\mathrm{a},t} &\sim \mathcal{N}(0, \Sigma_\mathrm{a}), & \Sigma_\mathrm{a} &= 0.0005, \\ e_{\mathrm{m},t} &\sim \mathcal{N}(0, \Sigma_\mathrm{m}), & \Sigma_\mathrm{m} &= 0.0003 \end{aligned} \quad (25)$$



(a) Training profile (b) Inference profile (simple)

(c) Inference profile (medium) (d) Inference profile (complicated)

Fig. 2. The angular velocity profiles used for training and inference.

During inference, the covariance of the measurement noise Σ_ω is increased to 0.03. The hyperparameters of Algorithm 1 are shown in Table I. The last 20% of training data is used for validation. We independently train 20 policies and select the one with the lowest validation error for inference. The initial state estimate is sampled from a normal distribution with actual initial orientation as mean and a standard deviation of 0.1. The inference results of our proposed method are shown in Fig. 3, for all three different angular velocity profiles with accurate estimation.

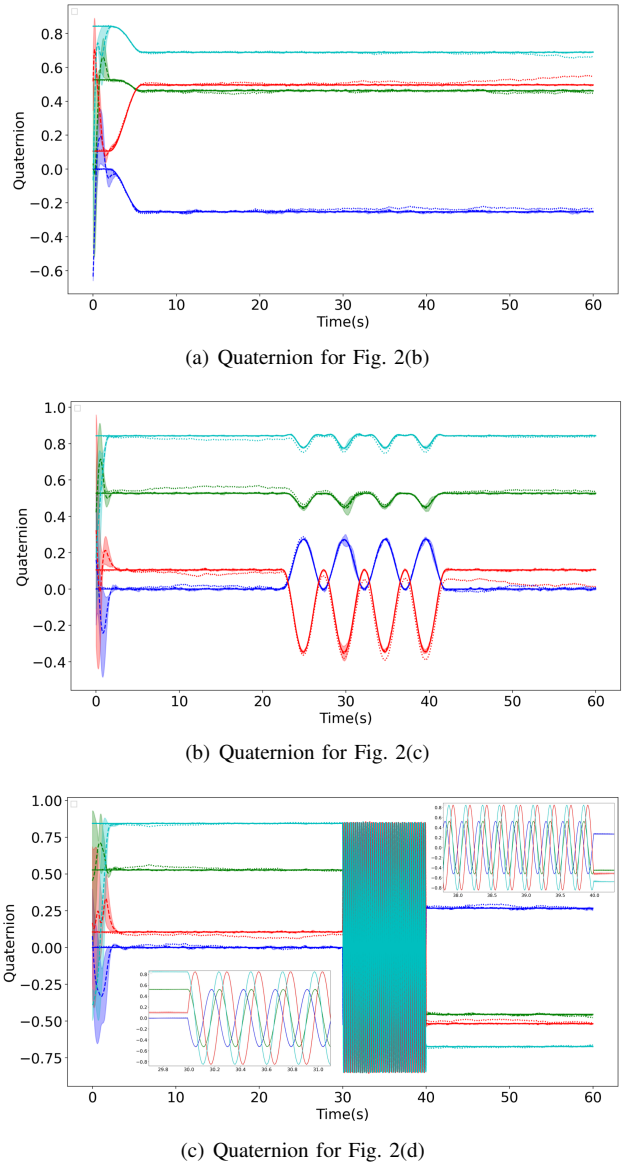
Furthermore, we compare our proposed DRL-based estimation method with the EKF [5], the UKF [10], and the complementary filter (CF) [6]. The root of mean square errors (RMSE) of the decoupled Euler parameters for pitch, ϕ, roll, θ and heading, ψ angles, corresponding to rotations around x, y, z-axis respectively, is chosen as the estimation performance and 200 Monte Carlo simulations are run for each algorithm. As shown in Fig. 4, the DRL-based estimation method can achieve good performance using the inference profiles in Fig. 2(c).

For a drastic angular velocity profile, the orientation estimation results of the EKF, UKF and CF can be challenging, as shown in Fig. 5. It can be found that the UKF yields a significant estimate error under a high noise level, and both the EKF and complementary filter perform poorly as the estimate error accumulates in the long period of rapid rotation. In comparison, our proposed DRL-based estimation method is more robust for "wild" movement profiles.

B. Results for real dataset

We apply our algorithm for real dataset from [3] (see Fig. 6). The data is collected from the Trivisio Colibri Wireless IMU [30] with a logging rate of 100Hz. The reference measurement of the orientation is provided as ground truth from a motion capture equipment [31] by tracking the optimal markers fixed to the sensor platform. The optical and IMU data has been time-synchronised and aligned beforehand.

The dataset is 100 seconds long and split into training and inference dataset separately. The first half of the collected data is used for training and the rest for inference. In the training dataset, we randomly selected a consecutive sequence



(a) Quaternion for Fig. 2(b)

(b) Quaternion for Fig. 2(c)

(c) Quaternion for Fig. 2(d)

Fig. 3. Quaternion for different angular velocities in Fig. 2(b), 2(c) and 2(d). The solid, dotted and dashed lines correspond to ground truth, integrated and estimated quaternion, respectively. The shaded areas correspond to the standard deviation of over 20 independent runs. For the high-velocity profile, a zoom-in for the high angular velocity period is also showed in Fig. 3(c).

of a length of 1000 samples as a training episode. We test in two scenarios (see Fig. 6): (1) inference including the training dataset from $t = 0$ where the initial estimation $\hat{q}^{\text{nb}}_{t=0s}$ is simply the measurement; (2) inference without training dataset from $t = 50s$ where $\hat{q}^{\text{nb}}_{t=50s}$ is normally distributed around the true initial orientation with a standard deviation of 0.01. 50 independent trials are performed. The estimation results are showed in Fig. 7.

In the second scenario, the second half of the dataset is also tested on the EKF, UKF and complementary filter methods. The estimation results are quantified and summarised in Table II. Since the initial estimations and the gyroscope measurements are relatively accurate, results indicate that our proposed algorithm has achieved comparable performance

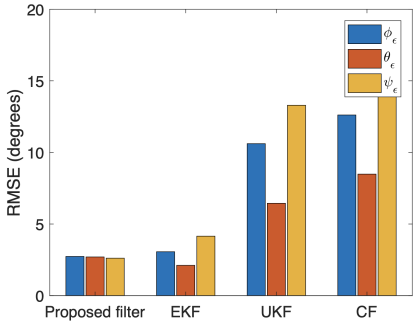
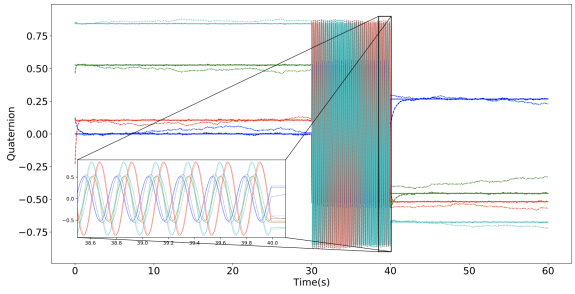
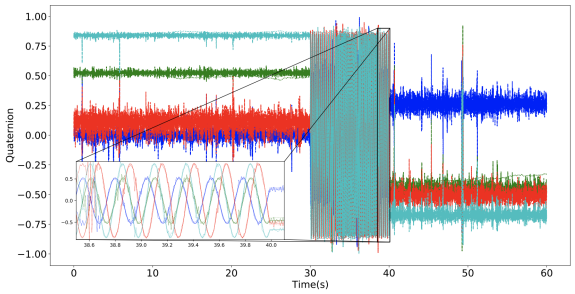


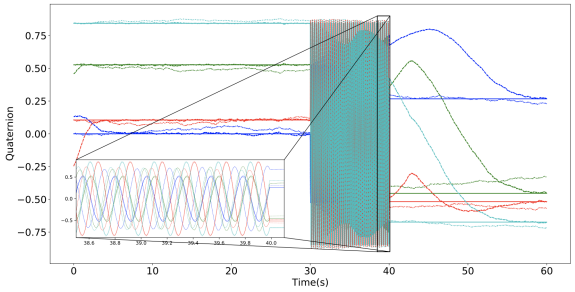
Fig. 4. The RMSE of the proposed filter, Extended Kalman filter (EKF), Unscented Kalman filter (UKF) and Complementary filter (CF) for the inference profiles in Fig. 2(c), 2(c) and 2(d). Because the quaternions double-cover the rotation space which imposes ambiguity in comparison, we use Euler angles instead of quaternions here.



(a) EKF based quaternion estimation for Fig.2(d)



(b) UKF based quaternion estimation for Fig.2(d)



(c) CF based quaternion estimation in Fig.2(d)

Fig. 5. Comparison with other classic estimation algorithms for a drastic angular velocity profile in Fig. 2(d).

with the other three state-of-art algorithms.

VI. Conclusion

Orientation estimation using inertial sensors combined with a magnetometer is well studied, and many algorithms have been proposed. However, there hardly exist any algorithms

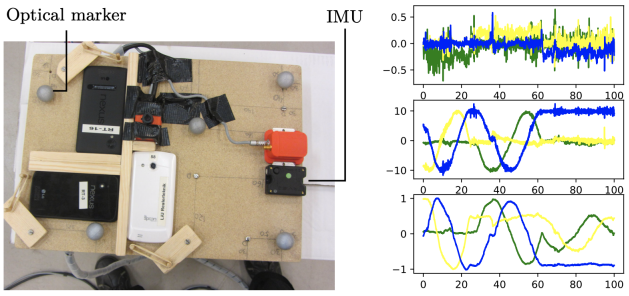


Fig. 6. Real dataset (adapted from Fig. 4.2 and 4.3 in [3]). Left: A snapshot of the platform for collecting real dataset Right: Measurements from an accelerometer ($y_{\mathrm{a},t}$, top), a gyroscope ($y_{\omega,t}$, middle) and a magnetometer ($y_{\mathrm{m},t}$, bottom) for 100 seconds of data collected with the IMU showed in the left figure.).

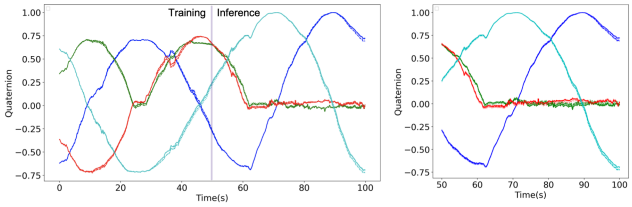


Fig. 7. Quaternion for real data. The solid, dotted and dashed lines correspond to ground truth, integrated and estimated quaternion respectively. The left and right figures are corresponding to scenarios 1 and 2 respectively. In the right figure, the dashed line is obtained by averaging over 50 independent trails.

TABLE II
RMSE of the orientation estimates

RMSE	Yaw[°]	Pitch[°]	Roll[°]
Proposed algorithm	1.9423	2.1048	0.8353
Extended Kalman Filter	2.0411	1.5272	1.2488
Unscented Kalman Filter	20.1370	20.3494	38.6775
Complementary Filter	1.2015	1.3892	0.8972

with theoretical guarantees of estimation convergence. This paper proposes a reinforcement learning-based orientation estimation method and proves that its estimate error converges exponentially to a positive bound. The proposed method shows superior estimation performance compared with some well-known ones in terms of arbitrary estimation initialisation and adaptation to a drastic angular velocity profile.

VII. Acknowledgment

We thank Rick Staa (TU Delft) for implementing the Algorithm 1 in TensorFlow [32]. We are grateful for the help and equipment provided by the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division at the Department of Computer and Information Science, Link ”oping University, Sweden. We thank Gustaf Hendeby, Niklas Wahlstr ”om, Hanna Nyqvist and Manon Kok who collected the real data and allow us to use. This work is supported by Huawei, AnKobot and China Scholarship Council (No.202006890020).

REFERENCES

[1] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

[2] H. Zhou and H. Hu, "Human motion tracking for rehabilitation—a survey," *Biomedical signal processing and control*, vol. 3, no. 1, pp. 1–18, 2008.

[3] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Foundations and Trends in Signal Processing*, vol. 11, no. 1-2, pp. 1–153, 2017.

[4] 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.

[5] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, "An extended kalman filter for quaternion-based orientation estimation using marg sensors," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS).*, vol. 4. IEEE, 2001, pp. 2003–2011.

[6] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for imus and margs," *Sensors*, vol. 15, no. 8, pp. 19 302–19 330, 2015.

[7] S. Wang, H. Wen, R. Clark, and N. Trigoni, "Keyframe based large-scale indoor localisation using geomagnetic field and motion pattern," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 1910–1917.

[8] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "Vinet: visual-inertial odometry as a sequence-to-sequence learning problem," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 3995–4001.

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

[10] E. Kraft, "A quaternion-based unscented kalman filter for orientation tracking," in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 1, no. 1. IEEE Cairns, Queensland, Australia, 2003, pp. 47–54.

[11] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE international conference on rehabilitation robotics*. IEEE, 2011, pp. 1–7.

[12] M. Kok, J. Hol, and T. Schön, "An optimization-based approach to human body motion capture using inertial sensors," in *19th World Congress of the International Federation of Automatic Control (IFAC), Cape Town, South Africa, August 24-29, 2014*. International Federation of Automatic Control, 2014, pp. 79–85.

[13] M. Brossard, S. Bonnabel, and A. Barrau, "Denoising imu gyroscopes with deep learning for open-loop attitude estimation," *arXiv preprint arXiv:2002.10718*, 2020.

[14] J. Morimoto and K. Doya, "Reinforcement learning state estimator," *Neural Computation*, vol. 19, no. 3, pp. 730–756, 2007.

[15] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.

[18] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[19] V. Petridis and S. Petridis, "Construction of neural network based lyapunov functions," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 5059–5065.

[20] M. Han, L. Zhang, J. Wang, and W. Pan, "Actor-critic reinforcement learning for control with stability guarantee," *IEEE Robotics and Automation Letters (RA-L & IROS), accepted and in press*, 2020. [Online]. Available: arXiv:2004.14288

[21] 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.

[22] J. Sola, "Quaternion kinematics for the error-state kf," *Laboratoire dAnalyse et dArchitecture des Systemes-Centre national de la recherche scientifique (LAAS-CNRS), Toulouse, France, Tech. Rep*, 2012.

[23] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen, "Stochastic stability of the discrete-time extended kalman filter," *IEEE Transactions on Automatic Control*, vol. 44, no. 4, pp. 714–728, 1999.

[24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introductions*, 2nd ed. The MIT Press, 2018.

[25] L. Hu, C. Wu, and W. Pan, "Lyapunov-based reinforcement learning state estimator," *arXiv preprint arXiv:2010.13529*, 2020.

[26] Q. Zhang, W. Pan, and V. Reppa, "Model-reference reinforcement learning control of autonomous surface vehicles," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 5291–5296.

[27] ——, "Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles," *arXiv preprint arXiv:2008.07240*, 2020.

[28] T. J. Perkins and A. G. Barto, "Lyapunov design for safe reinforcement learning," *Journal of Machine Learning Research*, vol. 3, no. Dec, pp. 803–832, 2002.

[29] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, vol. 80, Stockholmsmässan, Stockholm Sweden, Jul. 2018, pp. 1861–1870.

[30] "Trivisio prototyping gmbh," http://www.trivisio.com, 2016.

[31] "Vicon," url:http://www.vicon.com, Accessed on August 5., 2016.

[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.