Model Predictive Coverage Control ⋆

Andrea Carron * **Melanie N. Zeilinger** *

* *Institute for Dynamic Systems and Control, ETH Zurich, Switzerland*
(e-mail: [carrona][mzeilinger]@ethz.ch)

Abstract: Cooperative robotic problems often require coordination in space in order to complete a given task, important examples include search and rescue, operations in hazardous environments, and autonomous taxi deployment. Events can be quickly detected by partitioning the working environment and assigning one robot to each partition. However, a crucial factor that limits the effectiveness and usage of coverage algorithms is related to the ability of taking decisions in the presence of constraints. In this paper, we propose a coverage control algorithm that is capable of handling nonlinear dynamics, and state and input constraints. The proposed algorithm is based on a nonlinear tracking model predictive controller and is proven to converge to a centroidal Voronoi configuration. We also introduce a procedure to design the terminal ingredients of the model predictive controller. The effectiveness of the algorithm is then highlighted with a numerical simulation.

Keywords: coverage control, Voronoi partitioning, model predictive control.

1. INTRODUCTION

Collaborative robotics has the potential to improve the quality of our lives. Possible applications include: swarms of autonomous robots that quickly map and explore unknown and dangerous environments to perform search and rescue operations; a fleet of underwater vehicles that detect the source of an oil spill and mitigate the environmental damages fixing the leak and cleaning the surrounding area; self-driving taxis that coordinate themselves to quickly serve customers, minimize travel time and emissions. In all these examples, the robots can effectively detect events by partitioning the environment in which they are operating. Optimal partitions can be collaboratively and iteratively computed when the robots are deployed. However, while iteratively determining the partition, it is crucial to take into account that robots' actions are subject to limitations, e.g., actuation constraints or constraints induced by dynamics. Without taking into account these limitations, the group of agents can be led into dangerous scenarios, e.g., two agents can crash into each other if they overestimate their breaking capabilities.

In this work, we address the problem of optimally covering an area of interest while satisfying robot dynamics, and state and input constraints. Specifically, results from Voronoi partitioning and coverage control are combined with model predictive control (MPC) techniques, resulting in an MPC-based algorithm that is proven to converge to a centroidal Voronoi configuration. The presented algorithm is based on the nonlinear tracking MPC formulation introduced by Ferramosca et al. (2009), and a result on discrete-time Lloyd descent algorithms presented in Cortes et al. (2004).

Related Work: There has been a considerable effort in the analysis of coverage and constrained control problems separately. Classical approaches to coverage problems assume the system dynamics to be a single integrator. In Cortés and Bullo (2005) and Bullo et al. (2012), gradient-based solutions are proven to converge to centroidal Voronoi partitions. Non-uniform coverage over a line has been studied in Leonard and Olshevsky (2011); Davison et al. (2012, 2015), where the authors determine a policy that is proven to converge to an optimal partition. In Lee et al. (2015), the coverage control problem under known time-varying sensory functions is analyzed. In Schwager et al. (2009); Carron et al. (2015); Todescato et al. (2017) the problem of concurrently estimating the sensory function and covering the environment is studied. To the best of our knowledge, the only work that considers systems with dynamics is the one proposed by Cortes et al. (2004), however, the authors consider passive systems without state or input constraints. Conversely, in Mohseni et al. (2017), the authors develop a receding horizon controller that is proven to satisfy state and input constraints, but the robots have single integrator dynamics. Finally, in Patel et al. (2013), a coverage control problem is studied where the area of the partitions is constrained.

Model predictive control received a considerable attention in the last decades thanks to its capability of systematically handling state and input constraints. For a comprehensive review about model predictive control, we refer the reader to Mayne et al. (2000); Mayne (2014); Borrelli et al. (2017). While there is a variety of techniques, the approach most relevant to this paper is the work presented by Ferramosca et al. (2009). The authors present a tracking MPC formulation for nonlinear systems with recursive feasibility and convergence guarantees. This is crucial for coverage control problems where systems are nonlinear and constrained, and have to track a given reference.

⋆ This work was supported by the Swiss National Centre of Competence in Research NCCR Digital Fabrication, and by the ETH Career Seed Grant 19-18-2.

Contributions: The main contributions of this paper are twofold: the first is a coverage control algorithm capable of converging to centroidal Voronoi configuration while satisfying nonlinear system dynamics, and state and input constraints. The second contribution is to show that the results in Chen and Allgöwer (1998) can be utilized to design the nonlinear tracking MPC terminal ingredients.

The remainder of the paper is organized as follows. Section 2 recalls preliminaries about coverage control. In Sections 3 and 4, we formulate the coverage control problem, introduce the model predictive coverage control algorithm, and prove its convergence to a centroidal Voronoi configuration. Section 5 presents a procedure to compute the terminal ingredients for the proposed nonlinear MPC scheme. In Section 6, the numerical results are discussed, and Section 7 concludes the paper.

2. COVERAGE CONTROL

Let $\mathcal{Q}$ be a convex polytope in $\mathbb{R}^n$. We define a partition of $\mathcal{Q}$ as a collection of M polytopes $\mathcal{W} = \{\mathcal{W}_1, \dots, \mathcal{W}_M\}$ with disjoint interiors, the union of which is $\mathcal{Q}$.

Let $p = [p_1, \dots, p_M]$ be the location of M robots, each moving in $\mathcal{Q}$. The Voronoi partition of $\mathcal{Q}$ generated by p is given by $\mathcal{V}(p) = \{\mathcal{V}_1(p), \dots, \mathcal{V}_M(p)\}$, see e.g. Okabe et al. (1992), which is defined as

$$\mathcal{V}_i(p) = \{q \in \mathcal{Q} \mid \|q - p_i\| \leq \|q - p_j\|, \forall j \neq i\},$$

where $\|\cdot\|$ denotes the Euclidean norm. Given that the set $\mathcal{Q}$ is convex, it can be shown that the Voronoi regions are convex as well, Du et al. (1999).

Let $\Phi : \mathcal{Q} \to R_+$ be a probability density function that represents a measure of information or probability that some event takes place over $\mathcal{Q}$. Given a partition $\mathcal{W}$, we define for each region its centroid with respect to the probability density function Φ as

$$c_i(\mathcal{W}_i) = \left(\int_{\mathcal{W}_i} \Phi(q)dq\right)^{-1} \int_{\mathcal{W}_i} q\Phi(q)dq.$$

We denote with $\mathbf{c}(\mathcal{W}) = [c_1(\mathcal{W}_1), \dots, c_M(\mathcal{W}_M)]$ the vector containing the partition' centroids. A partition $\mathcal{W}$ is said to be a centroidal Voronoi partition of the pair $(\mathcal{Q}, \Phi)$ if $\mathcal{W} = \mathcal{V}(\mathbf{c}(\mathcal{W}))$, i.e., $\mathcal{W}$ coincides with the partition generated by $\mathbf{c}(\mathcal{W})$. To simplify the notation, we denote with $c_i(p)$ the centroid computed with respect to the Voronoi partition generated by p, i.e., $c_i(\mathcal{V}_i(p))$.

We consider the task of minimizing the locational optimization function, introduced in Cortes et al. (2004),

$$H(p, \mathcal{W}) = \sum_{i=1}^{M} \int_{\mathcal{W}_i} g(\|q - p_i\|)\Phi(q)dq, \tag{1}$$

where $p_i \in \mathcal{W}_i$, and $g(\cdot)$ denotes the sensing performance of the robot at point q from point p_i. Note that the function is to be minimized with respect to the robot locations p and the partitions $\mathcal{W}$. In Du et al. (1999), it is shown that for a fixed probability density function, the set of local minima of $H(p, \mathcal{W})$ coincides with the centroids of a Voronoi partition of the pair $(\mathcal{Q}, \Phi)$.

Let $p(0)$ be the robots' initial location, and assuming that the robots have a single integrator dynamics, without state and input constraints, the update

$$p(k+1) = c_i(p(k))$$

asymptotically converges to the set of centroidal Voronoi partitions, which are extreme points of the coverage function (1), Cortes et al. (2004). This algorithm is also known as Loyd algorithm, Lloyd (1982).

A generalization of the Lloyd algorithm is given by the following proposition.

Proposition 1. (Cortes et al. (2004)). Let $T : \mathcal{Q}^M \to \mathcal{Q}^M$ be a continuous mapping satisfying the properties

(1) for all $i \in \{1, \dots, M\}$, $\|T_i(p) - c_i(p)\| \leq \|p_i - c_i(p)\|$, where T_i denotes the ith component of T;
(2) if p is not centroidal, then there exists a j such that $\|T_j(p) - c_j(p)\| < \|p_j - c_j(p)\|$.

Let $p(0) \in \mathcal{Q}^M$ denote the initial robots' location. Then, the sequence $\{T^m(p(0)) | m \in \mathbb{N}\}$ converges to a centroidal Voronoi configuration.

3. PROBLEM FORMULATION

We consider a group of M robots, which move in the region $\mathcal{Q}$. The goal of the robots is to optimally cover the region $\mathcal{Q}$ with respect to the cost (1) under a fixed and known probability density function Φ.

Robots move according to the following discrete-time nonlinear dynamics

$$\begin{aligned} x_i(k+1) &= f_i(x_i(k), u_i(k)) \\ y_i(k) &= h_i(x_i(k), u_i(k)) \\ p_i(k) &= C_i x_i(k), \end{aligned} \tag{2}$$

where, for all $i \in \{1, \dots, M\}$, $x_i(k) \in \mathbb{R}^{n_i}$, $u_i(k) \in \mathbb{R}^{m_i}$ and $y_i(k) \in \mathbb{R}^{p_i}$ are state, input, and output vectors of the ith robot, respectively, f_i represents the state dynamics, h_i the output function, $C_i \in \mathbb{R}^{n \times p_i}$ is a matrix that selects the robot position among the state variables. The dynamics models $f_i(x_i, u_i)$ and the output models $h_i(x_i, u_i)$ are assumed to be known and Lipschitz continuous.

All robots are subject to polytopic state and input constraints containing the origin in their interior, i.e.,

$$\begin{aligned} \mathbb{X}_i &= \{x_i | A_x x_i \leq b_x\} \\ \mathbb{U}_i &= \{u_i | A_u u_i \leq b_u\}, \end{aligned} \tag{3}$$

where $A_x \in \mathbb{R}^{n_{x_i} \times n_i}, b_x \in \mathbb{R}^{n_{x_i}}, A_u \in \mathbb{R}^{n_{u_i} \times n_u}$ and $b_u \in \mathbb{R}^{n_{u_i}}$ for all $i \in \{1, \dots, M\}$. Without loss of generality, we assume that the first element of the state x_i is the position p_i of the i-th robot and that the operator $\psi(p_i) = \left[p_i^T \; 0 \cdots 0\right] \in \mathbb{R}^{n_{x_i}}$ generates a vector of the dimension of the state of the ith agent, where the first element is the position p_i of the ith robot, and the others are equal to zero.

Assumption 1. The robots' dynamics (2) are position invariant, i.e., $f(x(k) + \psi(p), u(k)) = x(k+1) + \psi(o)$ $\forall p \in \mathbb{R}^n, x(k), u(k)$. Moreover, we assume $\mathcal{Q} \subseteq C_i\mathbb{X}_i \; \forall i \in \{1, \dots, M\}$.

Assumption 1 restricts the results to agents with dynamics where the position appears only in an integrator. This condition is, however, satisfied by the vast majority of mobile ground, aerial, and underwater robots.

This paper presents a MPC-based coverage controller for robots with nonlinear dynamics, and polytopic state and input constraints. The controller is guaranteed to converge to a centroidal Voronoi partition, which is an extreme point of the coverage cost function (1).

4. MODEL PREDICTIVE COVERAGE CONTROL

In this section, we introduce the model predictive coverage controller. An MPC is used to track the centroids, which are iteratively updated to converge to an optimal Voronoi partition. To guarantee convergence, the tracked reference centroid is updated if and only if all robots have decreased their distance from their centroids. The controller is described in Algorithm 1, and defines a map which satisfies the assumptions of Proposition 1.

Algorithm 1 works as follows. During the initialization phase (line 1), the robots, based on their initial position $p(0)$, compute their Voronoi partitions, centroids, and the distance between their location $p(0)$ and the computed centroids. The centroid is set as reference of the tracking-MPC problem (4), i.e.,

$$r_i = c_i(p(0)),$$

and the distance between their location and the computed centroid is set as tracking-error, i.e.,

$$e_{i,r} = \|p_i(0) - c_i(p(0))\|.$$

Then, at every iteration, each agent first measures its own state (line 4), then solves the nonlinear tracking-MPC problem (4) (line 5), and finally applies the first input of the optimal input sequence (line 6). The last step of the algorithm (lines 8-10) checks if the reference of the MPC controller can be updated. In particular, two condititions must to be satisfied. First, all agents are required to not increase their tracking error, i.e.,

$$\|p_i(k) - r_i\| \leq e_{i,r},$$

and second, at least one agent $j \in \{1, \dots, M\}$ is required to decrease its tracking error, i.e.,

$$\|p_i(k) - r_j\| < e_{j,r}.$$

If these two conditions are fulfilled, then all agents update their reference $r_i = c_i(p(k))$ and their tracking error $e_{i,r}$. Note that the only step that requires a centralized coordination is the MPC reference update detection (line 8), all other steps can be performed using only local information.

The nonlinear tracking MPC controller defined in (4) is inspired by the work in Ferramosca et al. (2009). This MPC formulation is suitable for the coverage control problem because it guarantees recursive feasibility independently of how the reference changes. Moreover, if the reference is constant, the controller converges to the closest feasible reference. The MPC controller is defined by the following optimization problem

$$\min_{u,\bar{x},\bar{u},\bar{r}} J_i(x_i, u_i, \bar{x}_i, \bar{u}_i, r_i, \bar{r}_i) \tag{4a}$$
$$\text{s.t. } l = \{1, \dots, N-1\} \tag{4b}$$
$$x_{i,0} = x_i(k) \tag{4c}$$
$$x_{i,l+1} = f_i(x_{i,l}, u_{i,l}) \tag{4d}$$
$$(x_{i,l}, u_{i,l}) \in (\mathbb{X}_i, \mathbb{U}_i) \tag{4e}$$
$$(x_{i,N}, \bar{r}_i) \in \mathcal{X}_{\text{terminal}} \tag{4f}$$
$$\bar{r}_i = h_i(\bar{x}_i, \bar{u}_i) \tag{4g}$$
$$\bar{x}_i = f_i(\bar{x}_i, \bar{u}_i). \tag{4h}$$

Given a reference r_i, which in our case is the current centroid, the optimization problem computes the system's

Algorithm 1 Model Predictive Coverage Control

1: **Initialize:** compute $r_i = c_i(p(0))$ and
$e_{i,r} = \|p_i(0) - c_i(p(0))\| \ \forall i = \{1, \dots, M\}$
2: **for** k=1,2,... **do**
3: **for** every agent i **do**
4: Measure state x_i
5: Solve optimization problem (4)
6: Apply $u^\star_{i,0}$
7: **end for**
8: **if** $\|h_i(x_i(k), u_i(k)) - r_i\| \leq e_{i,r} \ \forall i \in \{1, \dots, M\}$ and $\exists j \in \{1, \dots, M\}$ s.t. $\|h_j(x_j(k), u_j(k)) - r_j\| < e_{j,r}$ **then**
9: Update $r_i = c_i(p(k))$ and $e_{i,r} = \|p_i(k) - c_i(p(k))\|$
10: **end if**
11: **end for**

equilibrium point (4g)-(4h), and the optimal input sequence. The cost function in problem (4) is defined as

$$\begin{aligned} J_i(x_i, u_i, \bar{x}_i, \bar{u}_i, r_i, \bar{r}_i) = \sum_{l=1}^{N-1} &\ell_{i,l}(x_{i,l} - \bar{x}_i, u_{i,l} - \bar{u}_i) + \\ &\ell_{i,N}(x_{i,l} - \bar{x}_i) + \ell_{i,t}(\bar{r}_i - r_i), \end{aligned} \tag{5}$$

where $\ell_{i,l}$, $\ell_{i,N}$, and $\ell_{i,t}$ are the stage cost, the terminal cost, and the target cost, respectively.

The MPC controller is locally computed by each agent and is used in a receding horizon fashion, which means that the control law is given by $\kappa_i(x_i, r_i) = u^\star_{i,0} \ \forall i \in \{1, \dots, M\}$, where $u^\star$ is the optimal solution of (4). Note that the feasible region of problem (4) does not depend on the reference r_i, because it just appears in the cost function $J_i(x_i, u_i, \bar{x}_i, \bar{u}_i, r_i, \bar{r}_i)$.

Consider the following assumptions on the controller parameters, which are commonly used in nonlinear tracking MPC to guarantee recursive feasibility and convergence, see e.g. Limon et al. (2009); Ferramosca et al. (2009). Note that the assumptions on the terminal ingredients are similar to those in standard MPC but extended to a set of equilibrium points. While it can generally be difficult to satisfy these assumptions, in the considered problem satisfying Assumption 1 on position-invariance, which makes any position a steady-state, the conditions are easily satisfied. See also Section 5 for a design procedure.

Assumption 2. Consider the optimization problem (4):

(1) Let $\left[\bar{x}_i^T, \ \bar{u}_i^T\right]^T = g_i(r_i)$ be a function that defines the steady state given the reference. Assume g_i to be Lipschitz continuous $\forall i \in \{1, \dots, M\}$.
(2) Let $\mathcal{R}$ be the set of admissible targets, i.e, $\mathcal{R} = \{(x, u) | r_i = Cx_i, \ x_i = f_i(x_i, u_i), (x_i, u_i) \in (\mathbb{X}_i, \mathbb{U}_i), \forall i \in \{1, \dots, M\}\}$. Assume the set $\mathcal{R}$ to be convex.
(3) Let $\kappa_i(x_i, r_i)$ be a continuous control law such that for all $r_i \in \mathcal{R}$, the steady- state $(\bar{x}_i, \bar{u}_i)$ is asymptotically stable for the system system $f_i(x_i(k), \kappa_i(x_i(k), r_i))$.
(4) Let $\mathcal{X}_{\text{terminal}}$ be an invariant set for system $f_i(x_i(k), \kappa_i(x_i(k), r_i))$ under state and input constraints (3).
(5) Let $\ell_{i,N}(x, \bar{x})$ be Lyapunov functions for their associated systems $x_{i,l+1} = f_i(x_{i,l}, \kappa_i(x_i(k), r_i))$.
(6) Let $\ell_{i,l}(x, u, \bar{x}, \bar{u})$ be positive definite functions.

(7) Let $\ell_{i,t}(r, \bar{r})$ be convex, positive definite and subdifferentiable functions.

In the following, we show that Algorithm 1 converges to a centroidal Voronoi configuration by satisfying the assumption of Proposition 1.

Proposition 3. Let Assumptions 1 and 2 be satisfied. Let $p_i(0) = C_i x_i(0) \in Q \; \forall i \in \{1, \ldots, M\}$ be the initial location of the robots subject to dynamics (2) and constraints (3), and $x_i(0) \; \forall i \in \{1, \ldots, M\}$ be a feasible initial condition for problem (4). Then problem (4) is recursively feasible and Algorithm 1 converges to a centroidal Voronoi configuration.

Proof. Recursive feasibility and convergence to the closest feasible reference follow directly from Assumption 2 and the results in Ferramosca et al. (2009). Due to Assumption 1, any centroid is a feasible reference. Thus the MPC scheme converges to the given references, and there exists a subsequence of the iterations of Algorithm 1 that satisfies Proposition 1. $\square$

5. MPCC DESIGN

In the previous section, an algorithm that solves the coverage control problem under nonlinear state dynamics and polytopic state and input constraints has been introduced. The proposed solution is based on an MPC formulation that requires the design of the terminal ingredients, i.e., a terminal control law, a terminal invariant set and a terminal cost function. In the following, we present a procedure to design these ingredients. Compared to the work of Ferramosca et al. (2009), we neither propose an equality terminal constraint nor a terminal set for an LTV representation of the nonlinear system, but a method to directly compute the terminal ingredients for the nonlinear system.

5.1 Terminal control law

The terminal control law is computed for the linearized system. The equilibrium $(\bar{x}, \bar{u})$ under which the linearization is performed is chosen as follows. Thanks to assumption 1, the position is not relevant for the selection of the equilibrium, and hence, without loss of generality, is set to zero. All remaining states are problem dependent, and can be chosen according to a given metric, i.e., to minimize a cost function. The linearization gives the following state and input matrices

$$A_i = \left.\frac{\partial f_i(x_i, u_i)}{\partial x_i}\right|_{(\bar{x}_i, \bar{u}_i)}, \quad B_i = \left.\frac{\partial f_i(x_i, u_i)}{\partial u_i}\right|_{(\bar{x}_i, \bar{u}_i)}.$$

The terminal control law $\kappa_i(x_i, \bar{x}_i) = K_i(x_i - \bar{x}_i)$ is chosen as an LQR controller for the linearized system, which is known to asymptotically stabilize the nonlinear system in a neighborhood of the equilibrium $(\bar{x}_i, \bar{u}_i)$, see e.g. Kirk (1970).

5.2 Terminal invariant set and cost

The terminal invariant set and cost computation are based on the work of Chen and Allgöwer (1998), but in discrete-time domain. The following lemma provides a region of attraction and a performance bound for the nonlinear system controlled by a local linear state feedback. We will later use these results to outline a design procedure for the terminal set and cost.

Lemma 4. Assume the closed-loop system $x(k+1) = f(x(k), Kx(k))$ to be stabilizable at the origin, and let $A_K = A + BK$ denote the linearized closed-loop system under the stabilizing controller $u(k) = Kx(k)$. Then,

(1) the following Lyapunov equation

$$\left(\frac{1}{\sqrt{1-c}} A_K\right)^T P \left(\frac{1}{\sqrt{1-c}} A_K\right) - P = -Q^\star \quad (6)$$

admits a unique positive-definite and symmetric solution P, where $Q^\star = Q + K^T R K$ is positive definite and symmetric, and c satisfies

$$c < 1 - |\lambda_{\max}(A_K)|^2.$$

(2) There exists a constant $\alpha \in (0, +\infty)$ specifying a neighborhood

$$\mathcal{X}_{\text{terminal}}(\alpha) = \{x \in \mathbb{R}^n | x^T P x \le \alpha\} \subseteq \mathbb{X}$$

such that
  (a) the input $Kx \in \mathbb{U}$ for all $x \in \mathcal{X}_{\text{terminal}}(\alpha)$, i.e., the linear feedback controller respects the input constraints in $\mathcal{X}_{\text{terminal}}(\alpha)$,
  (b) $\mathcal{X}_{\text{terminal}}(\alpha)$ is invariant for the nonlinear system (2) controlled by the linear feedback $u = Kx$,
  (c) for any $\bar{x} \in \mathcal{X}_{\text{terminal}}(\alpha)$, the infinite horizon cost

$$J^\infty(\bar{x}, u) = \sum_{k=\bar{k}}^{+\infty} x(k)^T Q x(k) + u(k)^T R u(k)$$

  subject to nonlinear dynamics (2), starting from $x(\bar{k}) = \bar{x}$ and controller by the local linear state feedback $u = Kx$, is bounded from above as follows

$$J^\infty(\bar{x}, u) \le \bar{x}^T P \bar{x}.$$

From Lemma 4, we can derive a procedure to compute the terminal set and cost, which is described in Algorithm 2. The procedure is a direct extension of the approach in Chen and Allgöwer (1998) for continuous-time systems, to which we refer for more details.

Algorithm 2 Terminal Ingredient Computation

1: Compute the LQR controller for the linearization of (2)
2: Choose a constant $c < 1 - |\lambda_{\max}(A_K)|^2$ and solve (6)
3: Find the largest α_1 such that $Kx \in \mathbb{U}$ for all $x \in \mathcal{X}_{\text{terminal}}(\alpha_1)$
4: Find the largest $\alpha \in (0, \alpha_1]$, c_1 and c_2, with $c = c_1 - 2c_2$, such that $x^T A^T P \Phi(x) \le c_1 \|x\|_P^2$ and $x^T A^T P \Phi(x) \le c_2 \|x\|_P^2$ are satisfied in $\mathcal{X}_{\text{terminal}}$, where $\Phi(x) = A_K x - f(x, Kx)$

6. SIMULATIONS

In this section, we show the effectiveness of the model predictive coverage control algorithm in simulation. The simulation is performed in Python on an Intel Core i7 3.3GHz machine with 16 GB of RAM, using CasADI and IPOPT as solver.

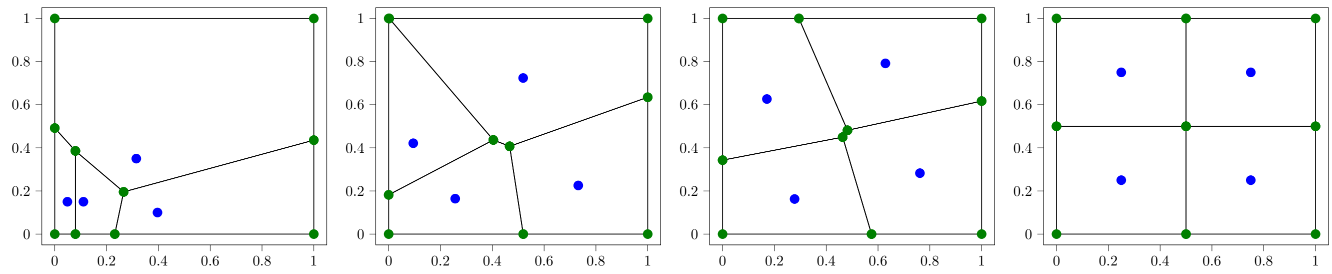

Fig. 1. Robot positions (blue dots) and Voronoi partitions (green dots and black lines) at iterations $k = 1, 10, 20, 100$.

The environment to cover is defined as $\mathcal{Q} = [0, 1] \times [0, 1]$, with density function $\Phi(q) = 1$. We consider a fleet of 4 robots, each modeled as a bicycle model. Each dynamics is represented by the following non-linear system

$$\begin{aligned} p_{x_i}(k+1) &= p_{x_i}(k) + T_s \cos(\theta) v_i(k) \\ p_{y_i}(k+1) &= p_{y_i}(k) + T_s \sin(\theta) v_i(k) \\ \theta_i(k+1) &= \theta(k) + T_s \tan(\gamma_i(k)) \frac{v_i(k)}{L} \\ \gamma_i(k+1) &= \gamma_i(k) + T_s \delta(k), \end{aligned} \tag{7}$$

where p_{x_i} and p_{y_i} represent the position, θ_i the yaw, and γ_i the steering angle of the ith robot. The two inputs are the velocity v_i and the steering rate δ_i. The sampling time T_s is 0.1s, and the wheelbase of the vehicle L is 0.005. We assume the output function to be equal to the state, i.e., $h_i(x_i, u_i) = x_i$, and the output matrix

$$C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The position of each robot is constrained to be within $\mathcal{Q}$, the steering angle $\gamma_i \in [-\frac{\pi}{2.1}, \frac{\pi}{2.1}]$, and the yaw $\theta_i \in [-\pi, \pi]$. The velocity and the steering rate are limited to $|v_i| \leq 1$ and $|\delta_i| \leq 1$. The robot initial positions are $p_1 = [0.05, 0.15]$, $p_2 = [0.1, 0.15]$, $p_3 = [0.25, 0.35]$, and $p_4 = [0.3, 0.1]$. Figure 1 shows the robot positions at iterations $k = 1, 10, 20, 100$. It can be seen from the pictures that the algorithm converges to 4 rectangles of equal area. In Figure 2, the coverage cost function (1) is plotted for the first 100 iterations.

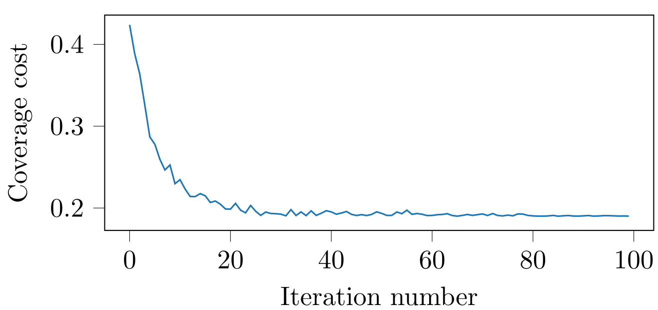

Fig. 2. Evolution of the coverage cost function (1) over 100 iterations.

Note that the coverage cost is not monotonic, as it only decreases when the centroids are updated but can increase during the MPC iterations. Finally, Figure 3 shows the trajectories of the robots.

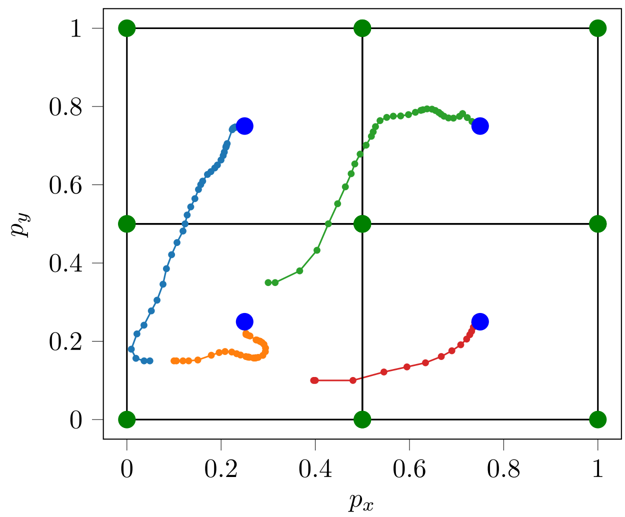

Fig. 3. Trajectories of the 4 agents (colored dotted lines), Voronoi partition (green dots and black lines), and centroids (blue dots).

7. CONCLUSIONS

In this paper, we propose a coverage control algorithm that is capable of handling nonlinear state dynamics, and state and input constraints. The proposed controller is based on a nonlinear tracking MPC, and it is proven to converge to a centroidal Voronoi configuration. Numerical results show the effectiveness of the algorithm.

REFERENCES

Borrelli, F., Bemporad, A., and Morari, M. (2017). *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, New York, NY, USA, 1st edition.

Bullo, F., Carli, R., and Frasca, P. (2012). Gossip coverage control for robotic networks: Dynamical systems on the space of partitions. *SIAM Journal on Control and Optimization*, 50(1), 419–447.

Carron, A., Todescato, M., Carli, R., Schenato, L., and Pillonetto, G. (2015). Multi-agents adaptive estimation and coverage control using gaussian regression. *European Control Conference (ECC'15)*.

Chen, H. and Allgöwer, F. (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10), 1205 – 1217. doi: https://doi.org/10.1016/S0005-1098(98)00073-9.

Cortés, J. and Bullo, F. (2005). Coordination and geometric optimization via distributed dynamical systems. *SIAM Journal on Control and Optimization*, 44(5), 1543–1574.

Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20(2), 243–255.

Davison, P., Leonard, N., Olshevsky, A., and Schwemmer, M. (2015). Nonuniform line coverage from noisy scalar measurements. *Automatic Control, IEEE Transactions on*, 60(7), 1975–1980.

Davison, P., Schwemmer, M., and Leonard, N. (2012). Distributed nonuniform coverage with limited scalar measurements. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, 1455–1460. IEEE.

Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4), 637–676. doi: 10.1137/S0036144599352836.

Ferramosca, A., Limon, D., Alvarado, I., Alamo, T., and Camacho, E.F. (2009). MPC for tracking of constrained nonlinear systems. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 7978–7983. doi:10.1109/CDC.2009.5400618.

Kirk, D. (1970). *Optimal Control Theory: An Introduction.* Englewood Cliffs: Prentice-Hall.

Lee, S., Diaz-Mercado, Y., and Egerstedt, M. (2015). Multirobot control using time-varying density functions. *Robotics, IEEE Transactions on*, 31(2), 489–493.

Leonard, N. and Olshevsky, A. (2011). Nonuniform coverage control on the line. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 753–758. IEEE.

Limon, D., Ferramosca, A., Alvarado, I., Alamo, T., and Camacho, E.F. (2009). *MPC for Tracking of Constrained Nonlinear Systems*, 315–323. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lloyd, S. (1982). Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2), 129–137.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967 – 2986. doi: https://doi.org/10.1016/j.automatica.2014.10.128.

Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789 – 814. doi: https://doi.org/10.1016/S0005-1098(99)00214-9.

Mohseni, F., Doustmohammadi, A., and Menhaj, M.B. (2017). Distributed model predictive coverage control for decoupled mobile robots. *Robotica*, 35(4), 922941. doi:10.1017/S0263574715000892.

Okabe, A., Boots, B., and Sugihara, K. (1992). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* John Wiley & Sons, Inc., New York, NY, USA.

Patel, R., Frasca, P., and Bullo, F. (2013). Centroidal Area-Constrained Partitioning for Robotic Networks. Dynamic Systems and Control Conference.

Schwager, M., Rus, D., and Slotine, J.J. (2009). Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3), 357–375.

Todescato, M., Carron, A., Carli, R., Pillonetto, G., and Schenato, L. (2017). Multi-robots gaussian estimation and coverage control: From client–server to peer-to-peer architectures. *Automatica*, 80, 284–294.