A Decentralized Kernel Density Estimation Approach to Distributed Robot Path Planning

Greg Foderaro
Department of Mechanical Engineering and Materials Science
Duke University
Durham, NC 27708
greg.foderaro@duke.edu

Silvia Ferrari
Department of Mechanical Engineering and Materials Science
Duke University
Durham, NC 27708
sferrari@duke.edu

Michael M. Zavlanos
Department of Mechanical Engineering and Materials Science
Duke University
Durham, NC 27708
michael.zavlanos@duke.edu

Abstract

This paper presents a decentralized kernel density estimation (KDE) technique for computing the actual positional density of the robots in a distributed network, which constitutes the feedback in the robot's control law. The goal of the feedback control law is to plan the paths of a distributed robot network in order to follow a known, optimal time-varying robot distribution or probability density function (PDF). Thus, knowledge of the actual positional density of the robots is needed to compute the robot feedback control law, such that the optimal PDF is achieved over time by the network. The optimal PDF is computed using a distributed optimal control (DOC) approach that guarantees the robots avoid collisions with obstacles, while minimizing the energy required to meet a goal distribution. This novel approach generates a potential function, and corresponding control law, for each robot through the decentralized computation of the robots' probability density function (PDF) obtained from the individual states of the robots. The methodology is demonstrated through a numerical simulation of a large distributed network of robots navigating an obstacle-populated region of interest.

1 Introduction

Distributed robot path planning has a broad range of applications from sensor networks [1, 2] to animal behavior [3]. These systems typically involve many cooperative robots that are each governed by a small system of ordinary differential equations (ODEs), referred to as *microscopic dynamics*. On large spatial and temporal scales, the interaction and collaboration between the robots give rise to a macroscopic behavior that can then be described by a restriction operator, such as an evolving (time-varying) probability density function (PDF) of the (microscopic) robots' states, and/or its first moments of the PDF. The evolution of the time-varying PDF can often be modeled as a system

of partial differential equations (PDEs), and the PDF can be optimized subject to the microscopic dynamics through the distributed optimal control problem presented by the authors in [2]. In this paper, the optimal, time-varying PDF is modeled as a time-varying mixture of Gaussians, and is assumed given.

The microscopic control law of each robot can then be derived from the optimal time-varying PDF, and from the actual robot PDF, using a potential field approach that pushes the robots toward an attractive potential computed from the difference between the two PDFs [1]. While the computational complexity of the DOC approach for computing the optimal PDF has been shown to be far reduced compared to classical optimal control [2], the calculation of the actual robot PDF requires knowledge of all individual robot states, and, thus, in principle, tends to be expensive and centralized.

This paper presents a novel approach based on decentralized kernel density estimation (KDE) for estimating the actual distribution (PDF) of the robot states in a decentralized fashion, through the nonparametric technique for estimating distributions from empirical data [4]. In the decentralized KDE algorithm, the robots use an information spreading protocol to exchange state information with their neighbors, and then perform local kernel density estimations. The density estimations calculated locally are asymptotically consistant with the solutions obtained using centralized kernel density estimation [5].

Several distributed parametric techniques also exist for estimating a distribution from a data set, where the unknown density is represented by a mixture of Gaussians, and the parameters are determined by an expectation maximization (EM) algorithm [6, 7]. However, these methods typically exhibit large inaccuracies on small data sets, often getting stuck in local maxima, and are sensitive to the initial parameter choices. The distributed KDE approach does not suffer from any of these downfalls, and is, therefore, considered to be the best choice for the DOC problem. The approach, presented in Sections 2-3, is demonstrated on a distributed path-planning problem involving a large network of robots in Section 4.

2 Probability Density Function Approach to Robot Path Planning

Potential field is a well-known approach to robot motion planning that treats the robot as a particle under the influence of an artificial potential field or function, U, that captures the geometric characteristics of an obstacle-populated workspace, and a goal configuration. In this paper, we assume that the workspace is a two-dimensional rectangular region of interest (ROI), denoted by $\mathcal{W} = [0,\ L] \times [0,\ L] \subset \Re^2$. Typically, the potential function is the sum of an attractive potential U_{att} that "pulls" the robot toward the goal configuration $\mathbf{x}_f$, and a repulsive potential U_{rep} that "pushes" the robot away from the obstacles [8]. After U is defined, the method is implemented by discretizing the robot workspace $\mathcal{W}$, and by evaluating the potential function for all discrete values of $\mathbf{x}$ in $\mathcal{W}$, using a finite resolution grid [9].

In [1], the authors presented a potential field approach for planning the paths of n cooperative sensors, such that they follow (or are "pulled" toward) a time-varying probability density function (PDF), comprised of a mixture of Gaussians. This PDF-based approach differs from potential field methods previously presented in the literature in that it is based on the joint PDF of the state of multiple robotic sensors, it is time-varying, and avoids collisions between sensors and multiple obstacles while minimizing the power consumption. Consider the problem of planning the paths of a network of N robots in $\mathcal{W}$, where each robot can be described by a small number of ordinary differential equations (ODEs), referred to as *microscopic dynamics*, which, in this paper, are given by the unicycle model

$$\dot{x}_i = v_i \cos\theta_i, \quad \dot{y}_i = v_i \sin\theta_i, \quad \dot{\theta}_i = \omega_i \tag{1}$$

The microscopic i^{th} robot configuration, $\mathbf{q}_i = [x_i\ y_i\ \theta_i]^T$, consists of the xy-coordinates, x_i and y_i, and of its heading angle, θ_i, where $i = 1, ..., N$. The microscopic control vector, $\mathbf{u}_i = [v_i\ \omega_i]^T$, consists of the linear velocity, v_i, and the angular velocity, ω_i. Then, at any time t_k an artificial force $\mathbf{F}(\mathbf{q}_i) \propto -\nabla U(\mathbf{q}_i, t_k)$, proportional to the negative gradient of the artificial potential, is applied to the robot in order to follow the steepest-descent direction of U.

The interactions of the N robots give rise to a macroscopic coherent behavior with coarse dynamics that can be modeled by partial differential equations (PDEs) [2]. Using a restriction operator that maps the microscopic states to the macroscopic description, as described in [10]. In this paper, it is

assumed that the macroscopic description of the robot network is a time-varying PDF, denoted by $\wp[\mathbf{x}_i, t]$, such that the probability of the i^{th} robot being in state $\mathbf{x}_i$ at time t is given by $\wp[\mathbf{x}_i, t]$. In other words, $\wp$ represents the robot density over the ROI, $\mathcal{W}$. In many sensor and unmanned vehicle applications, the macroscopic network performance can be shown to be a function of the restriction operator, or robot distribution, and controls, and, thus, can be expressed as an integral function of $\wp$ and $\mathbf{u}_i$ [2]. Then, the optimal robot PDF, $\wp^*$, and control laws, $\mathbf{u}_i^*$, can be determined using the distributed optimal control (DOC) method described in [2].

The attractive potential of each robot can then be defined based on the optimal PDF, $\wp^*(\mathbf{x}_i, t_k)$, which represents the goal density of the robots. Where, when integrated over a region $\mathcal{R} \subset \mathcal{W}$, $\wp^*$ provides the probability that the ith robot is located in $\mathcal{R}$ at t_k, i.e., the probability mass, $\Pr(\mathbf{x}_i \in \mathcal{R}, t_k) = \int_{\mathcal{R}} \wp^*(\mathbf{x}_i, t_k) d\mathbf{x}$. Since $\wp^*$ represents the goal joint PDF for the N robots, the effect of a robot moving to a state $\mathbf{x}_i$ should downgrade the probability mass, such that the probability that another robot in the network takes the same state value is decreased. In principle, each robot i can construct a feedback control law from the artificial force, using its knowledge of the optimal robot distribution $\wp^*$, and the actual robot distribution in $\mathcal{W}$ as follows. Let the attractive potential for the i^{th} robot be defined as the difference between the actual agent distribution y_i as seen by the i^{th} robot, and the time-shifted optimal distribution

$$U_{att}^i[\mathbf{x}_i(t), t] = y_i[\mathbf{x}_i(t), t] - \wp^*[\mathbf{x}_i(t + t_d), t + t_d] \tag{2}$$

Where, t_d is a time-shift parameter that allows the control law to look ahead in time to the optimal distribution, preventing the robots from lagging behind.

The potential function for robot i can then be defined as the sum,

$$U_i(\mathbf{x}_i, t) = w_a U_{att}^i(\mathbf{x}_i, t) + w_r U_{rep}^i(\mathbf{x}_i, t) \tag{3}$$

where, U_{rep}^i is a repulsive potential constructed based on the obstacles in the ROI [1], and w_a and w_r are user-defined weighting coefficients. The control law for each unicycle robot (1) is obtained from the negative gradient of the potential function in (3)

$$\mathbf{u}_i = [v_c \quad Q(\hat{\theta}_i, -\nabla U_i)]^T \tag{4}$$

Where, the minimum difference between the desired heading angle $\Theta(-\nabla U_i)$ and the i^{th} robot's actual heading angle $\hat{\theta}_i$ is,

$$Q(\cdot) = \{a(\hat{\theta}_i) - a[\Theta(-\nabla U_i)]\} sgn\{a[\Theta(-\nabla U_i)] - a(\hat{(\theta)}_i)\}, \tag{5}$$

and where $sgn(\cdot)$ is the sign function, $a(\cdot)$ is an angle wrapping function, and v_c is the robot vehicle's speed, which is set equal to a constant for simplicity [11].

In order to utilize this approach for a decentralized network of robots, the actual robot density y_i must be approximated locally by the i^{th} robot, without requiring direct communication with all other $(N-1)$ robots in the network. This can be achieved through a decentralized adaptation of the nonparametric technique known as kernel density estimation (KDE). In KDE, each node of a decentralized network repeatedly exchanges data with its neighbors through information spreading, and then performs a local KDE calculation. Through this process, each local estimate separately converges asymptotically to the distribution one would obtain using the centralized KDE method. Other decentralized techniques have been presented for estimating a distribution from a data set which use distributed expectation maximization (EM) algorithms, but they suffer from several disadvantages such as poor performance with small data sets, sensitivity to initial parameter choices, and the potential to get trapped in local maxima. Whereas, the distributed KDE approach presented in the next section does not suffer from any of these limitations.

3 Decentralized Kernel Density Estimation for Distributed Robot Path Planning

Kernel density estimation is a well-known non-parametric approach for estimating the probability density function (PDF) from which a set of independent and indentically distributed data samples were taken. Given a data set $\mathbf{y}_j, j = 1, ..., N_y, \mathbf{y}_j \in \Re^d$ that is assumed to be from some unknown

PDF f, the kernel density estimation takes the form [4],

$$\hat{f} = \sum_{j=1}^{N_y} w_j K_{\mathbf{H}_j}(\mathbf{y} - \mathbf{y}_j) \tag{6}$$

where $w_j, j = 1, ..., N_y$ is the weighting coefficients satisfying the condition $\sum_{j=1}^{N_y} w_j = 1$, and the j^{th} kernel centered at $\mathbf{y}_j$ is defined as,

$$K_{\mathbf{H}_j}(\mathbf{y} - \mathbf{y}_j) = |\mathbf{H}_j|^{-\frac{1}{2}} K(\mathbf{H}_j^{-\frac{1}{2}}(\mathbf{y} - \mathbf{y}_j)) \tag{7}$$

The kernel function K is a user-defined d-variate non-negative symmetric real-value function [4]. The band-width matrix $\mathbf{H}_j$ is a parameter that controls the smoothing of the KDE algorithm, and it must be positive definite and symmetric. With appropriate parameter choices, KDE has been shown to be an effective method for estimating the underlying PDF and often only requires a few samples to give adequate results [4]. However the general KDE algorithm described above requires centralized processing due to the summation, which might not be feasible in a distributed robot network.

Alternatively, a distributed KDE algorithm based on information spreading can be used that does not require centralized processing and is asymptotically consistent with the centralized version in cases where the network is fully connected [5]. The primary difference is that the distributed KDE algorithm uses an information sharing protocol to incrementally exchange kernel information between sensors until a complete and accurate approximation of the global KDE is achieved by each robot. It has been shown in [12] that as long as the network is fully connected, the connectivity structure will only affect the convergence speed and will not worsen the estimation accuracy. Therefore, KDE can be performed in a distributed manner with accuracies that are nearly identical to the centralized method and with the only requirement being the full connectivity of the network.

Each robot maintains a local estimation of the robot distribution, governed by a stored kernel set $\mathbf{S}_i = \{< w_{i,k}, \mathbf{x}_{i,k}, \mathbf{H}_{i,k} >, k = 1, ..., N_i\}$, where $\mathbf{x}_{i,k}$ denotes the position of robot k perceived by the i^{th} robot, N_i is the number of kernels stored by robot i, and $\mathbf{H}_{i,k}$ and $w_{i,k}$ are the bandwidth matrix and weighting coefficient of the k^{th} kernel stored by robot i. At time t_0, the kernel set of each robot only contains the kernel generated using its own position. The robots also maintain a neighbor set, where the i^{th} robot's neighbors are defined as any robots located within the distance of a communication radius r. Then through an information spreading process, the robots choose a random neighbor and compare their kernel sets with one another. If a robot sees that its neighbor has newer or previously unknown kernel information, they will save the information to their own stored kernel set. Then a new random neighbor will be chosen, and the process repeats.

In practice, the information communicated would include the sensors' positions and kernel parameters to construct the kernels and the corresponding sensors' indices and positional measurement timestamps to enable the overwriting of old data. Note that for many homogeneous robot networks (networks with identical sensors), the bandwidth matrices $\mathbf{H}_{i,k}$ and weighting parameters $w_{i,k}$ may be defined to be consistant across the network, making their communication unnecessary and reducing communication requirements. For simplicity, in this paper the bandwidth matrix is defined as $\mathbf{H}_{i,k} = c\mathbf{I}_2 \forall i, k$, where c is a constant and $\mathbf{I}_2$ is the two-dimensional identity matrix, and the weighting parameters are calculated as $w_{i,k} = \frac{1}{N_i} \forall i, k$. Then the purpose of the communications essentially becomes to give each robot full positional knowledge of all robots in the network.

Using their known sets of robot positions, each robot can then generate the corresponding kernels and combine them to obtain a local estimation of the PDF. For simplicity, the standard two-dimensional Gaussian kernel function is chosen in this paper and defined as,

$$K(\mathbf{x}) = \frac{1}{2\pi} e^{-\frac{1}{2}\mathbf{x}^T\mathbf{x}} \tag{8}$$

which is used to construct the kernels as follows,

$$K_{\mathbf{H}_{i,k}}(\mathbf{x} - \mathbf{x}_i) = |\mathbf{H}_{i,k}|^{-\frac{1}{2}} K(\mathbf{H}_{i,k}^{-\frac{1}{2}}(\mathbf{x} - \mathbf{x}_i)) \tag{9}$$

Then the local estimation of the PDF can be calculated by each robot as,

$$\hat{f}_i = \sum_{k=1}^{N_i} w_{i,k} K_{\mathbf{H}_{i,k}}(\mathbf{x} - \mathbf{x}_{i,k}) \tag{10}$$

Once the i^{th} robot has a local density estimation, the attractive potential (2) can be computed with $y \approx \hat{f}_i$, which leads to the generation of the potential function (3). Then the feedback control law is constructed from the negative gradient of the potential function $-\nabla U_i$, as in (4). Therefore by using the decentralized kernel density estimation approach, each robot in the network can derive its own feedback control law without centralized network computations.

4 Simulations and Results

The path planning approach presented in the previous sections is demonstrated through an example numerical simulation. Consider a network of $N = 200$ robots that are governed by the unicycle dynamics in (1) and have communication radii $r = 30$ Km. The network is deployed in a rectangular ROI denoted by $\mathcal{A} = [0,\ L_1] \times [0,\ L_2]$, where $L_1, L_2 = 160$ Km, and is given the objective of following an optimal time-varying robot distribution, avoiding obstacles, and minimizing energy consumed. The robot network operates over the time interval $t \in (t_0,\ t_f]$, where $t_0 = 0$ and $t_f = 15$ hr. The initial robot distribution $\wp^*(t_0)$ and geometric obstacles are shown in Figure 1, and the initial robot states are sampled from $\wp^*(t_0)$. The optimal robot distribution $\wp^*$ is assumed to be previously computed as a mixture of Gaussians via the DOC approach described in [2].

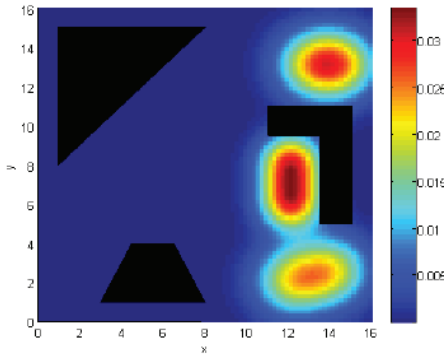


Figure 1: Initial robot distribution is plotted on the background, and the geometric obstacles are plotted as solid black.

With $\wp^*$ known, the individual control policies can be computed by each robot based on the potential function explained in Section 2 and its local estimation of the actual robot density obtained using the decentralized KDE method from Section 3. The robot navigation is illustrated in Figure 2, where both the optimal macroscopic distribution and the robot positions are plotted. It can be seen that the sensors follow $\wp^*$ while also avoiding collisions with obstacles.

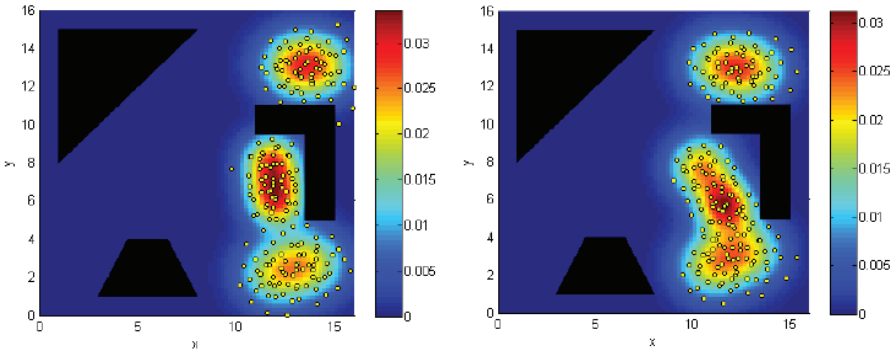
Since the DOC approach solves for the macroscopic time-varying robot distribution, instead of individual robot trajectories, the methodology is scalable and is shown to handle a large number of sensors. However, by using the decentralized kernel density estimation method, the network size can be limited by the storage constraints of the distributed sensors or by slow communication speeds across the network relative to the robot vehicle velocities.

5 Conclusions and Future Work

This paper presents a decentralized KDE approach that calculates the distribution of robot states in a distributed network. The technique is applied to a potential field approach for planning the paths of a distributed robot network with the objectives of matching a known time-varying optimal robot distribution, avoiding collisions with obstacles, and minimizing energy consumed by controlling the robots. The approach is illustrated through a numerical simulation of a large distributed network of robots that must navigate within an obstacle-populated region of interest.

Acknowledgments

This paper is supported by ONR grant (N00014-11-1-0688).



(a) $t = 0$ hr (b) $t = 5$ hr

(c) $t = 10$ hr (d) $t = 15$ hr

Figure 2: Simulated network of $N = 200$ sensors navigating to follow an evolving optimal PDF using distributed kernel density estimation and a potential field approach, where, the robot positions are plotted as black/yellow circles, and obstacles are plotted in solid black.

References

[1] S. Ferrari, G. Foderaro, and A. Tremblay. A probability density function approach to distributed sensors path planning. In *Proc. IEEE International Conference on Robotics and Automation*, pages 432–439, Anchorage, AK, 2010.

[2] G. Foderaro and S. Ferrari. Necessary conditions for optimality for a distributed optimal control problem. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4831–4838, Atlanta, GA, December 15-17 2010.

[3] K. M. Passino Y. Liu. Stable social foraging swarms in a noisy environment. *IEEE Transactions on Automatic Control*, 49(1), 2004.

[4] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley-Interscience, 1992.

[5] J. Lou J. Li Y. Hu, H. Chen. Distributed density estimation using non-parametric statistics. *International Conference on Distributed Computing Systems*, 2007.

[6] D. Gu. Distributed em algorithm for gaussian mixtures in sensor networks. *IEEE Transactions on Neural Networks*, 19(7), 2008.

[7] R. Nowak. Distributed em algorithms for density estimation and clustering in sensor networks. *IEEE Transactions on Signal Processing*, 51(8), 2003.

[8] S. Ge and Y. Cui. New potential functions for mobile robot path planning. *IEEE Transactions on Robotics and Automation*, 16(5), 2000.

[9] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2), 1992.

[10] O. Runbord I.G. Kevrekidis C.W. Gear C. Theodoropoulos J.M. Hyman, P.G. Kevrekidis. Equation-free coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4), 2003.

[11] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[12] O. Babaoglu M. Jelasity, A. Montresor. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems*, 23(3), 2005.