

Dynamic Maximum Entropy Algorithms for Clustering and Coverage Control

Yunwen Xu, S. Salapaka and C. L. Beck

Abstract—The dynamic coverage problem is increasingly found in a wide variety of areas, for example, from the development of mobile sensor networks, to the analysis of clustering in spatio-temporal dynamics of brain signals. In this paper, we apply control-theoretic methods to locate and track cluster center dynamics and show that dynamic control design is necessary to achieve dynamic coverage of mobile objects under *acceleration* fields. This is the first work to consider tracking cluster centers when site dynamics involve accelerations. We focus on the relationship between the objective of maximizing coverage in real-time and the Maximum Entropy Principle, and develop the ability to identify inherent cluster dynamics in a dataset. Algorithms are presented that guarantee asymptotic tracking of cluster centers, and for which we prove continuity and boundedness of the corresponding control laws. Simulations are provided to corroborate these results.

I. INTRODUCTION

With recent advancements in geographic information systems, distributed sensor networks and wireless communications, a wide range of applications could benefit from the ability to efficiently track a large set of mobile objects, distributed over some prespecified region, using a small number of mobile communication nodes. Such problems include placement of autonomous vehicles to perform distributed sensing tasks in a dynamic environment with the goal of achieving and maintaining a required sensor coverage criterion [1], [2]; the allocation and coordination of multiple resources and facilities in disaster response [3]; and placement and management of sensors in networks [4]. Specifically, these problems can be described in terms of dynamics of multiple elements in a domain Ω (to simplify the exposition, we consider $\Omega \subseteq \mathbb{R}^2$) with the main objective being to identify the *group dynamic* properties of these elements. In particular, at each time instance, we would like to determine a partition of the domain and identify a set of *resource locations* or *cluster centers*, such that each (effective) resource is assigned to a cell in the partition. The proceeding discussion gives a general sense of the *Dynamic Coverage Problem*, the main objective of which can be summarized as: identify *clusters* of mobile objects and determine the optimal locations and dynamics of resources such that these resources continuously provide adequate coverage of the mobile objects, throughout the time horizon of interest.

This work was partially supported by the NSF grant ECS - 0725708.

Yunwen Xu and C. L. Beck are with the department of Industrial and Enterprise Systems Engineering, S. Salapaka is with the department of Mechanical Science and Engineering. They are all in the University of Illinois at Urbana-Champaign, Urbana, Illinois, USA. Email: [xu27, salapaka, beck3]@illinois.edu

Dynamic coverage problems inherit the computational complexity of *facility location problems* that arise in a variety of *static* applications such as locational optimization, facility location, optimal coding, pattern recognition and learning, and data clustering and classification [5]–[8]. The static coverage problems are known to be NP-hard [9]–[12], where the cost functions are non-convex and are typically riddled with multiple local minima. Many popular algorithms, such as *Lloyd's* or *k-means* [13] are sensitive to the initial placement of resources, and typically get trapped in a local minimum. This complexity is compounded further by the inclusion of dynamics of constituent elements. Even in the static setting, there are relatively few algorithms that develop mechanisms to inhibit getting trapped at local minima and reduce sensitivity to initial conditions. One such algorithm is the Deterministic Annealing (DA) algorithm [14], which forms the foundation on which we develop our algorithm for the dynamic coverage problem. We will discuss the DA in more detail later in the paper.

Problems related to dynamic coverage are considered in [1], [2], [15]–[17], where the emphasis is on distributed implementations, i.e., under limited information flow between individual elements. These algorithms have the advantage of distributed implementation, however they are sensitive to the initial placement of the resources and suffer from drawbacks analogous to those found in Lloyd's algorithm. In contrast to this distributed approach, there is scant research that addresses the development of algorithms for problems of a non-distributed nature, or that aim simultaneously to attain global solutions and maintain low computational expense. In [18], [19], a maximum entropy principle (MEP)-based approach is discussed where dynamic coverage of mobile objects under given *velocity* fields is achieved by designing corresponding velocity fields for the resources.

The work in this paper has close parallels to [18], [19] in terms of problem formulation and objectives. By adopting a *free energy* function as a metric of coverage, the solutions associate each object to each resource with an association weighting parameter, hence successively overcoming the dependence on initial resource placements of distributed algorithms. These solutions are shown to be remarkably less expensive computationally than *frame-by-frame* methods, which implement *static*-clustering algorithms at fixed instances of time.

The novelty and practical importances of this work is that the trajectories of the resources can be manipulated through design of their acceleration fields, while in previous work only velocity fields were considered. Therefore this paper extends the class of dynamics of the mobile objects

to allow for general acceleration dynamics and control. This generalization is motivated mainly by multi-vehicle systems, such as those that may arise in disaster relief, search and rescue, and reconnaissance operations, since in these settings control is often implemented using thrust actuators. This modification on the assumptions about the problem dynamics results in considerable differences in the requirements on the control designs. Therefore in this work, we propose novel control laws aimed at satisfying the tracking requirements in such systems. This paper also presents a constructive *dynamic* control design that achieves coverage in this setting.

II. PROBLEM FORMULATION

A. Problem Setting

We consider a problem where the goal is to detect and track a group of moving objects (henceforth referred to as *sites*) in a given area. We form this as a dynamic coverage problem, where the aim is to successively identify locations and velocities of the resources, such that they provide adequate *coverage* of the moving sites at all times, that is the goal is to make resources track cluster centers. The sites may move in clusters and change cluster associations, and the clusters themselves can split and rejoin over time. An illustration is given in Figure 1.

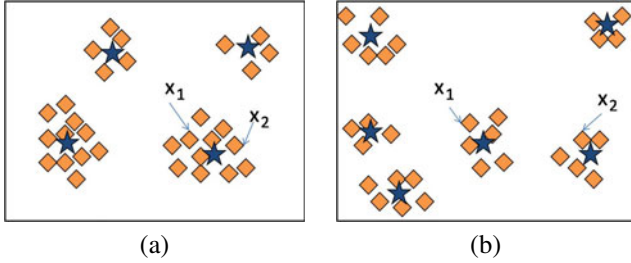


Fig. 1. Clustering moving objects in a given area. (a) and (b) denote two snapshots of an area with dynamic sites and resources. The squares and stars denote the positions of the sites and resources respectively. Sites x_1 and x_2 reside in the same cluster at the time instance shown in (a). A split occurs and causes them to reside in different clusters at the time instance shown in (b).

Specifically, for notational simplicity, we consider a domain $\Omega \subset \mathbb{R}^2$ (although our results extend directly to $\Omega \subset \mathbb{R}^n$) with N mobile sites and M resources, $N \gg M$, on a time horizon in $[0, +\infty)$. The site locations at time $t \in \mathbb{R}_+$ are given by $z_i(t) = [\xi_i(t) \ \eta_i(t)]^T \in \mathbb{R}^2, i = 1, \dots, N$ and the resource locations are given by $y_j(t) = [\rho_j(t) \ \omega_j(t)]^T \in \mathbb{R}^2, j = 1, \dots, M$. The mobile sites move under a (prescribed) continuous differentiable force field $\gamma(z, \dot{z}) \in (\mathbb{R}^2)^M$ with $\gamma_i \in \mathbb{R}^2$ denoting the force applied on the i th resource. Similarly, the resources' motions are driven by control input $u(t) \in (\mathbb{R}^2)^M$. Thus, the combined dynamics of sites and resources is represented by the state space equations

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \gamma(x_1(t), x_2(t)) \\ \dot{x}_3(t) &= x_4(t) \\ \dot{x}_4(t) &= u(t) \end{aligned} \quad (1)$$

where $x_1(t) = [z_1(t) \ z_2(t) \ \dots \ z_N(t)]^T$ and $x_3(t) = [y_1(t) \ y_2(t) \ \dots \ y_M(t)]^T$.

B. Metric

We adopt the concept of *distortion* and its variants from the data compression literature as a metric for coverage, and adapt this to make it suitable to a dynamic setting. Distortion, in a static coverage problem, is a measure of the (weighted) average distance of any site location to its nearest resource location; and the corresponding distortion minimization problem is given by

$$\min_{r_j} \sum_{i=1}^N p_i \min_{1 \leq j \leq N} d(s_i, r_j) \quad (2)$$

where $d(s_i, r_j) \in \mathbb{R}_+$ is the distance between the i th site s_i and j th resource r_j and p_i is a given distribution denoting the weight of the i th site. For static problems, $d(s_i, r_j)$ is typically chosen to be the square Euclidean distance $\|x_{1i} - x_{3j}\|^2$ [14], [20]–[23]. As a result, the resource locations y_j from solutions of (2) are located at the ‘centroid’ of the static clusters of sites z_i .

In the dynamic setting, we desire the resources to dynamically track the clusters, that is, not only should each resource y_j be near a centroid of the cluster but also its heading (velocity) should be representative of the cluster heading it represents. For instance, a resource that is at the centroid of a cluster but has a different velocity (magnitude or direction) than the average velocity of sites in the cluster cannot be thought of as *covering* (or *following* or *tracking*) the cluster. Accordingly, in the dynamic setting we define the *distance* function by

$$d(s_i, r_j) = \|x_{1i} - x_{3j}\|^2 + \theta \|x_{2i} - x_{4j}\|^2, \quad (3)$$

where the constant θ denotes a weight that characterizes the emphasis on the locational clustering relative to the velocity-based clustering. With this distance function, covering means that $[x_{3j}^T \ x_{4j}^T]^T$ is close to the corresponding cluster centroid position and cluster averaged velocity denoted by $[x_{3j}^{cT} \ x_{4j}^{cT}]^T$. Also note that this distance function can be interpreted as total energy in a mechanical system, where $\|x_{1i} - x_{3j}\|^2$ and $\|x_{2i} - x_{4j}\|^2$ represent *potential* and *kinetic* energies respectively.

Note that any change in position or velocity of a particular site i in the distortion function in (2) is reflected by changes in the distance $d(s_i, r_j)$ only with the *closest* resource j . This distributed aspect makes most algorithms (such as Lloyd’s) sensitive to the initial allocation of resources. In order to overcome this sensitivity, we modify this coverage function (along the lines of the DA algorithm [14]) by associating every site s_i to every resource r_j through weighting functions as shown below

$$D(s, r) = \sum_{i=1}^N p_i \sum_{j=1}^M p(r_j | s_i) d(s_i, r_j). \quad (4)$$

Here the weighting parameters $p(r_j | s_i) > 0$ can be chosen without loss of generality to satisfy $0 \leq p(r_j | s_i) \leq 1$ and $\sum_{j=1}^M p(r_j | s_i) = 1, i = 1, \dots, N, j = 1, \dots, M$. Note that we have replaced the *distance of a site to its closest resource* in (2) by the *weighted average distance of a site to all resources* in (4). The choice of the weights, $p(r_j | s_i)$, is

critical in assessing the trade-off between the decreasing local influence and the deviation of the distortion term (4) from the original cost function (2). We adopt MEP as the criterion determining a weighting function. The merit of the MEP is it provides a systematic method to compute a weighting function that achieves a specific feasible value of distortion, and thereby achieves a prespecified tradeoff in the above context [24], [25]. More specifically, we seek a weight distribution $\{p(r_j|s_i)\}$ that maximizes the Shannon Entropy $H(r|s) = -\sum_{i=1}^N p_i \sum_{j=1}^M p(r_j|s_i) \log(p(r_j|s_i))$, at a given (feasible) level of coverage indicated by the distortion $D(s, r) = D_0$. Therefore in this framework, we first maximize the unconstrained Lagrangian $H(r|s) - \beta(D(s, r) - D_0)$ w.r.t. $p(r_j|s_i)$, where β is a Lagrange multiplier. This unconstrained maximization problem has an explicit solution parametrized by β , known as the *Gibbs distribution*

$$p(r_j|s_i) = \frac{\exp\{-\beta d(s_i, r_j)\}}{\sum_{k=1}^M \exp\{-\beta d(s_i, r_k)\}}. \quad (5)$$

Remark: The above entropy maximization problem is equivalent to the following unconstrained minimization problem:

$$\min_{\{p(r_j|s_i)\}} D(s, r) - TH(r|s) \triangleq F \quad (6)$$

where $T \triangleq \frac{1}{\beta}$, which is the Lagrange version of distortion minimization under an entropy constraint. We call T and F *temperature* and *free energy* respectively, due to a close analogy to quantities in statistical thermodynamics [26]. On substituting this weight distribution (5) into the free energy term (6), we obtain the following cost function

$$F(y, \bar{y}) = -\frac{1}{\beta} \sum_{i=1}^N p_i \log \sum_{k=1}^M \exp\{-\beta d(s_i, r_k)\} \quad (7)$$

We replace distortion by free energy as a metric of coverage and the resource locations and velocities (x_3 and x_4) are obtained by minimizing F .

In the DA algorithm an *annealing* process is incorporated such that the minimization problem (6) is repeatedly solved at different values $\beta = \beta_k$, where β_k increases with k . For $\beta = 0$, this is entropy maximization, (a convex problem), and for $\beta = \infty$, the cost in (6) is identical to the distortion in (2); therefore this annealing process is insensitive to the initial allocation of resources and eventually achieves an allocation that closely approximates the true solution to (2).

In our work, we adapt the static DA algorithm to the dynamic setting. The following two key features of the DA algorithm allow this:

(1) *Phase-Transition Property:* The resource locations and velocities $r \in (\mathbb{R}^2)^M$ with $r_j = \{x_{3j}, x_{4j}\}$ for every j obtained by setting $\frac{\partial F}{\partial (x_3, x_4)} = 0$ give a local minimum for free energy F at every value of β , except at *critical temperatures* when $\beta = \beta_c$ (i.e. when $\frac{\partial^2 F}{\partial (x_{3j}, x_{3j})^2} = 0$). The critical temperature is given by $\beta_c^{-1} = 2\lambda_{\max}(C_{s|r_j})$ for some $1 \leq j \leq M$, where $\lambda_{\max}(\cdot)$ represents the largest eigenvalue of $C_{s|r_j} = \sum_{i=1}^N p(s_i|r_j)(s_i - r_j)(s_i - r_j)^T$. Moreover, the number of distinct locations when $\beta > \beta_c$ is always greater than the number when $\beta < \beta_c$.

(2) *Sensitivity-to-Temperature Property:* If the value of β is far from the critical values β_c , that is the minimum eigenvalue of $(I - 2\beta C_{s|r_j}) \geq \Delta > 0$ for $1 \leq j \leq M$, then $\|\frac{dr}{d\beta}\| \triangleq \left(\sum_j \|\frac{dr_j}{d\beta}\|^2\right)^{\frac{1}{2}} \leq c(\beta)/\Delta$ (see [19]), where $c(\beta)$ monotonically decreases to zero with β and is completely determined by β and the size of the space Ω .

These properties imply that between two consecutive critical temperatures, the number and locations of the resources are insensitive to changes in temperature value. We exploit this property in our algorithm and choose not to change the temperature value except near the critical conditions, thus reducing the number of computations, while not deviating much from the DA algorithm. Since tracking cluster centers is necessary for the occurrence of critical temperature conditions, the temperatures remain constant while the resources are far from the cluster centers. Once the cluster centers are tracked, the temperature values can be reduced to effect the critical temperature condition. As β crosses β_c , (x_3, x_4) is no longer a local minimum of F , and additional resource locations are required to capture a local minimum. This increase in the number of distinct resource locations is called *Splitting*. It should be noted that the algorithm induces *Splitting* only when identification of *finer* clusters is sought. If finer clusters are not sought, temperature values need not be changed thus avoiding the need for adding new resources (See [14], [18], [19] for details).

III. PROBLEM SOLUTION

The DA algorithm can be used directly to provide a “Frame-by-Frame approach” for the dynamic coverage problem by solving a static converge problem at each instance of time. However, this approach is computationally impractical. If we choose a large time interval for repeating the DA, the solution is incapable of providing smooth trajectories for resources. Alternatively, if we choose a small interval, the underlying sites do not move much, and neither do the corresponding cluster centers, which results in redundant iterations of the DA. We adopt Dynamic-Maximum-Entropy (DME) framework, proposed in [18], where β_k is changed only after the resources adequately track the cluster centers and more resolution is sought.

We make the following assumptions on the site dynamics and cluster-center definition:

Smoothness Assumption: The site-dynamics $\gamma(x_1, x_2)$ in (1) are continuously differentiable.

Positive Mass Assumption: All clusters have *positive mass*, that is $\min_{1 \leq j \leq M} p(r_j) > \nu > 0$.

Our goal is to make $x_3 \rightarrow x_3^c$ and $x_4 \rightarrow x_4^c$, where x_3^c and x_4^c are the instantaneous cluster position and velocity centers. Hence, these conditions require all resources to asymptotically track the cluster centers. In this paper, we approach this problem by making the derivative of the free energy (coverage metric), $\dot{F}(t) \leq 0$. Specifically, we aim to design the control law u in (1) to guarantee $\dot{F} = \frac{\partial F}{\partial \xi} \dot{\xi} \leq 0$. In fact, the control inputs steer the resources toward the position and velocity centroids of the clusters. Once the cluster centers are reached, our aim then is to split the resources

(by taking advantage of the phase transition property), when coverage with finer resolution is sought. In this paper we are primarily concerned with the design of the control law so that resources find and follow the cluster centers. Note that $\frac{\partial F}{\partial \xi}$ has the following structure, which we exploit for control design: $(\partial F / \partial \xi)^T = 2\Gamma \xi$, where

$$\Gamma = \begin{bmatrix} P_{1\otimes} & -P_{12\otimes} & & \\ & \theta P_{1\otimes} & -\theta P_{12\otimes} & \\ -P_{12\otimes}^T & & P_{2\otimes} & \\ & -\theta P_{12\otimes}^T & & \theta P_{2\otimes} \end{bmatrix}, \quad (8)$$

$P_1 \in \mathbb{R}^N, P_2 \in \mathbb{R}^M$ and $P_{12}, Q_1, Q_2 \in \mathbb{R}^{N \times M}$, with $P_1 = \text{diag}(p_1, \dots, p_N), P_2 = \text{diag}(p(r_1), \dots, p(r_M)), Q_1(i, j) = p(r_i | s_j), Q_2(i, j) = p(s_i | r_j) = p(r_j | s_i) / (\sum_l p(r_j | s_l))$ denoting the posterior probability matrix and $P_{12}(i, j) = p(s_i, r_j)$ the joint probability matrix. Then we have $P_{12} = P_1 Q_1 = Q_2 P_2$. Note that \otimes denotes the Kronecker product, and $P_{1\otimes}$ is short for $I_2 \otimes P_1$.

A. Static Control

The affine structure of the state equation (1) with respect to control u renders \dot{F} itself affine in u , that is, \dot{F} is of the form $a(x) + b(x)u$. The control authority is lost when $b(x) = 0$. Since $b(x)$ in this case is given by $(-x_2^T P_{12\otimes} + x_4^T P_{2\otimes}) = P_{2\otimes}(x_4 - Q_{2\otimes}^T x_2)^T$, $b(x) = 0$ only implies $x_4 = Q_{2\otimes} x_2$, (P_2 is diagonal positive definite and bounded away from zero). Thus when the resources attain the cluster center velocities, the static control becomes ineffective, regardless of their positions.

B. Dynamic Control

In order to avoid static tracking error, we design a dynamic controller in the form $\dot{x}_5 = \bar{u}$ and $x_5 = u$, where \bar{u} is the control input that we design and $x_5 = u$ is the augmented state. From now on we use $\xi_{cl} = [x_1^T x_2^T x_3^T x_4^T x_5^T]$ to denote the closed-loop states. By examining the above closed-loop system equation, it is easy to note that if x_3 and x_4 could be arbitrarily and separately controlled, then (x_1, x_3) and (x_2, x_4) form two independent second-order dynamic systems. Both systems are in the form presented in [18]. Motivated by this observation, we rewrite $\dot{x}_3 = v_f + (x_4 - v_f)$ and $\dot{x}_4 = u_f + (x_5 - u_f)$, where v_f and $u_f \in (\mathbb{R}^2)^M$ can be selected independently and then design \bar{u} to drive $x_4 - v_f$ and $x_5 - u_f$ to 0. In our design, we seek \bar{u} that makes $\dot{V} \leq 0$, where V is given by: $V(\xi) = F(\xi) + \frac{1}{2}x_5^T x_5$ and F is the system free energy given in (7). Note that V has the following properties.

1) *Bounded Below:*

$$F + \frac{1}{\beta} \log M = \frac{1}{\beta} \sum_i p_i \log \left\{ \frac{M}{\exp(-\beta d(s_i, r_j))} \right\} \geq 0$$

and $Q(t) \geq 0$. Thus the system energy term V is bounded below by $-\frac{1}{\beta} \log M$.

2) *Structured Derivative:* The time derivative of V is affine in \bar{u} (and is derived from (8)):

$$\dot{V} = \dot{F} + x_5^T \bar{u} = a_1 + b_1^T [v_f^T \ u_f^T]^T + a_2 + x_5^T \bar{u},$$

where

$$a_1 = 2 \begin{bmatrix} x_1 - Q_1 x_3 \\ \theta(x_2 - Q_1 x_4) \end{bmatrix}^T P_1 \begin{bmatrix} x_2 \\ \gamma \end{bmatrix}; \quad b_1 = 2 \begin{bmatrix} x_3 - Q_2^T x_1 \\ \theta(x_4 - Q_2^T x_2) \end{bmatrix}^T P_2$$

and

$$a_2 = 2 \begin{bmatrix} x_3 - Q_2^T x_1 \\ \theta(x_4 - Q_2^T x_2) \end{bmatrix}^T P_2 \begin{bmatrix} x_4 - v_f \\ x_5 - u_f \end{bmatrix}.$$

We note that $\dot{V}(t)$ becomes independent of the control \bar{u} only when $x_5 = 0$. Exploring this affine structure, we design v_f, u_f , and \bar{u} to guarantee $\dot{V} \leq 0$ whenever the resources are not at the cluster centers (i.e. $b_1 \neq 0$).

$$\begin{bmatrix} v_f \\ u_f \end{bmatrix} = \begin{cases} - \left[k_1 + \frac{a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2}}{b_1^T b_1} \right] b_1 & \text{if } b_1 \neq 0 \\ 0 & \text{if } b_1 = 0 \end{cases} \quad (9)$$

$$\bar{u} = \begin{cases} - \left[k_2 + \frac{a_2 + \sqrt{a_2^2 + (x_5^T x_5)^2}}{x_5^T x_5} \right] x_5 & \text{if } x_5 \neq 0 \\ 0 & \text{if } x_5 = 0 \end{cases} \quad (10)$$

in which k_1 and k_2 are arbitrary positive constants.

Remark: (9) and (10) should be determined in the order presented, since v_f and u_f appear in a_2 , and therefore are needed to determine \bar{u} .

We now state the main result of the paper, which shows the above control asymptotically tracks cluster center locations and velocities.

Theorem 1: For the system with site dynamics given by (1), under the *Smoothness and Positive Mass assumptions* and with v_f, u_f given by (9), the control input $\bar{u}(v_f, u_f, x_5)$ given by (10) achieves asymptotic tracking. That is $x_3 \rightarrow x_3^c$ and $x_4 \rightarrow x_4^c$ as $t \rightarrow \infty$ for all $j = 1, 2, \dots, M$.

Proof Let $Q(x_5) = x_5^T x_5$. Its derivative is given by

$$\dot{Q}(t) = \begin{cases} -k_2 x_5^T x_5 - \sqrt{a_2^2 + (x_5^T x_5)^2} - a_2 < 0 & \text{if } x_5 \neq 0 \\ 0 & \text{if } x_5 = 0 \end{cases}, \quad (11)$$

Therefore Q is positive definite, and \dot{Q} is negative definite. Since $Q(t)$ is a bounded monotonic function, there exists a finite constant Q_∞ such that $Q(t) \rightarrow Q_\infty$. Further, $Q(t)$ is of bounded variation, since $\int_0^\infty |\dot{Q}(t)| dt = -\int_0^\infty \dot{Q}(t) dt = Q(0) - Q_\infty < \infty$. As a result, we know $\lim_{t \rightarrow \infty} \dot{Q}(t) = 0$, which implies $\lim_{t \rightarrow \infty} x_5 = 0$ and $\lim_{t \rightarrow \infty} Q(t) = 0$.

Also since, \bar{u} given by (10) is Lipschitz at $x_5 = 0$ (see [27] and the discussion in Section V), Q and \dot{Q} are continuous functions. Since $x_5 \rightarrow 0$ as $t \rightarrow \infty$ and $\dot{Q} \leq 0$ from (11), these imply $a_2 \leq 0$ as $t \rightarrow \infty$. Moreover, this result is independent of the choice of v_f and u_f , or equivalently, independent of the selection of k_1 .

We now analyze tracking in two cases:

Case 1: $x_5 \rightarrow 0$ in infinite time. In this case, \bar{u} is always nonzero.

If $b_1 \neq 0$ for all finite t , then $\dot{V}(t) = -k_1 b_1^T b_1 - \sqrt{a_1^2 + (b_1^T b_1)^2} - k_2 x_5^T x_5 - \sqrt{a_2^2 + (x_5^T x_5)^2} \leq 0$, and V is a continuous function which is bounded below. Then, again by the bounded monotonic theorem, V_∞ exists with $|V_\infty| < \infty$, such that $\lim_{t \rightarrow \infty} V(t) = V_\infty$. $V(t)$ is also of bounded variation, that is $\int_0^\infty |\dot{V}(t)| dt = -\int_0^\infty \dot{V}(t) dt = V(0) - V_\infty < \infty$. Therefore $\lim_{t \rightarrow \infty} |\dot{V}(t)| = 0$. Since each term in \dot{V} is non-positive, $k_1 b_1^T b_1 \leq |\dot{V}(t)|$ for $k_1 > 0$, thus $b_1(t) \rightarrow 0$ as $t \rightarrow \infty$. Now as we assumed P_2 is positive definite with elements

bounded away from zero, we have $x_3(t) - Q_2^T x_1(t) \rightarrow 0$ and $x_4(t) - Q_2^T x_2(t) \rightarrow 0$, that is, the resources asymptotically track cluster centers. Similarly, $k_2 x_5^T x_5 \leq |\dot{V}(t)| \Rightarrow x_5(t) \rightarrow 0$ as $t \rightarrow \infty$, which is compatible with this case.

If $b_1(\hat{t}) = 0$ for some finite \hat{t} , then the centroid is achieved at that time instance and the control value at that time is irrelevant. However, if we desire $\dot{V} \leq 0$ when resources are at the cluster centers, then we require an additional assumption on *cluster consistency*: *The acceleration field imposed on the sites satisfies $(P_1 x_2)^T \bar{x}_1^c \leq 0$ and $(P_1 \gamma)^T \bar{x}_2^c \leq 0$ throughout the time horizon, where $\bar{x}_1^c := x_1 - Q_{1 \otimes} x_3^c$ and $\bar{x}_2^c := x_2 - Q_{1 \otimes} x_4^c$.* Note that $\bar{x}_1^c = x_1 - Q_{1 \otimes} Q_{2 \otimes}^T x_1$ and $\bar{x}_2^c = x_2 - Q_{1 \otimes} Q_{2 \otimes}^T x_2$ are essentially the vectors from each site to the weighted cluster centers (distances are in position and velocity, respectively and weighted by $p(r|s)$). Further since the weights $p(r|s)$ are given by the Gibbs distribution (5), $Q_{1 \otimes} x_3^c$ and $Q_{1 \otimes} x_4^c$ are dominated by the nearest cluster center. Therefore, this assumption simply states that as a site moves, the site direction and the vector to the average cluster center form an obtuse angle. Or equivalently, the sites move toward the weighted cluster center in an average sense. Based on this assumption, for any t with $b_1 = 0$, $\dot{V} = a_1^T + x_5^T \bar{u} = a_1^T - (k_2 + 1)x_5^T x_5$ is still negative. Thus the control input continues to steer the resources so as to decrease V . However, this additional result has relatively less significance, since once the resources are at the cluster centers, the focus should be on finding higher resolution (i.e. monitoring splitting conditions) rather than manipulating the resources dynamics.

Case 2: $x_5(t_1) = 0$ for some finite t_1 . In this case, $\bar{u}(t_1) = 0$ by (10) and $Q(t_1) = \dot{Q}(t_1) = 0$, which implies $x_5(t) \equiv 0$ for all $t > t_1$. Since otherwise if $x_5(t_2) \neq 0$ for $t_2 > t_1$, we must have $Q(t_2) > 0$, which contradicts $\dot{Q} \leq 0$. As a result, $\dot{x}_4 = x_5 \equiv 0$ for $t > t_1$, implying $x_4 \equiv x_{4c}$ for some constant vector x_{4c} .

Recall the property of $Q(t)$: as $t \rightarrow \infty$, $a_2(t) \leq 0$ for all v_f and u_f chosen by (9). Thus for any t with $b_1 \neq 0$, the expression $a_2|_{x_5=0} = b_1^T \begin{pmatrix} x_{4c} - v_f \\ -u_f \end{pmatrix} = b_1^T x_{4c} + k_1 b_1^T b_1 + a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2}$ is nonpositive. However, for any constant x_{4c} , the second term will dominate the first term by properly selecting the positive constant k_1 . So $b_1^T x_{4c} + k_1 b_1^T b_1 \geq 0$ and $a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2} \geq |a_1| + a_1 \geq 0$ for all $t > T$. Thus, a_2 will be greater than or equal to 0. To qualify both inequalities, a_2 must be zero for $t > T$, which implies $b_1 = 0$. Therefore even when x_5 is zero in finite time, the centroid condition still holds. \square

IV. SIMULATION

For the purpose of simulation, we chose a scenario with 64 mobile sites, each of which has the same weight, and which comprise four natural clusters (Fig. 2). Their initial velocities are chosen randomly, and their accelerations are chosen such that the clusters are maintained. (Note that the average velocity of each cluster is marked with an arrow).

To show the tracking results, we begin with a snapshot of a dynamic process which contains four resources. As the

sites move under the acceleration field, the algorithm progressively updates the association probabilities and computes v_f, u_f and \bar{u} using equation (9), (10), and thus generates the resource trajectories. The simulation result demonstrates that the resources asymptotically (identify and) track the cluster centers. When the cluster centers are tracked, the temperature is decreased to force splitting. To show the occurrence of resource splitting, we select a period that begins at very high temperature. At first, there is a single resource located at the centroid of the site positions (and velocities, which are not shown explicitly in the figure). As temperature decreases and cross the critical temperatures, a single resource splits into multiple distinct resources with lower associated weights. This process repeats and the system undergoes a series of phase transitions. Therefore, in this experiment, the resources move under the acceleration control input given by $\dot{x}_5 = \bar{u}(t)$, $u = x_5$, except at those time instances when the splitting occurs.

In this simulation, strong tracking performance (of the cluster centers) was achieved by the algorithm even when the site dynamics did not satisfy the consistency assumption. As a part of our ongoing work, we are analyzing if and how these assumptions can be relaxed, and are further testing our algorithm over a larger class of site dynamics.

V. ANALYSIS AND DISCUSSION

A. Static versus Dynamic Controllers

In the previous section, we discussed two control designs: static and dynamic. When the controls are effective, both designs render the derivatives of the system energy function negative, therefore pushing the resources toward the cluster centers. The key difference between the two designs is the effect resulting from losing control authority. The static control is disabled once the resources are at the velocity centers, while the dynamic control is effective as long as the position centers and velocity centers are not both reached simultaneously. In short, the dynamic control law is necessary to guarantee cluster tracking when the control input can not directly manipulate the resources velocity.

B. Lipschitz Continuity of the Control

We first define ξ_e of the closed-loop system to be such that $b_1 = 0$ and $x_5 = 0$. Equations (9), (10) suggest that, v_f, u_f and \bar{u} may become unbounded when ξ_{cl} is near ξ_e . However, we show that if there exists any \hat{v}_f, \hat{u}_f and \hat{u} which are Lipschitz at ξ_e , and guarantee that $\dot{V}_1 = a_1 + b_1 \begin{bmatrix} \hat{v}_f^T & \hat{u}_f^T \end{bmatrix}^T \leq 0$ and $\dot{V}_2 = a_2 + x_5^T \hat{u} \leq 0$, then the proposed v_f, u_f and \bar{u} are also Lipschitz at ξ_e .

This argument can be developed analogously to the proof of Proposition 3.43 in [27]. Since \hat{v}_f and \hat{u}_f are Lipschitz at ξ_e , there exist constants $r_1 > 0$ and $\delta > 0$ such that $\|\hat{v}_f^T \hat{u}_f^T\| \leq r_1 \|\xi_{cl} - \xi_e\|$ for all $\|\xi_{cl} - \xi_e\| < \delta$. Since $\dot{V}_1(\hat{v}_f, \hat{u}_f) \leq 0$, if $a_1(\xi_{cl}) > 0$, for $b_1 \neq 0$, we have $|a_1(\xi_{cl})| = a_1(\xi_{cl}) < -b_1 \begin{bmatrix} \hat{v}_f^T & \hat{u}_f^T \end{bmatrix}^T \leq r_1 \|b_1\| \cdot \|\xi_{cl} - \xi_e\|$. So $|b_1^T \begin{bmatrix} \hat{v}_f^T & \hat{u}_f^T \end{bmatrix}^T| \leq 2a_1 + (k_1 + 1)b_1^T b_1$. And $0 < \frac{a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2}}{b_1^T b_1} + k_1 \leq 1 +$

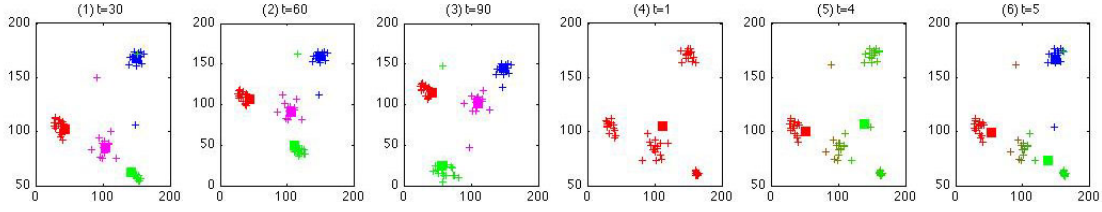


Fig. 2. Snapshots showing locations of mobile sites (plus symbols) and resources (square symbols). The left three figures display the resources tracking site dynamics as time progresses. The right three figures demonstrate the occurrence of splitting. Note that a unique resource (red square in (4)) gradually splits into two (a green square is added, while the color of plus symbols is a mixture of red and green), and then finally becomes three resources.

$k_0 + 2 \frac{r_1 \|\xi_{cl} - \xi_e\|}{\|b_1\|}$. Therefore, $\|\hat{v}_f^T \hat{u}_f^T\| \leq 2r_1 \|\xi_{cl} - \xi_e\| + (k_0 + 1)\|b_1\| \leq (2r_1 + r_2 + 1)\|\xi_{cl} - \xi_e\|$, where in the last equality, we use the fact b_1 is a bounded function of ξ_{cl} , thus we can find r_2 satisfying $\|b_1\| \leq r_2 \|\xi_{cl} - \xi_e\|$. If in the case that $a_1(\xi_{cl}) \leq 0$, then $0 \leq a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2} \leq b_1^T b_1$, and $\|\hat{v}_f^T \hat{u}_f^T\| \leq (a_1 + r_1)r_2 \|\xi_{cl} - \xi_e\|$. This completes the case for v_f and u_f , and similar results can be derived in the same fashion. Finally we note that we did not apply any additional assumptions to show the Lipschitz property, which indicates this is a general property of our control design.

C. Computational Complexity

The proposed approach avoids local minima by adopting a centralized approach, that is by associating each site with each resource through a weighting parameter given by the Gibbs distribution (5). Computing these exponential functions is one of the most costly computational parts of the algorithm. To address this, we consider approximating the Gibbs distributions by alternative lower-complexity distributions [28]. As well, as temperature decreases, the association between distant resources and sites becomes very small, and as $T \rightarrow 0$, the Gibbs distribution actually leads to nonzero association weights only with respect to the nearest resource. Therefore the algorithm can be approximated by gradually eliminating the influence of remote sites and redefining the algorithm so that we have, essentially, a local algorithm. Further approaches to improving scalability of the algorithm are under investigation.

REFERENCES

- [1] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage Control for Mobile Sensing Networks: Variations on a Theme," *Proceedings of Mediterranean Conference on Control and Automation*, July 2002.
- [2] E. Frazzoli and F. Bullo, "Decentralized Algorithms for Vehicle Routing in a Stochastic Time-varying Environment," *IEEE Conference on Decision and Control*, pp. 3357–3363, 2004.
- [3] F. Friedrich, F. Gehbauer, and U. Rickers, "Optimized resource allocation for emergency response after earthquake disasters," *Safety Science*, vol. 35, pp. 41–57, 2000.
- [4] W. Zhang, G. Xue, and S. Misra, "Fault-tolerant relay node placement in wireless sensor networks: problems and algorithms," in *Proceedings of IEEE INFOCOM*, Anchorage, AL, May 7-12 2007.
- [5] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, 1st ed. Kluwer, Boston, Massachusetts, 1991.
- [6] Z. Drezner, *Facility Location: A Survey of Applications and Methods*. New York: Springer Verlag, 1995.
- [7] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi Tessellations: Applications and Algorithms," *SIAM Review*, vol. 41, no. 4, pp. 637–676, December 1999.
- [8] P. Sharma, S. Salapaka, and C. Beck, "A Scalable Approach to Combinatorial Library Design for Drug Discovery," *Journal of Chemical Information and Modeling*, vol. 48, no. 1, pp. 27–41, 2008.
- [9] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, Nov 1984.
- [10] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k-means problem is np-hard," in *WALCOM '09: Proceedings of the 3rd International Workshop on Algorithms and Computation*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 274–285.
- [11] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "Np-hardness of euclidean sum-of-squares clustering," *Mach. Learn.*, vol. 75, no. 2, pp. 245–248, 2009.
- [12] P. Drineas, A. Friex, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Mach. Learn.*, vol. 56, pp. 9–33, 2004.
- [13] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [14] K. Rose, "Deterministic Annealing for Clustering, Compression, Classification, Regression and Related Optimization Problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–39, November 1998.
- [15] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage Control for Mobile Sensing Networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [16] D. Chudova, S. Gaffney, E. Mjolsness, and P. Smyth, "Translation-invariant Mixture Models for Curve Clustering," in *Proceedings of the 9th ACM SIGKDD*, 2003.
- [17] J.-B. Sheu, "A Fuzzy Clustering-Based Approach to Automatic Freeway Incident Detection And Characterization," *Fuzzy Sets Syst.*, vol. 128, no. 3, pp. 377–388, 2002.
- [18] P. Sharma, S. Salapaka, and C. Beck, "Entropy Based Algorithm for Combinatorial Optimization Problems with Mobile Sites and Resources," *Proceedings of American Control Conference*, pp. 1255–1260, 2008.
- [19] —, "Entropy-based framework for dynamic coverage and clustering problems," *accepted to IEEE Transactions on Automatic Control*, 2010.
- [20] —, "A deterministic annealing approach to combinatorial library design for drug discovery," *Proceedings of American Control Conference*, pp. 979–984, 2005.
- [21] —, "A scalable deterministic annealing algorithm for resource allocation problems," *Proceedings of American Control Conference*, pp. 3092 – 3097, 2006.
- [22] —, "A Maximum Entropy Based Scalable Algorithm for Resource Allocation Problems," *Proceedings of American Control Conference*, pp. 516–521, 2007.
- [23] S. Salapaka and A. Khalak, "Constraints on Locational Optimization Problems," *Proceedings of IEEE Control and Decision Conference*, pp. 1741–1746, December 2003.
- [24] E. T. Jaynes, "Information theory and statistical mechanics," *Physical Review*, no. 4, pp. 620–630, 1957.
- [25] —, *Probability Theory - The Logic of Science*. Cambridge University Press, 2003.
- [26] L. D. Landau and E. M. Lifshitz, *Statistical Physics, Part 1*, 3rd ed. Oxford, 1980, vol. 3.
- [27] R. Sepulchre, M. Jankovic, and P. Kokotovic, *Constructive Nonlinear Control*. Springer-Verlag, 1997. [Online]. Available: <http://www.montefiore.ulg.ac.be/services/stochastic/pubs/1997/SJK97a>
- [28] K. Demirciler and A. Ortega, "Reduced-Complexity Deterministic Annealing for Vector Quantizer Design," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 1807–1820, 2005.