CELLO: A fast algorithm for Covariance Estimation

In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2013).

CELLO: A Fast Algorithm for Covariance Estimation

William Vega-Brown, Abraham Bachrach, Adam Bry, Jonathan Kelly, and Nicholas Roy

***Abstract*— We present CELLO (Covariance Estimation and Learning through Likelihood Optimization), an algorithm for predicting the covariances of measurements based on any available informative features. This algorithm is intended to improve the accuracy and reliability of on-line state estimation by providing a principled way to extend the conventional fixed-covariance Gaussian measurement model. We show that in experiments, CELLO learns to predict measurement covariances that agree with empirical covariances obtained by manually annotating sensor regimes. We also show that using the learned covariances during filtering provides substantial quantitative improvement to the overall state estimate.**

I. Introduction

Reliable state estimation, often using information from multiple sensors, is essential for all robotics applications. Many state estimation problems are well-represented by temporally discrete hidden Markov models, with a latent state $\mathbf{x}_i \in \mathbb{R}^q$ at time t_i, and a corresponding observation $\mathbf{z}_i \in \mathbb{R}^p$. Using knowledge of the distributions over the initial state $p(\mathbf{x}_0)$, the transitions between states $p(\mathbf{x}_i|\mathbf{x}_{i-1})$, and the observations $p(\mathbf{z}_i|\mathbf{x}_i)$, together with the sequence of received observations $\{\mathbf{z}_0, \dots, \mathbf{z}_i\}$, we may make inferences about the latent states.

In real problems, these distributions are unknown. In order to make meaningful inferences, we are forced to choose approximations. To be useful, these approximatine distributions must be accurate enough to meet the constraints of the problem, while also permitting tractable filtering given the computational resources available—that is, they must permit efficient computation of an approximate posterior distribution for $\mathbf{x}_i$ conditioned on all previous observations $\{\mathbf{z}_k |\, k \in [1, i]\}$ from the approximating measurement distribution $p(\mathbf{z}_i|\mathbf{x}_i)$.

For a wide range of problems, these constraints are satisfied by choosing both $p(\mathbf{z}_i|\mathbf{x}_i)$ and $p(\mathbf{x}_i|\mathbf{x}_{i-1})$ to be multivariate normal distributions. Given deterministic (and possibly non-linear) functions $f(\mathbf{x})$ and $h(\mathbf{x})$ describing respectively the nominal state transition and measurement, along with symmetric positive definite matrices $\mathbf{Q}$ and $\mathbf{R}$ to parameterize the covariances associated with the state transition and measurement distributions, the model distribution may be written

$$\mathbf{x}_i \sim \mathcal{N}\left(f_i(\mathbf{x}_{i-1}), \mathbf{Q}_i\right) \tag{1}$$

$$\mathbf{z}_i \sim \mathcal{N}\left(h(\mathbf{x}_i), \mathbf{R}_i\right). \tag{2}$$

This research was sponsored by Ken Lodding and NASA Langley; Yakup Genc and Siemens Corporate Research; Behzad Kamgar-Parsi and ONR under the DR-IRIS MURI; and the ARL MAST Consortium. Their support is gratefully acknowledged.

The choice of multivariate normal distributions allows us to capture our estimate $\hat{\mathbf{x}}_i$ as a multivariate Gaussian, with deterministic update functions for mean and variance.

$$\hat{\mathbf{x}}_i \sim \mathcal{N}\left(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \tag{3}$$

$$\boldsymbol{\mu}_i = K_{\boldsymbol{\mu}}(\mathbf{x}_{i-1}, \mathbf{z}_i, \mathbf{Q}_{i-1}, \mathbf{R}_i) \tag{4}$$

$$\boldsymbol{\Sigma}_i = K_{\boldsymbol{\Sigma}}(\mathbf{x}_{i-1}, \mathbf{z}_i, \mathbf{Q}_{i-1}, \mathbf{R}_i) \tag{5}$$

The form of these update functions is prescribed by the chosen filter. If the transition and measurement functions are linear, then the Kalman filter [1] is the optimal choice; if they are nonlinear, there are a variety of suitable extensions, such as the extended [2] or unscented [3] Kalman filters.

In practice, this formulation works extremely well over a large space of problems. Provided the underlying distributions are unimodal and have effectively infinite support, the mean and covariance are often representative of the problem of interest even if the noise is not truly Gaussian, and the Kalman filter is fast enough to scale to very large or fast systems. The Kalman filter also provides a principled way of fusing data from multiple sensors. The *measurement covariance* $\mathbf{R}_i$ associated with each sensor defines the weight the estimator places on the measurements of that sensor, as well as how the estimate covariance $\boldsymbol{\Sigma}_i$ evolves, and how rapidly the estimate mean $\boldsymbol{\mu}_i$ adapts to new data.

The sensitivity of the performance of the Kalman filter to the values of the measurement and process noise covariances $\mathbf{R}$ and $\mathbf{Q}$ presents an often overlooked limitation: while there exist principled ways of determining suitable functions $f(\mathbf{x})$ and $h(\mathbf{x})$, there are few approaches to determining the corresponding covariance terms. Most implementations of the Kalman filter assume fixed $\mathbf{Q}$ and $\mathbf{R}$, taken as either equal to the empirical covariance of a body of sample data, or treated as parameters to be tuned. There is in general no reason to expect them to be time-invariant; in many cases sensor performance may vary with environmental parameters. Vision-based systems will degrade in darkness and at high speeds, while range-finding systems may perform poorly when faced with reflective or transparent surfaces, or when no surface is in range of the sensor. Moreover, if the measurement incorporated into the estimator is the output of an algorithm, such as a scan-matcher or visual odometry system, then algorithmic idiosyncrasies can further cause the covariance to vary. Choosing an appropriate covariance matrix becomes a matter of touch and tuning, and is generally settled by trading off the cost of uncertainty associated with an inflated covariance against the risk of inconsistency or incorrectness if the covariance is set too small.

Our approach is to attempt to predict the instantaneous measurement covariance $\mathbf{R}_i$, based on a vector of *predictors*, $\boldsymbol{\phi} \in \mathbb{R}^M$, computed from any data available to the estimator: features of the raw data an algorithm operates on, ancillary data produced by the algorithm, position in space, or even the current state estimate—anything expected to correlate with measurement covariance. The covariance associated with any given measurement is then given by

$$\mathbf{R}_i = \mathbf{R}(\boldsymbol{\phi}_i) \tag{6}$$

We use machine learning techniques to learn patterns from data drawn from experiments in the problem domain, rather than attempting to form an analytical estimate of this mapping.

In this paper, we present CELLO (Covariance Estimation and Learning through Likelihood Optimization), a novel algorithm for predicting the instantaneous covariances associated with sensor measurements. This algorithm extends the notion of an empirical covariance to describe a mapping from predictors to covariances, representing the covariance at any point in the predictor space as a weighted sum of outer products over body of training data. A method is presented for optimizing this weighting function to obtain maximal performance from a given body of data; by exploiting finite support in the weighting function and using existing data structures for fast nearest neighbor searches, we enable fast prediction at run-time.

We begin by outlining the theoretical roots of the algorithm, then describe in detail the methods used both for learning and to make predictions. We then describe our experimental results, both in simulation and on a micro air vehicle. We show the algorithm quantitatively improves estimates obtained from Kalman filtering, and is capable of making fast predictions even on very high-dimensional predictor spaces: our experiments included as many as thirty-eight predictors. Finally we describe limitations and possible routes to overcome them.

II. Related Work

The problem of learning a covariance mapping from data has attracted attention for many years. There have been many attempts to learn the covariances of many individual algorithms; Brenna [4], for instance, presented extensive work on covariance estimation for a laser scan matching algorithm. Most such efforts are tightly coupled to their parent algorithm, and they rarely generalize well. One early attempt at generalized covariance estimation was the adaptive Kalman filter [5], which modifies the elements of the measurement noise covariance $\mathbf{R}$ and the process noise covariance $\mathbf{Q}$ online. This learning relies purely on local noise characteristics; it is a reactive, rather than a predictive, scheme. Changes in the noise parameters will always be delayed while adaption occurs. This greatly reduces its effectiveness in cases of abrupt dramatic changes, as in outlier rejection.

Attempts to quantify the volatility of markets have led to advances by the computational finance community, yielding state-space methods like the multivariate dynamic linear models of Quintana and West [6], as well as regressive models like multivariate ARCH and its derivatives [7]. These methods are related to CELLO in that they also form weighted sums over outer products of prior data to generate their covariance predictions; however, they predict the instantaneous covariances as a time series, rather than as a map from an arbitrary predictor space, and as such share the disadvantages of the adaptive Kalman filter: they are reactive rather than predictive.

Bayesian non-parametric methods for covariance prediction have begun to emerge in the machine learning community. Wilson and Ghahramani [8] presented a kernel-based method which predicts a Wishart distribution over possible covariance matrices, along with a framework for doing Bayesian inference on such a model. While their formulation performs well for high-dimensional predictions, learning is prohibitively slow for high-dimensional feature spaces or large sample sizes. Further, prediction under the model is comparatively slow, rendering the algorithm unsuitable for use online. The multi-kernel Gaussian process of Melkumyan and Ramos [9] offers a different approach; rather than form an explicit distribution over positive definite matrices, it employs Gaussian processes to generate distributions over the elements of such matrices, using the choice of kernel to ensure positive definiteness. Like the Wishart process, it requires $\mathcal{O}(N)$ kernel computations to make a prediction, which will in general be too slow for online predictions. Other multi-output kernel methods, such as that of Álvarez et al. [10], have similar drawbacks.

III. Outer Product Estimation

Our goal is predict the covariance associated with sensor measurements. We define $\mathbf{e}_i \in \mathbb{R}^p$ as a vector representing measurement error:

$$\mathbf{e}_i = \mathbf{z}_i - h(\mathbf{x}_i) \tag{7}$$

We assume this error vector is drawn from a point in predictor space $\boldsymbol{\phi}_i \in \mathbb{R}^M$, and that the error is due to measurement stochasticity; we assume it to have mean zero, and denote its covariance as $\mathbf{R}_i$. We seek to predict this covariance given the predictor vector $\boldsymbol{\phi}_i$: $\mathbf{R}_i = \mathbf{R}(\boldsymbol{\phi}_i)$. We assume that we have available a data set

$$\mathcal{D} = \{\mathbf{e}_i, \boldsymbol{\phi}_i | \forall i \in [1, N]\}. \tag{8}$$

Note that a data set of this type requires that we have access to the error vectors $\mathbf{e}_i$, implying accurate knowledge of the true system state. In practice, this translates to requiring the training data be taken in a controlled environment, where state estimation is not an issue.

A. Local Empirical Covariance

In the limit of infinite training data, the maximum likelihood estimate of $\mathbf{R}(\boldsymbol{\phi})$ is the empirical covariance of all error vectors drawn from point $\boldsymbol{\phi}$.

$$\hat{\mathbf{R}}(\boldsymbol{\phi}_i) = \frac{1}{\sum_{k=1}^{N} \mathbf{1}_{\boldsymbol{\phi}_i = \boldsymbol{\phi}_k}} \sum_{k=1}^{N} \mathbf{1}_{\boldsymbol{\phi}_i = \boldsymbol{\phi}_k} \mathbf{e}_k \mathbf{e}_k{}^{\top}. \tag{9}$$

The indicator function $\mathbf{1}_{\boldsymbol{\phi}_i=\boldsymbol{\phi}_k}$ is equal to 1 if the subscripted condition is true and 0 otherwise. In practice, this approach is nearly useless; collecting data from the exact same point in feature space is difficult or impossible. To circumvent this limitation, we must incorporate data from points near the target point, but it is unclear which subset of $\mathcal{D}$ should be used to provide a good estimate of $\hat{\mathbf{R}}$ for a given $\boldsymbol{\phi}$.

To decide which subset of data may be incorporated, we first explicitly define the notion of 'nearness'. We then demonstrate that in the limit of infinite data and assuming continuity of the optimal map from predictors to covariances, taking the empirical covariance of error samples with associated predictors lying in a neighborhood of a test point will always converge to an estimate which differs from the optimal map by an amount bounded in the radius of the neighborhood. It follows that by shrinking that neighborhood, the error bound can be made arbitrarily small.

Let the function $\mathbf{R}(\boldsymbol{\phi})$ be a Lipschitz continuous map from $\mathbb{R}^M$, the space of predictors, onto $\mathbb{R}^{p\times p} > 0$, the space of symmetric positive definite matrices. This continuity requirement is minimally restrictive; without it, the map $\mathbf{R}(\boldsymbol{\phi})$ may vary arbitrarily fast and we cannot do better than the map described in equation (9). Given continuity, there must exist a constant K such that

$$|\mathbf{R}^{mn}(\boldsymbol{\phi})-\mathbf{R}^{mn}(\boldsymbol{\phi}')|<K\rho(\boldsymbol{\phi},\boldsymbol{\phi}') \quad \forall\{m,n\}\in[1,p]. \tag{10}$$

Here, superscripts indicate an element of a matrix, while subscripts indicate sample indices; $\rho(\boldsymbol{\phi},\boldsymbol{\phi}')$ is a distance metric on the space of predictors. Using this metric we define a ball $\mathbb{B}$ of radius ϵ centered at $\boldsymbol{\phi}_0$, and use the samples contained in the ball to estimate $\mathbf{R}(\boldsymbol{\phi}_0)$. We then have

$$|\mathbf{R}^{mn}(\boldsymbol{\phi}_0) - \hat{\mathbf{R}}^{mn}(\boldsymbol{\phi})| < K\epsilon \qquad \forall\boldsymbol{\phi}\in\mathbb{B}, \tag{11}$$

Defining the matrices $\mathbf{R}_0 = \mathbf{R}(\boldsymbol{\phi}_0)$ and $\boldsymbol{\Delta}_i = \mathbf{R}(\boldsymbol{\phi}_i) - \mathbf{R}(\boldsymbol{\phi}_0)$, algebraic manipulation yields

$$\mathrm{E}\left[\mathbf{e}_i^m\mathbf{e}_i^n\right] = \mathbf{R}_0^{mn} + \boldsymbol{\Delta}_i^{mn} \tag{12}$$

$$\mathrm{Var}\left[\mathbf{e}_i^m\mathbf{e}_i^n\right] = \begin{cases} |\mathbf{R}_0+\boldsymbol{\Delta}_i|^{mn} & m\neq n \\ 2(\mathbf{R}_0^{mm}+\boldsymbol{\Delta}_i^{mm})^2 & m=n \end{cases}, \tag{13}$$

where $|\mathbf{M}|^{mn} = \mathbf{M}^{mm}\mathbf{M}^{nn} - (\mathbf{M}^{mn})^2$ is the complement of the (m,n) second minor of $\mathbf{M}$. Determining the mean and variance of the outer product sum is then straightforward. We let $N_{\mathbb{B}}$ be the number of samples in the ball, and note

$$\hat{\mathbf{R}}^{mn} = \frac{1}{N_{\mathbb{B}}}\sum_{i=1}^{N_{\mathbb{B}}}\mathbf{e}_i^m\mathbf{e}_i^n. \tag{14}$$

Well-known properties of the expectation and variance yield

$$\mathrm{E}\left[\hat{\mathbf{R}}^{mn}\right] = \mathbf{R}_0^{mn} + \frac{1}{N_{\mathbb{B}}}\sum_{i=1}^{N_{\mathbb{B}}}\boldsymbol{\Delta}_i^{mn} \tag{15}$$

$$\mathrm{Var}\left[\hat{\mathbf{R}}^{mn}\right] = \frac{1}{N_{\mathbb{B}}^2}\sum_{i=1}^{N_{\mathbb{B}}}\mathrm{Var}\left[\mathbf{e}_i^m\mathbf{e}_i^n\right]. \tag{16}$$

Noting that the magnitude of $\boldsymbol{\Delta}_i^{mn}$ is bounded by $K\epsilon$, we have shown

$$|\mathrm{E}\left[\hat{\mathbf{R}}^{mn}\right] - \mathbf{R}_0^{mn}| < K\epsilon \tag{17}$$

$$\mathrm{Var}\left[\hat{\mathbf{R}}^{mn}\right] < \frac{1}{N_{\mathbb{B}}}\begin{cases} |\mathbf{R}_0|^{mn} + B^{mn}K\epsilon & m\neq n \\ 2((\mathbf{R}_0^{mm})+K\epsilon)^2 & m=n \end{cases}, \tag{18}$$

where $B^{mn} = (\mathbf{R}_0^{mm} + \mathbf{R}_0^{nn} + 2\mathbf{R}_0^{mn})$ is introduced to simplify notation. Equation (18) implies the variance on our estimate will always converge to zero as the number of samples tends to infinity, and equation (17) implies it will be a consistent estimate in the limit as ϵ tends to zero. Given a fixed amount of data, then, the choice of ϵ balances accuracy against precision of the estimate.

B. Learning the Metric

Choosing ϵ in a principled way requires knowledge not just of $\mathbf{R}(\boldsymbol{\phi})$, but also of the Lipschitz constant K; in practice, neither is available. Moreover, it is not obvious what is a sensible choice for $\rho(\boldsymbol{\phi},\boldsymbol{\phi}')$: the elements of the vector $\boldsymbol{\phi}$ may be taken from functions with vastly different scales, or even different ranges. There is no intuitive way to compare an image brightness to its contrast ratio, or a vehicle velocity to a fraction of inliers. Heuristic estimates can be of some assistance; we could, for instance, rescale each element to lie between zero and one, given a known finite range of possible values. However, doing so implicitly assigns each element equal importance, tacitly asserting that the elements of $\mathbf{R}_0$ are expected to vary at the same rate in every direction.

Given the difficulties in choosing a metric and a ball radius ϵ, we instead choose to learn these parameters from data, which permits us to explicitly choose how to balance the likelihood of our model against the precision of the implied estimate. We set $\epsilon = 1$ and choose a family of distance metrics parameterized by a vector $\boldsymbol{\theta}$; by rescaling the distance metric we can effectively force the radius of the ball to be whatever we wish. The simplest choice of distance metric, and one that proved effective in experiments, is to simply choose a scale factor for each element of $\boldsymbol{\phi}$, and use the Euclidean norm of this rescaled vector.

C. Non-Unit Weights

We may rewrite the local empirical sum of equation (14) in the form of equation (9), as a sum over all samples.

$$\hat{\mathbf{R}}(\boldsymbol{\phi}_0) = \frac{1}{\sum\mathbf{1}_{\rho(\boldsymbol{\phi}_i,\boldsymbol{\phi}_0)<1}}\sum_{i=1}^{N}\mathbf{1}_{\rho(\boldsymbol{\phi}_i,\boldsymbol{\phi}_0)<1}\mathbf{e}_i\mathbf{e}_i{}^{\top} \tag{19}$$

This is a weighted sum, where all weights have either unit or zero magnitude. This choice of weighting function is somewhat arbitrary; in general, any positive function $k(\rho(\boldsymbol{\phi},\boldsymbol{\phi}')) > 0$ will still create valid covariance matrices. Provided the function is decreasing in $\rho(\boldsymbol{\phi},\boldsymbol{\phi}')$, we can make statements about the asymptotic error and variance of the resulting matrices similar to those in section III-A.

Defining $k_i = k(\rho(\boldsymbol{\phi}_i, \boldsymbol{\phi}_0))$ and $N_{\text{eff}} = \sum_{i=1}^{N} k_i$, we write equations analogous to equations (15) and (17).

$$\mathrm{E}\left[\hat{\mathbf{R}}^{mn}\right] = \mathbf{R}_0^{mn} + \frac{1}{N_{\text{eff}}} \sum_{i=1}^{N} k_i \boldsymbol{\Delta}_i^{mn} \tag{20}$$

$$\therefore \quad |\mathrm{E}\left[\hat{\mathbf{R}}^{mn}\right] - \mathbf{R}_0^{mn}| < K \sum_{i=1}^{N} \frac{k_i \rho(\boldsymbol{\phi}_i, \boldsymbol{\phi}_0)}{N_{\text{eff}}} \tag{21}$$

Choosing $k(\cdot,\cdot)$ decreasing in ρ, and noting $\frac{k_i}{N_{\text{eff}}} \leq 1$ and $\rho \leq \epsilon$ by construction, it follows that

$$|\mathrm{E}\left[\hat{\mathbf{R}}^{mn}\right] - \mathbf{R}_0^{mn}| < K \sum_{i=1}^{N} \frac{k_i \rho(\boldsymbol{\phi}_i, \boldsymbol{\phi}_0)}{N_{\text{eff}}} < K\epsilon. \tag{22}$$

Non-unit weighting functions strictly reduce the bound on our error between the estimated and optimal covariance map: we can only improve the accuracy of our estimate by using a radial weighting function. The corresponding result for variance is

$$\mathrm{Var}\left[\hat{\mathbf{R}}^{mn}\right] = \frac{1}{N_{\text{eff}}^2} \sum_{i=1}^{N} k_i \mathrm{Var}\left[\mathbf{e}_i^m \mathbf{e}_i^n\right] \tag{23}$$

$$< \frac{1}{N_{\text{eff}}} \begin{cases} A^{mn} + B^{mn} K\hat{\epsilon} & m \neq n \\ 2((\mathbf{R}_0^{mm}) + K\hat{\epsilon})^2 & m = n \end{cases}, \tag{24}$$

where $\hat{\epsilon} = \sum_{i=1}^{N} \frac{k_i \rho(\boldsymbol{\phi}_i, \boldsymbol{\phi}_0)}{N_{\text{eff}}} \leq \epsilon$. Since $N_{\text{eff}} < N_{\mathbb{B}}$, non-unit weighting functions decrease the precision of our estimate. If samples are uniformly distributed in predictor-space with sample density σ, choosing a new radius $\epsilon' > \epsilon$ such that $\hat{\epsilon}' = \epsilon$ will allow us to include an expected $\sigma(\epsilon'^M - \epsilon^M)$ additional samples, offsetting this loss. By choosing non-unit weights and manipulating the weighting function, we may expect to obtain increased precision and accuracy with the same available data.

IV. Covariance Estimation

Given a dataset composed of sampled pairs of features and errors, $\mathcal{D} = \{\boldsymbol{\phi}_i, \mathbf{e}_i\}$. the framework induces a general form for predicted covariances.

$$\hat{\mathbf{R}}(\boldsymbol{\phi}) = \frac{1}{N_{\text{eff}}} \sum_{i=1}^{N} k(\rho(\boldsymbol{\phi}, \boldsymbol{\phi}_i)) \mathbf{e}_i \mathbf{e}_i{}^{\top} \tag{25}$$

To make predictions, we also require a metric on the M-dimensional space of predictors, $\rho(\boldsymbol{\phi}, \boldsymbol{\phi}')$ and a decreasing positive weighting function $k(\rho(\boldsymbol{\phi}, \boldsymbol{\phi}'))$. While choosing these functions optimally is an open question for future research, simple choices perform well in practice.

We employ a scaled Euclidean form for the metric, parameterized by a vector $\boldsymbol{\theta}$.

$$\rho(\boldsymbol{\phi}, \boldsymbol{\phi}') = (\boldsymbol{\phi} - \boldsymbol{\phi}')^{\top} \boldsymbol{\Theta}^{\top} \boldsymbol{\Theta} (\boldsymbol{\phi} - \boldsymbol{\phi}') \tag{26}$$

The *scale matrix* $\boldsymbol{\Theta}$ is chosen upper triangular, with nonzero elements equal to the elements of the parameter vector $\boldsymbol{\theta}$; forming the distance metric in this way guarantees positivity for all $\boldsymbol{\theta}$. Moreover, if we choose a weighting function $k(\rho(\boldsymbol{\phi}, \boldsymbol{\phi}'))$ with compact support, then we only need calculate the weights for samples within a fixed distance of the target point. Identifying these samples is the well-studied problem of nearest neighbor search in a Euclidean space, which may be done in sublinear time by storing $\{\boldsymbol{\Theta}\boldsymbol{\phi}_i | \forall i \in [1, N]\}$ in a k-D tree.

There are many suitable weighting functions $k(\rho(\boldsymbol{\phi}, \boldsymbol{\phi}'))$; we chose the triangle function

$$k(\rho(\boldsymbol{\phi}, \boldsymbol{\phi}')) = \begin{cases} 1 - \rho(\boldsymbol{\phi}, \boldsymbol{\phi}') & \text{if } \rho(\boldsymbol{\phi}, \boldsymbol{\phi}') < 1 \\ 0 & \text{else} \end{cases} \tag{27}$$

for its simplicity. In practice, any decreasing positive function will do, and the choice has minimal impact on the resulting predictions.

The assumption of Gaussianity implies

$$\mathbf{e}_i \sim \mathcal{N}\left(\mathbf{0}_p, \hat{\mathbf{R}}(\boldsymbol{\phi}_i | \boldsymbol{\theta})\right). \tag{28}$$

Using the standard Gaussian likelihood function, we may evaluate the logarithm of the likelihood of this model for a given parameter vector $\boldsymbol{\theta}$.

$$\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = -\frac{1}{2} \sum_{i=1}^{N} \left(\log |\hat{\mathbf{R}}(\boldsymbol{\phi}_i)| + \mathbf{e}_i{}^{\top} \hat{\mathbf{R}}(\boldsymbol{\phi}_i)^{-1} \mathbf{e}_i\right) \tag{29}$$

The maximum likelihood parameters $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$ are the natural choice for making predictions. However, this choice risks overfitting. In addition, if the samples are not uniformly distributed in the space of possible predictor vectors, then the covariance predictions will be high-variance in regions where data is sparse, and will become singular if the number of available data pairs is less than the dimensionality of the measurements.

We observe better results by including regularization in two forms. First, we employ 'leave-one-out' validation in the optimization; we evaluate the covariance of $\boldsymbol{\phi}_i$ using data tuples $\{\boldsymbol{\phi}_j, \mathbf{e}_j | \forall j \in [1, N] \setminus i\}$—that is, all data excluding the point of interest. Second, we include an explicit regularization term in the optimization.

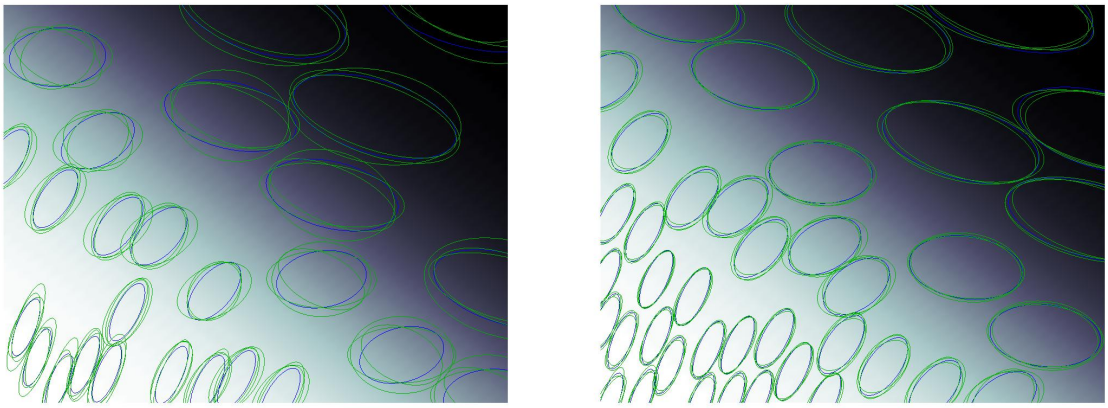
$$\mathcal{R}(\boldsymbol{\theta}|\mathcal{D}) = \sum_{i=1}^{N} \log(k(\rho(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j))) \tag{30}$$

Including this term gives the optimizer an indication of how to modify the parameter vector to increase data density.

If we define a regularization constant $\alpha \in [0, 1]$ we may write our objective function as

$$\mathcal{F}(\boldsymbol{\theta}|\mathcal{D}) = \sum_{i=1}^{N} (1 - \alpha)\mathcal{L}_i(\hat{\mathbf{R}}_i|\mathcal{D}) + \alpha\mathcal{R}_i(\boldsymbol{\theta}|\mathcal{D}). \tag{31}$$

Choosing α requires experimentation in the problem domain; typical values are quite small, on the order of 10^{-3}. Note that this objective function is continuous and twice analytically differentiable; since we have access to both the Jacobian and the Hessian matrix, we may then perform optimization using any desired method. Algorithm 1 presents a method employing stochastic gradient descent, which proved effective in experiment.



(a) *100 samples* (b) *1000 samples*

Fig. 1: *Two-dimensional covariance learning. The blue ellipses are true measurement covariances drawn from the data set; the green ellipses are the optimized estimated covariances, with results drawn from several independent data sets superimposed. In each case, the estimates appear unbiased, and the variance of the covariance ellipses sampled from the learned model is visible smaller in places with more data.*

Algorithm 1 Optimization Algorithm

Randomly initialize parameter vector $\boldsymbol{\theta}$ and learning rate η
repeat
  $\mathcal{I} \leftarrow$ SHUFFLE($[1, \ldots, N]$)
  for $i \in \mathcal{I}$ **do**
    $\hat{\mathbf{R}}_i \leftarrow$PREDICTCOVARIANCE($\boldsymbol{\phi}^{(i)}$)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla \mathcal{F}_i(\hat{\mathbf{R}}_i | \mathcal{D})$
  end for
until convergence
function PREDICTCOVARIANCE($\boldsymbol{\phi}$)
  $\hat{\mathbf{R}} \leftarrow \mathbf{0}_{p \times p}$
  $N_{\boldsymbol{\phi}} \leftarrow$NEARESTNEIGHBORS($\boldsymbol{\phi}$)
  for $i \in N_{\boldsymbol{\phi}}$ **do**
    $\hat{\mathbf{R}} \leftarrow \hat{\mathbf{R}} + k(\rho(\boldsymbol{\phi}, \boldsymbol{\phi}_i)) \mathbf{e}_i \mathbf{e}_i{}^\top$
    $N_{\text{eff}} \leftarrow N_{\text{eff}} + k(\rho(\boldsymbol{\phi}, \boldsymbol{\phi}_i))$
  end for
  $\hat{\mathbf{R}} \leftarrow \frac{1}{N_{\text{eff}}} \hat{\mathbf{R}}$
  return $\hat{\mathbf{R}}$
end function

V. SIMULATION RESULTS

A. Dark Room

The algorithm was first validated in simulation on a toy problem representative of target problem domains. We consider a fictional robot taking position measurements in a room of varying brightness. The fictional position sensor performs well in the light, but poorly in the darkness; the robot navigates the room, and compares its measurements to ground truth values at many locations, storing the location as a predictor vector $\boldsymbol{\phi}_i$ and the difference between observed position and true position as an error vector $\mathbf{e}_i$. This provides the data set $\mathcal{D}$ required to use algorithm 1; we use the Euclidean metric presented in equation (26), and learn the elements of the scale matrix $\boldsymbol{\Theta}$. The results of the learning process are presented in Figure (1), when (1a) 100 and (1b) 1000 samples were used for training.

B. Scan-Matching

The planar LIDAR unit is ubiquitous in two-dimensional localization and mapping tasks; it returns range measurements at a fixed sequence of angles, and provides an excellent balance of small size, light weight, low price, and high accuracy. The range scans may be used to build maps [11], to localize within a known map [12], or—by matching sequential scans—for motion estimation via dead reckoning [13].

However, one limitation of the sensor that often creates estimation challenges is its finite range. In a hallway-like environment, where the laser can see the walls of the hallway but not the ends, we expect a low uncertainty in the direction of the walls, where information is available, but a high uncertainty along the length of the hallway, where information is missing. In the limiting case of parallel planar walls and no sensor noise, we can obtain our transverse position in the hallway exactly, but make no guess as to our longitudinal position. This behavior can be very difficult to capture with fixed measurement covariances without resorting to modelling the range scan measurement in terms of sophisticated environment-specific features.

We demonstrate the benefits of an adaptive covariance scheme through prediction of the covariances of the transform parameters between matched sequential simulated laser scans in a hallway environment, using a predictor vector composed of histograms of angles returning viable ranges, and the angles formed by lines between sequential points. The predicted covariance ellipses are drawn in figure (2) at several locations in a hallway; the predictions consistently align the covariance ellipse with the hallway, regardless of the robot orientation.

We also simulate accelerometer data, and incorporate these predicted covariances into an unscented Kalman filter. A

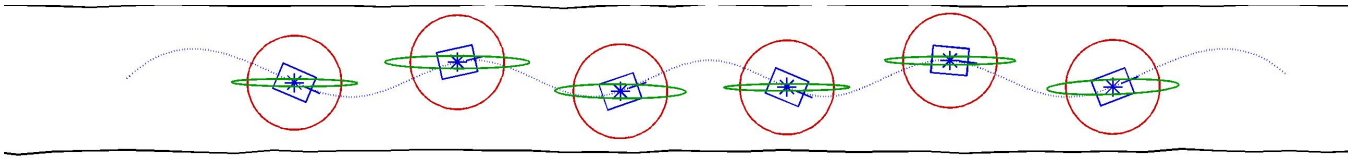
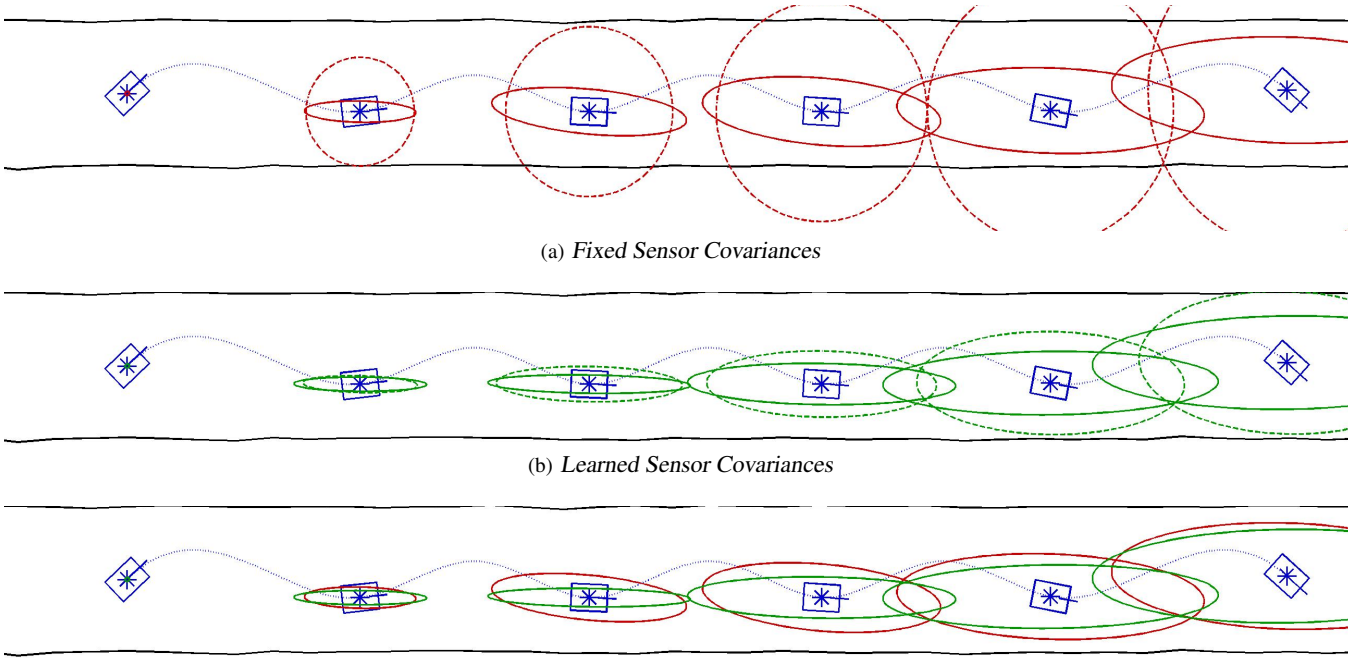


Fig. 2: *Learned measurement covariances (green) for a scan matching algorithm in a hallway, compared to fixed measurement covariances (red) taken as the empirical covariance of the available sample data. Measurements are displacements between successive scans, consequently they have the same domain as the position variables; covariance ellipses are presented at arbitrary scale, hence it is only the orientation and eccentricity of these ellipses that is relevant. The learned covariances consistently indicate a large measurement variation in the longitudinal direction, and a small variation in the transverse direction, reflecting the availability of information in only one direction. Note the learning was done using predictor features taken purely from the scan, and not including the position or orientation of the robot—the rotation of the ellipse is an emergent behavior.*



(a) *Fixed Sensor Covariances*

(b) *Learned Sensor Covariances*

(c) *Comparison of Simulated Covariances*

Fig. 3: *Comparison of filter performance using learned and fixed measurement covariances. Note that in contrast to figure 2, these covariances are state covariances. Covariances are drawn as 95% confidence margin ellipses at fixed time intervals. Each estimated covariance (dashed lines) is the mean filter covariances for a given time interval i, $\mathrm{E}\left[\mathbf{\Sigma}_i\right]$, over 250 simulated trials. Each simulated covariance (solid lines) is the covariance of the estimated trajectories for the time interval i, $\mathrm{Cov}\left[\boldsymbol{\mu}_i\right]$, again over 250 simulations. Filtering was done using an unscented Kalman filter with noisy accelerometer data and the scan-matching output for measurements. Note the distortion of the ellipse in the learned case, reflecting increased uncertainty in the longitudinal direction of the corridor. Note also that the fixed covariance scheme underestimates its uncertainty in the longitudinal direction.*

comparison of filter performance using (3a) fixed covariances and (3b) learned covariances indicates the weaknesses of a fixed covariance scheme; the filter underestimates its uncertainty in the longitudinal direction but grossly overestimates it in the transverse direction. The result is reflected in the distribution of estimated trajectories (solid lines). The learned covariance scheme does a much better job restricting its estimates to the interior of the hallway.

VI. Experimental Results

The learning and prediction process was experimentally evaluated on the output of an optical flow algorithm, operating on an image stream from a downward facing camera on a quad-rotor helicopter. The measurement vector $\mathbf{z}$ consists of the apparent motion of the image along its two axes, along with its apparent rotation and scale shift. By tracking the motion of the image stream and assuming a flat ground plane, it is possible to infer the motion of the vehicle: image translation implies either roll, pitch, or horizontal translation, while image rotation implies yaw and image scale shift implies vertical motion. In low-texture environments, in darkness, or when images are blurred by rapid motion, the image registration process is impaired and motion estimates degrade.

Eighteen predictors were chosen and calculated for each image pair, to produce a predictor vector $\boldsymbol{\phi}$; these predictors were chosen to be indicative of image contrast and structure, as well as the goodness of fit of the image registration process. The vehicle was flown in a motion capture room; the

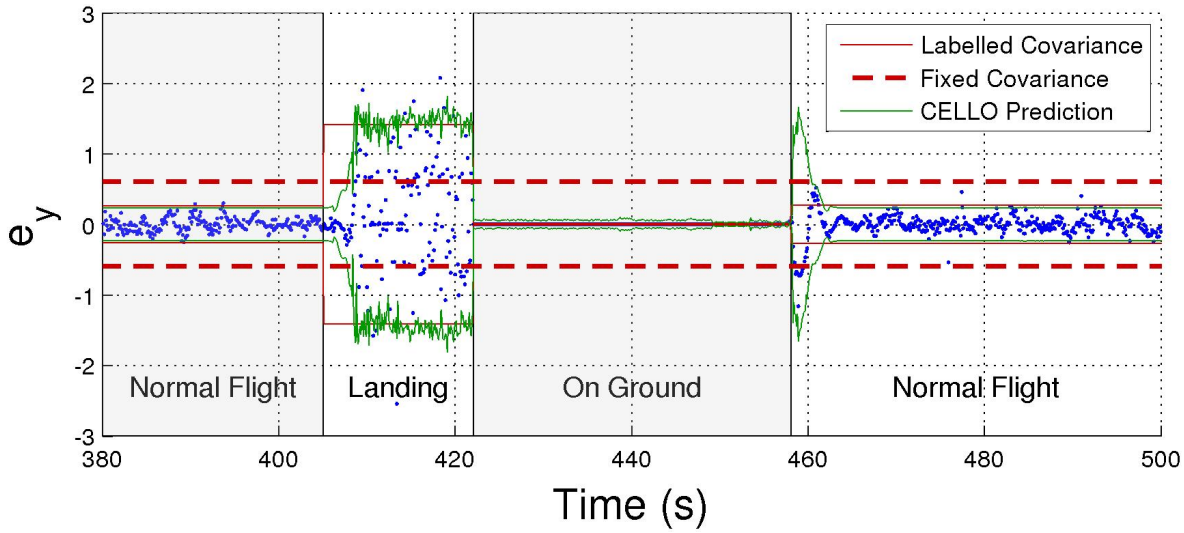


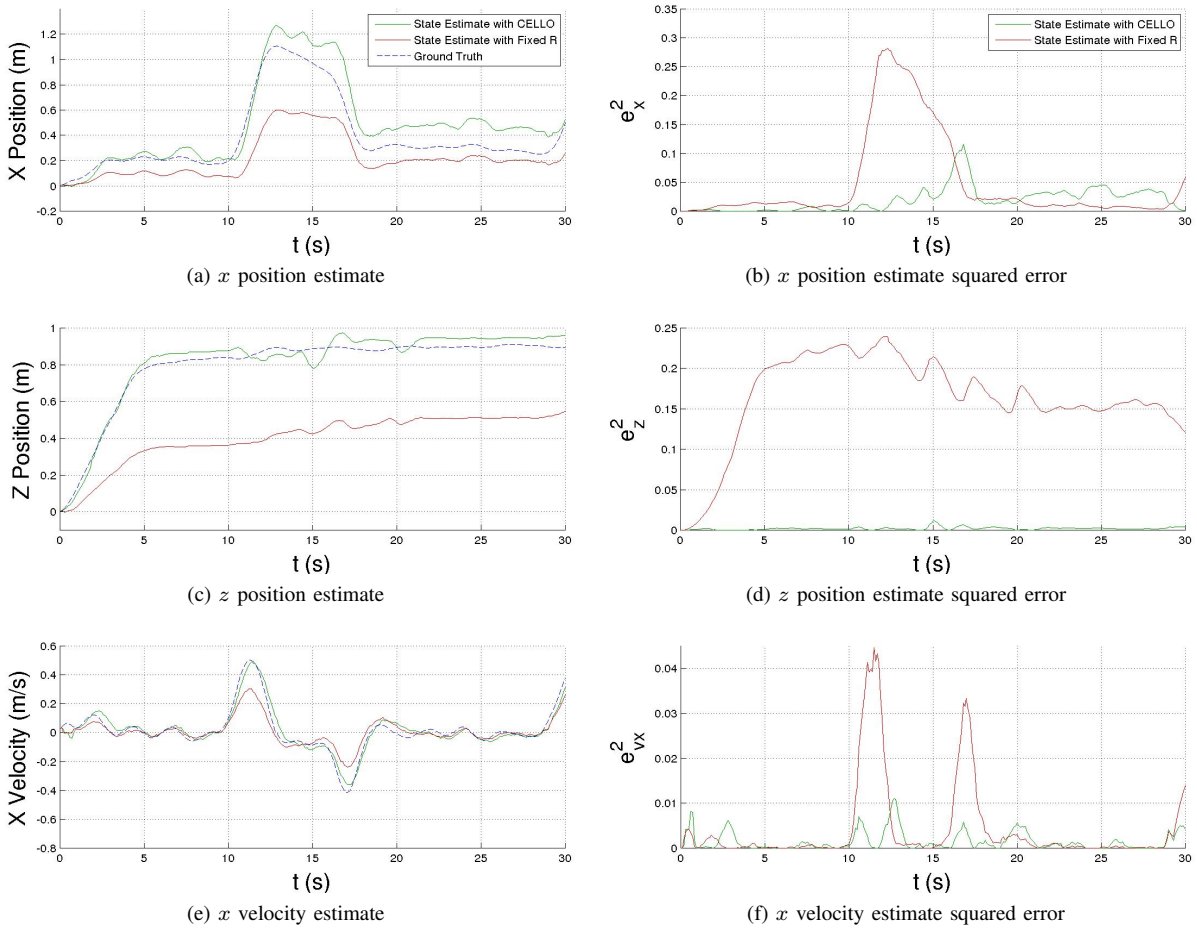
Fig. 4: *Comparison of marginal measurement covariances for the vertical (y) component of flow. Blue dots are sample error vectors; covariances are drawn as 95% confidence margins. The labelled covariances are the empirical covariance of each region; the fixed covariance is the empirical covariance of the entire data set; the CELLO predictions are the predicted covariances for each sample. Note how the learned covariances offer increased flexibility, even within a known region; the regions of higher noise due to takeoff and landing are assigned an appropriately larger covariance.*

motion capture system provides extremely accurate measurements of the vehicle state $\mathbf{x}$, and this state information was used to predict the measurements $\{h(\mathbf{x}_i)\}$. This allows us to form the error vectors $\{\mathbf{e}_i = \mathbf{z}_i - h(\mathbf{x}_i)\}$, yielding a data set of the form required for algorithm 1. The result of the learning is shown in figure (4). These results were attained in a few minutes on a desktop machine, given a data set of approximately ten thousand samples, corresponding to just under ten minutes of flight. In regions where the images are of low quality, the image registration process performs erratically; this is reflected by a covariance orders of magnitude larger than is typical. Typically, such measurements would be rejected by heuristic outlier detectors; such heuristics require careful tuning to avoid discarding useful data while ensuring all invalid points are discarded. CELLO handles this rejection natively, assigning those points a covariance large enough to mitigate any effect they would have on a state estimate, with no tuning or other input from the user required. Additionally, when the vehicle is stopped and the impact of motion blur is completely eradicated, the image matching process becomes very accurate, and the predicted covariance is correspondingly small.

To illustrate the benefits of a predicted covariance scheme, we compare the performance of predicted and labeled (i.e. fixed) covariances in online state estimation using an unscented Kalman filter. The filter incorporates both optical flow and accelerometer data; these sensors are in many ways complementary. The accelerometer operates at a high frequency and with reasonable accuracy; but observes only the derivatives of the system state, and so must be integrated twice, resulting in a large accumulated errors due to drift. The optical flow sensor operates at a much lower frequency, but provides measurements of velocity and height, greatly reducing drift errors. However, the covariance of the optical flow sensor is highly environment-dependent, as seen in figure (4). Choosing a small fixed covariance for the measurement model leads to filter divergence due to singularities in the optical flow measurement function; choosing a large covariance makes the filter slow to incorporate data. Adapting the covariance using the predictions from CELLO allows for the use of smaller covariances only when appropriate, and creates a more robust and accurate filter, as seen in figure (5).

VII. Conclusion

We have presented a method for predicting the covariances of sensor measurements given a vector of predictor functions. This algorithm successfully formed covariances in different regimes as well as a human could by hand-labelling the data; it did so in an automated fashion and without any specific knowledge of what those regimes might be. Furthermore, using these predictions in state estimation demonstrated at least three concrete advantages over using fixed covariances. First, they used available data more efficiently, producing more consistent estimates in the domains tested. Second, the covariances estimated by the filter using adapted measurement covariances better approximated the distribution of estimates in simulation. Third, learned covariances provide an intuitive and parameter-free way of handling outlier rejection in real data, improving the robustness of the state estimator to unmodeled sensor failures. These improvements came without the introduction of any significant overhead in the estimation process. We believe our algorithm represents a significant practical advancement in on-line state estimation, replacing the notoriously difficult process of tuning covariances and heuristic parameters with a simple, automated procedure. Future work includes developing better ways of selecting features, applying the method to modeling the process noise



(a) x position estimate

(b) x position estimate squared error

(c) z position estimate

(d) z position estimate squared error

(e) x velocity estimate

(f) x velocity estimate squared error

Fig. 5: *Online state estimation performance of an unscented Kalman filter using fixed and learned covariances to integrate optical flow and accelerometer data. Only three of twelve states are displayed: the x position, the height above ground z, and the translational velocity in the x direction. Fixed covariances the empirical measurement covariances of the manually annotated sensor regime. The adapted covariances produced by CELLO allow for smaller covariances without introducing inconsistency, precluding the need for outlier rejection and allowing the filter to safely place greater confidence in new data.*

covariance, as well as mitigating the need for ground truth knowledge of state.

VIII. Acknowledgements

This work was supported by ONR under MURI N00014-10-1-0936, MURI N00014-09-1-1052, Science of Autonomy grant N00014-10-1-0936. Their support is gratefully acknowledged.

References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82 (Series D), pp. 35–45, 1960.

[2] M. Athans, R. P. Wishner, and A. Bertolini, "Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements," *IEEE Transactions on Automatic Control*, vol. AC-13, pp. 504–514, 1968.

[3] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, 1997.

[4] M. Brenna, "Scan matching covariance estimation and slam: models and solutions for the scanslam algorithm," Ph.D. dissertation, Artificial Intelligence and Robotics Laboratory Politecnico di Milano, 2009.

[5] R. K. Mehra, "On the identification of variances and adaptive kalman filtering," *IEEE Transactions on Automatic Control*, vol. AC-15, pp. 175–184, 1970.

[6] J. M. Quintana and M. West, "An analysis of international exchange rates using multivariate dlms," vol. 36, pp. 275–281, 1987.

[7] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation," *Econometrica*, vol. 50, pp. 987–1007, 1982.

[8] A. G. Wilson and Z. Ghahramani, "Generalised wishart processes," *Uncertainty in Artificial Intelligence*, 2011.

[9] A. Melkumyan and F. Ramos, "Multi-kernel gaussian processes," *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 1408–1413, 2011.

[10] M. A. Alvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence, "Efficient multioutput gaussian processes through variational inducing kernels," *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 25–32, 2010.

[11] H. Durrant-Whyte, S. Majumder, S. Thrun, M. de Battista, and S. Scheding, "A bayesian algorithm for simultaneous localisation and map building," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, R. Jarvis and A. Zelinsky, Eds. Springer Berlin / Heidelberg, 2003, vol. 6, pp. 49–60.

[12] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proceedings of the IEEE International Conference on Robotics & Automation*, 1999.

[13] F. Lu and E. E. Milios, "Robot pose estimation in unknown environments by matching 2d range scans," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994.