Artif Intell Rev
DOI 10.1007/s10462-011-9248-4

Coverage control in unknown environments using neural networks

Alireza Dirafzoon · Saba Emrani · S. M. Amin Salehizadeh · Mohammad Bagher Menhaj

Abstract This paper proposes a distributed adaptive control algorithm for coverage control in unknown environments with networked mobile sensors. An online neural network weight tuning algorithm is used in order for the robots to estimate the sensory function of the environment, and the control law is derived according to the feedforward neural network estimation of the distribution density function of the environment. It is distributed in that it only takes advantage of local information of each robot. A Lyapunov function is introduced in order to show that the proposed control law causes the network to converge to a near-optimal sensing configuration. Due to neural network nonlinear approximation property, a major advantage of the proposed method is that in contrary to previous well known approaches for coverage, it is not restricted to a linear regression form. Finally the controller is demonstrated in numerical simulations. Simulation results have been shown that the proposed controller outperforms the previous adaptive approaches in the sense of performance and convergence rate.

Keywords Mobile sensor networks · Neural networks · Coverage control · Lyapunov stability analysis

1 Introduction

Sensor networks have a broad application in environmental sampling, ecosystem monitoring and military surveillance, which call for effective network control methods and coordination

A. Dirafzoon (✉) · S. Emrani · S. M. A. Salehizadeh · M. B. Menhaj
Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran
e-mail: adirafzoon@ieee.org

S. Emrani
e-mail: saba.emrani@gmail.com

S. M. A. Salehizadeh
e-mail: smasalehizadeh@gmail.com

M. B. Menhaj
e-mail: tmenhaj@ieee.org

Published online: 27 July 2011

tools to achieve specific performances. Especially in term of sensor network coverage, the following questions need to be addressed, how to deploy a sensor network, how to setup proper protocols among sensor agents to control the information flow, and how to efficiently maximize the coverage area with service quality ensured. The deployment of large groups of autonomous vehicles is rapidly becoming possible because of technological advances in networking systems.

Mobile sensor network, comparing to the static sensor network, has the following promising advantages. First, it offers the flexibility and convenience for the network to reorganize according to dynamic changes; for example, once a suspicious target is found, the whole mobile sensor network is able to rearrange its layout according to the detection. Second, mobility facilitates the sensor carriers loading with highly accurate and precise sensors, which provides insight into the trade-off between rich information and limited computation capability/expense budget. Third, mobile sensor network provides utility maintain service (Wang and Guo 2008).

One of the fundamental problems of multi-robot control is how to deploy a group of robots to spread out over an environment to carry out sensing, surveillance, data collection, or distributed servicing tasks. This operation is called coverage control, and several methods have been proposed to accomplish it in a distributed and efficient way.

The coverage problem is a fundamental issue of a sensor network system. Researchers have proposed various solutions to a lot of interesting sensor network coverage problems. In Du et al. (1999), the authors introduce the centroidal Voronoi tessellations as a comprehensive solution to a series of partition problems which sheds light on the sensor coverage problem, and also provide a centralized algorithms under deterministic and probabilistic domains. Cortes et al. (2004) proposes a decentralized control law for multi-robot coverage of an area partitioned into Voronoi diagram, in the sense that continually driving the robots toward the centroids of their Voronoi cells. A recent text that presents much of this work in a cohesive fashion is Bullo et al. (2008) and an excellent overview is given in Martinez et al. (2007). Different extensions of the framework devised in Cortes et al. (2004) have been proposed in the literature. In Cortez et al. (2005) the problem of limited-range interaction between agents was addressed. In Kwok and Martinez (2008) the basic approach was extended to deal with agents with limited energy. The problem of learning the distribution density functions online, while moving toward the optimal locations was addressed in Schwager et al. (2006, 2007). There has been another extension to heterogeneous groups of robots and non-convex environments in Pimenta et al. (2008). Coverage controllers also have been successfully implemented on robotic systems in Schwager et al. (2009, 2008). There are also a number of other notions of multi-robot sensory coverage (e.g. Choset 2001; Latimer et al. 2002; Butler and Rus 2004; Gren et al. 2004). In Hussein and Stipanovic (2007), address the dynamical coverage control law over a given bounded region, given the fleet fully connected; it may reach effective coverage with collision avoidance under different sensing models and parameters. In Li and Cassandras (2005), the authors propose a distributed coverage control scheme based on a probabilistic sensing range model to maximize the joint detection probability and minimize the communication cost. In this paper, we use the notion of an optimal sensing configuration developed by Cortes et al. (2004) and propose a neural network based controller for near-optimal coverage in an unknown environment that causes a network of robots to spread out over an environment while aggregating in areas of high sensory interest. This is in contrast to other works (e.g. Cortez et al. 2005; Salapaka et al. 2003; Pimenta et al. 2008) in which the distribution of sensory information in the environment is required to be known a priori by all robots. This a priori requirement was first relaxed in Schwager et al. (2006) by introducing a controller with a simple memory less approximation from

sensor measurements. It was removed by introducing an adaptive controller in Schwager et al. (2007). This work improved in Schwager et al. (2008) by introducing a consensus (or flocking) term in the adaptation law to allow sharing of parameters among neighbors resulted in increasing parameter convergence rates.

The main drawbacks of these works are that firstly they are only possible to proceed if the sensory function is linear in the known parameters. The second drawback is due to the relatively large amount of prior information which must be incorporated into these designs in the form of the exact structure of the required basis functions. Each robot needs to have the vector of basis functions in the regression matrix available for the computation of its control law, because it uses a copy of the vector of basis functions at each time step to compute its estimation of the sensory function. In contrast, in our proposed controller, we get rid of the first problem using the neural network approximation property for nonlinear functions. Also, applying neural networks for estimation of this function, the robots do not have to know anything either about the structure of that function or the value of basis functions in the regression matrix. The only information each robot needs to know about the environment is the value of the sensory function at its position.

Our proposed controller is fully decentralized, adaptive in the sense of the changes in the environment and the network of the robots, provably convergent. Any application in which a group of automated mobile agents is required to provide collective sensing over an environment can benefit from the proposed control law. The most important feature of our controller is that robots learn a representation of the sensing environment in a closed loop way without a prior knowledge about the structure of the sensory function.

The remainder of the paper is organized as follows. Section 2 provides some background on the geometrical coverage control problem setup and the structure of the neural network. In Sect. 3, we present the proposed neural network based controller, and a stability analysis of the proposed controller is reported at the end of the section. Numerical simulation results are described in Sect. 4. Finally, we conclude the paper in Sect. 5.

2 Problem setup

In this section, we state the basic definitions and results from locational optimization that will be useful in this work. More thorough discussions were given by Cortes et al. (2004) and Schwager et al. (2006).

2.1 Locational optimization

Let there be n robots in a bounded, convex environment $Q \in \mathbb{R}^2$. An arbitrary point in Q is denoted by q, the position of the ith robot is denoted by $p_i \in Q$. Let $\{V_1, V_2, \ldots, V_n\}$ be the Voronoi partition of Q, for which the robots' positions are the generator points. Specifically,

$$V_i = \left\{ q \in Q \,|\, \|q - p_i\| \leq \left\|q - p_j\right\|, \forall j \neq i \right\} \tag{1}$$

Let the unreliability of the sensor measurement be denoted by a quadratic function $f(x)$; specifically, $f(\|q - p_i\|) = 1/2\,(\|q - p_i\|)^2$ describes how unreliable is the measurement of the information at q by a sensor at p_i. This form of $f(\|q - p_i\|)$ is physically appealing since it is reasonable that sensing will become more unreliable farther from the sensor (Schwager et al. 2009).

Define the sensory function to be a continuous function $Q \to R_{>0}$ (where $R_{>0}$ is the set of strictly positive real numbers). The sensory function should be thought of as a weighting

of importance over Q. We want to have many robots where $\phi(q)$ is large, and few where it is small. The function $\phi(q)$, is not known by the robots in the network, but the robots have sensors that can measure $\phi(p_i)$, at the robots position p_i.

As a measure of the system performance, we define the coverage functional as follows

$$\mathcal{H}(p_1, \ldots, p_N) = \sum_{i=1}^{N} \int_{V_i} \frac{1}{2} (\|q - p_i\|)^2 \phi(q) dq \tag{2}$$

Note that unreliable sensing is expensive and high values of the sensory function $\phi(q)$ are also expensive. Qualitatively, a low value of $\mathcal{H}$ corresponds to a good configuration for sensory coverage of the environment Q.

Then one can define three properties analogous to mass moments of rigid bodies. The mass, first moment, and centroid of a Voronoi region V_i are defined as

$$M_{V_i} = \int_{V_i} \phi(q)\, dq, \tag{3}$$

$$L_{V_i} = \int_{V_i} q\phi(q) dq \tag{4}$$

$$C_{V_i} = \frac{L_{V_i}}{M_{V_i}} \tag{5}$$

respectively. Thus, M_{V_i} and C_{V_i} have properties intrinsic to physical masses and centroids. The moments and the Voronoi regions themselves depend on the robot positions.

Remarkably, despite this dependency, a standard result from locational optimization (Drezner 1995) is that

$$\frac{\partial \mathcal{H}}{\partial p_i} = -\int_{V_i} (q - p_i)\, \phi(q)\, dq = -M_{V_i} \left(C_{V_i} - p_i\right) \tag{6}$$

Equation (6) implies that critical points of $\mathcal{H}$ correspond to the configurations such that $p_i = C_{V_i}$ for all i, that is, each agent is located at the centroid of its Voronoi region. This brings us to the concept of optimal coverage as follows: a robot network is said to be in a (locally) optimal coverage configuration if every robot is positioned at the centroid of its Voronoi region, $p_i = C_{V_i}$ for all i (Schwager et al. 2009).

2.2 Feedforward neural networks

Feedforward neural networks are the networks in which the information flows in a unidirectional way, and are classically organized by layers. Feedforward networks have at least one input layer, composed of neurons that receive the network input signals, and one output layer, composed of one or more neurons that send the network response to the outside. There are one or more intermediate (hidden) layers between the input layer and the output layer, which are so called because the neurons are not directly accessible. The hidden neurons extract important features contained in the input data and provide the network with its ability to generalize. In theory, a neural network with one hidden layer with sufficient hidden neurons is capable of approximating any continuous function. In practice, neural networks with one and occasionally two hidden layers are widely used and have to perform very well (Hertz et al. 1995).

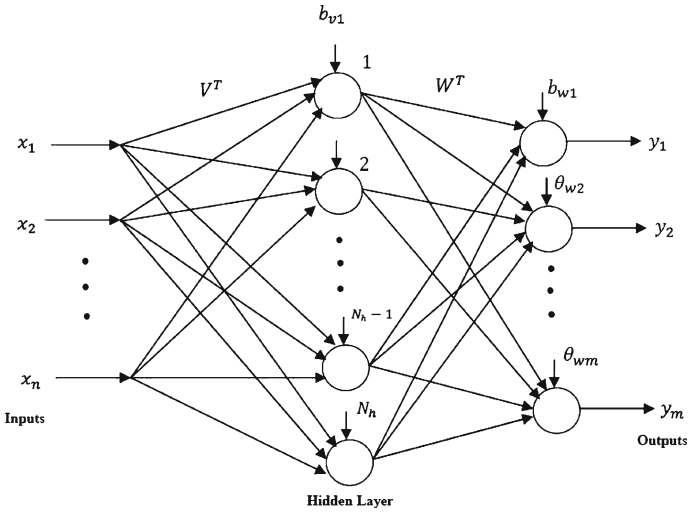

Fig. 1 Structure of a two-layer feedforward neural network

A two-layer feedforward NN shown in Fig. 1 has two layers of adjustable weights. The NN output $\mathbf{y}$ is a vector with components y_i which are determined in terms of the n components of the input vector x by the formula

$$y_i = \sum_{j=1}^{N_h} \left[w_{ij} \sigma \left(\sum_{k=1}^{n} v_{jk} x_k + b_{vj} \right) + b_{wi} \right], \quad i = 1, \ldots, N \tag{7}$$

where the activation functions are $\sigma(.)$ and N_h is the number of hidden-layer neurons. The inputs-to-hidden-layer interconnection weights are denoted by $v_{jk}\psi$ and the hidden-layer to outputs interconnection weights by w_{ij}. The threshold offsets are denoted by b_{wi} and b_{vj}. The activation functions used for neural networks are of a great variety, and have the biggest impact on the behavior and performance of the network. A number of activation functions $\sigma(.)$ have been used in literature, including sigmoid, hyperbolic tangent, and Gaussian functions. Here we will use the most common one, sigmoid activation function

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{8}$$

The most important advantages of sigmoid function are its differentiability within the context of finding a steepest decent gradient for the backpropagation algorithm and also mapping a wide domain of values into the interval [0, 1]. By collecting all the NN weightsv_{jk}, $w_{ij}\psi$ into vectors of weights V^T and $W^T\psi$ one can write the NN equation is terms of vectors as

$$\mathbf{y} = W^T \sigma(V^T \mathbf{x}) \tag{9}$$

with the vector of activation functions defined by $\sigma(z) = [\sigma(z_1), \ldots, \sigma(z_n)]^T$ for a vector $z \in \mathbb{R}^n$. The thresholds are included as the first columns of the weight matrices. To accommodate this, the vectors x and $\sigma(.)$ need to be augmented by placing a 1 as their first element (i.e., $\mathbf{x} \equiv [1, x_1, \ldots, x_n]^T$). Any tuning of V and W then includes tuning of the thresholds as well. The main property of a NN we shall be concerned with for controls purposes is the function approximation property (Cybenko 1989; Hornik et al. 1989). Let $f(x)$ be a smooth

function from $\mathbb{R}^n$ to $\mathbb{R}^m$. Then, it can be shown that, as long as x is restricted to a compact set U_x of $\mathbb{R}^n$, for some number of hidden layer neurons N_h, there exist weights and thresholds such that one has

$$f(\mathbf{x}) = W^T \sigma(V^T \mathbf{x}) + \varepsilon \tag{10}$$

This equation means that an NN can approximate any function in a compact set. The value of $\varepsilon\psi$ is called the NN functional approximation error. In fact, for any choice of a positive number ε_n, one can find a NN such that $\varepsilon < \varepsilon_n$ in U_x. For controls purposes, all one needs to know is that, for a specified value of $\varepsilon_n\psi$ these ideal approximating NN weights exist. Then, an estimate of $f(x)$can be given by

$$\hat{f}(\mathbf{x}) = \hat{W}^T \sigma(\hat{V}^T \mathbf{x}) \tag{11}$$

where $\hat{W}^T$ and $\hat{V}^T$ are estimates of the ideal NN weights that the scalar sigmoid activation function (8), the hidden-layer output gradient is are provided by some on-line weight tuning algorithms. For the scalar sigmoid activation function (8), the hidden-layer output gradient is

$$\frac{\partial \sigma}{\partial z} = \sigma(z)[1 - \sigma(z)] \tag{12}$$

The hidden-layer output gradient or Jacobian may be explicitly computed; for the sigmoid activation functions, it is

$$\hat{\sigma}\prime \equiv diag\{\sigma(\hat{V}^T x)\}\left[I - diag\{\sigma(\hat{V}^T x)\}\right] \tag{13}$$

where I denotes the identity matrix, and $diag(z)$ means a diagonal matrix whose diagonal elements are the components of vector z.

3 Distributed coverage control algorithm

3.1 Distributed coverage controller

Let the dynamics of each robot assumed to be modeled by the first-order equation

$$\dot{p}_i = u_i \tag{14}$$

where u_i is the control input. We can assume there is a low-level controller in place to cancel existing dynamics and enforce (16). We assume that the robots also are able to compute their own Voronoi cell, V_i as well as Cortes et al. (2004), Salapaka et al. (2003), Pimenta et al. (2008). Also, we assume that the sensory function $\phi(q)$ is static for the purposes of our analysis, that is, it does not change with time. The control law can be written as follows (Cortes et al. 2004):

$$u_i = k_i(C_{V_i} - p_i) \tag{15}$$

This control law guarantees that the system converges to an optimal coverage configuration only if the sensory function $\phi(q)$ is known. In this study we consider that the sensory function $\phi(q)$ is not known for the robots, but each robot can measure $\phi(p_i(t))$ at its position p_i.

We want a controller to position the robots at their Voronoi centroids, because we want to drive them to a near-optimal configuration. It is not easy to position a robot at its Voronoi centroid because firstly the robot does not know the sensory function $\phi(q)$ of the environment

which is required to calculate its centroid, and secondly the centroid moves as a non-linear function of the robots position (Schwager et al. 2007). To cope with the first problem, our controller learns an approximation of the centroid using an online weight tuning algorithm in a feedforward two layer NN. To deal with the second problem, the controller causes each robot to track its estimated centroid. We will prove that the robots converge to a near-optimal configuration, and that every robot learns an approximation of the sensory function.

Consider the case that each robot approximates the sensory function $\phi(q)$ using a two-layer feed-forward NN. We denote the approximation of the sensory function at point q for robot i and at time instant t by

$$\hat{\phi}_i(q,t) = \hat{W}_i \sigma(\hat{V}_i^T q) \tag{16}$$

where $\hat{W}_i$ and $\hat{V}_i$ are tuned according to the NN weight-tuning laws that will be described in the next subsection.

Then the moment approximations are as follows

$$\hat{M}_{V_i} = \int_{V_i} \hat{\phi}_i(q)\, dq, \tag{17}$$

$$\hat{L}_{V_i} = \int_{V_i} \tilde{p}\hat{\phi}_i(q)\, dq, \tag{18}$$

$$\hat{C}_{V_i} = \frac{\hat{L}_{V_i}}{\hat{M}_{V_i}} \tag{19}$$

which are analogous to (3), (4), and (5), respectively. The mass moment errors are

$$\tilde{M}_{V_i} = \int_{V_i} \tilde{\phi}_i(q)\, dq = \hat{M}_{V_i} - M_{V_i}, \tag{20}$$

$$\tilde{L}_{V_i} = \int_{V_i} q\tilde{\phi}_i(q)\, dq = \hat{L}_{V_i} - L_{V_i} \tag{21}$$

$$C_{V_i} = \frac{\tilde{L}_{V_i}}{\tilde{M}_{V_i}} \tag{22}$$

From (20), (21), and (22) one can find that

$$L_{V_i} = M_{V_i}\hat{C}_{V_i} + \tilde{M}_{V_i}\ \ (\hat{C}_{V_i} - \tilde{C}_{V_i}). \tag{23}$$

Now the sensory function estimation error of robot i at point q is defined as

$$\tilde{\phi}_i(q,t) = \phi(q) - \hat{\phi}_i(q,t) \tag{24}$$

For convenient, in the rest of the paper we use shorthand $\phi_i = \phi(p_i(t))$ and $\tilde{\phi}_i = \tilde{\phi}_i(p_i(t))$ for the value of the sensory function and the sensory function approximation error at the position of robot i, respectively, and $\tilde{\phi}_i^q = \tilde{\phi}_i(q,t)$ for the value of the sensory function approximation error at point q.

Next, define two error vectors, the actual error,

$$e_i = C_{V_i} - p_i \tag{25}$$

and the estimated error,

$$\hat{e}_i = \hat{C}_{V_i} - p_i. \tag{26}$$

We now propose to use the control law

$$u_i = k_{1_i}(\hat{C}_{V_i} - p_i) \tag{27}$$

where $k_{1_i} > 0$ is a scalar gain matrix.

3.2 NN control design for sensory function estimation

Here, we are going to present the online estimation procedure using NN. At first, we define the weight estimation errors for the robot i as $\tilde{V}_i = V_i - \hat{V}_i$, $\tilde{W}_i = W_i - \hat{W}_i$. Next define the hidden-layer output error of the ith robot's approximation for a given q as

$$\tilde{\sigma}_i = \sigma_i - \hat{\sigma}_i = \sigma(V_i^T q) - \sigma(\hat{V}_i^T q) \tag{28}$$

For our analysis, we require that the following assumption holds.

Assumption 1 On any compact subset of $\mathbb{R}^n$, the ideal NN weights are bounded by known positive values so that $||V||_F \leq V_M$ and $||W||_F \leq W_M$, where $||A||_F$ denotes the Frobenius norm of the matrix A as follows:

$$||A||_F^2 = tr\{A^T A\} = \sum_{i,j} a_{ij}^2 \tag{29}$$

Assumption 2 The NN weights of the hidden layer satisfy the following condition

$$W_i \geq \mathbf{1} w_{\mathrm{m}} \tag{30}$$

where $w_{\mathrm{m}} \in \mathbb{R}^{>0}$ is a lower bound known by each robot, $\mathbf{1}$ is the vector of ones and $\geq$ is applied element-wise. Note that Assumption 1 is always satisfied in practical situations (Hornik et al. 1989). Requirements such as Assumption 2 are also common for adaptive controllers. In theory, this assumption is not limiting since any function (with some smoothness requirements) over a bounded domain can be approximated arbitrarily well by some set of basis functions (Sanner et al. 1992).

The NN weights $\hat{W}_i$ and $\hat{V}_i$ used to calculate $\hat{C}_{V_i}$ are adjusted according to a set of weight-tuning laws which are introduced as follows

$$\dot{\hat{W}}_{ip} = \begin{cases} F\hat{\sigma}_i\tilde{\phi}_i - k_{2_i}\frac{W_M}{\tilde{\phi}_i}\hat{\sigma}_i^+ F^{-1T}\int\limits_{V_i}\left|\hat{e}_i^T(q - \hat{C}_{V_i})\right| dq - k_{3_i}F|\tilde{\phi}_i|\hat{W}_i, & \text{if } \tilde{\phi}_i \neq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{31}$$

$$\dot{\hat{V}}_i = Gq(\hat{\sigma}_i^T\hat{W}_i\tilde{\phi}_i)^T - \tilde{\phi}_i q\hat{W}_i^T G\hat{\sigma}_i' - k_{3_i}G|\tilde{\phi}_i|\hat{V}_i. \tag{32}$$

where $F \in \mathrm{R}^{N_h \times N_h}$ and $G \in \mathrm{R}^{2N_h \times 2N_h}$ are positive definite gain matrices with appropriate dimensions, k_{2_i} and k_{3_i}, and a are positive scalar gains, and $\hat{\sigma}_i^+$ denotes the pseudo inverse of $\hat{\sigma}_i$, i.e.$\hat{\sigma}_i\hat{\sigma}_i^+ = I$.

Equation (31) is the main weight-tuning algorithm for $\hat{W}_i$, however the controller (27) has a singularity at $\hat{W}_i = 0$. Note that because the output of the function $\sigma(.)$ is always above zero, there is no such a singularity in $\dot{\hat{V}}_i = 0$. Hence we need just to deal with the case that $\dot{\hat{W}}_i = 0$. We prevent the parameters from dropping below $w_{\min}$ and rising above W_M using a parameter projection (Slotine and Coetsee 1986) algorithm:

$$\dot{\hat{W}}_i = \dot{\hat{W}}_{ip} - I_{proj}\dot{\hat{W}}_{ip} \tag{33}$$

where the diagonal matrix I_{proj} is defined element-wise as

$$I_{proj}(j) = \begin{cases} 0 & \text{if } w_{\min} < \hat{W}_i(j) < w_{\max}, \\ 0 & \text{if } \hat{W}_i(j) = w_m \text{ and } \dot{\hat{W}}_{ip}(j) \geq 0 \\ 0 & \text{if } \hat{W}_i(j) = W_M \text{ and } \dot{\hat{W}}_{ip}(j) \leq 0 \\ 1 & \text{otherwise.} \end{cases}, \tag{34}$$

and $I_{proj}(j) = 0, \forall i \neq j$, where $\hat{W}_i(j)$ and $I_{proj}(j)$ denotes the jth element of the vector $\hat{W}_i$ and the jth diagonal element of the matrix I_{proj}, respectively.

Theorem 1 (convergence theorem) *under assumptions 1 and 2, a network of robots with dynamics (14), control law (27), and weight-tuning laws (31), (32) and (33), converges to a near-optimal coverage configuration. Furthermore, sensory function approximation error for each robot converges to zero over time.*

Proof We propose the following Lyapunov candidate function

$$V = \mathcal{H} + V_1 \tag{35}$$

where $\mathcal{H}$ is defined in (2) and V_1 is as follows:

$$V_1 = \frac{1}{2} \sum_{i=1}^{n} \left[a\tilde{\phi}_i^2 + tr\left\{ \tilde{W}_i^T F^{-1} \tilde{W}_i + \tilde{V}_i^T G^{-1} \tilde{V}_i \right\} \right] \tag{36}$$

where $a > 0$ is a scalar term. At first, we take the time derivative of $\mathcal{H}$ along the trajectories of the system which gives

$$\dot{\mathcal{H}} = \sum_{i=1}^{n} \frac{\partial \mathcal{H}}{\partial p_i}^T \dot{p}_i \tag{37}$$

Substituting from (6) yields

$$\dot{\mathcal{H}} = \sum_{i=1}^{n} M_{V_i} (p_i - C_{V_i})^T \dot{p}_i \tag{38}$$

Now we use the property in (24) to obtain

$$\dot{\mathcal{H}} = \sum_{i=1}^{n} \left[M_{V_i} (p_i - \hat{C}_{V_i})^T \dot{p}_i + \tilde{M}_{V_i} (\tilde{C}_{V_i} - \hat{C}_{V_i})^T \dot{p}_i \right] \tag{39}$$

Substituting for $\dot{p}_i$ with (14) and (27) and simplifying with $\tilde{L}_{V_i} = \tilde{M}_{V_i} \tilde{C}_{V_i}$, we have

$$\dot{\mathcal{H}} = \sum_{i=1}^{n} \left[-M_{V_i} (\hat{C}_{V_i} - p_i)^T k_{1_i} (\hat{C}_{V_i} - p_i) + (\tilde{L}_{V_i} - \tilde{M}_{V_i} \hat{C}_{V_i})^T k_{1_i} (\hat{C}_{V_i} - p_i) \right] \tag{40}$$

Simplifying further with $\hat{e}_i = \hat{C}_{V_i} - p_i$ and expanding $\tilde{L}_{V_i}$ and $\tilde{M}_{V_i}$ with (21) and (22) leads to

$$\dot{\mathcal{H}} = \sum_{i=1}^{n} \left[-M_{V_i} \hat{e}_i^T k_{1_i} \hat{e}_i + k_{1_i} \int_{V_i} \tilde{\phi}_i^q (q - \hat{C}_{V_i})^T dq \hat{e}_i \right]. \tag{41}$$

Note that the first term in (41) is obviously non-positive. Now taking the time derivative of V_1 yields:

$$\begin{aligned}\dot{V}_1 &= \sum_{i=1}^{n}\left[-a\tilde{\phi}_i\dot{\hat{\phi}}_i - tr\left\{\tilde{V}_i^T G^{-1}\dot{\hat{V}}_i + \tilde{W}_i^T F^{-1}\dot{\hat{W}}_i\right\}\right] \\ &= \sum_{i=1}^{n}\left[-a\tilde{\phi}_i(\dot{\hat{W}}_i^T\hat{\sigma}_i + q^T\dot{\hat{V}}_i\hat{\sigma}_i'^T\hat{W}_i) \; -tr\left\{\tilde{V}_i^T G^{-1}\dot{\hat{V}}_i + \tilde{W}_i^T F^{-1}\dot{\hat{W}}\right\}\right] \end{aligned} \tag{42}$$

Substituting $\dot{\hat{W}}_i^T$ from (33) yields:

$$\begin{aligned}\dot{V}_1 &= \sum_{i=1}^{n}\Big[-a\tilde{\phi}_i(\dot{\hat{W}}_{ip} - I_{proj}\dot{\hat{W}}_{ip})^T\hat{\sigma}_i - a\tilde{\phi}_i q^T\dot{\hat{V}}_i\hat{\sigma}_i'^T\hat{W}_i \\ &\quad - tr\left\{\tilde{W}_i^T(\dot{\hat{W}}_{ip} - I_{proj}\dot{\hat{W}}_{ip}) + \tilde{V}_i^T G^{-1}\dot{\hat{V}}_i\right\}\Big] \\ &= \sum_{i=1}^{n}\Big[-a\tilde{\phi}_i\dot{\hat{W}}_{ip}^T\hat{\sigma}_i + a\tilde{\phi}_i\dot{\hat{W}}_{ip}^T I_{proj}\hat{\sigma}_i - a\tilde{\phi}_i q^T\dot{\hat{V}}_i\hat{\sigma}_i'^T\hat{W}_i \\ &\quad - tr\left\{\tilde{W}_i^T(\dot{\hat{W}}_{ip} - I_{proj}\dot{\hat{W}}_{ip})\right\} \; -tr\left\{\tilde{V}_i^T G^{-1}\dot{\hat{V}}_i\right\}\Big] \end{aligned} \tag{43}$$

Then substituting for $\dot{\hat{W}}_{ip}$ and $\dot{\hat{V}}_i$ with (31) and (32) we have

$$\begin{aligned}\dot{V}_1 &= \sum_{i=1}^{n}\Bigg[-a\tilde{\phi}_i^2\hat{\sigma}_i^T F\hat{\sigma}_i + ak_{3_i}\tilde{\phi}_i\left|\tilde{\phi}_i\right|\hat{W}_i^T F\hat{\sigma}_i \\ &\quad - a\tilde{\phi}_i\left[\frac{k_{2_i}W_M}{\tilde{\phi}_i}\int_{V_i}\left|(q-\hat{C}_{V_i})^T\hat{e}_i\right|dq\hat{\sigma}_i^+F^{-1}\right]F\hat{\sigma}_i + a\tilde{\phi}_i\hat{W}_{ip}I_{proj}\hat{\sigma}_i \\ &\quad - tr\left\{a\tilde{\phi}_i\hat{\sigma}_i'^T\hat{W}_i q^T\hat{V}_1\right\} \; -tr\left\{(\tilde{W}_i^T\dot{\hat{W}}_{ip} - \tilde{W}_i^T I_{proj}\dot{\hat{W}}_{ip}) + \tilde{V}_i^T G^{-1}\dot{\hat{V}}_i\right\}\Bigg] \\ &= \sum_{i=1}^{n} -a\tilde{\phi}_i^2\hat{\sigma}_i^T F\hat{\sigma}_i + ak_{3_i}\tilde{\phi}_i\left|\tilde{\phi}_i\right|\hat{W}_i^T F\hat{\sigma}_i + a\tilde{\phi}_i\hat{W}_{ip}I_{proj}\hat{\sigma}_i \\ &\quad - ak_{2_i}W_M\int_{V_i}\left|(q-\hat{C}_{V_i})^T\hat{e}_i\right|dq\hat{\sigma}_i^+\hat{\sigma}_i + \tilde{W}_i^T I_{proj}\dot{\hat{W}}_{ip} \\ &\quad - tr\left\{a\tilde{\phi}_i^2\hat{\sigma}_i^T\hat{W}_i q^T G q\hat{W}_i^T\hat{\sigma}_i' - ak\tilde{\phi}_i\left|\tilde{\phi}_i\right|\hat{\sigma}_i'^T\hat{W}_i q^T G_i\hat{V}_i\right. \\ &\quad + a\tilde{\phi}_i^2\hat{\sigma}_i'^T\hat{W}_i q^T q\hat{W}_i^T\hat{\sigma}_i' + \tilde{\phi}_i\tilde{W}_i^T\hat{\sigma}_i - \tilde{\phi}_i\tilde{W}_i^T\hat{\sigma}_i - k_{3_i}\left|\tilde{\phi}_i\right|\tilde{W}_i^T\hat{W}_i \\ &\quad \left. + \tilde{V}_i^T q\tilde{\phi}_i\hat{W}_i^T\hat{\sigma}_i' - \tilde{V}_i^T q\tilde{\phi}_i\hat{W}_i^T\hat{\sigma}_i' - k_{3_i}\left|\tilde{\phi}_i\right|\tilde{V}_i^T\hat{V}_i\right\} \end{aligned} \tag{44}$$

Simplifying the above equation yields

$$\begin{aligned}\dot{V}_1 = \sum_{i=1}^{n} \Bigg[& -a\tilde{\phi}_i^2 \hat{\sigma}_i^T F \hat{\sigma}_i + ak_{3_i}\tilde{\phi}_i \left|\tilde{\phi}_i\right| \hat{W}_i^T F \hat{\sigma}_i \\ & - ak_{2_i} W_M \int_{V_i} \left|(q - \hat{C}_{V_i})^T \hat{e}_i\right| dq \\ & + a\tilde{\phi}_i \hat{W}_{ip} I_{proj} \hat{\sigma}_i + \tilde{W}_i^T I_{proj} \dot{\hat{W}}_{ip} \\ & - tr\left\{a\tilde{\phi}_i^2 \hat{\sigma}_i^T \hat{W}_i q^T G q \hat{W}_i^T \hat{\sigma}_i' - ak_{3_i}\tilde{\phi}_i \left|\tilde{\phi}_i\right| \hat{\sigma}_i'^T \hat{W}_i q^T G_i \hat{V}_i \right. \\ & \left. + a\tilde{\phi}_i^2 \hat{\sigma}_i'^T \hat{W}_i q^T q \hat{W}_i^T \hat{\sigma}_i'\right\} \\ & - k_{3_i}\left|\tilde{\phi}_i\right| tr\left\{\tilde{W}_i^T (W - \tilde{W}_i) + \tilde{V}_i^T (V - \tilde{V}_i)\right\} \Bigg] \end{aligned} \tag{45}$$

For notational convenience we define the matrix of all the NN weights $Z_i = diag\{W_i, V_i\}$ of the ith robot and the estimated value by each robot as $\hat{Z}_i = diag\{\hat{W}_i, \hat{V}_i\}$. Noticing assumption 1, one can easily conclude that $||Z_i||_F < Z_m$ where Z_m is a known positive constant. According to (41) and (45) we have

$$\begin{aligned}\dot{V} = \sum_{i=1}^{n} & -M_{V_i}\hat{e}_i^T k_{1_i}\hat{e}_i + k_{1_i}\int_{V_i}\tilde{\phi}_i^q (q - \hat{C}_{V_i})^T dq \hat{e}_i - a\tilde{\phi}_i^2 \hat{\sigma}_i^T F \hat{\sigma}_i + ak_{3_i}\tilde{\phi}_i \left|\tilde{\phi}_i\right| \hat{W}_i^T F \hat{\sigma}_i \\ & - ak_{2_i} W_M \int_{V_i} \left|(q - \hat{C}_{V_i})^T \hat{e}_i\right| dq + a\tilde{\phi}_i \hat{W}_{ip} I_{proj} \hat{\sigma}_i + \tilde{W}_i^T I_{proj} \dot{\hat{W}}_{ip} \\ & - tr\left\{a\tilde{\phi}_i^2 \hat{\sigma}_i^T \hat{W}_i q^T G q \hat{W}_i^T \hat{\sigma}_i' - ak_{3_i}\tilde{\phi}_i \left|\tilde{\phi}_i\right| \hat{\sigma}_i'^T \hat{W}_i q^T G_i \hat{V}_i + a\tilde{\phi}_i^2 \hat{\sigma}_i'^T \hat{W}_i q^T q \hat{W}_i^T \hat{\sigma}_i'\right\} \\ & + k_{3_i}\left|\tilde{\phi}_i\right| tr\left\{\tilde{Z}_i^T (Z - \tilde{Z}_i)\right\} \end{aligned} \tag{46}$$

Since

$$tr\left\{\tilde{Z}_i^T (Z - \tilde{Z}_i)\right\} = \left\langle \tilde{Z}_i, Z\right\rangle_F - \left\|\tilde{Z}_i\right\|_F^2 \leq \left\|\tilde{Z}_i\right\|_F \|Z\|_F - \left\|\tilde{Z}_i\right\|_F^2, \tag{47}$$

by selecting $k_{2_i} = \frac{k_{1_i}}{a}$ we obtain

$$\begin{aligned}\dot{V} \leq \sum_{i=1}^{n} & -M_{V_i}\hat{e}_i^T k_{1_i}\hat{e}_i + k_{1_i}\int_{V_i}\tilde{\phi}_i^q (q - \hat{C}_{V_i})^T dq \hat{e}_i - k_{1_i} W_M \int_{V_i}\left|(q - \hat{C}_{V_i})^T \hat{e}_i\right| dq \\ & + a\tilde{\phi}_i \left(\left|\tilde{\phi}_i\right| k_{3_i} \hat{W}_i^T F \hat{\sigma}_i - \tilde{\phi}_i \hat{\sigma}_i^T F \hat{\sigma}_i\right) + \tilde{W}_i^T I_{proj} \dot{\hat{W}}_{ip} + a\tilde{\phi}_i \hat{W}_{ip} I_{proj} \hat{\sigma}_i \\ & + k_{3_i}\left|\tilde{\phi}_i\right| \left\|\tilde{Z}_i\right\|_F \left(\left\|\tilde{Z}_i\right\|_F - Z_M\right) \end{aligned} \tag{48}$$

The term $-M_{V_i}\hat{e}_i^T k_{1_i}\hat{e}_i$ is always non-positive, so (48) reduces to:

$$\begin{aligned}\dot{V} \leq & \sum_{i=1}^{n} k_{1_i} \int_{V_i} \tilde{\phi}_i^q (q-\hat{C}_{V_i})^T \hat{e}_i dq - k_{1_i} W_M \int_{V_i} \left|(q-\hat{C}_{V_i})^T \hat{e}_i\right| dq \\ & + a\tilde{\phi}_i \left(\left|\tilde{\phi}_i\right| k_{3_i} \hat{W}_i^T F\hat{\sigma}_i - \tilde{\phi}_i \hat{\sigma}_i^T F\hat{\sigma}_i\right) + \tilde{W}_i^T I_{proj} \dot{\hat{W}}_{ip} + a\tilde{\phi}_i \hat{W}_{ip} I_{proj} \hat{\sigma}_i \\ & - tr\left\{a\tilde{\phi}_i^2 \hat{\sigma}_i^T \hat{W}_i q^T G_i q \hat{W}_i^T \hat{\sigma}_i'\right\} - a\tilde{\phi}_i tr\left\{\hat{\sigma}_i'^T \hat{W}_i q^T \left(\tilde{\phi}_i q \hat{W}_i^T \hat{\sigma}_i' - k_{3_i}\left|\tilde{\phi}_i\right| G_i \hat{V}_i\right)\right\} \\ & + k_{3_i}\left|\tilde{\phi}_i\right| \left\|\tilde{Z}_i\right\|_F \left(\left\|\tilde{Z}_i\right\|_F - Z_M\right) \end{aligned} \tag{49}$$

First consider the case that $\tilde{\phi}_i \neq 0$. Then, for the first and second term, according to (16) and assumption 1, we can say that $\hat{\phi}_i(q,t) \leq W_M$, so $\tilde{\phi}_i^q \leq W_M$. In addition,$(q-\hat{C}_{V_i})^T\hat{e}_i \leq \left|(q-\hat{C}_{V_i})^T\hat{e}_i\right|$ always holds. Therefore, one can conclude that $k_{1_i}\int_{V_i}\tilde{\phi}_i^q(q-\hat{C}_{V_i})^T dq\hat{e}_i - k_{1_i}W_M\int_{V_i}\left|(q-\hat{C}_{V_i})^T\hat{e}_i\right| dq \leq 0$. So (49) can be rewritten as

$$\begin{aligned}\dot{V} \leq & \sum_{i=1}^{n} a\tilde{\phi}_i \left(\left|\tilde{\phi}_i\right| k_{3_i} \hat{W}_i^T F\hat{\sigma}_i - \tilde{\phi}_i \hat{\sigma}_i^T F\hat{\sigma}_i\right) + \tilde{W}_i^T I_{proj} \dot{\hat{W}}_{ip} + a\tilde{\phi}_i \hat{W}_{ip} I_{proj} \hat{\sigma}_i \\ & - tr\left\{a\tilde{\phi}_i^2 \hat{\sigma}_i^T \hat{W}_i q^T G_i q \hat{W}_i^T \hat{\sigma}_i'\right\} - a\tilde{\phi}_i tr\left\{\hat{\sigma}_i'^T \hat{W}_i q^T \left(\tilde{\phi}_i q \hat{W}_i^T \hat{\sigma}_i' - k_{3_i}\left|\tilde{\phi}_i\right| G_i \hat{V}_i\right)\right\} \\ & + k_{3_i}\left|\tilde{\phi}_i\right| \left\|\tilde{Z}_i\right\|_F \left(\left\|\tilde{Z}_i\right\|_F - Z_M\right) \end{aligned} \tag{50}$$

The forth term in (50) is obviously non-positive and the last term maintains negative as long as $||\tilde{Z}_i||_F < Z_m$. Expanding the second term as a sum of scalar terms, one can see that the j^{th} scalar term is of the form

$$\tilde{W}_i(j) I_{proj}(j) \dot{\hat{W}}_{ip}(j). \tag{51}$$

From (34) if $\hat{W}_i(j) > w_m$ or $\hat{W}_i(j) = w_m$ and $\dot{\hat{W}}_i(j) \geq 0$ then $I_{proj}(j) = 0$ and the term vanishes. Now in the case $\hat{W}_i(j) = w_m$ and $\dot{\hat{W}}_{ip}(j) \leq 0$ we have $\tilde{W}_i(j) = W(j) - \dot{\hat{W}}_i(j) \geq 0$. Moreover, if $I_{proj}(j) = 1$ and $\dot{\hat{W}}_i(j) < 0$, the term is negative; therefore, in all cases the term $\tilde{W}_i I_{proj} \dot{\hat{W}}_{ip}$ is negative. The third term can be shown to be non-negative as well as the second term according to the above discussion.

Now we continue by investigating the contribution of the first term, i.e. $a\tilde{\phi}_i(\left|\tilde{\phi}_i\right| k_{3_i}\hat{W}_i^T F\hat{\sigma}_i - \tilde{\phi}_i\hat{\sigma}_i^T F\hat{\sigma}_i)$. Similar to the previous case if $\tilde{\phi}_i < 0$, the term remains non-positive. For the case that $\tilde{\phi}_i > 0$ the term simplifies to $-a\tilde{\phi}_i^2\left(\hat{\sigma}_i^T F\hat{\sigma}_i - k_{3_i}\hat{W}_i^T F\hat{\sigma}_i\right)$. According to the fact that the neural network weights are assumed to be positive, we can say that $\frac{1}{2} < \hat{\sigma}_i(j) < 1$. Moreover from assumption 1 and (33) and (34), we know that $0 < \hat{W}_i(j) < W_M$. Therefore, as long as k_{3_i} is selected such that

$$k_{3_i} \leq \frac{1}{2W_M}, \tag{52}$$

the term will remain non-positive.

Now consider the fifth term of $\dot{V}$, $-a\tilde{\phi}_i tr\left\{\hat{\sigma}_i'^T \hat{W}_i q^T \left(\tilde{\phi}_i q \hat{W}_i^T \hat{\sigma}_i' - k_{3_i}\left|\tilde{\phi}_i\right| G_i \hat{V}_i\right)\right\}$. Note that as long as $\tilde{\phi}_i < 0$ this term clearly remains non-negative. Then consider the case that

$\tilde{\phi}_i > 0$. In this case the above term simplifies to $-a\tilde{\phi}_i^2 tr\left\{\hat{\sigma}_i'^T \hat{W}_i q^T \left(q\hat{W}_i^T \hat{\sigma}_i' - k_{3_i} G_i \hat{V}_i\right)\right\}$ whose sign depends on $q\hat{W}_i^T \hat{\sigma}_i' - k_{3_i} G_i \hat{V}_i$. From (13) one can write

$$\hat{\sigma}_i'^T = \hat{\sigma}_i \left(\mathbf{1} - \hat{\sigma}_i\right) I_{N_h} \tag{53}$$

where $\mathbf{1}$ is the $N_h \times 1$ vector of 1's. Then the term simplifies as follows:

$$\begin{aligned} q\hat{W}_i^T \hat{\sigma}_i(\mathbf{1} - \hat{\sigma}_i) I_{N_h} - k_{3_i} G_i \hat{V}_i &= q\hat{W}_i^T \hat{\sigma}_i(\mathbf{1} - \hat{\sigma}_i) I_{N_h} - k_{3_i} G_i \hat{V}_i \\ &= q\hat{\phi}_i(\mathbf{1} - \hat{\sigma}_i) I_{N_h} - k_{3_i} G \hat{V}_i \end{aligned} \tag{54}$$

For this term to be non-positive one way is to have

$$q\hat{\phi}_i(\mathbf{1} - \hat{\sigma}_i) I_{N_h} > k_{3_i} G \hat{V}_i \tag{55}$$

where the operator $>$ is applied element wise. Now by choosing

$$k_{3_i} = \frac{w_m}{\|G\|_F V_M} \tag{56}$$

one can guarantee that the elements of (54) all remain non-positive. In order for k_{3_i} to satisfy both (51) and (55), one can choose the gain matrix G such that

$$\|G\|_F > \frac{2w_m W_M}{V_M} \tag{57}$$

Now we investigate the case that, $\tilde{\phi}_i = 0$ in which (49) reduces to

$$\dot{V} \leq \sum_{i=1}^{n} k_{1_i} \int_{V_i} \tilde{\phi}_i^q (q - \hat{C}_{V_i})^T \hat{e}_i dq - k_{1_i} W_M \int_{V_i} \left|(q - \hat{C}_{V_i})^T \hat{e}_i\right| dq + \tilde{W}_i^T I_{proj} \dot{\hat{W}}_{ip} \tag{58}$$

which can be shown to be negative according to the above discussions.

Using the fact that u_i is continuous, one can conclude that the Lyapunov function V has continuous first partial derivative; hence it is a radially unbounded function. Therefore, $\dot{V} \leq 0$ implies that $\dot{V}$ is uniformly continuous, and by Barbalat's lemma $\lim_{t\to\infty} \dot{V} = 0$, which directly proves Theorem 1. □

We should emphasize that our proposed method outperforms previous adaptive control approaches (i.e. Schwager et al. 2007) in two ways. Firstly, the NN estimation can always be accomplished, due to the neural network approximation property (Cybenko 1989). By contrast, in adaptive control approaches it is only possible to proceed if $\phi(q)$ is linear in the known parameters. Furthermore, tedious analysis is needed to compute a regression matrix. Secondly, in our method, robots do not have to know a prior knowledge even about the structure of the nonlinear sensory function of the environment $\phi(q)$. On the other hand, in a linear regression based adaptive controller (Schwager et al. 2007), each robot needs to have the vector of basis functions available for the computation of its control law, since it uses a copy of the vector of basis functions at each time step to compute its estimation of the sensory function. In comparison, here each robot just needs to know the value of the sensory function at its position and there is no need to any extra information about the structure of the function and the value of any basis function.

4 Simulation results

The proposed distributed coverage algorithm has been demonstrated via numerical simulations in Matlab environment. A team of 20 mobile robots is waiting to be deployed into a unit square Environment. The dynamics in (14) with the control law in (27) and the weight-tuning algorithms in (31–33) were modeled as a system of coupled differential equations. The Matlab numerical solver ode45 was used to integrate the equations, and the spatial integrals in (12) required for the computation of the centroids were computed by discretizing each Voronoi region and summing contributions of the integrand over the grid. Voronoi regions were computed using a decentralized algorithm similar to Cortes et al. (2004). The gain parameter for the basic controller (27) was selected as $k_{1_i} = 1$. For the Neural Network, we selected a two layer one with sigmoid activation function and $N_h = 10$ hidden layers and $w_{\mathrm{m}} = 0.1$, $W_M = 2000$, and $V_M = 1000$. The weight tuning parameters were selected as $k_{2_i} = 0.3, k_{3_i} = 0.00005$, $F = 10I_{N_h}$, and $G = 10I_2$. The robots were started from random initial positions and they know nothing about the nonlinear structure of the sensory distribution function. The estimated parameters $\hat{W}_i$ and $\hat{V}_i$ for each robot were started at random value between zero and 10. The spatial integrals in (21), (22), (31) and (32) required for the control law were computed by discretizing each Voronoi region V_i into a 10×10 grid and summing contributions of the integrand over the grid.

The simulations are carried out via three scenarios.

Scenario 1 In order to compare the proposed algorithm with previous adaptive procedures of coverage control, for the first scenario, the following sensory function is considered as in Schwager et al. (2007)

$$\phi(q) = (a/\sqrt{2\pi}\sigma_\phi)\left[\exp(-(q-\mu_1)^2/2\sigma_\phi^2) + \exp(-(q-\mu_2)^2/2\sigma_\phi^2)\right] \tag{59}$$

where $a = 500$, $\sigma_\phi = 0.2$, $\mu_1 = [0.17, 0.17]^T$, and $\mu_2 = [0.83, 0.83]^T$ producing a bimodal Gaussian distribution. Figure 2 shows the results of a simulation run with the $\phi(q)$ given by (61). The initial configuration of the network, the sensory function, the trajectories of the robots (dashed lines), and the final configuration of the robots are shown in Fig. 2a–d, respectively. It can be easily seen that the robots converge to a near-optimal configuration. In Fig. 2e, the norm of the estimated errors averaged over all the robots are shown for the controller in Schwager et al. (2007) and the proposed controller described above. We can see that both controllers make the robots move to the estimated centroid of their Voronoi regions and hence to a near-optimal configuration. But it is easy to verify that our proposed controller has a higher convergence rate than the former in the mean estimated error. Figure 2f shows the mean sensory function approximation of the robots along their trajectory over time. The plot shows the sensory function estimation error averaged over all of the robots in the network converging to zero. Note that the jagged time history is a result of the discretized spatial integral computation over the Voronoi region. Some of the NN weights are depicted in Fig. 3g.

Scenario 2 In the case of the second scenario, the sensory function is considered as

$$\phi(x, y) = a\exp(-b(x-y)^2) \tag{60}$$

with $a = 1000$, $b = -10$. Here, the point is that the sensory function is not a multi modal Gaussian function; therefore it cannot be considered as a linear combination of the Gaussian basis functions. The simulation parameters are the same as the first scenario. The initial

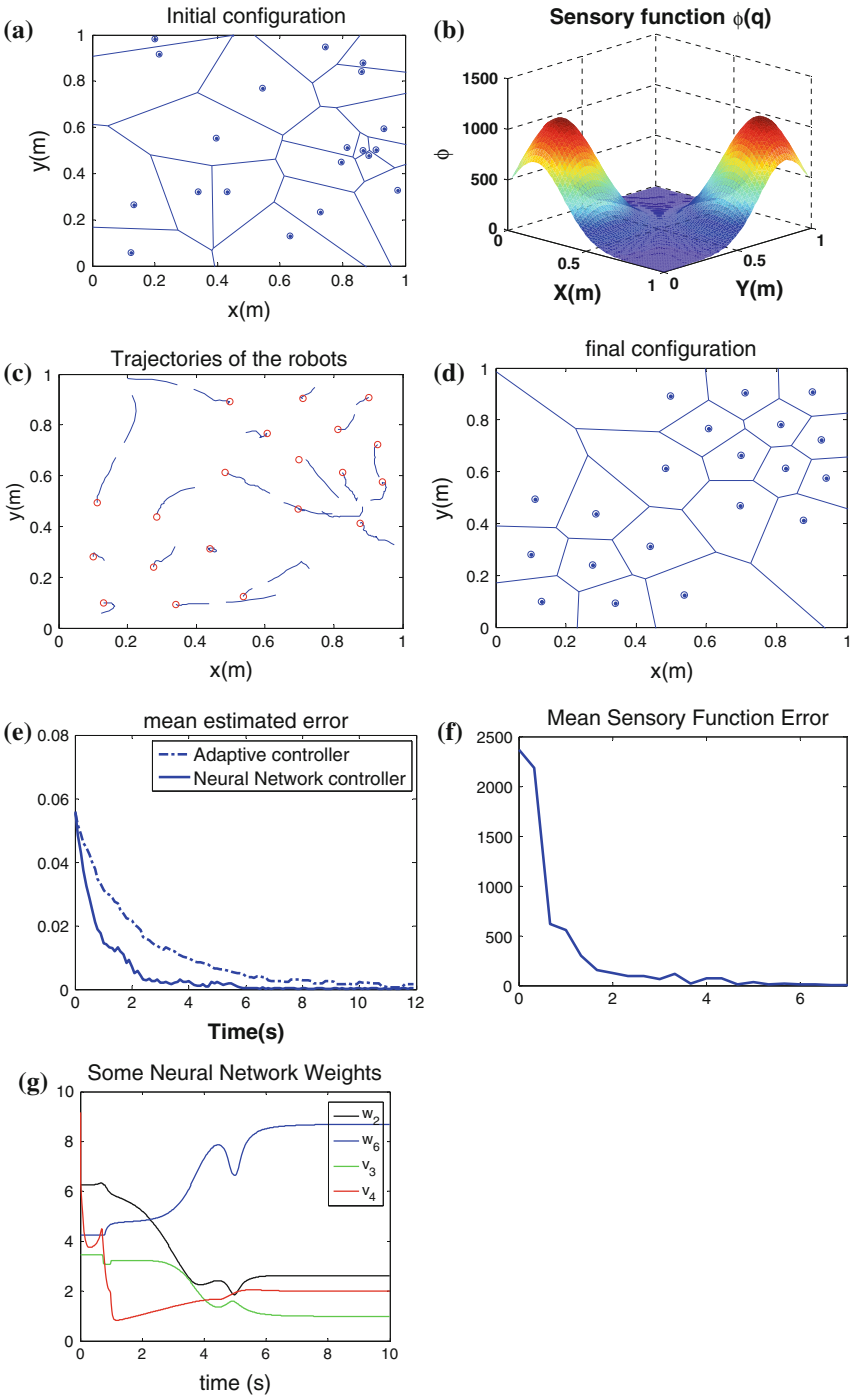

Fig. 2 Scenario 1, **a** the initial configuration of the robot, **b** the sensory function surface, **c** the trajectories of the robot, **d** the final configuration of the robot, **e** the norm of the estimated error averaged over all the robots, **f** sensory function estimation error averaged over all the robots, and **g** some of the neural network weights

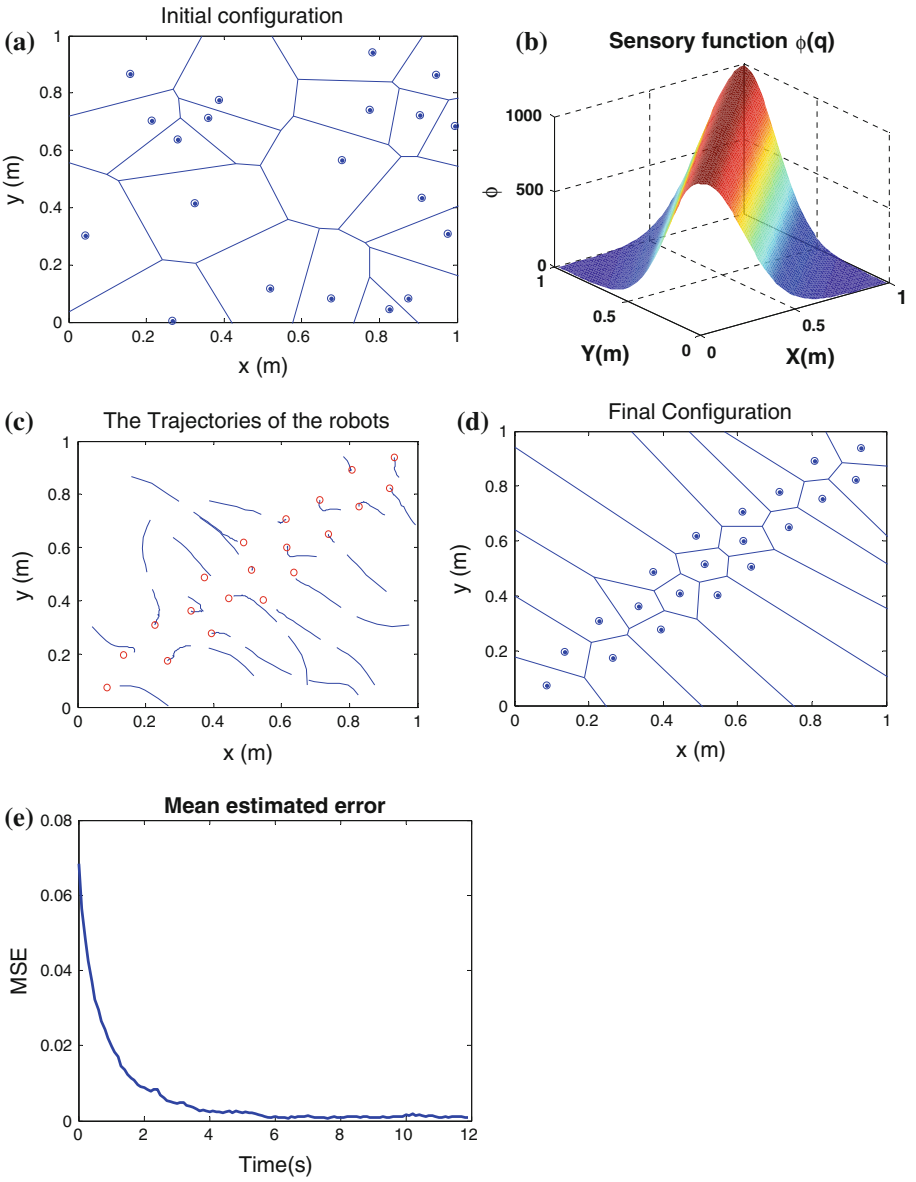

Fig. 3 Scenario 2, **a** the initial configuration of the robots, **b** sensory function, **c** the trajectories, **d** the final configuration of the robots, **e** mean estimated error

configuration, trajectories, and the final configuration of the robots are depicted in Fig. 3a–d, respectively. Due to nonlinear approximation of the NN, the robots converge to a near-optimal configuration, and the mean estimated error converges to zero.

Scenario 3 In the third scenario, we select the following function as the sensory function

$$\phi\left(q_x, q_y\right) = aq_x + bq_y + c \tag{61}$$

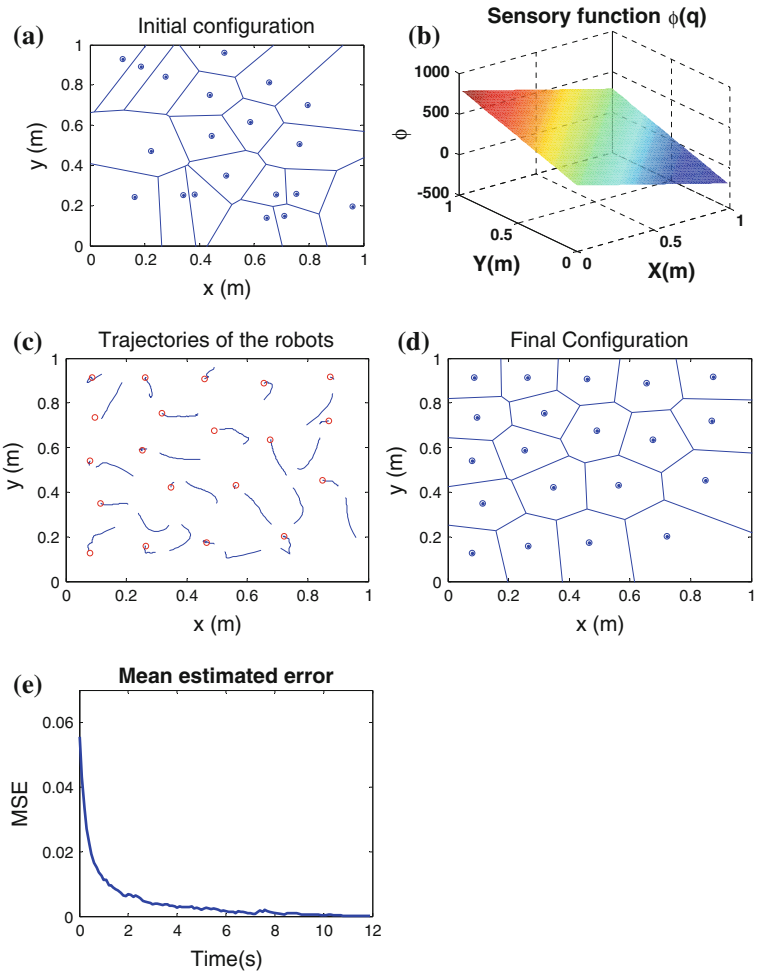

Fig. 4 Scenario 3, **a** the initial configuration of the robots, **b** sensory function, **c** the trajectories, **d** the final configuration of the robots, and **e** the mean estimated error

where $a = -500$, $b = 500$, $c = 1000$, which is the equation of a plane. This function, like the previous one, is not a multi modal Gaussian function. The simulation results for this sensory function are presented in Fig. 4a–e. One can easily see that the adaptive controller performs in such a way that the estimated error of the positions with respect to estimated Voronoi centroids converge to zero and the robots positions converge to the centroidal Voronoi configurations.

5 Conclusion and future work

In this paper, we proposed a distributed control algorithm in order to deploy a network of mobile robots in an unknown environment in a near-optimal manner. We used a feedforward neural network so as to the robots to learn the sensory distribution function of the environment.

We proved that the controller drives the robots to the estimated centroids of their Voronoi partitions and also causes their estimations of the sensory distribution to improve over time. The numerical simulation results have shown that the proposed controller can effectively position a group of robots sensing over an environment, and increases parameter convergence rates in comparison with the previously proposed adaptive controllers. The authors plan to extend this work by incorporating the consensus algorithm with the proposed method as a future work. In consensus algorithm, the robots can share estimated parameters with their neighbors through an agreement algorithm, which improves convergence rate of the estimation algorithm.

References

Arsie A, Frazzoli E (2007) Efficient routing of multiple vehicles with no explicit communications. Int J Robust Nonlinear Control 2(18):154–164

Bullo F, Cortes J, Martinez S (2008) Distributed control of robotic networks. Electronically available at http://coordinationbook.info

Butler ZJ, Rus D (2004) Controlling mobile sensors for monitoring events with coverage constraints. In: Proceedings of the IEEE international conference of robotics and automation (ICRA), New Orleans, LA, pp 1563–1573

Choset H (2001) Coverage for robotics: a survey of recent results. Ann Math Artif Intell 31:113–126

Cortes J, Martinez S, Bullo F (2005) Spatially-distributed coverage optimization and control with limited-range interactions. ESIAM Control optim Calc Var 11:691–719

Cortes J, Martinez S, Karatas T, Bullo F (2004) Coverage control for mobile sensing networks. IEEE Trans Rob Autom 20:243–255

Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 2:303–314

Drezner Z (1995) Facility location: a survey of applications and methods. Springer Series in Operations Research, Springer, Berlin

Du Q, Faber V, Gunzburger M (1994) Centroidal voronoi tessellations: applications and algorithms. SIAM Rev 41(4):637–676

Gren P, Fiorelli E, Leonard NE (2004) Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment. IEEE Trans Automat Control 8(49):1292–1302

Hertz J, Krogh A, Palmer RG (1991) Introduction to the theory of neural computation. Addison-Wesley, Red-wood City

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2:359–366

Hussein II, Stipanovic DM (2007) Effective coverage control for mobile sensor networks with guaranteed collision avoidance. IEEE Trans Control Syst Technol 4(15):642–657

Kwok A, Martinez S (2008) Deployment algorithms for a power constrained mobile sensor network. In: Proceedings of IEEE International Conference Robotic & Automation, Pasadena, USA, pp 140–145

Latimer D IV, Srinivasa S, Lee-Shue V, Sonne S, Choset H, Hurst A (2002) Towards sensor based coverage with robot teams. In: Proceedings of the IEEE international conference on robotics and automation, vol 1, pp 961–967

Li W, Cassandras CG (2005) Distributed cooperative coverage control of sensor networks. In: IEEE conference on decision and control, European control conference, pp 2542–2547

Martinez S, Cortes J, Bullo F (2007) Motion coordination with distributed information. IEEE Control Syst Mag 4(27):75–88

Pimenta LCA, et al. (2008) Simultaneous coverage and tracking (SCAT) of moving targets with robot networks. In: Proceedings of the 8th international workshop on the algorithmic foundations of robotics (WAFR), Guanajuato, Mexico

Pimenta LCA, Kumar V, Mesquita RC, Pereira GAS (2008) Sensing and coverage for a network of heterogeneous robots. In: CDC, pp 3947–3952

Sanner R, Slotine J (1992) Gaussian networks for direct adaptive control. IEEE Trans Neural Netw 3(6): 837–863

Salapaka S, Khalak A, Dahleh MA (2003) Constraints on locational optimization problems. In: Proceedings of the conference on decision and Control, Maui, HI, vol 2, pp 1430–1435

Schwager M, Julian B, Rus D (2009) Optimal coverage for multiple hovering robots with downward-facing cameras. In: Proceedings of international conference on robotics and automation(ICRA), Kobe, Japan

Schwager M, McLurkin J, Rus D (2006) Distributed coverage control with sensory feedback for networked robots. In: Proceedings of Robotics: Science and Systems, Philadelphia, PA

Schwager M, McLurkin J, Slotine JJE, Rus D (2008) From theory to practice: distributed coverage control experiments withgroups of robots. In: Proceedings of the international symposium on experimental robotics (ISER 08), Athens, Greece

Schwager M, Rus D, Slotine J-JE (2009) Decentralized, adaptive control for coverage with networked robots. Int J Rob Res 28:357–375

Schwager M, Slotine J-J, Rus D (2007) Decentralized, adaptive control for coverage with networked robots. In: international conference on robotics and automation(ICRA), Rome

Schwager M, Slotine J-J, Rus D (2008) Consensus learning for distributed coverage control. In: Proceedings of international conference on robotics and automation, Pasadena, CA, pp 1042–1048

Slotine JJE, Coetsee JA (1986) Adaptive sliding controller synthesis for nonlinear systems. Int J Control 43(6):1631–1651

Wang H, Guo Y (2008) A decentralized control for mobile sensor network effective coverage. In: 7th World congress on intelligent control, Chongqing, China