Maximum likelihood estimation of the extended Kalman filter's parameters with natural gradient

Colin Parellier, Camille Chapdelaine, Axel Barrau, Silvère Bonnabel

Maximum likelihood estimation of the extended Kalman filter's parameters with natural gradient

Colin Parellier, Camille Chapdelaine, Axel Barrau and Silvère Bonnabel

***Abstract*— The extended Kalman filter (EKF) relies on noise parameters, notably the covariance matrix of the observation noise. To identify them using real data, the standard approach consists in maximizing the likelihood of the EKF's estimates. To perform the optimization, we propose in this paper to use Amari's natural gradient descent, in a way that preserves positive semi-definiteness of the covariance parameter. We derive the corresponding equations, and we bring the method to bear on a real-world experiment, where we identify the covariance matrix of a GNSS for a vehicle localization problem.**

***Index Terms*— State estimation, Kalman filter, Information geometry, Identification.**

I. Introduction

The Kalman filter (KF), or its extended version (EKF), is pervasive in state estimation. One recurrent engineering problem, though, is that of tuning the EKF's multiple parameters. As the EKF's performance critically depends on those parameters in practice, they require a great deal of manual "tweaking", and engineering know-how [1], [26]. Hence, automatic tuning of the KF (or of the EKF), where the unknown parameters are optimized (in other words "learned") from data so as to maximize filter's predictions, dates back to the early days of filtering. The optimization procedure relies on gradient ascent of a likelihood function, where the gradient is computed through the sensitivity equations [16], see also [27]. Such gradient descent may lead to instability that various methods have seeked to combat, see [18], [25]. Derivative-free alternatives were also proposed, using the expectation-maximization (EM) algorithm [9], [24].

In this paper, we consider the problem of gradient descent based maximum likelihood estimation of the noise covariance matrix R used by an EKF. We leverage a special type of gradient descent, namely the "natural gradient" of Amari [4] from information geometry [3], [20]. Natural gradient has recently proved very powerful for machine learning [19] and for state estimation in robotics when estimating covariance matrices, see [5], [6]. We show that it leads to a well-behaved gradient descent algorithm, as compared to constant step size tuning. However, the problem with (natural) gradient descent with respect to a positive semi-definite covariance matrix R is that the updated matrix at each step may lose its symmetry, and the eigenvalues may become negative, which is meaningless. A way to enforce symmetry and positive semi-definiteness is to work with a "square-root" factorization $R = LL^T$ and to perform natural gradient on factor L instead. This then requires to derive the natural gradient w.r.t. L, which is our first main contribution.

To assess the method, we consider an application to accurate localization of a vehicle, based on the fusion of an inertial measurement unit (IMU) and a GPS. Our goal is to learn the covariance of the GPS sensor, so as to maximize the likelihood associated with the state estimates. The dynamical model involves a 18-dimensional state space, and the learning problem involves a 73 minutes long trajectory that spans a 5 km-wide loop. The experiments are similar to those of [1], but with a significant upscaling: they use a 5-dimensional state space to model a 2D vehicle performing a 15 meters wide loop, while we use a 18-dimensional state space for a 3D vehicle performing a 5 km wide loop. The covariance matrix encoding the GPS's uncertainty is generally not known accurately, and is learned from data by maximizing the log-likelihood. Because of the size of the problem, we utilize our recent closed-form sensitivity equations, see [21], that lead to drastic speed-ups of the gradient computation, rendering the method tractable in spite of the length of the data sequence. This application, based on our own unpublished real-world experiments with an actual car, is our second main contribution.

The paper is organized as follows. Section II presents the proposed method for learning the noise covariance parameter. Section III presents the extended Kalman filter (EKF) used for sensor fusion, and the localization problem to which our learning method is applied. Section IV reports on our own experimental results.

II. Maximum likelihood estimate for the Kalman filter parameters with natural gradient descent

In this paper, we consider a system

$$x_{n+1} = f(x_n, u_n) + w_n, \quad y_n = h(x_n) + v_n,$$

where x_n is the state, u_n the control input, $w_n \sim \mathcal{N}(0, Q)$ a white noise, and where y_n denote noisy observations, with $v_n \sim \mathcal{N}(0, R)$ a noise.

An extended Kalman filter (EKF) may be devised for this system. When doing so, the covariance matrices Q, R are considered as tuning parameters. In practice, they are hard to tune manually, and can be learned using data. In the sequel,

This work is supported by CIFRE Grant 2019/1974 from French Agence Nationale de la Recherche et de la Technologie (ANRT).

C. Parellier and S. Bonnabel are with Mines Paris - PSL, PSL Research University, Centre for Robotics, 60 Bd Saint-Michel, 75006 Paris, France. firstname.lastname@minesparis.psl.eu

C. Parellier and C. Chapdelaine are with Safran Tech, Groupe Safran, Rue des Jeunes Bois-Chateaufort, 78772, Magny Les Hameaux, France. firstname.lastname@safrangroup.com

A. Barrau is with Offroad, 5 Rue Charles de Gaulle, 94140 Alfortville, France.

however, we only focus on the noise covariance parameter, namely matrix R.

A. Maximum likelihood parameter learning

A longstanding approach to learning R is by maximum likelihood over a sequence of data. Indeed, one may compute the negative log-likelihood (NLL) associated to a sequence of measurements $y_1, \dots, y_N$, given noise covariance R:

$$\mathscr{L} = -\log p\left(y_1, \dots, y_N | R\right). \tag{1}$$

The gradient of the scalar loss function $\mathscr{L}$ with respect to $R \in \mathscr{M}_p(\mathbb{R})$ is a $p \times p$ matrix $\frac{\partial \mathscr{L}}{\partial R}$ whose ij's entry is the partial derivative of $\mathscr{L}$ w.r.t. R_{ij}. A simple method to automatically tune R is then through gradient descent, that is, one iterates over

$$R \leftarrow R - \eta \frac{\partial \mathscr{L}}{\partial R}, \tag{2}$$

with η the step size, to be tuned by the user.

B. Natural gradient descent method

Gradient descent (2) with a fixed scalar step size η is a first simple method to minimize the NLL. However, this fixed step size is often complicated to tune, given the cost function is complicated and nonconvex. To improve it, one could think of implementing line search. However, evaluating the cost function $\mathscr{L}$ is computationally demanding, as it is based on the EKF's estimates and thus requires running an EKF over the entire data sequence (the NLL expression is deferred to Section III, see (18), not to interrupt to flow of reading). In the same way, one could think of using Newton's method to help conditioning the gradient with the inverse of the Hessian. However, computing the second order derivatives seems very costly, numerically. Moreover we do not deal with a vector parameter, but with a matrix parameter, making the Hessian a complicated object. For those reasons, we propose in this work to use the method of natural gradient from information geometry [4], where a Riemannian metric "distorts" the parameter space and renders the gradient much more adaptive (notably it becomes invariant to a change of units). Natural gradient is very effective, but often proves too difficult or too costly to compute [19].

Natural gradient requires deriving the Fisher metric associated to the NLL. This seems a difficult task, though, owing to the complexity of the NLL in Kalman filtering. In this paper, we focus on the natural gradient associated to the family of Gaussian distributions as a proxy.

C. Fisher Information Metric (FIM) and natural gradient for multivariate Gaussians

Let us consider the set of centered multivariate Gaussians $\mathscr{N}(0,R)$. They are parameterized by their covariance matrix $R \in \mathbb{R}^{p \times p}$.

Proposition 1: The FIM associated to the family of centered multivariate Gaussian distributions $\mathscr{N}(0,R)$ writes $\langle \delta R, \delta R \rangle_R = Tr(R^{-1}\delta R R^{-1}\delta R)$.

Proof: We start from the Kullback-Leibler (KL) divergence. It can be considered as a definition of the FIM that $KL(\mathscr{N}(0,R+\delta R)||\mathscr{N}(0,R)) = \frac{1}{2}\langle \delta R, \delta R \rangle_R + O(||\delta R||^3)$.

On the other hand, we may compute explicitly the KL between Gaussian distributions and we find that

$$\begin{aligned} &KL(\mathscr{N}(0,R+\delta R)||\mathscr{N}(0,R)) \\ &= \frac{1}{2} Tr(R^{-1}(R+\delta R)) + \frac{1}{2}\log|R| - \frac{1}{2}\log|R+\delta R| - \frac{d}{2}. \end{aligned} \tag{3}$$

The proof is concluded using that $\log|R+\delta R| = \log|R| + Tr(R^{-1}\delta R) + O(||\delta R||)^2$, which allows for computing the gradient of $R \mapsto \log|R|$, and in turn using $(R+\delta R)^{-1} = R^{-1} - R^{-1}\delta R R^{-1} + O(||\delta R||)^2$, which allows for computing the Jacobian of this gradient, yielding the Hessian we are looking for. ∎

Proposition 2: The natural gradient of a scalar objective function $\mathscr{L}$ associated with the family of centered Gaussian distributions writes

$$\tilde{\nabla}\mathscr{L}(R) = R\frac{\partial \mathscr{L}}{\partial R}R. \tag{4}$$

Proof: The natural gradient is defined as the Riemannian gradient, see [11], [20], of the function in the sense of the FIM. The scalar product at R associated to the FIM reads

$$\langle \delta R_1, \delta R_2 \rangle_R = Tr(R^{-1}\delta R_1 R^{-1}\delta R_2). \tag{5}$$

We then write for a scalar function $\mathscr{L}(R+\delta R) = \mathscr{L}(R) + \langle \tilde{\nabla}\mathscr{L}(R), \delta R \rangle_R + O(||\delta R||^2)$ where $\tilde{\nabla}\mathscr{L}(R) \in \mathscr{M}_d(\mathbb{R})$ denotes the natural gradient. Using the standard Euclidean scalar product for matrices $\langle A, B \rangle := Tr(A^T B)$ the first order term also writes $\langle \frac{\partial \mathscr{L}}{\partial R}, \delta R \rangle$, proving the result by identification as $Tr(R^{-1}\tilde{\nabla}\mathscr{L}(R)R^{-1}\delta R) = Tr(\frac{\partial \mathscr{L}}{\partial R}\delta R)$ for all symmetric δR. ∎

Albeit well-known, see e.g. [5], the latter proof was included for completeness, especially in view of results to come.

The natural gradient algorithm then consists in replacing the standard gradient in (2) with its "natural" counterpart

$$R \leftarrow R - \eta\tilde{\nabla}\mathscr{L}(R) = R - \eta R\frac{\partial \mathscr{L}}{\partial R}R. \tag{6}$$

We immediately observe that, as $\mathscr{L}$ is unitless, (6) is invariant to a change of units $R \mapsto sR$, contrary to (2). A suitable stepsize tuning η for R expressed in, e.g., meters, will work equally if R is expressed in kilometers, which makes the method more adaptive.

D. Natural gradient in factorized form LL^T

For the covariance parameter output by the optimization algorithm to make sense, one must maintain the covariance matrix parameter R positive semi-definite (PSD) over the descent procedure. However, for too large a stepsize, (6) may become negative (but small step sizes lead to slow convergence). Following a standard approach [10], we write $R = LL^T$ with L a "square-root" matrix of the same size as R. By maintaining $L \in \mathscr{M}_d(\mathbb{R})$ instead of R, one does not have to numerically maintain positive semi-definiteness and symmetry of R in the gradient descent, since it is inherent to the decomposition. We hence propose to derive a *novel*

natural gradient in factorized form, to enforce positive semi-definiteness. Our result is as follows.

Proposition 3: The natural gradient of a function $\mathscr{L}$ associated with the density of Gaussian distributions $\mathscr{N}(0,R)$ parameterized in factorized form $R = LL^T$, reads :

$$\tilde{\nabla}\mathscr{L}(L) = \frac{1}{2}LL^T\frac{\partial \mathscr{L}}{\partial R}L = \frac{1}{2}R\frac{\partial \mathscr{L}}{\partial R}L. \tag{7}$$

Proof: The first step is to derive the FIM in this form.

Lemma 1: The FIM associated with the family $\mathscr{N}(0,LL^T)$ parameterized by L writes $\langle \delta L, \delta L\rangle_L = 2Tr(L^{-1}\delta L L^{-1}\delta L + (\delta L)^T L^{-T}L^{-1}\delta L)$.

Proof: We may compute the FIM with respect to parameter L associated with the family $\mathscr{N}(0,LL^T)$. From Proposition 1, the second-order expansion of the KL writes

$$\begin{aligned} &Tr(R^{-1}\delta R R^{-1}\delta R) \\ &= Tr\left((LL^T)^{-1}(\delta L L^T + L\delta L^T)(LL^T)^{-1})(\delta L L^T + L\delta L^T)\right) \\ &= 2Tr(L^{-1}\delta L L^{-1}\delta L + (\delta L)^T L^{-T}L^{-1}\delta L). \end{aligned}$$

This corresponds to the scalar product $\langle \delta L, \delta J\rangle_L = 2Tr(L^{-1}\delta L L^{-1}\delta J) + 2Tr\left((\delta L)^T(LL^T)^{-1}\delta J\right)$. ∎

Let us now conclude the proof. To retrieve the natural gradient we need to equate the first-order expansions, that is, $\langle \tilde{\nabla}\mathscr{L}, \delta L\rangle_L = \langle \frac{\partial \mathscr{L}}{\partial L}, \delta L\rangle$ where the latter denotes the standard Euclidean scalar product for matrices $\langle A,B\rangle := Tr(A^TB)$. We thus need to find the natural gradient at L, denoted by $\tilde{\nabla}\mathscr{L}(L)$ here, that ensures we have for all δL the following

$$\begin{aligned} &2Tr(L^{-1}(\tilde{\nabla}\mathscr{L})L^{-1}\delta L) + 2Tr\left((\delta L)^T(LL^T)^{-1}\tilde{\nabla}\mathscr{L}\right) \\ &= Tr\left((\delta L)^T\frac{\partial \mathscr{L}}{\partial L}\right). \end{aligned}$$

This is equivalent to

$$L^{-1}\tilde{\nabla}\mathscr{L} + (L^{-1}\tilde{\nabla}\mathscr{L})^T = \frac{1}{2}L^T\frac{\partial \mathscr{L}}{\partial L}.$$

Now if we momentarily admit that for the Euclidean gradient $\frac{\partial \mathscr{L}}{\partial L} = 2\frac{\partial \mathscr{L}}{\partial R}L$, we have

$$L^{-1}\tilde{\nabla}\mathscr{L} + (L^{-1}\tilde{\nabla}\mathscr{L})^T = L^T\frac{\partial \mathscr{L}}{\partial R}L.$$

As at each gradient step the gradient needs to be symmetric to preserve symmetry of the parameter (otherwise we may symmetrize it), we may assume $\frac{\partial \mathscr{L}}{\partial R}$ to be symmetric. As the antisymmetric part of $L^{-1}\tilde{\nabla}\mathscr{L}$ does not play any role in the equation, we may fix it by imposing that $L^{-1}\tilde{\nabla}\mathscr{L}$ be symmetric, which yields (7).

To prove the remaining unproved formula, we write $\mathscr{L}\left((L+\delta L)(L+\delta L)^T\right) \approx \mathscr{L}(LL^T) + \mathrm{tr}\left((\frac{\partial \mathscr{L}}{\partial R})^T(L\delta L^T + \delta L L^T)\right) = \mathscr{L}(LL^T) + 2\mathrm{tr}\left((\frac{\partial \mathscr{L}}{\partial R}L)^T\delta L\right)$, using that $\frac{\partial \mathscr{L}}{\partial R}$ is symmetric. Denoting $\mathscr{L}(R) = \bar{\mathscr{L}}(L)$ we also have $\bar{\mathscr{L}}(L+\delta L) \approx \bar{\mathscr{L}}(L) + \mathrm{tr}\left((\frac{\partial \bar{\mathscr{L}}}{\partial L})^T\delta L\right)$, and thus $\frac{\partial \bar{\mathscr{L}}}{\partial L} = 2\frac{\partial \mathscr{L}}{\partial R}L$. ∎

When working with the square-root factor L, the gradient descent using the Euclidean gradient reads

$$L \leftarrow L - \eta\frac{\partial \mathscr{L}}{\partial L}. \tag{8}$$

The natural gradient algorithm then consists in replacing the standard gradient in (8) with its natural counterpart, that is,

$$L \leftarrow L - \eta\tilde{\nabla}\mathscr{L}(L) = L - \eta\frac{1}{2}R\frac{\partial \mathscr{L}}{\partial R}L, \tag{9}$$

and to let $R = LL^T$ be the estimated covariance matrix.

III. Application to identification of GNSS uncertainty

To assess our novel gradient descent scheme for EKF parameter identification, we consider a vehicle equipped with an inertial measurement unit (IMU) and other sensors such as Global Positioning System (GPS or more generally GNSS). We denote the orientation of the vehicle by $\Omega_n \in SO(3)$, where $SO(3)$ is the special orthogonal group. Its inertial velocity is denoted by $\upsilon_n \in \mathbb{R}^3$, and its position by $p_n \in \mathbb{R}^3$.

A. Modelling of the IMU

The IMU measures rotation rates $\omega_n \in \mathbb{R}^3$, and accelerations $a_n \in \mathbb{R}^3$. We denote by $\Phi \in \mathbb{R}^3$ the Earth rotation vector and by δ_n the sampling period of the IMU. We further introduce the exponential operator $\exp : \mathbb{R}^3 \to SO(3)$, which maps each $v \in \mathbb{R}^3$ to matrix $\exp(v) = \exp_m((v)_\times)$, where $\exp_m$ denotes matrix exponential, and $(\cdot)_\times$ denotes Rodrigues operator which associates to $v \in \mathbb{R}^3$ the skew-symmetric matrix $(v)_\times$ such that, $\forall u \in \mathbb{R}^3$, $(v)_\times u = v \times u$, where $\times$ denotes cross product. The equations used for IMU modelling read

$$\begin{cases} \Omega_{n+1} &= \Gamma_n\Omega_n\exp\left[\delta_n(\omega_n - d_n - w_n^\omega)\right] \\ \upsilon_{n+1} &= \Gamma_n(\upsilon_n + \delta_n\Omega_n(a_n - b_n - w_n^a) + \delta_n g(p_n)) \\ p_{n+1} &= \Gamma_n(p_n + \delta_n\upsilon_n) \end{cases} \tag{10}$$

where $\Gamma_n = \exp(-\delta_n\Phi)$, and $g(p_n)$ is the gravity vector at p_n [15] [1]. In these equations, w_n^ω and w_n^a are white Gaussian noises with known variances σ_ω^2 and σ_a^2 respectively: $w_n^\omega \sim \mathscr{N}(0,\sigma_\omega^2 I)$,$w_n^a \sim \mathscr{N}(0,\sigma_a^2 I)$, where I is the identity matrix. Gyroscope and acceleration measurements have unknown biases, respectively denoted by $d_n \in \mathbb{R}^3$ and $b_n \in \mathbb{R}^3$ in Equation (10). The temporal evolution of these biases is described by

$$d_{n+1} = d_n + w_n^d, \quad b_{n+1} = b_n + w_n^b, \tag{11}$$

where w_n^d and w_n^b are white Gaussian noises with known variances σ_d^2 and σ_b^2 respectively: $w_n^d \sim \mathscr{N}(0,\sigma_d^2 I)$ and $w_n^b \sim \mathscr{N}(0,\sigma_b^2 I)$.

B. Sensor fusion with extended Kalman filter (EKF)

As IMUs lead to estimates that inevitably drift over time, they need to be aided by other sensors, through observations of the form

$$y_n = h(\chi_n) + v_n, \quad v_n \sim \mathscr{N}(0,R). \tag{12}$$

The measurements (12) read for the considered application

$$y_n = p_n + \Delta_n(\upsilon_n - \Phi\times p_n) + \Omega_n l_n + v_n \tag{13}$$

[1] To be consistent with our experiments, we use equations of aerospace engineering associated to an IMU that is accurate enough to measure the rotation of the Earth. The method also applies to cheaper IMUs, though.

where $v_n \sim \mathcal{N}(0,R)$. Here, $l_n \in \mathbb{R}^3$ is the unknown lever arm between the IMU frame and the GPS frame, and needs be estimated. Δ_n corresponds to the time lapse between the last IMU measurement and the considered GPS measurement. We assume that GPS data are not biased with respect to our model, so the noise v_n is considered as zero-mean.

The goal of sensor fusion (or observer design) is to estimate at each time step n the system's state:

$$\chi_n = (\Omega_n, \upsilon_n, p_n, d_n, b_n, l_n). \tag{14}$$

To estimate χ_n, we resort to an extended Kalman filter (EKF), that is still the state of the art in the navigation industry. To do so we rewrite Equations (10) and (11) in the general form:

$$\chi_{n+1} = f(\chi_n, u_n, w_n) \tag{15}$$

where $u_n^T = \left(\omega_n^T, a_n^T\right)$ and $w_n^T = \left(w_n^{\omega^T}, w_n^{a^T}, w_n^{d^T}, w_n^{b^T}\right)$ is a centered Gaussian noise. At each timestep, the EKF computes an estimation $\hat{\chi}_n$ of the system state χ_n and of the error covariance matrix P_n. The EKF consists in a prediction step alternated with an update step. In the prediction step, the variables of the system state χ_n are evolved through the noise-free dynamical model, given by propagation equation (15) and its linearization:

$$\begin{cases} \hat{\chi}_{n|n-1} & = f\left(\hat{\chi}_{n-1|n-1}, u_n, 0\right) \\ P_{n|n-1} & = F_n P_{n-1|n-1} F_n^T + G_n Q G_n^T \end{cases}. \tag{16}$$

where matrices F_n are the Jacobians of f with respect to χ_n (in the sense of a certain state error, see below), and G_n is the Jacobian of f with respect to w_n. In the update step, the state χ_n is corrected in the light of the measurements y_n:

$$\begin{aligned} S_n = H_n P_{n|n-1} H_n^T + R, \quad K_n = P_{n|n-1} H_n^T S_n^{-1}, \\ P_{n|n} = (I - K_n H_n) P_{n|n-1}, \quad \hat{\chi}_{n|n} = \hat{\chi}_{n|n-1} \oplus K_n z_n \end{aligned} \tag{17}$$

where z_n is the innovation $z_n = y_n - h(\hat{\chi}_n)$ and H_n denote the Jacobians of h with respect to χ_n. To update the state $\hat{\chi}_{n|n}$, the definition of the $\oplus$ operator depends on the choice of the error state vector. In the present paper, our choice for $\oplus$ is based on the Lie group $SE_2(3)$ that was introduced by the Invariant EKF (IEKF) theory of [7], and which was recently thoroughly treated in [13]. The IEKF finds its roots in the invariant observer theory [2], [12], and has led to various successes in the industry and in robotics, e.g., [8], [14], [17], [22]. An explicit description is given in Appendix A.

A standard computation (e.g., [1], [16]) shows that up to an additive constant the NLL writes

$$\mathscr{L} = \sum_{n=1}^{N} \log|S_n| + z_n^T S_n^{-1} z_n. \tag{18}$$

C. The parameters of the EKF

The EKF for sensor fusion detailed above depends on covariance matrices Q and R. The IMU (process) noise covariance Q related to σ_ω^2 σ_a^2, σ_d^2 and σ_b^2 is well-known from the characteristics of the IMU. On the contrary, the measurement covariance R of the GPS is hard to characterize, and is known with less accuracy, see [1], [26], since GPS errors are often correlated over time, whereas the EKF assumes that the errors are independent. Hence, the learned covariance R is not exactly that of the sensor: it is the parameter that maximizes the performance of the filter, in the sense of likelihood of data, and allows it to cope with unmodeled effects such as measurement correlation.

Fig. 1: Actual car used for the experiments

D. Gradient computation by backpropagation

The expression (18) of $\mathscr{L}$ depends on the noise parameter R in a complicated manner, through equations (16)-(17). The derivatives of $\mathscr{L}$ with respect to R may be obtained through the well-known sensitivity equations [16]. However, this comes at a heavy computational price: for each entry of the matrix R_{ij}, one needs to run the equivalent of a Kalman filter over the entire dataset, see Appendix A.3 of [23], to get each $\frac{\partial \mathscr{L}}{\partial R_{ij}}$. In [21], we very recently proposed an alternative method that requires the equivalent of running one Kalman filter over the dataset to compute the derivative with respect to the entire matrix, that is, $\frac{\partial \mathscr{L}}{\partial R}$, leading to drastic reduction in computation time. Owing to the length of the data sequence and the dimension of the state, we opt for this method to get the gradients, allowing for implementation of the proposed square-root natural gradient descent.

IV. Experimental results on real data

The method is tested on real data, acquired from a wheeled vehicle equipped with an IMU and a GPS. The vehicle used for the experiments is shown in Figure 1 and is an experimental vehicle of the company Safran, which participates in the present paper.

We estimate noise covariance square-root factor L such that $R = LL^T$ using the proposed method. The trajectory is displayed in Figure 2. This trajectory is 73 mn long and contains $N = 4428$ GPS measurements. The IMU signal is sampled at 100 Hz and the GPS at 1 Hz.

A. Assessing the EKF

We first evaluate the estimation of the trajectory performed by the EKF detailed in Section III, and compare it with a ground truth given by the IMU high precision commercial software, when using a coarse approximation of the covariance matrix: we take R to be the identity, which corresponds to a 1 meter standard deviation that matches specifications of the used GPS sensor. Table I shows the root-mean square error $RMSE =$

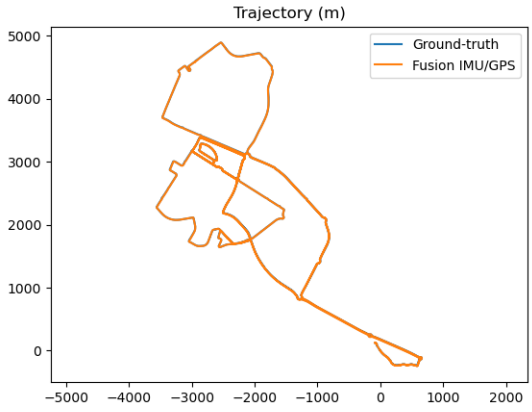


Fig. 2: Trajectory used when estimating L (in the form $R = LL^T$). The figure shows the trajectory estimated by the proposed EKF, as well as the ground truth. Data were acquired on a closed road near Paris, France.

$\sqrt{\frac{1}{N}\sum_n^N \left(\hat{x}_n - x_n^{ground-truth}\right)}$ and median absolute deviation (MAD): $MAD = \text{median}_{n=1,\dots,N} \left|\hat{x}_n - x_n^{ground-truth}\right|$. The MAD's interest is to mitigate initial errors. In Table I, RMSE and MAD are sufficiently low to validate the EKF indeed.

TABLE I: Errors for the run used for the training, on positions over each axis (in meters) and on yaw (in degrees)

	Position-x	Position-y	Position-z	yaw
RMSE	2.33 m	3.53 m	2.63 m	3.62°
MAD	1.45 m	2.08 m	2.26 m	0.58°

B. Comparison between Euclidean and natural gradient

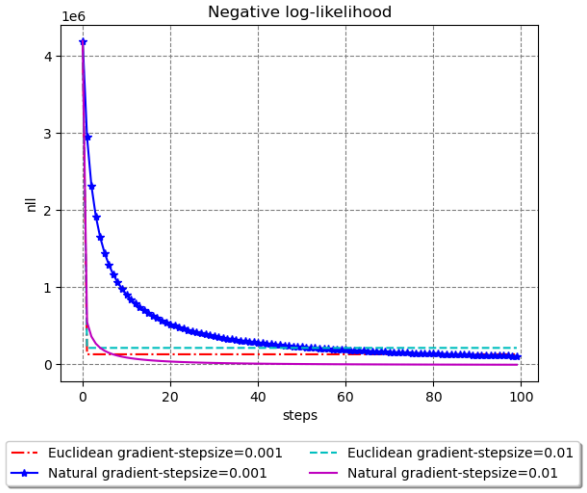
Based on the proposed EKF, the NLL $\mathscr{L}$ given by equation (18) is minimized by gradient descent with fixed step size η, using the backpropagation based gradients of [21].

We compare the results obtained using Euclidean gradient, see (8), or our natural gradient, see (9), with respect to L. For each gradient descent, 100 steps are performed.

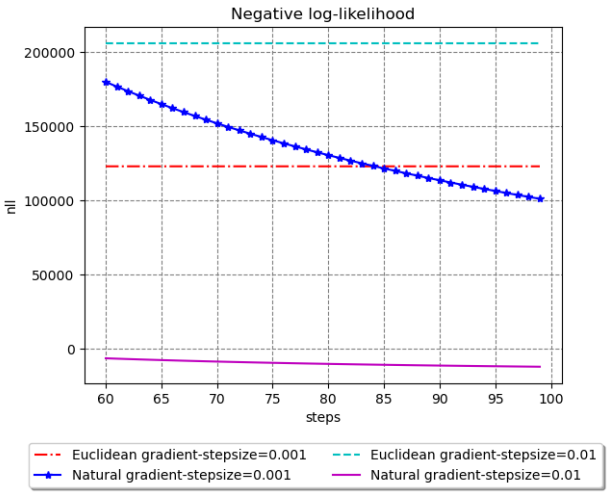
Figure 3a displays the NLL over the gradient descent steps using Euclidean or natural gradient, with fixed step size $\eta = 0.001$ or $\eta = 0.01$. To highlight the differences, Figure 3b displays a zoom on the 40 last steps. As one can see, whatever the value of the step size is ($\eta = 0.001$ or $\eta = 0.01$), the proposed natural gradient outperforms the Euclidean gradient descent in terms of final value reached. Besides, the larger step $\eta = 0.01$ leads to faster convergence.

C. Generalization performance

We now assess how the learned covariance performs in practice on unseen data. In Table II, we compute, for 7 unseen runs, the NLL obtained with each estimated L, i.e. with Euclidean or natural gradient, with step size $\eta = 0.01$ or $\eta = 0.001$. An example of a trajectory corresponding to data in one of these runs is shown in Figure 4. In Table II, one



(a)



(b)

Fig. 3: Negative log-likelihood over the gradient descent with different step sizes, using Euclidean or the proposed natural gradient: (a) over all the steps, (b) zoom over the last steps.

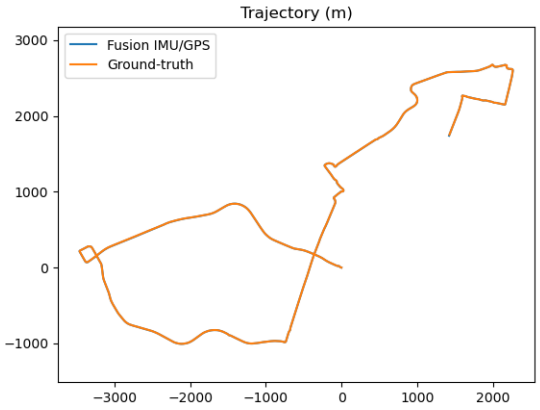


Fig. 4: Example of trajectory used for the tests, namely run 7 in Table II.

TABLE II: Generalization results: final Negative Log Likelihood ($\times 10^4$) depending on the method, on 7 unseen datasets

Run	Euclidean gradient - $\eta = 0.01$	Natural gradient - $\eta = 0.01$	Euclidean gradient - $\eta = 0.001$	Natural gradient - $\eta = 0.001$
Run 1	3.623	1.802	2.165	13.464
Run 2	14.951	-0.734	8.919	6.945
Run 3	7.943	3.252	4.742	16.168
Run 4	20.614	0.577	12.296	17.666
Run 5	14.951	-1.967	8.919	2.062
Run 6	1.482	-0.244	0.887	-0.120
Run 7	6.769	1.587	4.039	14.910

can see that the proposed method, that is, natural gradient on the factor L, with $\eta = 0.01$, consistently outperforms all the other methods. Interestingly, going from $\eta = 0.01$ to $\eta = 0.001$ improves Euclidean gradient but to a point that is still outperformed by the natural gradient with $\eta = 0.01$.

V. Conclusion

In this paper, we have proposed a novel gradient descent algorithm to estimate the noise covariance parameter of an EKF over data, by likelihood maximization. The method was successfully applied to a challenging example of engineering interest, and was demonstrated on real experiments. In the future, we would like to attempt to derive natural gradient w.r.t. the likelihood $\mathscr{L}$, as in [4], instead of the proxy that we used, i.e., that associated with Gaussian families.

Appendix

A. Error state vector used in the invariant EKF

To define the error state vector, denoted as ξ_n, the system state χ_n given in (14) is seen as the concatenation of two states $\chi_n = (\Psi_n, \beta_n)$, where $\Psi_n = (\Omega_n, \upsilon_n, p_n) \in SE_2(3)$ and $\beta_n^T = \left(d_n^T, b_n^T, l_n^T\right) \in \mathbb{R}^9$. The error state vector is split according to this decomposition: $\xi_n^T = \left(\xi_n^{\Psi^T}, \xi_n^{\beta^T}\right)$. The error $\xi_n^\Psi \in \mathbb{R}^9$, associated with Ψ_n, is chosen as left-invariant [7], while the error ξ_n^β, associated with β_n, is chosen as linear.

Given the total error ξ_n, the $\oplus$ operator used for updating the state in equation (17) is defined as follows. The vector $\tilde{\xi}_n = K_n z_n$ is split according to the splitting of the error state vector: $\tilde{\xi}_n^T = \left(\tilde{\xi}_n^{\Psi^T}, \tilde{\xi}_n^{\beta^T}\right)$, and, in the update (17), each part of the system state $\chi_n = (\Psi_n, \beta_n)$ is updated as :

$$\hat{\Psi}_{n|n} = \hat{\Psi}_{n|n-1} \odot \exp\left(\tilde{\xi}_n^\Psi\right), \quad \hat{\beta}_{n|n} = \hat{\beta}_{n|n-1} + \tilde{\xi}_n^\beta, \qquad (19)$$

where $\odot$ is the internal binary operation on $SE_2(3)$, and exp denotes the exponential map on $SE_2(3)$ [7], [13].

References

[1] Abbeel, P., Coates, A., Montemerlo, M., Ng, A.Y., Thrun, S.: Discriminative training of Kalman filters. In: Robotics: Science and systems. vol. 2, p. 1 (2005)

[2] Aghannan, N., Rouchon, P.: On invariant asymptotic observers. In: Proceedings of the 41st IEEE Conference on Decision and Control, 2002. vol. 2, pp. 1479–1484. IEEE (2002)

[3] Amari, S.I.: A foundation of information geometry. Electronics and Communications in Japan (Part I: Communications) **66**(6), 1–10 (1983)

[4] Amari, S.I.: Natural gradient works efficiently in learning. Neural Computation **10**(2), 251–276 (1998)

[5] Barfoot, T.D.: Multivariate Gaussian variational inference by natural gradient descent. arXiv preprint arXiv:2001.10025 (2020)

[6] Barfoot, T.D., Forbes, J.R., Yoon, D.J.: Exactly sparse gaussian variational inference with application to derivative-free batch nonlinear state estimation. The International Journal of Robotics Research **39**(13), 1473–1502 (2020)

[7] Barrau, A., Bonnabel, S.: The invariant extended Kalman filter as a stable observer. IEEE Transactions on Automatic Control **62**(4), 1797–1812 (2016)

[8] Barrau, A., Bonnabel, S.: Invariant kalman filtering. Annual Review of Control, Robotics, and Autonomous Systems **1**, 237–257 (2018)

[9] Bavdekar, V.A., Deshpande, A.P., Patwardhan, S.C.: Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter. Journal of Process Control **21**(4), 585–601 (2011)

[10] Bierman, G.J.: Factorization methods for discrete sequential estimation. Mathematics in Science and Engineering **128** (1977)

[11] Bonnabel, S.: Stochastic gradient descent on Riemannian manifolds. IEEE Transactions on Automatic Control **58**(9), 2217–2229 (2013)

[12] Bonnabel, S., Rouchon, P.: On invariant observers. Control and observer design for nonlinear finite and infinite dimensional systems, Lecture Notes in Control and Information Science, Springer pp. 53–65 (2005)

[13] Brossard, M., Barrau, A., Chauchat, P., Bonnabel, S.: Associating uncertainty to extended poses for on Lie group IMU preintegration with rotating Earth. IEEE Transactions on Robotics **38**(2), 998–1015 (2021)

[14] van Der Laan, N., Cohen, M., Arsenault, J., Forbes, J.R.: The invariant rauch-tung-striebel smoother. IEEE Robotics and Automation Letters **5**(4), 5067–5074 (2020)

[15] Eckman, R.A., Brown, A.J., Adamo, D.R., Gottlieb, R.G.: Normalization and implementation of three gravitational acceleration models. Tech. Rep. TP-2016-218604, NASA (2016)

[16] Gupta, N., Mehra, R.: Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. IEEE Transactions on Automatic Control **19**(6), 774–783 (1974)

[17] Hartley, R., Ghaffari, M., Eustice, R.M., Grizzle, J.W.: Contact-aided invariant extended kalman filtering for robot state estimation. The International Journal of Robotics Research **39**(4), 402–430 (2020)

[18] Kulikova, M.V., Tsyganova, J.V.: Constructing numerically stable Kalman filter-based algorithms for gradient-based adaptive filtering. International Journal of Adaptive Control and Signal Processing **29**(11), 1411–1426 (2015)

[19] Martens, J.: New insights and perspectives on the natural gradient method. The Journal of Machine Learning Research **21**(1), 5776–5851 (2020)

[20] Nielsen, F.: An elementary introduction to information geometry. Entropy **22**(10), 1100 (2020)

[21] Parellier, C., Barrau, A., Bonnabel, S.: Speeding-up backpropagation of gradients through the kalman filter via closed-form expressions. IEEE Transactions on Automatic Control **68**(12), 8171–8177 (2023)

[22] Pavlasek, N., Walsh, A., Forbes, J.R.: Invariant extended kalman filtering using two position receivers for extended pose estimation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 5582–5588. IEEE (2021)

[23] Särkkä, S., Svensson, L.: Bayesian filtering and smoothing, vol. 17. Cambridge university press (2023)

[24] Shumway, R.H., Stoffer, D.S.: An approach to time series smoothing and forecasting using the EM algorithm. Journal of Time Series Analysis **3**(4), 253–264 (1982)

[25] Tsyganova, J.V., Kulikova, M.V.: SVD-based Kalman filter derivative computation. IEEE Transactions on Automatic Control **62**(9), 4869–4875 (2017)

[26] Wu, F., Luo, H., Jia, H., Zhao, F., Xiao, Y., Gao, X.: Predicting the noise covariance with a multitask learning model for kalman filter-based gnss/ins integrated navigation. IEEE Transactions on Instrumentation and Measurement **70**, 1–13 (2020)

[27] Young, P.: Parameter estimation for continuous-time models—a survey. Automatica **17**(1), 23–39 (1981)