Robust Linear-complexity Approach to Full SLAM Problems: Stochastic Variational Bayes Inference

Xiaoyue Jiang[‡*], Hang Yu[†], Michael Hoy[*], Justin Dauwels[*†]
[*] ST Engineering – NTU Corporate Lab, [†]Centre for System Intelligence and Efficiency (EXQUISITUS)
School of Electrical and Electronic Engineering, Nanyang Technological University Singapore

***Abstract*—The simultaneous localization and mapping (SLAM) problem involves using the measurements of sensors to construct an environmental map, while simultaneously recovering the vehicle trajectory within this map. There are broadly two strategies for SLAM: on-line and off-line. In this paper, we focus on the off-line SLAM (a.k.a. full SLAM) problem and propose a variational Bayes inference algorithm to address it. Specifically, the intractable posterior distribution of the vehicle poses given the measurements is approximated by a tractable variational distribution, resulting in estimates of the vehicle poses as well as their uncertainties. In contrast with the existing off-line methods, the inverse variances of the additive noise are updated along with the posterior distribution instead of being fixed, thus, the proposed method is robust to unknown noises. Furthermore, the computational complexity of the proposed method is only *linear* in the number of frames and the computational bottleneck of the algorithm can be easily parallelized to achieve further acceleration. Numerical results show that the proposed method is insensitive to the selection of the noise parameters. More importantly, it is superior in efficiency to the state-of-the-art method, especially for large-scale SLAM problems.**

I. Introduction

To bring the usefulness of autonomous vehicles into full play for practical tasks, such as search, rescue, military, and transportation, an accurate estimate of the environmental map and the associated vehicles' positions is indispensable. The acquisition of maps under pose uncertainty is generally referred to as the SLAM problem.

The research on SLAM is generally divided into two groups: on-line and off-line. Both are of unique signifiance and wide applicability in practical systems. Specifically, on-line SLAM generally aims for real-time tracking [1]. It sequentially updates the current state of the vehicle pose and the map, given the previous state and the current measurements. Such problems can be solved by various filtering methods, including Extended Kalman Filter (EKF) [2], Rao-Blackwellized particle Filter (i.e., FastSLAM; see e.g., [3]), Unscented Kalman Filter (UKF) [4] and Extended Information Filter (EIF) [5]. On the other hand, off-line SLAM is required in local mapping and loop closing tasks to optimize a set of frames and map points [1]. It estimates all unknown vehicle poses (i.e., the entire path of the vehicle) and the global map given all measurements, and thus, it is typically formulated as a smoothing problem. Nonlinear least squares (NLLS) minimization is often used in the literature [6]–[8] to address this problem, by seeking the MAP (maximum a posterior) estimates. Of particular note is the popular graph-based SLAM method proposed in [8]. It introduces a graph to describe the SLAM problem, in which the nodes represent the vehicle poses and the map and the edges represent the measurements. The off-line SLAM problem is then interpreted as finding the best configuration of the nodes that satisfy the constraints imposed by the edges. However, the computational complexity of graph-based SLAM is $\mathcal{O}(K^3M^3+K^2MN)$, where K is the number of frames, M is the dimension of the vehicle pose, and N is the dimension of observation vector for each frame. Consequently, for practical use, graph-based SLAM usually restricts its attention to only a subset of the vehicle path (a.k.a., key-frames) or a small part of the whole graph [9], [10]. Unfortunately, such remedies might jeopardize the accuracy of the estimation [11]. Moreover, graph-based SLAM only provides the point estimates of the vehicle path and the map without their uncertainties.

In this paper, we propose a stochastic variational Bayes inference (SVBI) method to solve off-line SLAM. The intractable posterior of all unknown vehicle poses across time is approximated with a Gaussian variational distribution by minimizing their Kullback-Leibler (KL) divergence. In addition, we simultaneously estimate the global map and the hyperparameters, such as the inverse variance of the noise. The KL divergence is minimized using stochastic gradient descent. The computational complexity of the proposed method is only linear, i.e., $\mathcal{O}(KMN)$. The computational bottleneck of this method lies in the calculation of the stochastic gradients, but these calculations can be implemented in parallel. As a result, the proposed SVBI algorithm is tuning free of hyperparameters, while at the same time possessing superior computational efficiency over graph-based SLAM, and being amenable to parallelization.

To the best of our knowledge, we are among the first to propose variational inference algorithms for general off-line SLAM problems. We notice that the classical variational Bayes expectation maximization (VBEM) algorithm is applied in [13], [14]. However, VBEM requires that all priors must be conjugate to the likelihood. To satisfy this requirement, only conditionally-conjugate distributions are used to build the Bayesian model, thus limiting the practical modeling capabilities. As oppose to [13], [14], the proposed method can deal with non-conjugacy in Bayesian models. On the other hand, a so-called variational Bayes filter is proposed in [15] to tackle the SLAM problems. The observation and the motion function are approximated by neural networks.

[‡]Corresponding author: XJIANG007@e.ntu.edu.sg.
The research was partially supported by the ST Engineering – NTU Corporate Lab through the NRF corporate lab@university scheme. Hang Yu is supported by MOE (Singapore) Tier 2 project MOE2017-T2-2-126.

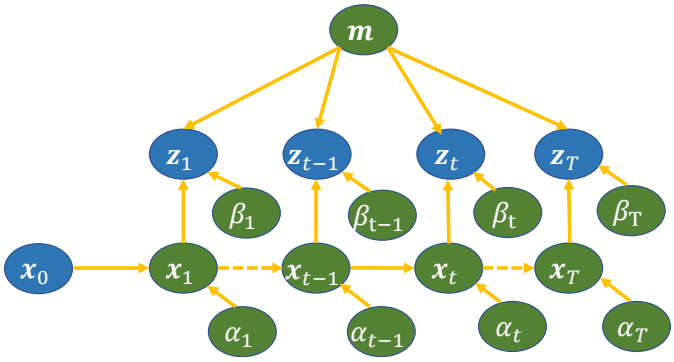


Fig. 1: The graphical model of the off-line SLAM problem.

The posterior distribution of the map is approximated by a Gaussian distribution using the VB framework, whereas the posterior distribution of the vehicle path is estimated using particle filtering. Consequently, training data is required to learn the parameters of the neural networks, which is impractical for real data. Moreover, particle filtering can be quite computationally burdensome.

To assess the performance of the proposed method, we simulate data from the commonly used visual SLAM system and then compare our SVBI method with the state-of-the-art graph-based SLAM algorithm under different settings. The proposed SVBI is shown be to robust to the choice of the hyperparameters in comparison with graph-based SLAM. Furthermore, it realizes significant efficiency gain for SLAM problems with a large number of frames.

This paper is organized as follows: we introduce a probabilistic formulation of off-line SLAM in Section II. We then derive the proposed SVBI algorithm in Section III. Numerical results are presented in Sections IV. Finally, we offer concluding remarks in Section V.

II. Probabilistic Formulation of Off-line SLAM

The general SLAM problem can be descried by a graphical model as shown in Fig. 1. Here, we use the blue and green nodes to represent observed and latent variables respectively. More specifically, $\boldsymbol{m}$ is the global environmental map, $\boldsymbol{x}_{1:K} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K\}$ represent the vehicle's path from time 1 to K starting from $\boldsymbol{x}_0$, and $\boldsymbol{z}_{1:K}$ are the corresponding measurements. In addition, $\alpha_{1:K}$ and $\beta_{1:K}$ represent the inverse variance of the motion, observation and map noise respectively. The Bayesian model corresponding to the graph can be factorized as:

$$\begin{aligned} &p(\boldsymbol{x}_{1:K}, \boldsymbol{z}_{1:K} | \boldsymbol{x}_0, \boldsymbol{m}, \alpha_{1:K}, \beta_{1:K}) \\ &\propto p(\boldsymbol{x}_0) \prod_{k=1}^{K} p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}, \alpha_k) p(\boldsymbol{z}_k | \boldsymbol{m}, \boldsymbol{x}_k, \beta_k). \end{aligned} \tag{1}$$

In the above expression, the pose evolution distribution $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \alpha_k)$ is typically referred to as the motion model, the measurement emission process $p(\boldsymbol{z}_k|\boldsymbol{x}_k, \beta_k)$ is often described as the observation model. Equivalently, these two models can be respectively expressed by the following two functions:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{\epsilon}_k, \tag{2a}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{m}) + \boldsymbol{\delta}_k, \tag{2b}$$

where $\boldsymbol{\epsilon}_k$ and $\boldsymbol{\delta}_k$ denote the additive motion and observation noise perturbations respectively, and f and h are motion and nonlinear observation functions respectively.

As mentioned in Section I, here we consider a 3D feature-based visual SLAM problem. We use "W" to denote the fixed world frame and "R" the camera frame. Consequently, $\boldsymbol{x}_k$ is a vector with $M = 6$ dimensions that comprises the 3D metric position $\boldsymbol{r}_k^{\mathrm{W}}$ and the Euler angles $\boldsymbol{e}_k^{\mathrm{WR}}$ (i.e., roll, pitch, yaw), where the superscripts denote that the quantities are expressed relative to each frame. In particular, we convert $\boldsymbol{e}_k^{\mathrm{WR}}$ to a 3×3 rotation matrix R_k^{WR} in the calculations when required. Suppose that the observed key points are all monocular, and thus, $\boldsymbol{z}_k := [u_i, v_i]$ is a set of 2D coordinate positions in the images. The global map $\boldsymbol{m}$ is composed of a set of 3D points $[\boldsymbol{l}_i^{\mathrm{W}}]$ in the real world coordinates. By assuming that the time interval between consecutive two frames is a constant $\triangle t$, f and h in (2) can be written as:

$$f(\boldsymbol{x}_{k-1}) = \begin{pmatrix} \boldsymbol{r}_k^{\mathrm{W}} \\ \boldsymbol{e}_k^{\mathrm{WR}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}_{k-1}^{\mathrm{W}} + \boldsymbol{v}_k^{\mathrm{W}} \triangle t \\ R2e\left(R_{k-1}^{\mathrm{W}} e2R\left(\boldsymbol{\omega}_k^{R} \triangle t\right)\right) \end{pmatrix}, \tag{3a}$$

$$[h(\boldsymbol{x}_k, \boldsymbol{m})]_i = h(\boldsymbol{x}_k, \boldsymbol{l}_i^{\mathrm{W}}) = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} u_0 + f_c \frac{X_{ik}^{\mathrm{R}}}{Z_{ik}^{\mathrm{R}}} \\ v_0 + f_c \frac{Y_{ik}^{\mathrm{R}}}{Z_{ik}^{\mathrm{R}}} \end{bmatrix}, \tag{3b}$$

where $\boldsymbol{v}^{\mathrm{W}}$ is the velocity vector, $\boldsymbol{w}^{\mathrm{R}}$ is the angular velocity, $R2e$ and $e2R$ are functions converting the Rotation matrix to and from the Euler angles respectively, f_c ,u_0, and v_0 are camera calibration parameters, and $[X_{ik}^{\mathrm{R}}, Y_{ik}^{\mathrm{R}}, Z_{ik}^{\mathrm{R}}]$ denote the relative position of $\boldsymbol{l}_i^{\mathrm{W}}$ in the k^{th} frame, which can be calculated as:

$$[X_i^{\mathrm{R}}, Y_i^{\mathrm{R}}, Z_i^{\mathrm{R}}]^T = R_k^{\mathrm{RW}}(\boldsymbol{l}_i^{\mathrm{W}} - \boldsymbol{r}_k^{\mathrm{W}}). \tag{4}$$

We further assume that the noise terms $\boldsymbol{\epsilon}_k$ and $\boldsymbol{\delta}_k$ in (2) follow Gaussian distributions $\mathcal{N}(0; \alpha_k^{-1} I_M)$ and $\mathcal{N}(0; \beta_k^{-1} I_N)$ respectively, where I_M denotes a $M \times M$ identity matrix, and α_k, β_k denote the inverse variance of the noise. Hence, the distributions of the motion and observation models in (1) can be expressed as:

$$\begin{aligned} &p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \alpha_k) \propto \\ &\sqrt{\alpha_k^M} \exp\left\{ -\frac{\alpha_k}{2} \left(\boldsymbol{x}_k - f(\boldsymbol{x}_{k-1})\right)^{\mathrm{T}} \left(\boldsymbol{x}_k - f(\boldsymbol{x}_{k-1})\right) \right\}, \end{aligned} \tag{5a}$$

$$\begin{aligned} &p(\boldsymbol{z}_k|\boldsymbol{m}, \boldsymbol{x}_k, \beta_k) \propto \\ &\sqrt{\beta_k^M} \exp\left\{ -\frac{\beta_k}{2} (\boldsymbol{z}_k - h\left(\boldsymbol{x}_k, \boldsymbol{m}\right))^{\mathrm{T}} (\boldsymbol{z}_k - h\left(\boldsymbol{x}_k, \boldsymbol{m}\right)) \right\}. \end{aligned} \tag{5b}$$

Note that in previous works [1], the noise hyperparameters $\alpha_{1:K}$ and $\beta_{1:K}$ are typically selected either by experience [2] or estimated through bootstrapping or heuristic approaches [1], [2] beforehand. In other words, they are treated as known parameters in the models. However, the predefined values of these parameters can be unreliable in practice, especially for the case where the noise level changes over time. To overcome this issue, we update $\alpha_{1:K}$,$\beta_{1:K}$ along with the posterior distribution, which will be discussed in the next section. The resulting approach is thus robust to the given values of the hyperparameters.

III. Stochastic Variational Bayes Inference for Off-line SLAM

Following the empirical Bayes approach, we aim to obtain the point estimates of the global map m the hyperparameters $\alpha_{1:K}$ and $\beta_{1:K}$ by maximizing their likelihood $\log p(\boldsymbol{z}_{1:K}, \boldsymbol{x}_0|\boldsymbol{m}, \alpha_{1:K}, \beta_{1:K})$ and then to compute the posterior distribution of the vehicle path $p(\boldsymbol{x}_{1:K}|\boldsymbol{z}_{1:K}, \boldsymbol{x}_0, \boldsymbol{m}, \alpha_{1:K}, \beta_{1:K})$ given the estimates of the map and the hyperparameters. However, the analytical expressions of both distributions are unavailable. Instead, we employ the variational Bayes framework and maximize the evidence lower bound (ELBO) $\mathcal{L}$ of $\log p(\boldsymbol{z}_{1:K}, \boldsymbol{x}_0|\boldsymbol{m}, \alpha_{1:K}, \beta_{1:K})$:

$$\begin{aligned}
&\log p(\boldsymbol{z}_{1:K}, \boldsymbol{x}_0|\boldsymbol{m}, \alpha_{1:K}, \beta_{1:K}) \\
&\geq \int q(\boldsymbol{x}_{1:K}) \log \frac{p(\boldsymbol{z}_{1:K}, \boldsymbol{x}_0, \boldsymbol{x}_{1:K}|\boldsymbol{m}, \alpha_{1:K}, \beta_{1:K})}{q(\boldsymbol{x}_{1:K})} d\boldsymbol{x}_{1:K}, \\
&= \mathcal{L}. \qquad (6)
\end{aligned}$$

The above inequality follows directly from the Jensen's inequality. In other words, we aim to find the variational distribution $q(\boldsymbol{x}_{1:K})$ and the point estimates of the map $\boldsymbol{m}$ and the hyperparameters $\alpha_{1:T}$ and $\beta_{1:T}$ that maximize $\mathcal{L}$.

As a first step, we choose the variational distribution for $\boldsymbol{x}_{1:K}$. We first factorize $q(\boldsymbol{x}_{1:K}) = \prod_{i=1}^{M} q(\boldsymbol{x}^i_{1:K})$, where M denotes the dimension of $\boldsymbol{x}_k$ and x^i_k denotes entry i in $\boldsymbol{x}_k$, and then set $q(\boldsymbol{x}^i_{1:K}) = \mathcal{N}(\boldsymbol{\mu}^i, {J^i}^{-1})$, where $\boldsymbol{\mu}^i$ is the mean and J^i is the precision matrix (i.e., inverse covariance). Note that for Gaussian distributions the zero pattern of J^i encodes the conditional dependence between $\boldsymbol{x}^i_{1:k}$: $x^i_{k_1}$ and $x^i_{k_2}$ are conditionally dependent given the remaining variables if and only if entry (k_1, k_2) in J^i is nonzero. On the other hand, we can tell from Fig. 1 that x^i_k only have conditional dependence on x^i_{k-1} and x^i_{k+1}. To describe such conditional dependence, we specify the precision matrix J^i to be a tridiagonal matrix for all i. To facilitate the variational inference, we further parameterize $q(\boldsymbol{x}^i_{1:K}) = \mathcal{N}(\boldsymbol{\mu}^i, {U^i}^{-1}{U^i}^{-T})$ such that

$$\boldsymbol{x}^i_{1:K} = {U^i}^{-1}\boldsymbol{e} + \boldsymbol{\mu}^i_{1:K}, \qquad (7)$$

where $\boldsymbol{e}$ follows the standard normal distribution $\phi(\boldsymbol{e}) = \mathcal{N}(\boldsymbol{0}, I_K)$, and U^i is the Cholesky decomposition of J^i, that is, $J^i = {U^i}^T U^i$. As a result, U^i is an upper triangular bidiagonal matrix, and its two diagonals are denotesd as D^i_0 and D^i_1. Note that D^i_0 is the main diagonal of U^i and so all entries inside are positive.

As a result, (6) can be rewritten as:

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_q[\log p(\boldsymbol{z}_{1:K}, \boldsymbol{x}_0, \boldsymbol{x}_{1:K}|\boldsymbol{m}, \alpha_{1:K}, \beta_{1:K})] + \mathcal{H}_q \\
&= \mathbb{E}_{\phi(\boldsymbol{e})}[\sum_{k=1}^{K} \log p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \alpha_k) p(\boldsymbol{z}_k|\boldsymbol{x}_k, \beta_k, \boldsymbol{m})] \\
&\quad - \sum_{i=1}^{M} \log|U^i| + c_0, \qquad (8)
\end{aligned}$$

where $\mathcal{H}_q = -\int q(\boldsymbol{x}_{1:K}) \log q(\boldsymbol{x}_{1:K}) d\boldsymbol{x}_{1:K}$ is the entropy of $q(\boldsymbol{x}_{1:K})$, $\log|U^i| = \sum_{k=1}^{K} \log U^i_{kk}$, and c_0 summarizes all irrelevant constants.

Algorithm 1: Stochastic variational Bayes inference for off-line SLAM problems.

Data: $\boldsymbol{z}_{1:K}, \boldsymbol{x}_0$
Result: $\boldsymbol{x}_{1:K}, \boldsymbol{m}$
Initialize $\boldsymbol{\mu}^{(0)}_{1:K}, U^{(0)}, \alpha^{(0)}_{1:K}, \beta^{(0)}_{1:K}$, $t = 0$;
while *convergence criterion is not met* **do**
  $t = t + 1$;
  $\boldsymbol{e} \sim \phi(\boldsymbol{e})$;
  $\boldsymbol{x}^{(t-1)}_{1:K} = (U^{-1})^{(t-1)}\boldsymbol{e} + \boldsymbol{u}^{(t-1)}_{1:K}$;
  For a variable $\nu \sim \{\boldsymbol{\mu}_{1:K}, U, \alpha_{1:T}, \beta_{1:K}\}$,
  Calculate the gradient $\nabla_\nu \mathcal{L}^{(t)}$ through (10) with $\boldsymbol{x}^{(t-1)}_{1:K}$;
  $\nu^{(t)} = \nu^{(t-1)} + \rho_t \nabla_\nu \mathcal{L}^{(t)}$;
end

In order to maximize the ELBO $\mathcal{L}$ in (8), we resort to the gradient ascent method. For notational simplicity, we define $\Phi_k = \boldsymbol{x}_k - f(\boldsymbol{x}_{k-1})$ and $\Psi_k = \boldsymbol{z}_k - h(\boldsymbol{x}_k, \boldsymbol{m})$. The gradient of $\mathcal{L}$ w.r.t. μ^i, D^i_0, D^i_1, m, α_k, and β_k can then be expressed as:

$$\nabla_{\boldsymbol{\mu}^i_k} \mathcal{L} = \mathbb{E}_{\phi(\boldsymbol{e})}[\Delta^i_k], \qquad (9a)$$

$$\nabla_{D^i_{0k}} \mathcal{L} = -\mathbb{E}_{\phi(\boldsymbol{e})}\Big[\Big({U^i}^{-T}\Delta^i_{1:K}\Big)_k \Big({U^i}^{-1}\boldsymbol{e}\Big)_k\Big] - \frac{1}{D^i_{0k}}, \qquad (9b)$$

$$\nabla_{D^i_{1k}} \mathcal{L} = -\mathbb{E}_{\phi(\boldsymbol{e})}\Big[\Big({U^i}^{-T}\Delta^i_{1:K}\Big)_k \Big({U^i}^{-1}\boldsymbol{e}\Big)_{k+1}\Big], \qquad (9c)$$

$$\nabla_{\boldsymbol{m}} \mathcal{L} = \sum_{k=1}^{K} \beta_k \mathbb{E}_{\phi(\boldsymbol{e})}\left[\left(\nabla_{\boldsymbol{m}} h(\boldsymbol{x}_k, \boldsymbol{m})\right)^{\mathrm{T}} \Psi_k\right], \qquad (9d)$$

$$\nabla_{\alpha_k} \mathcal{L} = \frac{M}{2\alpha_k} - \frac{1}{2}\mathbb{E}_{\phi(\boldsymbol{e})}\left[\Phi^T_k \Phi_k\right], \qquad (9e)$$

$$\nabla_{\beta_k} \mathcal{L} = \frac{N}{2\beta_k} - \frac{1}{2}\mathbb{E}_{\phi(\boldsymbol{e})}\left[\Psi^T_k \Psi_k\right], \qquad (9f)$$

$$\begin{aligned}
\Delta_k = &-\alpha_k \Phi_k + \alpha_{k+1} \nabla_{\boldsymbol{x}_k} f(\boldsymbol{x}_k) \circ \Phi_{k+1} \\
&+ \beta_k \left(\nabla_{\boldsymbol{x}_k} h(\boldsymbol{x}_k, \boldsymbol{m})\right)^{\mathrm{T}} \Psi_k, \qquad (9g)
\end{aligned}$$

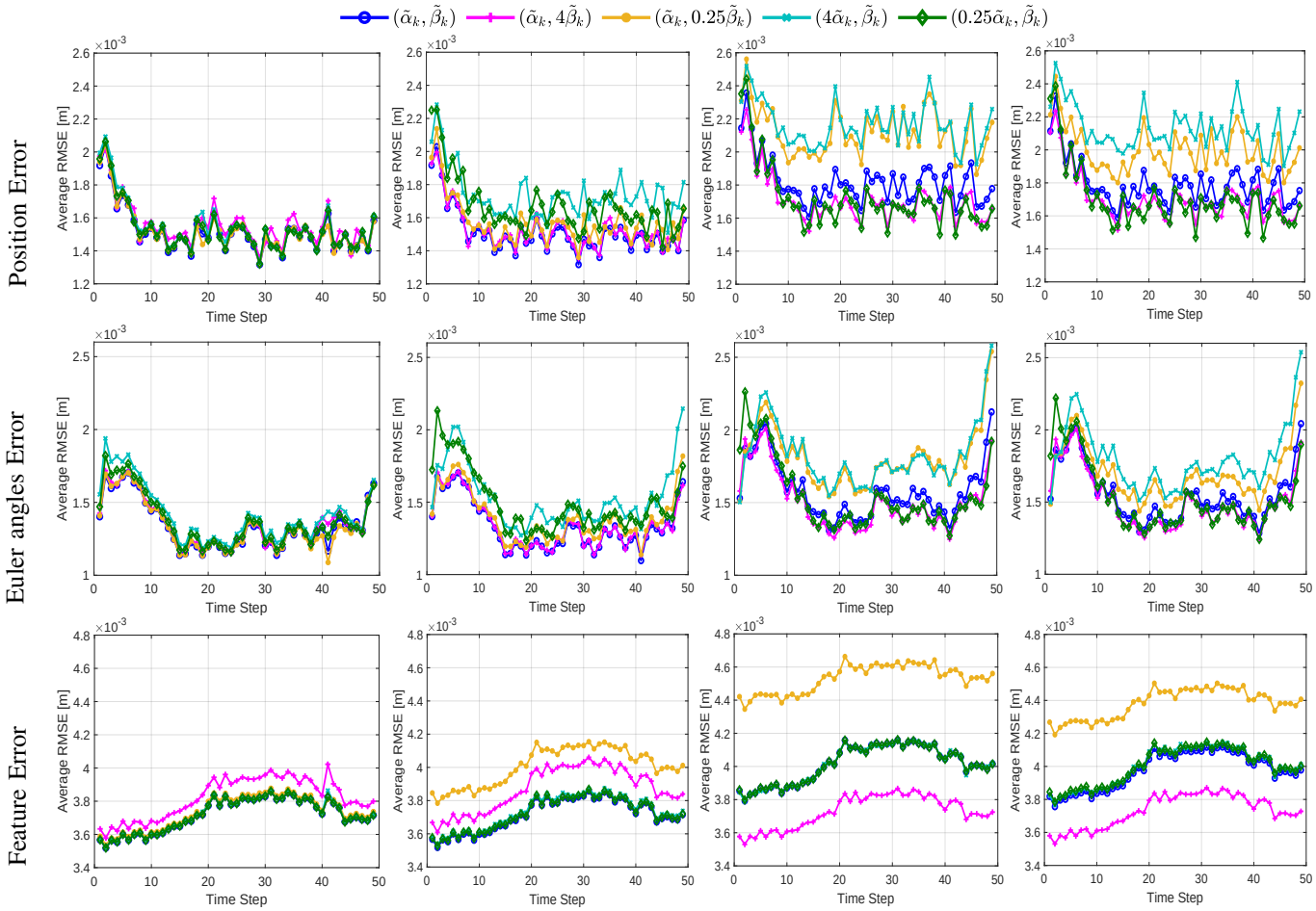
where all vectors in the above expressions are assumed to be column vectors, $[\boldsymbol{a}]_k$ denotes entry k in the vector $\boldsymbol{a}$, and $\circ$ denotes the elementwise product.

Unfortunately, the expectations in (9) cannot be computed analytically. To counteract this problem, we exploit stochastic gradient instead of the above exact gradient to maximize $\mathcal{L}$. Stochastic gradients are unbiased estimates of the exact gradient that can be easily evaluated. In our case, we approximate the expectations over $\phi(\boldsymbol{e})$ in an unbiased manner by drawing one sample from $\phi(\boldsymbol{e})$ and evaluating the value of the functions inside the expectation based on this sample. Therefore, the unbiased stochastic approximation of the exact gradients (9) can be written as:

$$\nabla_{\boldsymbol{\mu}^i_k} \mathcal{L} \approx \Delta_k, \qquad (10a)$$

$$\nabla_{D^i_{0k}} \mathcal{L} \approx -\Big[{U^i}^{-T}\Delta^i_{1:K}\Big]_k \Big[{U^i}^{-1}\boldsymbol{e}\Big]_k - \frac{1}{D^i_{0k}}, \qquad (10b)$$

$$\nabla_{D^i_{1k}} \mathcal{L} \approx -\Big[{U^i}^{-T}\Delta^i_{1:K}\Big]_k \Big[{U^i}^{-1}\boldsymbol{e}\Big]_{k+1}, \qquad (10c)$$



(a) SVBI. (b) GO. (c) GO_AIC. (d) GO_BIC.

Fig. 2: The average RMSE of (a) SVBI, (b) graph-based optimization with given hyperparameters (GO), (c) graph-based optimization with AIC (GO_AIC), and (d) graph-based optimization with BIC (GO_BIC).

$$\nabla_{\boldsymbol{m}} \mathcal{L} \approx \sum_{k=1}^{K} \beta_k \left(\nabla_{\boldsymbol{m}} h(\boldsymbol{x}_k, \boldsymbol{m})\right)^{\mathrm{T}} \Psi_k, \tag{10d}$$

$$\nabla_{\alpha_k} \mathcal{L} \approx \frac{M}{2\alpha_k} - \frac{1}{2}\Phi_k^T \Phi_k, \tag{10e}$$

$$\nabla_{\beta_k} \mathcal{L} \approx \frac{N}{2\beta_k} - \frac{1}{2}\Psi_k^T \Psi_k. \tag{10f}$$

Note that in the above expressions $U^{i^{-1}}\boldsymbol{a}$ can be computed with linear complexity since U^i is a bidiagonal matrix, where $\boldsymbol{a}$ is an arbitrary K-dimensional column vector. Given the stochastic gradients, we can optimize (8) iteratively as demonstrated in Algorithm 1, where ρ_t is the step size in the t^{th} iteration. According to the Robbins-Monro theorem [17], the stochastic gradient ascent algorithm in Algorithm 1 is guaranteed to converge to a local maximum of $\mathcal{L}$ if the step sizes satisfy $\sum \rho_t = \infty$ and $\sum \rho_t^2 < \infty$. In practice, we employ ADAM [18] to determine the step in every iteration. The computational complexity of the proposed algorithm is only $\mathcal{O}(KMN)$. The computational bottleneck lies in the calculations of the stochastic gradients (10), which can be easily parallelized.

IV. Experiments

In this section, we benchmark the proposed SVBI in Algorithm 1 against graph-based optimization [8], [9] on synthetic data simulated from the commonly-used visual SLAM system model (2). In the system model, we assume that the camera is moving in a 3D space and its calibration parameters follow the settings in [2]. The translation and angular velocity on each axis are set to be $0.05\ m$ and $0.02\ rad$ per frame. In addition, all additive noises follow Gaussian distributions with inverse variances $\alpha_k^t = 1/(0.005\ m)^2, \alpha_k^\theta = 1/(0.002\ rad)^2, \beta_k = 1$ for $k = 1, \cdots, K$.

We first investigate whether the proposed SVBI algorithm is robust to the given values of the hyperparameters α_k and β_k. As mentioned in Section I, the prior statistics of these hyperparameters are often unknown or inaccurate in real-world applications. Concretely, we test the cases where the given value of (α_k, β_k) is set as $(\tilde{\alpha}_k, \tilde{\beta}_k), (\tilde{\alpha}_k, 4\tilde{\beta}_k), (\tilde{\alpha}_k, 0.25\tilde{\beta}_k), (4\tilde{\alpha}_k, \tilde{\beta}_k), (0.25\tilde{\alpha}_k, \tilde{\beta}_k)$ respectively, where $(\tilde{\alpha}_k, \tilde{\beta}_k)$ denotes the true values of (α_k, β_k). For each pair of the given (α_k, β_k), we test both SVBI and graph-based optimization [8], compute the RMSE (root mean squre error) between the estimated and true values of the vechicle position, Euler angles, and the map, and further depict the RMSE as a function of the number of frames K in Fig. 2a and 2b respectively. We consider K up to 50. All results are averaged over 100 trials. For SVBI, we regard the given values of (α_k, β_k) as the initializations of these parameters in

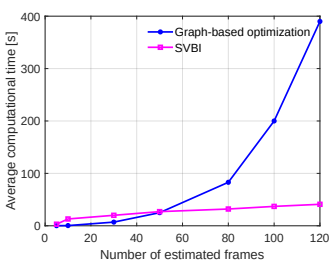


Fig. 3: The average computational time of graph-based optimization and SVBI w.r.t the number of estimated frames.

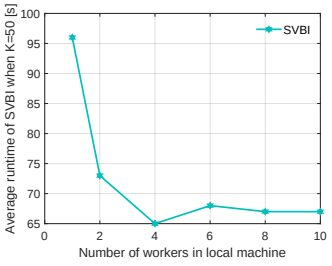


Fig. 4: The average computational time of SVBI w.r.t the number of cores in the server when $K = 50$

Algorithm 1 and further update their values using stochastic gradient ascent. On the other hand, since graph-based SLAM cannot estimate these parameters, we simply keep the given values fixed as the algorithm proceeds. We can tell from Figs. 2a and 2b that SVBI achieves comparable or higher accuracy than graph-based SLAM. Moreover, it can be observed from Fig. 2a that for SVBI the curves resulting from different values of α_k and β_k overlap with each other. As expected, SVBI is robust to the given values of the hyperparameters, as these parameters can be further updated in the algorithm. By contrast, as shown in Fig. 2b, the results of graph-based SLAM is sensitive the given valuse of (α_k, β_k), especially the estimates of the global map.

We further consider estimating (α_k, β_k) using Akaike information criterion (AIC) [19] and Bayesian information criterion (BIC) [20] in graph-based SLAM, which are classical methods for selecting the noise parameters (α_k, β_k). Specifically, let us denote the given values as $(\hat{\alpha}_k, \hat{\beta}_k)$, the candidate set of α_k and β_k is specified as $\mathcal{A} = \{0.25\hat{\alpha}_k, 0.5\hat{\alpha}_k, \hat{\alpha}_k, 2\hat{\alpha}_k, 4\hat{\alpha}_k\}$, and $\mathcal{B} = \{0.25\hat{\beta}_k, 0.5\hat{\beta}_k, \hat{\beta}_k, 2\hat{\beta}_k, 4\hat{\beta}_k\}$ respectively. We then test every combination of (α_k, β_k) selected from $\mathcal{A}$ and $\mathcal{B}$, and choose the pair with the smallest AIC or BIC score. We emphasize that the candidate sets are different for different given values of (α_k, β_k). The results are summarized in Figs. 2c and 2d. Although AIC and BIC may help graph-based SLAM to obtain comparable results to SVBI sometimes, the performance of AIC and BIC is sensitive to the candidate sets of α and β. In practice, however, it is impossible to traverse through all possible values of the hyperparameters. Moreover, AIC and BIC increases the computational cost of graph-based SLAM, as the algorithm has to be run for every combination of α_k and β_k. Indeed, the computational time of SVBI, graph-based SLAM with given values of the hyperparameters, AIC, and BIC is respectively 0.476s, 0.482s, 10.988s, and 11.000s.

Next, we present the computational time of SVBI and graph-based SLAM as a function of the number of frames K in Fig. 3. The computational time of SVBI is approximately a linear function of K, whereas that of graph-based SLAM is a cubic function. The results are consistent with the theoretical computational complexity. This implies that SVBI is more applicable to SLAM problems with a large number of frames. Additionally, we implement the SVBI algorithm in parallel for one example when $K = 50$. Fig. 4 shows how the computational time changes with the number of cores. The running time can be further decreased for the proposed SVBI algorithm when parallel computing is utilized. Such speedup is expected to be more significant when the scale of the problem becomes larger.

V. Conclusions

We propose a stochastic variational Bayes inference for offline SLAM problems. The proposed algorithm approximates the posterior distribution of the vehicle path with a variational distribution while updating the noise parameters. As a result, the proposed SVBI approach is robust to the choice of the noise parameters. Furthermore, the computational complexity of the proposed method is only linear, thus, SVBI is applicable to large-scale SLAM problems. The advantages of the proposed method is validated on synthetic data. In the future work, we will apply SVBI to real-life datasets.

References

[1] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: an Open-Source SLAM system for monocular, stereo and RGB-D cameras", *IEEE Transactions on Robotics*, 33(5): 1255-1262, 2017.

[2] C. Ye, and Y. Zhao. "An EKF-SLAM method with filter consistency test for mobile robots using a 3D camera", *In Proceedings of 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015.

[3] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm", in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006.

[4] H. Wang, et al., "An adaptive UKF based SLAM method for unmanned underwater vehicle", *Mathematical Problems in Engineering*, 2013.

[5] S. Thrun, et al., "Simultaneous localization and mapping with sparse extended information filters", *The International Journal of Robotics Research*, 23.7-8(2004): 693-716, 2004.

[6] F. Dellaert, and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing", *The International Journal of Robotics Research*, 25.12: 1181-1203, 2004.

[7] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates", *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, IEEE, 2006.

[8] G. Grisetti, et al., "A tutorial on graph-based SLAM", *IEEE Intelligent Transportation Systems Magazine*, 2.4: 31-43, 2010.

[9] S. Leutenegger, et al. "Keyframe-based visualinertial odometry using nonlinear optimization", *The International Journal of Robotics Research*, 34.3: 314-334, 2015.

[10] H. Strasdat, et al. "Double window optimisation for constant time visual SLAM", *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011.

[11] H. Strasdat, J.M.M. Montiel, and Andrew J. Davison. "Visual SLAM: why filter?", *Image and Vision Computing*, 30.2: 65-77, 2012.

[12] X. Jiang, et al. "Linear-complexity stochastic variational Bayes inference for SLAM", *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, IEEE, 2017.

[13] M. Fatemi, et al., "Variational Bayesian EM for SLAM", *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop on*, IEEE, 2015.

[14] M. Lundgren, L. Svensson, and L. Hammarstrand. "Variational Bayesian Expectation Maximization for Radar Map Estimation", *IEEE Trans. Signal Processing*, 64.6: 1391-1404, 2016.

[15] A. Mirchev, et al., "Approximate Bayesian inference in spatial environments", *arXiv preprint arXiv:1805.07206*, 2018.

[16] M.K. Titsias, and M. Lzaro-Gredilla, "Doubly stochastic variational Bayes for non-conjugate inference", *ICML*, 2014.

[17] H. Robbins, and S. Monro, "A stochastic approximation method", *Herbert Robbins Selected Papers*, Springer, New York, NY, 1985. 102-109.

[18] D. Kingma, and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[19] T.W. Arnold, "Uninformative parameters and model selection using Akaike's Information Criterion", *The Journal of Wildlife Management*, 74.6: 1175-1178, 2010.

[20] H. S.Bhat, and N. Kumar. "On the derivation of the Bayesian Information Criterion", *School of Natural Sciences*, University of California, 2010.