Inference Over Heterogeneous Finite-/Infinite-Dimensional Systems Using Factor Graphs and Gaussian Processes

David M. Rosen, Guoquan Huang, and John J. Leonard

***Abstract*—The ability to reason over partially observable networks of interacting states is a fundamental competency in probabilistic robotics. While the well-known factor graph and Gaussian process models provide flexible and computationally efficient solutions for this inference problem in the special cases in which all of the hidden states are either finite-dimensional parameters or real-valued functions, respectively, in many cases we are interested in reasoning about heterogeneous networks whose hidden states are comprised of both finite-dimensional parameters *and* functions. To that end, in this paper we propose a novel probabilistic generative model that incorporates both factor graphs and Gaussian processes to model these heterogeneous systems. Our model improves upon prior approaches to inference within these networks by removing the assumption of any specific set of conditional independences amongst the modeled states, thereby significantly expanding the class of systems that can be represented. Furthermore, we show that inference within this model can always be performed by means of a two-stage procedure involving inference within a factor graph followed by inference over a Gaussian process; by exploiting fast inference methods for the individual factor graph and Gaussian process models to solve each of these subproblems in succession, we thus obtain a general framework for computationally efficient inference over heterogeneous finite-/infinite-dimensional systems.**

I. Introduction

Many fundamental problems in robotics can be formulated as instances of inference over a network of interacting random states in which the goal is to estimate the values of some subset of the states given noisy observations of others; for example, the canonical problems of filtering, smoothing, localization, and mapping all belong to this class [1]. The development of computationally efficient inference methods for solving problems of this type is thus of significant practical import.

For the common special case in which all of the states are finite-dimensional parameters, factor graphs [2] have proven to be particularly useful: these models generalize and encompass both Bayesian networks and Markov random fields [3] (thus providing a unified theoretical framework for inference), and recent work in the robotics community has led to the development of efficient inference algorithms for these models that can solve problems involving tens of thousands of continuous random variables in real-time on a single processor [4]–[7].

More recently, Gaussian processes [8] have emerged as another useful class of models for the special case in which the hidden states are infinite collections of real values (i.e. real-valued functions); for example, recent work has applied these models to develop generalized Bayes filters [9] and Rauch-Tung-Striebel smoothers [10] in which the entire process and observation models can be estimated nonparametrically directly from data.

However, we are often interested in modeling heterogeneous networks whose states are comprised of both finite-dimensional parameters *and* entire functions (this is the case, for example, when modeling hybrid discrete-/continuous-time systems). While some special cases of this problem have recently been addressed in the robotics literature (e.g. in [9]–[11]), the approaches presented therein are developed for networks that assume the conditional independence of specific subsets of the modeled states (e.g. the hidden Markov models in [9], [10]), and therefore may not generalize to broader classes of networks in which these relations no longer hold. Indeed, to the best of our knowledge, there has been no discussion in the robotics literature of efficient inference methods for heterogeneous finite-/infinite-dimensional systems in the general case.

To that end, in this paper we develop a probabilistic generative model incorporating factor graphs and Gaussian processes to represent generic heterogeneous finite-/infinite-dimensional systems. In contrast to prior work, our model does not assume any specific set of conditional independences amongst the modeled states; the only requirement is that any interaction between states must be mediated by a finite set of finite-dimensional values (a significantly weaker condition). Furthermore, we show that inference within this model can always be performed by means of a two-stage procedure involving inference within a factor graph followed by inference over a Gaussian process; by applying efficient inference methods (based upon recent advances in sparse numerical linear algebra [12]–[14]) for the individual factor graph and Gaussian process models to solve each of these subproblems in succession, we are thus still able to opportunistically exploit whatever conditional independences *do* hold in a particular application (in the form of sparse linear systems, as described in Section II) to achieve fast computation *without* the need to explicitly require these independences in the design of our model itself. Through this approach we thus obtain a flexible and computationally efficient framework for inference over heterogeneous finite-/infinite-dimensional systems in the general case.

II. Review of mathematical preliminaries

In this section we review the formulations of the factor graph and Gaussian process models together with their associated inference algorithms.

The authors are with the Computer Science and Artificial Intelligence Laboratory of the Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Email: {dmrosen,gqhuang,jleonard}@mit.edu.

A. Factor graphs

1) Model formulation: A *factor graph* [2] is a bipartite graph that encodes the factorization of a probability distribution: given a distribution $p \colon \Omega \to \mathbb{R}$ over several variables $\Theta = (\theta_1, \ldots, \theta_n) \in \Omega$ with factorization

$$p(\Theta) = \prod_{i=1}^{m} p_i(\Theta_i), \tag{1}$$

where $\Theta_i \subseteq \{\theta_1, \ldots, \theta_n\}$ for all $1 \leq i \leq m$, the corresponding factor graph $\mathcal{G} = (\mathcal{F}, \Theta, \mathcal{E})$ is:

$$\begin{aligned} \mathcal{F} &= \{p_1, \ldots, p_m\}, \\ \Theta &= \{\theta_1, \ldots, \theta_n\}, \\ \mathcal{E} &= \{(p_i, \theta_j) \mid \theta_j \in \Theta_i\}. \end{aligned} \tag{2}$$

By (2), the *factor nodes* $p_i \in \mathcal{F}$ of $\mathcal{G}$ are in one-to-one correspondence with the factors of p in (1), the *variable nodes* $\theta_j \in \Theta$ are in one-to-one correspondence with the arguments of p, and factor node p_i and variable node θ_j share an edge $e_{ij} = (p_i, \theta_j) \in \mathcal{E}$ if and only if the variable θ_j appears as an argument to factor p_i in (1).

2) Inference in factor graphs: In general, we will be interested in the problem of Bayesian inference: given a hidden parameter Θ, an observable variable Z, and the joint distribution $p(Z, \Theta) = p(Z|\Theta) \cdot p(\Theta)$ relating the two, we wish to obtain the posterior belief $p(\Theta|Z)$ for Θ given a measured value of Z:

$$p(\Theta|Z) = \frac{p(Z, \Theta)}{p(Z)}. \tag{3}$$

Unfortunately, without assuming special structure in the joint distribution $p(Z, \Theta)$ (e.g. prior conjugacy, etc.), the computation of the exact posterior $p(\Theta|Z)$ is generally intractable, since this requires the evaluation of a (possibly very high-dimensional) integral to obtain the evidence $p(Z)$:

$$p(Z) = \int p(Z, \Theta) \, d\Theta. \tag{4}$$

On the other hand, it is often relatively straightforward to obtain the maximum *a posteriori* (MAP) point estimate $\hat{\Theta}_{\text{MAP}}$:

$$\hat{\Theta}_{\text{MAP}} = \operatorname*{argmax}_{\Theta} p(\Theta|Z); \tag{5}$$

for even if $p(Z)$ in (3) is unknown, it is *constant* with respect to Θ, and therefore (by virtue of (3) and (5)),

$$\hat{\Theta}_{\text{MAP}} = \operatorname*{argmax}_{\Theta} \frac{p(Z, \Theta)}{p(Z)} = \operatorname*{argmax}_{\Theta} p(Z, \Theta). \tag{6}$$

Computation of $\hat{\Theta}_{\text{MAP}}$ thus only requires that it be tractable to optimize the joint distribution $p(Z, \Theta)$ as a function of Θ.

Let us assume in the sequel (as is commonly the case in practice) that $p(Z, \Theta)$ is a twice-differentiable probability density function, and that it factors as

$$p(Z, \Theta) = \prod_{i=1}^{m} p_i(Z_i, \Theta_i). \tag{7}$$

Under these conditions, the computational cost of optimizing (6) using Newton-type methods [15] turns out to be determined by the sparsity pattern of the factor graph corresponding to (7). To see this, observe that

$$\begin{aligned} \hat{\Theta}_{\text{MAP}} &= \operatorname*{argmax}_{\Theta} p(Z, \Theta) \\ &= \operatorname*{argmin}_{\Theta} -\ln p(Z, \Theta) \\ &= \operatorname*{argmin}_{\Theta} -\sum_{i=1}^{m} \ln p_i(Z_i, \Theta_i), \end{aligned} \tag{8}$$

so that (by virtue of the final line of (8)) the Hessian

$$H = \frac{\partial^2}{\partial \Theta^2} \left[-\ln p(Z, \Theta) \right] = (H_{jk})_{j,k=1}^{\dim(\Theta)} \tag{9}$$

has $H_{jk} \neq 0$ only if $\theta_j, \theta_k \in \Theta_i$ for some factor p_i in (7), i.e., only if variable nodes θ_j and θ_k are connected to a common factor node p_i in $\mathcal{G}$. The edge set $\mathcal{E}$ of the factor graph $\mathcal{G}$ corresponding to (7) thus directly encodes the sparsity pattern of the Hessian H, and this in turn determines the cost of computing the update step during each iteration of the optimization (8) (cf. e.g. [5]–[7] for details).

After computing the point estimate $\hat{\Theta}_{\text{MAP}}$ in (8), one can additionally recover an approximation for the *entire* posterior $p(\Theta|Z)$ by means of the *Laplace approximation* [16, Sec. 4.4]:

$$\Theta|Z \approx \mathcal{N}\left(\hat{\Theta}_{\text{MAP}}, \Lambda_{\Theta|Z}^{-1}\right) \tag{10a}$$

$$\Lambda_{\Theta|Z} = \frac{\partial^2}{\partial \Theta^2} \left[-\ln p(Z, \Theta) \right] \Bigg|_{\Theta = \hat{\Theta}_{\text{MAP}}}; \tag{10b}$$

this approach approximates the true posterior using a Gaussian distribution that is locally fitted to $p(\Theta|Z)$ at $\hat{\Theta}_{\text{MAP}}$ (more precisely, it fits a second-order Taylor series expansion to $-\ln p(Z, \Theta)$ at $\hat{\Theta}_{\text{MAP}}$). We observe that the information matrix $\Lambda_{\Theta|Z}$ defined in (10b) is simply the Hessian in (9) evaluated at $\hat{\Theta}_{\text{MAP}}$; since Newton-type methods compute this matrix (or an approximation to it) as part of solving the optimization (8), it is thus available at no additional computational cost beyond that needed to compute $\hat{\Theta}_{\text{MAP}}$ itself.

B. Gaussian processes

1) Model formulation: Formally, a *Gaussian process* is a collection of random variables $\{F_x\}_{x \in \mathcal{X}} \subseteq \mathbb{R}^d$ indexed by some (possibly infinite) set $\mathcal{X}$, any finite subset of which have a jointly Gaussian distribution [8], [17].

Since finite-dimensional Gaussian distributions are completely determined by their means and (co)variances, the joint distribution for a finite subset $\{F_{x_i}\}_{i=1}^{n}$ of Gaussian process-distributed random variables can be completely specified by first and second moments of the form $E[F_x]$ and $E[(F_x - E[F_x])(F_{x'} - E[F_{x'}])^T]$ for $x, x' \in \{x_i\}_{i=1}^{n}$. Since these moments exist for any choices of $x, x' \in \mathcal{X}$, we may define functions according to

$$\begin{aligned} &m \colon \mathcal{X} \to \mathbb{R}^d \\ &m(x) = E[F_x] \end{aligned} \tag{11}$$

and

$$\begin{aligned} k&\colon \mathcal{X}\times\mathcal{X}\to\mathbb{R}^{d\times d}\\ k(x,x')&=E\left[(F_x-E[F_x])(F_{x'}-E[F_{x'}])^T\right]\\ &=E\left[(F_x-m(x))(F_{x'}-m(x'))^T\right]; \end{aligned} \tag{12}$$

the functions m and k defined in (11) and (12) are called the *mean function* and *covariance function* for the Gaussian process, respectively. Conversely, given any function $m\colon \mathcal{X}\to\mathbb{R}^d$ and any positive-definite matrix-valued kernel $k\colon \mathcal{X}\times\mathcal{X}\to\mathbb{R}^{d\times d}$, m and k determine (by means of (11) and (12)) a Gaussian process $\{F_x\}_{x\in\mathcal{X}}\subseteq\mathbb{R}^d$ for which they are the mean and covariance functions [8], [17]. We write

$$f\sim\mathcal{GP}(m,k) \tag{13}$$

to denote that $f\triangleq\{F_x\}_{x\in\mathcal{X}}$ is a collection of random variables distributed according to the Gaussian process with mean function m and covariance function k.

Since the Gaussian process (13) assigns to each $x\in\mathcal{X}$ a Gaussian-distributed random value $F_x\in\mathbb{R}^d$, the *entire collection* of random variables $f=\{F_x\}_{x\in\mathcal{X}}$ can be thought of as a *random function* $f\colon\mathcal{X}\to\mathbb{R}^d$. In this view (the so-called *function-space view* [8]), a Gaussian process $\mathcal{GP}(m,k)$ specifies a probability distribution over the *entire set of functions* $\{f\colon\mathcal{X}\to\mathbb{R}^d\}$ whose domain is $\mathcal{X}$. Gaussian processes thus provide a very useful class of priors for performing *nonparametric* regression and interpolation/extrapolation in a Bayesian setting when the true parametric form of the regression function is itself uncertain.

2) Inference in Gaussian processes: When performing inference with Gaussian process models, we will generally be interested in determining the posterior belief for a function with a Gaussian process prior given observations of its values at several points. More precisely, we wish to determine the belief for $\bar{f}\triangleq f|F$, where $f\sim\mathcal{GP}(m,k)$ and

$$\begin{aligned} X&=(x_1,\ldots,x_{n_X})\in\mathcal{X}^{n_X},\\ F&=f(X)=(f(x_1),\ldots,f(x_{n_X}))\in\mathbb{R}^{dn_X} \end{aligned} \tag{14}$$

denote a vector of n_X points in $\mathcal{X}$ and the corresponding vector of f's values at those points, respectively.

First, we observe that in order to recover the posterior belief for $\bar{f}$, it suffices to determine the posterior belief $F_*|F$ for f's values $F_*=f(X_*)$ on some second finite subset of test points $X_*\in\mathcal{X}^{n_{X_*}}$, since we can then obtain $f|F$ from $F_*|F$ by simply taking $X_*=x_*\in\mathcal{X}$ to be a single test point and then allowing it vary pointwise over the entire domain $\mathcal{X}$.

To that end, consider the joint distribution for (F,F_*):

$$\begin{bmatrix}F\\F_*\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}M\\M_*\end{bmatrix},\begin{bmatrix}K(X,X) & K(X,X_*)\\K(X_*,X) & K(X_*,X_*)\end{bmatrix}\right), \tag{15}$$

where

$$M=m(X)=(m(x_1),\ldots,m(x_{n_X}))\in\mathbb{R}^{dn_X}, \tag{16}$$

$M_*=m(X_*)$ analogously, and $K(X,Y)$ denotes the Gram matrix

$$K(X,Y)=\begin{bmatrix}k(x_1,y_1) & \cdots & k(x_1,y_{n_Y})\\ \vdots & & \vdots\\ k(x_{n_X},y_1) & \cdots & k(x_{n_X},y_{n_Y})\end{bmatrix}\in\mathbb{R}^{dn_X\times dn_Y} \tag{17}$$

for two vectors $X\in\mathcal{X}^{n_X}$ and $Y\in\mathcal{X}^{n_Y}$. Since F and F_* are jointly Gaussian-distributed according to (15), the conditional distribution $F_*|F$ is also Gaussian-distributed according to

$$\begin{aligned} F_*|F&\sim\mathcal{N}\left(M_{F_*|F},\Sigma_{F_*|F}\right),\\ M_{F_*|F}&=M_*+K(X_*,X)K_X^{-1}(F-M),\\ \Sigma_{F_*|F}&=K(X_*,X_*)-K(X_*,X)K_X^{-1}K(X,X_*), \end{aligned} \tag{18}$$

where we have introduced the notation $K_X\triangleq K(X,X)$ since this quantity is constant with respect to X_*.

Now since (18) holds for *every* choice of X and X_*, then letting $X_*=x_*\in\mathcal{X}$ be a single point (so that $F_*=f(x_*)$ is just the value of f at x_*), we find that the posterior belief for $\bar{f}$ is again Gaussian process-distributed according to

$$\bar{f}\sim\mathcal{GP}(\bar{m},\bar{k}), \tag{19a}$$

$$\bar{m}(x)=m(x)+k_X(x)K_X^{-1}(F-M), \tag{19b}$$

$$\bar{k}(x,x')=k(x,x')-k_X(x)K_X^{-1}k_X(x')^T, \tag{19c}$$

where k_X is the single-variable function defined by

$$\begin{aligned} k_X&\colon\mathcal{X}\to\mathbb{R}^{d\times dn_X}\\ k_X(x)&=K(x,X) \end{aligned} \tag{20}$$

for fixed X. Gaussian processes thus provide a class of priors on the set of functions $\{f\colon\mathcal{X}\to\mathbb{R}^d\}$ that is closed under posterior updates given observations of the function values $F=f(X)$ on some finite subset of points $X\in\mathcal{X}^{n_X}$.

3) A word on the design of Gaussian process models: As in all kernel-based methods, the covariance (i.e. kernel) function $k(x,x')$ plays a crucial role in Gaussian process models. In this subsection, we show how to characterize several practically-important properties of Gaussian processes (e.g. sample function differentiability class) in terms of easily-ascertained properties of their kernel functions, and provide a few guidelines for the design of kernels in application.

Given $f\sim\mathcal{GP}(m,k)$, equations (19)–(20) show how to obtain the posterior belief for $\bar{f}=f|F$ after incorporating knowledge of f's values $F=f(X)$ on a vector X of inputs. When using this posterior belief to predict the value of f at other (unobserved) inputs, the corresponding point estimator is just the posterior mean $\bar{m}\colon\mathcal{X}\to\mathbb{R}^d$, since

$$\bar{m}(x_*)=E[f(x_*)|f(X)]=\operatorname*{argmax}_{f(x_*)\in\mathbb{R}^d}p(f(x_*)|f(X)) \tag{21}$$

for all $x_*\in\mathcal{X}$ by virtue of (19). Since K_X, F, and M are constant with respect to x_*, equations (17) and (19b) imply that the posterior prediction function $\bar{m}$ is a linear combination of the prior mean function m and terms of the form $k(\cdot,x_i)v_i$, where $v_i\in\mathbb{R}^d$ for $1\le i\le n_X$; in particular, if $m\equiv 0$ (as is commonly assumed in practice), the predictor $\bar{m}$ is just a linear combination of the terms $k(\cdot, x_i)v_i$. Domain-specific knowledge can thus inform the design of the kernel function k so as to obtain a predictor $\bar{m}$ with a parametric form that is well-suited to the prediction task at hand.

The kernel function $k(x, x')$ also serves to define a notion of “similarity” between points $x, x' \in \mathcal{X}$ in the input space, or equivalently, how tightly coupled the function values $f(x)$ and $f(x')$ are. In particular, if the kernel function $k(\cdot, x_i) \colon \mathcal{X} \to \mathbb{R}$ has a neighborhood $N(x_i)$ outside of which $\|k(\cdot, x_i)\|$ is negligibly small, then $f(x_*)$ and $f(x_i)$ are only weakly coupled whenever $x_* \notin N(x_i)$, and in this case equation (19b) shows that f’s value $f(x_i)$ at x_i does not significantly affect the value of the posterior mean prediction $\bar{m}(x_*)$. The choice of a kernel function k with a characteristic spatial scale (e.g. the radial basis kernel) thus gives rise to a Gaussian process whose sample functions f and posterior mean $\bar{m}$ likewise have a characteristic spatial scale over which their values vary (cf. [8, Sec. 4.2]). Thus, if the system being modeled has a characteristic spatial scale, this knowledge can again be incorporated in the design of k.

Furthermore, knowledge of such a characteristic spatial scale can also be exploited in the design of k to reduce the computational cost of performing inference over the resulting Gaussian process model. As shown in (19), computing the posterior mean and covariance functions $\bar{m}$ and $\bar{k}$ requires the evaluation of a matrix-vector or matrix-matrix product with the inverse of the kernel matrix $K_X \in \mathbb{R}^{dn_X \times dn_X}$; in general, these operations are $O(d^3 n_X^3)$, which can quickly become prohibitively expensive as the number of observations n_X increases. However, if k is chosen such that each $k(\cdot, x_i)$ is supported on a compact set whose size is determined by the spatial scale in the modeled system, then many of the elements $k(x_i, x_j)$ in K_X will be zero, i.e. K_X (and likewise $k_X(x)$) will be block sparse; this sparsity can then be exploited to significantly reduce the computational cost of prediction (19).

Finally, for cases in which $\mathcal{X} \subseteq \mathbb{R}^n$, the kernel function also determines the continuity and differentiability classes of the sample functions drawn from a mean-zero Gaussian process. If $f \sim \mathcal{GP}(0, k)$ with $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (so that f is scalar-valued) and there exists $\epsilon > 0$ and $C > 0$ such that

$$E\left[|f(x) - f(y)|^2\right] \leq \frac{C}{|\log\|x - y\||^{1+\epsilon}} \tag{22}$$

for all $x, y \in I$ with $I \subset \mathcal{X}$ a compact set, then f is continuous on I with probability 1; for the common special case in which k is a *stationary* kernel (i.e. in which $k(x, y) = \rho(x - y)$ for some $\rho \colon \mathcal{X} \to \mathbb{R}$), condition (22) simplifies to

$$\rho(0) - \rho(x) \leq \frac{C}{|\log\|x\||^{1+\epsilon}} \tag{23}$$

(cf. [18, pgs. 60–62]). Conditions (22) and (23) can be used to establish vector-valued sample function differentiability up order D as follows. Let $f \sim \mathcal{GP}(0, k)$ with $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{d \times d}$, and write f and k in coordinates as

$$\begin{aligned} f(x) &= (f_1(x), \ldots, f_d(x)) \in \mathbb{R}^d, \\ k(x, x') &= (k_{ij}(x, x'))_{i,j=1}^d \in \mathbb{R}^{d \times d} \\ &= \begin{bmatrix} k_{11}(x, x') & \cdots & k_{1d}(x, x') \\ \vdots & & \vdots \\ k_{d1}(x, x') & \cdots & k_{dd}(x, x') \end{bmatrix}. \end{aligned} \tag{24}$$

If $\frac{\partial^2 k_{ii}}{\partial x_a \partial x'_a}(x, x')$ exists at $(x_*, x_*) \in \mathcal{X} \times \mathcal{X}$, then the mean-square partial derivative $\frac{\partial f_i}{\partial x_a}$ exists at x_* and is jointly mean-zero Gaussian process-distributed with f according to

$$\begin{aligned} E\left[f_i(x) \, \frac{\partial f_j}{\partial x_b}(x')\right] &= \frac{\partial k_{ij}}{\partial x'_b}(x, x') \\ E\left[\frac{\partial f_i}{\partial x_a}(x) \, \frac{\partial f_j}{\partial x_b}(x')\right] &= \frac{\partial k_{ij}}{\partial x_a \partial x'_b}(x, x') \end{aligned} \tag{25}$$

where $1 \leq a, b \leq n$ (cf. [8, Secs. 4.1.1 and 9.4]). By induction, if each of the mixed partial derivatives $\frac{\partial^{\alpha+\beta} k_{ij}}{\partial x^\alpha \partial x'^\beta}$ exists at $(x_*, x_*) \in \mathcal{X} \times \mathcal{X}$ for all multi-indices $\alpha, \beta \in \mathbb{N}^n$ with $0 \leq |\alpha|, |\beta| \leq D$, then f has mean-square mixed partial derivatives $\frac{\partial^\alpha f_i}{\partial x^\alpha}(x_*)$ of all orders up to and including D, and all of these partial derivatives are jointly mean-zero Gaussian process-distributed with f according to

$$E\left[\frac{\partial^\alpha f_i}{\partial x^\alpha}(x) \, \frac{\partial^\beta f_j}{\partial x^\beta}(x')\right] = \frac{\partial^{\alpha+\beta} k_{ij}}{\partial x^\alpha \partial x'^\beta}(x, x'). \tag{26}$$

Equation (26) and conditions (22)–(23) can be used to establish that $f \in C^D(I, \mathbb{R}^d)$ with probability 1 by showing that each of the partial derivatives $\frac{\partial^\alpha f_i}{\partial x^\alpha}$ has continuous sample paths on $I \subset \mathcal{X}$ with probability 1 for all $1 \leq i \leq d$ and all $0 \leq |\alpha| \leq D$ (cf. [19, Sec. 2.5]). Thus, for applications in which sample functions must belong to a certain smoothness class (e.g. in the case of physical mechanical systems obeying Newton’s laws, for which the trajectory is the second integral of the applied forces with respect to time), this knowledge can once again be incorporated into the design of the kernel k. Furthermore, the fact that f and its derivatives $\frac{\partial^\alpha f_i}{\partial x^\alpha}$ are *jointly* Gaussian process-distributed according to (26) allows the integration of observations of f’s derivatives into the inference framework outlined in Section II-B2 whenever such observations are available (cf. [8, Sec. 9.4] and [20]).

Interested readers are encouraged to consult [17], [21] for more information on kernel-based machine learning techniques in general (including an extensive listing of commonly-used kernel functions and methods for constructing new kernels out of old), and [8], [18], [19], [21] for more information on the design of Gaussian process models in particular.

III. Inference over heterogeneous finite-/infinite-dimensional systems

The factor graph and Gaussian process models of Section II provide extremely useful approaches for probabilistic inference over finite-dimensional parameters or entire functions, respectively; in this section, we show how to incorporate both of these models into a framework that enables inference

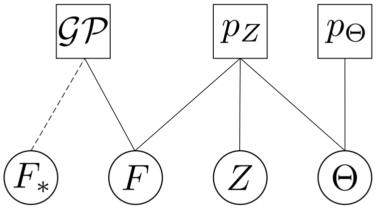


Fig. 1. The factor graph describing the joint distribution of the finite-dimensional parameters in the model (27): here Θ is a hidden parameter with prior $p_\Theta(\cdot)$, $F = f(X)$ is the hidden vector of (random) values of a function $f \sim \mathcal{GP}(m, k)$ evaluated on the input points $X \in \mathcal{X}^{n_X}$, and Z is a vector of observations related to F and Θ through the likelihood (i.e. measurement) model $p_Z(\cdot|F, \Theta)$. We also introduce an additional hidden vector of function values $F_* = f(X_*)$ on the inputs $X_* \in \mathcal{X}^{n_{X_*}}$ that does not directly influence the observation Z, but whose posterior distribution we nevertheless wish to infer using the Gaussian process prior on f.

over heterogeneous systems whose states are comprised of both finite-dimensional parameters *and* entire functions. We begin in Section III-A by introducing a probabilistic generative model to describe these systems, and then derive a set of computationally efficient algorithms for performing (approximate) Bayesian inference within this model in Section III-B.

A. Model formulation

We are interested in performing inference over heterogeneous systems whose hidden states consist of both finite-dimensional parameters *and* functions. To that end, we define the following probabilistic generative model:

$$\begin{aligned} \Theta &\sim p_\Theta(\cdot) \\ f &\sim \mathcal{GP}(m, k) \\ F &= f(X) \\ Z|F, \Theta &\sim p_Z(\cdot|F, \Theta), \end{aligned} \tag{27}$$

where Θ is a finite-dimensional parameter with prior $p_\Theta(\cdot)$, $f \colon \mathcal{X} \to \mathbb{R}^d$ is a function with prior $\mathcal{GP}(m, k)$ taking values $F = f(X)$ on $X \in \mathcal{X}^{n_X}$, and Z is a finite-dimensional vector of observations related to F and Θ through the likelihood (i.e. measurement) model $p_Z(\cdot|F, \Theta)$. The factor graph describing the joint distribution of the finite-dimensional parameters Θ, F, and Z in (27) is shown in Fig. 1.

As can be seen in Fig. 1, (27) models the observation Z as arising from the interaction of a finite-dimensional hidden state Θ and a finite set of values F of a hidden function f. It does not enforce any conditional independence relations between Θ, F, and Z (equivalently, it does not require that $p(\Theta)$, $p(F)$, or $p(Z|F, \Theta)$ admit any nontrivial factorizations), and therefore suffices to model *any* measurement arising from *any* interaction amongst (finite- or infinite-dimensional) hidden states that is mediated by a finite-dimensional set of values, as claimed. Finally, we point out that although (27) does constrain the observation Z to be finite-dimensional, this is not actually a limitation in practice, as physical sensors are only capable of collecting finitely many measurements.

B. Inference

In this section we derive inference methods for computing the joint posterior distribution $p(f, \Theta|Z)$ and the marginals $p(f|Z)$ and $p(\Theta|Z)$ in the model (27).

To begin, we observe that (by the same logic as was used in Section II-B2) performing inference over the entire function f is equivalent to replacing f with $F_* = f(X_*)$ and then allowing $X_* = x_* \in \mathcal{X}$ to vary pointwise over all of $\mathcal{X}$. To that end, we consider the joint posterior $p(F_*, F, \Theta|Z)$:

$$\begin{aligned} p(F_*, F, \Theta|Z) &= p(F_*|F, \Theta, Z) \cdot p(F, \Theta|Z) \\ &= p(F_*|F) \cdot p(F, \Theta|Z), \end{aligned} \tag{28}$$

where the first equality follows from the chain rule of probability and the second from the fact that $F_* \perp\!\!\!\perp (Z, \Theta)|F$ (cf. Fig. 1). The first distribution $p(F_*|F)$ in (28) comes directly from the Gaussian process prior on f and is given in closed form by (18); the second $p(F, \Theta|Z)$ can be approximated by applying the Laplace approximation (10) to the subgraph in Fig. 1 determined by the solid edges (as described in Section II-A2) to produce:

$$\begin{bmatrix} F \\ \Theta \end{bmatrix} |Z \sim \mathcal{N}\left(\begin{bmatrix} \mu_{\bar{F}} \\ \mu_{\bar{\Theta}} \end{bmatrix}, \begin{bmatrix} \Sigma_{\bar{F}\bar{F}} & \Sigma_{\bar{F}\bar{\Theta}} \\ \Sigma_{\bar{\Theta}\bar{F}} & \Sigma_{\bar{\Theta}\bar{\Theta}} \end{bmatrix} \right). \tag{29}$$

We can further decompose this distribution as

$$p(F, \Theta|Z) = p(F|\Theta, Z) \cdot p(\Theta|Z), \tag{30}$$

and by virtue of (29) we have

$$\Theta|Z \sim \mathcal{N}\left(\mu_{\bar{\Theta}}, \Sigma_{\bar{\Theta}\bar{\Theta}}\right) \tag{31}$$

and

$$\begin{aligned} F|\Theta, Z &\sim \mathcal{N}\left(\mu_{\bar{F}|\bar{\Theta}}, \Sigma_{\bar{F}|\bar{\Theta}}\right), \\ \mu_{\bar{F}|\bar{\Theta}} &= \mu_{\bar{F}} + \Sigma_{\bar{F}\bar{\Theta}} \Sigma_{\bar{\Theta}\bar{\Theta}}^{-1} \left(\Theta - \mu_{\bar{\Theta}}\right), \\ \Sigma_{\bar{F}|\bar{\Theta}} &= \Sigma_{\bar{F}\bar{F}} - \Sigma_{\bar{F}\bar{\Theta}} \Sigma_{\bar{\Theta}\bar{\Theta}}^{-1} \Sigma_{\bar{\Theta}\bar{F}}. \end{aligned} \tag{32}$$

Now, we can obtain the joint distribution $p(F_*, \Theta|Z)$ from $p(F_*, F, \Theta|Z)$ by marginalizing F. By virtue of (28) and (30), this can be written as:

$$\begin{aligned} p(F_*, \Theta|Z) &= \int p(F_*, F, \Theta|Z)\, dF \\ &= p(\Theta|Z) \int p(F_*|F) \cdot p(F|\Theta, Z)\, dF. \end{aligned} \tag{33}$$

We observe that the distributions $p(F_*|F)$ and $p(F|\Theta, Z)$ in the integrand in the final line of (33) can be interpreted as a (Gaussian) likelihood for F_* given F and a (Gaussian) prior for F, respectively, so that the entire integral simply represents the marginal distribution for F_* given Θ and Z:

$$p(F_*|\Theta, Z) = \int p(F_*|F) \cdot p(F|\Theta, Z)\, dF. \tag{34}$$

In general, given the Gaussian prior and likelihood models:

$$\begin{aligned} x &\sim \mathcal{N}(\mu_x, \Sigma_x) \\ y|x &\sim \mathcal{N}\left(Ax + b, \Sigma_{y|x}\right), \end{aligned} \tag{35}$$

the marginal distribution of y is:

$$\begin{aligned} y &\sim \mathcal{N}(\mu_y, \Sigma_y), \\ \mu_y &= A\mu_x + b, \\ \Sigma_y &= \Sigma_{y|x} + A\Sigma_x A^T \end{aligned} \tag{36}$$

(cf. e.g. [16, Sec. 2.3.3]). Equations (18), (32), and (34)–(36) together imply that the distribution $p(F_* | \Theta, Z)$ in (34) is given in closed form by:

$$\begin{aligned} F_* | \Theta, Z &\sim \mathcal{N}\left(\mu_{\bar{F}_* | \bar{\Theta}}, \Sigma_{\bar{F}_* | \bar{\Theta}}\right), \\ \mu_{\bar{F}_* | \bar{\Theta}} &= M(X_*) + K(X_*, X) K_X^{-1} \left(\mu_{\bar{F} | \bar{\Theta}} - M\right) \\ \Sigma_{\bar{F}_* | \bar{\Theta}} &= K(X_*, X_*) - K(X_*, X) K_X^{-1} K(X, X_*) \\ &\quad + K(X_*, X) K_X^{-1} \Sigma_{\bar{F} | \bar{\Theta}} K_X^{-1} K(X, X_*). \end{aligned} \tag{37}$$

Finally, (33), (34), and (37) in turn imply that the joint posterior distribution $p(f, \Theta | Z)$ we seek is given by:

$$p(f, \Theta | Z) = p(f | \Theta, Z) \cdot p(\Theta | Z), \tag{38}$$

where $p(\Theta | Z)$ is given by (29) and (31), and

$$\begin{aligned} f | \Theta, Z &\sim \mathcal{GP}\left(m_{\bar{f} | \bar{\Theta}},\ k_{\bar{f} | \bar{\Theta}}\right) \\ m_{\bar{f} | \bar{\Theta}}(x) &= m(x) + k_X(x) K_X^{-1} (\mu_{\bar{F}} + \Sigma_{\bar{F}\bar{\Theta}} \Sigma_{\bar{\Theta}\bar{\Theta}}^{-1} (\Theta - \mu_{\bar{\Theta}})) \\ k_{\bar{f} | \bar{\Theta}}(x, x') &= k(x, x') + k_X(x) K_X^{-1} k_X(x')^T \\ &\quad + k_X(x) K_X^{-1} \Sigma_{\bar{F} | \bar{\Theta}} K_X^{-1} k_X(x')^T. \end{aligned} \tag{39}$$

Now it remains only to determine the marginal distribution $p(f | Z)$ (the marginal distribution $p(\Theta | Z)$ having already been determined in (29) and (31)). We observe that

$$p(F_* | Z) = \int p(F_* | F) \cdot p(F | Z)\, dF \tag{40}$$

with $p(F_* | F)$ given by (18) and $F | Z \sim \mathcal{N}\left(\mu_{\bar{F}}, \Sigma_{\bar{F}\bar{F}}\right)$ by (29). A second application of equations (35) then shows that

$$\begin{aligned} F_* | Z &\sim \mathcal{N}\left(\mu_{\bar{F}_*}, \Sigma_{\bar{F}_*}\right), \\ \mu_{\bar{F}_*} &= M(X_*) + K(X_*, X) K_X^{-1} (\mu_{\bar{F}} - M), \\ \Sigma_{\bar{F}_*} &= K(X_*, X_*) - K(X_*, X) K_X^{-1} K(X, X_*) \\ &\quad + K(X_*, X) K_X^{-1} \Sigma_{\bar{F}\bar{F}} K_X^{-1} K(X, X_*). \end{aligned} \tag{41}$$

Thus, the marginal posterior distribution $p(f | Z)$ is given by:

$$\begin{aligned} f | Z &\sim \mathcal{GP}(m_{\bar{f}}, k_{\bar{f}}), \\ m_{\bar{f}}(x) &= m(x) + k_X(x) K_X^{-1} (\mu_{\bar{F}} - M), \\ k_{\bar{f}}(x, x') &= k(x, x') - k_X(x) K_X^{-1} k_X(x')^T \\ &\quad + k_X(x) K_X^{-1} \Sigma_{\bar{F}\bar{F}} K_X^{-1} k_X(x')^T. \end{aligned} \tag{42}$$

Equations (29) and (31), (38)–(39), and (42) admit the computation of $p(f, \Theta | Z)$ and its marginals using a two-stage inference procedure: first we compute the Laplace approximation for $(F, \Theta) | Z$ in (29) by applying the method of Section II-A2 to the factor graph determined by the solid edges in Fig. 1, and then we recover $p(f, \Theta | Z)$ or $p(f | Z)$ by fusing $p(F, \Theta | Z)$ with the conditional distribution for $f | F$ induced by the Gaussian process prior over f using (38)–(39) or (42), respectively.

Finally, we observe that although our algorithmic development has thus far involved only a single function f, parameter Θ, and observation Z, the fact that all of these may be vector-valued implies that this procedure immediately generalizes to incorporate any number of functions $f_1, \ldots, f_{n_f}$, parameters $\Theta_1, \ldots, \Theta_{n_\Theta}$ and observations $Z_1, \ldots, Z_{n_Z}$ by simply defining $f = (f_1, \ldots, f_{n_f})$, $\Theta = (\Theta_1, \ldots, \Theta_{n_\Theta})$, and $Z = (Z_1, \ldots, Z_{n_Z})$. Any conditional independence relationships that hold amongst these variables can subsequently be exploited in the form of factor graph sparsity (when performing inference over the finite-dimensional parameters F, Θ, and Z) or block sparsity of kernel matrices (when inferring the posterior distributions for $\bar{f} = (\bar{f}_1, \ldots, \bar{f}_{n_f})$). As we will see in the next section, the exploitation of sparsity enables fast inference over networks of the form (27) containing hundreds or thousands of continuous random variables.

IV. An example application

In this section we demonstrate the application of the inference framework developed in Section III using a toy target-tracking example; specifically, we consider a novel hybrid discrete-/continuous-time formulation of the cooperative localization and target tracking (CLATT) problem. For ease of exposition, in this demonstration we will consider only a single mobile robot and a single target; however, the inference algorithm that we derive immediately generalizes to arbitrary numbers of robots and targets following the argument given at the end of Section III-B.

To that end, we consider a single mobile robot attempting to track a single target, both moving in the plane. We model the robot pose at time t_i as $s_i = (x_i^r, y_i^r, \theta_i^r)$, where $(x_i^r, y_i^r) \in \mathbb{R}^2$ is the robot's position in the plane and $\theta_i^r \in (-\pi, \pi]$ its heading angle. We assume that the robot is equipped with a proprioceptive sensor (e.g. an inertial measurement unit) that enables it to estimate its ego-motion $\Delta s_{i,i+1}$ between two subsequent poses s_i and s_{i+1} according to:

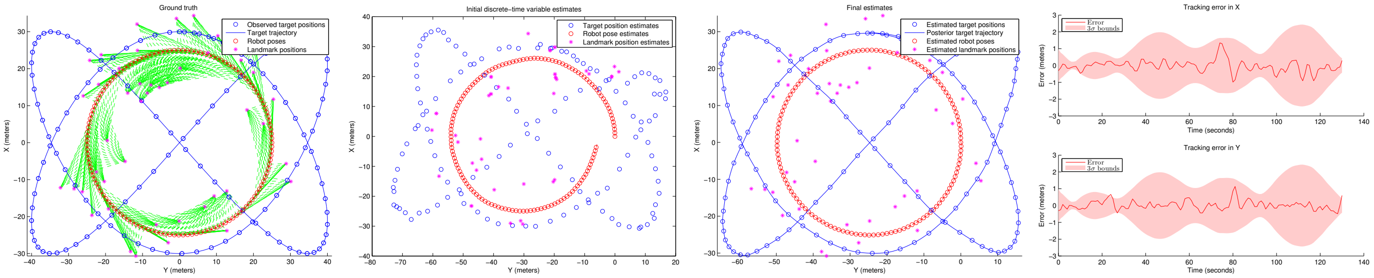
$$\Delta s_{i,i+1} = \begin{bmatrix} (x_{i+1}^r - x_i^r)\cos(\theta_i^r) + (y_{i+1}^r - y_i^r)\sin(\theta_i^r) \\ (x_{i+1}^r - x_i^r)\sin(\theta_i^r) - (y_{i+1}^r - y_i^r)\cos(\theta_i^r) \\ \theta_{i+1}^r - \theta_i^r \end{bmatrix}, \tag{43}$$

and that these measurements are subject to mean-zero additive Gaussian noise with a standard deviation of .03 meters in each translational direction and 1 degree in rotation.

To prevent the accumulation of drift in its own state estimate, the robot also estimates the positions of, and relocalizes itself with respect to, any landmarks that it discovers as it moves through its environment (more precisely, it performs smoothing SLAM [1]). For this purpose, we assume that the robot is equipped with a sensor that enables it to measure the relative range and bearing $m_{i,j} = (r_{i,j}, \theta_{i,j})$ from its current pose s_i to a landmark at position $l_j = (x_j, y_j)$:

$$\begin{aligned} \Delta x_{i,j} &= x_j - x_i^r, \\ \Delta y_{i,j} &= y_j - y_i^r, \\ r_{i,j} &= \sqrt{\Delta x_{i,j}^2 + \Delta y_{i,j}^2} \\ \theta_{i,j} &= \arctan(\Delta y_{i,j}, \Delta x_{i,j}) - \theta_i^r. \end{aligned} \tag{44}$$

We assume that the measurements (44) are subject to zero-mean additive Gaussian noise with a standard deviation of .10



(a) Ground truth

(b) Initial discrete-time estimates

(c) Final estimates

(d) Tracking errors

Fig. 2. Tracking a target with a mobile robot. (a): Ground truth for this experiment, showing the robot poses (red circles), landmark locations (magenta asterisks), and the target's continuous-time trajectory (blue curve) and positions at which it was observed (blue circles). Landmark observations are indicated by a dashed green line connecting the landmark and the robot pose from which it was observed. (b): The initial estimate of the discrete-time variables (robot poses, observed target positions, and landmark locations) used to initialize the numerical optimization to compute the MAP estimate as described in Section II-A2. (c): The final MAP estimates for the discrete-time variables and the posterior marginal estimate for the entire target trajectory obtained as described in Section III-B. (d): The globally-registered target tracking errors in x and y, together with the 3σ confidence bounds reported by the inference method.

meters in range and 1 degree in bearing, and that the sensor has a 180-degree forward-facing field of view and a maximum range of 20 meters.

Finally, we assume that the robot is also equipped with a sensor that enables it to measure the relative range and bearing from its current pose to the tracked target according to (44). For the sake of simplicity, in this example we will assume that this sensor is *always* able to observe the target, independent of its position relative to the robot, and that it is likewise subject to zero-mean additive Gaussian noise with a .10 meter standard deviation in range and a 1 degree standard deviation in bearing.

A discrete-time formulation suffices to model the robot state in this problem because we assume direct access to odometry measurements; these observations form a "backbone" of pose-to-pose constraints that (in combination with the landmark observations) enables smoothing over the robot's trajectory. Unfortunately, direct odometric measurements are generally not available for the target in target-tracking applications. In practice, it is common to replace odometry measurements with constraints derived from an *assumed* target motion model (e.g. constant-velocity models); however, it is not always clear *a priori* how to select an appropriate parametric class for such a model (particularly when the target is highly maneuverable or unpredictable), which renders this approach vulnerable to model misspecification.

Bearing these considerations in mind, in this example we propose to model the target position as an unknown continuous-time function $f\colon \mathbb{R} \to \mathbb{R}^2$ with a Gaussian process prior; this avoids the necessity of specifying a particular parametric form for the target motion model (hence also the model misspecification problem) while still enabling smoothing over the target's observed positions by enforcing smoothness in the target's estimated trajectory through an appropriate design of the covariance function k (as described in Section II-B3). To that end, we suppose that $f \sim \mathcal{GP}(0, k_f)$, where

$$\begin{aligned} k_f &\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}^2 \\ k_f(t, t') &= k_{pp1,2}\left(\frac{|t - t'|}{l}\right) \begin{bmatrix} \sigma_f^2 & 0 \\ 0 & \sigma_f^2 \end{bmatrix}, \end{aligned} \tag{45}$$

and $k_{pp1,2}\colon \mathbb{R} \to \mathbb{R}$ is the piecewise-polynomial radial basis kernel defined in [8, pg. 88]. This kernel has several desirable properties for this application, including stationarity, compact support on intervals of length l (corresponding to the length of the "sliding window" over which we wish to smooth the target path, and thus the sparsity of the kernel matrix K_X), a maximum (co)variance magnitude of σ_f^2 (specifying the strength of the applied smoothing), and first-order sample function differentiability (thus enforcing the condition that the target trajectory should be at least first-order differentiable, as we should expect for any physical system obeying Newton's laws). Based on prior knowledge of the target's maneuverability characteristics, in this example we will take $\sigma_f = 5$ meters and $l = 10$ seconds.

The experimental setup is shown in Fig. 2(a). The simulated environment is an 80×60 meter grid containing 45 randomly distributed landmarks. The robot traverses a single counterclockwise loop of radius 25 meters through this environment at a constant speed of 1.2 m/s, while the tracked target follows a Lissajous curve at speeds between 1.7 and 6.0 m/s (with an average speed of 4.1 m/s). The robot measures the target position, its own ego-motion relative to its prior pose, and any nearby landmarks once every second, for a duration of 130 seconds. The entire simulation thus comprises 45 landmarks (with 745 observations), 131 robot poses (with 130 odometry measurements connecting them), and 131 target positions (each with one measurement).

This raw data was batch-processed using the two-stage procedure outlined in Section III-B. First, an initial estimate for the discrete-time variables (the robot poses, landmark positions, and measured target positions) was obtained by integrating the robot odometry measurements and initializing the landmarks and target positions relative to the robot pose from which they were first observed (Fig. 2(b)). This estimate was then used as the initialization for an iterative numerical optimization to compute the MAP estimate and the Laplace approximation to the posterior distribution of the discrete-time variables as outlined in Section II-A2. Optimization was performed using MATLAB's `lsqnonlin` with the `trust-region-reflective` method (requiring 11 iterations and 3.37 seconds); the resulting MAP estimate $\hat{\Theta}_{\text{MAP}}$ and the Hessian approximation $H(\hat{\Theta}_{\text{MAP}}) \approx 2J^T(\hat{\Theta}_{\text{MAP}})J(\hat{\Theta}_{\text{MAP}})$ were then used to approximate the posterior distribution according to (10). Finally, this approximate posterior distribution for the discrete-time variables was used to compute the marginal posterior distribution for the unknown target path $f(t)$ using (42). The final estimates are shown in Fig. 2(c).

To evaluate the performance of this method, we computed the least-squares-optimal registration between the robot's coordinate system and the global coordinate system based upon aligning the landmark position estimates (as would be done in practice when localizing the robot with respect to the global reference frame), and then computed the target tracking error as the difference between the robot's globally registered estimate and the ground truth; results are shown in Fig. 2(d). In this experiment we observed median tracking errors of $-.018$ and $-.016$ meters in x and y, respectively, and a total RMS error of $.44$ meters. Comparison of Figs. 2(a) and 2(d) reveals that most of this error arises when tracking the target through the aggressive turns at the corners of the environment; furthermore, while the tracking error may be somewhat high in these regions, the posterior 3σ confidence bounds produced by the inference method correctly capture the greater uncertainty in these sections of the estimate, thereby preserving consistency. This demonstrates the feasibility of the proposed nonparametric hybrid discrete-/continuous-time approach to the CLATT problem.

V. Conclusion

In this paper we developed a probabilistic generative model and an associated set of inference algorithms for reasoning over general heterogeneous finite-/infinite-dimensional systems. Our approach generalizes prior work by relaxing the requirement that a specific set of conditional independences amongst the modeled states must hold, yet is nevertheless still able to opportunistically exploit whatever conditional independences *do* obtain in a particular application to achieve fast computation. Through this approach we thus obtain a flexible and computationally efficient framework for inference over heterogeneous finite-/infinite-dimensional systems in the general case.

In closing, we remark that while the inference framework developed herein has been formulated in the batch setting, we believe it should be possible to adapt this approach to the online case by applying incremental methods (e.g. [5]–[7]) to efficiently solve the sequences of individual factor graph and Gaussian process subproblems. We intend to investigate this possibility in future research.

Acknowledgements

This work was partially supported by the Office of Naval Research under grants N00014-10-1-0936, N00014-11-1-0688 and N00014-13-1-0588 and by the National Science Foundation under grant IIS-1318392, which we gratefully acknowledge.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: The MIT Press, 2008.

[2] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[3] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: The MIT Press, 2009.

[4] M. Kaess, V. Ila, R. Roberts, and F. Dellaert, "The Bayes tree: An algorithmic foundation for probabilistic robot mapping," in *Intl. Workshop on the Algorithmic Foundations of Robotics, WAFR*, Singapore, Dec. 2010.

[5] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. of Robotics Research*, vol. 31, no. 2, pp. 216–235, Feb. 2012.

[6] D. Rosen, M. Kaess, and J. Leonard, "An incremental trust-region method for robust online sparse least-squares estimation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, St. Paul, MN, May 2012, pp. 1262–1269.

[7] ——, "Robust incremental online inference over sparse factor graphs: Beyond the Gaussian case," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013, pp. 1017–1024.

[8] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press, 2006.

[9] J. Ko and D. Fox, "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, Jul. 2009.

[10] M. Deisenroth, R. Turner, M. Huber, U. Hanebeck, and C. Rasmussen, "Robust filtering and smoothing with Gaussian processes," *IEEE Trans. on Automatic Control*, vol. 57, no. 7, pp. 1865–1871, Jul. 2012.

[11] C. Tong, P. Furgale, and T. Barfoot, "Gaussian process Gauss-Newton for non-parametric simultaneous localization and mapping," *Intl. J. of Robotics Research*, vol. 32, no. 5, pp. 507–525, May 2013.

[12] P. Matstoms, "Sparse QR factorization in MATLAB," *ACM Trans. Math. Softw.*, vol. 20, no. 1, pp. 136–159, Mar. 1994.

[13] T. Davis, J. Gilbert, S. Larimore, and E. Ng, "A column approximate minimum degree ordering algorithm," *ACM Trans. Math. Softw.*, vol. 30, no. 3, pp. 353–376, Sep. 2004.

[14] Y. Chen, T. Davis, W. Hager, and S. Rajamanickam, "Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate," *ACM Trans. Math. Softw.*, vol. 35, no. 3, pp. 22:1–22:14, Oct. 2008.

[15] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. New York: Springer Science+Business Media, 2006.

[16] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer Science+Business Media, 2006.

[17] M. Álvarez, L. Rosasco, and N. Lawrence, "Kernels for vector-valued functions: A review," Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Tech. Rep. MIT-CSAIL-TR-2011-033, CBCL-301, Jun. 2011.

[18] R. Adler, *The Geometry of Random Fields*. Philadelphia: The Society for Industrial and Applied Mathematics (SIAM), 2010.

[19] C. Paciorek, "Nonstationary Gaussian processes for regression and spatial modeling," Ph.D. dissertation, Carnegie Mellon University, 2003.

[20] E. Solak, R. Murray-Smith, W. Leithead, D. Leith, and C. Rasmussen, "Derivative observations in Gaussian process models of dynamic systems," in *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2003, pp. 1033–1040.

[21] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: The MIT Press, 2002.