

Variational Inference With Parameter Learning Applied to Vehicle Trajectory Estimation

Jeremy Nathan Wong , David Juny Yoon , Angela P. Schoellig , and Timothy D. Barfoot 

Abstract—We present parameter learning in a Gaussian variational inference setting using only noisy measurements (i.e., no groundtruth). This is demonstrated in the context of vehicle trajectory estimation, although the method we propose is general. The letter extends the Exactly Sparse Gaussian Variational Inference (ESGVI) framework, which has previously been used for large-scale nonlinear batch state estimation. Our contribution is to additionally learn parameters of our system models (which may be difficult to choose in practice) within the ESGVI framework. In this letter, we learn the covariances for the motion and sensor models used within vehicle trajectory estimation. Specifically, we learn the parameters of a white-noise-on-acceleration motion model and the parameters of an Inverse-Wishart prior over measurement covariances for our sensor model. We demonstrate our technique using a 36 km dataset consisting of a car using lidar to localize against a high-definition map; we learn the parameters on a training section of the data and then show that we achieve high-quality state estimates on a test section, even in the presence of outliers. Lastly, we show that our framework can be used to solve pose graph optimization even with many false loop closures.

Index Terms—SLAM, localization.

I. INTRODUCTION

PROBABILISTIC state estimation is a core component of mobile robot navigation. While the estimation machinery is reasonably mature, there are robot model parameters that are difficult to determine from first principles and vary with each new platform and sensor. Our vision is to develop a learning framework that allows the deployment of a robot with arbitrary sensors onboard, and have it learn the model parameters required for estimation (and planning/control) solely from the sensor data. This can be viewed as a form of nonlinear system identification, although we will approach the problem using modern machine learning techniques.

In this letter, we show that we can learn the parameters of a nonlinear system in concert with a nonlinear batch state estimation framework, namely Exactly Sparse Gaussian Variational Inference (ESGVI) [1]. ESGVI exploits the fact that

Manuscript received February 24, 2020; accepted June 12, 2020. Date of publication July 7, 2020; date of current version July 16, 2020. This letter was recommended for publication by Associate Editor J. Guivant and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by Applanix Corporation and in part by the Natural Sciences and Engineering Research Council of Canada. (*Corresponding author: Jeremy Nathan Wong.*)

The authors are with the Institute for Aerospace Studies, University of Toronto, Toronto, ON M3H 5T6, Canada, and also with the Vector Institute for Artificial Intelligence, University of Toronto, Toronto, ON M5G 1M1, Canada (e-mail: jeremy.wong@robotics.utias.utoronto.ca; david.yoon@robotics.utias.utoronto.ca; angela.schoellig@utoronto.ca; schoellig@utias.utoronto.ca; tim.barfoot@utoronto.ca).

Digital Object Identifier 10.1109/LRA.2020.3007381

the joint likelihood between the observed measurements (data) and the latent state can be factored, which provides a family of scalable state estimation tools starting from a variational inference objective. To extend this to parameter learning, we use Expectation Maximization (EM). In the E-step, we fix all model parameters and optimize a bound on the data log-likelihood, the so-called Evidence Lower Bound (ELBO); this is equivalent to ESGVI latent state inference. In the M-step, we hold the latent state estimate fixed and optimize the ELBO for the parameters. Our method is general and applicable to any nonlinear system identification problem, even when the factorization of the joint likelihood has cycles (e.g., Simultaneous Localization and Mapping (SLAM)). Barfoot *et al.* [1] hint at the ESGVI extension to parameter learning, but do not demonstrate it in practice.

Our demonstration of parameter learning focuses on robot noise models. The noise models of the motion prior and observed measurements are often assumed to be known or tuned by trial and error. Our previous work demonstrated parameter learning for vehicle motion priors, but required accurate and complete (i.e., observation of the complete latent state) groundtruth [2]. However, often times, collecting such groundtruth is not possible or extremely expensive. We demonstrate the ability to learn these noise models from only noisy measurements. If groundtruth is available, we treat it simply as another (noisy) measurement that can be included in the framework. We also demonstrate that an Inverse-Wishart (IW) prior over the time-varying measurement covariances, using a Maximum A Posteriori (MAP) treatment in the variational setting, achieves outlier rejection in both parameter learning and latent state inference. We then demonstrate our parameter learning method on a real-world lidar dataset and a pose graph optimization problem created from a front-end pose graph SLAM algorithm. We show that our parameter learning method is able to handle both noisy measurements and outliers during training and testing.

In summary, the main contribution of this letter is a detailed investigation and experimental demonstration of parameter learning as part of ESGVI. Our application focuses on trajectory estimation, where we show nonlinear system identification using noisy measurements, without groundtruth. We also include outlier rejection in the variational setting by placing an IW prior over covariances.

In Section II we review previous work. An overview of the ESGVI framework with parameter learning is provided in III. Section IV presents the noise models we use and how we learn their parameters. An experimental evaluation of our parameter learning method is presented in V. In Section VI, we provide concluding remarks and discuss future work.

II. RELATED WORK

System identification has been an active research area for decades [3]–[6]. In the interest of space, we restrict our review of related work to the techniques that are most similar to our proposed approach. In the domain of parameter learning, the most common approach is to find parameters that maximize the likelihood of the data. One way to do this is to directly maximize the likelihood function with respect to the parameters [2], [7], [8]. This can be a difficult problem to solve, particularly when the model depends on missing or unobserved variables. In this case, an indirect approach can be taken by introducing a latent state to the problem, which can be estimated alongside of the parameters. This is known as Expectation Maximization (EM), an iterative algorithm that alternates between optimizing for a distribution over the latent state and the parameters.

Past work has shown how to estimate all the parameters of a linear dynamical system using EM, with Kalman smoothing in the E-step to update states and calculating analytic equations for parameter updates in the M-step [9]. There have also been methods that attempt parameter learning for nonlinear systems with EM. Ghahramani and Roweis [10] learn a full nonlinear model using Gaussian Radial Basis Functions (RBFs) to approximate the nonlinear expectations that would otherwise be intractable to compute. This method was applied to learn a simple sigmoid nonlinearity. Other methods approximate the required expectation using particle smoothing [11] or sigmapoint smoothing [7], [8], [12]. These methods, however, did not learn a full nonlinear model, but only learned parameters of a mostly predefined model (e.g., calibration parameters), and were tested only in simulation.

Unlike all these other methods, we use ESGVI within the EM parameter learning framework, which is a more general method not limited to problems with a specific factorization of the joint likelihood between the data and the latent state (e.g., smoothing problems with a block-tridiagonal inverse covariance). We also demonstrate a practical application of parameter learning by estimating the parameters of our motion prior and measurement noise models in a batch estimation framework.

While we are interested in batch estimation, previous work has investigated learning the noise model parameters of filters. Abbeel *et al.* [13] learn the noise model parameters of the Kalman Filter offline. However, these parameters are assumed to be static and do not vary with time. One popular area of study that handles changing covariances is Adaptive Kalman Filtering, where the measurement covariance is updated in an online fashion based on the statistics of the measurement innovation [14]–[16]. The measurement covariance in these cases is updated based solely on the data seen during inference, whereas we incorporate a prior.

Ko and Fox [17] apply Gaussian process regression to learn robot measurement and motion models, but because they do not exploit sparsity, need to resort to using their learned models in a filter estimation framework. We exploit sparsity for batch estimation. Recent methods take advantage of deep neural networks (DNNs) to learn the robot noise models [18]–[20] but in many cases require groundtruth to train the DNN. We bypass this requirement by simultaneously estimating a distribution over the latent state.

Barfoot *et al.* [1] show how to learn a constant covariance using ESGVI through EM but do not demonstrate it in practice. Our main contributions compared to [1] is demonstrating

parameter learning for a specific application and learning time-varying covariances by introducing an IW prior over our covariances, which enables outlier rejection. As an alternate method for outlier rejection, Chebroly *et al.* [21] use EM to learn a tuning parameter for M-estimation but treat their latent variables as point estimates. The IW distribution has been used as a prior over covariances before, but the parameters were assumed to be known [22]. We seek to learn at least some of the parameters of the prior.

To the best of our knowledge, the work we present in this letter is the first attempt at wrapping parameter learning into the ESGVI framework to solve a practical problem, and shows that we can achieve a robust extension of ESGVI (with an outlier rejection scheme) by placing an IW prior on our measurement covariances. We also show comparable trajectory estimation performance between learning parameters with and without groundtruth.

III. ESGVI WITH PARAMETER LEARNING

A. Variational Setup

We begin with the maximum-likelihood problem for the given data, \mathbf{z} , which is expressed as

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} p(\mathbf{z}|\boldsymbol{\theta}), \quad (1)$$

where $\boldsymbol{\theta}$ represents the parameters of our system that we wish to learn.

We define the loss that we wish to minimize as the negative log-likelihood of the data and introduce the latent state, \mathbf{x} . Applying the usual EM decomposition results in

$$\begin{aligned} \mathcal{L} = -\ln p(\mathbf{z}|\boldsymbol{\theta}) &= \underbrace{\int q(\mathbf{x}) \ln \left(\frac{p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})}{q(\mathbf{x})} \right) d\mathbf{x}}_{\leq 0} \\ &\quad - \underbrace{\int q(\mathbf{x}) \ln \left(\frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{x})} \right) d\mathbf{x}}_{\text{upper bound}}, \end{aligned} \quad (2)$$

where we define our approximate posterior as a multivariate Gaussian distribution, $q(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We now proceed iteratively in two steps, the expectation step (E-step) and the maximization step (M-step).¹

As commonly done in the EM framework, in both the E-step and the M-step, we optimize the upper bound term in (2), which is also known as the (negative) Evidence Lower Bound (ELBO). Using the expression for the entropy, $-\int q(\mathbf{x}) \ln q(\mathbf{x}) d\mathbf{x}$, for a Gaussian and dropping constants, the upper bound term is written as the loss functional of ESGVI,

$$V(q|\boldsymbol{\theta}) = \mathbb{E}_q[\phi(\mathbf{x}|\boldsymbol{\theta})] + \frac{1}{2} \ln (|\boldsymbol{\Sigma}^{-1}|), \quad (3)$$

where we define $\phi(\mathbf{x}|\boldsymbol{\theta}) = -\ln p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$, $\mathbb{E}_q[\cdot]$ is the expectation conditioned on the distribution $q(\mathbf{x})$, and $|\cdot|$ is the matrix determinant. We drop \mathbf{z} in the notation for convenience as our expectation is over \mathbf{x} .

Taking the derivatives of the loss functional with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}^{-1}$, Barfoot *et al.* [1] developed a Newton-style iterative

¹We are working with the negative log-likelihood so we are technically applying Expectation Minimization, but the acronym stays the same.

optimizer to update our estimate of $q(\mathbf{x})$. We summarize the optimization scheme here as

$$(\Sigma^{-1})^{(i+1)} = \sum_{k=1}^K \mathbf{P}_k^T \mathbb{E}_{q_k^{(i)}} \left[\frac{\partial^2 \phi_k(\mathbf{x}_k | \boldsymbol{\theta})}{\partial \mathbf{x}_k^T \partial \mathbf{x}_k} \right] \mathbf{P}_k, \quad (4a)$$

$$(\Sigma^{-1})^{(i+1)} \delta \boldsymbol{\mu} = - \sum_{k=1}^K \mathbf{P}_k^T \mathbb{E}_{q_k^{(i)}} \left[\frac{\partial \phi_k(\mathbf{x}_k | \boldsymbol{\theta})}{\partial \mathbf{x}_k^T} \right], \quad (4b)$$

$$\boldsymbol{\mu}^{(i+1)} = \boldsymbol{\mu}^{(i)} + \delta \boldsymbol{\mu}, \quad (4c)$$

where superscript i is used to denote variables at the i^{th} iteration. We have exploited the factorization of the joint log-likelihood into K factors as

$$\phi(\mathbf{x} | \boldsymbol{\theta}) = \sum_{k=1}^K \phi_k(\mathbf{x}_k | \boldsymbol{\theta}). \quad (5)$$

For generality we have each factor, ϕ_k , affected by the entire parameter set, $\boldsymbol{\theta}$, but in practice it can be a subset. \mathbf{P}_k is a projection matrix that extracts \mathbf{x}_k from \mathbf{x} (i.e. $\mathbf{x}_k = \mathbf{P}_k \mathbf{x}$). The marginal of q associated with \mathbf{x}_k is

$$q_k(\mathbf{x}_k) = \mathcal{N}(\mathbf{P}_k \boldsymbol{\mu}, \mathbf{P}_k \Sigma \mathbf{P}_k^T). \quad (6)$$

Critical to the efficiency of the ESGVI framework is the ability to compute the required marginals in (4a) and (4b), without ever constructing the complete (dense) covariance matrix, Σ . A sparse solver based on the method of Takahashi *et al.* [23] is used to achieve this in [1].

The expectations in (4a) and (4b) can be approximated using Gaussian cubature samples (e.g., sigmapoints) of the marginal posterior. Importantly, approximating the expectations at only the mean of the posterior is equivalent to the MAP batch optimization with Newton's method. Barfoot *et al.* [1] also provide a derivative-free optimization scheme with only Gaussian cubature, which we do not show here.

In the M-step, we hold $q(\mathbf{x})$ fixed and optimize the upper bound for the parameters, $\boldsymbol{\theta}$. We can optimize for $\boldsymbol{\theta}$ by taking the derivative of the loss functional as follows:

$$\begin{aligned} \frac{\partial V(q | \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_q[\phi(\mathbf{x} | \boldsymbol{\theta})] = \frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_q \left[\sum_{k=1}^K \phi_k(\mathbf{x}_k | \boldsymbol{\theta}) \right] \\ &= \sum_{k=1}^K \mathbb{E}_{q_k} \left[\frac{\partial}{\partial \boldsymbol{\theta}} \phi_k(\mathbf{x}_k | \boldsymbol{\theta}) \right]. \end{aligned} \quad (7)$$

In the last step, the expectation reduces from being over the full Gaussian, q , to the marginal associated with the variables in each factor, q_k . We can then set the derivative to zero and isolate $\boldsymbol{\theta}$ for a critical point, formulating an M-step. If isolation is not possible, we can use the gradient in (7) for a partial M-step, which is known as Generalized Expectation Maximization (GEM) [24].

B. Alternate Loss Functional

In the E-step, we hold $\boldsymbol{\theta}$ fixed and optimize $q(\mathbf{x})$ for the best possible Gaussian fit. Barfoot *et al.* [1] present an alternate, Gauss-Newton-style loss functional for when the negative log-likelihood takes the form

$$\phi(\mathbf{x} | \mathbf{w}) = \frac{1}{2} (\mathbf{e}(\mathbf{x})^T \mathbf{w}^{-1} \mathbf{e}(\mathbf{x}) - \ln(|\mathbf{w}^{-1}|)), \quad (8)$$

where $\boldsymbol{\theta}$ is now a covariance matrix, \mathbf{w} . With Jensen's inequality [25] and dropping the second term since \mathbf{w} is a constant in the E-step, we can write

$$\mathbb{E}_q[\mathbf{e}(\mathbf{x})]^T \mathbf{w}^{-1} \mathbb{E}_q[\mathbf{e}(\mathbf{x})] \leq \mathbb{E}_q[\mathbf{e}(\mathbf{x})^T \mathbf{w}^{-1} \mathbf{e}(\mathbf{x})]. \quad (9)$$

Motivated by this relationship, we can define a new loss functional for the E-step as

$$V'(q) = \frac{1}{2} \mathbb{E}_q[\mathbf{e}(\mathbf{x})]^T \mathbf{w}^{-1} \mathbb{E}_q[\mathbf{e}(\mathbf{x})] + \frac{1}{2} \ln(|\Sigma^{-1}|), \quad (10)$$

which is a conservative approximation of $V(q)$, appropriate for mild nonlinearities and/or concentrated posteriors. The alternate loss functional is simpler to implement in practice as it does not require the second derivative of the factors.² Also note how evaluating the expectation only at the mean of the posterior is equivalent to MAP Gauss-Newton.

IV. PARAMETER LEARNING FOR ROBOT NOISE MODELS

A. Constant Covariance

Barfoot *et al.* [1] outline parameter learning (M-step) for constant covariance noise models, which we summarize here. Our loss functional is

$$V(q | \mathbf{w}) = \mathbb{E}_q[\phi^m(\mathbf{x} | \mathbf{w})] + \frac{1}{2} \ln(|\Sigma^{-1}|). \quad (11)$$

This expression is similar to (8), but we can exploit the factorization of $\phi^m(\mathbf{x} | \mathbf{w})$ to write:

$$\begin{aligned} \phi^m(\mathbf{x} | \mathbf{w}) &= \sum_{k=1}^K \phi_k^m(\mathbf{x}_k | \mathbf{w}) \\ &= \sum_{k=1}^K \frac{1}{2} (\mathbf{e}_k(\mathbf{x}_k)^T \mathbf{w}^{-1} \mathbf{e}_k(\mathbf{x}_k) - \ln(|\mathbf{w}^{-1}|)), \end{aligned} \quad (12)$$

where the K factors (in practice, it could be a subset) are affected by the unknown parameter \mathbf{w} , a constant covariance matrix. Evaluating the derivative, as shown in (7), with respect to \mathbf{w}^{-1} and setting to zero for computing a minimum results in the optimal \mathbf{w} to be

$$\mathbf{w} = \frac{1}{K} \sum_{k=1}^K \mathbb{E}_{q_k} [\mathbf{e}_k(\mathbf{x}_k) \mathbf{e}_k(\mathbf{x}_k)^T], \quad (13)$$

which can be approximated with Gaussian cubature if the error functions, $\mathbf{e}_k(\mathbf{x}_k)$, are nonlinear.

Alternatively, we can choose to linearize $\mathbf{e}_k(\mathbf{x}_k)$ at the posterior marginal, $q_k = \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$, resulting in the following M-step:

$$\begin{aligned} \mathbf{w} &\approx \frac{1}{K} \sum_{k=1}^K \mathbb{E}_{q_k} \left[(\mathbf{e}_k(\boldsymbol{\mu}_k) + \mathbf{e}_k \delta \mathbf{x}_k) (\mathbf{e}_k(\boldsymbol{\mu}_k) + \mathbf{e}_k \delta \mathbf{x}_k)^T \right] \\ &= \frac{1}{K} \sum_{k=1}^K (\mathbf{e}_k(\boldsymbol{\mu}_k) \mathbf{e}_k^T(\boldsymbol{\mu}_k) + \mathbf{e}_k \Sigma_k \mathbf{e}_k^T), \end{aligned} \quad (14)$$

²Another alternative is the derivative-free optimization scheme with cubature sampling, at the cost of requiring more cubature samples (i.e., more computation). A derivative-free scheme for the alternate loss functional is also possible [1].

where $\mathbf{e}_k = \frac{\partial \mathbf{e}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \Big|_{\mathbf{x}_k = \boldsymbol{\mu}_k}$.

B. White-Noise-on-Acceleration Prior on Latent State

We next demonstrate parameter learning for the situation where the covariance of a factor is indirectly estimated through another quantity. Consider the example of a white-noise-on-acceleration (WNOA) motion prior on the latent state, where the parameter we wish to estimate is the power-spectral density matrix, \mathbf{Q}_c [26]. More specifically, let us study the application of the prior in $SE(3)$ [27], which is defined as follows:

$$\begin{aligned} \dot{\mathbf{T}}(t) &= \boldsymbol{\omega}(t) \wedge \mathbf{T}(t), \\ \dot{\boldsymbol{\omega}} &= \mathbf{w}(t), \quad \mathbf{w}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_c \delta(t - t')), \end{aligned} \quad (15)$$

where $\mathbf{T}(t) \in SE(3)$ is the pose expressed as a transformation matrix, $\boldsymbol{\omega}(t) \in \mathbb{R}^6$ is the body-centric velocity, $\mathbf{w}(t) \in \mathbb{R}^6$ is a zero-mean, white-noise Gaussian process, and the operator, \wedge , transforms an element of \mathbb{R}^6 into a member of Lie algebra, $\mathfrak{se}(3)$. The state at time t_k is $\mathbf{x}_k = \{\mathbf{T}_k, \boldsymbol{\omega}_k\}$, and similarly, $\mathbf{x}_{k-1,k}$ is the state at two consecutive times, t_{k-1} and t_k .

We express the factors of our loss functional from (3), but with only WNOA prior factors for simplicity:

$$\begin{aligned} \phi^p(\mathbf{x}|\mathbf{Q}_c) &= \sum_{k=2}^K \phi_k^p(\mathbf{x}_{k-1,k}|\mathbf{Q}_c) \\ &= \sum_{k=2}^K \frac{1}{2} (\mathbf{e}_{p,k}^T \mathbf{Q}_c^{-1} \mathbf{e}_{p,k} + \ln |\mathbf{Q}_c|), \end{aligned} \quad (16)$$

where

$$\mathbf{e}_{p,k} = \begin{bmatrix} \ln(\mathbf{T}_k \mathbf{T}_{k-1}^{-1})^\vee - (t_k - t_{k-1}) \boldsymbol{\omega}_{k-1} \\ \mathcal{J}^{-1}(\ln(\mathbf{T}_k \mathbf{T}_{k-1}^{-1})^\vee) \boldsymbol{\omega}_k - \boldsymbol{\omega}_{k-1} \end{bmatrix}, \quad (17)$$

and the covariance of the prior, \mathbf{Q}_k , is defined as [26]

$$\mathbf{Q}_k = \mathbf{Q}_{\Delta t} \otimes \mathbf{Q}_c, \quad \mathbf{Q}_k^{-1} = \mathbf{Q}_{\Delta t}^{-1} \otimes \mathbf{Q}_c^{-1},$$

$$\mathbf{Q}_{\Delta t} = \begin{bmatrix} \frac{1}{3} \Delta t^3 & \frac{1}{2} \Delta t^2 \\ \frac{1}{2} \Delta t^2 & \Delta t \end{bmatrix}, \quad \mathbf{Q}_{\Delta t}^{-1} = \begin{bmatrix} 12 \Delta t^{-3} & -6 \Delta t^{-2} \\ -6 \Delta t^{-2} & 4 \Delta t^{-1} \end{bmatrix},$$

where \otimes is the Kronecker product. Solving for the derivative with respect to $Q_{c_{ij}}$, the (i, j) matrix element of \mathbf{Q}_c , we have

$$\begin{aligned} \frac{\partial V(q|\boldsymbol{\theta})}{\partial Q_{c_{ij}}} &= \frac{1}{2} \text{tr} \left(\sum_{k=2}^K \mathbb{E}_{q_{k-1,k}} [\mathbf{e}_{p,k} \mathbf{e}_{p,k}^T] (\mathbf{Q}_{\Delta t}^{-1} \otimes \mathbf{1}_{ij}) \right) \\ &\quad - \frac{1}{2} (K-1) \dim(\mathbf{Q}_{\Delta t}) \mathbf{Q}_{c_{ij}}, \end{aligned} \quad (18)$$

where $q_{k-1,k}$ is the marginal posterior at two consecutive times, t_{k-1} and t_k . Setting the derivative to zero, the optimal estimate of our parameter is then

$$Q_{c_{ij}} = \frac{\text{tr} \left(\sum_{k=2}^K \mathbb{E}_{q_{k-1,k}} [\mathbf{e}_{p,k} \mathbf{e}_{p,k}^T] (\mathbf{Q}_{\Delta t}^{-1} \otimes \mathbf{1}_{ij}) \right)}{\dim(\mathbf{Q}_{\Delta t}) (K-1)}. \quad (19)$$

As explained for (13), the expectation in (19) can be approximated with Gaussian cubature or linearization.

C. Inverse-Wishart Prior on Covariance

We further extend covariance estimation by incorporating a prior. Instead of treating the covariance as a static parameter, we treat it as a random variable and place an IW prior on it. We then learn some of the parameters of the prior. In order to do so, we redefine our joint likelihood as

$$p(\mathbf{x}, \mathbf{z}, \mathbf{R}) = p(\mathbf{x}, \mathbf{z}|\mathbf{R}) p(\mathbf{R}), \quad (20)$$

where now we also include the covariances, $\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_K\}$, as random variables. We also redefine our posterior estimate to be

$$q'(\mathbf{x}) = q(\mathbf{x}) s(\mathbf{R}), \quad (21)$$

a product between a Gaussian $q(\mathbf{x})$ and a posterior distribution for the covariances, $s(\mathbf{R})$.

The upper bound term in the EM decomposition of (2) can now be written as

$$- \int \int q(\mathbf{x}) s(\mathbf{R}) \ln \left(\frac{p(\mathbf{x}, \mathbf{z}|\mathbf{R}) p(\mathbf{R})}{q(\mathbf{x}) s(\mathbf{R})} \right) d\mathbf{x} d\mathbf{R}. \quad (22)$$

We define the posterior over the covariances as

$$s(\mathbf{R}) = \delta(\mathbf{R} - \boldsymbol{\Upsilon}), \quad (23)$$

where $\delta(\cdot)$ is the Dirac delta function (interpreted as a probability density function) and $\boldsymbol{\Upsilon} = \{\boldsymbol{\Upsilon}_1, \boldsymbol{\Upsilon}_2, \dots, \boldsymbol{\Upsilon}_K\}$ is the set of optimal covariances. The upper bound now simplifies to

$$\begin{aligned} &- \int q(\mathbf{x}) \ln (p(\mathbf{x}, \mathbf{z}|\boldsymbol{\Upsilon}) p(\boldsymbol{\Upsilon})) d\mathbf{x} \\ &+ \int q(\mathbf{x}) \ln q(\mathbf{x}) d\mathbf{x} + \underbrace{\int s(\mathbf{R}) \ln s(\mathbf{R}) d\mathbf{R}}_{\text{indep. of } \boldsymbol{\Upsilon}}, \end{aligned} \quad (24)$$

where we have abused notation and written $p(\mathbf{R} = \boldsymbol{\Upsilon})$ as $p(\boldsymbol{\Upsilon})$, and similarly will later write $p(\mathbf{R}_k = \boldsymbol{\Upsilon}_k)$ as $p(\boldsymbol{\Upsilon}_k)$. We view our selection of the delta function as a convenient way of showing how we can approximate a Gaussian distribution for the trajectory and a MAP approximation of the covariances in a single variational framework. The last term is the differential entropy of a Dirac delta function, and because it is independent of our variational parameter, $\boldsymbol{\Upsilon}$, we choose to drop it from our loss functional.

We assume $p(\boldsymbol{\Upsilon})$ factors as $p(\boldsymbol{\Upsilon}) = \prod_{k=1}^K p(\boldsymbol{\Upsilon}_k)$. We apply an IW prior over our covariances by defining

$$p(\boldsymbol{\Upsilon}_k) = \frac{|\boldsymbol{\Psi}|^{\nu/2}}{2^{\frac{\nu d}{2}} \Gamma_d(\frac{\nu}{2})} |\boldsymbol{\Upsilon}_k|^{-\frac{\nu+d+1}{2}} \exp \left(-\frac{1}{2} \text{tr}(\boldsymbol{\Psi} \boldsymbol{\Upsilon}_k^{-1}) \right), \quad (25)$$

where d is the dimension of $\boldsymbol{\Upsilon}_k$, $\boldsymbol{\Psi} \in \mathbb{R}^{d \times d} > 0$ is the scale matrix, $\nu > d - 1$ is the degrees-of-freedom (DOF), and $\Gamma_d(\cdot)$ is the multivariate Gamma function. The IW distribution has been used as a prior over covariance matrices before, which led to outlier rejection at inference [28], [29], but the parameters of the prior were assumed to be known. We choose to estimate the scale matrix parameter $\boldsymbol{\Psi}$ and leave the degrees-of-freedom ν as a metaparameter.

Now we define our factors as

$$\begin{aligned} -\ln(\Upsilon) &= \sum_{k=1}^K -\ln p(\Upsilon_k) \\ &= \sum_{k=1}^K \phi_k^w(\Upsilon_k|\Psi) = \phi^w(\Upsilon|\Psi). \end{aligned} \quad (26)$$

Dropping constant terms, the loss functional can finally be written as

$$\begin{aligned} V(q|\Upsilon, \Psi) &= \sum_{k=1}^K \mathbb{E}_{q_k} [\phi_k^m(\mathbf{x}_k|\Upsilon_k) + \phi_k^w(\Upsilon_k|\Psi)] \\ &\quad + \frac{1}{2} \ln(|\Sigma^{-1}|), \end{aligned} \quad (27)$$

where

$$\begin{aligned} \phi_k^m(\mathbf{x}_k|\Upsilon_k) &= \frac{1}{2} (\mathbf{e}_k(\mathbf{x}_k)^T \Upsilon_k^{-1} \mathbf{e}_k(\mathbf{x}_k) - \ln(|\Upsilon_k^{-1}|)), \quad (28) \\ \phi_k^w(\Upsilon_k|\Psi) &= -\frac{\alpha-1}{2} \ln|\Upsilon_k^{-1}| - \frac{\nu}{2} \ln|\Psi| + \frac{1}{2} \text{tr}(\Psi \Upsilon_k^{-1}), \end{aligned} \quad (29)$$

with $\alpha = \nu + d + 2$.

In the E-step, we hold Ψ fixed and optimize for Υ_k , which we accomplish by taking the derivative of the loss functional as follows:

$$\frac{\partial V}{\partial \Upsilon_k^{-1}} = \frac{1}{2} \mathbb{E}_{q_k} [\mathbf{e}_k(\mathbf{x}_k) \mathbf{e}_k(\mathbf{x}_k)^T] - \frac{1}{2} \alpha \Upsilon_k + \frac{1}{2} \Psi. \quad (30)$$

Setting the derivative to zero,

$$\begin{aligned} \Upsilon_k &= \frac{1}{\alpha} \Psi + \frac{1}{\alpha} \mathbb{E}_{q_k} [\mathbf{e}_k(\mathbf{x}_k) \mathbf{e}_k(\mathbf{x}_k)^T] \\ &= \frac{\alpha-1}{\alpha} \underbrace{\left(\frac{\Psi}{\alpha-1} \right)}_{\text{IW mode}} + \frac{1}{\alpha} \mathbb{E}_{q_k} [\mathbf{e}_k(\mathbf{x}_k) \mathbf{e}_k(\mathbf{x}_k)^T], \end{aligned} \quad (31)$$

where we see the optimal Υ_k is a weighted average between the mode of the IW distribution and the optimal static covariance estimate from (13) at a single marginal factor. Since our E-step in ESGVI is already iterative, we can seamlessly extend it by applying (31) as iteratively reweighted least squares (IRLS).

In the M-step, we hold Υ fixed and optimize for Ψ , which we accomplish by taking the derivative of the loss functional as follows:

$$\frac{\partial V}{\partial \Psi} = \sum_{k=1}^K \left(-\frac{\nu}{2} \Psi^{-1} + \frac{1}{2} \mathbf{R}_k^{-1} \right). \quad (32)$$

Setting the derivative to zero,

$$\Psi^{-1} = \frac{1}{K\nu} \sum_{k=1}^K \Upsilon_k^{-1}. \quad (33)$$

Applying (31) in the E-step and (33) in the M-step, we found that our optimization scheme was still ill-posed, and our covariance estimates tended toward the positive-definite boundary (i.e., the zero matrix). We propose constraining the determinant of Ψ to be a constant β , which can be thought of as constraining the volume of the uncertainty ellipsoid of the corresponding measurements

to be fixed. We accomplish this by scaling the latest Ψ update as follows:

$$\Psi_{\text{constrained}} \leftarrow (\beta |\Psi|^{-1})^{\frac{1}{d}} \Psi. \quad (34)$$

We then rely on the noise models of other factors (e.g., the motion prior) to adapt to our selection of β during training.

V. EXPERIMENTAL VALIDATION

To evaluate our parameter learning method, we will be working with the vehicle dataset collected and used in our previous work [2]. The dataset consists of 36 km of driving, with Velodyne VLS-128 lidar data and an Applanix POS-LV positioning system. There are two sources of 6-DOF vehicle pose measurements. The first is from the POS-LV system, which we treat as groundtruth. The second is from a lidar localization system from Applanix, which localizes the lidar data to a prebuilt high-definition map.

We use Route A,³ our 16 km long training set, to learn the parameters of our noise models. For inference, we perform a batch trajectory optimization on Route B,⁴ our 20 km long test set, using the learned noise model parameters of our motion prior and measurements.

Finally, we evaluate our method on a pose graph optimization problem with false loop closures.

A. Training With and Without Groundtruth

In Experiment A, our first experiment, we only use the lidar localization measurements to learn our model parameters (training without groundtruth). As a benchmark, we also learn another set of model parameters where we additionally include groundtruth poses in our training (training with incomplete groundtruth). This is different from our previous work [2] where the training method required groundtruth of the entire state (training with complete groundtruth), which for our problem setup is pose and body-centric velocity. Additionally, in that paper, the measurement covariances were assumed to be known and not learned.

The loss functional corresponding to this experiment is

$$\begin{aligned} V(q|\Upsilon, \Psi, \mathbf{w}_{gt}, \mathbf{Q}_c) &= \mathbb{E}_{q'} [\phi^p(\mathbf{x}|\mathbf{Q}_c) + \phi^m(\mathbf{x}|\mathbf{w}_{gt}) \\ &\quad + \phi^m(\mathbf{x}|\Upsilon) + \phi^w(\Upsilon|\Psi)] + \frac{1}{2} \ln(|\Sigma^{-1}|), \end{aligned} \quad (35)$$

where $\phi^p(\mathbf{x}|\mathbf{Q}_c)$ are the WNOA prior factors, $\phi^m(\mathbf{x}|\mathbf{w}_{gt})$ are the groundtruth factors (when available), and $\phi^m(\mathbf{x}|\Upsilon)$ and $\phi^w(\Upsilon|\Psi)$ are the lidar measurement factors with an IW prior over the covariances. See (16) for the definition of $\phi^p(\mathbf{x}|\mathbf{Q}_c)$ and (12) for the definition of $\phi^m(\mathbf{x}|\mathbf{w}_{gt})$ and $\phi^m(\mathbf{x}|\Upsilon)$. For the definition of $\phi^w(\Upsilon|\Psi)$, see (26) and (29).

The WNOA error function (required for ϕ^p) is shown in (17), and the error function for pose measurements (required for ϕ^m) is defined as

$$\mathbf{e}_{m,k} = \ln(\mathbf{T}_k \mathbf{T}_{\text{meas},k}^{-1}). \quad (36)$$

The estimation problem in this experiment can be represented by the factor graph in Fig. 1, where we can train with or

³Map available at: <https://tinyurl.com/trjgxaaj>

⁴Map available at: <https://tinyurl.com/r5m78nq>

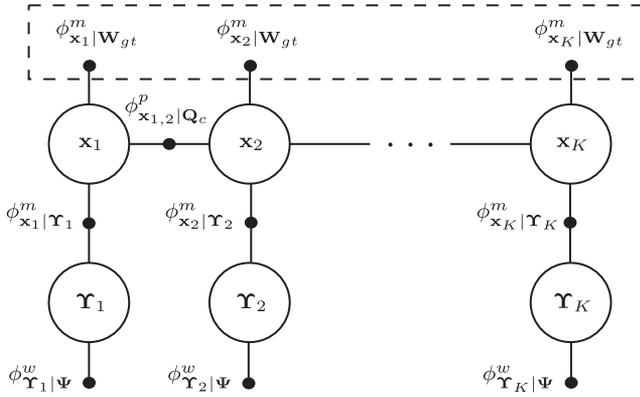


Fig. 1. Factor graph for our vehicle estimation problem in Experiment A (see Section V-A). White circles represent random variables to be estimated (vehicle state \mathbf{x} and measurement covariances Υ). Small black dots represent factors in the joint likelihood of the data and the state. Binary motion prior factors, $\phi_{\mathbf{x}_{k-1,k}^p | \mathbf{Q}_c}$, depend on parameter \mathbf{Q}_c . Unary groundtruth pose factors (if available), $\phi_{\mathbf{x}_k}^m | \mathbf{w}_{gt}$, depend on parameter \mathbf{w}_{gt} . Factors $\phi_{\mathbf{x}_k}^m | \Upsilon_k$ and $\phi_{\Upsilon_k}^w | \Psi$ are for applying an Inverse-Wishart prior over our measurement pose covariances, Υ , and depend on parameter Ψ . We are able to learn parameters \mathbf{Q}_c and Ψ , even without groundtruth factors (factors inside dashed box).

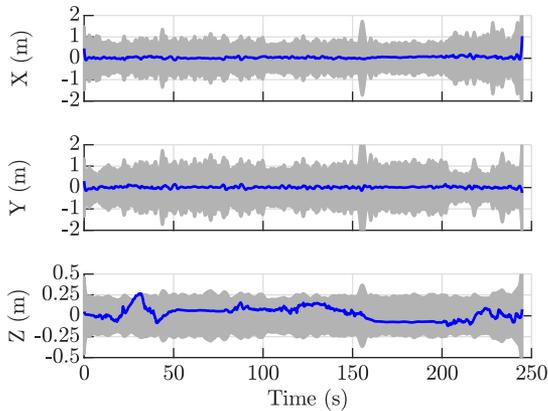


Fig. 2. Experiment A - Error plots (blue lines) along with the 3σ covariance envelopes (gray background) when parameters are trained without groundtruth.

without the groundtruth factors, which are shown inside the dashed box. For the sake of conciseness in our notation, we denote $\phi^p(\mathbf{x}_{k-1,k} | \mathbf{Q}_c)$ as $\phi_{\mathbf{x}_{k-1,k}^p | \mathbf{Q}_c}^p$, $\phi^m(\mathbf{x}_k | \mathbf{w}_{gt})$ as $\phi_{\mathbf{x}_k}^m | \mathbf{w}_{gt}$, $\phi^m(\mathbf{x}_k | \Upsilon_k)$ as $\phi_{\mathbf{x}_k}^m | \Upsilon_k$, and $\phi^w(\Upsilon_k | \Psi)$ as $\phi_{\Upsilon_k}^w | \Psi$.

We choose to fix the parameters to $\nu = 6$ and $\beta = 1$ and learn the parameters Ψ , \mathbf{w}_{gt} (when groundtruth is available), and \mathbf{Q}_c . For both sets of learned parameters, we then perform trajectory estimation on our test set, where we only use the lidar localization measurements with our learned covariance model and our learned motion prior.

Fig. 2 shows the error plots where we have trained without groundtruth for our estimated x , y , and z positions, along with their 3σ covariance envelopes. As can be seen, the errors consistently remain within the covariance envelopes. We do however note that our estimator appears to be underconfident. We believe that this is a result of our decision to constrain $|\Psi| = \beta = 1$ in order for our training method to work in practice. This decision is analogous to fixing the volume of the covariance ellipsoid to be constant. In doing so, we relied on the learned covariance of the

TABLE I

EXPERIMENT A - COMPARISON OF TRANSLATIONAL ERRORS ON TEST SET BETWEEN TRAINING WITH COMPLETE GROUNDTRUTH, WITH INCOMPLETE GROUNDTRUTH, AND WITHOUT GROUNDTRUTH. WE NOTE THAT THE FIRST COLUMN, OUR PREVIOUS WORK, DID NOT LEARN THE MEASUREMENT COVARIANCES

Seq no.	Trained with complete GT [2] (m)	Trained with incomplete GT (m)	Trained without GT (m)
0	0.0690	0.0720	0.0717
1	0.0888	0.1003	0.0925
2	0.4071	0.4148	0.4106
3	0.1947	0.1908	0.1847
4	0.2868	0.2866	0.2820
5	0.5703	0.5592	0.5549
6	0.3292	0.3014	0.2965
7	0.2207	0.2248	0.2230
8	0.1115	0.1151	0.1199
9	0.0979	0.1026	0.0997
overall	0.2376	0.2368	0.2335

TABLE II

EXPERIMENT A - ANALYSIS OF HOW INCREASING NOISE ON MEASUREMENTS AFFECTS THE PARAMETER LEARNING METHOD. EVEN WITH MEASUREMENT ERRORS OF OVER 1.6 m, THE ERRORS ON THE ESTIMATED TRAJECTORY ARE UNDER 0.5 m

Measurement errors (m)	Estimated trajectory errors (m)
0.2407	0.2335
0.5010	0.2909
0.8653	0.3289
1.2481	0.3936
1.6383	0.4566

motion prior to adjust relative to the measurement covariances. The posterior mean is unaffected by this choice but not the posterior covariance.

Table I shows the resulting mean translational errors from both training methods on all test sequences. We also include the results from our previous work where we trained using complete groundtruth for comparison.

While we achieve very similar errors across all training methods, the benefit is that we now do not require any groundtruth. Neither of the three training methods seem to outperform the others. We believe this is because our lidar localization measurements are quite accurate relative to groundtruth [2].

To further validate our method and show that we can indeed train with noisy measurements, we decided to artificially add additional noise to the measurements, where the noise statistics are unknown to the training process. We use the following $SE(3)$ perturbation scheme [28], [30] to inject noise into the position portion of our pose measurements:

$$\mathbf{T}_{\text{noisy}} = \exp(\xi^\wedge) \mathbf{T}_{\text{meas}}, \quad (37)$$

where $\xi = \begin{bmatrix} \xi_{1:3} \\ \mathbf{0} \end{bmatrix}$, $\xi_{1:3} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$.

We vary σ from 0.25 m to 1 m, injecting the same amount of noise into the test measurements and training measurements.

Table II shows how our test errors change with increasing noise on measurements in both our training and test set. While measurement error increases significantly, up to over 1.6 m, we are still able to achieve translational errors of below 0.5 m on our estimated trajectory. This shows that we are still able to learn reasonable parameters of our system even without any groundtruth and quite noisy measurements.

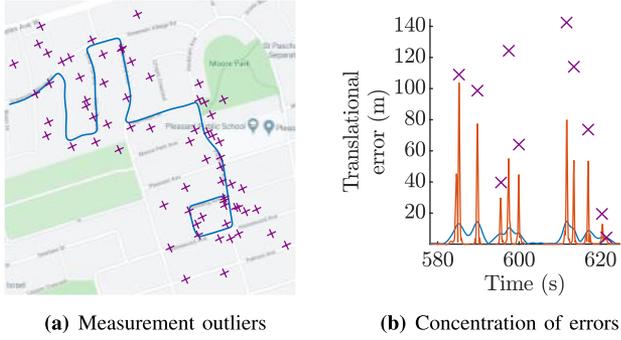


Fig. 3. Experiments B & C - (a) Measurement outliers (purple) overlaid with the groundtruth trajectory (blue). (b) Translational errors for the static covariance method on a portion of the test set containing measurement outliers (purple) when training with and without outliers (Experiments B in blue and C in orange, respectively).

B. Training and Testing With Measurement Outliers

In Experiment B, we show that estimating time-varying covariances for each of our measurements with an IW prior results in outlier rejection. We artificially introduce outliers in our training and test set using the following method. With 5% probability, we apply the following perturbation to the actual pose measurement:

$$\mathbf{T}_{\text{outlier}} = \exp(\boldsymbol{\xi}^{\wedge})\mathbf{T}_{\text{meas}}, \quad (38)$$

with $\boldsymbol{\xi} \in \mathbb{R}^6 \sim \mathcal{U}(-200, 200)$.

Fig. 3(a) shows an example of the measurement outliers on sequence 3 of our test set.

We now seek to compare the performance between the cases where we have treated the measurement covariance, \mathbf{w} , as a static parameter to be learned, and where we have treated the measurement covariance at each time as a random variable and learn the parameter, Ψ , of the IW prior.

The loss functional corresponding to the static measurement covariance is

$$V(q'|\mathbf{w}, \mathbf{Q}_c) = \mathbb{E}_{q'}[\phi^p(\mathbf{x}|\mathbf{Q}_c) + \phi^m(\mathbf{x}|\mathbf{w})] + \frac{1}{2} \ln(|\Sigma^{-1}|), \quad (39)$$

where as for the IW prior on the measurement covariances, the loss functional is

$$V(q'|\Upsilon, \Psi, \mathbf{Q}_c) = \mathbb{E}_{q'}[\phi^p(\mathbf{x}|\mathbf{Q}_c) + \phi^m(\mathbf{x}|\Upsilon)] + \phi^w(\Upsilon|\Psi) + \frac{1}{2} \ln(|\Sigma^{-1}|). \quad (40)$$

Table III shows the resulting translational errors on our test trajectory. We can see that without the IW prior, the estimation framework fails to reject outliers, resulting in an overall translation error of above 5 m. However, using the IW prior, we see that the error is only 0.2365 m. When we did not have any outliers at all in our data, the error was 0.2335 m (Table I), meaning the average translational error on our test set only increased by 0.003 m.

From this experiment, we can see that using the IW prior allows for the handling of outliers in both training and testing due to our ability to estimate measurement covariances.

TABLE III
EXPERIMENTS B & C - TRANSLATIONAL ERRORS USING A STATIC MEASUREMENT COVARIANCE COMPARED TO USING AN IW PRIOR WHEN WE HAVE OUTLIERS IN OUR TEST SET. IN EXPERIMENT B, WE TRAIN WITH OUTLIERS AND IN EXPERIMENT C, WE TRAIN WITHOUT OUTLIERS

Seq no.	Experiment B		Experiment C	
	Static \mathbf{W} (m)	IW prior (m)	Static \mathbf{W} (m)	IW prior (m)
0	6.1976	0.0773	7.3504	0.0731
1	5.8371	0.0979	6.0754	0.0948
2	5.3652	0.4125	5.5771	0.4096
3	5.1217	0.1860	5.8157	0.1873
4	5.5186	0.2807	5.5503	0.2826
5	5.4780	0.5563	6.3057	0.5554
6	6.3936	0.3004	7.1858	0.2995
7	5.6898	0.2274	6.0332	0.2256
8	6.3717	0.1233	9.3079	0.1224
9	6.8032	0.1036	8.1237	0.1046
overall	5.8776	0.2365	6.7325	0.2355

C. Training Without and Testing With Measurement Outliers

In Experiment B, we included outlier measurements in both the training and test set and saw that the IW prior allows us to achieve comparable errors to the case with no outliers. To see if this still holds even when we do not see any outliers in training, in Experiment C we train without any outliers but test with outliers. As the only difference between Experiment B and Experiment C is that we now train without any outliers, the loss functionals remain the same.

Table III shows that the resulting translational errors are again very high when we simply learn a static measurement covariance, but that we can still achieve reasonably low errors when learning the parameters of our IW prior. By incorporating the IW prior instead of learning a static measurement covariance, we decrease error from above 6 m to 0.2355 m. Compared to the error of 0.2335 m when there are no outliers in our test set (Table I), we see an increase in error of only 0.002 m with the IW prior. This experiment shows that we can indeed still benefit from the outlier rejection scheme that comes with using an IW prior even when there are no outliers in our training set.

Comparing the results from Experiments B and C, we see that in both cases, incorporating the IW prior helps to reject outliers in the test set, regardless of whether there were any outliers in the training set. While errors for the static measurement covariance were similarly poor in both experiments, we note that the concentration of the errors are different as shown in Fig. 3(b). What we see is that when we train without any outliers (Experiment C), the errors are concentrated where the outliers are in the test set. This result is unsurprising given that no outliers were seen in training, which in turn is reflected in our learned noise model parameters. When we train with outliers (Experiment B), the errors still peak around the outliers, but are more spread out over the entire trajectory. Regardless of this difference, we can see that using the IW prior is robust to both cases and still results in low translational errors.

D. Bicocca Dataset

We also evaluate our method on the Bicocca 25b dataset from [31], which provides a set of odometry and loop closure constraints (represented as a pose graph) created from a bag of words place recognition system run on data collected during the

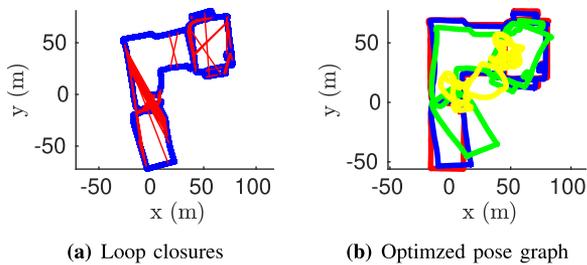


Fig. 4. (a) Odometric trajectory (blue) with loop closures (red). (b) Optimized trajectory using IW prior (blue), static covariance learning (green) and no covariance learning (yellow) overlaid with groundtruth (red).

TABLE IV
AVERAGE TRAJECTORY ERROR AS CALCULATED BY THE RAWSEEDS TOOLKIT

IW prior (m)	Static \mathbf{W} (m)	No covariance learning (m)
2.3292	11.9059	29.4418

Rawseeds project [32]. Fig. 4(a) shows the loop closures in red, including many false ones. We optimize this pose graph using our framework and show the results in Fig. 4(b). We see that our framework with the IW prior (blue) is closest to the groundtruth (red) while learning a static covariance (green) and no covariance learning (yellow) are much more negatively affected by the false loop closures. Table IV also shows the Average Trajectory Error as calculated by the Rawseeds Toolkit for each of the methods.

VI. CONCLUSION

In this letter, we presented parameter learning for ESGVI. We showed that our parameter learning method does not need groundtruth, and is robust to noisy measurements and outliers. This is desirable because in many cases, we do not have a way of obtaining accurate groundtruth of robot trajectories. The implication of our work is that we now have a framework for estimating robot parameters based solely on whatever sensors are available. We experimentally demonstrated our method on a 36 km vehicle dataset.

However, we still assumed two parameters to be known: ν , the DOF parameter for the IW distribution, and β , the determinant constraint on the scale matrix, Ψ . For future work, we will investigate how to also learn ν and eliminate the need to constrain the determinant of Ψ to be a constant, β .

In this work, we chose to learn the noise model parameters as a useful practical application of our framework. However, our future intention with ESGVI is to learn entire robot models that are represented by rich modelling techniques, such as DNNs.

ACKNOWLEDGMENT

The test vehicle used in this letter was donated by General Motors (GM) Canada.

REFERENCES

- [1] T. D. Barfoot, J. R. Forbes, and D. J. Yoon, "Exactly sparse gaussian variational inference with application to derivative-free batch nonlinear state estimation," *Int. J. Robot. Res.*, to be published.
- [2] J. N. Wong, D. J. Yoon, A. P. Schoellig, and T. Barfoot, "A data-driven motion prior for continuous-time trajectory estimation on SE(3)," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1429–1436, Apr. 2020.
- [3] K. J. Åström and P. Eykhoff, "System identification—A survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.
- [4] M. Gevers, "A personal view of the development of system identification: A 30-year journey through an exciting field," *IEEE Control Syst. Mag.*, vol. 26, no. 6, pp. 93–105, Dec. 2006.
- [5] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval, "Past, present and future of nonlinear system identification in structural dynamics," *Mech. Syst. Signal Process.*, vol. 20, no. 3, pp. 505–592, 2006.
- [6] L. Fu and P. Li, "The research survey of system identification method," in *Proc. Int. Conf. Intell. Human-Mach. Syst. Cybern.*, 2013, vol. 2, pp. 397–401.
- [7] J. Kokkala, A. Solin, and S. Särkkä, "Expectation maximization based parameter estimation by sigma-point and particle smoothing," in *Proc. Int. Conf. Inf. Fusion*, 2014, pp. 1–8.
- [8] J. Kokkala, A. Solin, and S. Särkkä, "Sigma-point filtering and smoothing based parameter estimation in nonlinear dynamic systems," *J. Advances Inf. Fusion*, vol. 11, no. 1, pp. 15–30, 2016.
- [9] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *J. Time Ser. Anal.*, vol. 3, no. 4, pp. 253–264, 1982.
- [10] Z. Ghahramani and S. T. Roweis, "Learning nonlinear dynamical systems using an em algorithm," in *Proc. Conf. Neural Inf. Process. Syst.*, 1999, pp. 431–437.
- [11] T. B. Schön, A. Wills, and B. Ninness, "System ident. of nonlinear state-space models," *Automatica*, vol. 47, no. 1, pp. 39–49, 2011.
- [12] M. Gašperin and D. Juričić, "Application of unscented transformation in nonlinear system identification," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 4428–4433, 2011.
- [13] P. Abbeel, A. Coates, M. Montemerlo, A. Y. Ng, and S. Thrun, "Discriminative training of Kalman filters," in *Proc. Robot.: Sci. Syst.*, 2005, vol. 2, pp. 289–296.
- [14] A. Mohamed and K. Schwarz, "Adaptive Kalman filtering for ins/gps," *J. Geodesy*, vol. 73, no. 4, pp. 193–203, 1999.
- [15] C. Hu *et al.*, "Adaptive Kalman filtering for vehicle navigation," *J. Global Positioning Syst.*, vol. 2, no. 1, pp. 42–47, 2003.
- [16] Y. Yang and W. Gao, "An optimal adaptive Kalman filter," *J. Geodesy*, vol. 80, no. 4, pp. 177–183, 2006.
- [17] J. Ko and D. Fox, "Learning GP-bayesfilters via gaussian process latent variable models," *Auton. Robots*, vol. 30, no. 1, pp. 3–23, 2011.
- [18] M. Brossard, A. Barrau, and S. Bonnabel, "AI-IMU dead-reckoning," *IEEE Trans. Intell. Vehicles*, 2020, doi: 10.1109/ITV.2020.2980758.
- [19] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep inference for covariance estimation: Learning Gaussian noise models for state estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1436–1443.
- [20] R. L. Russell, and C. Reale, "Multivariate uncertainty in deep learning," 2019, *arXiv:1910.14215*.
- [21] N. Chebrolu, T. Läbe, O. Vysotska, J. Behley, and C. Stachniss, "Adaptive robust kernels for non-linear least squares problems," 2020, *arXiv:2004.14938*.
- [22] A. G. Wilson and Z. Ghahramani, "Generalised wishart processes," in *Proc. Conf. Uncertainty Artif. Intell.*, Arlington, VA, USA, 2011, pp. 736–744.
- [23] K. Takahashi, J. Fagan, and M.-S. Chen, "A sparse bus impedance matrix and its application to short circuit study," in *Proc. PICA Conf.*, 1973.
- [24] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.
- [25] J. L. W. V. Jensen, "Sur les fonctions convexes et les inégalités entre les valeurs moyennes," *Acta Math.*, vol. 30, pp. 175–193, 1906.
- [26] T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch continuous-time trajectory estimation as exactly sparse gaussian process regression," in *Proc. Robot.: Sci. Syst.*, 2014, doi: 10.15607/RSS.2014.X.001.
- [27] S. Anderson and T. D. Barfoot, "Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE(3)," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2015, pp. 157–164.
- [28] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [29] V. Peretroukhin, W. Vega-Brown, N. Roy, and J. Kelly, "PROBE-GK: Predictive robust estimation using generalized kernels," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 817–824.
- [30] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Trans. Robot.*, vol. 30, no. 3, pp. 679–693, Jun. 2014.
- [31] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time for pose graph SLAM," *Int. J. Robot. Res.*, vol. 32, pp. 1611–1626, 2013.
- [32] S. Ceriani *et al.*, "Rawseeds ground truth collection systems for indoor self-localization and mapping," *Auton. Robots*, vol. 27, pp. 353–371, 2009, doi: 10.1007/s10514-009-9156-5.