Coverage Control Overview

Simon Hu

1 Coverage Control Overview

1.1 Background

Let $\mathbf{p} = (p_1, p_2, \ldots, p_N) \in \mathbb{R}^N$ be a vector whose elements are the positions of the i-th agent, given by $p_i \in \mathbb{R}^d$ where $d = 1, 2, \ldots$. The goal of the coverage control algorithm is to solve the following problem.

$$\max_{\mathbf{p}\in\Xi^N} \mathcal{H}_\varphi(\mathbf{p}, t) := \max_{\mathbf{p}\in\Xi^N} \int_\Xi \min_{i=1,2,\ldots,N} \|\boldsymbol{\xi} - \mathbf{p}_i\|_2^2 \, \varphi(\boldsymbol{\xi}, t) \, \mathrm{d}\boldsymbol{\xi} \tag{1.1}$$

In other words, the goal is to find the optimal configuration of agent positions $\mathbf{p}$ so that the desired area to cover, which is encoded in $\varphi : \mathbb{R}^N \times \mathbb{R}^+ \to \mathbb{R}$, is covered by all agents, which ensuring that agents are assigned an area, which is encoded in $\|\boldsymbol{\xi} - \mathbf{p}\|_2^2$, that is maximal with respect to Ξ.

The Voronoi partition $\mathcal{V} \equiv \mathcal{V}(\Xi, \mathbf{p})$ of Ξ given the current agent positions $\mathbf{p}$ is a set $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_N\}$. Here, each cell $\mathcal{V}_i$ is given by

$$\mathcal{V}_i = \left\{ \xi \mid \|\xi - p_i\|_2^2 \leq \left\|\xi - p_j\right\|_2^2 \; \forall\, i \neq j, \, \forall\, j = 1, 2, \ldots, n \right\}.$$

An example of a Voronoi partition is shown in Figure (insert figure reference here). Using the definition of a Voronoi partition, we can rewrite (1.1) as

$$\max_{\mathbf{p}\in\Xi^N} \mathcal{H}_\varphi(\mathbf{p}, t) := \max_{p\in\Xi^N} \sum_{i=1}^{N} \int_{\mathcal{V}_i} \|\boldsymbol{\xi} - \mathbf{p}_i\|_2^2 \, \varphi(\boldsymbol{\xi}, t) \, \mathrm{d}\boldsymbol{\xi}. \tag{1.2}$$

In other words, the problem reduces to finding the configuration of agents so that effective coverage is maintained, but each agent is also assigned an area (i.e. a cell) that it is responsible for. There are two advantages of using the Voronoi partitions. First, the min term inside the integral is removed by the construction of the Voronoi cells. Second, any algorithm that uses these partitions will be *distributed*, which means agents only need to use information from its Voronoi neighbors, defined by agents that share a cell boundary (i.e. agents i and j are neighbors if and only if $\partial\mathcal{V}_i \cap \partial\mathcal{V}_j \neq \emptyset$), so that communications between agents can be reduced.

The mass M_i and centroid C_i of the i-th cell is given by

$$M_i = \int_{\mathcal{V}_i} \varphi(\boldsymbol{\xi}, t) \, \mathrm{d}\boldsymbol{\xi}, \quad C_i = \frac{1}{M_i} \int_{\mathcal{V}_i} \boldsymbol{\xi} \, \varphi(\boldsymbol{\xi}, t) \, \mathrm{d}\boldsymbol{\xi}. \tag{1.3}$$

To solve the maximization problem, agents move towards the centroid of their Voronoi cell [**?**]. Intuitively, the agents move towards the area they need to cover since the φ acts as an attracting force that draws the center of mass towards the desired coverage area. Algorithm (insert algorithm reference here) describes one step of the coverage control algorithm.

1.2 Approach

For the purpose of this project, we consider $d = 3$, i.e. 3-D Euclidean space. Though orientation information about the robot may be present, so that $d \neq 3$, we only use the position information as this algorithm spits out waypoints for the agents to travel to. There are multiple assumptions and simplifications that are made so that the 3-D implementation is achieved.