A State-Space Solution to the Estimation of Interacting Vehicle Trajectories with Deep Neural Networks and Variational Bayes Filtering

Tristan Klempka, Patrick Danès

A State Space Solution to the Estimation of Interacting Vehicle Trajectories with Deep Neural Networks and Variational Bayes Filtering

Tristan Klempka
LAAS-CNRS, Université de Toulouse, CNRS, UPS
Continental Digital Services France
Toulouse, France
tristan.klempka@laas.fr

Patrick Danès
LAAS-CNRS, Université de Toulouse, CNRS, UPS
Toulouse, France
patrick.danes@laas.fr

***Abstract*—This paper addresses the estimation of trajectories of interacting vehicles at a microscopic scale, as a prerequisite to their prediction for risk assessment. A state-space solution is investigated, where both the Markov hidden state (continuous-valued, which captures the joint histories of vehicles) and the measurements (low-dimensional and noisy) admit a vehicle-wise structure. The vehicles' transition models are assumed independent of each other, time- and vehicle-invariant, and coequal to an "egocentric" prior dynamics pdf. To cope with the vehicles' interactions, this pdf is conditioned on the full state vector as the past time index, which imposes a centralized estimation/prediction of the fleet motion. The two fundamental pillars of the approach are developed: learning of a Gaussian mixture egocentric transition model by means of Deep Neural Networks; synthesis of a stochastic variational Bayes filtering algorithm which features a decentralized vehicle-wise structure but takes into account interactions. Tests on highway scenarios are presented.**

***Index Terms*—Autonomous vehicles, Intelligent transportation systems, Robotics, Interactive vehicle dynamics, Neural-network models, Stochastic filtering techniques.**

I. Introduction

Detecting and predicting dangerous situations is one of the major challenges of road safety. It allows preventing measures to be taken in order to avoid potentially deadly outcomes. To do so, Advanced Driver Assistance Systems (ADAS) must be able to model the current traffic situation and its evolution. The complex and evolving dependencies between drivers constitute an important issue for such systems.

Vehicle motion models are reviewed in [1]. They can be classified as physics-based, maneuver-based or interaction-aware. The first two models cannot constitute the basis for midterm motion prediction because of their lack of representation power. Only interaction-aware models can capture ego-motions together with road configurations and others drivers. So, they alone are suited to risk assessment based on drivers' joint motion estimation and prediction [2].

This work has been partially funded by the French National Research and Technology Agency, in the framework of a CIFRE project involving LAAS-CNRS and Continental Digital Services France.

Interaction-aware models are often made of two distinct highly coupled parts : intent estimation and motion prediction. According to [3], one way of handling their complexity is to consider multiple discrete maneuver hypotheses and estimate their posterior probabilities. The posterior maneuver estimate can in turn constitute a basis for continuous motion prediction. Intention estimators usually rely on the discriminative classification of specific maneuvers with respect to the vehicles' status and environmental cues, *e.g.*, by using support vector machines (SVMs) [4], Long-Short Term Memory (LSTM) neural networks [5], heuristic rules via expert knowledge [6], or generative Hidden Markov Models [7]. Motion prediction can be addressed by means of Gaussian processes [7], Mixture Density Networks [8], LSTM Networks [5], Conditional Variational Auto-Encoders [9], or optimization-based planning [3].

Intention and motion estimation can be jointly performed. In [10], a driver model is learned by Inverse Reinforcement Learning so as to predict the actions of all vehicles in a highway traffic scene considering various intentions/maneuvers. In [11], a LSTM Network is trained with relational features between drivers so as to predict their future motions, implicitly capturing their intentions if the dataset is large enough. Stochastic filtering techniques have also been used to estimate intention and predict motion. The road scene is often modeled as a Markov process by means of a Dynamic Bayesian Network, which captures the dependencies between agents [12]–[15]. This setting often implies mixed (discrete-continuous) state-spaces and nonlinear models, so that conventional techniques do not apply. Moreover, no instantaneous measure of drivers' intentions is available.

This paper proposes a state-space approach to motion estimation where both the Markov hidden state (continuous-valued, which captures the joint histories of vehicles) and the measurements (low-dimensional and noisy) admit a vehicle-wise structure. Section II states the problem, and introduces the "egocentric" prior dynamics pdf common to all vehicles at all times. Then, Section III introduces the two cornerstones of the approach, viz. the learning of a Gaussian mixture egocentric transition model by means of Deep Neural Networks,

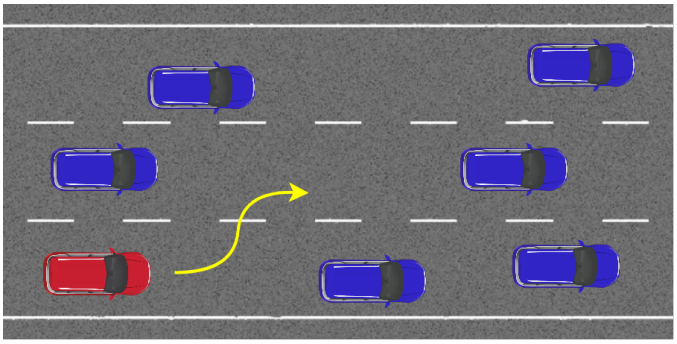

Fig. 1. Typical configuration of an highway traffic scenario that leads to strong interactions between drivers. The red car motion is influenced by blue cars around it.

and the stochastic variational Bayes filtering algorithm which enables the decentralized vehicle-wise motion estimation while taking into account interactions. Tests on highway scenarios constitute Section IV. Open issues conclude the paper.

II. Problem Statement and Context of Its Solution

A. Trajectories estimation as a prerequisite for prediction

A multiple-lane highway section is considered, which comprises no intersection nor access ramp (Figure 1). At any time instant nT_s, with $n = 0, \dots, N$ and T_s the sampling period, a set $\mathcal{V}$ of K vehicles $\mathcal{V}_1, \dots, \mathcal{V}_K$ (cars or trucks) moves rightwards. The considered situations entail interactions among $\mathcal{V}$ and can be coarsely depicted in terms of the maneuvers being executed, *e.g.*, *free_ride*, *following*, *braking*, *overtaking*, etc.

As a prerequisite to subsequent risk assessment, the prediction of the trajectories of all vehicles must be performed at a microscopic scale, *i.e.*, over horizons ranging between 1 and 5 seconds (as commonly found in the literature).

B. Towards a State-Space Solution

1) Overview: A state-space approach is sought for, in which a hidden random state vector captures the status of the whole fleet $\mathcal{V}$. This vector, termed x_n at time n, comes as the stacking of subvectors $\{x_n^k\}_{k=1,\dots,K}$. For any pair k, k' in $\{1, \dots, K\}$, x_n^k and $x_n^{k'}$ report the same kind of information on vehicles $\mathcal{V}_k$ and $\mathcal{V}_{k'}$. Importantly, the hidden state random process $x_{0:n} = x_0, \dots, x_n$ is assumed Markov. In the same way as $x_n = \left((x_n^1)^T, \dots, (x_n^K)^T\right)^T$, with $.^T$ the transpose operator, the measurement random variable at any time n features a vehicle-wise structure and writes as $z_n = \left((z_n^1)^T, \dots, (z_n^K)^T\right)^T$. Given a sequence $z_{0:n}$ of noisy low-dimensional measurements[1], the aim is to determine the state vector filtering probability density function (pdf) $p(x_n|z_{0:n})$, with first moments $\hat{x}_{n|n}, P_{n|n}$. From this distribution, m-step-ahead prediction pdfs $p(x_{n+m}|z_{0:n})$ can be obtained for the whole fleet $\mathcal{V}$, and risk assessment can be conducted, *e.g.*, on the basis of their approximate confidence sets.

The prior knowledge on the hidden state process is composed of the initial pdf $p(x_0)$ and the prior dynamics pdf $p(x_n|x_{n-1})$, $n \geq 1$. It is henceforth assumed that, conditioned on the previous state of the whole fleet, all the vehicles dynamics are mutually independent, *i.e.*,

$$\forall n \geq 1,\ p(x_n|x_{n-1}) = \prod_{k=1}^{K} p(x_n^k|x_{n-1}). \tag{1}$$

In addition, it can reasonably be postulated that the admissible transitions of vehicle $\mathcal{V}_k$ between times $n-1$ and n only depend on its state x_{n-1}^k and on the states of the set $\mathcal{N}_{n-1}(\mathcal{V}_k)$ of its neighbors at time $n-1$, *i.e.*,

$$p(x_n|x_{n-1}) = \prod_{k=1}^{K} p(x_n^k|x_{n-1}^k, \{x_{n-1}^l\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)}). \tag{2}$$

Last, the vehicle-wise (or "egocentric") transition model $p(x_n^k|x_{n-1}^k, \{x_{n-1}^l\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)})$ can be deemed time-invariant (independent of n) and common to all vehicles (independent of the selected ego-vehicle $\mathcal{V}_k$). Note that this set of assumptions does not prevent the prior dynamics pdf from fully capturing all the interactions among vehicles.

As for the measurement model $p(z_n|x_n)$, besides vehicle-wise structure, vehicle-wise independence is assumed, so that

$$\forall n \geq 0,\ p(z_n|x_n) = \prod_{k=1}^{K} p(z_n^k|x_n^k). \tag{3}$$

2) Potential benefits and Challenges: This fairly classical state-space statement fundamentally differs from recent powerful approaches which aim to predict output time sequences on the basis of, say, Recurrent or Long Short Term Memory Neural Networks, in two respects: the prediction from the sequence of past measurements entails the explicit definition and selection of a Markov state vector which, at any time, captures the history of $\mathcal{V}$; at any time, the measurements are noisy low-dimensional expressions of the hidden state. Each state subvector x_n^k associated to $\mathcal{V}_k$ may be made of discrete-valued indexes expressing driver intentions and/or maneuvers, together with continuous-valued variables depicting the fine-grained nature of the motion.

Nevertheless, two major challenges are raised by the context of interactive vehicles. On the one hand, hand-crafting of the prior dynamics model would be difficult and limited by its ability to capture the huge variability of situations. For instance, not only the combinatorics of admissible intentions and maneuvers is very high, but their thorough description through a model like (2) would be cumbersome. On the other hand, conventional recursive Bayesian filtering or prediction techniques may be unsuitable to this problem, as the very high-dimensional state-space necessary to depict $\mathcal{V}$ may induce an inability to compute/approximate the posterior pdfs, a prohibitive computational complexity, or even unacceptable systematic computation errors. These challenges have given rise to the cornerstones of the approach described next.

[1]Throughout the paper, the same notation is used for random variables/processes and for their outcomes in a random experiment.

III. Fundamentals of The Approach

A. Definition of the Hidden State Process

The state vector x_n includes no discrete-valued variable. The (continuous-valued) entries of each subvector x_n^k are the x- (longitudinal) position and velocities of vehicle $\mathcal{V}_k$ followed by its y- (transversal) positions and velocities. It is expected that this definition of $x_n^k \in \mathbb{R}^4$ is rich enough so that the egocentric transition model $p(x_n^k|x_{n-1}^k, \{x_{n-1}^l\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)})$ involved in (2) can grasp a broad range of admissible motions. This is an important issue, especially because such an egocentric model must implicitly encode the drivers' intentions/maneuvers[2]. Strikingly, if intentions/maneuvers were explicitly captured by discrete state variables, these would not follow a homogeneous Markov chain. Instead, their prior dynamics would take the form of a transition matrix which depends on the continuous-valued (past) state variables in a hardly identifiable way. In summary, defining the state vector as continuous-valued simplifies the problem in terms of required prior knowledge and estimation complexity.

Closed-form transition models do exist, *e.g.*, the intelligent driver model [16] or social forces for pedestrian motion. However, they convey limited representation possibilities, as they can only capture simple interactions involving few agents. As Gaussian Mixture Models (GMMs) can approximate a huge set of probability density functions [17], the egocentric prior dynamics pdf has instead been set to a conditional GMM of the form (":=" means "is defined as")

$$p(x_n^k|x_{n-1}^k, \{x_{n-1}^l\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)}) := \sum_{m=1}^{M} \lambda_m(x_{n-1}^k, \diamond)\mathcal{G}(x_n^k; \mu_m(x_{n-1}^k, \diamond), \Sigma_m(x_{n-1}^k, \diamond)), \quad \text{where } \diamond = \{x_{n-1}^l\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)} \tag{4}$$

and $\mathcal{G}(x; \mu, \Sigma)$ terms the multivariate Gaussian pdf of x with mean μ and covariance matrix Σ.

At any time n, the observations are assumed to be independent noisy measurements of the genuine x- and y- positions of the vehicles, so that (3) is defined as

$$p(z_n|x_n) := \prod_{k=1}^{K} \mathcal{G}(z_n^k; H_n^k x_n^k, R_n^k), \;\; H_n^k = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \tag{5}$$

with $R_n^k \in \mathbb{R}^{2\times 2}$ the known (maybe time- and vehicle-dependent) covariance matrix of the additive zero-mean measurement noise.

B. Learning Prior Dynamics with Deep Neural Networks

The weight $\lambda_m \in [0; 1]$, mean vector $\mu_m \in \mathbb{R}^4$ and covariance matrix $\Sigma_m \in \mathbb{R}^{4\times 4}$ associated to each m^{th} hypothesis of the egocentric transition GMM (4) are functions of the state vector entries associated at the past time index to the ego-vehicle given in argument and to its neighbors. Though with some loss of generality and at the expense of handling needlessly conservative (spreaded) prior dynamics, the covariance matrices Σ_m are henceforth assumed diagonal, *i.e.*, with $\diamond$ defined above and for all $m \in \{1, \dots, M\}$,

$$\Sigma_m(\diamond) := \text{diag}(\sigma_{m,1}^2(x_{n-1}^k, \diamond), \dots, \sigma_{m,4}^2(x_{n-1}^k, \diamond)). \tag{6}$$

Assuming that a dataset of "clean" sequences of state vectors is available, the consequent model (4),(6), *i.e.*, the dependency of λ_m, μ_m, $\sigma_{m,1}, \dots, \sigma_{m,4}$ on their arguments, is learned by machine learning techniques. Mixture Density Networks (MDNs) were proposed in [18] as a direct way of learning the GMM parameters in (4). However, in our problem, the high dimensions of the inputs and outputs of the MDN lead to mode collapsing, a phenomenon acknowledged in [19]: the native MDN implementation reduces the sought multimodal prior dynamics to a single Gaussian distribution.

To overcome this limitation, GMM parameters are learned separately by distinct feed-forward Deep Neural Networks (DNNs). First, a DNN learns the parameter functions $\{\lambda_m(x_{n-1}^k, \diamond)\}_{m=1,\dots,M}$, which map the states at time $n-1$ of the ego-vehicle and its neighbors to the probability of each m^{th} intended maneuver by the ego-vehicle at time n. This DNN entails multiple fully connected layers: an input layer of the size of the states of the ego-vehicle and its neighbors; several fully connected hidden layers; an output layer which size is the number M of considered maneuvers. To ensure that $\sum_{m=1}^{M} \lambda_m(x_{n-1}^k, \diamond) = 1$, the activation function of last layer is set to the Log-Softmax function

$$\log \lambda_m = y_m^\lambda - \log\Big(\sum_{l=1}^{M} \exp y_l^\lambda\Big), \tag{7}$$

with $\{y_m^\lambda \in \mathbb{R}\}_{m=1,\dots,M}$ the outputs from the last layer. The Log-Softmax function is preferred to the conventional Softmax activation function in view of its superior numerical stability. If the dataset is structured into KN tuples $\big\{\big(\mathsf{x}_\mathsf{n}^\mathsf{k}, \{\mathsf{x}_\mathsf{n}^\mathsf{l}\}_{\mathcal{V}_{\mathsf{l}\in\mathcal{N}_\mathsf{n}(\mathcal{V}_\mathsf{k})}}, \mathsf{m}_\mathsf{n}^\mathsf{k}\big)\big\}$ gathered on ego-vehicles $\mathsf{k} = 1, \dots, \mathsf{K}$ over times $\mathsf{n} = 1, \dots, \mathsf{N}$, where $\mathsf{m}_\mathsf{n}^\mathsf{k}$ terms the genuine pre-labeled maneuver, then the classification network is trained so as to minimize the loss function

$$\text{Loss} := -\sum_{\mathsf{n}=1,\mathsf{k}=1}^{\mathsf{n}=\mathsf{N},\mathsf{k}=\mathsf{K}} \log p(\mathsf{m}_\mathsf{n}^\mathsf{k}|\mathsf{x}_\mathsf{n}^\mathsf{k}, \{\mathsf{x}_\mathsf{n}^\mathsf{l}\}_{\mathcal{V}_{\mathsf{l}\in\mathcal{N}_\mathsf{n}(\mathcal{V}_\mathsf{k})}}). \tag{8}$$

Secondly, once the probability distribution of the maneuvers intended by the driver has been learned, M independent DNNs are trained. Each m^{th} DNN learns the functions $\mu_m(x_{n-1}^k, \diamond)$ and $\Sigma_m(x_{n-1}^k, \diamond)$, which map the states at time $n-1$ of the ego-vehicle undergoing the m^{th} maneuver and the states of its neighbors to the moments of the Gaussian distribution from which the state at time n of the ego-vehicle is sampled. Each m^{th} DNN also features multiple connected layers. Its last layer outputs are $y_m^\mu \in \mathbb{R}^2$ and $y_m^{\sigma_1}, \dots, y_m^{\sigma_4} \in \mathbb{R}$. The sought functions μ_m, $\sigma_{m,1}, \dots, \sigma_{m,4}$, which must fulfill $\sigma_{m,1} > 0, \dots, \sigma_{m,4} > 0$, are obtained by

$$\mu_m = y_m^\mu, \;\; \sigma_{m,i} = \exp(y_m^{\sigma_i}), \;\; i = 1, \dots, 4. \tag{9}$$

[2]Note that the implicit encoding of discrete maneuvers into a transition pdf on continuous random vectors does not preclude the explicit learning of their temporal dynamics, see Section III-B.

Each m^{th} network is trained with $\mathsf{K}_m\mathsf{N}_m$ tuples $\big\{\big(\mathsf{x}^{\mathsf{k}}_{\mathsf{n}-1,m},\{\mathsf{x}^{\mathsf{l}}_{\mathsf{n}-1,m}\}_{\mathcal{V}_{\mathsf{l}}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_{\mathsf{k}})},\mathsf{x}^{\mathsf{k}}_{\mathsf{n},m}\big)\big\}$ gathered on K_m vehicles engaged in the maneuver indexed by m over N_m times. The corresponding loss function writes as

$$\text{Loss}_m := -\sum_{\mathsf{n}=1,\mathsf{k}=1}^{\mathsf{n}=\mathsf{N}_m,\mathsf{k}=\mathsf{K}_m} \log \mathcal{G}\big(\mathsf{x}^{\mathsf{k}}_{\mathsf{n},m};\mu_m(x^{\mathsf{k}}_{\mathsf{n}-1,m},\mathsf{o}),\Sigma_m(x^{\mathsf{k}}_{\mathsf{n}-1,m},\mathsf{o})\big), \quad \text{with } \mathsf{o}=\{\mathsf{x}^{\mathsf{l}}_{\mathsf{n}-1}\}_{\mathcal{V}_{\mathsf{l}}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_{\mathsf{k}})}. \quad (10)$$

C. State Estimation by Variational Bayes Particle Filtering

1) Basics: Particle filtering can in principle get an approximation of the filtering pdf. However, the underlying random sampling of the high-dimensional state-space may induce the "gaps and clusters" phenomenon, and lead to a poor efficiency at a high computational cost. This is why variational Bayes approximations have been envisaged.

Let x be a random variable and $x^1,\dots,x^K$ its partition into K blocks. Let $\mathcal{Q}$ be the set of separable pdfs of x, *i.e.*, $\mathcal{Q}=\{\tilde{q}(x)=\prod_{k=1}^K \tilde{q}(x^k)\}$. A (separable) pdf $q(x)$ is termed the (separable) *variational Bayes (VB) approximation* of a given pdf $p(x)$ if it minimizes, over $\mathcal{Q}$, the Kullback-Leibler divergence to $p(x)$, *i.e.*,

$$q(x):=\prod_{k=1}^K q(x^k)=\arg\min_{\tilde{q}(x):=\prod_{k=1}^K \tilde{q}(x^k)} \text{KL}\big(\tilde{q}(x)\|p(x)\big). \quad (11)$$

The well-known solution to (11) writes as

$$q(x):=\prod_{k=1}^K q(x^k),\quad q(x^k)\propto\exp\Big(\mathbb{E}_{q(x^{k-})}\big[\ln p(x)\big]\Big), \quad (12)$$

where x^{k-} stands for the complement of x^k in x. Definition (11) implies that $q(x)$ cannot take high values while $p(x)$ is small, or, equivalently, that the modes of $q(x)$ capture at least some modes of $p(x)$. Its solution (12) shows that the imposed stochastic independence of the marginals of $q(x)$ is traded off with the mathematical dependence of each $q(x^k)$ on expectations involving $\{q(x^{k'})\}_{k'\neq k}$.

Variational methods have long been used for inference and learning [20], [21]. Their potential for Bayesian filtering [22], [23] has also been investigated, with the aim to recursively approximate the filtering pdf $p(x_n|z_{0:n})$ by $q(x_n|z_{0:n}) := \prod_{k=1}^K q(x^k_n|z_{0:n})$ which minimizes $\text{KL}\big(\tilde{q}(x_n|z_{0:n})\|p(x_n|z_{0:n})\big)$ over separable pdfs $\tilde{q}(x_n|z_{0:n})$. [24] derive a computationally efficient "multiple" filtering strategy, which, thanks to VB approximations, enables a solution by running K separate (though interacting) particle filters on respective partitions of the hidden state vector. Figure 2 instantiates their algorithm for a simplified problem. One key point lies in the VB approximation $q(x^k_n|x^k_{n-1},z_{0:n-1})$ of the genuine partition-wise prior dynamics $p(x^k_n|x_{n-1})$, which eliminates its conditioning on x^{k-}_{n-1} (*i.e.*, the complement of x^k_{n-1} in x_{n-1}):

$$q(x^k_n|x^k_{n-1},z_{0:n-1})\propto\exp\Big(\mathbb{E}_{q(x^{k-}_{n-1}|z_{0:n-1})}[\ln p(x^k_n|x_{n-1})]\Big), \quad (13)$$

Figure 2 also reports the way how to approximately sample from such a pdf.

2) Application to interacting vehicles: For the considered filtering problem, the state vector is partitioned vehicle-wise, so that K can stand both for the number of partitions and the number of vehicles. However, the assumptions (2),(4) stated in the considered context of interacting vehicles do not match with these of [24]. Nevertheless, by simplifying the problem, a solution can be set up, through at the expense of an extension of [24]. The simplification consists in approximating the GMM transition model (4) by its moment-matched single Gaussian approximation

$$\begin{aligned} p(x^k_n|x^k_{n-1},\diamond)&=\textstyle\sum_{m=1}^M \lambda_m(x^k_{n-1},\diamond)\mathcal{G}(x^k_n;\mu_m(x^k_{n-1},\diamond),\Sigma_m(x^k_{n-1},\diamond)),\\ &\approx \bar{p}(x^k_n|x^k_{n-1},\diamond):=\mathcal{G}(x^k_n;\mu(x^k_{n-1},\diamond),\Sigma(x^k_{n-1},\diamond)), \quad (14)\\ \text{with } \mu(.)&=\textstyle\sum_{m=1}^M \lambda_m(.)\mu_m(.)\\ \Sigma(.)&=\textstyle\sum_{m=1}^M \lambda_m(.)\big(\Sigma_m(.)+(\mu_m(.)-\mu(.))(\mu_m(.)-\mu(.))^T\big). \end{aligned}$$

As it is, Part I of Figure 2 cannot handle Gaussian prior dynamics such as (14). The whole algorithm from [24] must then be extended to the case when the matrix $Q_n := \text{blkdiag}(Q^1_n,\dots,Q^K_n)$ is a function of x_n, *i.e.*, $Q_n(x_n) := \text{blkdiag}(Q^1_n(x_n),\dots,Q^K_n(x_n))$. Going back to (13), and doing the calculations leads to replacing the first item of Part III by

$$q(x^k_n|x^k_{n-1},z_{0:n-1})=\mathcal{G}(x^k_n;m^k_n(x^k_{n-1}),\Lambda^k_n(x^k_{n-1})), \quad (15)$$

$$\text{with } \Lambda^k_n(x^k_{n-1}):=\Big(\mathbb{E}_{q(x^{k-}_{n-1}|z_{0:n-1})}\big[\big(Q^k_{n-1}(x_{n-1})\big)^{-1}\big]\Big)^{-1} \text{ and}$$

$$m^k_n(x^k_{n-1}):=\Lambda^k_n(x^k_{n-1})\mathbb{E}_{q(x^{k-}_{n-1}|z_{0:n-1})}\Big[(Q^k_{n-1}(x_{n-1}))^{-1}f^k_{n-1}(x_{n-1})\Big].$$

Monte Carlo approximations of $m^k_n(x^k_{n-1}),\Lambda^k_n(x^k_{n-1})$ can still be obtained in the vein of the second item of Part III. Importantly, (15) again shows that though the variational Bayes multiple particle filter can lead to a separable approximation to the posterior pdf, this is not at the expense of neglecting the interactions among vehicles.

IV. Case Study

A. Implementation

Using 4K footage recorded from an aerial drone and state-of-the-art deep learning video segmentation techniques, the HighD dataset provides positions and velocities of interacting vehicles on a highway segment about 420 meters long, during a visibility median duration of 13.6 s per vehicle. Data points are recorded at 25 fps. A downsampling factor of 13 is applied, so that the final time step is $T_s = 0.52$s.

HighD is deemed accurate enough to be used as a surrogate for the genuine dataset $\mathcal{X}$ of hidden states. As stated in Section III-A, vehicle longitudinal+lateral positions+velocities are collected. $\mathcal{X}$ is then manually segmented and labeled with the maneuver intended by each vehicle, among: *forward* (the vehicle stays at least 3 seconds into the same lane); *sheer_in* and *sheer_out* (the vehicle moves from its current lane to its nearest right or left lane, respectively, during a 3 seconds interval).

I - PROBLEM

- Assumptions
  - $x_{n+1} = \begin{pmatrix} x_{n+1}^1 \\ \vdots \\ x_{n+1}^K \end{pmatrix} = f_n(x_n) + w_n = \begin{pmatrix} f_n^1(x_n)+w_n^1 \\ \vdots \\ f_n^K(x_n)+w_n^K \end{pmatrix}$, $w_n \sim \mathcal{G}(0, Q_n := \text{blkdiag}(Q_n^1, \ldots, Q_n^K))$, $w_{0:n}$ white;
  - $z_n = \begin{pmatrix} z_n^1 \\ \vdots \\ z_n^K \end{pmatrix} = Hx_n + v_n = \begin{pmatrix} H_n^1 x_n^1 + v_n^1 \\ \vdots \\ H_n^K x_n^K + v_n^K \end{pmatrix}$, $v_n \sim \mathcal{G}(0, R_n := \text{blkdiag}(R_n^1, \ldots, R_n^K))$, $v_{0:n}$ white indep. of $w_{0:n}$ and x_0.
- Goal of each k^{th} separate particle filter
  - Given the discrete pdf $\hat{q}(x_{n-1}^k|z_{0:n-1}) = \sum_{s=1}^S \lambda_{n-1}^{k,(s)} \delta(x_{n-1}^k - \tilde{x}_{n-1}^{k,(s)})$ which approximates $q(x_{n-1}^k|z_{0:n-1})$ such that $q(x_{n-1}|z_{0:n-1}) := \prod_{k=1}^K q(x_{n-1}^k|z_{0:n-1}) \approx p(x_{n-1}|z_{0:n-1})$, determine the approximation $\hat{q}(x_n^k|z_{0:n}) = \sum_{s=1}^S \lambda_n^{k,(s)} \delta(x_n^k - \tilde{x}_n^{k,(s)})$ of $q(x_n^k|z_{0:n})$ which satisfies $q(x_n|z_{0:n}) := \prod_{k=1}^K q(x_n^k|z_{0:n}) \approx p(x_n|z_{0:n})$.

II - MAIN ALGORITHM

- [If $n = 0$, then: replace steps 1 and 2 below by sample $\{x_0^{k,(s)}\}_{s=1,\ldots,S}$ i.i.d. from $p(x_0)$; then do step 2b.]
- For each $k = 1, \ldots, K$,
  1) Sample $\{x_{n-1}^{k,(s)}\}_{s=1,\ldots,S}$ i.i.d. from $\hat{q}(x_{n-1}^k|z_{0:n-1})$ (*i.e.*, resample the weighted particle set $\{(\tilde{x}_{n-1}^{k,(s)}, \lambda_{n-1}^{k,(s)})\}_{s=1,\ldots,S}$ into $\{(x_{n-1}^{k,(s)}, \frac{1}{S})\}_{s=1,\ldots,S}$).
  2) For each $s \in \{1, \ldots, S\}$,
     a) draw $\tilde{x}_n^{k,(s)} \sim q(x_n^k|x_{n-1}^{k,(s)}, z_{0:n-1})$, where $q(x_n^k|x_{n-1}^k, z_{0:n-1})$ is the approximation of the genuine partition-wise prior dynamics $p(x_n^k|x_{n-1})$ (which is also equal to $p(x_n^k|x_{n-1}, z_{0:n-1})$) by variational Bayes arguments so as to drop the statistical dependence between x_n^k and x_{n-1}^{k-} conditionally on x_{n-1}^k and $z_{0:n-1}$;
     b) set $\lambda_n^{k,(s)} = p(z_n^k|x_n^{k,(s)})$.
  3) Renormalize the weights, *i.e.*, for each $s \in \{1, \ldots, S\}$, set $\lambda_n^{k,(s)} = \frac{\lambda_n^{k,(s)}}{\sum_{r=1}^S \lambda_n^{k,(r)}}$.
  4) From the discrete pdf $\hat{q}(x_n^k|z_{0:n}) = \sum_{s=1}^S \lambda_n^{k,(s)} \delta(x_n^k - \tilde{x}_n^{k,(s)})$ which approximates $q(x_n^k|z_{0:n})$, deduce the k^{th} partition-wise approximation $\hat{x}_{n|n}^k = \sum_{s=1}^S \lambda_n^{k,(s)} \tilde{x}_n^{k,(s)}$ and $\hat{P}_{n|n}^k = \sum_{s=1}^S \lambda_n^{k,(s)} (\tilde{x}_n^{k,(s)} - \hat{x}_{n|n}^k)(\tilde{x}_n^{k,(s)} - \hat{x}_{n|n}^k)^T$ of the genuine posterior mean $x_{n|n}$ and posterior covariance $P_{n|n}$ of $p(x_n|z_{0:n})$.

III - DETAIL OF STEP 2a: HOW TO SAMPLE FROM $q(x_n^k|x_{n-1}^{k,(s)}, z_{0:n-1})$

- Equation (13) leads to $q(x_n^k|x_{n-1}^k, z_{0:n-1}) = \mathcal{G}(x_n^k; m_n^k(x_{n-1}^k), Q_{n-1}^k)$ with $m_n^k(x_{n-1}^k) := \mathbb{E}_{q(x_{n-1}^{k-}|z_{0:n-1})}\left[f_{n-1}^k(x_{n-1})\right]$.
- $m_n^k(x_{n-1}^k)$ cannot be computed in closed-form, but a Monte Carlo approximation $\hat{m}_n^k(x_{n-1}^k) = \frac{1}{J}\sum_{j=1}^J f_{n-1}^k(\mathcal{X}_{k,n-1}^{(s_j)})$, *e.g.*, with $J = S$, can be easily obtained by setting $\mathcal{X}_{k,n-1}^{(s_j)} := \left((x_{n-1}^{1,(s_j)})^T, \ldots, (x_{n-1}^{k-1,(s_j)})^T, (x_{n-1}^k)^T, (x_{n-1}^{k+1,(s_j)})^T, \ldots, (x_{n-1}^{K,(s_j)})^T\right)^T$, where for each $r \neq k$: $\forall j = 1, \ldots, J$, $x_{n-1}^{r,(s_j)} \overset{\text{i.i.d.}}{\sim} \sum_{s=1}^S \frac{1}{S}\delta(x_{n-1}^r - x_{n-1}^{r,(s)})$. In other words, for each k^{th} partition, $\hat{m}_n^k(x_{n-1}^k)$ is computed as the average of $\{f_{n-1}^k(\mathcal{X}_{k,n-1}^{(s_j)})\}_{j=1,\ldots,J}$, where: whatever $j = 1, \ldots, J$, the k^{th} subvector of $\mathcal{X}_{k,n-1}^{(s_j)}$ is set to the variable x_{n-1}^k; each other r^{th} subvector of $\mathcal{X}_{k,n-1}^{(s_j)}$, $r \neq k$, is obtained by sampling within the set $\{x_{n-1}^{r,(1)}, \ldots, x_{n-1}^{r,(S)}\}$ obtained after step 1, this process being of course repeated for $j = 1, \ldots, J$.

Fig. 2. Variational Bayes multiple particle filter of [24] for a simplified problem.

This labeled dataset $\left\{\left(\mathsf{x}_\mathsf{n}^\mathsf{k}, \{\mathsf{x}_\mathsf{n}^\mathsf{l}\}_{\mathsf{l}\in\mathcal{N}_\mathsf{n}(\mathcal{V}_\mathsf{k})}, \mathsf{m}_\mathsf{n}^\mathsf{k}\right)\right\}$ constitutes the basis of DNN-based prior dynamics learning, as per Section III-B. Each vehicle is considered in turn as the ego-vehicle. The set $\mathcal{N}_\cdot(\mathcal{V}_\mathsf{k})$ associated to any k^{th} ego-vehicle is made up with eight neighbors: the preceding vehicle, the vehicle in front of the preceding vehicle, the following vehicle, the vehicle behind the following vehicle, the vehicles on the right and left lanes which are preceding or driving alongside the ego-vehicle. When a neighbor does not exist, its longitudinal distance is set to a large number (in front of behind) and its velocity and lateral position are set to these of the ego-vehicle. A total of $90,000$ data points is splitted into 80% for training and 20% for validation. Half of the training dataset is labeled as the *forward* maneuver, while the remaining half is evenly distributed between the *sheer_in* and the *sheer_out* maneuvers. DNNs Performance metrics are calculated on a test set of $20,000$ different frames. Submetric RMSEs were obtained for the three regression networks. The classification network obtained a balanced accuracy of 97%.

The maneuver classification network is composed of 3 hidden layers of 128 neurons each, with the Tanh activation function. A dropout layer is added after each hidden layer in order to improve generalization during training. Incidentally, the inputs to this DNN are $\left\{\left(\mathsf{x}_\mathsf{n}^\mathsf{k}, \{\mathsf{x}_\mathsf{n}^\mathsf{k} - \mathsf{x}_\mathsf{n}^\mathsf{l}\}_{\mathsf{l}\in\mathcal{N}_\mathsf{n}(\mathcal{V}_\mathsf{k})}, \mathsf{m}_\mathsf{n}^\mathsf{k}\right)\right\}$. The M subsequent (independent) trajectory DNNs feature a common architecture: they are composed of 2 hidden layers of 512 neurons each, with Tanh activation functions. All the DNNs are trained using the Adam optimizer with a learning rate of $\alpha = 0.00004$ for the maneuver classification network and $\alpha = 0.0008$ for the trajectory networks. The models are implemented using PyTorch [25]. Note that once the estimation is done, the plausible future of all vehicles can

	PF		VBPF	
	pos.	vel.	pos.	vel.
Scenario 1	4.35	2.50	**0.45**	**0.35**
Scenario 2	∞	∞	**0.32**	**0.34**
Scenario 3	4.31	2.40	**0.37**	**0.31**
Scenario 4	2.24	1.72	**0.27**	**0.26**
Scenario 5	5.72	2.65	**0.64**	**0.42**
Scenario 6	1.96	1.80	**0.34**	**0.40**

TABLE I
RMS POSITION AND VELOCITY ERRORS OVER TIME AND OVER VEHICLES FOR THE SIX CONSIDERED SCENARIOS. THE GAUSSIAN MEASUREMENT NOISE 99%-PROBABILITY CONFIDENCE INTERVAL SPANS $\pm\frac{1}{4}$STANDARD_VEHICLE_SIZE.

be predicted by means of the dynamic model learned this way.

The set $\{z^k_{0:n}\}$ used in the estimation scheme are also derived on the basis of the HighD dataset, by adding to each genuine vehicle position a Gaussian zero-mean measurement noise of given variance, *i.e.*, following (5) with

$$R^k_n = \begin{pmatrix} \sigma^2_{k(x)} & 0 \\ 0 & \sigma^2_{k(y)} \end{pmatrix}. \tag{16}$$

Values for $\sigma^2_{k(x)}, \sigma^2_{k(y)}$ are selected to simulate a measurement noise 99%-probability confidence interval of $\pm\frac{1}{4} \times$ (the size of a standard vehicle).

B. Results

To validate the method and compare the implemented filtering algorithms, 6 scenarios are built from the highD dataset. They comply with the assumptions made in Section II: common highway section; common three lanes; same direction of motion; no intersection; no access ramp.

- *Scenario 1* is composed of 8 vehicles driving across the whole highway segment without changing lane.
- *Scenario 2* features 6 vehicles driving across the whole highway segment with multiple changing lane maneuvers and a vehicle inserting itself between two others.
- *Scenario 3 and 4* is composed of 1 and 2 vehicles overtaking another one respectively.
- *Scenario 5 and 6* involves multiple overtakes and lane changes between 5 and 6 vehicles respectively.

The cumulative duration of all scenarios is approximately two minutes. The proposed Variational Bayes Particle Filter (VBPF) and a standard Particle Filter (PF) are compared. Their respective numbers of particles are set to 120 and 10000, so that the induced complexities are reasonable.

Table I shows the root mean squared position and velocity errors (RMSEs) over time and over vehicles for both filtering algorithms. These attest the good performances of the algorithm as a prerequisite to motion prediction. The VBPF leads to a good overall RMSE but is slightly optimistic. As for the PF, in view of the sharpness of the measurement likelihood function, its number of efficient particles drops to 1 right at the first couple of estimation steps. This leads to inconsistent estimates and a poor RMSE.

Figures IV-A sketches the position and velocity estimates along time together with their covariances on the second and third scenarios. The VBPF estimates are close to the ground-truth across the whole road segment for both scenarios.

V. PROSPECTS

Instead of considering diagonal matrices for the egocentric transition models, full covariance matrices will be considered. They can indeed lead to less spreaded prior dynamics, by capturing cross-correlation between state variables. This will be done by modifying the trajectory DNNs so as to learn the Cholesky square roots of the (non-diagonal) covariance matrices $\{\Sigma_m(x^k_{n-1}, \diamond)\}$.

Several improvements can be brought to the DNN architectures used for prior dynamics learning. Convolutional layers can be considered as a way of taking in account the spatial structure of the road scene data. Also, generative models such as CVAEs may be able to extract more meaningful information from the data. Last, degeneracy of Mixture Density Networks must be revisited.

The state-space can be augmented with a discrete index representing the vehicle type as this information can realistically be acquired from various sensors. Models can be re-learned accordingly.

On the computational side, parallelizing possibilities offered by the VBPF together with synchronization requirements will be investigated, in order to get real time performance.

REFERENCES

[1] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent," *ROBOMECH*, 2014, http://www.robomechjournal.com/content/1/1/1.

[2] S. Lefèvre, "Estimation du risque aux intersections pour applications sécuritaires avec véhicules communicants," Ph.D. dissertation, Grenoble Univ. (in French), 2013.

[3] J. Schulz, K. Hirsenkorn, J. Löchner, M. Werling, and D. Burschka, "Estimation of collective maneuvers through cooperative multi-agent planning," in *IEEE Intelligent Vehicles Symposium (IV'2017)*, Los Angeles, CA, 2017.

[4] G. Aoude, V. Desaraju, L. Stephens, and J. How, "Behavior classification algorithms at intersections and validation using naturalistic data," in *IEEE Intelligent Vehicles Symposium (IV'2011)*, Baden-Baden, Germany, 2011.

[5] N. Deo and M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *IEEE Intelligent Vehicles Symposium (IV'2018)*, Changshu, Suzhou, China, 2018.

[6] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr, "A combined model- and learning-based framework for interaction-aware maneuver prediction," *IEEE Trans. on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1538–1550, 2016.

[7] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Risk assessment at road intersections: Comparing intention and expectation," in *IEEE Intelligent Vehicles Symposium (IV'2012)*, Alcalá de Henares, Spain, 2012.

[8] Y. Hu, W. Zhan, and M. Tomizuka, "Probabilistic prediction of vehicle semantic intention and motion," in *IEEE Intelligent Vehicles Symposium (IV'2018)*, Changshu, Suzhou, China, 2018.

[9] ——, "A framework for probabilistic generic traffic scene prediction," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC'2018)*, Maui, HI, 2018.

[10] D. Sierra González, J. Dibangoye, and C. Laugier, "High-speed highway scene prediction based on driver models learned from demonstrations," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC'2016)*, Rio de Janeiro, Brazil, 2016.

[11] F. Altché and A. de la Fortelle, "An LSTM network for highway trajectory prediction," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC'2017)*, Yokohama, Japan, 2017.

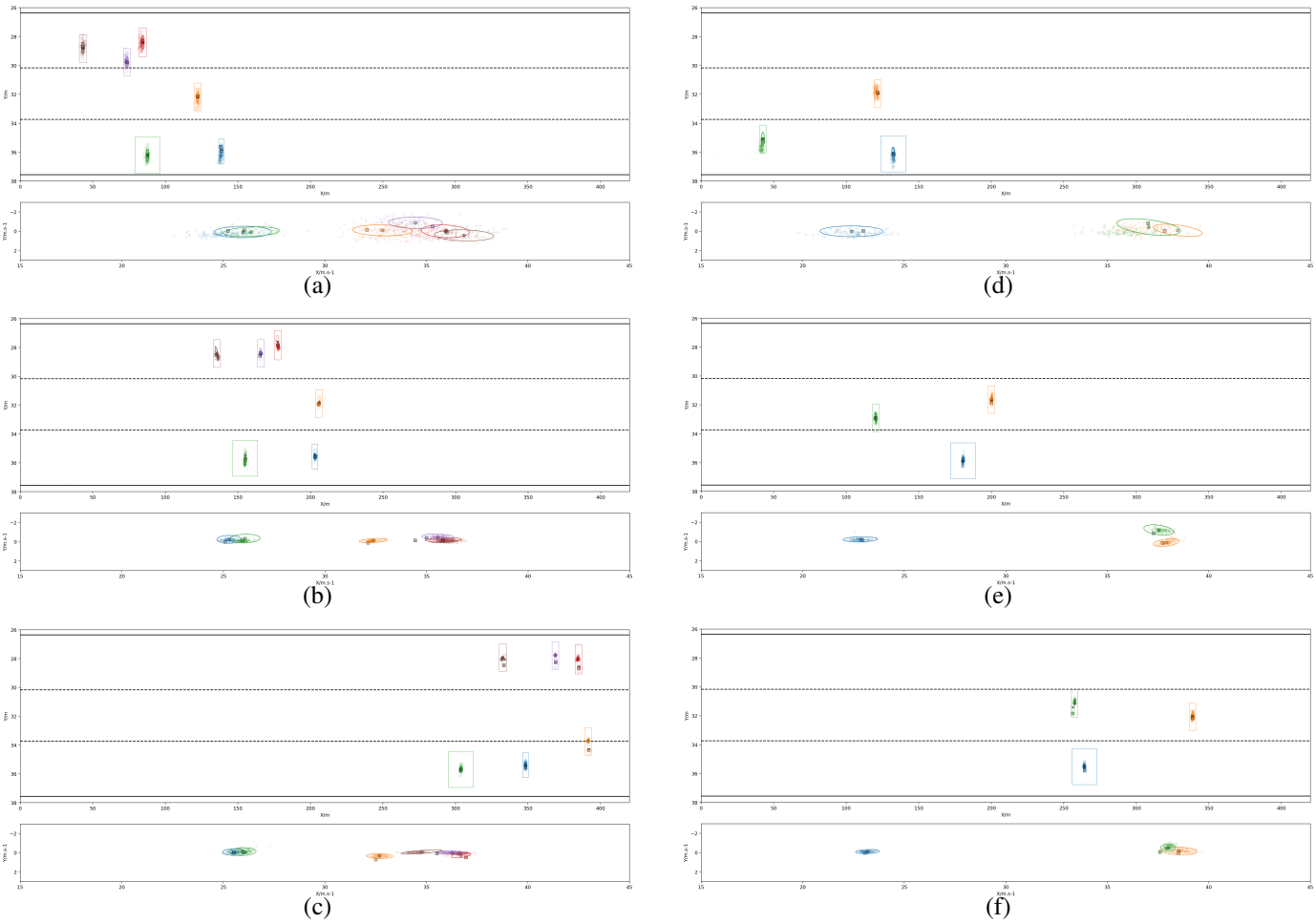

Fig. 3. Schematic top view of the application of VBPF to two highway scenarios for which the measurement noise 99%-probability confidence interval is $\pm\frac{1}{4}\times$ (the size of a standard vehicle); Estimated positions are depicted by circled markers. Their associated confidence ellipses should enclose the genuine positions (squared markers) with 99%-probability. Measured positions are represented by stars. Lanes are represented by straight and dotted lines. (a-b-c) : Throughout Scenario 2, even during strong interactions, all the vehicles are consistently estimated with convenient state error statistics. (d-e-f) : The three overtaking vehicles in Scenario 3 are also correctly estimated during the whole road segment.

[12] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC'2010)*, Madeira, Portugal, 2010.

[13] ——, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, 2015.

[14] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka, "Multiple model unscented kalman filtering in dynamic Bayesian networks for intention estimation and trajectory prediction," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC'2018)*, Maui, HI, 2018.

[15] ——, "Multiple model unscented kalman filtering in dynamic Bayesian networks for intention estimation and trajectory prediction," in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS'2018)*, Madrid, Spain, 2018.

[16] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.

[17] K. Plataniotis and D. Hatzinakos, "Gaussian mixtures and their applications to signal processing," in *Advanced Signal Processing Handbook: Theory and Implementation for Radar, Sonar, and Medical Imaging Real Time Systems*, S. Stergiopoulos, Ed. CRC Press, Boca Raton, 2000.

[18] C. Bishop, "Mixture density networks," Neural Computing Research Group, Aston Univ., Birmingham, United Kingdom, Tech. Rep., 1994.

[19] O. Makansi, E. Ilg, O. Cicek, and T. Brox, "Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'2019)*, Long Beach, CA, 2019.

[20] M. Jordan, Z. Ghahramani, T. Jaakkola, and S. L., *Machine Learning*. Kluwer, 1999, ch. An Introduction to Variational Methods for Graphical Models.

[21] D. Tzikas, A. Likas, and N. Galatsanos, "The variational approximation for Bayesian inference - life after the EM algorithm," *IEEE Signal Processing Magazine*, pp. 131–146, 2008.

[22] V. Smidl and A. Quinn, *The Variational Bayes Method in Signal Processing*. New York, NY: Springer, 2006.

[23] ——, "Variational Bayesian filtering," *IEEE Trans. on Signal Processing*, vol. 56, no. 10, pp. 5020–5030, 2008.

[24] B. Ait-El-Fquih and I. Hoteit, "A variational Bayesian multiple particle filtering scheme for large-dimensional systems," *IEEE Trans. on Signal Processing*, vol. 64, no. 20, pp. 5409–5422, 2016.

[25] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.