The Extended Kalman Filter is a Natural Gradient Descent in Trajectory Space

Yann Ollivier

Abstract

The extended Kalman filter is perhaps the most standard tool to estimate in real time the state of a dynamical system from noisy measurements of some function of the system, with extensive practical applications (such as position tracking via GPS). While the plain Kalman filter for linear systems is well-understood, the extended Kalman filter relies on linearizations which have been debated.

We recover the exact extended Kalman filter equations from first principles in statistical learning: the extended Kalman filter is equal to Amari's *online natural gradient*, applied in the space of trajectories of the system. Namely, each possible trajectory of the dynamical system defines a probability law over possible observations. In principle this makes it possible to treat the underlying trajectory as the parameter of a statistical model of the observations. Then the parameter can be learned by gradient ascent on the log-likelihood of observations, as they become available. Using Amari's *natural* gradient from information geometry (a gradient descent preconditioned with the Fisher matrix, which provides parameterization-invariance) exactly recovers the extended Kalman filter.

This applies only to a particular choice of process noise in the Kalman filter, namely, taking noise proportional to the posterior covariance—a canonical choice in the absence of specific model information.

Overview. State estimation consists in estimating the current state of a dynamical system given noisy observations of a function of this system. Namely, consider a dynamical system with state s_t, inputs u_t and dynamics f, namely,

$$s_t = f(s_{t-1}, u_t) \tag{1}$$

and assume we have access to noisy observations y_t of some function h of the system,

$$y_t = h(s_t, u_t) + \mathcal{N}(0, R) \tag{2}$$

with covariance matrix R. One of the main problems of filtering theory is to estimate the current state s_t given the observations y_t (assuming that f, h, R, and the inputs or control variables u_t are known).

We prove the exact equivalence of two methods to tackle this problem:

- The *extended Kalman filter*, the most standard tool designed to deal with this problem: it is built in a Bayesian setting as a real-time approximation of the posterior mean and covariance of the state given the observations. (We use a particular variant of the filter where the process noise on s_t is modeled as proportional to the posterior covariance, $Q_t \propto P_{t|t-1}$ in Def. 3 [Nel00, §3.2.2] [Hay01, §5.2.2], a canonical choice in the absence of further information. This choice introduces "fading memory" which robustifies the filter [Sim06, §5.5].)

- The *online natural gradient*, a classical tool from statistical learning to estimate the parameters of a probabilistic model. Here, the hidden parameter to be estimated is the whole *trajectory* $\mathbf{s} = (s_t)_{t\geqslant 0}$. Letting $\mathcal{S}$ be the set of trajectories of (1), each possible trajectory $\mathbf{s} = (s_t) \in \mathcal{S}$ defines a probability distribution $p(\mathbf{y}|\mathbf{s})$ on observation sequences $\mathbf{y} = (y_t)_{t\geqslant 1}$ via the observation model (2). So $\mathbf{s}$ can be seen as the parameter of a probabilistic model on $\mathbf{y}$. Then, in principle, $\mathbf{s}$ can be learned by online gradient descent $\frac{\ln p(y_t|\mathbf{s})}{\partial \mathbf{s}}$ in the space of trajectories: each time a new observation y_t becomes available, one can re-estimate $\mathbf{s}$ using a gradient step on the log-likelihood of y_t knowing $\mathbf{s}$.

  The *natural* gradient descent [Ama98] preconditions the gradient steps by the inverse Fisher matrix of the model $p(\mathbf{y}|\mathbf{s})$ with respect to $\mathbf{s}$. This is motivated by invariance to changes of variables over which the model is expressed, and by theorems of asymptotic optimality [Ama98].

We claim that these two methods yield the same estimate of s_t at time t (Thm. 5). The same holds in continuous time for the extended Kalman–Bucy filter (Thm. 17).

This largely extends a previous result by the author, which dealt with the case $f = \mathrm{Id}$: namely, it was shown in [Oll18] that the natural gradient descent to estimate the parameter θ of a probabilistic model from observations y_t, is equivalent to applying a Kalman filter to the hidden state $s_t = \theta$ for all t. Thus the previous result viewed the natural gradient as a particular case of an extended Kalman filter with "static" dynamics; here we view the extended Kalman filter as a natural gradient descent in the space of trajectories, and recover the previous result when $f = \mathrm{Id}$.

This result may contribute to the understanding of the extended Kalman filter. The use of Kalman-like filters in navigation systems (GPS, vehicle control, spacecraft...), time series analysis, econometrics, etc. [Sä13], is extensive to the point it has been described as one of the greater discoveries of mathematical engineering [GA15]. But while the plain Kalman filter (which deals with linear f) is exactly optimal, the extended Kalman filter relies on linear expansions. Variants of the extended filter have been proposed, for instance using higher-order expansions for certain terms, though with

more limited use [RG11]. On the other hand, the natural gradient can be constructed from first principles. Remarkably, the quite complicated formulas defining the extended Kalman filter can be derived exactly from its natural gradient interpretation.

However, two technical points make the precise statement of the correspondence (Theorem 5) more subtle.

First, an important choice when applying the extended Kalman filter is the choice of system noise $\mathcal{N}(0, Q)$ that is added to the dynamical system (1). (One may think of the process noise Q in the Kalman filter either as actual noise in a stochastic system, or as a modeling tool to apply the Kalman filter when knowledge of the deterministic system f is imperfect; the results below hold regardless of interpretation.) Often, Q is adjusted by trial and error. A canonical choice is to take Q proportional to the posterior covariance on s (Def. 4, [Nel00, §3.2.2] [Hay01, §5.2.2]); this is equivalent to introducing *fading memory* into the filter [Sim06, §7.4].

Our result applies only in the latter case; this is certainly a restriction. Fundamentally, choices such as $Q = \mathrm{Id}$ define a preferred basis in state space, while the extended Kalman filter with Q proportional to the posterior variance can be expressed in an abstract, basis-free vector space. Since the natural gradient is basis-invariant, it can only be equivalent to another basis-invariant algorithm.[1]

Our results relate the extended Kalman filter with nonzero Q to the natural gradient over trajectories of the *noiseless* system (1). The choice of noise Q for applying the Kalman filter corresponds to different natural gradient learning rates: the particular choice $Q = 0$ corresponds to a learning rate $1/t$ in the natural gradient, while positive Q correspond to larger learning rates.

Second, the natural gradient uses quantities expressed in an abstract Riemannian manifold of trajectories $\mathbf{s}$; still, to perform an actual update of $\mathbf{s}$, a numerical representation of $\mathbf{s}$ has to be used. (The direction of the natural gradient is parameterization-invariant, but the actual step requires an explicit parameterization, whose influence vanishes only in the limit of small learning rates.) The space of trajectories $\mathbf{s}$ could be parameterized, for instance, by the initial state s_0, or the state s_t at any time t provided f is invertible. The correspondence turns out to be exact if, when the observation y_t becomes available at time t, the natural gradient update uses the current state s_t to parameterize of the trajectory $\mathbf{s}$. One one hand this seems quite natural, and computationally convenient at time t; on the other hand, it means we are performing a natural gradient descent in a coordinate system that shifts in time.

[1] Other choices of Q, such as $Q = \mathrm{Id}$, do have an interpretation as gradient descents in trajectory space, but using quite artificial preconditioning matrices instead of the Fisher matrix; we do not develop this point.

Example: recovering the natural gradient from the extended Kalman filter for statistical learning problems. The correspondence works both ways: in particular, it can be used to view the online natural gradient on a parameter θ of a statistical model, as a particular instance of extended Kalman filtering. This important example corresponds to $f = \mathrm{Id}$ above, and is the case treated in [Oll18]. We summarize it again for convenience.

Let $p(y_t|u_t, \theta)$ be a statistical model to predict a quantity y_t from an input u_t given a parameter θ. We assume that the model can be written as $y_t \sim p_{\text{obs}}(y_t|h(\theta, u_t))$ where $h(\theta, u_t)$ is a function that encodes the prediction on y_t, and the noise model p_{obs} is an exponential family with mean parameter $h(\theta, u_t)$, such as $y_t = h(\theta, u_t) + \mathcal{N}(0, R)$. The function h may be anything, such as $h(\theta, u_t) = \theta^\top u_t$ for a linear model, or a feedforward neural network with input u_t and parameters θ.

A standard approach for this problem would be stochastic gradient descent: updating the parameter θ via gradient descent of $\ln p(y_t|u_t\theta)$ for each new observation pair (u_t, y_t). But the extended Kalman filter can also be applied to this problem by viewing θ as the hidden state of a static system, namely, $s_t = \theta$ and $f = \mathrm{Id}$, and treating the y_t as observations of θ knowing u_t. See eg [SW88] for an early example with neural networks. Following [Oll18], we extend the extended Kalman filter in Def. 3 to cover any exponential family as the model for y_t given $h(s_t, u_t)$: this allows the Kalman filter to deal with discrete/categorical data y_t, for instance, by letting $h(\theta, u_t)$ be the list of probabilities of all classes.

The main result from [Oll18] states that the extended Kalman filter for this problem, is exactly equivalent to the online natural gradient on θ. This is a corollary of the present work by taking $f = \mathrm{Id}$ and $s_t = \theta$: indeed, with $f = \mathrm{Id}$ we can identify the set of trajectories $\mathbf{s} \in \mathcal{S}$ with their value at any time, and the gradient descent on $\mathbf{s}$ becomes a gradient on θ.

So the online natural gradient for a statistical problem with parameter θ appears as a particular instance of the extended Kalman filter on a static system $f = \mathrm{Id}$, while the extended Kalman filter for general f appears as a particular case of the online natural gradient in the more abstract space of trajectories.

Does this provide a convergence proof for the extended Kalman filter, via the theory of stochastic gradient descent? Not really, as consecutive observations in a dynamical system are not independent and identically distibuted. The online natural gradient on a dynamical system is not quite an instance of stochastic gradient descent.

Related work. The role of the information matrix in Kalman filtering was recognized early [Jaz70, §7.5], and led to the formulation of the Kalman filter using the inverse covariance matrix known as the "information filter"

[Sim06, §6.2]. However, except in the static case treated in [Oll18], this does not immediately translate into an equivalence between extended Kalman filtering and natural gradient descent, as is clear from the amount of work needed to prove our results.

Several recent works make a link between Kalman filtering and preconditioned gradient descent in some particular cases. [RRK+92] argue that for neural networks, backpropagation, i.e., ordinary gradient descent, "is a degenerate form of the extended Kalman filter". [Ber96] identifies the extended Kalman filter with a Gauss–Newton gradient descent for the specific case of nonlinear regression. [dFNG00] interprets process noise in the static Kalman filter as an adaptive, per-parameter learning rate, thus akin to a preconditioning matrix. [ŠKT01] uses the Fisher information matrix to study the variance of parameter estimation in Kalman-like filters, without using a natural gradient; [BL03] comment on the similarity between Kalman filtering and a version of Amari's natural gradient for the specific case of least squares regression; [Mar14] and [Oll15] mention the relationship between natural gradient and the Gauss–Newton Hessian approximation; [Pat16] exploits the relationship between second-order gradient descent and Kalman filtering in specific cases including linear regression; [LCL+17] use a natural gradient descent over Gaussian distributions for an auxiliary problem arising in Kalman-like Bayesian filtering, a problem independent from the one treated here.

[HRW12] interpret the Kalman filter as an online Newton method over a variable representing the trajectory. Namely, defining the "past trajectory" of the system as $z_t := (s_1, \dots, s_t)$, and denoting the log-likelihood function by $J_t(z_t) := \frac{1}{2}\sum_{s=1}^{t} \|s_t - f(s_{t-1}, u_t)\|^2_{Q^{-1}} + \frac{1}{2}\sum_{s=1}^{t} \|y_t - h(s_t, u_t)\|^2_{R^{-1}}$, they prove that the Kalman filter can be seen, at each time step, as one step of the Newton method on z_t to find the minimum of J_t. This is somewhat reminiscent of the approach taken here. However, the derivation for the nonlinear case is incomplete (otherwise, this would prove optimality of the extended Kalman filter even in the nonlinear case). Their result states that assuming $\hat{z}_{t-1}$ minimizes J_{t-1}, then the extended Kalman filter tries to find the state s_t that minimizes J_t, via one Newton step. In the linear case, J_t is quadratic, so this Newton step successfully finds the minimum of J_t, so $\hat{z}_t$ minimizes J_t and the idea can be iterated. Therefore the plain (non-extended) Kalman filter can be seen as an online Newton method on $z_t = (s_1, \dots, s_t)$. However, if $\hat{z}_t$ does not exactly minimize J_t then the extended Kalman filter at time $t+1$ does not coincide with a Newton step anymore. Using the Fisher information matrix instead of the Hessian, namely, a natural gradient instead of a Newton method, helps with this issue.

Notation conventions. In statistical learning, the external inputs or regressor variables are often denoted x. In Kalman filtering, x often denotes

the state of the system, while the external inputs are often u. Thus we will avoid x altogether and denote by u the inputs and by s the state of the system.

The variable to be predicted at time t will be y_t, and $\hat{y}_t$ is the corresponding prediction. In general $\hat{y}_t$ and y_t may be different objects in that $\hat{y}_t$ encodes a full probabilistic prediction for y_t. For Gaussians with known variance, $\hat{y}_t$ is just the predicted mean of y_t, so in this case y_t and $\hat{y}_t$ are the same type of object. For Gaussians with unknown variance, $\hat{y}$ encodes both the mean and second moment of y. For discrete categorical data, $\hat{y}$ encodes the probability of each possible outcome y.

The natural gradient descent on parameter θ_t will use the Fisher matrix J_t. The Kalman filter will have posterior covariance matrix P_t.

For multidimensional quantities x and $y = f(x)$, we denote by $\frac{\partial y}{\partial x}$ the Jacobian matrix of y w.r.t. x, whose (i, j) entry is $\frac{\partial f_i(x)}{\partial x_j}$. This satisfies the chain rule $\frac{\partial z}{\partial y}\frac{\partial y}{\partial x} = \frac{\partial z}{\partial x}$. With this convention, gradients of real-valued functions are *row* vectors, so that a gradient descent takes the form $x \leftarrow x - \eta\,(\partial f/\partial x)^\top$.

For a column vector u, $u^{\otimes 2}$ is synonymous with $uu^\top$, and with $u^\top u$ for a row vector.

1 Natural Gradient Descent

A standard approach to optimize the parameter θ of a probabilistic model, given a sequence of observations (y_t), is an online gradient descent

$$\theta_t \leftarrow \theta_{t-1} + \eta_t \frac{\partial \ln p(y_t|\theta)}{\partial \theta}^\top \tag{3}$$

with learning rate η_t. This simple gradient descent is particularly suitable for large datasets and large-dimensional models [BL03], and has become a staple of current statistical learning, but has several practical and theoretical shortcomings. For instance, it uses the same non-adaptive learning rate for all parameter components. Moreover, simple changes in parameter encoding or in data presentation (e.g., encoding black and white in images by 0/1 or 1/0) can result in different learning performance.

This motivated the introduction of the *natural gradient* [Ama98]. It is built to achieve invariance with respect to parameter re-encoding; in particular, learning become insensitive to the characteristic scale of each parameter direction, so that different directions naturally get suitable learning rates. The natural gradient is the only general way to achieve such invariance [AN00, §2.4].

The natural gradient preconditions the gradient descent with $J(\theta)^{-1}$ where J is the *Fisher information matrix* [Kul97] with respect to the parameter θ. For a smooth probabilistic model $p(y|\theta)$ over a random variable y

with parameter θ, the latter is defined as

$$J(\theta) := \mathbb{E}_{y\sim p(y|\theta)}\left[\frac{\partial \ln p(y|\theta)}{\partial\theta}^{\otimes 2}\right] = -\mathbb{E}_{y\sim p(y|\theta)}\left[\frac{\partial^2 \ln p(y|\theta)}{\partial\theta^2}\right] \tag{4}$$

If the model for y involves an input u, then an additional expectation or empirical average over the input is introduced in the definition of J [AN00, §8.2] [Mar14, §5].

Intuitively, J captures the change in the distribution p_θ when θ changes infinitesimally, measured by the relative entropy (Kullback–Leibler divergence), namely

$$\mathrm{KL}(p_{\theta+\delta\theta}|p_\theta) = \frac{1}{2}\delta\theta^\top J(\theta)\delta\theta + O(\delta\theta^3) \tag{5}$$

In particular, this only depends on θ via p_θ. Namely, making a change of variables in the parameters θ of a probabilistic model p_θ will not change $\mathrm{KL}(p_{\theta+\delta\theta}|p_\theta)$; the norm $\delta\theta^\top J(\theta)\delta\theta$ is a parameter-invariant way to measure the change of p_θ induced by θ, and this turns Θ into a Riemannian manifold [AN00].

Intuitively the natural gradient is thus the steepest gradient direction in Kullback–Leibler distance: the natural gradient direction $J(\theta)^{-1}\partial\ell(\theta)/\partial\theta$ of a function $\ell(\theta)$ gives, at first order, the direction $\delta\theta$ with steepest increase of f for the minimum change $\mathrm{KL}(p_{\theta+\delta\theta}|p_\theta)$ of p_θ [OAAH17].

However, this comes at a large computational cost for large-dimensional models: just storing the Fisher matrix already costs $O((\dim\theta)^2)$. Various strategies are available to approximate the natural gradient for complex models such as neural networks, using diagonal or block-diagonal approximation schemes for the Fisher matrix, e.g., [LMB07, Oll15, MCO16, GS15, MG15].

Definition 1 below formally introduces the *online* natural gradient.

Definition 1 (Online natural gradient). *Consider a statistical model with parameter θ that predicts an output y given an input u, via a model $y \sim p(y|u,\theta)$. Given observation pairs (u_t, y_t), the goal is to minimize, online, the log-likelihood loss function*

$$-\sum_t \ln p(y_t|u_t,\theta) \tag{6}$$

as a function of θ.

The online natural gradient *maintains a current estimate θ_t of the parameter θ, and a current approximation J_t of the Fisher matrix. The parameter is estimated by a gradient descent with preconditioning matrix J_t^{-1}, namely*

$$J_t \leftarrow (1-\gamma_t)J_{t-1} + \gamma_t\,\mathbb{E}_{y\sim p(y|u_t,\theta)}\left[\frac{\partial \ln p(y|u_t,\theta)}{\partial\theta}^{\otimes 2}\right] \tag{7}$$

$$\theta_t \leftarrow \theta_{t-1} + \eta_t\, J_t^{-1}\left(\frac{\partial \ln p(y_t|u_t,\theta)}{\partial\theta}\right)^\top \tag{8}$$

with learning rate η_t and Fisher matrix decay rate γ_t.

In the Fisher matrix update, the expectation over all possible values $y \sim p(y|\hat{y})$ can often be computed algebraically (for a given input u_t), but this is sometimes computationally bothersome (for instance, in neural networks, it requires $\dim(\hat{y}_t)$ distinct backpropagation steps [Oll15]). A common solution [APF00, LMB07, Oll15, PB13] is to just use the value $y = y_t$ (*outer product* approximation) instead of the expectation over y. Another is to use a Monte Carlo approximation with a single sample of $y \sim p(y|\hat{y}_t)$ [Oll15, MCO16], namely, using the gradient of a synthetic sample instead of the actual observation y_t in the Fisher matrix. These latter two solutions are often confused; only the latter provides an unbiased estimate, see discussion in [Oll15, PB13].

The online "smoothed" update of the Fisher matrix in (7) reuses Fisher matrix values computed at previous values of θ_t and u_t, instead of using the exact Fisher matrix at θ_t in (8). Such or similar updates are used in [LMB07, MCO16]. The reason is at least twofold. First, the exact Fisher matrix involves an expectation over the inputs u_t [AN00, §8.2], so using it would mean recomputing the value of the Fisher matrix on all previous observations each time θ_t is updated. Instead, to keep the algorithm online, (7) reuses values computed on previous observations u_t, even though they were computed using an out-of-date parameter θ. The decay rate γ_t controls this moving average over observations (e.g., $\gamma_t = 1/t$ realizes an equal-weight average over all inputs seen so far). Second, the expectation over $y \sim p(y|u_t, \theta)$ in (7) is often replaced with a Monte Carlo estimation with only one value of y, and averaging over time compensates for this Monte Carlo sampling.

As a consequence, since θ_t changes over time, this means that the estimate J_t mixes values obtained at different values of θ, and converges to the Fisher matrix only if θ_t changes slowly, i.e., if $\eta_t \to 0$. The correspondence below with Kalman filtering suggests using $\gamma_t = \eta_t$.

Natural gradient descent in different charts of a manifold. One motivation for natural gradient is its invariance to a change of parameterization of the model REF. However, this holds only in the limit of small learning rates $\eta_t \to 0$, or in continuous time; otherwise this is true only up to $O(\eta_t^2)$. Indeed, if θ belongs to a manifold, the object $J_t^{-1} \left(\frac{\partial \ell_t}{\partial \theta}\right)^\top$ is a well-defined tangent vector at θ, but the additive update $\theta \leftarrow \theta - \eta_t J_t^{-1} \left(\frac{\partial \ell_t}{\partial \theta}\right)^\top$ is still performed on an explicit parameterization (chart).[2] Thus, each time an explicit update is performed, a coordinate system must be chosen.

Thus, from now on we will explicitly separate the abstract points $\vartheta \in \Theta$

[2] A possible solution is to use the geodesics of the Riemannian manifold defined by the Fisher metric, but this is rarely convenient except in particular situations where these geodesics are known explicitly (e.g., [Ben15, Bon13]).

in the abstract parameter manifold Θ, and their expression $\theta \in \mathbb{R}^{\dim(\Theta)}$ in a coordinate system. Likewise, we will denote J the Fisher matrix in a coordinate system, and $\mathcal{J}$ the corresponding abstract Fisher metric, a $(0,2)$-tensor on Θ. We will denote $\mathfrak{p}_t(y_t|\vartheta)$ the probability distribution on observations at time t knowing the parameter $\vartheta \in \Theta$, and $p_t(y_t|\theta)$ the same model in a coordinate system. The loss function to be minimized is $-\sum_t \ln \mathfrak{p}_t(y_t|\vartheta)$.

In particular, when observing y_t at time t, the natural gradient direction with Fisher metric tensor $\mathcal{J}$ is

$$\mathcal{J}^{-1} \frac{\partial \ln \mathfrak{p}_t(y_t|\vartheta)}{\partial \vartheta} \tag{9}$$

where $\frac{\partial \ln \mathfrak{p}_t(y_t|\vartheta)}{\partial \vartheta}$ is a cotangent vector, which becomes a tangent vector after applying $\mathcal{J}^{-1}$. Then we would like to consider an update of the type

$$\vartheta \leftarrow \vartheta + \eta_t \, \mathcal{J}^{-1} \frac{\partial \ln \mathfrak{p}_t(y_t|\vartheta)}{\partial \vartheta} \tag{10}$$

with learning rate η_t. However, this $+$ sign does not make sense in a manifold, so we have to apply this in an explicit parameterization (chart), then jump back to the manifold.[3]

This is the object of the next definition: at each step, we first jump to a chart Φ, apply the natural gradient update in that chart, and jump back to the manifold via Φ^{-1}. For a given chart $\Phi\colon \Theta \to \mathbb{R}^{\dim(\Theta)}$ on a manifold Θ, and an abstract tensor g at $\vartheta \in \Theta$, we denote the coordinate expression of g in chart Φ by $\mathbf{T}\Phi(g)$ (specifying θ is not needed since an abstract tensor g includes its basepoint information). Given a numerical tensor g with coordinates expressed in the chart Φ, we denote $\mathbf{T}_\vartheta \Phi^{-1}(g)$ the corresponding abstract tensor at $\vartheta \in \Theta$.

Definition 2 (Online natural gradient in charts on a manifold). *Let Θ be a smooth manifold. For each $t \geqslant 1$, let $\mathfrak{p}_t(y|\vartheta)$ be a probabilistic model on some variable y, depending smoothly on $\vartheta \in \Theta$.*

For each time $t \geqslant 1$, let $\Phi_t\colon \Theta \to \mathbb{R}^{\dim(\Theta)}$ be a chart on Θ.

The online natural gradient descent *for the observations y_t, in the sequence of charts Φ_t, maintains an element $\vartheta_t \in \Theta$ and a metric tensor $\mathcal{J}_t$ at ϑ_t,*

[3] This only matters at second order in the learning rate: two different parameterizations will provide updates differing by $O(\eta_t^2)$. In particular, in continuous time these considerations disappear, and the continuous-time trajectory

$$\frac{\mathrm{d}\vartheta}{\mathrm{d}t} = \mathcal{J}^{-1} \frac{\partial \ln \mathfrak{p}_t(y_t|\vartheta)}{\partial \vartheta} \tag{11}$$

is parameterization-independent.

defined inductively by

$$\theta \leftarrow \Phi_t(\vartheta_{t-1}), \qquad J \leftarrow \mathbf{T}\Phi_t(\mathcal{J}_{t-1}) \tag{12}$$

$$J_t \leftarrow (1-\gamma_t)J + \gamma_t \mathbf{T}\Phi_t \left(\mathbb{E}_{y\sim \mathfrak{p}_t(y|\vartheta_{t-1})} \left[\frac{\partial \ln \mathfrak{p}_t(y|\vartheta)}{\partial \vartheta_{t-1}}^{\otimes 2} \right] \right) \tag{13}$$

$$\theta_t \leftarrow \theta + \eta_t \, J_t^{-1} \mathbf{T}\Phi_t \left(\frac{\partial \ln \mathfrak{p}_t(y_t|\vartheta)}{\partial \vartheta_{t-1}} \right)^\top \tag{14}$$

$$\vartheta_t \leftarrow \Phi_t^{-1}(\theta_t), \qquad \mathcal{J}_t \leftarrow \mathbf{T}_{\vartheta_t}\Phi_t^{-1}(J_t) \tag{15}$$

with learning rate η_t and Fisher matrix decay rate γ_t.

If the chart Φ_t is constant in time, then this reduces to the ordinary online natural gradient on θ_t (Lemma 6 below). Indeed, if $\Phi_t = \Phi_{t+1}$ then applying Φ_t^{-1} at the last step then applying Φ_{t+1} in the next step cancels out, so this amounts to just disregarding ϑ and working on θ.

2 Kalman Filtering

One possible definition of the extended Kalman filter is as follows [Sim06, §15.1]. We are trying to estimate the current state of a dynamical system s_t whose evolution equation is known but whose precise value is unknown; at each time step, we have access to a noisy measurement y_t of a quantity $\hat{y}_t = h(s_t)$ which depends on this state.

The Kalman filter maintains an approximation of a Bayesian posterior on s_t given the observations $y_1, \dots, y_t$. The posterior distribution after t observations is approximated by a Gaussian with mean s_t and covariance matrix P_t. (Indeed, Bayesian posteriors always tend to Gaussians asymptotically under mild conditions, by the Bernstein–von Mises theorem [vdV00].) The Kalman filter prescribes a way to update s_t and P_t when new observations become available.

The Kalman filter update is summarized in Definition 3 below. It is built to provide the *exact* value of the Bayesian posterior in the case of *linear* dynamical systems with Gaussian measurements and a Gaussian prior. In that sense, it is exact at first order.

Definition 3 (Extended Kalman filter). *Consider a dynamical system with state s_t, inputs u_t and outputs y_t,*

$$s_t = f(s_{t-1}, u_t) + \mathcal{N}(0, Q_t), \qquad \hat{y}_t = h(s_t, u_t), \qquad y_t \sim p_{\text{obs}}(y|\hat{y}_t) \tag{16}$$

where $p_{\text{obs}}(\cdot|\hat{y})$ denotes an exponential family with mean parameter $\hat{y}$ (e.g., $y = \mathcal{N}(\hat{y}, R)$ with fixed covariance matrix R).

The extended Kalman filter *for this dynamical system estimates the current state s_t given observations $y_1, \dots, y_t$ in a Bayesian fashion. At*

each time, the Bayesian posterior distribution of the state given $y_1, \dots, y_t$ is approximated by a Gaussian $\mathcal{N}(s_t, P_t)$ so that s_t is the approximate maximum a posteriori, and P_t is the approximate posterior covariance matrix. (The prior is $\mathcal{N}(s_0, P_0)$ at time 0.) Each time a new observation y_t is available, these estimates are updated as follows.

The transition step (before observing y_t) is

$$s_{t|t-1} \leftarrow f(s_{t-1}, u_t) \tag{17}$$

$$F_{t-1} \leftarrow \left.\frac{\partial f}{\partial s}\right|_{(s_{t-1}, u_t)} \tag{18}$$

$$P_{t|t-1} \leftarrow F_{t-1} P_{t-1} F_{t-1}^\top + Q_t \tag{19}$$

$$\hat{y}_t \leftarrow h(s_{t|t-1}, u_t) \tag{20}$$

and the observation step after observing y_t is

$$E_t \leftarrow \textit{sufficient statistics}(y_t) - \hat{y}_t \tag{21}$$

$$R_t \leftarrow \mathrm{Cov}(\textit{sufficient statistics}(y) | \hat{y}_t) \tag{22}$$

where the sufficient statistics are those of the exponential family p_{obs} (for a Gaussian model $y = \mathcal{N}(\hat{y}, R)$ with known R these are just the error $E_t = y_t - \hat{y}_t$ and the covariance matrix $R_t = R$)

$$H_t \leftarrow \left.\frac{\partial h}{\partial s}\right|_{(s_{t|t-1}, u_t)} \tag{23}$$

$$K_t \leftarrow P_{t|t-1} H_t^\top \left(H_t P_{t|t-1} H_t^\top + R_t \right)^{-1} \tag{24}$$

$$P_t \leftarrow (\mathrm{Id} - K_t H_t)\, P_{t|t-1} \tag{25}$$

$$s_t \leftarrow s_{t|t-1} + K_t E_t \tag{26}$$

Defining the output noise via an exponential family $p_{\mathrm{obs}}(y|\hat{y})$ allows for a straightforward treatment of various output models, such as discrete outputs (by letting $\hat{y}$ encode the probabilities of each class) or Gaussians with unknown variance. In the Gaussian case with known variance our definition is fully standard. However, for continuous variables with non-Gaussian output noise, the definition of E_t and R_t above differs from the practice of modelling non-Gaussian noise via a nonlinear function applied to Gaussian noise.[4]

Definition 4 (Pure fading-memory Kalman filter). *The* pure fading-memory Kalman filter *consists in taking the process noise Q_t proportional to $P_{t|t-1}$, so that the noise on the dynamics of s_t is modeled to be*

[4]Non-Gaussian output noise is often modelled in Kalman filtering via a continuous nonlinear function applied to a Gaussian noise [Sim06, 13.1]; this cannot easily represent discrete random variables. Moreover, since the filter linearizes the function around the 0 value of the noise [Sim06, 13.1], in that approach the noise is still implicitly Gaussian, though with a state-dependent variance.

proportional to the current uncertainty on s_t. Specifically, given a sequence of weights $\alpha_t \geqslant 0$, we call pure fading-memory extended Kalman filter *the choice*

$$Q_t = \alpha_t F_{t-1} P_{t-1} F_{t-1}^\top \tag{27}$$

so that the transition equation (19) for P becomes

$$P_{t|t-1} \leftarrow (1+\alpha_t) F_{t-1} P_{t-1} F_{t-1}^\top \tag{28}$$

and Q_t is proportional to $P_{t|t-1}$.

In the Bayesian interpretation of the Kalman filter, this amounts to giving more weights to the likelihood of recent observations: the weight for the likelihood of previous observations decreases by a factor $1/(1+\alpha_t)$ at each step, hence the name "fading memory". This prevents the filter from ultimately growing stale and corresponds to larger learning rates for the natural gradient descent. The choice $\alpha_t = 0$, on the other hand, corresponds to a learning rate $1/t$ for the natural gradient descent.

3 Statement of the Correspondence

Notation for the dynamical system. *We consider a dynamical system with state $s_t \in \mathbb{R}^{\dim(s)}$, inputs $u_t \in \mathbb{R}^{\dim(u)}$ and dynamics f, namely,*

$$s_t = f(s_{t-1}, u_t) \tag{29}$$

where f is a smooth function from $\mathbb{R}^{\dim(s)} \times \mathbb{R}^{\dim(u)}$ to $\mathbb{R}^{\dim(s)}$. Predictions $\hat{y}_t \in \mathbb{R}^{\dim(\hat{y})}$ are made on observations $y_t \in \mathbb{R}^{\dim(y)}$ via

$$\hat{y}_t = h(s_t, u_t) \tag{30}$$

where h is a smooth function from $\mathbb{R}^{\dim(s)} \times \mathbb{R}^{\dim(u)}$ to $\mathbb{R}^{\dim(\hat{y})}$, and the observation model on y_t is

$$y_t \sim p_{\text{obs}}(y_t | \hat{y}_t) \tag{31}$$

where p_{obs} is some exponential family with mean parameter $\hat{y}_t$ (such as a Gaussian with mean $\hat{y}_t$ and known variance).

We refer to Appendix A for a reminder on exponential families.

Notation for natural gradient on trajectories. *Given a dynamical system as above and a sequence of inputs $(u_t)_{t\geqslant 1}$, we denote by $\mathcal{S}$ the set of trajectories of the dynamical system, i.e., the set of sequences $\mathbf{s} = (s_t)_{t\geqslant 0}$ such that $s_t = f(s_{t-1}, u_t)$ for all $t \geqslant 1$.*

We also define the chart Φ_t that parameterizes trajectories by their state at time t:

$$\Phi_t \colon \mathbf{s} \mapsto s_t \tag{32}$$

Each trajectory $\mathbf{s} = (s_t)_{t\geqslant 0} \in \mathcal{S}$ *defines a probability distribution on observations by setting*

$$\mathfrak{p}_t(y_t|\mathbf{s}) := p_{\mathrm{obs}}(y_t|\hat{y}_t) = p_{\mathrm{obs}}(y_t|h(\Phi_t(\mathbf{s}), u_t)) \tag{33}$$

where $\hat{y}_t = h(\Phi_t(\mathbf{s}), u_t)$ *is the prediction made at time* t *from trajectory* $\mathbf{s}$*.*

Thus, we can apply the general definition of online natural gradient (Definition 2) to the models $\mathfrak{p}_t$ in the charts Φ_t: this provides an online natural gradient descent on $\mathbf{s}$ given the observations y_t.

Thus, given an initial trajectory $\mathbf{s}^0$ (typically defined by its initial state s_0^0), the online natural gradient descent produces an estimated trajectory $\mathbf{s}^t$ after observing $y_1, \dots, y_t$. Let us denote

$$s_t := \Phi_t(\mathbf{s}^t), \qquad s_{t|t-1} := \Phi_t(\mathbf{s}^{t-1}) \tag{34}$$

the estimated states at time t of the trajectories $\mathbf{s}^t$ and $\mathbf{s}^{t-1}$, respectively. In the notation of Definition 2, the space Θ is $\mathcal{S}$, the parameter ϑ_t is $\mathbf{s}^t$, its expression θ_t in the chart Φ_t is s_t, and the intermediate value θ is $s_{t|t-1}$.

Our goal is to show that these satisfy the same evolution equations as in the extended Kalman filter. Namely, we will prove the following.

Theorem 5 (Extended Kalman filter as a natural gradient on trajectories). *Consider a dynamical system as above, an initial state* s_0*, a sequence of inputs* $(u_t)_{t\geqslant 1}$*, and a sequence of observations* $(y_t)_{t\geqslant 1}$*.*

Let s_t *be the state estimated at time* t *by the pure fading-memory Kalman filter (Defs. 3 and 4) with observations* (y_t)*, initial state* s_0 *and initial covariance matrix* P_0*.*

Let $\mathbf{s}^t \in \mathcal{S}$ *be the trajectory estimated after* t *steps of the online natural gradient for the model* $\mathfrak{p}_t(y_t|\mathbf{s})$ *in the chart* Φ_t *(Def. 2), initialized at the trajectory* $\mathbf{s}^0$ *starting at* s_0 *(*$\mathbf{s}_0^0 = s_0$*), and with initial Fisher matrix* J_0*.*

Assume that the initializations and hyperparameters (learning rate, fading memory rate) of the two algorithms are related via

$$P_0 = \eta_0 J_0^{-1}, \qquad \eta_t = \gamma_t, \qquad \frac{1}{\eta_t} = \frac{1}{1+\alpha_t}\frac{1}{\eta_{t-1}} + 1 \tag{35}$$

Then for all $t \geqslant 0$*, the trajectory* $\mathbf{s}^t$ *passes through* s_t *at time* t*:*

$$\mathbf{s}_t^t = s_t \tag{36}$$

and the Kalman covariance and Fisher matrix satisfy $P_t = \eta_t J_t^{-1}$*.*

Let us give a few examples to illustrate the relation between hyperparameters: with $\alpha_t = 0$ (no process noise in the Kalman filter), the natural gradient learning rates must satisfy $1/\eta_t = 1 + 1/\eta_{t-1}$, which is satisfied by $\eta_t = 1/(t + \mathrm{cst})$. This is the classical asymptotic rate for parameter

identification in statistical theory; on a dynamical system it can be realized only in the absence of noise in the system. On the other hand, a constant $\alpha_t = \alpha > 0$ corresponds to a constant gradient learning rate $\eta_t = \frac{\alpha}{1+\alpha}$ (and any other choice of η_0 defines by induction a sequence η_t that tends to this value).

The extended Kalman filter appears as a natural gradient with the particular natural gradient setting $\eta_t = \gamma_t$: namely, the Fisher metric decay rate is equal to the natural gradient learning rate. This is commented in [Oll18] for the case $f = \mathrm{Id}$; in short, the natural gradient maintains a second-order approximation of the log-likelihood of recent observations as a function of the parameter, and $\eta_t = \gamma_t$ corresponds to using the same decay rate for old observations in the first-order and second-order terms.

Note that $\eta_t \to 1$ when $\alpha \to \infty$: infinite noise on the process corresponds to infinite forgetting of the past, and in that case the Kalman filter jumps to the maximum likelihood estimator for the latest observation (estimated at second order) [Sim06, p. 212]. With the natural gradient, a learning rate of 1 corresponds to directly jumping to the minimum for quadratic functions (learning rates above 1 overshoot with the natural gradient).

4 Proof of Theorem 5

The proof considers the online natural gradient from Definition 2 on the trajectory space of a dynamical system, and gradually makes all elements more explicit until we are left with the extended Kalman filter.

For the proof, we shall assume that the function $f(\cdot, u_t)$ that maps s_{t-1} to s_t is invertible; this guarantees that we can indeed parameterize trajectories by their value s_t at any time t. In particular, the quantities F_t in the extended Kalman filter are invertible. Without this assumption we would have to consider equivalence classes of trajectories having the same value at time t, as a function of time, which would make notation substantially heavier. This is not really needed: in the end all terms F_t^{-1} vanish from the expressions, and the statement of Theorem 5 only involves the value s_t and Fisher matrix J_t at time t, not the past trajectory back-computed from s_t.

4.1 Online Natural Gradient in Charts: Explicit Updates

Here we consider the general setting of Definition 2: Θ is a smooth manifold; for each $t \geqslant 1$, $\mathfrak{p}_t(y|\vartheta)$ is a probabilistic model on some variable y, depending smoothly on $\vartheta \in \Theta$; for each time $t \geqslant 1$, $\Phi_t \colon \Theta \to \mathbb{R}^{\dim(\Theta)}$ is some chart on Θ.

Lemma 6. *Denote*

$$p_t(y|\theta) := \mathfrak{p}_t(y|\Phi_t^{-1}(\theta)) \tag{37}$$

the expression of the probabilistic model in the chart Φ_t.

Then the updates (13)–(14) for J_t and θ_t in the online natural gradient are equivalent to

$$J_t \leftarrow (1-\gamma_t)J + \gamma_t \, \mathbb{E}_{y\sim p_t(y|\theta)} \left[\frac{\partial \ln p_t(y|\theta)}{\partial \theta}^{\otimes 2} \right] \tag{38}$$

$$\theta_t \leftarrow \theta + \eta_t \, J_t^{-1} \frac{\partial \ln p_t(y_t|\theta)}{\partial \theta}^{\top} \tag{39}$$

where θ and J are as in (12) above.

In particular, if the chart Φ_t is the same at all times, then the abstract online natural gradient reduces to the usual online natural gradient, because (12) and (15) cancel each other out.

Proof.
This follows by applying Lemma 26 from Appendix B to the function $\ln \mathfrak{p}_t(y|\vartheta)$. □

By studying the effect of a change of chart from applying (15) at one step and then (12) at the next step, we are ready to obtain fully explicit expressions for the online natural gradient that do not refer to manifold points ϑ or abstract tensors. The structure is closer to the extended Kalman filter.

Lemma 7. *Denote*

$$\psi_t := \Phi_{t+1} \circ \Phi_t^{-1}, \qquad \Psi_t := \left. \frac{\partial \psi_t(\theta)}{\partial \theta} \right|_{\theta=\theta_t} \tag{40}$$

the change of chart from t to $t+1$, and its derivative. Also denote $p_t(y|\theta) := \mathfrak{p}_t(y|\Phi_t^{-1}(\theta))$ as in Lemma 6 above.

Then the online natural gradient descent in the charts Φ_t is equivalent to

$$\theta \leftarrow \psi_{t-1}(\theta_{t-1}) \tag{41}$$

$$J \leftarrow (\Psi_{t-1}^{-1})^{\top} J_{t-1} \, \Psi_{t-1}^{-1} \tag{42}$$

$$J_t \leftarrow (1-\gamma_t)J + \gamma_t \, \mathbb{E}_{y\sim p_t(y|\theta)} \left[\frac{\partial \ln p_t(y|\theta)}{\partial \theta}^{\otimes 2} \right] \tag{43}$$

$$\theta_t \leftarrow \theta + \eta_t \, J_t^{-1} \frac{\partial \ln p_t(y_t|\theta)}{\partial \theta}^{\top} \tag{44}$$

Proof.
Consider the effect of following Definition 2: we apply (15) at one step and then (12) at the next step. Namely, we go from chart Φ_{t-1} to Φ_t. The transformation rule (41) for θ is a direct consequence of this. Similarly, the transformation rule (42) for J follows from the change of coordinate formula for a $(0,2)$-tensor, given in Lemma 28 in Appendix B, when going from chart Φ_{t-1} to Φ_t. The rest is copied from Lemma 6. □

4.2 Online Natural Gradient on Trajectories of a Dynamical System

We now specialize these results to the main situation considered in this text, that of observations of a dynamical system.

Let us translate Lemma 7 in this setting. We first need to explicit the function $\psi_t = \Phi_{t+1} \circ \Phi_t^{-1}$ and its derivative Ψ_t.

Lemma 8. *In the setting above, for any time $t \geqslant 1$ and state $s \in \mathbb{R}^{\dim(s)}$ we have*

$$\Phi_t(\Phi_{t-1}(s)) = f(s, u_t) \tag{45}$$

Proof.
Indeed, $\Phi_{t-1}(s)$ maps s to the trajectory $\mathbf{s} \in \mathcal{S}$ whose state at time $t-1$ is s. Then the state at time t of $\mathbf{s}$ is $f(s, u_t)$ by definition of $\mathcal{S}$. □

It is then immediate to translate Lemma 7 in this setting. Remember that we apply this lemma to $\Theta = \mathcal{S}$, $\theta_t = s_t$ and $\theta = s_{t|t-1}$ by definition.

Corollary 9 (Explicit form of the online natural gradient for a dynamical system). *The online natural gradient descent for the dynamical system above, in the sequence of charts Φ_t, is equivalent to*

$$s_{t|t-1} \leftarrow f(s_{t-1}, u_t) \tag{46}$$

$$F_{t-1} \leftarrow \frac{\partial f(s_{t-1}, u_t)}{\partial s_{t-1}} \tag{47}$$

$$\hat{y}_t \leftarrow h(s_{t|t-1}, u_t) \tag{48}$$

$$J \leftarrow (F_{t-1}^{-1})^\top J_{t-1} F_{t-1}^{-1} \tag{49}$$

$$J_t \leftarrow (1-\gamma_t) J + \gamma_t \, \mathbb{E}_{y \sim p_{\mathrm{obs}}(y|\hat{y}_t)} \left[\frac{\partial \ln p_{\mathrm{obs}}(y|\hat{y}_t)}{\partial s_{t|t-1}}^{\otimes 2} \right] \tag{50}$$

$$s_t \leftarrow s_{t|t-1} + \eta_t \, J_t^{-1} \frac{\partial \ln p_{\mathrm{obs}}(y_t|\hat{y}_t)}{\partial s_{t|t-1}}^\top \tag{51}$$

where the last expressions depend on $s_{t|t-1}$ via $\hat{y}_t = h(s_{t|t-1}, u_t)$.

Proof.
By Lemma 8, the function $\psi_{t-1} = \Phi_t \circ \Phi_{t-1}^{-1}$ appearing in Lemma 7 is $f(\cdot, u_t)$. Therefore, its derivative Ψ_{t-1} at point $\theta_{t-1} = s_{t-1}$ is

$$\Psi_{t-1} = \frac{\partial f(s_{t-1}, u_t)}{\partial s_{t-1}} \tag{52}$$

This provides the updates for $s_{t|t-1}$ and for J in the statement.

Next, the probability distribution $p_t(y|\theta)$ appearing in Lemma 7 is $\mathfrak{p}_t(y|\Phi_t^{-1}(\theta))$ by definition. Here $\theta = s_{t|t-1}$. In our situation, $\mathfrak{p}$ is defined by (33) namely $\mathfrak{p}_t(y|\mathbf{s}) = p_{\text{obs}}(y|h(\Phi_t(\mathbf{s}), u_t))$. Therefore, we obtain

$$\begin{aligned} p_t(y|\theta) &= \mathfrak{p}_t(y|\Phi_t^{-1}(s_{t|t-1})) && (53)\\ &= p_{\text{obs}}(y|h(\Phi_t(\Phi_t^{-1}(s_{t|t-1})), u_t)) && (54)\\ &= p_{\text{obs}}(y|h(s_{t|t-1}, u_t)) && (55)\\ &= p_{\text{obs}}(y|\hat{y}_t) && (56) \end{aligned}$$

and this ends the proof. □

4.3 The Kalman State Update as a Gradient Step

Here we recall some results from [Oll18] on the Kalman filter. These results interpret the update step in the Kalman filter as a gradient descent step preconditioned by the covariance matrix P, and make the relationship with the Fisher information matrix of the observation model p_{obs}.

This relies on the output noise model $p_{\text{obs}}(y|\hat{y})$ being an exponential family. This is satisfied in the most common case, when the model for y is Gaussian with mean $\hat{y}$, but also for other types of model, such as categorical outputs where $\hat{y}$ is the vector of probabilities of all classes.

The following statement is Proposition 6 in [Oll18].

Proposition 10 (Kalman filter as preconditioned gradient descent). *The update of the state s in a Kalman filter can be seen as an online gradient descent on data log-likelihood, with preconditioning matrix P_t. More precisely, the update (26) is equivalent to*

$$s_t = s_{t|t-1} + P_t \left(\frac{\partial \ln p_{\text{obs}}(y_t|\hat{y}_t)}{\partial s_{t|t-1}} \right)^\top \tag{57}$$

where this expression depends on $s_{t|t-1}$ via $\hat{y}_t = h(s_{t|t-1}, u_t)$.

The next proposition is known as the *information filter* in the Kalman filter literature, and states that the observation step for P is additive when considered on P^{-1} (see [Sim06, (6.33)] or Lemma 9 in [Oll18])

Lemma 11 (Information filter). *The update (24)–(25) of P_t in the extended Kalman filter is equivalent to*

$$P_t^{-1} \leftarrow P_{t|t-1}^{-1} + H_t^\top R_t^{-1} H_t \tag{58}$$

(assuming $P_{t|t-1}$ and R_t are invertible).

The next result (Lemma 10 from [Oll18]) states that after each observation, the Fisher information matrix of the latest observation is added to P^{-1}.

Lemma 12. *For exponential families $p_{\text{obs}}(y|\hat{y})$, the term $H_t^\top R_t^{-1} H_t$ appearing in Lemma 11 is equal to the Fisher information matrix of y with respect to the state s,*

$$H_t^\top R_t^{-1} H_t = \mathbb{E}_{y\sim p_{\text{obs}}(y|\hat{y}_t)}\left[\frac{\partial \ln p_{\text{obs}}(y|\hat{y}_t)}{\partial s_{t|t-1}}^{\otimes 2}\right] \tag{59}$$

where this expression depends on $s_{t|t-1}$ via $\hat{y}_t = h(s_{t|t-1}, u_t)$.

By collecting these results into the definition of the Kalman filter, one gets the following reformulation, which brings it closer to a natural gradient.

Corollary 13. *The extended Kalman filter can be rewritten as*

$$s_{t|t-1} \leftarrow f(s_{t-1}, u_t) \tag{60}$$

$$F_{t-1} \leftarrow \left.\frac{\partial f}{\partial s}\right|_{(s_{t-1},u_t)} \tag{61}$$

$$P_{t|t-1} \leftarrow F_{t-1} P_{t-1} F_{t-1}^\top + Q_t \tag{62}$$

$$\hat{y}_t \leftarrow h(s_{t|t-1}, u_t) \tag{63}$$

$$P_t^{-1} \leftarrow P_{t|t-1}^{-1} + \mathbb{E}_{y\sim p_{\text{obs}}(y|\hat{y}_t)}\left[\frac{\partial \ln p_{\text{obs}}(y|\hat{y}_t)}{\partial s_{t|t-1}}^{\otimes 2}\right] \tag{64}$$

$$s_t \leftarrow s_{t|t-1} + P_t \left(\frac{\partial \ln p_{\text{obs}}(y_t|\hat{y}_t)}{\partial s_{t|t-1}}\right)^\top \tag{65}$$

where the last expressions depend on $s_{t|t-1}$ via $\hat{y}_t = h(s_{t|t-1}, u_t)$.

In the pure fading-memory case, the update for $P_{t|t-1}$ is $P_{t|t-1} \leftarrow (1+\alpha_t) F_{t-1} P_{t-1} F_{t-1}^\top$. In that situation, comparing this rephrasing of the Kalman filter with the explicit form of the natural gradient in Corollary 9 makes it clear that J_t is proportional to the inverse of P_t. This is made precise in the following statement.

Proposition 14. *The Kalman algorithm in Corollary 13 in the pure fading-memory case, and the natural gradient algorithm in Corollary 9, are identical under the identification*

$$P_t = \eta_t J_t^{-1} \tag{66}$$

provided the hyperparameters η_t, γ_t and α_t satisfy the following relations:

$$\eta_t = \gamma_t, \qquad \frac{1}{\eta_t} = \frac{1}{1+\alpha_t}\frac{1}{\eta_{t-1}} + 1 \tag{67}$$

In particular, if these algorithms are initialized at the same point (same s_0, and $P_0 = \eta_0 J_0^{-1}$), they will remain identical at all times.

Proof.
Define $\tilde{J}_t := \eta_t P_t^{-1}$; we want to show that $\tilde{J}_t$ follows the same evolution equation as J_t.

If this holds, then the update of s_t will be identical in the two algorithms, since one uses P_t and the user uses $\eta_t J_t^{-1}$ to precondition the gradient.

From Corollary 13 in the pure fading-memory case we get

$$\tilde{J}_t = \eta_t P_t^{-1} \tag{68}$$

$$= \eta_t P_{t|t-1}^{-1} + \eta_t \mathbb{E}_{y\sim p_{\text{obs}}(y|\hat{y}_t)} \left[\frac{\partial \ln p_{\text{obs}}(y|\hat{y}_t)}{\partial s_{t|t-1}}^{\otimes 2} \right] \tag{69}$$

$$= \frac{\eta_t}{1+\alpha_t} (F_t^{-1})^\top P_{t-1}^{-1} F_t^{-1} + \eta_t \mathbb{E}_{y\sim p_{\text{obs}}(y|\hat{y}_t)} \left[\frac{\partial \ln p_{\text{obs}}(y|\hat{y}_t)}{\partial s_{t|t-1}}^{\otimes 2} \right] \tag{70}$$

$$= \frac{\eta_t}{1+\alpha_t} (F_t^{-1})^\top \frac{\tilde{J}_{t-1}}{\eta_{t-1}} F_t^{-1} + \eta_t \mathbb{E}_{y\sim p_{\text{obs}}(y|\hat{y}_t)} \left[\frac{\partial \ln p_{\text{obs}}(y|\hat{y}_t)}{\partial s_{t|t-1}}^{\otimes 2} \right] \tag{71}$$

while the full update for J_t in Cor. 9 is

$$J_t = (1-\gamma_t)(F_t^{-1})^\top J_{t-1} F_t^{-1} + \gamma_t \mathbb{E}_{y\sim p_{\text{obs}}(y|\hat{y}_t)} \left[\frac{\partial \ln p_{\text{obs}}(y|\hat{y}_t)}{\partial s_{t|t-1}}^{\otimes 2} \right] \tag{72}$$

Thus, the two updates coincide if

$$\gamma_t = \eta_t, \qquad 1 - \eta_t = \frac{\eta_t}{(1+\alpha_t)\eta_{t-1}} \tag{73}$$

and in this case, if the algorithms are identical at time $t-1$ then they will be identical at time t. This ends the proof of the proposition and of Theorem 5. □

5 Continuous-Time Case: the Kalman–Bucy Filter as a Natural Gradient

Consider now a continuous-time model of a dynamical system with state s and control or input u_t, with evolution equation

$$\dot{s}_t = f(s_t, u_t), \tag{74}$$

and we want to learn the current state of the system from observations y_t. As before, the observations are modeled via an observation function $h(s_t, u_t)$ plus noise,

$$y_t = h(s_t, u_t) + W_t \tag{75}$$

where W_t is a white noise process with known covariance matrix R_t.

The continuous-time analogue of the extended Kalman filter for this situation is the extended Kalman–Bucy filter, which can be described [Wik]

by the two evolution equations (which mix the transition and the observation steps of the discrete-time case)

$$\dot{s}_t = f(s_t, u_t) + K_t(y_t - h(s_t, u_t)) \tag{76}$$

$$\dot{P}_t = F_t P_t + P_t F_t^\top - K_t H_t P_t + Q_t \tag{77}$$

where

$$F_t := \frac{\partial f(s_t, u_t)}{\partial s_t}, \qquad H_t := \frac{\partial h(s_t, u_t)}{\partial s_t}, \qquad K_t := P_t H_t^\top R_t^{-1} \tag{78}$$

Here Q_t is the covariance of the noise used to model the uncertainty on the transitions of the system (for instance, if f is not known exactly), namely $\mathrm{d}s_t = f(s_t, u_t)\,\mathrm{d}t + \mathrm{d}B_t$ with B_t a Brownian motion with covariance matrix Q_t.

As in the discrete case, we will work with the *pure fading-memory* variant of the extended Kalman–Bucy filter, which assumes

$$Q_t = \alpha_t P_t \tag{79}$$

where $\alpha_t \geqslant 0$ is a hyperparameter. This choice of Q_t is canonical in the absence of further information on the system.

We will recover this filter fully in the course of proving Theorem 17 below, by starting with an abstract definition of the continuous-time online natural gradient, and making it explicit until we end up with the Kalman–Bucy filter.

The proof reveals a feature of the Kalman–Bucy filter: namely, to properly define it in a manifold setting, a choice of covariant derivative is needed to transfer the covariance matrix P_t at the current point s_t, to a new covariance matrix at $s_{t+\mathrm{d}t}$; in a manifold this is a non-trivial operation (for the consequences for Kalman filtering, see for instance the discussion and Fig. 9 in the review [BB18]). This results from the need to keep the algorithm online, and not recompute the Fisher matrix of past observations when the parameter is updated.

On the other hand, the evolution of the state s_t does not depend explicitly on a covariant derivative (or choice of chart), contrary to the discrete-time case: in the discrete-time case, a change of chart influences the update of s_t only at second order in the learning rate, and this disappears in continuous time because the learning rates become infinitesimal.

Natural gradient in continuous time. The statistical learning viewpoint on this problem is as follows: Each trajectory $\mathbf{s} = (s_t)_{t\geqslant 0}$ defines a probability distribution on generalized functions[5] $\mathbf{y} = (y_t)_{t\geqslant 0}$ in the Wiener space via the model (75).

[5]We will call *generalized functions* the elements of the Wiener space, i.e., a functional space in which samples of the white noise live. These can be seen, for instance, as random

The trajectories $\mathbf{s}$ of the system may be parameterized via their initial state s_0. Then the problem of estimating $\mathbf{s}$ from the observations $\mathbf{y}$ becomes a standard statistical estimation problem of estimating s_0, and methods such as the natural gradient may be applied to optimize s_0 knowing the observations.

Definition 15 (Observation model, instantaneous Fisher matrix, instantaneous log-likelihood). *We call* observation model *parameterized by θ in some manifold Θ, a probability distribution on generalized functions $\mathbf{y} = (y_t)_{t\in[0;T]}$ over $[0;T]$, which is absolutely continuous with density $p(\mathbf{y}|\theta)$ with respect to the Wiener measure.*

The Fisher information matrix of this model over $[0;T]$ is

$$J_{[0;T]}(\theta) := \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\theta)} \frac{\partial \ln p(\mathbf{y}|\theta)}{\partial\theta}^{\otimes 2} \tag{80}$$

if this quantity exists.

We define the instantaneous *Fisher information matrix to capture the amount of information brought by y_t at instant t, as*

$$j_t(\theta) := \frac{\mathrm{d}}{\mathrm{d}t} J_{[0;t]}(\theta) \tag{81}$$

and the instantaneous log-likelihood *of $\mathbf{y} = (y_t)_{t\in[0;T]}$, which captures the likelihood of y_t knowing the model, as*

$$\ell_t(\theta) := \frac{\mathrm{d}}{\mathrm{d}t} \ln p(y_{[0:t]}|\theta) \tag{82}$$

provided these derivatives exist.

The instantaneous log-likelihood ℓ_t is the continuous-time analogue of the log-likelihood of the current observation y_t used to update the parameter in Definition 1. Intuitively ℓ_t is equal to $\ln p(y_{[t;t+\mathrm{d}t]}|\theta, y_{[0;t)})$. We will formalize this intuition in Proposition 19 and Corollary 20 below: this corollary shows that for the model $y_t = h(s_t, u_t) + W_t$, the gradient of this log-likelihood is given by the error $y_t - h(s_t, u_t)$.

The instantaneous Fisher matrix j_t is the continuous-time analogue of the Fisher information matrix on a single observation $y \sim p(y|u_t, \theta)$ at time t used in Definition 1.[6]

distributions against which functions can be integrated. Namely, the white noise definition states that if w_t is sampled from a real-valued white noise with unit variance, then for every deterministic function f, the integral $\int f(t)\, w_t \,\mathrm{d}t$ is Gaussian with variance $\int f(t)^2 \,\mathrm{d}t$ [Jaz70, Thm 4.1]. Intuitively, the white noise w_t takes values $(1/\sqrt{\mathrm{d}t})\mathcal{N}(0,1)$ in each infinitesimal interval of size $\mathrm{d}t$. If $B_t = \int w_t \,\mathrm{d}t$ is the Brownian motion with derivative w_t, then $\int f(t)\, w_t \,\mathrm{d}t$ is the same as the Itô integral $\int f(t)\,\mathrm{d}B_t$. For the vector-valued case: if w_t has covariance matrix R_t, then for each vector-valued f the integral $\int w_t^\top f(t)\,\mathrm{d}t$ is Gaussian with variance $\int f(t)^\top R_t f(t)\,\mathrm{d}t$.

[6] Intuitively this is equal to

$$\mathrm{d}t\, \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\theta)} \frac{\partial \ln p_t(y_t|\theta)}{\partial\theta}^{\otimes 2} \tag{83}$$

The continuous-time analogue of the online natural gradient descent (Def. 1) may be defined as follows.

Definition 16 (Online natural gradient in continuous time). *Let $\mathbf{y} = (y_t)_{t\geqslant 0}$ be a continuous function of time. Let D be a covariant derivative on the manifold Θ. Given an observation model as above, we define the* online natural gradient *for learning θ based on the observations $\mathbf{y}$, as the solution of*

$$\frac{DJ_t}{\mathrm{d}t} = -\gamma_t J_t + \gamma_t\, j_t(\theta_t) \tag{84}$$

$$\dot\theta_t = \eta_t J_t^{-1} \frac{\partial \ell_t(\mathbf{y}|\theta)^\top}{\partial\theta} \tag{85}$$

initialized at some $\theta_0 \in \Theta$ with some positive definite metric tensor J_0.

The term $-\gamma_t J_t$ in the equation introduces a decay factor on J as in the discrete case.

On the covariant derivative D in the online natural gradient. The covariant derivative D is the continuous-time analogue of the choice of charts at each time t used in the discrete case. In the continuous-time case, it is needed only for J_t, not for θ_t. Indeed, (85) is a well-defined ordinary differential equation in the manifold Θ, whose right-hand term is a tangent vector at $\theta_t \in \Theta$ (see Lemma 27). [7]

On the Kalman filter side of the correspondence, the need to introduce a covariant derivative or coordinate system for J corresponds to the fact that the Kalman covariance matrix $P_{t|t-1}$ is translated from $s_{t|t-1}$ to s_t in the Kalman filter; this translation makes no sense in a Riemannian manifold.

A canonical choice for D would be the Levi-Civita covariant derivative associated with the metric $J(\theta)$. However, this does not result in a convenient algorithm. In the Kalman–Bucy filter, D turns out to be the covariant derivative associated with the chart s_t at time t; in particular, this D is time-dependent.

The *non-online* natural gradient would use

$$\dot\theta_t = \eta_t J(\theta_t)^{-1} \frac{\partial \ln p_t(y_t|\theta)^\top}{\partial\theta} \tag{86}$$

instead, which does not depend on a choice of covariant derivative. However, the Fisher matrix $J(\theta)$ is an average over the time interval $[0;t]$ (from the

which is formally closer to Def. 1; but this latter expression is not fully rigorous because samples $\mathbf{y} \sim p(\mathbf{y}|\theta)$ include white noise and thus have infinite values of y_t, which are compensated by the $\mathrm{d}t$ factor. This is why we use the rigorous expression (84) instead.

[7] We have assumed that the observations y_t are ordinary functions, not elements of the Wiener space; in the latter case, (85) would become a stochastic differential equation, requiring particular treatment to make it parameterization-independent in the manifold.

statistical learning point of view, the Fisher matrix is an expectation over inputs u_t): using $J(\theta_t)$ would necessitate to recompute an integral over the past for each new value of θ_t. Instead, the online version reuses values computed at previous times instead of recomputing the full Fisher matrix $J(\theta_t)$ for new values of θ_t. This is why some way of transferring J from previous values of θ_t to the current one is needed.

Thus, the appearance of a covariant derivative in Definition 16 results from the need for a convenient *online* algorithm.

The Kalman–Bucy filter as a natural gradient. The correspondence between the online natural gradient and the Kalman–Bucy filter is expressed as follows.

Theorem 17. *Consider a continuous-time dynamical system with state $s_t \in \mathbb{R}^{\dim(s)}$, inputs $u_t \in \mathbb{R}^{\dim(u)}$ and dynamics f, namely,*

$$\dot{s}_t = f(s_{t-1}, u_t) \tag{87}$$

where f is a smooth function from $\mathbb{R}^{\dim(s)} \times \mathbb{R}^{\dim(u)}$ to $\mathbb{R}^{\dim(s)}$. We assume that the solutions are regular on some time interval $[0; T]$ for some open domain of initial conditions $s_0 \in \mathbb{R}^{\dim(s)}$. Define the prediction model

$$y_t = h(s_t, u_t) + W_t \tag{88}$$

where h is a smooth function from $\mathbb{R}^{\dim(s)} \times \mathbb{R}^{\dim(u)}$ to $\mathbb{R}^{\dim(y)}$, and W_t is a white noise process with covariance matrix R_t.

Let Θ be the set of trajectories of the system, parameterized by their initial condition $\theta = s_0$. Thus, each $\theta \in \Theta$ defines a trajectory $s_t(\theta)$ and a probability distribution on observations $\mathbf{y} = (y_t)_{t\in[0;T]}$ via (88). For each time $t \geqslant 0$, let D^t be the covariant derivative on Θ associated with the chart $\theta \mapsto s_t(\theta)$ (namely, the covariant derivative D^t of a tensor is equal to its ordinary derivative when expressed in the chart s_t).

Let $(y_t)_{t\in[0;T]}$ be a smooth series of observations. Let θ_t be the trajectory at time t inferred by the online natural gradient (Def. 16) with observations (y_t), where the covariant derivative used at time t is D^t.

Then the state $s_t(\theta_t)$ inferred by the natural gradient at time t, is the same as the state s_t inferred at time t by the Kalman–Bucy filter with pure fading memory ($Q_t = \alpha_t P_t$ in the Kalman–Bucy equations), provided both are initialized at the same state s_0, with Kalman–Bucy initial covariance $P_0 = \eta_0 J_0^{-1}$, and provided the hyperparameters are related via

$$\gamma_t = \eta_t, \qquad \dot{\eta}_t = \alpha_t \eta_t - \eta_t^2 \tag{89}$$

Moreover, the Kalman–Bucy posterior covariance is related to the expression of the Fisher matrix J_t in chart s_t via $P_t = \eta_t J_t^{-1}$.

Let us comment once more on the hyperparameter settings. First, the extended Kalman–Bucy filter (with fading memory $Q_t = \alpha_t P_t$) is recovered as an online natural gradient with parameters $\gamma_t = \eta_t$ for the same reasons as in the discrete case.

Second, the equation on η_t is satisfied, for instance, if $\eta_t = \alpha_t$ for all t. Other solutions exist: solutions are better found by writing the equation on $1/\eta_t$ instead of η_t. For instance, the full-memory, noiseless case $\alpha_t = 0$ corresponds to the learning rate $\eta_t = 1/(t + \mathrm{cst})$, as in the discrete case.

6 Proofs for Continuous Time

The proof proceeds by working out more and more explicit expressions for the natural gradient, until we end up with the Kalman–Bucy filter.

We first compute an explicit form for the instantaneous log-likelihood and its gradient, for the case of the model $y_t = h(s_t, u_t) + W_t$. This is mostly a direct application of the Cameron–Martin theorem [CM44], and relates the gradient of the instantaneous loss ℓ_t to the error $y_t - h(s_t, u_t)$ at time t.

Theorem 18 (Cameron–Martin theorem). *Let h be a smooth real-valued function on $[0;T]$. Let W_t be a white noise on $[0;T]$. Let $\mathcal{W}$ be the Wiener measure (the law of W_t in the Wiener space). Let $\mathcal{W}_h$ be the Wiener measure translated by h, namely, the distribution of $h + W$ in the Wiener space. Then $\mathcal{W}_h$ is absolutely continuous with respect to $\mathcal{W}$, and its density at a function $\mathbf{y} = y(t)$ is*

$$\frac{\mathrm{d}\mathcal{W}_h}{\mathrm{d}\mathcal{W}}(\mathbf{y}) = \exp\left(\int_{[0;T]} y(t)h(t)\,\mathrm{d}t - \frac{1}{2}\int_{[0;T]} h(t)^2\,\mathrm{d}t\right) \tag{90}$$

Proof.
This is a rephrasing of Theorem 1 in [CM44]; the statement given here can be found as Theorem 1.2 in [Kuo75] applied to the abstract Wiener space on $L^2([0;T])$ with variance $t = 1$.

At an informal level, $y := h + W$ is a Gaussian centered at h while the white noise is a Gaussian centered at 0, so informally the ratio of the probability densities is the ratio of these two Gaussians,

$$\exp\left(-\frac{1}{2}\int_{t=0}^{T}(y(t) - h(t))^2\,\mathrm{d}t\right)\exp\left(\frac{1}{2}\int_{t=0}^{T} y(t)^2\,\mathrm{d}t\right) \tag{91}$$

but rigorously, the quantity $\int_{t=0}^{T} y(t)^2\,\mathrm{d}t$ is infinite under the white noise distribution. However, this quantity cancels out between the two parts of the expression, resulting in the Cameron–Martin theorem and the expression (90).

More rigorously, let E be a measurable set in the abstract Wiener space over $L^2([0;T])$. Then $\mathcal{W}_h(E) = \mathcal{W}(E - h)$ by definition of $\mathcal{W}_h$. Therefore, we can apply Theorem 1.2 in [Kuo75] to $-h$, which gives the result.

(Note that we express everything over the white noise W instead of the Brownian motion $B = \int W$, so the norm we use for the Wiener space is indeed the L^2 norm instead of the square norm of derivatives as found for instance in Theorem 1.1 of [Kuo75].) □

Proposition 19 (Instantaneous log-likelihood). *Let $s_t(\theta)$ be a set of trajectories smoothly parameterized by $\theta \in \Theta$. Consider the probability distribution on generalized functions $\mathbf{y}$ defined by (75), namely, $y_t = h(s_t(\theta, u_t)) + W_t$ with W_t a white noise with covariance matrix R_t.*

Then this probability distribution has a density $p(\mathbf{y}|\theta)$ with respect to the Wiener measure. For continuous functions $\mathbf{y}$, this density satisfies

$$\ln p(\mathbf{y}|\theta) = \int_{t=0}^{T} \ln p_t(y_t|\theta)\,\mathrm{d}t \tag{92}$$

where

$$\ln p_t(y_t|\mathbf{s}) := y_t^\top R_t^{-1} h(s_t, u_t) - \frac{1}{2} h(s_t, u_t)^\top R_t^{-1} h(s_t, u_t) \tag{93}$$

Consequently, the instantaneous log-likelihood of this model is equal to $\ell_t(\mathbf{y}|\theta) = \ln p_t(y_t|\mathbf{s})$.

(The probability density $p_t(y_t|\theta)$ does not sum to 1 over y_t because p_t is not a probability but a probability density wrt the Wiener measure; the Wiener measure contains the Gaussian factor on y_t.)

Proof.
The probability distribution on $\mathbf{y}$ is, by definition, a white noise centered at $h(s_t(\theta), u_t)$, with covariance matrix R_t. After changing variables by $R_t^{-1/2}$, we can assume without loss of generality that $R_t = \mathrm{Id}$. Then, by applying Thm. 18 in the vector-valued case, we find that the density of the law of $\mathbf{y}$ with respect to the Wiener measure is

$$p(\mathbf{y}|\mathbf{s}) = \exp\left(\int_{t=0}^{T} y_t^\top R_t^{-1} h(s_t, u_t)\,\mathrm{d}t - \frac{1}{2}\int_{t=0}^{T} h(s_t, u_t)^\top R_t^{-1} h(s_t, u_t)\,\mathrm{d}t\right) \tag{94}$$

which proves the claim. □

This immediately provides an explicit form for the parameter update (85) in the definition of the online natural gradient.

Corollary 20 (Gradient of instantaneous log-likelihood). *Let $\mathbf{s} = (s_t)_{t\in[0;T]}$ be a set of trajectories smoothly parameterized by $\theta \in \Theta$. Consider the associated observation model (75) as above.*

Then the natural gradient parameter update (85) for this model satisfies

$$\frac{\partial \ell_t(\mathbf{y}|\theta)}{\partial \theta} = (y_t - h(s_t(\theta), u_t))^\top R_t^{-1} \frac{\partial h(s_t, u_t)}{\partial \theta} \tag{95}$$

In particular, the gradient step will try to change the value of $h(s_t, u_t)$ to reduce the error $y_t - h(s_t, u_t)$, as expected.

Proof.
This is a direct consequence of (93). □

We now turn to the expression for the Fisher matrix.

Proposition 21 (Instantaneous Fisher matrix). *Let $\mathbf{s} = (s_t)_{t\in[0;T]}$ be a set of trajectories smoothly parameterized by $\theta \in \Theta$, and consider the observation model $y_t = h(s_t, u_t) + W_t$ with W_t a white noise with covariance R_t. Denote*

$$G_t := \frac{\partial s_t(\theta)}{\partial \theta}, \qquad H_t := \frac{\partial h(s_t, u_t)}{\partial s_t} \tag{96}$$

Then the Fisher matrix (80) for this model is

$$J_{[0:T]}(\theta) = \int_0^T G_t^\top H_t^\top R_t^{-1} H_t G_t \,\mathrm{d}t \tag{97}$$

and in particular, the instantaneous Fisher matrix (81) is equal to

$$j_t(\theta) = G_t^\top H_t^\top R_t^{-1} H_t G_t \tag{98}$$

In particular, if the trajectories are parameterized by their state s_t at time t then $G_t = \mathrm{Id}$ (for this particular t), and this is the continuous-time analogue of Lemma 11.

Lemma 22. *Let W_t be a vector-valued white noise on an interval $[0; T]$, with covariance matrix R_t. Let $f(t)$ and $g(t)$ be two vector-valued deterministic functions on $[0; T]$. Then*

$$\mathbb{E}\left[\left(\int_0^T W_t^\top f(t)\,\mathrm{d}t\right)\left(\int_0^T W_t^\top g(t)\,\mathrm{d}t\right)\right] = \int_0^T f(t)^\top R_t\, g(t)\,\mathrm{d}t \tag{99}$$

(where the integrals are in the Wiener or Itô sense).

Proof of Lemma 22.
First, consider the case of a real-valued white noise w_t with unit variance. The integral $\int f(t) w_t \,\mathrm{d}t$ is equal to $\int f(t)\,\mathrm{d}B_t$ where B_t is the Brownian motion whose derivative is w_t (namely $\mathrm{d}B_t = w_t\,\mathrm{d}t$). It is known [Jaz70, (4.23 for deterministic f and g)] that

$$\mathbb{E}\left[\left(\int_0^T f(t)\,\mathrm{d}B_t\right)\left(\int_0^t g(t)\,\mathrm{d}B_t\right)\right] = \int_0^T f(t)g(t)\,\mathrm{d}t \tag{100}$$

which gives the result for dimension 1 and unit variance.

Now a vector-valued white noise with covariance matrix R_t can be written as $W_t = R_t^{1/2}(w_t^1, \ldots, w_t^n)^\top$ where the w_t^i are independent real-valued white noises with unit variance. The result follows by applying the above to $R_t^{1/2} f(t)$ and $R_t^{1/2} g(t)$ and summing over components. □

Proof of Proposition 21.
The Fisher matrix for this model is, by (80) and (92),

$$J_{[0;t]} = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\theta)}\left[\left(\frac{\partial}{\partial\theta}\int_0^T \ln p_t(y_t|\theta)\,\mathrm{d}t\right)^{\otimes 2}\right] \tag{101}$$

and the expression (93) for $\frac{\partial}{\partial\theta}\ln p_t(y_t|\theta)$ yields

$$J_{[0;t]} = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\theta)}\left[\left(\int_0^T (y_t - h(s_t,u_t))^\top R_t^{-1}\,\frac{\partial h(s_t,u_t)}{\partial\theta}\,\mathrm{d}t\right)^{\otimes 2}\right] \tag{102}$$

Now, the model p_t was derived from the observation model (75): under this model, $y_t = h(s_t,u_t) + W_t$ with W_t a white noise with covariance R_t. Therefore, $y_t - h(s_t,u_t) = W_t$ and

$$J_{[0;t]} = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\theta)}\left[\left(\int_0^T W_t^\top R_t^{-1}\,\frac{\partial h(s_t,u_t)}{\partial\theta}\,\mathrm{d}t\right)^{\otimes 2}\right] \tag{103}$$

Now we can apply Lemma 22 to the components of the derivative with respect to θ, namely

$$f(t) = R_t^{-1}\frac{\partial h(s_t,u_t)}{\partial\theta_i} \tag{104}$$

and

$$g(t) = R_t^{-1}\frac{\partial h(s_t,u_t)}{\partial\theta_j} \tag{105}$$

and we find that the (i,j) entry of the Fisher matrix is

$$\int_0^T \frac{\partial h(s_t,u_t)}{\partial\theta_i}^\top R_t^{-1}\frac{\partial h(s_t,u_t)}{\partial\theta_j}\,\mathrm{d}t \tag{106}$$

hence the result. □

By putting these two results together, we get a more explicit form of the natural gradient for sets of trajectories.

Corollary 23. *Let $\mathbf{s} = (s_t)_{t\in[0;T]}$ be a set of trajectories smoothly parameterized by $\theta\in\Theta$, and consider the observation model $y_t = h(s_t,u_t)+W_t$ with W_t a white noise with covariance R_t (namely, Def. 15 with $p_t(y_t|\theta)$ given by (93)). Denote*

$$G_t := \frac{\partial s_t(\theta)}{\partial\theta}, \qquad H_t := \frac{\partial h(s_t,u_t)}{\partial s_t} \tag{107}$$

Let $\mathbf{y} = (y_t)_{t\in[0;T]}$ be a smooth function. Then the online natural gradient (Def. 16) for this model with observations $\mathbf{y}$ satisfies

$$\frac{D^t J_t}{\mathrm{d}t} = -\gamma_t J_t + \gamma_t\, G_t^\top H_t^\top R_t^{-1} H_t G_t \tag{108}$$

$$\dot\theta_t = \eta_t\, J_t^{-1} G_t^\top H_t^\top R_t^{-1}(y_t - h(s_t(\theta_t), u_t)) \tag{109}$$

where in these expressions, G_t is evaluated at θ_t and H_t at $s_t(\theta_t)$, and where D^t is the covariant derivative associated with the chart $\theta \mapsto s_t(\theta)$.

This result is still somewhat non-explicit due to the covariant derivative D^t. This will disappear by using s_t rather than θ as the parameterization of the trajectories at each time; this is more consistent with an algorithmic implementation at time t, and with the form of the Kalman–Bucy filter.

Assume that $\theta \mapsto s_t(\theta)$ is indeed a chart, namely, that s_t is smooth and one-to-one on its domain with smooth inverse. (This is the case under the assumptions of Thm. 17: then $s_t(\theta)$ is the solution of an ordinary differential equation with initial condition $\theta = s_0$, and if the function f defining the equation is regular, then the mapping from s_{t_1} to s_{t_2} is a diffeomorphism on its domain.) This implies that $G_t = \partial s_t(\theta)/\partial\theta$ is invertible.

Let $J_{t\downarrow t_0} := \mathbf{T}s_{t_0}(J_t)$ be the expression of J_t in chart s_{t_0}. Since J_t is a $(0,2)$-tensor at θ_t we have

$$J_{t\downarrow t_0} := \mathbf{T}s_{t_0}(J_t) = (G_{t_0}(\theta_t)^{-1})^\top J_t G_{t_0}(\theta_t)^{-1} \tag{110}$$

by Lemma 28 (where we interpret this as a matrix expression by just viewing θ as another chart).

We will be particularly interested in $J_{t\downarrow t}$, which represents the Fisher matrix with respect to the current state s_t rather than θ. For this, we first have to study how $J_{t\downarrow t_0}$ evolves when the reference chart t_0 changes. This works most finely when the trajectories parameterized by θ satisfy a differential equation $\frac{\partial}{\partial t} s_t(\theta) = f_t(s_t(\theta))$ for some f_t, as is the case in the Kalman–Bucy filter.

Lemma 24 (Time-varying charts). *Let $\mathbf{s} = (s_t)_{t\in[0;T]}$ be a set of trajectories smoothly parameterized by $\theta \in \Theta$. Assume that there exists a function $f_t(s)$ such that*

$$\frac{\partial s_t(\theta)}{\partial t} = f_t(s_t(\theta)) \tag{111}$$

and set $F_t(s) := \frac{\partial f_t(s)}{\partial s}$.

Let J be a $(0,2)$-tensor at some $\theta \in \Theta$. Then the expression $J_{\downarrow t_0}$ of J in the chart s_{t_0} evolves as

$$\frac{\mathrm{d}J_{\downarrow t_0}}{\mathrm{d}t_0} = -F_{t_0}^\top J_{\downarrow t_0} - J_{\downarrow t_0} F_{t_0} \tag{112}$$

where F_{t_0} is evaluated at $s_{t_0}(\theta)$.

(Note that $J_{t\downarrow t_0}$ is an expression in coordinates, so we can take its ordinary derivative without needing covariant derivatives.)

This expression is the continuous-time analogue of (49): it shows that the transition update $P_{t|t-1} \leftarrow F_{t-1}P_{t-1}F_{t-1}^\top$ in the discrete-time extended Kalman filter (and likewise the $F_tP_t + P_tF_t^\top$ term in the Kalman–Bucy filter) is just a result of reexpressing the covariance matrix in a coordinate system corresponding to the current state.

Proof.
By the coordinate expression for a $(0,2)$-tensor in chart $s_{t_0}(\theta)$, we have

$$J_{\downarrow t_0} = (G_{t_0}(\theta)^{-1})^\top J G_{t_0}(\theta)^{-1} \tag{113}$$

So we are left with studying the derivative of $G_{t_0}(\theta)$. Using $\frac{\mathrm{d}}{\mathrm{d}t_0}G_t^{-1} = -G_{t_0}^{-1}(\frac{\mathrm{d}}{\mathrm{d}t_0}G_{t_0})G_{t_0}^{-1}$ we find

$$\frac{\mathrm{d}J_{\downarrow t_0}}{\mathrm{d}t_0} = -(\dot{G}_{t_0}G_{t_0}^{-1})^\top J_{\downarrow t_0} - J_{\downarrow t_0}(\dot{G}_{t_0}G_{t_0}^{-1}) \tag{114}$$

where we have abbreviated $\dot{G}_{t_0} = \frac{\mathrm{d}}{\mathrm{d}t_0}G_{t_0}$.

Now, the derivative of $G_t(\theta)$ with respect to t satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}G_t(\theta) = \frac{\partial}{\partial t}\frac{\partial}{\partial\theta}s_t(\theta) = \frac{\partial}{\partial\theta}\frac{\partial}{\partial t}s_t(\theta) \tag{115}$$

$$= \frac{\partial}{\partial\theta}f_t(s_t(\theta)) = \frac{\partial f_t(s_t(\theta))}{\partial s_t}\frac{\partial s_t(\theta)}{\partial\theta} \tag{116}$$

$$= F_t(s_t(\theta))G_t(\theta) \tag{117}$$

or more synthetically, $\dot{G}_t = F_tG_t$. This proves the claim. □

Then the online natural gradient rewrites as follows. Note the similarity with the Kalman–Bucy filter for the state s_t, and for the Fisher matrix $J_{t\downarrow t}$ (which will ultimately be proportional to hte inverse of P_t).

Corollary 25 (Explicit online natural gradient). *Let $\mathbf{s} = (s_t)_{t\in[0;T]}$ be a set of trajectories smoothly parameterized by $\theta \in \Theta$. Assume that there exists a function $f_t(s)$ such that*

$$\frac{\partial s_t(\theta)}{\partial t} = f_t(s_t(\theta)) \tag{118}$$

Consider the observation model $y_t = h(s_t, u_t) + W_t$ with W_t a white noise with covariance R_t (namely, Def. 15 with $p_t(y_t|\theta)$ given by (93)). Denote

$$F_t(s) := \frac{\partial f_t(s)}{\partial s}, \qquad G_t := \frac{\partial s_t(\theta)}{\partial\theta}, \qquad H_t := \frac{\partial h(s_t,u_t)}{\partial s_t} \tag{119}$$

and assume that G_t is invertible.

Let $\mathbf{y} = (y_t)_{t\in[0;T]}$ *be a smooth function. Then the online natural gradient (Def. 16) for this model with observations* $\mathbf{y}$ *is equivalent to*

$$\frac{\mathrm{d}}{\mathrm{d}t} J_{t\downarrow t} = -F_t^\top J_{t\downarrow t} - J_{t\downarrow t} F_t - \gamma_t J_{t\downarrow t} + \gamma_t\, H_t^\top R_t^{-1} H_t \tag{120}$$

$$\dot\theta_t = \eta_t\, G_t^{-1} J_{t\downarrow t}^{-1} H_t^\top R_t^{-1} (y_t - h(s_t(\theta_t), u_t)) \tag{121}$$

initialized with $J_{0\downarrow 0} := (G_0^{-1})^\top J_0 G_0^{-1}$. *In these expressions,* F_t *and* H_t *are evaluated at* $s_t(\theta_t)$ *while* G_t *is evaluated at* θ_t.

Moreover the state $s_t(\theta_t)$ *learned at time* t *satisfies*

$$\frac{\mathrm{d}}{\mathrm{d}t} s_t(\theta_t) = f_t(s_t(\theta_t)) + \eta_t\, J_{t\downarrow t}^{-1} H_t^\top R_t^{-1} (y_t - h(s_t(\theta_t), u_t)) \tag{122}$$

Proof.
Since $J_{t\downarrow t} := (G_t^{-1})^\top J_t G_t^{-1}$, the relationship (121) holds by direct substitution of the equation evolution (109) for θ_t.

By definition, D^{t_0} is the covariant derivative associated with the chart s_{t_0} (Def. 29). Its expression is obtained by going into the chart s_{t_0}, taking ordinary derivatives, and going back, namely (Def. 29),

$$\frac{D^{t_0} J_t}{\mathrm{d}t} = \mathbf{T} s_{t_0}^{-1} \left(\frac{\mathrm{d}}{\mathrm{d}t} J_{t\downarrow t_0} \right) \tag{123}$$

By definition of the online natural gradient with covariant derivative $D = D^t$ at time t, one has, for $t_0 = t$

$$\frac{D^{t_0} J_t}{\mathrm{d}t} = -\gamma_t J_t + \gamma_t j_t(\theta_t) \tag{124}$$

Therefore, at time $t = t_0$, the expression above for D^{t_0} yields

$$\frac{\mathrm{d}}{\mathrm{d}t} J_{t\downarrow t_0} = \mathbf{T} s_{t_0} \left(\frac{D^{t_0} J_t}{\mathrm{d}t} \right) \tag{125}$$

$$= \mathbf{T} s_{t_0} \left(-\gamma_t J_t + \gamma_t j_t(\theta_t) \right) \tag{126}$$

$$= -\gamma_t J_{t\downarrow t_0} + \gamma_t (G_{t_0}^{-1})^\top j_t G_{t_0}^{-1} \tag{127}$$

by definition of $J_{t\downarrow t_0}$ and by the coordinate expression for $(0,2)$-tensors in chart s_{t_0}. Here G_{t_0} and j_t are evaluated at θ_t.

Prop. 21 provides the expression for the instantaneous Fisher matrix j_t,

$$j_t(\theta) = G_t^\top H_t^\top R_t^{-1} H_t G_t \tag{128}$$

so that for $t_0 = t$, we have $(G_{t_0}^{-1})^\top j_t G_{t_0}^{-1} = H_t^\top R_t^{-1} H_t$ and

$$\frac{\mathrm{d}}{\mathrm{d}t} J_{t\downarrow t_0} = -\gamma_t J_{t\downarrow t_0} + \gamma_t H_t^\top R_t^{-1} H_t \tag{129}$$

at $t_0 = t$.

Now we are interested in $J_{t\downarrow t}$; to get its evolution equation we must differentiate with respect to the two instances of t, one of which captures the intrinsic change in J and the other the change of chart:

$$\frac{\mathrm{d}}{\mathrm{d}t} J_{t\downarrow t} = \frac{\mathrm{d}}{\mathrm{d}t} J_{t\downarrow t_0}\Big|_{t_0=t} + \frac{\mathrm{d}}{\mathrm{d}t_0} J_{t\downarrow t_0}\Big|_{t_0=t} \tag{130}$$

We just computed $\frac{\mathrm{d}}{\mathrm{d}t} J_{t\downarrow t_0}\Big|_{t_0=t}$, and $\frac{\mathrm{d}}{\mathrm{d}t_0} J_{t\downarrow t_0}\Big|_{t_0=t}$ is provided by Lemma 24. This provides the full evolution equation (120) for $J_{t\downarrow t}$ in the statement.

To compute the evolution of the state $s_t(\theta_t)$ learned at time t, let us decompose

$$\frac{\mathrm{d}}{\mathrm{d}t} s_t(\theta_t) = \frac{\partial s_t(\theta)}{\partial t}\Big|_{\theta=\theta_t} + \frac{\partial s_t(\theta_t)}{\partial \theta_t}\frac{\partial \theta_t}{\partial t} \tag{131}$$

$$= f_t(s_t(\theta_t)) + G_t \dot{\theta}_t \tag{132}$$

hence the result after substituting for $\dot{\theta}_t$. □

Proof of Theorem 17.
First, by the assumptions of Theorem 17, the trajectories s_t satisfy the evolution equation $\dot{s}_t = f(s_t, u_t)$, so we can apply the results above with $f_t(s_t) := f(s_t, u_t)$, and the definition of F_t is consistent with the notation in the Kalman–Bucy filter.

Define

$$P_t := \eta_t J_{t\downarrow t}^{-1} \tag{133}$$

so that by definition, the evolution of the state (122) rewrites as

$$\frac{\mathrm{d}}{\mathrm{d}t} s_t(\theta_t) = f_t(s_t(\theta_t)) + P_t H_t^\top R_t^{-1}(y_t - h(s_t(\theta_t), u_t)) \tag{134}$$

which is the state evolution equation in the Kalman–Bucy filter. Thus we are left with checking the evolution equation for P_t.

We can compute the time derivative of P_t via (120). Using $\frac{\partial}{\partial t} J_{t\downarrow t}^{-1} = -J_{t\downarrow t}^{-1}(\frac{\partial}{\partial t} J_{t\downarrow t}) J_{t\downarrow t}^{-1}$, a direct computation yields

$$\dot{P}_t = \dot{\eta}_t J_{t\downarrow t}^{-1} - \eta_t J_{t\downarrow t}^{-1}\left(-F_t^\top J_{t\downarrow t} - J_{t\downarrow t} F_t - \gamma_t J_{t\downarrow t} + \gamma_t H_t^\top R_t^{-1} H_t\right) J_{t\downarrow t}^{-1} \tag{135}$$

$$= \frac{\dot{\eta}_t}{\eta_t} P_t + P_t F_t^\top + F_t P_t + \gamma_t P_t - \frac{\gamma_t}{\eta_t} P_t H_t^\top R_t^{-1} H_t P_t \tag{136}$$

If $\gamma_t = \eta_t$, this coincides with the Kalman–Bucy evolution equation for P_t with process noise

$$Q_t = \left(\eta_t + \frac{\dot{\eta}_t}{\eta_t}\right) P_t \tag{137}$$

which ends the proof. □

A Appendix: Reminder on Exponential Families

An *exponential family of probability distributions* on a variable x (discrete or continuous), with *sufficient statistics* $T_1(x), \dots, T_K(x)$, is the following family of distributions, parameterized by $\beta \in \mathbb{R}^K$:

$$p_\beta(x) = \frac{1}{Z(\beta)} \mathrm{e}^{\sum_k \beta_k T_k(x)} \lambda(\mathrm{d}x) \tag{138}$$

where $Z(\beta)$ is a normalizing constant, and $\lambda(\mathrm{d}x)$ is any reference measure on x, such as the Lebesgue measure or any discrete measure. The family is obtained by varying the parameter $\beta \in \mathbb{R}^K$, called the *natural* or *canonical* parameter. We will assume that the T_k are linearly independent as functions of x (and linearly independent from the constant function); this ensures that different values of β yield distinct distributions.

For instance, Bernoulli distributions are obtained with λ the uniform measure on $x \in \{0, 1\}$ and with a single sufficient statistic $T(0) = 0$, $T(1) = 1$. Gaussian distributions with a fixed variance are obtained with $\lambda(\mathrm{d}x)$ the Gaussian distribution centered on 0, and $T(x) = x$.

Another, often convenient parameterization of the same family is the following: each value of β gives rise to an average value $\bar{T}$ of the sufficient statistics,

$$\bar{T}_k := \mathbb{E}_{x \sim p_\beta} T_k(x) \tag{139}$$

For instance, for Gaussian distributions with fixed variance, this is the mean, and for a Bernoulli variable this is the probability to sample 1.

Exponential families satisfy the identities

$$\frac{\partial \ln p_\beta(x)}{\partial \beta_k} = T_k(x) - \bar{T}_k, \qquad \frac{\partial \ln Z}{\partial \beta_k} = \bar{T}_k \tag{140}$$

by a simple computation [AN00, (2.33)].

These identities are useful to compute the Fisher matrix J_β with respect to the variable β, as follows [AN00, (3.59)]:

$$\begin{aligned}
(J_\beta)_{ij} &:= \mathbb{E}_{x \sim p_\beta} \left[\frac{\partial \ln p_\beta(x)}{\partial \beta_i} \frac{\partial \ln p_\beta(x)}{\partial \beta_j} \right] && (141)\\
&= \mathbb{E}_{x \sim p_\beta} \left[(T_i(x) - \bar{T}_i)(T_j(x) - \bar{T}_j) \right] && (142)\\
&= \mathrm{Cov}(T_i, T_j) && (143)
\end{aligned}$$

or more synthetically

$$J_\beta = \mathrm{Cov}(T) \tag{144}$$

where the covariance is under the law p_β. That is, for exponential families the Fisher matrix is the covariance matrix of the sufficient statistics. In particular it can be estimated empirically, and is sometimes known algebraically.

In this work we need the Fisher matrix with respect to the mean parameter $\bar{T}$,

$$(J_{\bar{T}})_{ij} = \mathbb{E}_{x \sim p_\beta} \left[\frac{\partial \ln p_\beta(x)}{\partial \bar{T}_i} \frac{\partial \ln p_\beta(x)}{\partial \bar{T}_j} \right] \tag{145}$$

By substituting $\frac{\partial \ln p(x)}{\partial \alpha} = \frac{\partial \ln p(x)}{\partial \beta}\frac{\partial \beta}{\partial \alpha}$, the Fisher matrices J_α and J_β with respect to parameterizations α and β are related to each other via

$$J_\alpha = \frac{\partial \beta}{\partial \alpha}^\top J_\beta \frac{\partial \beta}{\partial \alpha} \tag{146}$$

(consistently with the interpretation of the Fisher matrix as a Riemannian metric and the behavior of metrics under change of coordinates [GHL87, §2.3]). So we need to compute $\partial \bar{T}/\partial \beta$. Using the log-trick

$$\partial \mathbb{E}_{x\sim p} f(x) = \mathbb{E}_{x\sim p}\left[f(x)\,\partial \ln p(x)\right] \tag{147}$$

together with (140), we find

$$\frac{\partial \bar{T}_i}{\partial \beta_j} = \frac{\partial \mathbb{E} T_i(x)}{\partial \beta_j} = \mathbb{E}\left[T_i(x)(T_j(x) - \bar{T}_j)\right] = \mathbb{E}\left[(T_i(x) - \bar{T}_i)(T_j(x) - \bar{T}_j)\right] = (J_\beta)_{ij} \tag{148}$$

so that

$$\frac{\partial \bar{T}}{\partial \beta} = J_\beta \tag{149}$$

(see [AN00, (3.32)], where η denotes the mean parameter) and consequently

$$\frac{\partial \beta}{\partial \bar{T}} = J_\beta^{-1} \tag{150}$$

so that we find the Fisher matrix with respect to $\bar{T}$ to be

$$J_{\bar{T}} = \frac{\partial \beta}{\partial \bar{T}}^\top J_\beta \frac{\partial \beta}{\partial \bar{T}} \tag{151}$$

$$= J_\beta^{-1} J_\beta J_\beta^{-1} \tag{152}$$

$$= J_\beta^{-1} = \mathrm{Cov}(T)^{-1} \tag{153}$$

that is, the Fisher matrix with respect to $\bar{T}$ is the inverse covariance matrix of the sufficient statistics.

This gives rise to a simple formula for the natural gradient of expectations with respect to the mean parameters. Denoting $\tilde{\nabla}$ the natural gradient,

$$\tilde{\nabla}_{\bar{T}}\, \mathbb{E} f(x) := J_{\bar{T}}^{-1} \frac{\partial \mathbb{E} f(x)}{\partial \bar{T}}^\top \tag{154}$$

$$= J_{\bar{T}}^{-1} \frac{\partial \beta}{\partial \bar{T}}^\top \frac{\partial \mathbb{E} f(x)}{\partial \beta}^\top \tag{155}$$

$$= J_\beta J_\beta^{-1} \frac{\partial \mathbb{E} f(x)}{\partial \beta}^\top \tag{156}$$

$$= \frac{\partial \mathbb{E} f(x)}{\partial \beta}^\top \tag{157}$$

$$= \mathbb{E}\left[f(x)\,\frac{\partial \ln p_\beta(x)}{\partial \beta}\right] \tag{158}$$

$$= \mathbb{E}\left[f(x)(T(x) - \bar{T})\right] \tag{159}$$

$$= \mathrm{Cov}(f, T) \tag{160}$$

which in particular, can be estimated empirically.

B Tensors and Charts for Manifolds

We state without proof some classical results from differential geometry.

Lemma 26. *Let Θ be a smooth manifold, and let $\mathcal{L}\colon \Theta \to \mathbb{R}$ be a real function on Θ. Let $\vartheta \in \Theta$ and let v be the derivative of $\mathcal{L}$ at ϑ, namely the cotangent vector*

$$v = \frac{\partial \mathcal{L}(\vartheta)}{\partial \vartheta} \tag{161}$$

Let $\Phi\colon \Theta \to \mathbb{R}^{\dim(\Theta)}$ be a chart on Θ. Then the expression of v in the chart Φ is

$$\mathbf{T}\Phi(v) = \left.\frac{\partial \ell(\theta)}{\partial \theta}\right|_{\theta=\Phi(\vartheta)} \tag{162}$$

where

$$\ell(\theta) := \mathcal{L}(\Phi^{-1}(\theta)) \tag{163}$$

is the expression of $\mathcal{L}$ in the chart.

Similarly, the expression of $v^{\otimes 2}$ in the chart is $\frac{\partial \ell(\theta)}{\partial \theta}^{\otimes 2}$.

Remember that a $(0,2)$-tensor can be seen as a map sending a tangent vector to a cotangent vector; therefore, if invertible, its inverse sends cotangent vectors to tangent vectors.

Lemma 27. *Let $\mathcal{J}$ be an invertible $(0,2)$-tensor on a manifold Θ, and let $\mathfrak{v}$ be a cotangent vector at some $\vartheta \in \Theta$. Let J and v be respectively the matrix and row vector representing $\mathcal{J}$ and $\mathfrak{v}$ in a chart. Then the expression of $\mathcal{J}^{-1}\mathfrak{v}$ in the chart is $J^{-1}v^\top$.*

Lemma 28. *Let Θ be a smooth manifold, and let $\mathcal{J}$ be a $(0,2)$-tensor at some $\vartheta \in \Theta$. Let Φ_1, Φ_2 be two charts on Θ and let $\psi := \Phi_2 \circ \Phi_1^{-1}$ be the change of chart.*

Let J_1 be the matrix representing $\mathcal{J}$ in chart Φ_1, and likewise for J_2. Then

$$J_2 = (\Psi^{-1})^\top J_1 \, \Psi^{-1} \tag{164}$$

where

$$\Psi := \left.\frac{\partial \psi(\theta)}{\partial \theta}\right|_{\theta=\Phi_1(\vartheta)} \tag{165}$$

Definition 29 (Covariant derivative associated with a chart). *Let Θ be a smooth manifold and let $\Phi\colon \Theta \to \mathbb{R}^{\dim(\Theta)}$ be a chart. The* covariant derivative associated with Φ *is the covariant derivative D which coincides with the usual derivative when expressed in chart Φ. Namely, for any curve (θ_t) in Θ and any tensor Z_t at θ_t,*

$$\frac{DZ_t}{\mathrm{d}t} := \mathbf{T}_{\theta_t}\Phi^{-1}\left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{T}\Phi(Z_t)\right) \tag{166}$$

This is indeed a covariant derivative, whose Christoffel symbols in chart Φ are 0.

References

[Ama98] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Comput.*, 10:251–276, February 1998.

[AN00] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, 2000. Translated from the 1993 Japanese original by Daishi Harada.

[APF00] Shun-ichi Amari, Hyeyoung Park, and Kenji Fukumizu. Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, 12(6):1399–1409, 2000.

[BB18] Axel Barrau and Silvère Bonnabel. Invariant kalman filtering. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:237–257, 2018.

[Ben15] Jérémy Bensadon. Black-box optimization using geodesics in statistical manifolds. *Entropy*, 17(1):304–345, 2015.

[Ber96] Dimitri P. Bertsekas. Incremental least squares methods and the extended Kalman filter. *SIAM Journal on Optimization*, 6(3):807–822, 1996.

[BL03] Léon Bottou and Yann LeCun. Large scale online learning. In *NIPS*, volume 30, page 77, 2003.

[Bon13] Silvère Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Trans. Automat. Contr.*, 58(9):2217–2229, 2013.

[CM44] Robert H Cameron and William T Martin. Transformations of Wiener integrals under translations. *Annals of Mathematics*, 45(2):386–396, 1944.

[dFNG00] João FG de Freitas, Mahesan Niranjan, and Andrew H. Gee. Hierarchical Bayesian models for regularization in sequential learning. *Neural computation*, 12(4):933–953, 2000.

[GA15] Mohinder S. Grewal and Angus P. Andrews. *Kalman filtering: Theory and practice using MATLAB*. Wiley, 2015. 4th edition.

[GHL87] S. Gallot, D. Hulin, and J. Lafontaine. *Riemannian geometry*. Universitext. Springer-Verlag, Berlin, 1987.

[GS15] Roger B. Grosse and Ruslan Salakhutdinov. Scaling up natural gradient by sparsely factorizing the inverse Fisher matrix. In *ICML*, pages 2304–2313, 2015.

[Hay01] Simon Haykin. *Kalman filtering and neural networks*. John Wiley & Sons, 2001.

[HRW12] Jeffrey Humpherys, Preston Redd, and Jeremy West. A fresh look at the kalman filter. *SIAM review*, 54(4):801–823, 2012.

[Jaz70] Andrew H. Jazwinski. *Stochastic processes and filtering theory*. Academic Press, 1970.

[Kul97] Solomon Kullback. *Information theory and statistics*. Dover Publications Inc., Mineola, NY, 1997. Reprint of the second (1968) edition.

[Kuo75] Hui-Hsiung Kuo. *Gaussian measures in Banach spaces*. Springer, 1975.

[LCL+17] Yubo Li, Yongqiang Cheng, Xiang Li, Xiaoqiang Hua, and Yuliang Qin. Information geometric approach to recursive update in nonlinear filtering. *Entropy*, 19(2):54, 2017.

[LMB07] Nicolas Le Roux, Pierre-Antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 849–856, 2007.

[Mar14] James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

[MCO16] Gaétan Marceau-Caron and Yann Ollivier. Practical Riemannian neural networks. arXiv preprint arXiv:1602.08007, 2016.

[MG15] James Martens and Roger B. Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *ICML*, pages 2408–2417, 2015.

[Nel00] Alex Tremain Nelson. Nonlinear estimation and modeling of noisy time-series by dual kalman filtering methods. 2000. PhD dissertation.

[OAAH17] Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 18(18):1–65, 2017.

[Oll15] Yann Ollivier. Riemannian metrics for neural networks I: feedforward networks. *Information and Inference*, 4(2):108–153, 2015.

[Oll18] Yann Ollivier. Online natural gradient as a kalman filter. *Electron. J. Statist.*, 12(2):2930–2961, 2018.

[Pat16] Vivak Patel. Kalman-based stochastic gradient method with stop condition and insensitivity to conditioning. *SIAM Journal on Optimization*, 26(4):2620–2648, 2016.

[PB13] Razvan Pascanu and Yoshua Bengio. Natural gradient revisited. *CoRR*, abs/1301.3584, 2013.

[RG11] Michael Roth and Fredrik Gustafsson. An efficient implementation of the second order extended kalman filter. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–6. IEEE, 2011.

[RRK+92] Dennis W. Ruck, Steven K. Rogers, Matthew Kabrisky, Peter S. Maybeck, and Mark E. Oxley. Comparative analysis of backpropagation and the extended Kalman filter for training multilayer perceptrons. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):686–691, 1992.

[Sim06] Dan Simon. *Optimal state estimation: Kalman, H_∞, and nonlinear approaches*. John Wiley & Sons, 2006.

[ŠKT01] Miroslav Šimandl, Jakub Královec, and Petr Tichavskỳ. Filtering, predictive, and smoothing Cramér–Rao bounds for discrete-time nonlinear dynamic systems. *Automatica*, 37(11):1703–1716, 2001.

[SW88] Sharad Singhal and Lance Wu. Training multilayer perceptrons with the extended Kalman algorithm. In *NIPS*, pages 133–140, 1988.

[Sä13] Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[vdV00] A.W. van der Vaart. *Asymptotic statistics*. Cambridge university press, 2000.

[Wik] Wikipedia. Extended Kalman filter. https://en.wikipedia.org/wiki/Extended_Kalman_filter#Continuous-time_extended_Kalman_filter, retrieved on 2018-12-03.