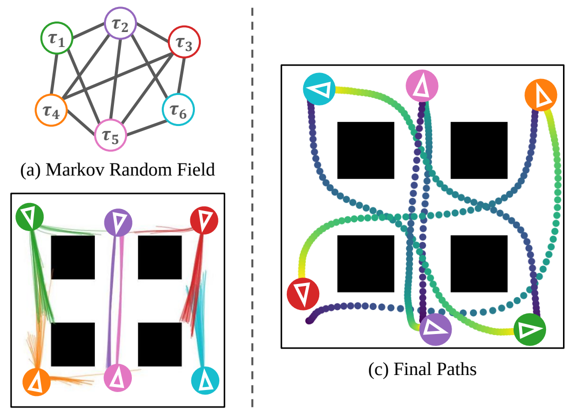
Stein Variational Belief Propagation for Decentralized Multi-Robot Control

Jana Pavlasek[1] Joshua Mah[1] Odest Chadwicke Jenkins[1] Fabio Ramos[2]

***Abstract*— Decentralized control for multi-robot systems involves planning in complex, high-dimensional spaces. The planning problem is particularly challenging in the presence of potential collisions between robots and obstacles, and different sources of uncertainty such as inaccurate dynamic models and sensor noise. A multi-robot system can be represented as a graphical model, in which nodes represent individual robots and edges represent communication between robots. This representation enables the use of graphical inference algorithms for solving multi-robot control. In this short paper, we introduce Stein Variational Belief Propogation (SVBP), a novel algorithm for performing inference over the marginal distributions of nodes in a graph. We present simulation results which demonstrate that our method can represent complex, multi-modal distributions in localization and control tasks.**



(a) Markov Random Field

(b) Trajectory Belief

(c) Final Paths

Fig. 1: Stein Variational Belief Propagation (SVBP) computes marginal trajectory distributions for each robot in a multi-robot system. SVBP represents the relationships between robots as a Markov Random Field (a) and maintains multi-modal distributions over each robot trajectory (b). The final trajectory generated by SVBP for each robot is shown in (c), where the colour represents time, and the robot is shown in its final position.

I. Introduction

Multi-robot coordination is an essential capability for applications involving teams of robots, such as industrial robots, delivery vehicles, and autonomous cars. Planning for multi-robot systems is challenging due to the high-dimensionality introduced by a large number of agents. Decentralized control is one strategy for multi-robot coordination, in which each robot performs local computations using information from neighboring robots. This approach is well-suited to multi-robot systems since it enables distributed algorithms which solve lower-dimensional, local problems compared to the expensive high-dimensional centralized approach.

Robot swarms have been represented as graphical models, where each robot is a node in the graph, and edges connect robots in communication range [1]–[4]. Decentralized control can then be performed via a graphical inference technique such as belief propagation. Belief propagation infers marginal posteriors for each node in a graph using only messages from immediate neighbors [5]. As a result, this algorithm enables multi-hop information to be passed through the graph. In this work, we represent a multi-robot system as a Markov Random Field (MRF) and perform belief propagation to infer the marginal belief for each robot (see Figure 1).

A number of belief propagation algorithms have been proposed in the literature. Gaussian Belief Propagation (GaBP) is an efficient algorithm when the node distributions and their corresponding factors can be represented as Gaussian [6], [7]. This method enables efficient computation and has been shown to be effective for multi-robot collision avoidance and localization [3], [4]. However, many applications in robotics are complex and multi-modal, and cannot be fully represented by Gaussian uncertainty. Nonparametric Belief Propagation (NBP) [8], [9] represents distributions nonparametrically as mixtures of Gaussians, but involves expensive product operations between mixture distributions. Particle Belief Propagation (PBP) [10] uses importance sampling to iteratively update a set of particles representing the belief, enabling the representation of arbitrarily complex distributions. PBP relies on the definition of a sampling distribution, which later work proposed to estimate via expectation maximization [11]. Importance sampling is prone to mode collapse, an effect which has been mitigated by using multiple sampling distributions [12]. Belief propagation has been applied to robotic perception of articulated objects, using an efficient sampling-based message product technique [13], learned unary factors [14], and end-to-end learned factors [15]. While these methods enable complex representations of belief distributions, they rely on expensive sequential sampling operations.

In this short paper, we propose Stein Variational Belief Propagation (SVBP), an efficient method for performing nonparametric belief propagation which employs Stein Variational Gradient Descent (SVGD) [16]. SVGD is a technique for inference in which distributions are represented nonparametrically using a set of particles which are iteratively updated using a gradient rule. This method has been shown to mitigate mode collapse and effectively represent multi-modal distributions compared to sampling-based methods through a repulsive

[1]Robotics Department, University of Michigan, Ann Arbor, USA. {pavlasek, joshmah, ocj}@umich.edu. [2]NVIDIA Corporation, Seattle, USA & School of Computer Science, University of Sydney, Sydney, Australia. fabio.ramos@sydney.edu.au.

force between particles arising from the use of the kernel in the update rule. Since the particles are deterministically updated through gradient steps, the algorithm is efficient and parallelizable. SVGD has proven useful in a number of robotic applications in recent years, including control, planning, and point cloud matching [17]–[20]. SVGD has been applied to graphical models to approximate the joint distribution using kernels over local node neighborhoods [21], and the conditional distributions over nodes [22]. Both these methods rely on the conditional independence structure of MRFs and as such only pass messages between immediate neighbors in the graph. In contrast, our proposed method computes the *marginal* beliefs over nodes using belief propagation, which involves passing messages through the whole graph.

We formulate our algorithm by leveraging the particle message update rules from PBP [10] combined with SVGD to update marginal distributions, eliminating the need for sampling and fully leveraging gradient information. We demonstrate preliminary results on 2D multi-robot control simulations which show that SVBP can maintain multi-modal belief distributions in uncertain environments, leading to improved performance compared to baselines.

II. Belief Propagation

Let $G = (V, E)$ denote a Markov Random Field (MRF) with nodes V and edges E. Let $\mathcal{X} = \{x_s \mid s \in V\}$ denote the set of all hidden nodes in the graph, and $\mathcal{Z} = \{z_s \mid s \in V\}$ denote the observed nodes corresponding to each hidden node. The joint probability of the graph G can be expressed as a product of its clique potentials:

$$p(\mathcal{X}, \mathcal{Z}) \propto \prod_{(s,t)\in E} \psi_{st}(x_s, x_t) \prod_{s\in V} \phi_s(x_s, z_s). \tag{1}$$

The function ψ_{st} is the *pairwise potential*, describing the correspondence between neighboring nodes, and ϕ_s is the *unary potential*, describing the correspondence of a hidden variable x_s with the observed variable z_s. Belief propagation estimates the marginal posterior distribution of a hidden node s using the following equation:

$$p(x_s \mid \mathcal{Z}) \propto \phi_s(x_s) \prod_{t\in\rho(s)} m_{t\to s}(x_s) \tag{2}$$

where $\rho(s)$ denotes the neighbors of s. The message from node t to node s, $m_{t\to s}$, is defined as:

$$m_{t\to s}(x_s) = \int_{x_t} \phi_t(x_t)\psi_{ts}(x_t, x_s) \prod_{u\in\rho(t)\setminus s} m_{u\to t}(x_t)\, dx_t \tag{3}$$

Note that we omit the observation, z_s, from the unary potential ϕ_s for brevity.

A. Particle Belief Propagation

Particle Belief Propagation (PBP) defines a sampling-based algorithm for computing the messages in Equation (3) for cases where the integral is intractable due to the complexity of the state space [10]. PBP represents the belief at each node with a set of N particles, $\left\{x_s^{(i)} \mid i \in 1\dots N\right\}$. Given samples $x_t^{(1)}, \dots, x_t^{(M)}$ drawn from a candidate distribution W_t, PBP defines the approximate message:

$$\hat{m}_{t\to s}(x_s^{(i)}) = \frac{1}{M}\sum_{j=1}^{M} \frac{\phi_t(x_t^{(j)})}{W_t(x_t^{(j)})} \psi_{ts}(x_t^{(j)}, x_s^{(i)}) \prod_{u\in\rho(t)\setminus s} m_{u\to t}(x_t^{(j)}). \tag{4}$$

This message definition is used to draw samples from the marginal posterior, $p(x_s \mid \mathcal{Z})$, using importance sampling.

III. Stein Variational Belief Propagation

Stein Variational Gradient Descent (SVGD) [16] approximates the true posterior distribution with a nonparametric candidate distribution which takes the form of a set of particles. SVGD iteratively updates the set of particles in order to minimize the KL-divergence between the true and candidate posteriors. At an iteration k, the following update rule is applied to each particle $x_s^{(i)}$:

$$x_s^{(i)}[k] \leftarrow x_s^{(i)}[k-1] + \epsilon\gamma(x_s^{(i)}[k-1]) \tag{5}$$

$$\gamma(x_s) = \frac{1}{N}\sum_{j=1}^{N} k(x_s^{(j)}, x_s)\nabla_{x_s^{(j)}} \log p(x_s^{(j)}) + \nabla_{x_s^{(j)}} k(x_s^{(j)}, x_s) \tag{6}$$

where $k(x_s, \cdot)$ is a kernel function between particles corresponding to node s. Intuitively, the kernel gradient term of the above equation acts as a repulsive force between particles. In practice, this characteristic prevents mode collapse in SVGD and often requires less particles to cover the space.

Stein Variational Belief Propagation (SVBP) applies SVGD to infer the marginal belief of each node, $p(x_s)$, from Equation (2). Note that we drop the observation $\mathcal{Z}$ from the posterior for brevity. Combining with Equation (6), the update rule for the particles at node s is:

$$\nabla_{x_s^{(j)}} \log p(x_s^{(j)}) = \nabla_{x_s^{(j)}} \log \phi_s(x_s^{(j)}) + \sum_{t\in\rho(s)} \nabla_{x_s^{(j)}} \log m_{t\to s}(x_s^{(j)}), \tag{7}$$

where $m_{t\to s}(x_s)$ is defined via the PBP message rule from Equation (4). A distinct set of Stein particles represents the posterior belief at each node. We note that the gradient update from Equation (7) only involves evaluating gradient information from immediate neighbors. This enables efficient gradient updates since the algorithm only requires back-propagating through immediate neighbors.

In practice, we use the current belief of the neighboring node, $p(x_t)$, as the sampling distribution, W_t, where $p(x_t)$ is represented by Stein particles for node t with equal weights. This enables efficient computation of the messages since it eliminates the need to run expensive sampling algorithms like MCMC, as originally proposed in the PBP algorithm.

We employ a synchronous message passing scheme in which all messages are computed prior to updating each node belief. This enables efficient batch computations of factors and messages suitable for execution on a GPU. However, our algorithm can be employed with other message passing schedules.

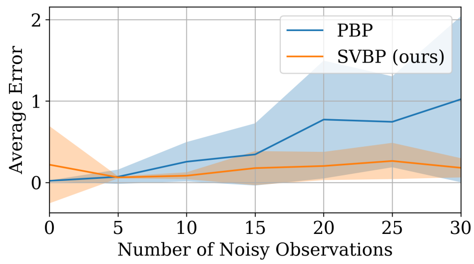


Fig. 2: Average error for each node estimate for multi-robot localization. Results are shown for varying levels of noise, corresponding to the number of noisy components added to the observation.

IV. Preliminary Results

To demonstrate the viability of the proposed approach, we conduct two simulated experiments which represent robotic problem domains in both multi-robot perception and control. We provide supplemental video results at the following webpage: https://progress.eecs.umich.edu/projects/stein-bp.

A. Multi-Robot Perception

The perception experiment involves localizing a collection of agents in which the observation for each agent is multi-modal. An example observation and the associated graph is shown in Figure 3. Each robot is represented by a distinct color, and its multi-modal sensor observation is represented in the same color. In addition to a sensor observation, robots observe each other, creating an edge between communicating robots (shown in red). This experiment is a version of the articulated "spider" localization problem from the NBP literature [9], [13], [15].

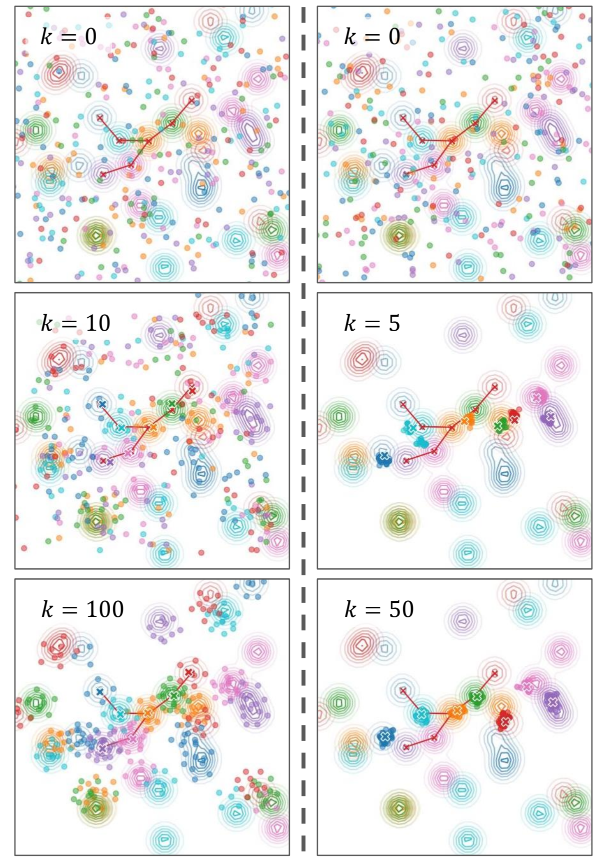
The unary potential for each robot is a mixture of Gaussians corresponding to the robot observation. The pairwise potential is a function of the observed translation d between neighboring robots:

$$\psi_{ts}(x_t, x_s) = \exp\Big(-\alpha\big(\|x_s - x_t\| - d\big)\Big). \tag{8}$$

Note that we assume that d is known (we use $d = 1$ for all edges). We restrict the graph to a chain structure for this problem.

Baseline: We implement Particle Belief Propagation as a baseline approach. We employ iterative importance sampling over the particles at each node, where each particle is weighted according to Equation (2) with the message definition of Equation (4). We use the current particle set at each node as the candidate distribution for message computation. We apply random noise at the beginning of each iteration. The same factor definitions and parameters are used for PBP and SVBP.

Results: To generate an estimate for each node's position, we select the highest weighted particle according to the marginal posteriors. The average error for each node over 10 runs for our SVBP algorithm against PBP is shown in Figure 2. The x-axis represents the number of noisy components added over all the Gaussian mixtures which represent the node observations. For each run, the noisy components are randomly assigned across nodes. SVBP ran for 100 iterations, and PBP ran for 50 iterations.



(a) SVBP (b) PBP

Fig. 3: Qualitative results for SVBP (ours) and PBP over iterations (k). The red lines represent the true position of the nodes, and the colored 'x' markers represent the maximum likelihood estimate for each node. The colored distributions represent the noisy observations for each node of the corresponding color. Best viewed in color.

Our SVBP method performs comparatively to PBP for low observation noise, but significantly outperforms PBP in noisy cases. We observed that PBP tends to converge quickly but was subjected to mode collapse which results in locally optimal estimates. In contrast, SVBP maintains multiple modes, making it more likely that the global solution is represented in the candidate particle set. A visualization of the belief distributions of SVBP and PBP for the highest noise observation is shown in Figure 3.

B. Decentralized Multi-Robot Control

Our second experiment involves decentralized control of a multi-robot system. Each robot must avoid obstacles and the other robots in its trajectory to the goal. We take a planning as inference approach [23]–[25] in which the nodes in the graph represent the trajectory distribution for each robot over a fixed horizon T, and the edges in the graph represent robots in communication, as in Figure 1. We consider the trajectories to be a sequence of acceleration commands, $\tau_s = \{u_{s,k} \mid 0 \le k \le T\}$, governed by known dynamics $x_{s,k+1} = f_s(x_{s,k}, u_{s,k})$, where $x_{s,k}$ is the position and velocity of robot s at time k.

For this experiment, we assume the graph is fully-connected. We employ a loopy version of belief propagation,

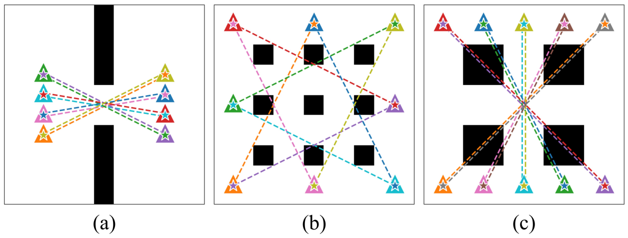


Fig. 4: Testing environments for the multi-robot control experiments. The start and goal positions are marked with a triangle and star respectively. Each color represents a single robot.

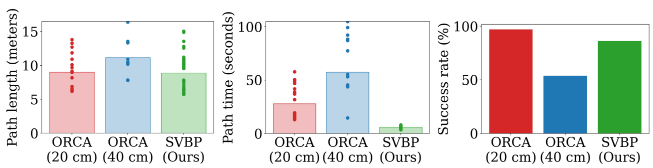


Fig. 5: Path length, path time, and success rate for each method considered for the multi-robot control example. A trajectory is successful if it reaches the goal within 15 cm. Only successful trajectories are included in path length and path time.

in which the messages are initialized and iteratively updated. This approach does not provide exact marginals but has proven to be effective in practice [26]. The unary potential for each robot trajectory consists of an obstacle avoidance potential, a quadratic cost, and a goal potential. The pairwise potential between communicating robots employs the following collision avoidance factor over the trajectory:

$$\psi_{ts}(\tau_t, \tau_s) = \sum_{k=0}^{T} \alpha_k \left(1 - \left(\frac{\|x_{s,k} - x_{t,k}\|}{r}\right)^{\beta}\right) \qquad (9)$$

where T is the time horizon, $\|x_{s,k} - x_{t,k}\|$ is the distance between the robot positions at timestep k, r is the desired collision radius, and α_k and $0 < \beta \leq 1$ are constants. In our experiments, we use $r = 0.5$ and $\beta = 0.3$. We set α_k to decrease linearly over the horizon. We assume known, linear dynamics which allows the gradients to be computed with respect to the acceleration commands. At each timestep, we execute the first action in the trajectory and replan, as in model predictive control (MPC). This approach is akin to a multi-robot version of Stein MPC [17].

Baseline: Our baseline for this scenario is the well-established Optimal Reciprocal Collision Avoidance (ORCA) algorithm [27]. ORCA assumes that neighboring agent's velocity can be perfectly known and calculates optimal reciprocally collision-avoiding velocities that are closest to the original preferred velocity. The scenario was implemented using RVO2 library [28] for robots with radii of 20 cm and 40 cm that could achieve maximum velocities of 2 m/s. We assume full connectivity.

Results: We present the total path length and the total path time for ORCA and SVBP in Figure 5. Since ORCA is sensitive to the robot radius parameter, we show results for both a radius of 20 cm and 40 cm. We perform 10 runs on each of the environments in Figure 4. The success rate in Figure 5 represents the percentage of trajectories which successfully reached the goal with a threshold of 15 cm. Path length and time are only computed for successful trajectories.

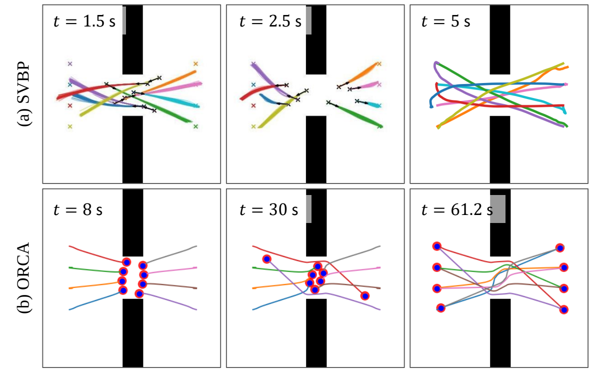


Fig. 6: Qualitative results for the multi-robot control experiment for SVBP and ORCA with a 20 cm collision radius.

We observe that the failure modes in SVBP are often due to imprecision in achieving the goal location due to the tradeoff between factors. Another failure mode results when robots get stuck in local minima around large obstacles, such as in the environment in Figure 4(c). OCRA with a 40 cm collision radius fails for all runs in the environment in Figure 4(a).

When the robot collision radius is small, ORCA results in comparable path lengths to SVBP, however, the robots move much more conservatively in ORCA, which results in higher path times. A qualitative comparison in the environment in which the robots must cross a narrow gap is shown in Figure 6.[1]

V. Discussion & Future Work

This paper describes Stein Variational Belief Propagation, an algorithm for performing inference over the marginal distributions within a graph using Stein Variational Gradient Descent. We present results in simulation on a multi-robot perception task and a multi-robot decentralized control task. Our results suggest that our algorithm is capable of maintaining complex, multi-modal distributions in uncertain and cluttered environments. We hypothesize that the particle distributions in SVBP can be used to provide an uncertainty estimate in noisy cases, such as the one shown in Figure 3. This could be employed in robotic applications to improve robustness to noise or within the context of active perception.

Future work will involve further characterization of the convergence properties of the algorithm proposed. We will characterize performance against other other baselines, including the message passing algorithm from Pacheco et al. [12], which shares the specific aim of maintaining modes in the belief distribution. Our findings for the multi-robot control experiment compared to the ORCA baseline are consistent with the findings from Patwardhan et al. [3], but we do not require that the trajectories are initialized with a trajectory planner. Future work will compare this algorithm to SVBP. We plan to evaluate our algorithm in the presence of real-world uncertainty through robot experiments, and in higher dimensional problems such as manipulation tasks.

[1]These results are best viewed in video form. See the website at: https://progress.eecs.umich.edu/projects/stein-bp.

References

[1] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, “Learning decentralized controllers for robot swarms with graph neural networks,” in *Conference on robot learning*. PMLR, 2020, pp. 671–682.

[2] E. Tolstaya, L. Butler, D. Mox, J. Paulos, V. Kumar, and A. Ribeiro, “Learning connectivity for data distribution in robot teams,” in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 413–420.

[3] A. Patwardhan, R. Murai, and A. J. Davison, “Distributing collaborative multi-robot planning with Gaussian belief propagation,” *Robotics and Automation Letters*, vol. 8, no. 2, pp. 552–559, 2022.

[4] R. Murai, J. Ortiz, S. Saeedi, P. H. Kelly, and A. J. Davison, “A robot web for distributed many-device localisation,” *arXiv preprint arXiv:2202.03314*, 2022.

[5] M. J. Wainwright, M. I. Jordan *et al.*, “Graphical models, exponential families, and variational inference,” *Foundations and Trends in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[6] Y. Weiss and W. Freeman, “Correctness of belief propagation in gaussian graphical models of arbitrary topology,” *Advances in neural information processing systems*, vol. 12, 1999.

[7] A. J. Davison and J. Ortiz, “FutureMapping 2: Gaussian belief propagation for spatial AI,” *arXiv preprint arXiv:1910.14139*, 2019.

[8] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, “Nonparametric belief propagation,” *Communications of the ACM*, vol. 53, no. 10, pp. 95–103, 2010.

[9] M. Isard, “PAMPAS: Real-valued graphical models for computer vision,” in *Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2003.

[10] A. Ihler and D. McAllester, “Particle belief propagation,” in *Artificial Intelligence and Statistics*, 2009, pp. 256–263.

[11] T. Lienart, Y. W. Teh, and A. Doucet, “Expectation particle belief propagation,” in *Advances in Neural Information Processing Systems*, 2015, pp. 3609–3617.

[12] J. Pacheco, S. Zuffi, M. Black, and E. Sudderth, “Preserving modes and messages via diverse particle selection,” in *International Conference on Machine Learning (ICML)*, vol. 32, no. 2, 2014, pp. 1152–1160.

[13] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins, “Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects,” *Science Robotics*, vol. 4, no. 30, 2019.

[14] J. Pavlasek, S. Lewis, K. Desingh, and O. C. Jenkins, “Parts-based articulated object localization in clutter using belief propagation,” in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[15] A. Opipari, J. Pavlasek, C. Chen, S. Wang, K. Desingh, and O. C. Jenkins, “Differentiable nonparametric belief propagation,” in *IEEE ICRA 2022 Workshop on Robotic Perception and Mapping: Emerging Techniques*, 2022.

[16] Q. Liu and D. Wang, “Stein variational gradient descent: A general purpose Bayesian inference algorithm,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[17] A. Lambert, A. Fishman, D. Fox, B. Boots, and F. Ramos, “Stein variational model predictive control,” in *Conference on Robot Learning (CoRL)*, 2020.

[18] L. Barcelos, A. Lambert, R. Oliveira, P. Borges, B. Boots, and F. Ramos, “Dual online Stein variational inference for control and dynamics,” in *Robotics: Science and Systems (RSS)*, 2021.

[19] F. A. Maken, F. Ramos, and L. Ott, “Stein ICP for uncertainty estimation in point cloud matching,” *Robotics and Automation Letters*, vol. 7, no. 2, pp. 1063–1070, 2022.

[20] A. Lambert, B. Hou, R. Scalise, S. S. Srinivasa, and B. Boots, “Stein variational probabilistic roadmaps,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11 094–11 101.

[21] D. Wang, Z. Zeng, and Q. Liu, “Stein variational message passing for continuous graphical models,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 5219–5227.

[22] J. Zhuo, C. Liu, J. Shi, J. Zhu, N. Chen, and B. Zhang, “Message passing Stein variational gradient descent,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 6018–6027.

[23] H. Attias, “Planning by probabilistic inference,” in *International Workshop on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. R4. PMLR, 2003, pp. 9–16.

[24] M. Toussaint, “Robot trajectory optimization using approximate inference,” in *International Conference on Machine Learning (ICML)*, 2009, pp. 1049–1056.

[25] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, “Information-theoretic model predictive control: Theory and applications to autonomous driving,” *Transactions on Robotics (T-RO)*, vol. 34, no. 6, pp. 1603–1622, 2018.

[26] K. P. Murphy, Y. Weiss, and M. I. Jordan, “Loopy belief propagation for approximate inference: An empirical study,” in *Conference on Uncertainty in Artificial Intelligence (UAI)*, ser. UAI’99. Morgan Kaufmann Publishers Inc., 1999, p. 467–475.

[27] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, “Reciprocal n-body collision avoidance,” in *Robotics Research: The 14th International Symposium (ISRR)*. Springer, 2011, pp. 3–19.

[28] J. Van Den Berg, S. J. Guy, J. Snape, M. Lin, and D. Manocha. (2016) Rvo2 library: Reciprocal collision avoidance for real-time multi-agent simulation. [Online]. Available: https://gamma.cs.unc.edu/ORCA/