# Clustering and Coverage Control for Systems With Acceleration-Driven Dynamics

Yunwen Xu, Srinivasa M. Salapaka, and Carolyn L. Beck

***Abstract*—In this technical note, we consider the dynamic coverage control problem from a clustering perspective, to which we apply control-theoretic methods to identify and track the cluster center dynamics. To the authors' knowledge, this is the first work to consider tracking cluster centers when the dynamics of the system elements involve *acceleration* fields. We show that a *dynamic* control design is *necessary* to achieve *dynamic* coverage under these acceleration fields. We pose the goal of maximizing the instantaneous coverage as a combinatorial optimization problem, and *propose* a framework that extends the concepts of the deterministic annealing algorithm to the dynamic setting. The resulting Lagrangian is used as a control Lyapunov function for designing coverage control. The algorithms we propose guarantee asymptotic tracking of cluster group dynamics, and we further establish continuity and boundedness of the corresponding control laws. Simulations are provided to corroborate these results.**

***Index Terms*— Clustering methods, deterministic annealing, dynamic coverage control, Lyapunov methods, optimization.**

## I. Introduction

The study of clustering and coverage control problems with dynamic objects arises in various applications, such as placement of autonomous sensors to perform distributed sensing tasks in a dynamic environment with the goal of achieving and maintaining a required sensor coverage criterion [1]–[3]; modeling CPU and database demands and their fluctuations in web-based software engineering [4]; and identifying the centroidal evolution in clusters within massive dynamic datasets containing varying features in database research [5]. These problems can be described in terms of dynamics of multiple elements (or *sites*) in a (general) domain with the main objective being to identify the *group dynamic* properties. Specifically, this goal can be viewed as a two-fold task: ( 1) partition the set of sites and place a *resource* in each cell of the partition, such that the averaged distance from a site to the nearest resource is minimized and 2) control the resource dynamics such that they track the corresponding dynamic cells.

The first task requires real-time decision-making and can be viewed as a *static* clustering or facility location problem, which has been studied in various contexts, such as coding, vector quantization and statistical learning [6]–[8]. These problems are typically posed as combinatorial optimization problems [9], where the cost functions are non-convex and are riddled with multiple local minima. Many popular algorithms, such as *Lloyd's* or $k$*-means* [10] are sensitive to the initial placement of resources, and typically get trapped in local minima. There are few algorithms with mechanisms to prevent being trapped in poor local minima and reduce sensitivity to initial conditions. One such algorithm is the deterministic annealing (DA) algorithm [8], which forms the foundation of our algorithm for the dynamic coverage problem. The second task adds additional computational and design challenges to the static problem by introducing site dynamics. Specifically, the main design challenges arise from the associated tracking problems of the dynamic clusters and their interactions.

Problems related to dynamic coverage are considered in [1]–[3], where the emphasis is on distributed implementations, i.e., under limited information flow between individual elements. Therefore, these algorithms are sensitive to the initial placement of the resources and suffer from drawbacks analogous to those found in Lloyd's algorithm. Also, these problems generally assume the underlying sites are static or fixed, and focus only on the dynamics of the resources; thus the resulting goal is to determine for each resource its optimal location and the path to reach its destination [2], [3]. In contrast to the distributed approach, there is scant research that seeks algorithms of a non-distributed nature that aim simultaneously to attain global solutions and maintain low computational expense in a dynamic environment. In [11] a Dynamic-Maximum-Entropy (DME) framework is proposed that treats dynamic coverage of mobile sites under given *velocity* fields by designing corresponding velocity fields for the resources.

The work in this technical note has close parallels to [11] in terms of problem formulation and objectives, however, we extend the class of site dynamics and develop altogether new control strategies. To successively overcome the local dependence of many distributed algorithms, we adopt a soft partitioning approach from the DA algorithm and associate each site to multiple resources via nonnegative association weights. The computational challenge induced by the underlying site dynamics is addressed by adopting an *energy function* that approximates the coverage metric as the *control Lyapunov function* of the system. Implementation of the proposed method is shown to be computationally less expensive than implementing repeated *static*-clustering of the data at fixed time instances.

The novelty and practical importance of this technical note is that it considers more realistic dynamics than [11], by allowing for general acceleration fields and control. Specifically, the trajectories of the resources can be manipulated through design of their acceleration fields, while in [11] only velocity fields are considered. This generalization is motivated mainly by multi-vehicle systems, such as those that may arise in disaster relief, search and rescue, and reconnaissance operations. This generalization in problem dynamics results in design challenges that are fundamentally different from those considered in [11]; we propose a new constructive *dynamic* control law that satisfies the tracking and coverage requirements in such systems (and note that the control law derived in [11] is static feedback, which is proved to be not sufficient here). A preliminary version of these results can be found in [12]. The main contributions of this technical note are summarized as follows.

1) Proof that a dynamic control law is necessary to track the centroidal dynamics of the mobile objects when their motions are driven by acceleration fields (note that in [11] a static control law is shown to suffice for the case of velocity fields);
2) Development of a nonlinear dynamic feedback control law for the resources, under which asymptotic tracking of cluster centers is achieved under mild conditions (see Theorem 1);
3) Proof that the constructive control law is non-conservative, that is, if asymptotic tracking can be achieved by some Lipschitz control near the instantaneous cluster center, then the proposed control is also Lipschitz and bounded (see Theorem 2).

Manuscript received December 10, 2012; revised May 26, 2013; accepted May 26, 2013. Date of publication November 25, 2013; date of current version April 18, 2014. This work was supported in part by the National Science Foundation under Grants ECS-0725708 and CMMI-1100257. Recommended by Associate Editor P. Shi.

The authors are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: xu27@illinois.edu; salapaka@illinois.edu; beck3@illinois.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2013.2292726

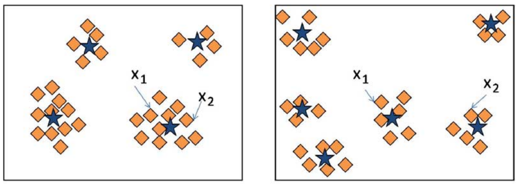


Fig. 1. Clustering moving objects. (a) and (b) denote two snapshots of an area with dynamic sites (squares) and resources (stars). Sites $x_1$ and $x_2$ reside in the same cluster at the time instance shown in (a). A split occurs and causes them to reside in different clusters at the time instance shown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

## II. Problem Formulation

### A. Problem Setting

For ease of illustration, we consider a geographical sensor coverage problem, where the goal is to detect and track a group of moving objects (or sites) in an area with a small number of autonomous vehicles (or resources) equipped with sensors. The locations and dynamics of the vehicles must be determined, such that the sensors continuously provide adequate *coverage* of the moving objects at all times. That is the sensors should track clusters of the moving objects when these objects may change cluster associations, and the clusters themselves can split and rejoin over time. (See Fig. 1.)

Let $\Omega \subset \mathbb{R}^2$ be a compact domain of interest; we note that the results directly extend to more general domains and higher dimensions. Suppose there are $N$ mobile sites denoted by $s$ and $M$ resources denoted by $r$ $(M \ll N)$, and the time horizon is $[0, +\infty)$. For arbitrary $t \geq 0$, let $z_i(t)$ and $y_j(t) \in \mathbb{R}^2$ denote the locations of the $i$th site and the $j$th resource, respectively, and $\dot{z}_i(t)$ and $\dot{y}_j(t)$ represent their instantaneous velocities. The sites move under a prescribed acceleration field $\gamma(z_1, \ldots, z_N, \dot{z}_1, \ldots, \dot{z}_N) = [\gamma_1^T, \gamma_2^T, \ldots, \gamma_N^T]^T \in \mathbb{R}^{2N}$, which is continuously differentiable, with $\gamma_i \in \mathbb{R}^2$ representing the acceleration of the $i$th site. Note that we have dropped the dependence on '$(t)$' for notational convenience unless required for clarification. Similarly, the resources move under a control acceleration field $u(t) = [u_1^T, u_2^T, \ldots, u_M^T]^T \in \mathbb{R}^{2M}$, which is to be designed. Thus, the combined dynamics of sites and resources is represented by the state space equations

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

where $x_1(t) = [z_1^T(t), z_2^T(t), \ldots, z_N^T(t)]^T \in \mathbb{R}^{2N}$ and $x_3(t) = [y_1^T(t), y_2^T(t), \ldots, y_M^T(t)]^T \in \mathbb{R}^{2M}$. We denote $x_{k_i}(t) = [x_{k_{2i-1}}(t), x_{k_{2i}}(t)]^T \in \mathbb{R}^2$ for $k = 1, \ldots, 4$. Let $\xi \triangleq [x_1^T, x_2^T, x_3^T, x_4^T]^T$ and $f = [x_2^T, \gamma^T, x_4^T, u^T]^T$, then the system (1) can be compactly written as $\dot{\xi} = f$.

### B. Coverage Metric

We adopt the concept of *distortion* from the data compression literature to measure instantaneous coverage (the lower the distortion, the better the coverage), and adapt this to a dynamic setting. For any time $t$, distortion is a measure of the average distance of a site to its nearest resource, given by

$$D(s, r) = \sum_{i=1}^{N} p_i \min_{1 \leq j \leq M} d(s_i(t), r_j(t)) \tag{2}$$

where $s_i(t) \triangleq [z_i^T(t), \dot{z}_i^T(t)]^T = [x_{1_i}^T(t), x_{2_i}^T(t)]^T \in \mathbb{R}^4$, $r_j(t) \triangleq [y_j^T(t), \dot{y}_j^T(t)]^T = [x_{3_j}^T(t), x_{4_j}^T(t)]^T \in \mathbb{R}^4$, $d(s_i(t), r_j(t)) \in \mathbb{R}_+$ denotes the distance between the $i$th site $s_i$ and $j$th resource $r_j$ and $p_i$ is a given positive constant (without loss of generality, we assume $\sum_i p_i = 1$) that denotes relative weight of the $i$th site. The distortion is defined for every time instance $t$ due to the implicitly assumed dynamics. For static coverage problems, $d(s_i, r_j)$ is typically chosen to be the squared Euclidean distance $\|x_{1_i} - x_{3_j}\|^2$. The resource locations, $y_j$'s (or $x_3$), that minimize (2) are at the 'centroid' of the clusters of sites $z_i$'s (or $x_1$) [8], [11], [13].

In the dynamic setting, we desire the resources to dynamically cover the clusters, that is, each resource $r_j$ should not only reach the instantaneous centroid of the cluster it represents at a particular time $t$, but should also consider the heading (velocity) of the cluster. For instance, a resource that is at the position 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 *tracking*) the cluster. Accordingly, in this technical note we define the *distance* by

$$d(s_i, r_j) = \left\|x_{1_i} - x_{3_j}\right\|^2 + \theta \left\|x_{2_i} - x_{4_j}\right\|^2 \tag{3}$$

where the weighting constant $\theta \geq 0$ characterizes the emphasis on the locational clustering relative to the velocity-based clustering. With this distance function, covering means that for every resource $j$, $r_j = [x_{3_j}^T, x_{4_j}^T]^T$ is close to the corresponding *cluster centroid position* and *cluster average velocity* (defined and quantified in Section III). Therefore, to maximize the coverage, we need to minimize the distortion $D(s, r)$ in (2) for all time $t$.

Note that any change in position or velocity of a particular site $i$ affects $d(s_i, r_j)$ only with respect to the *nearest* resource $j$ in the distortion (2). This distributed aspect makes most algorithms (such as Lloyd's) overly sensitive to the initial resource allocation. In order to overcome this sensitivity, we modify the distortion $D$ (following the DA algorithm [8]) by associating every site $i$ to every resource $j$ through an *association weight* $p(r_j|s_i)$:

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

Here we choose $\{p(r_j|s_i)\}$ to satisfy $0 \leq p(r_j|s_i) \leq 1$ and $\sum_{j=1}^{M} p(r_j|s_i) = 1$ without loss of generality. And thus we have replaced the *average distance of a site to its nearest resource* in (2) by the *weighted average distance of a site to all resources* in (4). The choice of $p(r_j|s_i)$ is critical in assessing the trade-off between decreasing the local influence and the deviation of $\bar{D}$ in (4) from the original distortion $D$ in (2). As in [8] and [11], we seek a weighting that achieves a trade-off between minimizing the distortion in (4) and maximizing the Shannon entropy term $H(r|s) = -\sum_{i=1}^{N} p_i \sum_{j=1}^{M} p(r_j|s_i) \log(p(r_j|s_i))$, which is equivalent to seeking $\{p(r_j|s_i)\}$ that minimize the *free energy*, or the Lagrangian, given by $F(r) \triangleq \bar{D}(s, r) - TH(r|s)$, where $T$ is a Lagrange multiplier, referred to as *temperature*. This yields a *Gibbs distribution*

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

with $\beta = 1/T$. By substituting the association weights (5), the free energy in (6) simplifies as

$$F(r) = -\frac{1}{\beta} \sum_{i=1}^{N} p_i \log \sum_{k=1}^{M} \exp\{-\beta d(s_i, r_k)\}. \tag{6}$$

In conclusion, we approach the instantaneous distortion minimization problem by solving an optimal assignment of resource locations and velocities that minimizes the free energy (6) with $d$ being specified in (3).

### C. An Iterative Solution

In the DA algorithm, an *annealing* process is incorporated such that the free energy minimization problem is repeatedly solved at a succession of values $\beta = \beta_k$, where $\beta_k$ increases with $k$. When $\beta \to 0$, this corresponds to entropy maximization, (a convex problem), and for $\beta \to \infty$, the association weights (5) become nearly binary and the free energy $F \approx \bar{D} \approx D$ in (2). Therefore the algorithm is insensitive to the initial resource allocations and eventually achieves an allocation that closely approximates a solution to (2).

The first order necessary condition of (6) (i.e., $\nabla_{(x_{3_j}, x_{4_j})} F = 0$) yields the stationary point

$$x^\star_{3_j} = \sum_{i=1}^{N} p(s_i|r_j) x_{1_i};\ x^\star_{4_j} = \sum_{i=1}^{N} p(s_i|r_j) x_{2_i},\ j = 1, \cdots, N \quad (7)$$

where $p(s_i|r_j) = [p(r_j|s_i)p_i]/p(r_j)$ and $p(r_j) = \sum_{i=1}^{N} p(r_j|s_i)p_i$ represent the posterior and total weight of a resource, respectively. These equations imply that $x^\star_3$ and $x^\star_4$ are the weighted centroids of all sites, where the weights are specified by $\{p(s|r)\}$. When the positions and velocities of all resources satisfy the centroid (7), we say the resources attain the *cluster centroids* and denote these states as $x^c_3$ and $x^c_4$. That is, $x^c_{3_j}$ and $x^c_{4_j}$ satisfy the following implicit equations:

$$x^c_{3_j} = \sum_{i=1}^{N} p(s_i|r^c_j) x_{1_i}, \quad x^c_{4_j} = \sum_{i=1}^{N} p(s_i|r^c_j) x_{2_i} \quad (8)$$

$$\text{and } p(r^c_j|s_i) = \frac{\exp\{-\beta d(s_i, r^c_j)\}}{\sum_k \exp\{-\beta d(s_i, r^c_k)\}}, \quad \forall j. \quad (9)$$

The DA algorithm exhibits a *phase-transition property* (see [8]). It has been shown that the solution in (7), parameterized by $\beta$, is indeed a local minimum of (6), except at a finite number of *critical values* $\beta_c$, (given by $\beta_c^{-1} = 2\lambda_{max}(C_{s|r_j})$ for some $1 \le j \le M$, where $\lambda_{max}(\cdot)$ represents the largest eigenvalue, and $C_{s|r_j} \triangleq \sum_{i=1}^{N} p(s_i|r_j)(s_i - r_j)(s_i - r_j)^T$). The number of *distinct* resource locations when $\beta > \beta_c$ is always greater than the number when $\beta < \beta_c$. Moreover, when the value of $\beta$ is far from $\beta_c$, the solution (7) is insensitive to the changes in the values of $\beta$ (see the *sensitivity-to-temperature property* derived in [11] for quantitative results).

Therefore, between two consecutive values of $\beta_c$, the number and assignment of the resources are insensitive to changes in the $\beta$ value. We exploit this insensitivity and choose not to change $\beta$ except near the critical conditions, thus greatly reducing the number of potential computations. Since tracking cluster centers is necessary for the occurrence of critical conditions, $\beta$ remains constant while the resources are far from the cluster centers. Once the cluster centers are reached, we increase $\beta$ to effect the critical condition. Here we assume that the time required to implement temperature changes in the algorithm is negligible compared to the time constant of the site dynamics. When $\beta$ surpasses the $\beta_c$ level, the resource(s) *split(s)*, that is the number of distinct resource locations increases and therefore more resources are required to track the minima of $F$. It should be noted that the algorithm induces *splitting* only when identification of *finer* clusters is sought. If finer clusters are not sought, $\beta$ values need not be changed thus avoiding adding new resources (See [11] for details).

## III. Control Design for the DME Framework

As proposed in the DME framework in [11], we update $\beta$ only after the resources adequately track the cluster centers and finer clustering is sought. Therefore, we focus on designing control laws that drive the resources to pursue the cluster centers. The central difference in the work we present herein versus [11] arises from the class of site-dynamics under consideration: in [11], all sites are moving under a velocity field, for which a *static* feedback control law governing the resource velocities is sufficient to drive all resources towards the cluster centers. In this article, we allow for more complicated site dynamics (1) by including acceleration fields, and seek a control law to guarantee asymptotic tracking of the cluster centroids; for this case we prove necessity of *dynamic* feedback control laws. We make the following assumptions regarding the site dynamics and cluster mass.

*A1 (Smoothness Assumption)*: The site dynamics $\gamma(\xi)$ in (1) are continuously differentiable.

*A2 (Positive Mass Assumption)*: All clusters have *positive mass*, that is, for all $j$, the weight of resource $r_j$ is bounded away from zero, or $p(r_j) \ge \varepsilon > 0$ for a constant $\varepsilon$.

The tracking objective can be interpreted as $x_3 \to x^c_3$ and $x_4 \to x^c_4$ as $t \to \infty$, with $x^c_3$ and $x^c_4$ being the instantaneous cluster position and velocity centers, as defined in (8). In other words, for a fixed value of $\beta$, we want to minimize the free energy (6) [thus minimize the distortion (4)] at all times. Following a similar argument as in [11], we approach this goal by designing a control $u$ in (1) that renders the free energy decreasing with time, that is $\dot{F} = (d/dt)F(t) \le 0$, whenever the cluster centers are not achieved ($[x_3; x_{4_j}] \ne [x^c_3; x^c_4]$). Once the cluster centers are reached, our aim then is to split the resources (by taking advantage of the phase transition property) if coverage with finer resolution is sought. Note that the derivative of the free energy function (6) is given by: $\dot{F} = (\partial F/\partial \xi)^T \dot{\xi} = 2\xi^T \Gamma f(\xi, \gamma, u)$, where

$$\Gamma = \begin{bmatrix} P_{1\otimes} & & -P_{12\otimes} & \\ & \theta P_{1\otimes} & & -\theta P_{12\otimes} \\ -P^T_{12\otimes} & & P_{2\otimes} & \\ & -\theta P^T_{12\otimes} & & \theta P_{2\otimes} \end{bmatrix} \quad (10)$$

with $P_1 \triangleq diag(p_1, \cdots, p_N) \in \mathbb{R}^{N\times N}$, $P_2 \triangleq diag(p(r_1), \cdots, p(r_M)) \in \mathbb{R}^{M\times M}$ and $[P_{12}]_{(i,j)} = p(s_i, r_j)$ denoting the joint association weights. We further define matrices $Q_1, Q_2 \in \mathbb{R}^{N\times M}$, where $[Q_1]_{(i,j)} = p(r_i|s_j)$ and $[Q_2]_{(i,j)} = p(s_i|r_j)$. Note that $P_{12} = P_1 Q_1 = Q_2 P_2$. As is standard, the notation $\otimes$ denotes the Kronecker product, and $P_{q\otimes} \triangleq I_2 \otimes P_q$ for $q \in \{1, 2, 12\}$.

### A. Necessity of Dynamic Control

We first show that unlike the velocity-field setting studied in [11], *static* feedback control laws are *not* sufficient to achieve the tracking objective in an acceleration-driven setting. Note that the derivative $\dot{F}$ is affine in $u$. More specifically, $\dot{F}$ is of the form $a(\xi) + b(\xi)u$, where $a(\xi) = 2x_1^T P_{1\otimes}(x_2 - Q_{1\otimes} x_4) - 2x_3^T P_{2\otimes}(Q^T_{2\otimes} x_2 - x_4) + 2\theta(x_2 - Q_1 x_4)^T P_1 \gamma$ and $b(\xi) = 2\theta(x_4 - Q^T_{2\otimes} x_2)^T P_{2\otimes}$. Since $\|b(\xi)\| = 2\theta \|P_2(x_4 - Q^T_{2\otimes} x_2)\| \ge 2\theta\varepsilon \|x_4 - Q^T_{2\otimes} x_2\|$ from *A2*, the control $u$ affects $\dot{F}$ only when the velocity centroid has not been reached (when $b(\xi) \ne 0$ or $x_4 \ne x^c_4$). That is, the control $u$ becomes ineffective when all resource velocities attain the cluster centroid velocities, regardless of their positions. Therefore the static control laws are insufficient for tracking cluster centers with these dynamics, for example, for initial conditions that satisfy $x_4 = x^c_4$, but $x_3 \ne x^c_3$.

### B. Constructive Dynamic Control Law

We propose a dynamic feedback control law in the form, $\dot{x}_5(t) = v(t)$; $u(t) = x_5(t)$, where we have introduced a new state variable $x_5$ and $v$ is the new design parameter. With this extension, the augmented system equation becomes $\dot{\xi}_{cl}(t) = g(\xi_{cl}, \gamma, v)$, where $\xi_{cl} \triangleq [x_1^T\ x_2^T\ x_3^T\ x_4^T\ x_5^T]^T$ denotes the closed-loop states, and $g = [x_2^T\ \gamma^T\ x_4^T\ x_5^T\ v^T]^T$. To complete our control design, we rewrite $\dot{x}_3 = \upsilon + (x_4 - \upsilon)$ and $\dot{x}_4 = \mu + (x_5 - \mu)$, where $\upsilon$ and $\mu$

are independently designed to control the states $x_3$ and $x_4$. We then design $v$ to drive $(x_4 - v)$ and $(x_5 - \mu)$ to 0. We use the following *augmented energy function* as the control Lyapunov function

$$V(\xi_{cl}) = F(\xi) + W(x_5) + \frac{1}{\beta}\log M \tag{11}$$

in which $F$ is the free energy given in (6) and $W(x_5) \triangleq (1/2)x_5^T x_5$. We seek $v$ that makes $\dot{V} \leq 0$ and thus guarantees asymptotic tracking. We can show that $V \geq 0$ since $V(\xi_{cl}) \geq F(\xi) + (1/\beta)\log M$ $=(1/\beta)\sum_i p_i \log\{(M/\sum_{j=1}^M \exp(-\beta d(s_i, r_j)))\} \geq 0$. Also the time derivative $\dot{V}$ is affine in $v$ since

$$\dot{V}(\xi_{cl}) = \underbrace{a_1(\xi) + b_1(\xi)^T \begin{bmatrix} v(\xi) \\ \mu(\xi) \end{bmatrix}}_{\dot{V}_1(\xi)} + \underbrace{a_2(\xi_{cl}) + x_5^T v(\xi_{cl})}_{\dot{V}_2(\xi_{cl})} \tag{12}$$

where

$$\begin{aligned} a_1(\xi) &\triangleq 2\begin{bmatrix} x_1 - Q_1 x_3 \\ \theta(x_2 - Q_1 x_4) \end{bmatrix}^T P_1 \begin{bmatrix} x_2 \\ \gamma \end{bmatrix} \\ b_1(\xi) &\triangleq 2P_2 \begin{bmatrix} x_3 - Q_2^T x_1 \\ \theta(x_4 - Q_2^T x_2) \end{bmatrix} \text{ and} \\ a_2(\xi_{cl}) &\triangleq b_1(\xi)^T \begin{bmatrix} x_4 - v \\ x_5 - \mu \end{bmatrix}. \end{aligned} \tag{13}$$

We exploit the affine structure in (12) to design $v, \mu$, and $v$ that guarantee $\dot{V} \leq 0$ whenever the resources are not at the cluster centers (i.e., $b_1(\xi) \neq 0$) and thus achieves asymptotic tracking. One such design can be constructed using the formulation proposed by Sontag in [14], giving us

$$\begin{bmatrix} v \\ \mu \end{bmatrix} = \begin{cases} -\left[k_1 + \frac{a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2}}{b_1^T b_1}\right] b_1 & \text{if } b_1 \neq 0 \\ 0 & \text{if } b_1 = 0 \end{cases} \tag{14}$$

$$\text{and } v = \begin{cases} -\left[k_2 + \frac{a_2 + \sqrt{a_2^2 + (x_5^T x_5)^2}}{x_5^T x_5}\right] x_5 & \text{if } x_5 \neq 0 \\ 0 & \text{if } x_5 = 0 \end{cases} \tag{15}$$

where $k_1$ and $k_2$ are arbitrary positive constants.

The following theorems state the main results of this technical note. Theorem 1 establishes that the control law $u$ based on $v$ in (14) asymptotically tracks cluster centroid locations and velocities.

*Theorem 1 (Asymptotic Tracking of Clusters):* For a system with site dynamics given by (1), under the assumptions A1 and A2, the control law $\dot{u} = v(v, \mu, x_5)$ with $v, \mu$ and $v$ given by (14) achieves asymptotic tracking. That is, $x_3 \to x_3^c$ and $x_4 \to x_4^c$ as $t \to \infty$ for all $j = 1, 2, \cdots, M$.

*Proof:* Note that $W(\xi_{cl}(t)) = (1/20x_5(t)^T x_5(t) \geq 0$ and $W(\xi_{cl}(t)) = 0$ only when $x_5(t) = 0$. Also $\dot{W}(\xi_{cl}(t)) = x_5(\xi_{cl}(t))^T v(\xi_{cl}(t)) \leq 0$ for all $t$ since $\dot{W}(\xi_{cl}(t)) =$

$$\begin{cases} -k_2 x_5^T x_5 - \sqrt{a_2^2 + (x_5^T x_5)^2} - a_2 < 0 & \text{if } x_5(t) \neq 0 \\ 0 & \text{if } x_5(t) = 0 \end{cases} \tag{16}$$

where $a_2$ is given in (13), and $\dot{W} < 0$ for all $x_5 \neq 0$. As a consequence, the real-valued function $W(\xi_{cl}(t))$ converges to a finite value $W_\infty$ as $t \to \infty$. Further, the smoothness of $\gamma$ ensures that $\dot{W}$ is of bounded variation, and since $\int_{t_0}^\infty -\dot{W}(\xi_{cl}(t))dt = W(\xi_{cl}(t_0)) - W_\infty < \infty$, we have $\dot{W}_\infty \triangleq \lim_{t\to\infty} \dot{W}(\xi_{cl}(t)) = 0$ from *Proposition A.1*. Therefore, from (16), $\lim_{t\to\infty} x_5(t) = 0$. Note that this result holds for any choice of positive constant $k_1$ in (14).

Now $x_5(t) \to 0$ can be studied under the following two cases. In Case 1, $x_5(t^*) = 0$ at some finite time $t^*$; in Case 2, $x_5(t) \neq 0$ at any finite time. We show that in both cases, as $t \to \infty$, $b_1(\xi(t)) \to 0$, or equivalently $x_3 \to x_3^c$ and $x_4 \to x_4^c$:

*Case 1: $x_5(t^*) = 0$ for $t^* < \infty$:* $x_5(t^*) = 0$ implies $v(t^*) = 0$ by (15), $W(\xi_{cl}(t^*)) = 0$, and $\dot{W}(\xi_{cl}(t^*)) = 0$ by (16). Then we conclude that $x_5(t) \equiv 0$ for all $t > t^*$, since otherwise, if $x_5(t') \neq 0$ for $t' > t^*$, $W(\xi_{cl}(t')) = (1/2)\|x_5(t')\|^2 > 0$, which yields $\dot{W} > 0$ for some $t^* < t < t'$ and contradicts $\dot{W} \leq 0$ for all $t$ (see (16)). Therefore, for all $t > t^*$, $\dot{x}_4 = x_5 \equiv 0$, implying $x_4 \equiv c_4$ for some constant vector $c_4$.

We then claim that there exists a time $T > t^*$, such that $a_2(\xi_{cl}(t)) = 0$ for all $t > T$. Since for all $t > t^*$, $x_5(t) = 0$ and $x_4(t) = c_4$, we can make $a_2(\xi_{cl}(t)) = a_2|_{x_5=0,x_4=c_4} = b_1^T c_4 + k_1 b_1^T b_1 + a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2} \geq 0$ by choosing a sufficiently large constant $k_1$. If this claim is false, we can find a sequence of time instances $\{t_k\} > t^*$ with $t_k \to \infty$ as $k \to \infty$ such that $a_2(\xi_{cl}(t_k)) > 0$, since $a_2$ is a continuous and bounded function of $t$. However, in this case, using (16) we can show that $\dot{W}(\xi_{cl}(t_k))|_{x_5=0} = -|a_2(t_k)| - a_2(t_k) = -2a_2(t_k) < 0$, which contradicts $\dot{W}(\xi_{cl}(t)) = 0$ for all $t > t^*$. Therefore, $a_2(\xi_{cl}(t)) = 0$ for all $t > t^*$. Note that since this equality is achieved only when $b_1(t) = 0$, we conclude that $b_1(t) = 0$ for all $t > t^*$.

*Case 2: $x_5 \to 0$ as $t \to \infty$ With $x_5(t) \neq 0$ for All $t < \infty$:* If $b_1(\hat{t}) = 0$ for some finite $\hat{t}$, then the cluster centroid is attained at the time instance $\hat{t}$, and no control authority is needed to *improve* tracking.

If $b_1(t) \neq 0$ for all finite $t$, that is, not all resources simultaneously track the cluster centers in finite time, then $\dot{V}(\xi_{cl}) = -k_1 b_1^T b_1 - \sqrt{a_1^2 + (b_1^T b_1)^2} - k_2 x_5^T x_5 - \sqrt{a_2^2 + (x_5^T x_5)^2} < 0$, for all $t < \infty$, thus $\dot{V}(\xi_{cl}(t)) \to 0$ as $t \to \infty$(from *Proposition A.1*). Since $k_1 > 0$, $0 \leq k_1 b_1^T b_1 \leq |\dot{V}(\xi_{cl})|$, we then have $\lim_{t\to\infty} b_1(t) = 0$. □

In the next theorem we show the control law (14) is non-conservative, that is, if there exists a Lipschitz control that makes $\dot{V} \leq 0$, then our design achieves the same.

*Theorem 2 (Boundedness of Control):* If there exists Lipschitz functions $\hat{v}, \hat{v}$ and $\hat{\mu}$ that asymptotically track cluster centers, that is, $\dot{V}_1(\xi) = a_1(\xi) + b_1(\xi)[\hat{v}^T \ \hat{\mu}^T]^T \leq 0$ and $\dot{V}_2(\xi_{cl}) = a_2(\xi_{cl}) + x_5^T \hat{v} \leq 0$ (see (12) for the definition of $V_1$ and $V_2$) whenever $\xi_{cl} \neq \xi_e \triangleq [x_1^T \ x_2^T \ x_3^{cT} \ x_4^{cT} \ 0]^T$, then $v$, $\mu$ and $v$ given in (14) and (15) also track the cluster centers and are Lipschitz.

Note that $a_1$, $a_2$, $b_1$, $x_3^c$ and $x_4^c$ in Theorem 2 are defined in (13) and (8). The proof of Theorem 2, which is analogous to the proof of [15, Proposition 3.43], is provided in Appendix B, where we exploit the algebraic structure of the control defined by (14) and (15). Once the tracking cluster objective is achieved, we exploit the phase-transition property to effect resource splitting (is desired) to achieve better coverage as discussed in Section II. If no higher resolution coverage is sought, resources continue to track the cluster centroids.

## IV. Simulation

We consider a scenario with 64 mobile sites, each of which has the same weight, and which comprise four natural clusters (Fig. 2). We choose site dynamics such that all sites have zero initial velocities, and sites within one cluster have similar accelerations so that the clusters are maintained. The initial acceleration of each cluster is indicated by an arrow in plot (a); the individual site accelerations are generated by adding small random perturbations on those cluster accelerations. Plots (a)–(h) record the featured time instances, in which plots (a)–(d) demonstrate that our DME algorithm seeks better coverage through resource

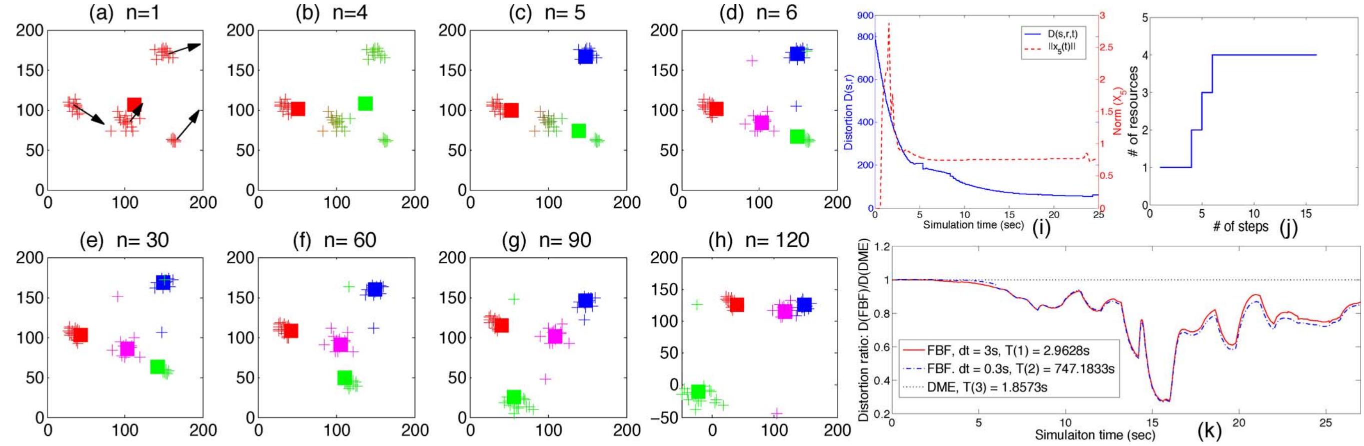


Fig. 2. Plots (a)–(h) show snapshots of a domain with mobile sites (plus symbols) and resources (square symbols). The nominal initial accelerations of natural clusters are given by $\gamma_{n1}(0) = [0.05; 0.12]$, $\gamma_{n2}(0) = [-0.02; 0.09]$, $\gamma_{n3}(0) = [-0.01; -0.17]$ and $\gamma_{n4}(0) = [-0.61; -0.29]$, respectively, whose directions and magnitudes are indicated by arrows and their lengths in plot (a). The accelerations of individual sites are from normal distributions around those nominal values. At $n = 1$, the algorithm is initiated at a high temperature value ($T = 5000$) and places all resources at the velocity and position centroid of all sites ($x_3(1) = x_3^c(1)$ and $x_4(1) = x_4^c(1)$) since all association weights $p(r_j|s_i)$ are nearly uniform. Hence the centroid condition is automatically satisfied. To improve coverage, more distinct resource locations are added through phase transitions, i.e., by increasing the $\beta$ value, until the resolution requirement is reached [plots (b)–(d)]. The color of a site is determined by the average rgb value of all resources, weighted by the association weights. The algorithm progressively computes $v, \mu$ and $v$ to adjust resource dynamics, and drives resources towards cluster centers [plots (e)–(h)], until all cluster centers are identified and tracked (plots (g), after 90 time steps). The temperature value remains the same during this period. Plot (i) depicts the distortion $D$ as time progresses, and it also provides the value changes of the control energy $\|x_5\|$, and its fluctuation indicates the effort used to correct resource dynamics for tracking. Initially, single-cluster placement achieves perfect tracking so no acceleration adjustment is needed. After several resource splits, the resources chase the cluster centers by modifying their accelerations (the peak of $\|x_5\|$). When all cluster centers are attained, the resource accelerations converge to the centroid accelerations, and $\|x_5\|$ tends to a steady-state value. Plot (j) shows the number of distinct resource locations. The comparison between the proposed DME algorithm and the FBF approaches is shown in plot (k), during which process we assume no occurrence of resource splittings. The distortion $D$ achieved by the DME algorithm is comparable with the FBF method using small time steps, while the computational time of DME is much smaller (1.8573 seconds versus 347.1833 seconds for the latter).

splits until the coverage requirement is met. Note that a single resource location gradually splits into two, three, and then finally becomes four resource locations (that is, all 4 clusters are successfully identified). Upon splitting, the instantaneous cluster centroid positions and velocities are assigned to the corresponding new resources. The subsequent deviations of the resources from the cluster centers due to initial mismatch of their accelerations from that of respective clusters, are corrected by using the tracking control in (15) and (16) [plots (d)–(g)].

Plot (k) compares the computational effort for tracking of cluster centers of our DME algorithm with a frame-by-frame (FBF) method, where DA-based static clustering is used in a predetermined sequence of sampled time instants. In this simulation, we applied the FBF method two times—the first with fewer time instants (10) and the second with many time instants (100), using static clustering in both cases. The DME algorithm, when compared to the FBF approach with 100 time instants, gives comparable distortion (within 88.79%) after a transient time of 25 sec [see plots (i) and (k)] while taking only 5.35% of the computation time. The DME algorithm takes 62.69% computation time when compared to the FBF with fewer time instants (see caption for details).

Note that the comparison of the resulting distortion values in the above simulations is conservative since we have assumed that the results of static clustering at each time sample of the FBF method is applied *instantaneously*. If we account for the drift of the sites during the computation of the static DA algorithm, the resulting distortions from the FBF method will be more likely to be larger.

## V. Analysis and Discussion

### A. Flexibility in Implementation

The DME framework presented herein enables cluster splits for better coverage as discussed in Section III. However, it does not explicitly monitor cluster mergers. An easy way to accommodate cluster mergers is to track the pairwise distance for all pairs of resources $d(r_i, r_j)$, and combine two resources if $d(r_i, r_j) < \sigma$ for some threshold $\sigma$. Alternatively, we can add a cost term to the distortion function that penalizes nearby resources. An example of one such term is $\sum_{i,j} -(1/\beta) \log d(r_i, r_j)$ under which two nearby resources will induce a large cost that will increase as the annealing process continues ($\beta$ increases), making splitting less likely. Inclusion of cluster mergers in our framework is part of our ongoing work.

The DME framework provides a fundamental algorithmic structure for dynamic coverage problems, where the notion of coverage is not limited to geographical objective functions and can easily be extended to include domain specific objectives for other problems. For example, in web-based software engineering, the coverage can be defined in terms of the *quality of service* function [4]; in network routing problems, where decisions on both the locations of communication centers and the routes are required, both distortion and communication costs can be accounted for in the distance function [16].

### B. Computational Complexity

The proposed method adopts the DA algorithm to solve the clustering objective whenever $\beta$ changes. Therefore the association weights given by (5) need to be calculated for all site-resource pairs, which can be computationally expensive. To reduce computational effort, we can approximate the Gibbs distributions by lower-complexity distributions [17]. Since the association between pairs of distant resources and sites becomes very small as $\beta$ increases, we can also gradually eliminate the influence of remote sites and perform progressively distributed computations, leading to a more tractable algorithm. Further approaches to improving scalability of the algorithm are under investigation, and some preliminary results are given in [18].

## APPENDIX

### A. Convergence of Functions With Bounded Variation

*Proposition A.1:* Let $\phi : \mathbb{R} \rightarrow \mathbb{R}$ be a nonnegative function of bounded variation on $[0, \infty)$. Suppose that $\lim_{t\rightarrow\infty} \int_{r=0}^{t} \phi(r)dr$ exists and is finite, then $\lim_{t\rightarrow\infty} \phi(t) = 0$.

*Proof:* If zero is not the limit, then there exists a $\delta > 0$ and a sequence $\{t_n\}$ with $t_n - t_{n-1} > 1$, such that $\phi(t_n) > \delta$ for all $n$. Moreover, since $s(t) \triangleq \int_{r=0}^{t} \phi(r)dr$ is increasing, and $s(t) \leq \bar{s} \triangleq \lim_{t\rightarrow\infty} s(t) < \infty$ as assumed, there exists a subsequence $\{t_{n_k}\} \subset \{t_n\}$, such that $|s(t_{n_k}) - \bar{s}| < (1/2)\delta, \forall k$. Let $\check{\phi}(n_k) \triangleq \min_{t_{n_k} \leq r \leq t_{n_{k+1}}} \phi(r)$, we have $\check{\phi}(n_k) < \check{\phi}(n_k)(t_{n_{k+1}} - t_{n_k}) \leq \int_{r=t_{n_k}}^{t_{n_{k+1}}} \phi(r)dr \leq |s(t_{n_k}) - \bar{s}| < (1/2)\delta$. Then $(1/2)\delta \leq \sup_{t_{n_k} \leq r_1, r_2 \leq t_{n_{k+1}}} |\phi(r_1) - \phi(r_2)|$. And the total variation of $\phi$ on interval $[0, t_N] \subset [0, \infty)$ is $T_0^{t_N} = \sup \sum_{i=1}^{N'} |\phi(r_i) - \phi(r_{i-1})| \geq \sum_{k=1}^{N} \phi(n_k) - \check{\phi}(r_k) \geq (1/2)\delta N$, in which the supremum is taken over all subdivisions $0 < r_1 < \cdots < r_{N'} = t_N$ with $N' \in \mathbb{N}$. This contradicts the bounded variation assumption. □

### B. Proof of Theorem 2 (Boundedness of Control)

Since $\hat{\upsilon}$, $\hat{\mu}$ and $\hat{\upsilon}$ are Lipschitz at $\xi_e$ by assumption, there exists an $r_1 > 0$ and $\delta > 0$ such that $\|[\hat{\upsilon}^T\ \hat{\mu}^T\ \hat{\upsilon}^T]^T\| \leq r_1\|\xi_{cl} - \xi_e\|$ for all $\xi \in \Omega_\delta \triangleq \{\xi_{cl} : \|\xi_{cl} - \xi_e\| < \delta\}$, thus $\max\{\|(\hat{\upsilon}^T\ \hat{\mu}^T)^T\|, \|\hat{\upsilon}\|\} \leq r_1\|\xi_{cl} - \xi_e\|$ in $\Omega_\delta$.

Case 1) $a_1(\xi) > 0$ and $b_1 \neq 0$. Since $\dot{V}_1(\hat{\upsilon}, \hat{\mu}) \leq 0$ by assumption, then $a_1(\xi) < -b_1[\hat{\upsilon}^T\ \hat{\mu}^T]^T \leq r_1\|b_1\|\|\xi_{cl} - \xi_e\|$ in $\Omega_\delta$ (12). This implies $\eta \triangleq (a_1/\|b_1\|^2) \leq (r_1\|\xi_{cl} - \xi_e\|/\|b_1\|)$. So we have $0 < ((a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2})/b_1^T b_1) + k_1 \leq k_1 + 1 + 2\eta \leq 1 + k_1 + 2(r_1\|\xi_{cl} - \xi_e\|/\|b_1\|)$. For $\upsilon$ and $\mu$ selected by (14) when $b_1 \neq 0$, we have $\|[\upsilon^T\ \mu^T]\| \leq \|k_1 + ((a_1 + \sqrt{a_1^2 + \|b_1\|^4})/\|b_1\|^2)\| \cdot \|b_1\| \leq (k_1 + 1)\|b_1\| + 2r_1\|\xi_{cl} - \xi_e\| \leq [2r_1 + (k_1 + 1)r_2]\|\xi_{cl} - \xi_e\|$, where in the last inequality, we use the fact $\|b_1\| \leq \|P_2\| \cdot \|\begin{bmatrix} x_3 - Q_2^T x_1 \\ \theta(x_4 - Q_2^T x_2) \end{bmatrix}\| \leq r_2\|\xi_{cl} - \xi_e\|$, for some $r_2 > 0$. Let $r_3 \triangleq [2r_1 + (k_1 + 1)r_2]$.

Case 2) $a_1(\xi) \leq 0$ and $b_1 \neq 0$, we have $0 \leq a_1 + \sqrt{a_1^2 + (b_1^T b_1)^2} \leq a_1 + |a_1| + \|b_1\|^2 = \|b_1\|^2$, and $\|[\hat{\upsilon}^T\ \hat{\mu}^T]\| \leq (1 + k_1)\|b_1\| \leq (1 + k_1)r_2\|\xi_{cl} - \xi_e\|$. Let $r_4 \triangleq (1 + k_1)r_2$.

In both cases, $\upsilon, \mu$ as defined in (14) are Lipschitz with constant $R \triangleq \max\{r_3, r_4\}$ in $\Omega_\delta$. $\upsilon$ is similarly shown to be Lipschitz. Since $\upsilon$ is Lipschitz in $\Omega_\delta$ and bounded outside $\Omega_\delta$ from (14), the boundedness of control follows.

## ACKNOWLEDGMENT

The authors would also like to thank the Associate Editor and the anonymous reviewers for their helpful suggestions.

## REFERENCES


[1] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *Proc. 43rd IEEE Conf. Decision and Control*, Dec. 2004, vol. 4, pp. 3357–3363.

[2] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, 2004.

[3] I. I. Hussein and D. M. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 4, pp. 642–657, 2007.

[4] H. Ghanbari, C. Barna, M. Litoiu, M. Woodside, T. Zheng, J. Wong, and G. Iszlai, "Tracking adaptive performance models using dynamic clustering of user classes," in *Proc. 2nd Joint WOSP/SIPEW Int. Conf. Performance Engineering*, 2011, pp. 179–188.

[5] Z. Zhang, Y. Yang, A. Tung, and D. Papadias, "Continuous k-means monitoring over moving objects," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1205–1216, Sep. 2008.

[6] Z. Drezner, *Facility Location: A Survey of Applications and Methods*. New York: Springer-Verlag, 1995.

[7] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, 1st ed. Boston, MA: Kluwer, 1991.

[8] K. Rose, "Deterministic annealing for clustering, compression, classification, regression and related optimization problems," *Proc. IEEE*, vol. 86, no. 11, pp. 2210–2239, Nov. 1998.

[9] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[10] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 2, pp. 129–137, 1982.

[11] P. Sharma, S. M. Salapaka, and C. L. Beck, "Entropy-based framework for dynamic coverage and clustering problems," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 135–150, Jan. 2012.

[12] Y. Xu, S. Salapaka, and C. L. Beck, "Dynamic maximum entropy algorithms for clustering and coverage control," in *Proc. 49th IEEE Conf. Decision and Control*, Dec. 2010, pp. 1836–1841.

[13] S. M. Salapaka, A. Khalak, and M. Dahleh, "Constraints on locational optimization problems," in *Proc. 42nd IEEE Conf. on Decision and Control*, Dec. 2003, vol. 2, pp. 1741–1746.

[14] E. D. Sontag, "A 'universal' construction of Artstein's theorem on nonlinear stabilization," *Syst. and Control Lett.*, vol. 13, no. 2, pp. 117–123, 1989.

[15] R. Sepulchre, M. Jankovic, and P. Kokotovic, *Constructive Nonlinear Control*. New York: Springer-Verlag, 1997.

[16] N. V. Kale and S. M. Salapaka, "Maximum entropy principle-based algorithm for simultaneous resource location and multihop routing in multiagent networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 4, pp. 591–602, Apr. 2012.

[17] K. Demirciler and A. Ortega, "Reduced-complexity deterministic annealing for vector quantizer design," *EURASIP J. Advances Signal Proces.*, vol. 2005, pp. 1807–1820, 2005.

[18] P. Sharma, S. M. Salapaka, and C. L. Beck, "A scalable approach to combinatorial library design for drug discovery," *J. Chem. Inform. and Modeling*, vol. 48, no. 1, pp. 27–41, 2008.