2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Rumor-robust Decentralized Gaussian Process Learning, Fusion, and Planning for Modeling Multiple Moving Targets

Chang Liu[1] *Member, IEEE*, Zhihao Liao[2], and Silvia Ferrari[3] *Senior Member, IEEE*

***Abstract*— This paper presents a decentralized Gaussian Process (GP) learning, fusion, and planning (RESIN) formalism for mobile sensor networks to actively learn target motion models. RESIN is characterized by both computational and communication efficiency, and the robustness to rumor propagation in sensor networks. By using the weighted exponential product rule and the Chernoff information, a rumor-robust decentralized GP fusion approach is developed to generate a globally consistent target trajectory prediction from local GP models. A decentralized information-driven path planning approach is then proposed for mobile sensors to generate informative sensing paths. A novel, constant-sized information sharing strategy is developed for sensing path coordination, and an analytical objective function is derived that significantly reduces the computational complexity of the path planning. The effectiveness of RESIN is demonstrated in simulations.**

I. INTRODUCTION

The problem of learning the behavior and dynamics of moving targets via mobile sensor networks has received significant attention in recent years because of important applications such as environmental monitoring [1], [2], security and surveillance [3], [4], and the internet of things [5]. Bayesian nonparametric (BNP) models, such as Gaussian Processes (GPs) and Dirichlet Process Gaussian Processes (DPGPs), have been shown very effective at modeling moving targets because of their flexibility, expressiveness, and data-driven nature [6]. Unlike traditional, model-based approaches, GPs require little prior information about target behavior, and are applicable when the number of targets of interest change over time, for example, as new targets enter and old targets leave the ROI [4], [7]. As a result, GP models provide a more flexible and systematic approach for modeling moving targets when compared to semi-Markov jump systems [8], linear stochastic models [9], and physics-based models [10], [11]. Early works on BNP sensor network control relied on centralized learning, data fusion and planning [12]. In many applications, however, contested communication and GPS-denied environments prevent centralized methods from performing robustly and reliably. This is because the central station or fusion agent may be unable to gather information and/or convey plans to all sensors consistently over time. This paper presents a decentralized BNP learning, data fusion, and planning formalism characterized by easiness for parallelization, scalability, and robustness to single-point failure compared to the centralized counterparts.

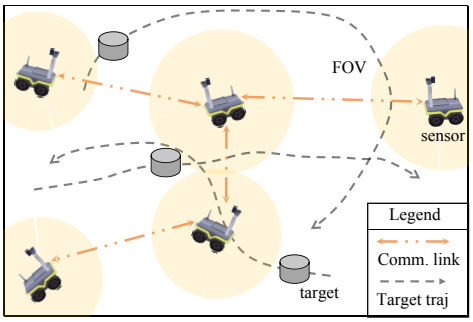

Fig. 1: Sensors share information between neighbors and coordinate their trajectories to actively learn the GP motion models of multiple moving targets.

Several decentralized GP learning and fusion approaches have been previously developed to distribute computations among independent local agents operating on subsets of the data. Two representative classes of methods include the mixture of experts (MoEs) [13] and the product of experts (PoEs) [14]. In MoEs, each agent locally learns a GP model for a different partition of the state space and the global prediction is made by collecting all of agents' local predictions. In contrast, in PoEs, agents share the same state space and each agent locally learns an independent GP model using a subset of training data. Then, the global prediction is made by the Bayes rule and the independence assumption of local predictions. PoE methods allow for efficient training and prediction and thus have attracted great interest. However, current PoE approaches cannot be directly applied to data fusion in sensor networks since PoE-based approaches cannot handle rumor propagation. This means common information between local agents, such as the simultaneous measurements of the same target, may be redundantly used and leads to incorrect fusion results [15]. This paper presents a "rumor-robust" PoE-based approach for fusing GP models such that rumor propagation is prevented.

A decentralized information-driven path planning (IPP) approach is also presented for controlling and coordinating sensor trajectories so as to obtain the most informative target measurements subject to communication constraints. Previous methods for decentralized IPP include a decentralized, gradient-based control approach that assumes all-to-all agent communications [16]. This method usually incurs large communication burden since the convergence of gradient-based optimization requires multiple iterations. More re-

[1]Chang Liu is a Software Engineer at Nvidia Corporation. He was a Postdoctoral Associate at Sibley School of Mechanical and Aerospace Engineering at Cornell University. Email: changliu@berkeley.edu.

[2]Zhihao Liao is a Software Engineer at TuSimple. He was a Master student at Sibley School of Mechanical and Aerospace Engineering at Cornell University. Email: zl673@cornell.edu.

[3]Silvia Ferrari is the John Brancaccio Professor of Mechanical and Aerospace Engineering at Cornell University, Ithaca, NY USA. Email: sf375@cornell.edu.

978-1-7281-7447-1/20/$31.00 ©2020 IEEE

cently, [17] proposes a multi-robot online sensing strategy for the construction of communication maps using GP. The work uses a leader-follower paradigm to coordinate with its follower. However, no coordination is ensured between pairs.

To overcome these challenges, this paper develops a rumor-robust decentralized GP learning, fusion, and planning (RESIN[1]) approach for mobile sensor networks to actively learn target motion models. To deal with time-varying target motion models, a spatio-temporal kernel is used in GP modeling. A rumor-robust decentralized GP learning and fusion algorithm is then proposed and applied to combine individual sensors' local prediction of target trajectories into a globally consistent one. The GP learning and fusion approach is computationally efficient and can avoid rumor propagation in the sensor network. We subsequently present a sequential optimal control approach that is efficient both in computation and communication for decentralized sensor path planning. In particular, an analytical objective function is derived via the use of decentralized GP fusion, which reduces the original mixed integer nonlinear programming problem into a low-dimensional nonlinear programming problem. Besides, a new information sharing strategy is proposed for coordination between sensors, which only requires a constant-sized communication overhead. In contrast, prior works [18], [19] require a communication overhead that is linear in the number of sensors.

II. Problem Formulation

Consider a network of N mobile sensors deployed to learn the motion models of M targets moving across a connected, compact, and non-empty workspace $\mathcal{W} \subset \mathbb{R}^2$ (Figure 1). Each sensor is equipped with a fixed stereo-camera and a wireless communication device. The number of targets is unknown a priori and can change over time due to, for example, targets entering or exiting the workspace. The motion model of each target, indexed by i, is represented by a time-varying, continuous, and differentiable function $\mathbf{f}_i : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ defined over $\mathcal{W}$, which maps the target position to its velocity, i.e.,

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}_i\left[\mathbf{x}_i(t), t\right] \triangleq \mathbf{v}_i(t), \tag{1}$$

where $\mathbf{x}_i(t) \in \mathcal{W}$ and $\mathbf{v}_i(t) \in \mathbb{R}^2$ represent the position and velocity of the target's center of mass, respectively.

The sensor's state, defined as $\mathbf{s} = [s_x \quad s_y \quad s_\theta \quad s_v]^T \in \mathbb{R}^4$, includes the sensor position $[s_x \quad s_y]^T \in \mathcal{W}$, orientation $s_\theta \in [0, 2\pi)$, and velocity $s_v \geq 0$. The sensor control input, defined as $\mathbf{u} = [a \quad \omega]^T \in \mathbb{R}^2$, includes the linear acceleration, $a \in \mathbb{R}$, and angular velocity, $\omega \in \mathbb{R}$. Let $\Delta T > 0$ represent the discretization interval so that the kth step corresponds to $t = k\Delta T$. The jth sensor's kinematic model can be represented by the following difference equation,

$$\mathbf{s}_j(k+1) = \mathbf{s}_j(k) + \begin{bmatrix} s_v(k)\cos(s_\theta(k)) \\ s_v(k)\sin(s_\theta(k)) \\ \mathbf{u}_j(k) \end{bmatrix} \Delta T \tag{2}$$

[1]RESIN is the acronym of "Rumor-robust decentralized gp lEarning, fuSIon, and planNing".

For simplicity, in the rest of the paper, we refer to the kinematic model (2) as $\mathbf{g} : \mathbb{R}^4 \times \mathbb{R}^2 \to \mathbb{R}^4$ such that

$$\mathbf{s}_j(k+1) = \mathbf{g}(\mathbf{s}_j(k), \mathbf{u}_j(k)). \tag{3}$$

Each sensor measures the target position and velocity by computing the sparse scene flow of the target. Camera frames containing targets are obtained when the targets are inside the sensor's FOV, defined as $\mathcal{F}\left(\mathbf{s}_j(k)\right) = \left\{\mathbf{w} \in \mathcal{W} \mid \left\|[s_x(k) \quad s_y(k)]^T - \mathbf{w}\right\|_2 \leq r_j\right\}$, where $r_j > 0$ denotes the jth sensor's sensing range. The camera obeys the following measurement model with additive Gaussian noise,

$$\mathbf{z}_{ij}(k) = \mathbf{v}_i(k) + \boldsymbol{\varepsilon}, \quad \text{if } \mathbf{x}_i(k) \in \mathcal{F}\left(\mathbf{s}_j(k)\right) \tag{4}$$

where $\mathbf{z}_{ij}(k) \in \mathbb{R}^2$ is the jth sensor's measured velocity of ith target, and $\boldsymbol{\varepsilon} \in \mathbb{R}^2$ is a zero-mean Gaussian white noise and it follows the distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\varepsilon)$, where $\boldsymbol{\Sigma}_\varepsilon = \epsilon_0^2\mathbf{I}$. Here we assume perfect data association between targets and sensor measurements, as the data association problem is out of the scope of this paper.

The sensors form a communication network where each sensor can constantly communicate with its neighboring sensors. Without loss of generality, it is assumed that the communication network forms a connected graph such that a tree-structured communication path exists at each time step.

A. Decentralized Learning and Fusion

In order to learn the target motion models, sensors must accurately predict target positions and actively decide their sensing trajectories to reduce the uncertainty in the target states estimates. Each sensor first locally learns a GP model based on its own sensor measurements to predict the targets' future trajectories and then fuses the local prediction into a global one via communicating with neighboring sensors. A typical issue in decentralized fusion is the rumor propagation, where the common information in sensors' local data are double counted [15]. Tracking and removing the common information is needed to avoid rumor propagation, but is usually computationally heavy. This paper proposes a decentralized GP fusion approach to combine local predictions to generate globally consistent prediction of target trajectories while avoiding double counting, as described in Section III.

B. Information-driven Path Planning (IPP) Algorithm

The IPP can be formulated as an optimal control problem. Let $\boldsymbol{u}_j\left(k : k_f\right) = \left[\boldsymbol{u}_j^T\left(k\right) \quad \ldots \quad \boldsymbol{u}_j^T\left(k_f\right)\right]^T$ represent the planned control inputs of jth sensor over the planning interval $[k, k_f]$ and $\boldsymbol{U}\left(k : k_f\right) = \left[\boldsymbol{u}_1\left(k : k_f\right) \quad \ldots \quad \boldsymbol{u}_N\left(k : k_f\right)\right]$ denote the control inputs of all N sensors. The optimal control of all sensors, $\boldsymbol{U}^*\left(k : k_f\right)$, is computed by maximizing the objective function $J\left(\boldsymbol{U}\left(k : k_f\right)\right)$ under system constraints, formulated as the following optimization problem,

$$\begin{aligned} \boldsymbol{U}^*\left(k : k_f\right) &= \arg\max_{\boldsymbol{U}(k:k_f)} J\left(\boldsymbol{U}\left(k : k_f\right)\right) \\ s.t.\ \mathbf{s}_j(\tau+1) &= \mathbf{g}\left(\mathbf{s}_j(\tau), \mathbf{u}_i(\tau)\right), \mathbf{s}_j(\tau) \in \mathcal{S},\ \mathbf{u}_j(\tau) \in \mathcal{U} \\ &\tau = k, \ldots, k_f - 1,\ j = 1, \ldots, N, \end{aligned} \tag{5}$$

where $\mathcal{S}$ and $\mathcal{U}$ represent the feasible set of sensor state and control input, respectively.

In this work, the objective function uses the mutual information (MI), which has been shown very effective for information-driven path planning [4], [16], [20]. In particular, define $\mathbf{X}(k:k_f) = [\mathbf{x}_1(k:k_f) \quad \dots \quad \mathbf{x}_m(k:k_f)]$ as the predicted target positions in the planning interval, which are obtained using decentralized GP fusion. Also define $\mathbf{Z}(k:k_f)$ as the predicted measurements of these targets from all sensors, then $J(\boldsymbol{U}(k:k_f))$ is defined as the conditional MI between $\mathbf{X}(k:k_f)$ and $\mathbf{Z}(k:k_f)$, given existing sensor measurements, i.e.,

$$J(\boldsymbol{U}(k:k_f)) = I(\mathbf{X}(k:k_f); \mathbf{Z}(k:k_f) \mid \mathbf{Z}(1:k-1)). \tag{6}$$

Solving the centralized IPP (5) is in general computationally expensive due to the exponentially growing search space with respect to the sensor number and the planning horizon. This paper propose a decentralized IPP algorithm to distribute the computation among sensors to efficiently solve the problem, as presented in Section IV.

III. Decentralized GP Learning and Fusion for Target Prediction

This works utilizes GP to model the velocity field of each target, with the target position and time being the input and the corresponding target velocity being the output of GP. In particular, let $\mathbf{X}_i(k) = [\mathbf{x}_i(1) \quad \dots \quad \mathbf{x}_i(k)]$ and $\mathbf{Z}_{ij}(k) = [\mathbf{z}_{ij}(1) \quad \dots \quad \mathbf{z}_{ij}(k)]$ represent the measured positions and velocities of ith target by the jth sensor at time steps 1 to k. Then jth sensor's local GP model can predict the velocity at time $\tau > 0$ at a query position $\boldsymbol{\xi} \in \mathcal{W}$, and the predicted value $\mathbf{z}_{ij,\boldsymbol{\xi}}(\tau)$ obeys a Gaussian distribution $\mathbf{z}_{ij,\boldsymbol{\xi}}(\tau) \sim \mathcal{N}\left(\boldsymbol{\mu}_{ij}(\boldsymbol{\xi}), \Sigma_{ij}(\boldsymbol{\xi})\right)$, where the mean $\boldsymbol{\mu}_{ij}(\boldsymbol{\xi})$ and covariance $\Sigma_{ij}(\boldsymbol{\xi})$ are [6]

$$\begin{aligned}\boldsymbol{\mu}_{ij}(\boldsymbol{\xi}) &= \mathbf{K}(\boldsymbol{\xi}, \mathbf{X})\left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \epsilon_0^2 \mathbf{I}\right)^{-1} \mathbf{Z} \\ \Sigma_{ij}(\boldsymbol{\xi}) &= \mathbf{K}(\boldsymbol{\xi}, \boldsymbol{\xi}) - \mathbf{K}(\boldsymbol{\xi}, \mathbf{X})\left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \epsilon_0^2 \mathbf{I}\right)^{-1} \mathbf{K}(\mathbf{X}, \boldsymbol{\xi}).\end{aligned} \tag{7}$$

The kernel matrix $\mathbf{K}(\cdot, \cdot)$ is the key component of GP and it encodes the similarity between input data points. This paper uses the following spatial-temporal Radial Basis functions as the spatio-temporal kernel to account for the time-varying nature of the motion model,

$$K(\mathbf{x}_i(t_i), \mathbf{x}_j(t_j)) = \sigma_s^2 e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2l_x^2}} e^{-\frac{(t_i - t_j)^2}{2l_\tau^2}},$$

where l_x and l_τ represent the spatial and temporal length scale, and σ_s is the hyperparameter for signal variance.

A. Local GP Learning and Prediction of Target Trajectory

Given the measurements of sensor j, the hyper-parameters of the local GP can be learned by maximizing the logarithm of the marginal likelihood function of the training data [6]. The resultant GP model is then used to predict the target positions in the planning interval $[k, k_f]$. Using the Bayes rule, the pdf of predicted target positions can be represented as follows [21],

$$\begin{aligned}&P_j(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k)) = \\ &\prod_{\tau=k}^{k_f-1} \int_{\mathbb{R}^2} \left[P(\mathbf{x}_i(\tau+1) \mid \mathbf{v}_i(\tau), \mathbf{x}_i(\tau)) P_j(\mathbf{v}_i(\tau) \mid \mathbf{x}_i(\tau))\right] d\mathbf{v}_i(\tau)\end{aligned} \tag{8}$$

where $P_j(\mathbf{v}_i(\tau) \mid \mathbf{x}_i(\tau))$ corresponds to jth sensor's local GP model and $P(\mathbf{x}_i(\tau+1) \mid \mathbf{v}_i(\tau), \mathbf{x}_i(\tau))$ can be obtained from the target motion model (1).

In general, there is no analytical form for $P_j(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k))$ when $k_f - k \geq 2$. To make the prediction tractable, we define a *nominal path* that is obtained by assuming that the target moves with the mean velocity given by GP and then approximate $P_j(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k))$ along the nominal path. In particular, define the sequence of nominal positions as $\hat{\mathbf{X}}_{ij}(k+1:k_f) = [\hat{\mathbf{x}}_{ij}(k+1) \quad \dots \quad \hat{\mathbf{x}}_{ij}(k_f)]$ where $\hat{\mathbf{x}}_{ij}(\tau+1) = \boldsymbol{\mu}_{ij}(\hat{\mathbf{x}}_{ij}(\tau))\Delta T + \hat{\mathbf{x}}_{ij}(\tau)$, $\tau = k, \dots, k_f - 1$, with the initial condition $\hat{\mathbf{x}}_{ij}(k) = \mathbf{x}_i(k)$. The velocity term $\boldsymbol{\mu}_{ij}(\hat{\mathbf{x}}_{ij}(\tau))$ is the mean vector computed using (7). Then the pdf of the predicted trajectory can be approximated as follows [21],

$$\begin{aligned}&P_j(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k)) \\ &\approx \prod_{\tau=k}^{k_f-1} P_j\left(\mathbf{v}_i(\tau) = \frac{\mathbf{x}_i(\tau+1) - \hat{\mathbf{x}}_{ij}(\tau)}{\Delta T} \mid \hat{\mathbf{x}}_{ij}(\tau)\right)\end{aligned} \tag{9}$$

where the equality (9) is obtained from the motion model (1). Equations (9) indicates that the pdf of the predicted target trajectory can be approximated as the pdf of predicted velocities along the nominal path, and the resultant pdf is a product of Gaussian distributions. By simple algebraic manipulation, it can be shown that (9) is actually a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{ij,loc}, \boldsymbol{\Sigma}_{ij,loc})$, where the mean vector $\boldsymbol{\mu}_{ij,loc}$ and the covariance matrix $\boldsymbol{\Sigma}_{ij,loc}$ is

$$\begin{aligned}\boldsymbol{\mu}_{ij,loc} &= \left[\hat{\mathbf{x}}_{ij}^T(k+1) \quad \dots \quad \hat{\mathbf{x}}_{ij}^T(k_f)\right]^T, \\ \boldsymbol{\Sigma}_{ij,loc} &= diag\left[\boldsymbol{\Sigma}_{ij}(\hat{\mathbf{x}}_{ij}(k+1)) \quad \dots \quad \boldsymbol{\Sigma}_{ij}(\hat{\mathbf{x}}_{ij}(k_f))\right],\end{aligned}$$

where $diag$ means the block diagonal matrix. It is easy to see the mean is the vector of nominal positions.

B. Decentralized Target Trajectory Fusion and Prediction

To coordinate sensing paths, it is important for sensors to fuse their local prediction to obtain a global consensus on targets' predicted trajectories. This subsection proposes a rumor-robust decentralized GP fusion approach. Consider the fusion of ith target's prediction from sensors j and l, where the pdfs of local prediction are $P_j(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k))$ and $P_l(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k))$, computed using (9). The proposed fusion rule is as follows,

$$\begin{aligned}P(\mathbf{X}_i(k+1:k_f)|\mathbf{X}_i(k)) \propto P_j^{\beta_j w^*}(\mathbf{X}_i(k+1:k_f)|\mathbf{X}_i(k)) \\ P_l^{\beta_l(1-w^*)}(\mathbf{X}_i(k+1:k_f)|\mathbf{X}_i(k)),\end{aligned} \tag{10}$$

where β_j and β_l are weighting factors that indicate each sensor's contribution to the combined prediction. Following the strategy in [14], β_j and β_l are chosen as the difference

in differential entropy between the prior and the posterior at $\mathbf{X}_i(k+1:k_f)$ to ensure that the more information an agent contains about the ith target's prediction, the more it contributes to the combined prediction. Using the fact that for a Gaussian distribution $P(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, its exponential is also a Gaussian distribution with a scaled covariance matrix [14], i.e. $P^{\alpha}(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \alpha^{-1}\boldsymbol{\Sigma})$, the predictive mean and covariance matrix of $P\left(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k)\right)$ are

$$\begin{aligned}\boldsymbol{\mu}_i^* &= \boldsymbol{\Sigma}_i^* \left(\beta_j w^* \boldsymbol{\Sigma}_{ij,loc}^{-1}\boldsymbol{\mu}_{ij,loc} + \beta_l(1-w^*)\boldsymbol{\Sigma}_{il,loc}^{-1}\boldsymbol{\mu}_{il,loc}\right)\\ \boldsymbol{\Sigma}_i^* &= \left(\beta_j w^* \boldsymbol{\Sigma}_{ij,loc}^{-1} + \beta_l(1-w^*)\boldsymbol{\Sigma}_{il,loc}^{-1}\right)^{-1}\end{aligned} \tag{11}$$

where $\left(\boldsymbol{\mu}_{ij,loc}, \boldsymbol{\Sigma}_{ij,loc}\right)$ and $\left(\boldsymbol{\mu}_{il,loc}, \boldsymbol{\Sigma}_{il,loc}\right)$ are the mean and covariance pairs of sensor j and l 's local GP prediction.

The use of the weighted exponential product w^* and $(1-w^*)$ in (10) has been effective to remove rumor propagation [22]. To compute the optimal fusion weight w^*, this work uses the Chernoff information as the metric for measuring distance between pdfs [22]. For two arbitrary pdfs $P_a(\mathbf{x})$ and $P_b(\mathbf{x})$, the optimal Chernoff weight is obtained by minimizing their Chernoff information, i.e.,

$$w^* = \arg\max_{w\in[0,1]} -\log \int \left[P_a(\mathbf{x})\right]^w \left[P_b(\mathbf{x})\right]^{1-w} d\mathbf{x}.$$

The combined pdf is then $P(\mathbf{x}) = \left[P_a(\mathbf{x})\right]^{w^*}\left[P_b(\mathbf{x})\right]^{1-w^*}$.

The main difficulty of using Chernoff weight for information fusion is that for genenral distributions, there is usually no analytic expression for their Chernoff information, thus computing the optimal weight causes significant computational overhead [22]. However, the Chernoff information for two multivariate Gaussian distributions, $P_a(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a)$ and $P_b(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b)$, can be expressed in the closed form [23], and the optimal Chernoff weight can be computed as follows,

$$\begin{aligned}w^* = \arg\min_{w\in[0,1]} &\frac{1}{2}\log\frac{|w\boldsymbol{\Sigma}_a + (1-w)\boldsymbol{\Sigma}_b|}{|\boldsymbol{\Sigma}_a|^w|\boldsymbol{\Sigma}_b|^{1-w}}\\ &+ \frac{w(1-w)}{2}\left(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\right)^T\left(w\boldsymbol{\Sigma}_a + (1-w)\boldsymbol{\Sigma}_b\right)\left(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\right)\end{aligned} \tag{12}$$

The optimal Chernoff weight, w^*, in (11) can therefore be computed between the following two Gaussian distributions,

$$\begin{aligned}P^{\beta_j}\left(\mathbf{X}_{ij}(k+1:k_f) \mid \mathbf{X}_{ij}(k)\right) &\sim \mathcal{N}(\boldsymbol{\mu}_{ij,loc}, \beta_j^{-1}\boldsymbol{\Sigma}_{ij,loc}),\\ P^{\beta_l}\left(\mathbf{X}_{il}(k+1:k_f) \mid \mathbf{X}_{il}(k)\right) &\sim \mathcal{N}(\boldsymbol{\mu}_{il,loc}, \beta_l^{-1}\boldsymbol{\Sigma}_{il,loc}),\end{aligned}$$

using nonlinear optimization algorithms. Using (11) and (12) for each pair of sensors along the tree-structured communication network, the rumor-robust decentralized GP fusion can be conducted efficiently. The fused prediction at the root sensor is propagated back to all sensors such that sensors have the same fused pdf, which we refer to as $P_{fuse}\left(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k)\right)$ for the ith target.

IV. Decentralized Sensor Planning

This section presents the decentralized IPP algorithm used in RESIN, which uses the sequential planning strategy [18], [19]. In the sequential planning, given a planning order, each sensor first receives the planning information from its predecessors (Section IV-A), then it computes its own optimal path (Section IV-B), and sends the new planning information to the next sensor in the sequence. Without loss of generality, we assume the planning order corresponds to sensors' indices in following analysis. The decentralized IPP algorithm is characterized by efficiency in both communication (Section IV-A) and computation (Section IV-B).

A. Fusing Predecessors' Plans

For the jth sensor, given the planned paths of the first $j-1$ sensors, $\mathbf{S}_{j-1}(k:k_f) = [\mathbf{s}_1(k:k_f) \quad \ldots \quad \mathbf{s}_{j-1}(k:k_f)]$, the local planning problem becomes

$$\begin{aligned}\mathbf{u}_j^*(k:k_f) = \arg\max_{\mathbf{u}_j(k:k_f)} &J_j\left(\mathbf{u}_j(k:k_f); \mathbf{S}_{j-1}(k:k_f)\right)\\ s.t.\ &\mathbf{s}_j(\tau+1) = \mathbf{g}\left(\mathbf{s}_j(\tau), \mathbf{u}_j(\tau)\right),\\ &\mathbf{s}_j(\tau) \in \mathcal{S},\ \mathbf{u}_j(\tau) \in \mathcal{U},\ \tau = k, \ldots, k_f\end{aligned} \tag{13}$$

The objective function is defined as the mutual information between target prediction and jth sensor's planned path, conditioned on the first $j-1$ sensors' plans, i.e.,

$$\begin{aligned}&J_j\left(\mathbf{u}_j(k:k_f); \mathbf{S}_{j-1}(k:k_f)\right) =\\ &\quad I\left(\mathbf{X}(k+1:k_f); \mathbf{z}_j(k+1:k_f) \mid \mathbf{Z}_{j-1}(k+1:k_f)\right)\end{aligned}$$

where $\mathbf{Z}_{j-1}(k+1:k_f)$ represents the predicted measurements of the first $j-1$ sensors' planned paths.

In order to fuse predecessor sensors' predicted measurements, the globally fused target prediction, $P_{fuse}\left(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k)\right)$, is treated as the prior distribution of targets' prediction. Similar to Section III-A, the Bayesian fusion approach is used to compute the posterior distribution conditioned on $\mathbf{Z}_{j-1}(k+1:k_f)$, and it can be shown that [21],

$$\begin{aligned}&P_{j,pre}\left(\mathbf{X}_i(k+1:k_f) \mid \mathbf{Z}_{j-1}(k+1:k_f)\right)\\ &\approx \prod_{\tau=k}^{k_f-1} P\left(\mathbf{v}_i(\tau) = \frac{\mathbf{x}_i(\tau+1) - \hat{\mathbf{x}}_i(\tau)}{\Delta T}\right)\prod_{l=1}^{j-1} P\left(\mathbf{z}_l(\tau) \mid \hat{\mathbf{x}}_i(\tau)\right)\end{aligned} \tag{14}$$

The predicted measurement from lth sensor, $\mathbf{z}_l(\tau)$, is assumed to be nonempty if the nominal position $\hat{\mathbf{x}}_i(\tau)$ lies in the sensor's FOV at τ.

The prior $P\left(\mathbf{v}_i(\tau) = \frac{\mathbf{x}_i(\tau+1) - \hat{\mathbf{x}}_i(\tau)}{\Delta T}\right)$ can be directly obtained by marginalizing $P_{fuse}\left(\mathbf{X}_i(k+1:k_f) \mid \mathbf{X}_i(k)\right)$ over all time steps except τ, and can be easily shown to be a Gaussian distribution, denoted as $\mathcal{N}(\boldsymbol{\mu}_{i,fuse}(\tau), \boldsymbol{\Sigma}_{i,fuse}(\tau))$. Since the measurement model is linear Gaussian, an analytical expression of $P_{j,pre}\left(\mathbf{X}_i(k+1:k_f) \mid \mathbf{Z}_{j-1}(k+1:k_f)\right)$ can be obtained. In particular, let $\mathbb{I}\{\hat{\mathbf{x}}_i(\tau) \in \mathcal{F}\left(\mathbf{s}_l(\tau)\right)\}$ represent the indicator function and it equals 1 if and only if the the nomial position $\hat{\mathbf{x}}_i(\tau)$ lies in the sensor's planned FOV at τ. Then it can be derived, using the conjugacy property of Gaussian prior and likelihood functions, that given the prior covariance matrix $\boldsymbol{\Sigma}_{i,fuse}(\tau)$ and let $n(\tau) = \sum_{l=1}^{j-1}\mathbb{I}\{\hat{\mathbf{x}}_i(\tau) \in \mathcal{F}\left(\mathbf{s}_l(\tau)\right)\}$ represent the number

of sensors in the first $j-1$ sensors that can measure the ith target at time τ, then the posterior covariance is

$$\boldsymbol{\Sigma}_{ij,pre}(\tau) = \left(\boldsymbol{\Sigma}_{i,fuse}^{-1}(\tau) + n(\tau)\boldsymbol{\Sigma}_{\varepsilon}^{-1}\right)^{-1}, \tau = k{+}1, \ldots, k_f.$$

Therefore, the fused pdf can be compactly represented as

$$P_{j,pre}\left(\mathbf{X}_i\left(k+1:k_f\right) \mid \mathbf{Z}_{j-1}(k+1:k_f)\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_{ij,pre}(k+1:k_f), \boldsymbol{\Sigma}_{ij,pre}(k+1:k_f)\right),$$

where the covariance matrix $\boldsymbol{\Sigma}_{ij,pre}(k+1:k_f)$ is

$$\boldsymbol{\Sigma}_{ij,pre}\left(k+1:k_f\right) = diag\left[\boldsymbol{\Sigma}_{ij,pre}(k+1) \quad \ldots \quad \boldsymbol{\Sigma}_{ij,pre}(k_f)\right].$$

Since the objective function only depends on covariance matrix, we ignore the expression of mean $\boldsymbol{\mu}_{ij,pre}(k{+}1:k_f)$.

The key observation here is that, to compute $\boldsymbol{\Sigma}_{ij,pre}\left(k+1:k_f\right)$, the only information needed from predecessor sensors is the times that each sensor can expect to detect the target in the planning interval, i.e., $n(t)$. Since all sensors share the same nominal path of the target, $\hat{\mathbf{X}}_i\left(k:k_f\right)$, thanks to the decentralized GP fusion, each sensor only needs to receive the counting from its predecessor, then add its own measurement times to the total counting, and send the updated counting to the next sensor. The communication overhead between each pair of sensors is therefore constant and independent of the number of predecessor sensors. In contrast, the transmitted information to each sensor in state-of-the-art works [18], [19] is the planned paths from all predecessors, which incurs a communication burden of $O(N)$.

B. Local Objective Function

The fused pdf $P_{j,pre}\left(\mathbf{X}_i\left(k:k_f\right) \mid \mathbf{Z}_{j-1}(k:k_f)\right)$ is now used as the prior pdf for jth sensor's path planning. Given the jth sensor's future control inputs $\mathbf{u}_j\left(k:k_f\right)$ and the consequent future measurements $\mathbf{z}_j\left(k:k_f\right)$, the posterior pdf can be obtained [21], i.e.,

$$P_{j,plan}\left(\mathbf{X}_i\left(k+1:k_f\right) \mid \mathbf{z}_j\left(k+1:k_f\right), \mathbf{Z}_{j-1}(k+1:k_f)\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_{ij,plan}(k+1:k_f), \boldsymbol{\Sigma}_{ij,plan}(k+1:k_f)\right),$$

where the covariance matrix is

$$\boldsymbol{\Sigma}_{ij,plan}\left(k+1:k_f\right) = diag\left[\boldsymbol{\Sigma}_{ij,plan}(k+1) \quad \ldots \quad \boldsymbol{\Sigma}_{ij,plan}(k_f)\right].$$

Again, using the conjugacy of Gaussian distribution, the covariance matrix of the posterior pdf can be computed in a closed-form, i.e., for $\tau = k+1, \ldots, k_f$,

$$\boldsymbol{\Sigma}_{ij,plan}(\tau) = \left(\boldsymbol{\Sigma}_{ij,pre}^{-1}(\tau) + \mathbb{I}\{\hat{\mathbf{x}}_i(\tau) \in \mathcal{F}\left(\mathbf{s}_j(\tau)\right)\}\boldsymbol{\Sigma}_{\varepsilon}^{-1}\right)^{-1}. \tag{15}$$

Now we derive the closed-form of the objective function. It it easy to show that [21],

$$J_j\left(\mathbf{u}_j\left(k:k_f\right) \mid \mathbf{S}_{j-1}(k:k_f)\right) = \sum_{i=1}^{M}\sum_{\tau=k}^{k_f}\frac{1}{2}\log\det\frac{\boldsymbol{\Sigma}_{ij,pre}(\tau)}{\boldsymbol{\Sigma}_{ij,plan}(\tau)}.$$

The indicator function in (15) makes the IPP problem a mixed integer nonlinear programming problem, which is notoriously difficult to solve. To overcome this difficulty, we consider an approximate objective function, where the indicator function is replaced by the constant 1 and the following weighting factor is added to the MI at each step,

$$\psi(\tau) = \max\left(0, 1 - \frac{\left(\|[s_{j,x}(\tau), s_{j,y}(\tau)]^T - \hat{\mathbf{x}}_i\left(\tau\right)\|_2 - \frac{r_j}{2}\right)^2}{\left(\frac{r_j}{2}\right)^2}\right).$$

Then the objective function becomes

$$J_j = \sum_{i=1}^{M}\sum_{\tau=k}^{k_f}\frac{\psi(\tau)}{2}\log\det\frac{\boldsymbol{\Sigma}_{ij,pre}(\tau)}{\tilde{\boldsymbol{\Sigma}}_{ij,plan}(\tau)} \tag{16}$$

where $\tilde{\boldsymbol{\Sigma}}_{ij,plan}(\tau) = \left(\boldsymbol{\Sigma}_{ij,pre}^{-1}(\tau) + \boldsymbol{\Sigma}_{\varepsilon}^{-1}\right)^{-1}$. The decentralized IPP problem (13) can be efficiently solved using nonlinear optimization algorithms.

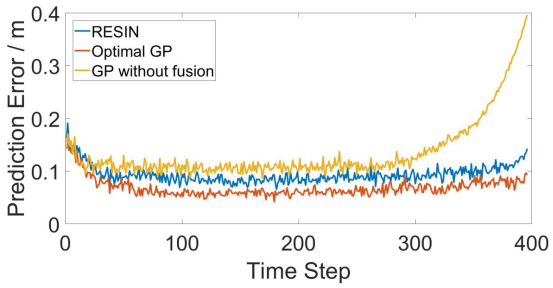

Fig. 2: Average prediction error of targets using different GP fusion strategies.

V. Simulation Setup and Results

Two simulations are conducted to evaluate the effectiveness of RESIN. The first simulation evaluates the decentralized GP learning and fusion in RESIN where stationary sensors are used to avoid the influence of planning algorithms. The second simulation considers mobile sensors and evaluates the full pipeline in RESIN.

A. Evaluating Decentralized GP Learning and Fusion

In this simulation, four stationary sensors are randomly placed in a $10m \times 10m$ workspace. A total of eight targets enter the workspace, each of which has a different trajectory. The sensor's sensing range is $r_j = 5m$. Decentralized GP learning and fusion in RESIN is compared to the centralized GP and the GP without fusion methods. In the centralized GP method, all sensors share their measurements with other sensors. For GP without fusion method, sensors do not communicate with each other and therefore the local GP learning and prediction only uses sensor's local measurements.

Figure 2 compares the average prediction error of all sensors' predicted trajectory of every target in the planning interval (five steps). As expected, the centralized GP has the minimum prediction error among all three approaches, as the sensors have direct access to all measurements and therefore are able to make the most accurate prediction. RESIN has a slightly larger prediction error than the centralized GP fusion. In contrast, GP without fusion leads to the worst prediction error since each sensor makes prediction only based on its own measurements. The simulation result shows that RESIN is an effective fusion approach that achieves similar performance as that of the centralized GP.

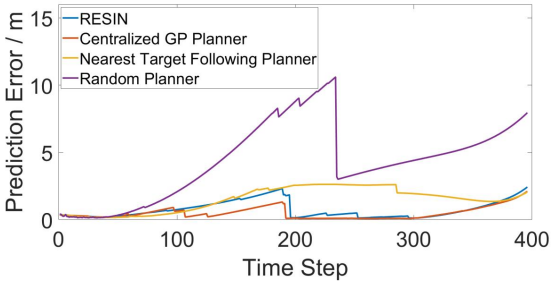

Fig. 3: Average prediction error of targets using different fusion and planning strategies.

B. Evaluating RESIN

In this simulation, four mobile sensors are randomly placed in a $30m \times 30m$ workspace. There is a total of eight moving targets and each target moves in a different pattern. The velocity of each sensor is bounded in the range of $[0, 3](m/s)$ and the control input of each robot is also bounded, defined as follows,

$$\left[-\frac{\pi}{6}, -5(m/s)\right]^T \leq \mathbf{u} \leq \left[\frac{\pi}{6}, 5(m/s)\right]^T.$$

RESIN is compared to three benchmark methods, including the centralized GP planner, nearest target following planner, and the random planner. In the centralized GP planner, the GP learning, prediction, and planning are conducted in a centralized way, where all sensors' measurements are shared, and the planning is conducted for all sensors simultaneously. The nearest target following planner drives a sensor to pursue the closest target based on its locally estimated target position. The random planner generates random control inputs.

Figure 3 compares the performance of these four planners, in which the average prediction error of all sensors' predicted trajectory of every target under different planning strategies are quantitatively compared. RESIN outperforms both the nearest target following planner and the random planner in general, and has very similar performance as that of the centralized GP planner.

VI. Conclusion

This paper proposes RESIN for sensor networks to actively learn GP motion models of moving targets. Characterized by the computational and communication efficiency, and the robustness to rumor propagation, RESIN is a powerful framework for mobile sensor networks. The future work will consider the data association in the decentralized fusion and also develop the parallel implementation of RESIN.

Acknowledgments

This work is supported by ONR grant N00014-17-1-2175.

References

[1] A. Singh, F. Ramos, H. D. Whyte, and W. J. Kaiser, "Modeling and decision making in spatio-temporal processes for environmental surveillance," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 5490–5497.

[2] C. Liu, S. E. Li, and J. K. Hedrick, "Measurement dissemination-based distributed bayesian filter using the latest-in-and-full-out exchange protocol for networked unmanned vehicles," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 11, pp. 8756–8766, 2017.

[3] C. Liu, S. E. Li, D. Yang, and J. K. Hedrick, "Distributed bayesian filter using measurement dissemination for multiple unmanned ground vehicles with dynamically changing interaction topologies," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 3, p. 030903, 2018.

[4] H. Wei, P. Zhu, M. Liu, J. P. How, and S. Ferrari, "Automatic pan-tilt camera control for learning dirichlet process gaussian process (dpgp) mixture models of multiple moving targets," *IEEE Transactions on Automatic Control*, 2018.

[5] S. Hong, D. Kim, M. Ha, S. Bae, S. J. Park, W. Jung, and J.-E. Kim, "Snail: an ip-based wireless sensor network approach to the internet of things," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34–42, 2010.

[6] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, 1 2006.

[7] C. Liu, Y. Chen, J. Gemerek, H. Yang, and S. Ferrari, "Learning recursive bayesian nonparametric modeling of moving targets via mobile decentralized sensors," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8034–8040.

[8] W. Lu, G. Zhang, S. Ferrari, R. Fierro, and I. Palunko, "An information potential approach for tracking and surveilling multiple moving targets using mobile sensor agents," in *Unmanned Systems Technology XIII*, vol. 8045. International Society for Optics and Photonics, 2011.

[9] S. MartíNez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[10] D. Lee, C. Liu, Y.-W. Liao, and J. K. Hedrick, "Parallel interacting multiple model-based human motion prediction for motion planning of companion robots," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 52–61, 2017.

[11] S. E. Li, G. Li, J. Yu, C. Liu, B. Cheng, J. Wang, and K. Li, "Kalman filter-based tracking of moving objects using linear ultrasonic sensor array for road vehicles," *Mechanical Systems and Signal Processing*, vol. 98, pp. 173–189, 2018.

[12] Y. Xu, J. Choi, and S. Oh, "Mobile sensor network navigation using gaussian processes with truncated observations," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1118–1131, 2011.

[13] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[14] Y. Cao and D. J. Fleet, "Generalized product of experts for automatic and principled fusion of gaussian process predictions," *arXiv preprint arXiv:1410.7827*, 2014.

[15] M. E. Campbell and N. R. Ahmed, "Distributed data fusion: Neighbors, rumors, and the art of collective knowledge," *IEEE Control Systems Magazine*, vol. 36, no. 4, pp. 83–109, 2016.

[16] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2010.

[17] A. Q. Li, P. K. Penumarthi, J. Banfi, N. Basilico, J. M. O'Kane, I. Rekleitis, S. Nelakuditi, and F. Amigoni, "Multi-robot online sensing strategies for the construction of communication maps," *Autonomous Robots*, pp. 1–21, 2019.

[18] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4775–4782.

[19] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1540–1553, 2017.

[20] H. Wei, W. Lu, P. Zhu, S. Ferrari, M. Liu, R. H. Klein, S. Omidshafiei, and J. P. How, "Information value in nonparametric dirichlet-process gaussian-process (dpgp) mixture models," *Automatica*, vol. 74, pp. 360–368, 2016.

[21] C. Liu, Z. Liao, and S. Ferrari, "Rumor-robust decentralized gaussian process learning, fusion, and planning for modeling multiple moving targets," *arXiv preprint arXiv:2009.06021*, 2020.

[22] N. R. Ahmed and M. Campbell, "Fast consistent chernoff fusion of gaussian mixtures for ad hoc sensor networks," *IEEE transactions on signal processing*, vol. 60, no. 12, pp. 6739–6745, 2012.

[23] F. Nielsen, "Chernoff information of exponential families," *arXiv preprint arXiv:1102.2684*, 2011.