nication requirements by up to 3 times when compared to a localization-only approach for a 20 robot network. In our experiments, we demonstrate cases where our tracking algorithm correctly determines the location of the robot, while processing each timestep independently fails to do so. Other approaches, for example those based on directly weighting temporal samples by inter-distance likelihoods, failed to localize the robots. Our approach, based on a novel decomposition of the temporal dynamics model, allows us to integrate multi-hop inter-distance readings in a single timestep and track the robots. We also show how NBP message updates should be scheduled to achieve robust, reliable convergence.

In the Experiments section, we compare our algorithm to a Monte-Carlo approach developed by Dil et al. [1]. We find that our algorithm is slower but 3 to 4 times more accurate, and more robust to noisy inter-distance sensor readings.

II. Related Work

Many robotics problems are related to our collaborative inter-distance tracking problem, including simultaneous localization and mapping (SLAM), simultaneous localization and tracking (SLAT), control of robotic swarms, and robot/sensor network localization. We distinguish between localization approaches, which use sensor readings from a single timestep (perhaps in series over multiple timesteps), and tracking approaches, which integrate location estimates using models of robot dynamics.

In classic SLAM problems, a single mobile robot produces a map of the environment while simultaneously determining its location. Several recent approaches allow SLAM problems to be efficiently solved with as many as 10^8 static features where distributions are modeled as multivariate Gaussians. Many of these methods are based on inference in an appropriate graphical model; for an overview, see [2]. SLAM has also been extended to domains with multiple robots, often under the assumption that the initial locations of the robots are known [7], [8]. In these approaches, robots share and localize using a global map, rather than through distributed observations of other robots as in our approach. Thrun and Liu [9] investigate merging multiple maps when initial locations of robots are unknown, and the landmarks are not uniquely identifiable. Howard [10] explores map merging when robots do not know their initial locations. In contrast, our approach uses a series of inter-robot sensor values.

The SLAT problem is a variant of SLAM in which static robots, which are typically nodes in a sensor network, use distance and bearing readings from a moving robot to localize themselves. The approach of Taylor et al. [11] avoids representing ring-shaped or multi-modal posterior distributions by assuming sensors roughly know their initial locations, and batch processing sets of distance measurements. Funiak et al. [12] propose a more complex method focused on SLAT problems in camera networks.

In distributed control, methods have been proposed for controlling robot swarms to maintain a specified formation [13], or using several mobile robots to localize a target [14]. These approaches assume sensors observe true locations plus Gaussian noise, resulting in posterior distributions which (unlike those arising in inter-distance tracking) are well approximated as Gaussian. However, our approach shares the goal of developing effective distributed algorithms for multiple robots.

A great deal of research focuses on distance-based localization in sensor networks; for a summary, see [15]. These localization techniques can be applied at each timestep to determine the location of each robot, but ignore dynamics. To address static localization problems, researchers have applied techniques such as multidimensional scaling (MDS), in both centralized [16] and distributed [17] implementations. Many localization methods produce global solutions by incrementally stiching local sensor maps together [17], for instance by recursively localizing new sensors with respect to previously localized sensors [18]. Other approaches solve more global inference problems, computing location estimates via iterative least squares [19], [20]. Such methods can be very effective when bearing estimates are available, as in the bundle adjustment methods widely used in camera networks [21]. However, with the greater posterior uncertainty produced by inter-distance measurements, the approximate marginal distributions computed by NBP are often more robust and effective [6].

In some multi-robot tracking applications, a sufficient number of beacon nodes (at least three) can be directly observed by all robots at all times [22], [23]. Other approaches assume sensors measure both orientation and distance, allowing for simplifying Gaussian assumptions [24]. Previous approaches to the inter-distance tracking problem include the work of Howard et al. [25], which formulates inter-distance tracking as numerical optimization; Park et al. [26], which solves a trilateration problem using quadratic approximations of the motion of robots relative to one another; and Dil et al. [1], which uses assumptions about maximum transmission radii to apply particle filters.

In contrast with previous localization methods for networks of mobile robots, our distributed tracking algorithm is based on applying the nonparametric BP algorithm to a probabilistic graphical model. Schiff et al. [27] previously demonstrated the feasibility of running BP in a real network of MICA2 motes for static sensor fusion.

III. Problem Formulation

A. Input, Assumptions, and Output

Our models supports tracking applications where n robots move around a space for T timesteps. The location of robot s at time τ is denoted by $x_{s,\tau} \in \mathbb{R}^2$. We currently model the space as a 1x1 unit square, but our approach can be easily relaxed to support other geometries. Each robot is equipped with sensors for inter-distance measurement. We denote the (noisy) distance estimate between robots s and t at time τ as $d_{st,\tau}$. As distance estimates are based on signals emitted by nearby robots, the likelihood a sensor will detect a nearby sensor diminishes with distance. We refer to the subset of