# Adaptive Inter-Robot Trust for Robust Multi-Robot Sensor Coverage

Alyssa Pierson and Mac Schwager

**Abstract** This paper proposes a new approach to both characterize inter-robot trust in multi-robot systems and adapt trust online in response to the relative performance of the robots. The approach is applied to a multi-robot coverage control scenario, in which a team of robots must spread out over an environment to provide sensing coverage. A decentralized algorithm is designed to control the positions of the robots, while simultaneously adapting their trust weightings. Robots with higher quality sensors take charge of a larger region in the environment, while robots with lower quality sensors have their regions reduced. Using a Lyapunov-type proof, it is proven that the robots converge to locally optimal positions for sensing that are as good as if the robots' sensor qualities were known beforehand. The algorithm is demonstrated in Matlab simulations.

## 1 Introduction

Multi-robot systems have the capacity to carry out large scale tasks efficiently. However, in order to be practical in real-world settings, multi-robot systems should be robust to the deficiencies of individual robots. In this work we consider the problem of decentralized coverage control in the case when different robots may have different, but unknown, sensing qualities. We propose an online, adaptive method to compensate for the relative differences in sensing quality using only information from the robots' sensor readings. The robots estimate a "trust weighting" online, which they use to adjust their sensing load.

The necessity for adaptive trust can be illustrated through several examples. First, consider a situation in which a group of robots is deployed over a region to take pictures following some disaster, such as an earthquake or building collapse. The quality of sensors may degrade differently, for example, due to dust or cracks on the

Alyssa Pierson and Mac Schwager
Boston University, Boston MA 02215, e-mail: pierson@bu.edu, schwager@bu.edu

camera lens. Our algorithm accounts for the sensor quality and adapts trust weightings accordingly. Even in the most benign situations, sensor creep can occur causing uneven degradation in sensing performance. As sensor creep occurs in the group, adaptive trust weightings adjust for the lower-performing robots and increases the overall integrity of the group data collected.

A common strategy for coverage control, first proposed by Cortés et al., is based on Voronoi tessellations of the environment ([2], [3]). This strategy drives all robots to the centroids of their Voronoi cells, also referred to as the move-to-centroid controller. It is known from previous research that the centroidal Voronoi configuration has optimal properties for minimizing distances to points [4], as well as applications in data compression [5]. Other extensions have been proposed with the weighted Voronoi cell where the weightings account for differences in agent performance and sensor qualities. Pavone et. al illustrated that using weighted Voronoi diagrams, also known as Power Diagrams, the different cell weights allow for different agents to take on varying sensing responsibility [12]. Another method considers the sensing radius as the Voronoi weighting [13], which is useful in a heterogeneous group of robots. Another application defined the weight as a measure of energy-efficiency of a robot, allowing the group to compensate for low-energy robots [9]. Marier et. al have used the Voronoi weightings to quantify sensor health of each robot, assigning low-performing robots smaller areas of coverage and higher sensing costs ([10], [11]).

While there is a wide variety of existing research on weighted Voronoi cells with respect to robot performance, most assume the correct weightings are known *a priori*. In contrast, our work proposes an algorithm to adapt trust weightings online using only comparisons between a robot's sensor measurements, and those of its neighbors. We integrate a measure of sensor discrepancy into a cost function for the group, and use this to derive an adaptation law for each robot to change its trust weightings online, while simultaneously performing a Voronoi based coverage control algorithm. We prove that the system converges to a local minimum of the cost function using a Lyapunov proof. The weightings serve as an adaptive way to assess trust between agents and improve the overall sensing quality of the group.

## 2 Problem Set-Up

Consider a set of $n$ robots in a bounded, convex environment $Q \subset \Re^2$. A given point in $Q$ is denoted $q$, and let the position of the $i$-th agent be $p_i \in Q$. Prior coverage control algorithms use the standard Voronoi partition of the environment. Let $\{V_1, ..., V_n\}$ be the Voronoi partition of $Q$, with each cell satisfying the Voronoi definition

$$V_i = \{q \in Q \mid \|q - p_i\| \leq \|q - p_j\|, \quad \forall j \neq i\}.$$

For our work, we use the weighted Voronoi partition, also known as the Power Diagram, with each weighting $w_i$ serving as the trust weighting for robot $i$. Let

$\{W_1, \dots, W_n\}$ be the weighted Voronoi partition of $Q$, with each cell satisfying

$$W_i = \{q \in Q \mid ||q - p_i||^2 - w_i \leq ||q - p_j||^2 - w_j, \quad \forall j \neq i\}. \tag{1}$$

The trust weighting for robot $i$ is $w_i$, and it has the effect of increasing or reducing the size of its associated Voronoi cell. Figure 1 illustrates the differences between a weighted and standard Voronoi cell.

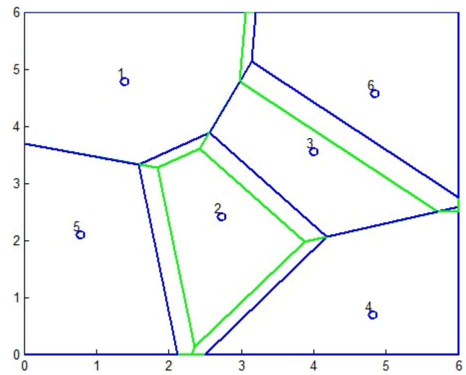


**Fig. 1** The regular (blue) and weighted (green) Voronoi cell for a six-robot configuration. Here, robot 2 has a lower weighting than the other robots and robot 6 has a higher weighting, which is reflected in the changes in cell area.

In Figure 1, both the regular and weighted Voronoi cells are drawn to illustrate the effect of the weightings on the Voronoi cell boundaries. As shown, the trust weighting of robot two is lower than its neighbors, giving it a decreased cell area. Conversely, the trust weighting of robot six is higher, giving it the increased area.

For our bounded region $Q$, we also define an integrable function $\phi : Q \to \Re_{>0}$ to represent the areas of importance in the environment. Areas with large values of $\phi(q)$ are more important than those with small values, and all the robots have knowledge of this function. When the robots do not know this function, techniques have been developed to learn it online from sensor data [16]. We also introduce the sensing function $\gamma_i(p_i, q)$ to model the relationship between sensor health and data sensed by the robot. This function uses a quadratic approximation to model how the health affects a sensor reading. Unlike $\phi(q)$, $\gamma_i(p_i, q)$ may take different values by different robots looking at the same point. For example, if robot $i$ uses a camera for sensing, $\gamma_i(p_i, q)$ may be the brightness of the pixel looking at $q$ while the robot is located at $p_i$. Robot $i$'s camera positioned at $p_i$ may have a different value for some point $q$ than robot $j$'s camera positioned at $p_j$.

### 2.1 Locational Optimization

Before introducing our problem formulation, we will state the basic nomenclature and results from Locational Optimization. A complete discussion can be found in [3]. We can formulate a cost function for the sensing network over the area $Q$ as

$$\mathscr{H}(p_1,...,p_n) = \sum_{i=1}^{n} \int_{V_i} \frac{1}{2} \|q-p_i\|^2 \phi(q) dq. \tag{2}$$

Note that sensing cost increases as robots move further away from high values of $\phi(q)$. Intuitively, a low value of $\mathscr{H}$ would indicate a good configuration of the robots for coverage of the environment. Two other useful quantities to define from this formulation are the mass and centroid of a Voronoi region $V_i$, respectively, as

$$M_{V_i} = \int_{V_i} \phi(q) dq, \text{ and } C_{V_i} = \frac{\int_{V_i} q\phi(q)dq}{M_{V_i}}.$$

Given that $\phi(q)$ is strictly positive, both $M_{V_i}$ and $C_{V_i}$ are analogous to physical masses and centroids of the Voronoi cells. Although there is a complex dependency between robot position and the geometry of the Voronoi cells, a surprising result from locational optimization [4] is that

$$\frac{\partial \mathscr{H}}{\partial p_i} = -\int_{V_i} (q-p_i)\phi(q)dq = -M_{V_i}(C_{V_i} - p_i). \tag{3}$$

Equation (3) implies that the critical points of $\mathscr{H}$ correspond to the configurations in which all robots are located at the centroid of their Voronoi cell, or $p_i = C_{V_i}$ for all $i$. Critical points can either correspond to local minimum, maximum, or saddle points. Cortés introduced a gradient-based controller that is guaranteed to drive the robots to the critical points corresponding to local minimum [3]. The controller we use here also has this property. We restrict ourselves to only considering local minima of $\mathscr{H}$ since global optimization of (2) is known to difficult (NP-hard). Thus, when we refer to optimal coverage configurations, we mean locally optimal configurations. Variations on the control law which attempt to find global minima through exploration are discussed by Salapaka et al. [14] and Schwager et al. [15].

Our formulation introduces trust weightings for each agent as an additional optimization variable. These weightings are used in calculating the weighted Voronoi cell, also known as the Power Cell, for each agent, given in (1). We still wish to formulate this as a locational optimization problem, and use a modified cost function written

$$\mathscr{W}(p_1,...,p_n,w_1,...,w_n) = \sum_{i=1}^{n} \int_{W_i} \frac{1}{2} \left(\|q-p_i\|^2 - w_i\right) \phi(q) dq, \tag{4}$$

where $W_i$ is the robot's weighted Voronoi cell, and $w_i$ is the robot's individual trust weighting. Note that this is almost identical to the formulation in (2), except the integral is calculated over the weighted Voronoi cell instead of the standard Voronoi cell. Additionally, we have added $w_i$ to the integrand, giving it the same form as the weighted Voronoi cell definition (1).

Similar to the original cost function, we can also define the mass and centroid of the weighted Voronoi cell, respectively, as

$$M_{W_i} = \int_{W_i} \phi(q)dq, \text{ and } C_{W_i} = \frac{\int_{W_i} q\phi(q)dq}{M_{W_i}}.$$

From this, we can take the partial derivative of the cost function with respect to agent positions, and we find

$$\frac{\partial \mathscr{W}}{\partial p_i} = -\int_{W_i} (q - p_i)\phi(q)dq = -M_{W_i}(C_{W_i} - p_i), \tag{5}$$

which implies that critical points of $\mathscr{W}$ will also correspond to robots positioned at the centroids of their weighted Voronoi cells [10]. Using (5), we will introduce a controller similar to the Cortés controller that only moves the robots towards the local minima.

## 2.2 Robot and Sensor Model

In this section, we describe our model for the dynamics of the robots and the quality of the sensor. First, we assume that the robots have integrator dynamics, where

$$\begin{aligned} \dot{p}_i &= u_{i,1}, \text{ and} \\ \dot{w}_i &= u_{i,2}. \end{aligned} \tag{6}$$

Here, $u_{i,1}$ is the control input to the robot, and $u_{i,2}$ is an adaptation law for the weightings. We can equivalently assume there are low-level controllers in place to cancel existing dynamics and enforce (6). We also assume that the robots will be able to communicate with their neighbors and share information about data sensed. The communication network is defined as an undirected graph in which two robots share an edge of the graph if they share Voronoi cell boundaries. This is also known as the Delaunay graph. We can then write the set of neighbors for any robot $i$ as $\mathscr{N}_i := \{j | V_i \cup V_j \neq 0\}$. Additionally, robots are able to compute their own weighted Voronoi cells, as defined by (1), which is a common assumption in the literature ([3], [10], [14]).

To model the effect of sensor health on performance, we consider a specific form for the sensor function $\gamma_i(p_i, q)$. This relates the impact of health as a quadratic approximation of the actual function near points of comparison. While in practice it is not necessary to know $\gamma_i$, for our convergence proofs we assume that $\gamma_i$ can be approximated by

$$\gamma_i(p_i, q) = -\alpha \left( \|q - p_i\|^2 - h_i \right), \tag{7}$$

where $h_i$ is some health offset indicative of sensor performance and $\alpha$ is some scaling factor. Note that this equation for $\gamma_i(p_i,q)$ shares a similar structure with the weighted Voronoi cell definition (1). It is not necessary for the robots to know $h_i$ or $\alpha$ for a given sensor so long as $\gamma_i(p_i,q)$ can be measured from the robot's sensor. For example, imagine that $\gamma_i(p_i,q)$ conveys pixel brightness captured from a camera. While the robot may not know the camera health, it is still capable of obtaining pixel brightness. The $\alpha$ and $h_i$ variables shape the approximation of how the health affects the quality at some point $q$ from $p_i$. We believe this is a valid model for sensor quality, as the performance of sensing some point $q$ decreases as $q$ moves away from the sensor at $p_i$. This also allows different sensors to have a different performance. We also note that $\gamma_i(p_i,q)$ can be extended to any 2D sensor model, not just cameras. Another example to consider is an implementation where the robots have lidar sensors to map an environment topography, and $\gamma_i(p_i,q)$ is the elevation measurement at point $q$ for the robot's sensor positioned at $p_i$.

## 3 Decentralized Control

The main goals of our work are to 1) drive the robots to an optimal coverage configuration in the environment and 2) adjust trust weightings to account for variations in sensing performance. To accomplish these goals, we propose one control law to change the positions of the robots and one adaptation law to change the weightings of the robots. We will then prove that both of these control laws will drive the robots to converge asymptotically to a stable equilibrium configuration corresponding to a local minimum of the sensing cost function.

With respect to the position controller, we will use the control law

$$\dot{p}_i = u_{i,1} = k_p(C_{W_i} - p_i), \tag{8}$$

where $k_p$ is a positive proportional gain constant, and $C_{W_i}$ is the centroid of the weighted Voronoi cell. This controller is commonly referred to as the move-to-centroid control law, first proposed by Cortés ([3]) and extended and modified in ([1],[10], [16]). While the original control law used the unweighted Voronoi cell centroid, $C_{V_i}$, it does not impact the performance of the controller to use the weighted Voronoi centroid, $C_{W_i}$ ([10], [13]).

For the weightings, we propose a new adaptation law

$$\dot{w}_i = u_{i,2} = \frac{k_w}{M_{W_i}} \sum_{j \in \mathcal{N}_i} \left( \int_{b_{ij}} \gamma_i(p_i,q_c)dq - \frac{\int_{b_{ij}} \gamma_i(p_i,q_c)dq + \int_{b_{ij}} \gamma_j(p_j,q_c)dq}{2} \right) \tag{9}$$

where $k_w$ is a positive proportional gain constant, and $b_{ij}$ is the cell boundary line between neighboring agents $i$ and $j$. Essentially, this compares values of sensing data between two neighbors over shared points along their boundaries. Figure 2 illustrates the shared boundary.

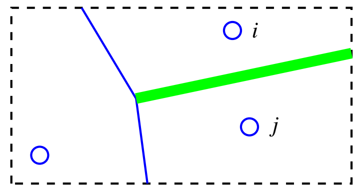


**Fig. 2** For neighbors $i$ and $j$ the green line highlights their shared Voronoi cell boundary. In the weightings adaptation law, sensing data is compared along points in this boundary.

The control law $u_{i,1}$ is referred to as the positional controller, and the control law $u_{i,2}$ is the weightings adaptation law. The behavior of the system with these control laws is formalized in the following theorem.

**Theorem 1.** *Using the positional control law* (8) *and the weightings adaptation law* (9)*, the robots converge to an asymptotically stable local minimum of the sensing cost function* $\mathscr{W}(p_1,...,p_n,w_1,...,w_n)$ (4)*. Furthermore, the positions of the robots satisfy*

$$\|p_i - C_{W_i}\| \to 0 \ \ \forall \ \ i \in n, \tag{10}$$

*and the weightings satisfy*

$$(w_i - w_j) \to (h_i - h_j) \ \ \forall \ \ i, j. \tag{11}$$

*Proof.* We will first show that the controllers drive the system to stable equilibria by using the cost function $\mathscr{W}$ as a Lyapunov function candidate. We will then introduce a new Lyapunov function to show that the adaptation law for the weightings, $u_{i,2}$, also drives the weightings to the set defined in (11).

Consider our cost function $\mathscr{W}$ in (4) as a Lyapunov-like function. Taking the time derivative of this function yields

$$\dot{\mathscr{W}} = \sum_{i=1}^{n} \int_{W_i} (q - p_i)^T \phi(q) dq \dot{p}_i + \sum_{i=1}^{n} \int_{W_i} \frac{1}{2} \phi(q) dq \dot{w}_i.$$

We can break this expression into two parts as

$$\dot{\mathscr{W}}_1 = \sum_{i=1}^{n} \int_{W_i} (q - p_i)^T \phi(q) dq \dot{p}_i, \quad \dot{\mathscr{W}}_2 = \sum_{i=1}^{n} \frac{1}{2} M_{W_i} \dot{w}_i.$$

Plugging in our adaptation law $u_{i,2}$ (9) for $\dot{w}_i$, $\dot{\mathscr{W}}_2$ simplifies as

$$\begin{aligned}
\dot{\mathscr{W}}_2 &= \sum_{i=1}^{n} \frac{1}{2} M_{W_i} \frac{k_w}{M_{W_i}} \sum_{j \in \mathscr{N}_i} \left( \frac{\int_{b_{ij}} \gamma_i(p_i, q) dq - \int_{b_{ij}} \gamma_j(p_j, q) dq}{2} \right) \\
&= \sum_{i=1}^{n} \frac{k_w}{4} \sum_{j \in \mathscr{N}_i} \int_{b_{ij}} [\gamma_i(p_i, q) - \gamma_j(p_j, q)] \, dq \\
&= 0.
\end{aligned}$$

Now consider $\dot{\mathscr{W}}_1$. By plugging in our controller $u_{i,1}$ (8) for $\dot{p}_i$, the time derivative of the cost function becomes

$$\begin{aligned}\dot{\mathscr{W}} = \dot{\mathscr{W}}_1 &= \sum_{i=1}^{n} \int_{W_i} (q-p_i)^T \phi(q) dq \left[k_p (C_{W_i} - p_i)\right] \\ &= \sum_{i=1}^{n} -k_p M_{W_i} \|C_{W_i} - p_i\|^2 \leq 0. \end{aligned} \tag{12}$$

Using La Salle's Invariance Principle [8], the robots converge to the largest invariant set such that $\dot{\mathscr{W}} = 0$. From (12), when $p_i = C_{W_i}$ for all $i$, then $\dot{\mathscr{W}} = 0$. From our control law (8), when $p_i = C_{W_i}$, $\dot{p}_i = 0$ for all $i$, therefore the centroidal Voronoi configuration $p_i = C_{W_i}$ $\forall$ $i$ is the largest invariant set. By La Salle's, the robots converge to the centroidal configuration, proving (10) from Theorem 1.

In order to prove (11) from Theorem 1, consider a second Lyapunov-like function,

$$\mathscr{V} = \sum_{i=1}^{n} \frac{1}{2} \|w_i - h_i\|^2$$

with time derivative

$$\begin{aligned}\dot{\mathscr{V}} &= \sum_{i=1}^{n} (w_i - h_i)^T \dot{w}_i \\ &= \sum_{i=1}^{n} (w_i - h_i)^T \frac{k_w}{2M_{W_i}} \sum_{j \in \mathscr{N}_i} \int_{b_{ij}} \left[\gamma_i(p_i, q) - \gamma_j(p_j, q)\right] dq. \end{aligned}$$

To simplify this expression, we notice from (7) that

$$\gamma_i(p_i, q) - \gamma(p_j, q) = -\alpha \left( \|q - p_i\|^2 - h_i - \|q - p_j\|^2 + h_j \right).$$

However, we are evaluating point $q$ at the cell boundary, so it will satisfy (1)

$$\|q - p_i\|^2 - w_i = \|q - p_j\|^2 - w_j.$$

Combining these expressions, we find

$$\gamma_i(p_i, q) - \gamma(p_j, q) = -\alpha \left(w_i - w_j - h_i + h_j\right)$$

The difference in sensing quality is constant between two neighboring robots along the boundary $b_{ij}$. Thus, when we plug this into our expression for $\dot{\mathscr{V}}$, we obtain

$$\begin{aligned}\dot{\mathscr{V}} &= \sum_{i=1}^{n} (w_i - h_i)^T \frac{k_w}{2M_{W_i}} \sum_{j \in \mathscr{N}_i} -\alpha \left(w_i - w_j - h_i + h_j\right) \int_{b_{ij}} dq \\ &= \sum_{i=1}^{n} (w_i - h_i)^T \frac{\alpha k_w}{2M_{W_i}} \sum_{j \in \mathscr{N}_i} \left(w_j - h_j - w_i + h_i\right) d_{ij} \end{aligned} \tag{13}$$

where $d_{ij}$ is the length of the boundary $b_{ij}$. It is advantageous to re-write this expression in matrix form. To do so, we will define

$$\tilde{w} = \begin{bmatrix} w_1 - h_1 \\ \vdots \\ w_n - h_n \end{bmatrix}, \qquad M^{-1} = \begin{bmatrix} \frac{1}{M_{W_1}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{M_{W_n}} \end{bmatrix}, \text{ and}$$

$$L = \begin{bmatrix} \ddots & & L_{ij} \\ & \sum_{j \in \mathscr{N}_i} d_{ij} & \\ L_{ij} & & \ddots \end{bmatrix}, \quad L_{ij} = \begin{cases} -d_{ij} & \text{for } j \in \mathscr{N}_i \\ 0 & \text{otherwise} \end{cases}.$$

Hence from (13) we can write the derivative of the Lyapunov function in matrix form as

$$\dot{V} = -\alpha k_w \tilde{w}^T M^{-1} L \tilde{w}.$$

$M^{-1}$ is a diagonal matrix of positive entries and $L$ is the weighted Laplacian of the neighbor graph, which is known to be positive semi-definite ([6], [7]). It can be shown that the product $M^{-1}L$ is positive semi-definite, which allows us to state

$$\dot{\mathscr{V}} \leq 0.$$

To complete the proof, we use La Salle's Invariance Principle to find the largest invariant set such that $\dot{\mathscr{V}} = 0$. The invariant set is defined as when $\tilde{w}$ is in the null space of $L$. From graph theory, we know this occurs when $\tilde{w}$ is a vector of identical entries, i.e. $\tilde{w}_i = \tilde{w}_j$ for all neighbors. This can also be written as the set of all $w_i$ such that

$$w_i - h_i = w_j - h_j \quad \forall\, i, j \in n$$

or

$$(w_i - w_j) = (h_i - h_j) \quad \forall\, i, j \in n,$$

proving (11) from Theorem 1. □

*Remark 1.* This proof shows that using the controller $u_{i,2}$ (9), our weightings converge to a set of values relating the robot trust weightings back to sensing performance. Overall, the convergence of the weightings implies they will reach static values, which in conjunction with the move-to-centroid controller means that the robots will find final locations in the environment. Although changing the weightings creates a change in boundaries and thus a change in the cell centroids, the weightings eventually converge to an invariant set, which means the positions of the robots will eventually reach their centroids.

*Remark 2.* Theorem 1 guarantees convergence to a relative difference between the weightings and the health factor, not the direct health value. This is as expected, since from our problem setup, trust is a relative notion among neighboring agents with no external authority. Additionally, weighted Voronoi cell boundaries are calculated from a relative difference (1), as any constant offset would be canceled out on either side.

*Remark 3.* The convergence of the robots to locally optimal locations in the environment is as good as if the correct robot trust weightings were known beforehand. If the weightings are correct, it implies all robots will agree in compared sensing data values. In this case, $\dot{w}_i$ goes to zero (9), while the positional controller $\dot{p}_i$ remains the same (8).

*Remark 4.* One simplification of the weightings adaptation law is to compare the sensing values between neighbors at any subset of points in $b_{ij}$, including a single point, instead of across the entire boundary. The motivation to compare sensing functions at fewer points, as illustrated by Figure 3, is that it may be quicker and computationally easier than the boundary calculation, albeit less robust. Corollary 1 shows that this simplification of any subset of points still maintains convergence of the weightings to an invariant set, as well as convergence of the location of the robots to their centroids.

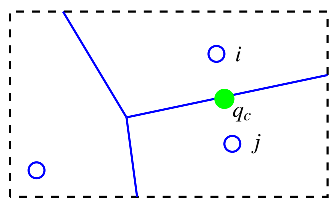


**Fig. 3** For neighboring robots $i$ and $j$, the weighted midpoint $q_c$ (green) lies along the shared Voronoi boundary (blue).

**Corollary 1.** *The claims of Theorem 1 also hold true for the adaptation law*

$$\dot{w}_i = u_{i,3} = \frac{k_w}{2M_{W_i}} \sum_{j \in \mathcal{N}_i} \left(\gamma_i(p_i, q_c) - \gamma_j(p_j, q_c)\right), \tag{14}$$

*where $q_c$ is any point in $b_{ij}$.*

*Proof.* Using (14) in place of the previous adaptation law (9) and noting that the weighted graph Laplacian becomes the normal graph Laplacian [6], the same proof and arguments hold from Theorem 1. □

## 4 Simulation Results

Simulations were carried out in a Matlab environment. The controllers in (8) and (14) were applied to a group of $n = 10$ robots. Riemann sums were used to approximate integrals in calculating the controllers, the weighted Voronoi cell masses and centroids, and the cost function (4). The environment $Q$ was defined as a square region. The information density function $\phi(q)$ was defined as constant in Scenario A and B, and as a sum of two Gaussian functions, with peaks in the upper right quadrant and lower left quadrant in Scenario C.

All robots are initialized with random positions, and three scenarios are included in this paper. Scenario A starts with equal weightings but unequal sensing performance. Scenario B starts with unequal weightings but equal sensing performance. Scenario C starts with randomized weightings and health factors for all agents.

## *4.1 Scenario A*

In this scenario, all robots start with equal trust weightings. However, robot $i = 2$ has a lower sensor health $h_2$, which implies it is not performing at its expected ability. The initial weighting and health values are

$$w_0 = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0],$$
$$h_0 = [1.0, 0.2, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0].$$

Over the course of the simulation we notice that the weighting $w_1$ decreases as a result of its lower health. Figure 4 shows a comparison between final configurations with and without the adaptive trust weightings, while Figure 5 shows the global sensing cost and trust weightings over time.

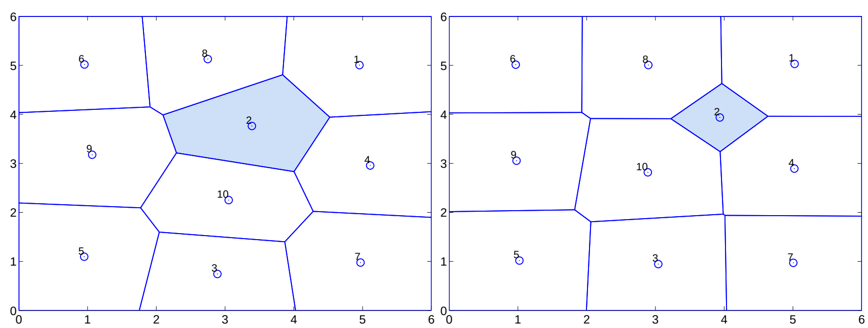


**Fig. 4** Final configurations without adaptive weightings (left) and with adaptive weightings.

We can see in Figure 4 that without adaptive trust weightings, the lower-health robot is able to get an equal share of the environment sensing load. With the adaptive weightings, its contribution is reduced to improve the overall group quality. To verify the system converges as predicted by Theorem 1, see Figure 5.

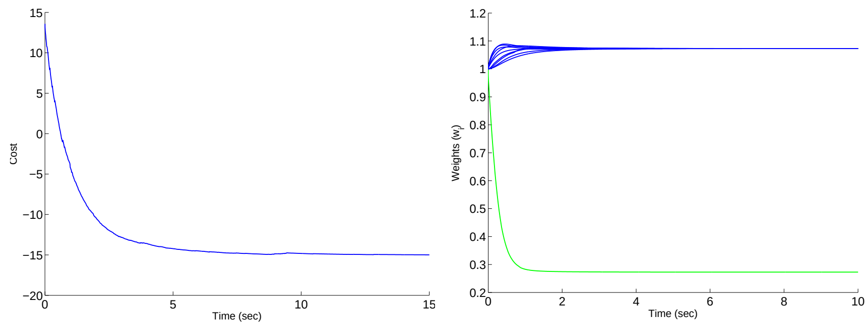


**Fig. 5** [Left] Global sensing cost of the system over time. [Right] Convergence of weighting values over time. Values for $w_2$ are shown in green, while the rest of the group is shown in blue.

The values of the global cost function decrease over time, a result of the move-to-centroid controller $u_{i,1}$. In addition, by looking at the new weighting values, we see that the weight $w_2$ did indeed drop in value, while the other weights remained equal. The final values for the weightings taken after 100 seconds were

$$\begin{aligned} w_f &= [1.07, 0.27, 1.07, 1.07, 1.07, 1.07, 1.07, 1.07, 1.07, 1.07], \\ w-h &= [0.07, 0.07, 0.07, 0.07, 0.07, 0.07, 0.07, 0.07, 0.07, 0.07]. \end{aligned}$$

Consistent with our predictions, the difference between $w_i$ and $h_i$ is equal in value for all robots. Note that the values of $w_i$ do not converge to the exact values of $h_i$, only the difference.

### *4.2 Scenario B*

To illustrate the converse of Scenario A, we create a situation in which the robots are performing equally, but robot $i = 3$ has been initially assigned a lower trust weighting than the rest of the group. We show that the weightings will eventually converge to the same value when the sensing healths are equal. Initially, the weightings and sensing healths were assigned to be

$$\begin{aligned} w_0 &= [1.0, 1.0, 0.2, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0], \\ h_0 &= [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]. \end{aligned}$$

As we have stated previously, it is not necessary in practice to know the sensor health $h_i$, but we will assign these values in simulation to show functionality. With all of the sensing health values set equal, we expect that the weightings will coverage to equal values as well.

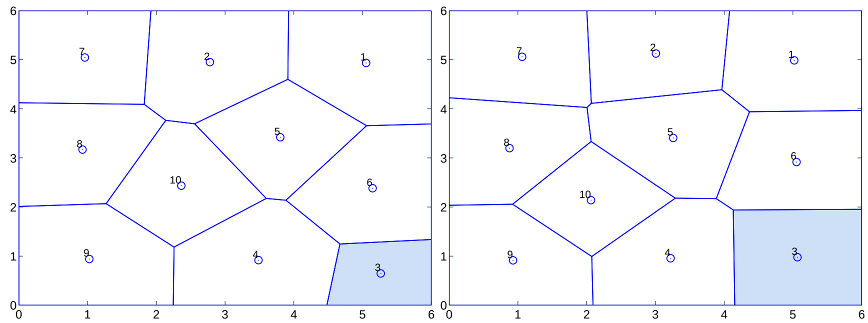

**Fig. 6** Final configurations without adaptive weightings (left) and with adaptive weightings.

Figure 6 shows the final configurations with and without adaptive weightings. By using the adaptive weightings, the incorrectly-assigned trust weighting is corrected at robot three is given a more-equal share of the sensing load. Figure 7 shows the cost plot and the weightings over time.

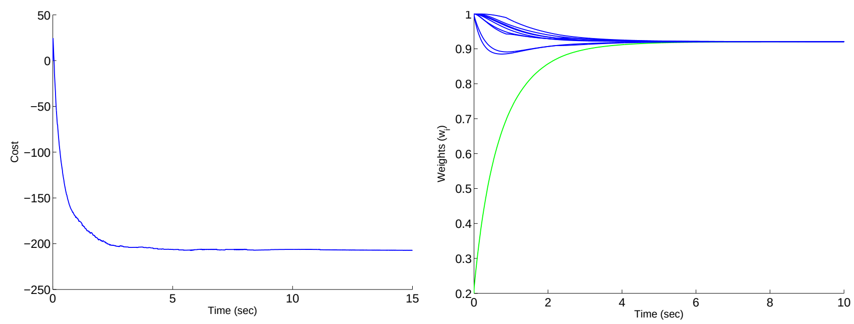


**Fig. 7** [Left] Global sensing cost of the system over time. [Right] Convergence of weighting values over time. Values for $w_3$ are shown in green, while the rest of the group is shown in blue.

From the plot of the weightings, we see that $w_3$ moves towards the other weights over time. The final weightings values after 100 seconds were

$$w_f = [0.92, 0.92, 0.92, 0.92, 0.92, 0.92, 0.92, 0.92, 0.92, 0.92],$$
$$w - h = [-0.08, -0.08, -0.08, -0.08, -0.08, -0.08, -0.08, -0.08, -0.08, -0.08].$$

As predicted, with all health values equal, the weightings converge to equal values. Note that even though the health hasn't changed, the values of the weightings decrease. From Remark 2, we know that the decrease is not important, so long as the difference between all agents is equal. It makes no difference on the Voronoi configuration if the weightings are all 0.92 or 1.0, because the relative difference is the same.

### 4.3 Scenario C

The previous two scenarios used simple initial values to show in detail how the weightings adapt over time. In this scenario, the weightings and the sensing health factors were initialized as random numbers drawn from the uniform distribution over $[0,1]$ to illustrate more complex functionality. The initial values were:

$$w_0 = [0.66, 0.63, 0.29, 0.43, 0.02, 0.98, 0.17, 0.11, 0.37, 0.20],$$
$$h_0 = [0.49, 0.34, 0.95, 0.92, 0.05, 0.74, 0.27, 0.42, 0.55, 0.94].$$

Similar to before, the simulation was run in Matlab, with initial and final configurations shown in Figure 8. We expect that our weightings controller will drive the difference $w_i - h_i$ to equal values amongst the group while still maintaining coverage control. From Figure 8, we see the algorithm is able to accomplish a centroidal Voronoi configuration from randomized initial positions.

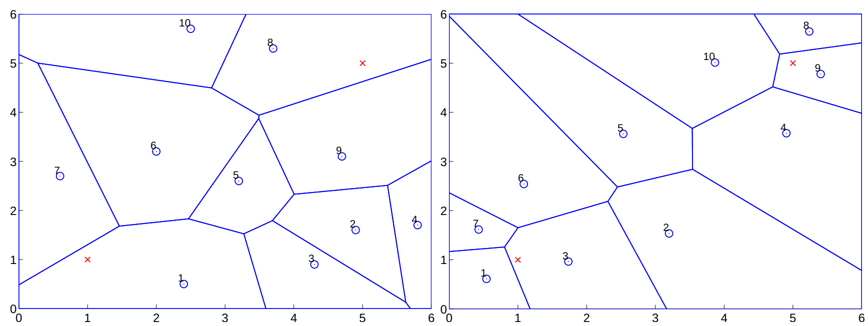

**Fig. 8** Initial (left) and final configurations of the robots.

To verify that this indeed is a better final position, we can see the cost function decreases to a final value in Figure 9. We also observe that the final values of $w-h$, taken after 100 seconds, are

$$w-h = [-0.15, -0.15, -0.15, -0.15, -0.15, -0.15, -0.15, -0.15, -0.15, -0.15].$$

These values show convergence to the common invariant set described in Theorem 1. Figure 9 shows that all of these weightings relative to their respective health factor converge over time. Note that this plot is the relative difference $w-h$, which is different than what is plotted in Scenario A and B.

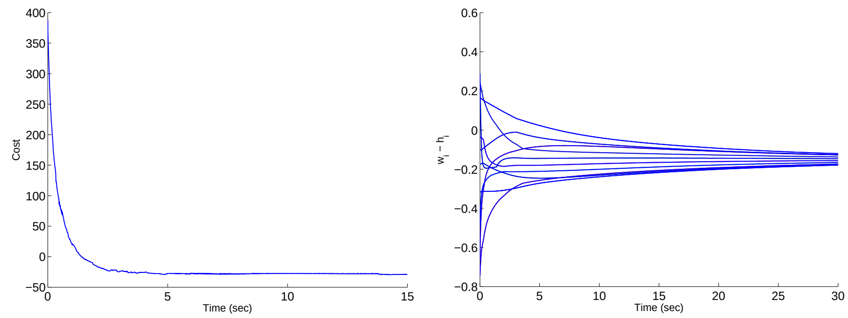


**Fig. 9** [Left] Global sensing cost of the system over time [Right] For each agent, the difference $w-h$ is plotted, showing convergence to a common value.

We plot the difference $w-h$ to better illustrate the convergence, as plotting the actual values of the weightings would seem random and erratic. From this, we see all values coming into agreement, as predicted by (11). In conjunction with the decreasing cost function, we can verify that the final position is locally optimal and stable.

## 5 Conclusion

In this paper we have described a method for quantifying robot-to-robot trust in a multi-robot coverage control application. Specifically, we allow the robots in the group to compare values of data sensed with their neighbors, and using an adaptive control law, adjust their weightings to better account for their performance. These trust weightings adjust the placement of the weighted Voronoi boundaries between neighboring robots. By controlling the weights on each Voronoi cell, we are able to adjust a robot's cell size relative to its neighbors, which is analogous to creating trust relationships between neighboring robots. The weightings adaptation law was proven to converge to an asymptotically stable invariant set, which is shown to be as good as knowing the health factors directly. The positional controller was similar to positional controllers in previous works in that it moved robots towards the centroid of their Voronoi cells.

Our method can be used to incorporate the robot sensor degradation into the overall decentralized algorithm while maintaining stability and performance. This will provide robustness in real-world coverage control applications. First, it can adjust for variations in the degradation of sensing performance caused by external factors, such as dust on a camera lens. Second, it provides robustness against sensor creep over long periods of time. Finally, in applications where there may be malicious robots in the group, these trust weightings can provide insight into identifying and mitigating against malicious robots. Future extensions of this work aim to further investigate the relationship of trust and adversarial robots. This paper provides the framework for quantifying trust, but additional steps are still needed to ensure that the impact caused by adversarial robots is limited. Another extension is to apply this concept of trust to applications beyond coverage control, such as multi-agent mapping, target tracking, or search.

**Acknowledgements** This work was supported in part by ONR grant N00014-12-1-1000, and by a Clare Boothe Luce Fellowship. We are grateful for this financial support.

## References

[1] Breitenmoser A, Schwager M, Metzger JC, Siegwart R, Rus D (2010) Voronoi coverage of non-convex environments with a group of networked robots. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, pp 4982–4989
[2] Cortés J (2010) Coverage optimization and spatial load balancing by robotic sensor networks. Automatic Control, IEEE Transactions on 55(3):749–754
[3] Cortes J, Martinez S, Karatas T, Bullo F (2004) Coverage control for mobile sensing networks. Robotics and Automation, IEEE Transactions on 20(2):243–255
[4] Drezner Z (1995) Facility location: a survey of applications and methods. Springer series in operations research, Springer
[5] Du Q, Faber V, Gunzburger M (1999) Centroidal voronoi tessellations: Applications and algorithms. SIAM review 41(4):637–676
[6] Godsil C, Royle G (2001) Algebraic Graph Theory. Graduate Texts in Mathematics, Springer-Verlag
[7] Horn RA, Johnson CR (1990) Matrix analysis. Cambridge university press
[8] Khalil H (2002) Nonlinear Systems. Prentice Hall PTR
[9] Kwok A, Martinez S (2007) Energy-balancing cooperative strategies for sensor deployment. In: Decision and Control, 2007 46th IEEE Conference on, IEEE, pp 6136–6141
[10] Marier JS, Rabbath CA, Léchevin N (2011) Optimizing the location of sensors subject to health degradation. In: American Control Conference (ACC), 2011, IEEE, pp 3760–3765
[11] Marier JS, Rabbath CA, Léchevin N (2012) Health-aware coverage control with application to a team of small uavs
[12] Pavone M, Arsie A, Frazzoli E, Bullo F (2009) Equitable partitioning policies for robotic networks. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, pp 2356–2361
[13] Pimenta L, Kumar V, Mesquita RC, Pereira G (2008) Sensing and coverage for a network of heterogeneous robots. In: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, IEEE, pp 3947–3952
[14] Salapaka S, Khalak A, Dahleh M (2003) Constraints on locational optimization problems. In: Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, IEEE, vol 2, pp 1741–1746
[15] Schwager M, Bullo F, Skelly D, Rus D (2008) A ladybug exploration strategy for distributed adaptive coverage control. In: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, IEEE, pp 2346–2353
[16] Schwager M, Rus D, Slotine JJ (2009) Decentralized, adaptive coverage control for networked robots. The International Journal of Robotics Research 28(3):357–375