

# A Data-Driven Motion Prior for Continuous-Time Trajectory Estimation on $SE(3)$

Jeremy N. Wong , David J. Yoon , Angela P. Schoellig , and Timothy D. Barfoot 

**Abstract**—Simultaneous trajectory estimation and mapping (STEAM) is a method for continuous-time trajectory estimation in which the trajectory is represented as a Gaussian Process (GP). Previous formulations of STEAM used a GP prior that assumed either white-noise-on-acceleration (WNOA) or white-noise-on-jerk (WNOJ). However, previous work did not provide a principled way to choose the continuous-time motion prior or its parameters on a real robotic system. This letter derives a novel data-driven motion prior where ground truth trajectories of a moving robot are used to train a motion prior that better represents the robot’s motion. In this approach, we use a prior where latent accelerations are represented as a GP with a Matérn covariance function and draw a connection to the Singer acceleration model. We then formulate a variation of STEAM using this new prior. We train the WNOA, WNOJ, and our new latent-force prior and evaluate their performance in the context of both lidar localization and lidar odometry of a car driving along a 20 km route, where we show improved state estimates compared to the two previous formulations.

**Index Terms**—SLAM, localization.

## I. INTRODUCTION

IT IS common for state estimation in robotics to be carried out in discrete time, which is an approximation of the robot’s continuous-time trajectory. This approximation is sufficient in many cases, but there are situations when it is inadequate. Examples include using continuous scanning-while-moving sensors (e.g., rolling-shutter camera or scanning laser rangefinder), or high-rate asynchronous sensors. In both these cases, the naive discrete approach of including a state at every measurement time can be expensive. Many people have attempted to address this problem by using ad-hoc assumptions about the smoothness of the trajectory in order to use a smaller number of discrete states and infer motion in between. However, smoothness can be built into the estimation exactly using a continuous-time motion prior, thus gaining the ability to incorporate measurements at any time along the trajectory without introducing additional states at those measurement times. Furthermore, continuous-time techniques benefit from the advantage that posterior estimates can be

Manuscript received September 10, 2019; accepted January 4, 2020. Date of publication January 24, 2020; date of current version February 3, 2020. This letter was recommended for publication by Associate Editor U. Frese and Editor S. Behnke upon evaluation of the reviewers’ comments. This work was supported in part by the 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 M3H 5T6, Canada (e-mail: jeremy.wong@robotics.utoronto.ca; david.yoon@robotics.utoronto.ca; angela.schoellig@utoronto.ca; tim.barfoot@utoronto.ca).

Digital Object Identifier 10.1109/LRA.2020.2969153

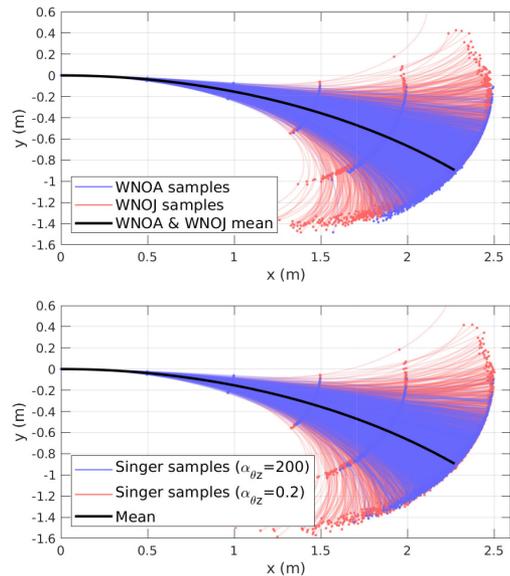


Fig. 1. While the white-noise-on-jerk prior can be formulated such that its mean matches that of any white-noise-on-acceleration prior, their covariances can never be the same (top). The Singer prior is a richer prior that can represent but is not limited to both types of trajectories (bottom).

queried at any time along the trajectory, not just at measurement times.

Current formulations of the continuous-time estimation framework employ either a white-noise-on-acceleration (WNOA) [1]–[3] or white-noise-on-jerk (WNOJ) [4], [5] motion prior, which assumes the prior mean is constant velocity or constant acceleration, respectively. The WNOJ prior can be formulated so that its mean matches the mean of any WNOA prior. However, the two motion priors will always have different covariances as seen in Fig. 1 (top). Thus the explicit choice of either one of these priors is limiting and we argue that it is much more principled to learn the parameters of a richer motion prior based on data.

With this in mind, we derive a motion prior that represents latent accelerations as a Gaussian Process (GP) with a Matérn covariance function and learn the hyperparameters of this GP from noisy ground truth data. Our derivation starts by transforming the GP into a stochastic differential equation (SDE), which we show to be equivalent to the Singer acceleration model [6]. This motion prior can represent but is not limited to both WNOA and WNOJ trajectories as seen in Fig. 1 (bottom).

We then show that on a real world lidar dataset with over 20 km of driving, our motion prior outperforms the WNOA and WNOJ

priors, which have also been properly trained using ground truth data.

The contribution of this letter is two-fold. The first is a principled method for hyperparameter training of continuous-time motion priors in  $SE(3)$ . This opens up the possibility of using far richer motion priors with more parameters. This leads to the second part of the paper, the derivation of a new richer data-driven motion prior.

In Section II we review previous work. An overview of the existing motion priors used for continuous-time estimation is provided in Section III. Section IV presents the derivation of a latent-force prior that we show to be equivalent to the Singer acceleration model. Section V outlines the method that we take to train motion prior hyperparameters given only noisy ground truth measurements of the states. An experimental comparison of the WNOA, WNOJ, and the latent-force prior is presented in Section VI. In Section VII we give concluding remarks and discuss future work.

## II. RELATED WORK

Most estimation problems are carried out in discrete time and need to use some ad-hoc interpolation scheme to recover a continuous trajectory. One such technique is using cubic splines as in [7] and [8]. However, the specific interpolation scheme chosen encodes certain assumptions about the robot motion that may not be accurate. By explicitly formulating the problem in a continuous-time framework, the need to make these often arbitrary smoothness assumptions is eliminated.

Tong *et al.* [9], [10] showed that batch continuous-time estimation can be carried out by representing the trajectory as a Gaussian process. Barfoot *et al.* [11] extended the GP approach to STEAM, using a class of linear time-invariant (LTI) stochastic differential equations (SDEs). This resulted in an inverse kernel matrix that is exactly sparse, making the solution more computationally efficient. Anderson and Barfoot [1] extended this work in  $SE(3)$  to enable continuous-time estimation of bodies undergoing translations and rotations in three-dimensional space using a white-noise-on-acceleration (WNOA) prior. Tang *et al.* [4] further extended this by using a white-noise-on-jerk (WNOJ) prior in  $SE(3)$ . STEAM has been used in applications such as motion planning [2], crop monitoring [12], and visual teach and repeat [13].

Both WNOA and WNOJ models are commonly used in target tracking [14]–[17]. These models incorporate assumptions about the motion of the target that may not be realistic but are attractive due to their simplicity. When white-noise models are insufficient, Markov process models are used where the control input is modelled as a Markov process instead of a white-noise process. One example is the Singer acceleration model [6].

Another way to view modelling is through the use of latent-force models, which combine mechanistic and data-driven approaches [18]. In latent-force models, typically an overly simplistic mechanistic model of the system is used but augmented with latent forces represented as a GP. The idea is that when training latent-force models through data, the GP can model interactions not captured by the mechanistic model. Thus the GP must be rich enough to fully learn these interactions. In the case of the WNOA and WNOJ prior, the GP used is simply a white-noise process, which struggles with representational power. Thus tuning the hyperparameters of WNOA and WNOJ models alone do not give the model enough flexibility to learn a good representation through data.

Hartikainen and Särkkä [19] show how Gaussian process regression models can be restated as linear-Gaussian state space models. In particular, they show that the Matérn family of covariance functions can be exactly reformulated as a SDE with white noise. Furthermore, these Gaussian process latent force models can be reformulated as a single linear SDE driven by white noise [20], which has the form of priors in which we are interested.

Using latent-force models in state estimation has been done before but in a discrete-time estimation framework and where the states were limited to be in  $\mathbb{R}^n$  [21]–[23]. Regarding hyperparameter training for Gaussian process models, it is usually carried out by minimizing the negative log likelihood of data given some parameters but this approach has only been carried out in  $\mathbb{R}^n$  [24], [25].

To the best of our knowledge, the derivation we present in this letter is the first attempt at modelling the trajectory using a latent-force model with a Matérn covariance in the context of continuous-time trajectory estimation on  $SE(3)$  and using a data-driven approach to estimate its parameters.

## III. EXISTING CONTINUOUS-TIME MOTION PRIORS

In this section, we show the details of the existing WNOA and WNOJ motion priors used in STEAM. In order to ensure that estimation can be done efficiently, we are interested in motion priors from a class of linear time-invariant (LTI) stochastic differential equations (SDEs) of the form

$$\begin{aligned}\dot{\boldsymbol{\gamma}}(t) &= \mathbf{A}\boldsymbol{\gamma}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}\mathbf{w}(t), \\ \mathbf{w}(t) &\sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_c\delta(t-t')), \end{aligned} \quad (1)$$

where  $\boldsymbol{\gamma}(t)$  is the state at timestep  $t$ ,  $\mathbf{u}(t)$  is a known input, and  $\mathbf{w}(t)$  is a zero-mean, white-noise GP with power spectral density matrix,  $\mathbf{Q}_c$ . If  $\mathbf{u}(t) = \mathbf{0}$ , the solution for the mean function is

$$\check{\boldsymbol{\gamma}}(\tau) = \boldsymbol{\Phi}(\tau, t_k)\check{\boldsymbol{\gamma}}(t_k), \quad (2)$$

where  $\check{\boldsymbol{\gamma}}$  is the prior mean, and  $\boldsymbol{\Phi}(\tau, t_k)$  is the state transition function from timestep  $t_k$  to timestep  $\tau$ . We use GP priors because of their rich mathematical connection to motion models and the propagation of uncertainty is well understood.

### A. WNOA Prior for $SE(3)$

The WNOA prior originally used by STEAM is defined as follows:

$$\begin{aligned}\dot{\mathbf{T}}(t) &= \boldsymbol{\varpi}(t) \wedge \mathbf{T}(t) \\ \dot{\boldsymbol{\varpi}} &= \mathbf{w}'(t), \quad \mathbf{w}'(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}'_c\delta(t-t')), \end{aligned} \quad (3)$$

where  $\mathbf{T}(t)$  is the pose expressed as a transformation matrix,  $\boldsymbol{\varpi}(t)$  is the body-centric velocity and the operator,  $\wedge$ , transforms an element of  $\mathbb{R}^6$  into a member of Lie algebra,  $\mathfrak{se}(3)$ . The state is

$$\mathbf{x}(t) = \{\mathbf{T}(t), \boldsymbol{\varpi}(t)\}. \quad (4)$$

The SDE in (3) is nonlinear and so cannot be cast into the form of (1) but [1] defines the local state variables

$$\boldsymbol{\xi}_i(t) = \ln(\mathbf{T}(t)\mathbf{T}_i^{-1})^\vee, \quad t_i \leq t \leq t_{i+1} \quad (5)$$

$$\dot{\boldsymbol{\xi}}_i(t) = \mathcal{J}(\boldsymbol{\xi}_i(t))^{-1}\boldsymbol{\varpi}(t), \quad (6)$$

where the operator,  $\vee$ , converts a member of  $\mathfrak{se}(3)$  to  $\boldsymbol{\xi}_i(t) \in \mathbb{R}^6$  and  $\mathcal{J}(\boldsymbol{\xi}) \in \mathbb{R}^{6 \times 6}$  is the left Jacobian of  $SE(3)$ .

These local variables can be used to define a sequence of local priors where the prior at each timestep is a LTI SDE in the form described in (1) with

$$\gamma_i(t) := \begin{bmatrix} \xi_i(t) \\ \dot{\xi}_i(t) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}. \quad (7)$$

This local prior is a good approximation of the global non-linear prior when  $\xi_i(t)$  is small or when velocity is near constant.

#### B. WNOJ Prior for $SE(3)$

The WNOJ prior from [4] estimates the global state

$$\mathbf{x}(t) = \{\mathbf{T}(t), \boldsymbol{\omega}(t), \dot{\boldsymbol{\omega}}(t)\}, \quad (8)$$

where  $\dot{\boldsymbol{\omega}}(t)$  is the body-centric acceleration. Using the idea of local pose variables, [4] defines a sequence of local priors as a LTI SDE in the form of (1):

$$\gamma_i(t) := \begin{bmatrix} \xi_i(t) \\ \dot{\xi}_i(t) \\ \ddot{\xi}_i(t) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}. \quad (9)$$

The relationship between  $\xi_i(t)$  and  $\dot{\xi}_i(t)$  and global state variables are shown in Equations (5) and (6). The relationship between  $\ddot{\xi}_i(t)$  and global state variables as shown in [4] is

$$\ddot{\xi}_i(t) \approx -\frac{1}{2}(\mathcal{J}(\xi_i(t))^{-1}\boldsymbol{\omega}(t))^\wedge \boldsymbol{\omega}(t) + \mathcal{J}(\xi_i(t))^{-1}\dot{\boldsymbol{\omega}}(t), \quad (10)$$

where the approximation holds as long as  $\xi_i(t)$  is small.

### IV. LATENT-FORCE MODEL GP PRIOR

In the new motion prior that we derive, we represent the latent accelerations that the robot undergoes in each of its 6 degrees of freedom as a GP with a Matérn covariance function:

$$\ddot{\xi}_i(t) \sim \mathcal{GP}(\mathbf{0}, \mathcal{K}_v(t, t')). \quad (11)$$

For our prior, we choose  $v = \frac{1}{2}$ , which yields the exponential covariance function defined as

$$\mathcal{K}_{1/2}(t, t') = \sigma^2 \exp(-\ell^{-1}|t - t'|), \quad (12)$$

where  $\ell$ , the length-scale and  $\sigma^2$ , the variance, are diagonal matrices in  $\mathbb{R}^{6 \times 6}$ , with each diagonal term representing one of the 6 degrees of freedom.

Following the approach of [19], we can express the GP representing acceleration as the SDE

$$\begin{aligned} \ddot{\xi}_i(t) &= -\boldsymbol{\alpha}\ddot{\xi}_i(t) + \mathbf{w}(t), \\ \mathbf{w}(t) &\sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_c\delta(t - t')), \end{aligned} \quad (13)$$

where  $\boldsymbol{\alpha} = \ell^{-1}$  and  $\mathbf{Q}_c = 2\boldsymbol{\alpha}\sigma^2$ .

Now following the approach of [20], the model can be augmented to form a joint model in the form of (1) that includes the states  $\xi_i(t)$  and  $\dot{\xi}_i(t)$ :

$$\gamma_i(t) := \begin{bmatrix} \xi_i(t) \\ \dot{\xi}_i(t) \\ \ddot{\xi}_i(t) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{\alpha} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}. \quad (14)$$

The global state remains the same as in (8).

Our new motion prior now includes length-scale and variance as hyperparameters embedded in  $\boldsymbol{\alpha}$  and  $\mathbf{Q}_c$ , respectively. This allows much greater flexibility in our motion prior compared to the previous WNOA and WNOJ formulations that did not have a tunable length-scale parameter. In fact, WNOA and WNOJ are special cases of this motion prior. WNOA is the case when length-scale approaches 0, meaning accelerations are uncorrelated. WNOJ is the case when length-scale approaches  $\infty$ , meaning that accelerations are correlated to accelerations at every other time.

This particular parameterization of the motion prior collapses to the exact form of the Singer acceleration model in [6]. As such, we will be referring to this prior as the Singer prior. In this work, we chose this particular Matérn covariance function but we could potentially use other covariance functions that have more representational power.

#### A. Cost Terms in Optimization

The cost function for our estimator consists of the prior and measurement cost terms. These come from the negative log likelihood of the data given the measurements and the hyperparameters:

$$J = -\log p(\mathbf{x}|\mathbf{y}, \mathbf{Q}_c, \boldsymbol{\alpha}) = \underbrace{\sum_i J_i}_{\text{prior}} + \underbrace{\sum_j J_j}_{\text{measurement}}, \quad (15)$$

where  $\mathbf{x}$  is the state,  $\mathbf{y}$  is the measurements, and  $\mathbf{Q}_c$  and  $\boldsymbol{\alpha}$  are the hyperparameters. The optimal state estimate is then

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} J(\mathbf{x}), \quad (16)$$

where the state  $\mathbf{x}$  consists of the trajectory poses, velocities and accelerations, as defined in (8). The optimization is solved using Gauss-Newton with an  $SE(3)$  perturbation scheme [26], [27] to update the states:

$$\begin{aligned} \mathbf{T}_{\text{op},i} &\leftarrow \exp(\delta\xi_i^\wedge)\mathbf{T}_{\text{op},i}, \\ \boldsymbol{\omega}_{\text{op},i} &\leftarrow \boldsymbol{\omega}_{\text{op},i} + \delta\boldsymbol{\omega}_i, \\ \dot{\boldsymbol{\omega}}_{\text{op},i} &\leftarrow \dot{\boldsymbol{\omega}}_{\text{op},i} + \delta\dot{\boldsymbol{\omega}}_i, \end{aligned} \quad (17)$$

where  $(\cdot)_{\text{op}}$  is the operating point. Each prior cost term is

$$J_i = \frac{1}{2}\mathbf{e}_i^T \mathbf{Q}_i^{-1} \mathbf{e}_i. \quad (18)$$

The formulation of the prior error terms,  $\mathbf{e}_i$ , and covariances,  $\mathbf{Q}_i$ , for the WNOA and WNOJ prior can be found in [1] and [4]. In the next section, we show the derivations for our Singer prior.

#### B. Prior Error Term

The state transition function is

$$\begin{aligned} \Phi(t, t_i) &= \exp(\mathbf{A}\Delta t_i) \\ &= \begin{bmatrix} \mathbf{1} & \Delta t_i \mathbf{1} & (\boldsymbol{\alpha}\Delta t_i - \mathbf{1} + \exp(-\boldsymbol{\alpha}\Delta t_i))\boldsymbol{\alpha}^{-1} \\ \mathbf{0} & \mathbf{1} & (\mathbf{1} - \exp(-\boldsymbol{\alpha}\Delta t_i))\boldsymbol{\alpha}^{-1} \\ \mathbf{0} & \mathbf{0} & \exp(-\boldsymbol{\alpha}\Delta t_i) \end{bmatrix}. \end{aligned} \quad (19)$$

Using Equations (5), (6) and (10), the local state variables can be written in terms of global state variables as

$$\begin{aligned} \gamma_i(t_i) &= [\mathbf{0}^T \quad \varpi_i^T \quad \dot{\varpi}_i^T]^T, \\ \gamma_i(t_{i+1}) &= \begin{bmatrix} \ln(\mathbf{T}_{i+1,i})^\vee \\ \mathcal{J}_{i+1,i}^{-1} \varpi_{i+1} \\ -\frac{1}{2}(\mathcal{J}_{i+1,i}^{-1} \varpi_{i+1})^\wedge \varpi_{i+1} + \mathcal{J}_{i+1,i}^{-1} \dot{\varpi}_{i+1} \end{bmatrix}. \end{aligned} \quad (20)$$

Now in terms of global state variables, the prior error term is

$$\mathbf{e}_i = \gamma_i(t_{i+1}) - \Phi(t_{i+1}, t_i) \gamma_i(t_i) = [\mathbf{e}_p^T \quad \mathbf{e}_v^T \quad \mathbf{e}_a^T]^T, \quad (21)$$

where

$$\begin{aligned} \mathbf{e}_p &= \ln(\mathbf{T}_{i+1,i})^\vee - (t_{i+1} - t_i) \varpi_i - \mathbf{C}_1 \dot{\varpi}_i, \\ \mathbf{e}_v &= \mathcal{J}_{i+1,i}^{-1} \varpi_{i+1} - \varpi_i - \mathbf{C}_2 \dot{\varpi}_i, \\ \mathbf{e}_a &= -\frac{1}{2}(\mathcal{J}_{i+1,i}^{-1} \varpi_{i+1})^\wedge \varpi_{i+1} + \mathcal{J}_{i+1,i}^{-1} \dot{\varpi}_{i+1} - \mathbf{C}_3 \dot{\varpi}_i, \end{aligned} \quad (22)$$

and we have defined

$$\begin{aligned} \mathbf{C}_1 &= \boldsymbol{\alpha}^{-2}(\boldsymbol{\alpha}(t_{i+1} - t_i) - \mathbf{1} + \exp(-\boldsymbol{\alpha}(t_{i+1} - t_i))), \\ \mathbf{C}_2 &= \boldsymbol{\alpha}^{-1}(\mathbf{1} - \exp(-\boldsymbol{\alpha}(t_{i+1} - t_i))), \\ \mathbf{C}_3 &= \exp(-\boldsymbol{\alpha}(t_{i+1} - t_i)). \end{aligned} \quad (23)$$

The covariance matrix can be computed as

$$\begin{aligned} \mathbf{Q}_i(t) &= \int_0^{\Delta t_i} \exp(\mathbf{A}(\Delta t_i - s)) \mathbf{L} \mathbf{Q}_c \mathbf{L}^T \exp(\mathbf{A}(\Delta t_i - s))^T ds \\ &= \begin{bmatrix} 2\alpha\sigma^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 2\alpha\sigma^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 2\alpha\sigma^2 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} & \mathbf{Q}_{13} \\ \mathbf{Q}_{12}^T & \mathbf{Q}_{22} & \mathbf{Q}_{23} \\ \mathbf{Q}_{13}^T & \mathbf{Q}_{23}^T & \mathbf{Q}_{33} \end{bmatrix}, \end{aligned} \quad (24)$$

where

$$\begin{aligned} \mathbf{Q}_{11} &= \frac{1}{2} \boldsymbol{\alpha}^{-5} (1 - e^{-2\boldsymbol{\alpha}\Delta t_i} + 2\boldsymbol{\alpha}\Delta t_i + \frac{2}{3} \boldsymbol{\alpha}^3 \Delta t_i^3 \\ &\quad - 2\boldsymbol{\alpha}^2 \Delta t_i^2 - 4\boldsymbol{\alpha} \Delta t_i e^{-\boldsymbol{\alpha}\Delta t_i}), \\ \mathbf{Q}_{12} &= \frac{1}{2} \boldsymbol{\alpha}^{-4} (e^{-2\boldsymbol{\alpha}\Delta t_i} + 1 - 2e^{-\boldsymbol{\alpha}\Delta t_i} \\ &\quad + 2\boldsymbol{\alpha}\Delta t_i e^{-\boldsymbol{\alpha}\Delta t_i} - 2\boldsymbol{\alpha}\Delta t_i + \boldsymbol{\alpha}^2 \Delta t_i^2), \\ \mathbf{Q}_{13} &= \frac{1}{2} \boldsymbol{\alpha}^{-3} (1 - e^{-2\boldsymbol{\alpha}\Delta t_i} - 2\boldsymbol{\alpha}\Delta t_i e^{-\boldsymbol{\alpha}\Delta t_i}), \\ \mathbf{Q}_{22} &= \frac{1}{2} \boldsymbol{\alpha}^{-3} (4e^{-\boldsymbol{\alpha}\Delta t_i} - 3 - e^{-2\boldsymbol{\alpha}\Delta t_i} + 2\boldsymbol{\alpha}\Delta t_i), \\ \mathbf{Q}_{23} &= \frac{1}{2} \boldsymbol{\alpha}^{-2} (e^{-2\boldsymbol{\alpha}\Delta t_i} + 1 - 2e^{-\boldsymbol{\alpha}\Delta t_i}), \\ \mathbf{Q}_{33} &= \frac{1}{2} \boldsymbol{\alpha}^{-1} (1 - e^{-2\boldsymbol{\alpha}\Delta t_i}). \end{aligned}$$

### C. Querying the Trajectory

Because the prior we formulated is in continuous time, we now have the advantage of being able to interpolate for a state estimate at any given time. Suppose we have states at times  $t_i$  and

$t_{i+1}$  but want to query the state at time  $\tau$  where  $t_i < \tau < t_{i+1}$ . This can be done using the results from [1]:

$$\begin{bmatrix} \xi_i(\tau) \\ \dot{\xi}_i(\tau) \\ \ddot{\xi}_i(\tau) \end{bmatrix} = \gamma_i(\tau) = \boldsymbol{\Lambda}(\tau) \gamma_i(t_i) + \boldsymbol{\Omega}(\tau) \gamma_i(t_{i+1}), \quad (25)$$

where  $\boldsymbol{\Lambda}(\tau) \in \mathbb{R}^{18 \times 18}$  and  $\boldsymbol{\Omega}(\tau) \in \mathbb{R}^{18 \times 18}$  are [11]

$$\begin{aligned} \boldsymbol{\Lambda}(\tau) &= \Phi(\tau, t_i) - \boldsymbol{\Omega}(\tau) \Phi(t_{i+1}, t_i), \\ \boldsymbol{\Omega}(\tau) &= \mathbf{Q}_i(\tau) \Phi(t_{i+1}, \tau)^T \mathbf{Q}_i(t_{i+1})^{-1}. \end{aligned} \quad (26)$$

Using the relationship between the local and global state variables, we have that

$$\begin{aligned} \mathbf{T}_\tau &= \exp(\xi_i(\tau))^\vee \mathbf{T}_i, \\ \varpi_\tau &= \mathcal{J}(\xi_i(\tau)) \dot{\xi}_i(\tau). \end{aligned} \quad (27)$$

## V. HYPERPARAMETER TRAINING

We developed a method for hyperparameter training from data in  $SE(3)$ . Even after applying this principled method of choosing hyperparameters for our WNOA and WNOJ priors, we found that these priors were limiting in the type of trajectories that could be accurately represented. As such, we proposed a new motion prior in Section IV that has more representational power, but more parameters. The larger number of parameters greater highlights the importance of a principled hyperparameter training method, which we show how to do for  $SE(3)$  in this section. We show the hyperparameter training method for the Singer prior but the WNOA and WNOJ priors were also trained with data in a similar way.

The standard approach for hyperparameter training is to minimize the negative log likelihood of the data given the parameters [24], [25]:

$$\begin{aligned} -\log p(\mathbf{y} | \mathbf{Q}_c, \boldsymbol{\alpha}) &= \frac{1}{2} (\mathbf{y} - \check{\mathbf{x}})^T \mathbf{P}^{-1} (\mathbf{y} - \check{\mathbf{x}}) \\ &\quad - \frac{1}{2} \log |\mathbf{P}| + \frac{n}{2} \log 2\pi, \\ \mathbf{P} &= \check{\mathbf{P}}(\mathbf{Q}_c, \boldsymbol{\alpha}) + \sigma_n^2 \mathbf{1}, \end{aligned} \quad (28)$$

where  $\mathbf{y}$  is a stacked vector of ground truth measurements with additive noise  $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{1})$ ,  $\check{\mathbf{x}}$  is a stacked vector of the mean function evaluated at the ground truth measurement times and  $\check{\mathbf{P}}$  is the covariance matrix associated with  $\check{\mathbf{x}}$  and generated using the hyperparameters,  $\mathbf{Q}_c$  and  $\boldsymbol{\alpha}$ .

This cost function in this form does not lend itself nicely to training since our states are no longer vectors, but are in  $SE(3)$ . However, we can instead write the objective function of the prior in factor form as

$$-\log p(\mathbf{y} | \mathbf{Q}_c, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{e}^T \mathbf{Q}^{-1} \mathbf{e} + \frac{1}{2} \log |\mathbf{Q}| + \frac{n}{2} \log 2\pi, \quad (30)$$

where  $\mathbf{e}$  is a stacked vector of error terms from (21) composed with the ground truth measurements and  $\mathbf{Q}$  is the block diagonally stacked  $\mathbf{Q}_i$  terms from (24).

By making this simple modification to the objective function, we are able to train hyperparameters in  $SE(3)$  with noise-free ground truth measurements. However, the process of incorporating noisy ground truth measurements in  $SE(3)$  is slightly more

involved. We must incorporate the noise coming from the measurements to obtain a new value of  $\mathbf{Q}$  to be used in the objective function in (30), otherwise the noise in the measurements will be attributed to the process noise thus inflating the estimate of  $\mathbf{Q}_c$ .

Incorporating noise into the error equations and taking the approximation that process noise, measurement noise, and timesteps are small, we see that

$$\mathbf{e}_p \approx \boldsymbol{\epsilon}_{p_{i+1}} - \boldsymbol{\epsilon}_{p_i} - (t_{i+1} - t_i)\boldsymbol{\epsilon}_{v_i} - \mathbf{C}_1\boldsymbol{\epsilon}_{a_i} + \boldsymbol{\xi}_{p_i}, \quad (31)$$

$$\mathbf{e}_v \approx \boldsymbol{\epsilon}_{v_{i+1}} - \boldsymbol{\epsilon}_{v_i} - \mathbf{C}_2\boldsymbol{\epsilon}_{a_i} + \boldsymbol{\xi}_{v_i}, \quad (32)$$

$$\mathbf{e}_a \approx \boldsymbol{\epsilon}_{a_{i+1}} - \mathbf{C}_3\boldsymbol{\epsilon}_{a_i} + \boldsymbol{\xi}_{a_i}, \quad (33)$$

where

$$\begin{bmatrix} \boldsymbol{\epsilon}_{p_i} \\ \boldsymbol{\epsilon}_{v_i} \\ \boldsymbol{\epsilon}_{a_i} \end{bmatrix} = \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \mathbf{R}_{pp}^i & \mathbf{R}_{pv}^i & \mathbf{R}_{pa}^i \\ \mathbf{R}_{vp}^i & \mathbf{R}_{vv}^i & \mathbf{R}_{va}^i \\ \mathbf{R}_{ap}^i & \mathbf{R}_{av}^i & \mathbf{R}_{aa}^i \end{bmatrix} \right) \quad (34)$$

is the noise on the measurements at timestep  $i$  and

$$\begin{bmatrix} \boldsymbol{\xi}_{p_i} \\ \boldsymbol{\xi}_{v_i} \\ \boldsymbol{\xi}_{a_i} \end{bmatrix} = \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \mathbf{Q}_{pp}^i & \mathbf{Q}_{pv}^i & \mathbf{Q}_{pa}^i \\ \mathbf{Q}_{vp}^i & \mathbf{Q}_{vv}^i & \mathbf{Q}_{va}^i \\ \mathbf{Q}_{ap}^i & \mathbf{Q}_{av}^i & \mathbf{Q}_{aa}^i \end{bmatrix} \right) \quad (35)$$

is the process noise between timestep  $i$  and  $i + 1$ .

We now have that

$$\begin{aligned} \text{Cov}(\mathbf{e}_{p_i}, \mathbf{e}_{p_i}) &= \mathbf{R}_{pp}^{i+1} + \mathbf{R}_{pp}^i + \mathbf{C}_1\mathbf{R}_{aa}^i\mathbf{C}_1^T \\ &\quad + (t_{i+1} - t_i)\mathbf{R}_{pv}^i + (t_{i+1} - t_i)\mathbf{R}_{vp}^i \\ &\quad + (t_{i+1} - t_i)^2\mathbf{R}_{vv}^i + \mathbf{Q}_{pp}^i, \end{aligned}$$

$$\begin{aligned} \text{Cov}(\mathbf{e}_{p_i}, \mathbf{e}_{v_i}) &= (t_{i+1} - t_i)\mathbf{R}_{vv}^i + \mathbf{C}_1\mathbf{R}_{aa}^i\mathbf{C}_2^T \\ &\quad + \mathbf{R}_{pv}^{i+1} + \mathbf{R}_{pv}^i + \mathbf{Q}_{pv}^i, \end{aligned}$$

$$\text{Cov}(\mathbf{e}_{p_i}, \mathbf{e}_{a_i}) = \mathbf{C}_1\mathbf{R}_{aa}^i\mathbf{C}_3^T + \mathbf{Q}_{pa}^i,$$

$$\text{Cov}(\mathbf{e}_{v_i}, \mathbf{e}_{v_i}) = \mathbf{R}_{vv}^i + \mathbf{R}_{vv}^{i+1} + \mathbf{C}_2\mathbf{R}_{aa}^i\mathbf{C}_2^T + \mathbf{Q}_{vv}^i,$$

$$\text{Cov}(\mathbf{e}_{v_i}, \mathbf{e}_{a_i}) = \mathbf{C}_2\mathbf{R}_{aa}^i\mathbf{C}_3^T + \mathbf{Q}_{va}^i,$$

$$\text{Cov}(\mathbf{e}_{a_i}, \mathbf{e}_{a_i}) = \mathbf{C}_3\mathbf{R}_{aa}^i\mathbf{C}_3^T + \mathbf{R}_{aa}^{i+1} + \mathbf{Q}_{aa}^i,$$

$$\text{Cov}(\mathbf{e}_{p_{i+1}}, \mathbf{e}_{p_i}) = -\mathbf{R}_{pp}^{i+1} - (t_{i+2} - t_{i+1})\mathbf{R}_{vp}^{i+1},$$

$$\text{Cov}(\mathbf{e}_{p_{i+1}}, \mathbf{e}_{v_i}) = -(t_{i+2} - t_{i+1})\mathbf{R}_{vv}^{i+1} - \mathbf{R}_{pv}^{i+1},$$

$$\text{Cov}(\mathbf{e}_{p_{i+1}}, \mathbf{e}_{a_i}) = -\mathbf{C}_1\mathbf{R}_{aa}^{i+1},$$

$$\text{Cov}(\mathbf{e}_{v_{i+1}}, \mathbf{e}_{p_i}) = -\mathbf{R}_{vp}^{i+1},$$

$$\text{Cov}(\mathbf{e}_{v_{i+1}}, \mathbf{e}_{v_i}) = -\mathbf{R}_{vv}^{i+1},$$

$$\text{Cov}(\mathbf{e}_{v_{i+1}}, \mathbf{e}_{a_i}) = -\mathbf{C}_2\mathbf{R}_{aa}^{i+1},$$

$$\text{Cov}(\mathbf{e}_{a_{i+1}}, \mathbf{e}_{a_i}) = -\mathbf{C}_3\mathbf{R}_{aa}^{i+1}.$$



Fig. 2. The Buick test vehicle used for data collection. The vehicle is equipped with a Velodyne VLS-128 lidar, and an Applanix POS-LV system.

TABLE I  
NUMBER OF PARAMETERS COMPARED WITH NUMBER OF ITERATIONS TO CONVERGENCE FOR EACH OF THE THREE MOTION PRIORS

Prior	WNOA	WNOJ	Singer
# of Parameters	6	6	12
# of Iterations to Convergence	26	48	178

Putting this all together, we arrive at the final expression for  $\mathbf{Q}$  when our ground truth measurements are noisy:

$$\mathbf{Q} = \begin{bmatrix} \Sigma_{0,0} & \Sigma_{1,0}^T & & & \\ \Sigma_{1,0} & \Sigma_{1,1} & \Sigma_{2,1}^T & & \\ & \Sigma_{2,1} & \ddots & \ddots & \\ & & & \ddots & \ddots \end{bmatrix}, \quad (36)$$

where

$$\Sigma_{i,j} = \begin{bmatrix} \text{Cov}(\mathbf{e}_{p_i}, \mathbf{e}_{p_j}) & \text{Cov}(\mathbf{e}_{p_i}, \mathbf{e}_{v_j}) & \text{Cov}(\mathbf{e}_{p_i}, \mathbf{e}_{a_j}) \\ \text{Cov}(\mathbf{e}_{p_i}, \mathbf{e}_{v_j})^T & \text{Cov}(\mathbf{e}_{v_i}, \mathbf{e}_{v_j}) & \text{Cov}(\mathbf{e}_{v_i}, \mathbf{e}_{a_j}) \\ \text{Cov}(\mathbf{e}_{p_i}, \mathbf{e}_{a_j})^T & \text{Cov}(\mathbf{e}_{v_i}, \mathbf{e}_{a_j})^T & \text{Cov}(\mathbf{e}_{a_i}, \mathbf{e}_{a_j}) \end{bmatrix}. \quad (37)$$

Because our final expression for  $\mathbf{Q}$  is block tridiagonal, we are still able to exploit sparsity in hyperparameter training when evaluating our cost function.

## VI. EXPERIMENTAL VALIDATION

To evaluate our motion priors, we will be working with a dataset consisting of 36 km of driving with both Velodyne VLS-128 lidar data and ground truth from an Applanix POS-LV system. The experimental setup also includes hardware time synchronization between the lidar and the POS-LV system. The test vehicle used for the dataset collection is shown in Fig. 2.

The dataset consists of Route A,<sup>1</sup> which is 16 km long, and Route B,<sup>2</sup> which is 20 km long. Each of the three motion priors were trained using the method from Section V with the POS ground truth data from Route A. Table I shows the number of parameters for each model along with the number of iterations it took for hyperparameter training to converge. While the Singer model takes the most iterations to converge, hyperparameter training is a procedure with an upfront cost that only needs to be done once for each robotic platform.

<sup>1</sup>Map available at: <https://tinyurl.com/trjgxa>

<sup>2</sup>Map available at: <https://tinyurl.com/r5m78nq>

TABLE II

TRANSLATIONAL ERRORS EVALUATED ON LIDAR LOCALIZATION ALONG ROUTE B FOR THE THREE MOTION PRIORS AND THE REDUCTION IN ERROR ACHIEVED BY THE SINGER PRIOR

Seq. no.	WNOA (m)	WNOJ (m)	Singer (m)	Reduction in error from WNOA	Reduction in error from WNOJ
0	0.0690	0.0729	0.0677	<b>1.85%</b>	<b>7.01%</b>
1	0.0888	0.0892	0.0835	<b>5.95%</b>	<b>6.34%</b>
2	0.4071	0.3984	0.3999	<b>1.76%</b>	<b>-0.38%</b>
3	0.1947	0.1683	0.1667	<b>14.36%</b>	<b>0.93%</b>
4	0.2868	0.2686	0.2655	<b>7.43%</b>	<b>1.17%</b>
5	0.5703	0.5474	0.5471	<b>4.07%</b>	<b>0.05%</b>
6	0.3292	0.2850	0.2863	<b>13.03%</b>	<b>-0.45%</b>
7	0.2207	0.2224	0.2146	<b>2.74%</b>	<b>3.50%</b>
8	0.1115	0.1182	0.1126	<b>-0.95%</b>	<b>4.77%</b>
9	0.0979	0.1050	0.0979	<b>-0.02%</b>	<b>6.73%</b>
overall	0.2376	0.2275	0.2242	<b>5.64%</b>	<b>1.47%</b>

TABLE III

LONGITUDINAL VELOCITY ERRORS EVALUATED ON LIDAR LOCALIZATION ALONG ROUTE B FOR THE THREE MOTION PRIORS AND THE REDUCTION IN ERROR ACHIEVED BY THE SINGER PRIOR

Seq. no.	WNOA (m/s)	WNOJ (m/s)	Singer (m/s)	Reduction in error from WNOA	Reduction in error from WNOJ
0	0.2561	0.1794	0.2010	<b>21.53%</b>	<b>-12.04%</b>
1	0.1585	0.1123	0.1216	<b>23.26%</b>	<b>-8.25%</b>
2	0.1656	0.1341	0.1332	<b>19.57%</b>	<b>0.65%</b>
3	0.1590	0.1109	0.1120	<b>29.55%</b>	<b>-1.04%</b>
4	0.1427	0.1342	0.1157	<b>18.88%</b>	<b>3.75%</b>
5	0.2279	0.1477	0.1452	<b>36.31%</b>	<b>1.69%</b>
6	0.2484	0.1642	0.1505	<b>39.42%</b>	<b>8.35%</b>
7	0.1655	0.1733	0.1744	<b>-5.34%</b>	<b>-0.59%</b>
8	0.1370	0.1744	0.1371	<b>-0.05%</b>	<b>21.39%</b>
9	0.1988	0.1511	0.1565	<b>21.28%</b>	<b>-3.59%</b>
overall	0.1860	0.1482	0.1447	<b>22.18%</b>	<b>2.32%</b>

### A. Lidar Localization

We perform lidar localization as a batch trajectory optimization using each of the three trained motion priors on Route B, our 20 km long test trajectory. We use data from one of the runs along Route B to create a map of the area and then use a different run of Route B to perform lidar localization against the previously created map. We obtain 6-DOF pose estimates along with their covariances at 10 Hz from a lidar-only localization pipeline provided by Applanix. Because this pipeline is a commercial product that has been vigorously validated internally, we can assume that the covariance estimates are reasonable. We treat the pose estimates from lidar localization as pose measurements in our continuous-time estimator where we incorporate one of our motion priors. To ensure a fair comparison, all aspects of the estimator except for the motion prior are kept the same.

Since part of evaluating a motion prior is the quality of interpolation, we also interpolate at the ground truth timesteps since they occur more frequently than the lidar localization pose measurements. This also allows us to directly calculate localization errors.

We break down the test trajectory into 10 sequences and evaluate the performance of lidar localization on each of the sequences. The translational errors are shown in Table II where we see that the Singer prior results in an overall reduction of error by 5.64% compared to WNOA and 1.47% compared to WNOJ.

The errors in longitudinal velocity are shown in Table III, where we see that the Singer prior results in an overall reduction of error by 22.18% compared to WNOA and 2.32% compared to WNOJ.

Because the pose measurements we obtain from lidar localization are reliably accurate relative to ground truth, we perform another experiment where we decrease the frequency at which we receive pose measurements from lidar localization but use the same motion prior hyperparameters trained from ground truth measurements every 0.1 s. In our new experiment, instead of receiving pose measurements every 0.1 s from lidar localization, we increase measurement dropout to 1 s and all the way up to 5 s. The results from this experiment are shown in Fig. 3. With a 5 s measurement dropout, the Singer prior has a 29.57% reduction in translational error compared to WNOA and 67.89% compared to WNOJ. The Singer prior also decreased longitudinal velocity error by 7.25% for WNOA and 15.38% for WNOJ.

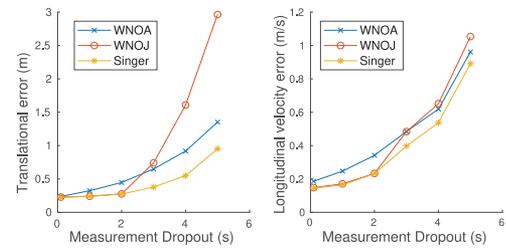


Fig. 3. Estimation errors for each of the three priors as measurement dropout is increased.

While the WNOJ prior outperforms the WNOA prior and is closely comparable to the Singer prior without measurement dropout, Fig. 3 shows it is not robust to increasing measurement dropout. This is because when the model chosen to represent trajectories cannot sufficiently represent the true model, hyperparameter training is more sensitive to the frequency of the ground truth measurements. The trained parameters work the best when the frequency of the ground truth training data is close to the frequency of the measurement test data. It was noted in [4] that the WNOJ prior is more sensitive to the hyperparameters chosen than the WNOA prior, which is consistent with our findings that the effect of measurement dropout on the WNOA prior is not as pronounced.

It is well known that if a model is too expressive, it may overfit to the training data and generalize poorly to new data. However, while the Singer model is more expressive, it is capable of a better fit to typical vehicle trajectories without overfitting to the training data. As a result, we see that the Singer model is able to maintain its performance advantage over the WNOA prior. Thus another advantage of the Singer prior over the WNOJ prior is that it is more robust to the frequency of measurements, which is desirable in a continuous-time estimation framework where the measurement frequency of the estimator does not need to be known beforehand.

As stated in Section IV, the WNOA prior assumes accelerations are uncorrelated with time while the WNOJ prior assumes that accelerations are correlated to accelerations at every other time. Both these assumptions are unrealistic because typical robot maneuvers will have accelerations correlated for a certain period of time (such as a car executing a turn). As such, the Singer prior allows the length scale of acceleration to be adjusted based on what we learn from data.

TABLE IV  
PERCENT TRANSLATION ERRORS EVALUATED ON LIDAR ODOMETRY ALONG ROUTE B AND THE REDUCTION IN ERROR ACHIEVED BY THE SINGER PRIOR

Seq. no.	WNOA	WNOJ (with patch)	WNOJ (without patch)	Singer	Reduction in error from WNOA	Reduction in error from WNOJ (with patch)	Reduction in error from WNOJ (without patch)
0	3.26%	3.23%	3.18%	3.13%	<b>4.14%</b>	<b>3.10%</b>	<b>1.54%</b>
1	2.89%	2.86%	2.85%	2.87%	<b>0.71%</b>	-0.20%	-0.52%
2	2.41%	2.39%	2.43%	2.41%	<b>0.08%</b>	-0.74%	<b>0.90%</b>
3	2.26%	2.25%	-	2.28%	-0.95%	-1.34%	-
4	2.37%	2.40%	-	2.39%	-0.83%	-	-
5	2.45%	2.40%	2.41%	2.38%	<b>2.75%</b>	<b>0.66%</b>	<b>0.91%</b>
6	2.56%	2.62%	2.57%	2.57%	-0.65%	<b>1.77%</b>	-0.01%
7	3.37%	3.33%	3.34%	3.34%	<b>0.72%</b>	-0.33%	-0.10%
8	3.57%	3.48%	3.48%	3.53%	<b>1.23%</b>	-1.39%	-1.30%
9	3.21%	3.17%	-	3.22%	-0.10%	-1.50%	-
overall (without 3,4,9)	2.929%	2.901%	2.894%	2.890%	<b>1.35%</b>	<b>0.39%</b>	<b>0.14%</b>
overall	2.834%	2.812%	-	2.811%	<b>0.82%</b>	<b>0.03%</b>	-

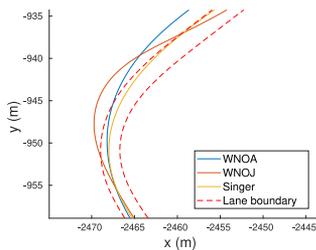


Fig. 4. The estimator using the Singer prior keeps the estimates within the lane boundaries while the WNOA and WNOJ estimates deviate outside.

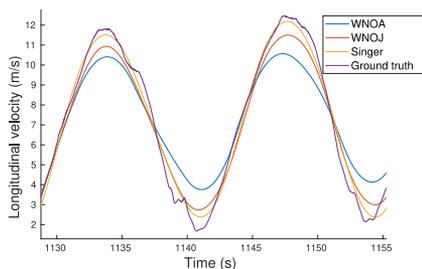


Fig. 5. The estimator using the Singer prior is able to better capture the peaks in longitudinal velocity which represent changes in acceleration.

Fig. 4 shows a curved portion of sequence 5 with a 4 s measurement dropout, where dotted red lines indicate the boundary at which the estimate of the car would cross the lane markings. We see that the Singer model is able to keep the estimate within the lane markings while both the WNOA and WNOJ estimates deviate outside of the markings for a short period of time.

Taking a look at Fig. 5, which shows the estimated velocities for another section of sequence 5 with a 4 s measurement dropout, we can see how the velocity estimates are much better using the Singer prior as the peaks are better matched to the ground truth. These peaks represent changes from acceleration to deceleration, which occur frequently in urban driving scenarios. The WNOA prior struggles the most to capture these motions while the Singer prior does the best.

### B. Lidar Odometry

We also evaluate our continuous-time motion priors with their trained hyperparameters on Route B, using the lidar-only odometry implementation from [28] and [4]. In this approach, a sliding-window is used rather than a batch optimization, where

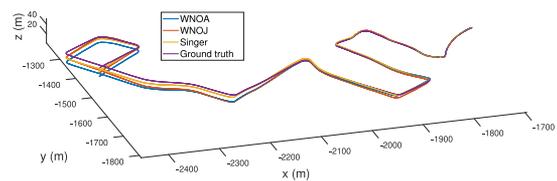


Fig. 6. 3D plot of odometry estimates for sequence 2 show that the estimator using the Singer prior is able to reduce drift in the  $z$  direction.

each window contains a reference point cloud consisting of 3 individual point clouds (where each is a single revolution of the lidar) and 2 target point clouds we are trying to align. We make use of the continuous-time interpolation scheme to handle point cloud motion distortion. We choose not to use the KITTI dataset because the point clouds have already been processed by the dataset authors to compensate for motion distortion.

Because of numerical instabilities, the WNOJ estimator in [4] required a patch that reverted to using a WNOA prior for a single window if any abnormalities were detected, such as a sudden increase in acceleration. We run lidar odometry with and without the patch and report results for both. Table IV shows the average percent translation errors over path segments of lengths 100, 200,  $\dots$ , 800 meters for each sequence, the same evaluation metric used in the KITTI odometry benchmark [29]. The WNOJ estimator without the patch failed for sequences 3, 4, and 9. Excluding these sequences, overall, the Singer prior decreases error by 1.35% from the WNOA prior and 0.14% from the WNOJ prior without the patch.

Fig. 6 shows the lidar odometry estimates for each of the motion priors for sequence 2, where the estimates for the WNOJ prior are without the patch. It can be seen that the estimator with the Singer prior reduces drift compared to the other two priors, most notably in the  $z$  direction.

We observe that the improvements in estimator accuracy with our lidar odometry experiment are not as significant compared to our lidar localization experiment. One major difference is that in lidar odometry, while the hyperparameters of the motion prior were trained in a principled way, the measurement covariances of the point cloud alignment, which are also hyperparameters of the estimator, were hand-picked. On the other hand, the pose measurements in lidar localization obtained from the Applanix pipeline have covariances attached, which were estimated in a principled manner.

It should be noted that the estimator using the Singer prior is able to run on all sequences without any patches, showing

its increased robustness. On all sequences, the Singer prior decreases error overall by 0.82% from the WNOA prior and maintains a very similar performance as the WNOJ prior, which used the patch.

## VII. CONCLUSIONS AND FUTURE WORK

In this letter, we showed that a continuous-time trajectory estimator in  $SE(3)$  using a Singer prior trained with data outperforms the existing white-noise-on-acceleration and white-noise-on-jerk priors also trained with data. It also exhibits more robustness compared to the WNOJ prior in the presence of extended measurement dropouts. This is because our hyperparameter training method allows us to use richer priors with more parameters to better fit the type of trajectories that our robot undergoes.

In our lidar odometry experiment, we did not have a principled way to choose the measurement covariances for our point cloud alignment. In the future, we can explore better methods to identify these measurement covariances to improve our estimator. Examples of work that estimate the covariance associated with point cloud alignment include [30] and [31].

In modelling the trajectories, regardless of the prior, we have represented each of the 6 pose variables as its own GP where there are no correlations between pose variables. However, this may not be the case as there can be correlations between them. As such, we could try to learn the parameters  $\mathbf{Q}_c$  and  $\alpha$  without constraining them to be diagonal matrices.

We have also assumed that we have ground truth measurements of all our states for the hyperparameter training. However, on some robotic platforms, this may not be possible. An extension would be a hyperparameter training method in  $SE(3)$  that only uses a subset of measurements.

Our new latent force model prior represented latent accelerations as a Matérn covariance function with  $\nu = 1/2$ , which we showed was equivalent to the Singer acceleration model. Another extension would be to explore the use of other covariance functions and then perform hyperparameter training to estimate their parameters.

Finally, we could consider incorporating control input data when training our motion prior to be able to better capture the dynamics of the robotic system.

## ACKNOWLEDGMENT

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

## REFERENCES

- [1] 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/RSSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 157–164.
- [2] M. Mukadam, J. Dong, F. Dellaert, and B. Boots, "Simultaneous trajectory estimation and planning via probabilistic inference," in *Robot.: Sci. Syst.*, 2017, doi: [10.15607/RSS.2017.XIII.025](https://doi.org/10.15607/RSS.2017.XIII.025).
- [3] X. Yan, V. Indelman, and B. Boots, "Incremental sparse GP regression for continuous-time trajectory estimation and mapping," *Robot. Auton. Syst.*, vol. 87, pp. 120–132, 2017.
- [4] T. Y. Tang, D. J. Yoon, and T. D. Barfoot, "A white-noise-on-jerk motion prior for continuous-time trajectory estimation on  $SE(3)$ ," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 594–601, Apr. 2019.
- [5] M. Mukadam, J. Dong, X. Yan, F. Dellaert, and B. Boots, "Continuous-time Gaussian process motion planning via probabilistic inference," *Int. J. Robot. Res.*, vol. 37, no. 11, pp. 1319–1340, Sep. 2018.
- [6] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-6, no. 4, pp. 473–483, Jul. 1970.
- [7] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spinning 2D laser," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 4312–4319.
- [8] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robot.: Sci. Syst.*, vol. 2, no. 9, 2014, doi: [10.15607/RSS.2014.X.007](https://doi.org/10.15607/RSS.2014.X.007).
- [9] C. H. Tong, P. T. Furgale, and T. D. Barfoot, "Gaussian process gauss-newton: Non-parametric state estimation," in *Proc. IEEE 9th Conf. Comput. Robot Vision*, 2012, pp. 206–213.
- [10] C. H. Tong, P. Furgale, and T. D. Barfoot, "Gaussian process gauss-newton for non-parametric simultaneous localization and mapping," *Int. J. Robot. Res.*, vol. 32, no. 5, pp. 507–525, 2013.
- [11] T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch continuous-time trajectory estimation as exactly sparse gaussian process regression," in *Robot.: Sci. Syst. Citeseer*, 2014, doi: [10.15607/RSS.2014.X.001](https://doi.org/10.15607/RSS.2014.X.001).
- [12] J. Dong, J. G. Burnham, B. Boots, G. Rains, and F. Dellaert, "4D crop monitoring: Spatio-Temporal Reconstruction for Agriculture," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3878–3885.
- [13] M. Warren, M. Greeff, B. Patel, J. Collier, A. P. Schoellig, and T. D. Barfoot, "There's no place like home: visual teach and repeat for emergency return of multirotor UAVs during GPS failure," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 161–168, Jan. 2019.
- [14] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. I: Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.
- [15] J. T. Jang *et al.*, "Trajectory generation with piecewise constant acceleration and tracking control of a quadcopter," in *Proc. IEEE Int. Conf. Ind. Technol.*, 2015, pp. 530–535.
- [16] T. Matsuzaki, H. Kameda, S. Tsujimichi, and K. Kosuge, "Maneuvering target tracking using constant velocity and constant angular velocity model," in *Proc. IEEE Int. Conf. Syst., Man Cybern*, 2000, vol. 5, pp. 3230–3234.
- [17] S. R. Jondhale and R. S. Deshpande, "Tracking target with constant acceleration motion using kalman filtering," in *Proc. IEEE Int. Conf. Adv. Commun. Comput. Technol.*, 2018, pp. 581–587.
- [18] M. Alvarez, D. Luengo, and N. D. Lawrence, "Latent force models," in *Proc. Artif. Intell. Statist.*, 2009, pp. 9–16.
- [19] J. Hartikainen and S. Särkkä, "Kalman filtering and smoothing solutions to temporal gaussian process regression models," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, 2010, pp. 379–384.
- [20] J. Hartikainen and S. Särkkä, "Sequential inference for latent force models," in *Proc. Conf. Uncertainty Artif. Intell.*, 2011, pp. 311–318.
- [21] J. Hartikainen, M. Seppänen, and S. Särkkä, "State-space inference for non-linear latent force models with application to satellite orbit prediction," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 723–730.
- [22] S. Särkkä, M. A. Alvarez, and N. D. Lawrence, "Gaussian process latent force models for learning and stochastic control of physical systems," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2953–2960, Jul. 2019.
- [23] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 283–298, Feb. 2008.
- [24] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006, vol. 2, no. 3.
- [25] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression," *Auton. Robots*, vol. 39, no. 3, pp. 221–238, 2015.
- [26] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [27] 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.
- [28] T. Tang, D. Yoon, F. Pomerleau, and T. D. Barfoot, "Learning a bias correction for lidar-only motion estimation," in *Proc. IEEE 15th Conf. Comput. Robot Vision*, 2018, pp. 166–173.
- [29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 3354–3361.
- [30] S. Bonnabel, M. Barczyk, and F. Goulette, "On the covariance of ICP-based scan-matching techniques," in *Proc. IEEE Amer. Control Conf.*, 2016, pp. 5498–5503.
- [31] D. Landry, F. Pomerleau, and P. Giguère, "CELLO-3D: Estimating the covariance of ICP in the real world," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 8190–8196.