

Predicting Environmental Boundary Behaviors with a Mobile Robot

David Saldaña¹, Renato Assunção¹, and Mario F. M. Campos¹

Abstract—Predicting the behavior of dangerous environmental boundaries, like spreading fire or oil spill, provides relevant information to mitigate the problem or even to support evacuation actions in order to save human or animal lives. In this paper, we present a model that uses a single robot moving around an environmental boundary in order to predict its shape by an analytical continuous function, which is based on the combination of polynomial approximation and Fourier Series. We show that the method converges to the exact boundary when we increase the sample frequency and the robot velocity. In order to evaluate the estimation quality, we performed experiments with simulated and actual robots. We applied our model in some dynamic boundaries presented in the literature, as in the application of plume-front estimation, showing that it accomplish accurate results.

Index Terms—Environment-aware Automation, Surveillance Systems, Probability and Statistical Methods

I. INTRODUCTION

ESTIMATING the shape of environmental boundaries has been a highlighted topic in robotics since the last decade. There is a variety of applications where robotic sensors have a relevant impact, specially when monitoring phenomena such as oil spills [1], [2], [3], forest fires [4], chemical leaks [5], and harmful algae blooms [6]. In these environments, where anomalies may grow and progressively move, being able to possess the ability to forecast the affected areas is a very desirable feature of an autonomous monitoring system.

In the robotics literature, the concept of estimating boundaries is directly related to estimating environmental sets, level curves, and perimeters. Preliminary works focused only on the identification of shapes with static boundaries [7], [1]. In [8] the authors propose a method to track dynamic level curves by means of the gradient information required to minimize the square error between the robot location and the level curve. In [9], approximation theory of convex bodies is used to estimate a given boundary with a polygon with a fixed number of vertices. This approach converges by increasing the interpolation points, but it requires many robots to check and

Manuscript received: August 29, 2015; Revised November 17, 2015; Accepted January 12, 2016.

This paper was recommended for publication by Editor Antonio Bicchi upon evaluation of the Associate Editor and Reviewers' comments. *The authors gratefully acknowledge the support of CAPES, CNPq and FAPEMIG. David Saldaña also thanks to COLCIENCIAS for its support.

*The authors appreciate the help of Mickey Whitzer, Philip Dames, and Vijay Kumar, for their unvaluable advice and support during the experiments. Part of this work was carried out at the GRASP laboratory, University of Pennsylvania.

¹D. Saldaña, R. Assunção and M.F.M. Campos are with the Computer Vision and Robotics Laboratory (VeRLab), Computer Science Department, Universidade Federal de Minas Gerais, MG, Brazil. E-mails: {saldana, assuncao, mario}@dcc.ufmg.br

Digital Object Identifier (DOI): see top of this page.

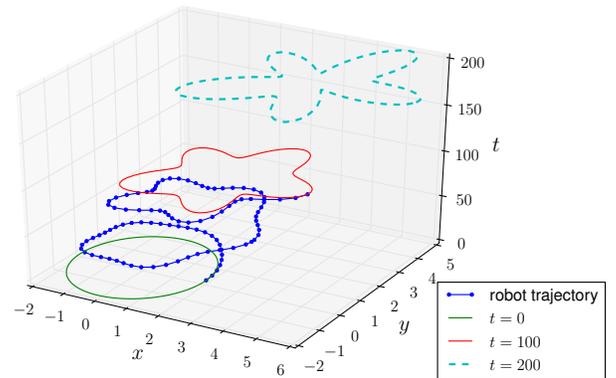


Figure 1: Dynamic boundary and the robot trajectory. The desired shape to be predicted is represented by the dashed line.

update the vertices. Some methods such as [10], [11], [12], and [13] have the advantage of integrating the detection, tracking and estimation processes, but the boundary is projected by interpolating the robot localization and the sampled points. It also requires a large number of robots, because the boundary evolves making the older sampled points less relevant to the estimation of the current shape. In [2] the authors present an efficient estimation method for liquid pollutants in marine environments. However, this method is specific for boundaries with dynamics satisfying the advection-diffusion behavior. The work of Duttagupta et al. [14] introduces the concept of prediction for non-closed curves by using robots with range sensors. However, this kind of method cannot be directly applied to any of the aforementioned contexts.

In a previous work [13], we proposed a hybrid approach to detect and to track multiple boundaries. In this paper, we focus on the problem of predicting the boundary behavior using a single robot. As an illustration of the problem, Figure 1 shows the evolution of a dynamic boundary at times $t = 0, 100, 200$. This boundary has simultaneous transformations like translation, shrinking and expansion. Starting at time $t = 0$ as a circle, it is deformed successively to assume a flower like shape at $t = 200$. The robot moves along the boundary from time $t = 0$ to $t = 100$ sampling its location. At each time t , the robot can track only one point in the curve. The sampled points are represented by small dots connected by the smooth spiral-shaped curve representing the robot's trajectory up to $t = 100$. The objective is to use the sampled collected positions to learn the entire boundary at all time points, including its forecasting

in future moments. For instance, at time $t = 100$, we want to estimate its present entire red-line shape as well as to predict that, at time $t = 200$, it will become the star-shaped dashed curve.

Our contributions are two fold. (1) Most of the related approaches relies on linear interpolation of the robot's positions in order to approximate the boundary by a polygon. Without knowledge of the boundary evolution dynamics, we approached the problem differently. We modeled the evolving boundary by a position-specific parametric analytical continuous function, where the parameters can be estimated using the sampled punctual locations. (2) Our proposal not only estimates the current closed-curve state, but it can also predict the future boundary behavior without previous knowledge of the phenomena.

II. PROBLEM STATEMENT

We are interested in a region within the environment where there is a phenomenon delimited by a perimeter. This region of interest $\Omega_t \subset \mathbb{R}^2$ is a connected set with finite area, indexed by time $t \in \mathbb{R}$, and enclosed by a boundary defined as

Definition 1 (dynamic boundary): A **dynamic boundary** is a set of planar points $\partial\Omega_t$ such that for each point $p \in \partial\Omega_t$, and for any arbitrarily specified $\xi > 0$, the open disc centered at p with radius ξ contains points of Ω_t and its complement Ω_t' .

We assume that the boundary $\partial\Omega_t$ is a simple closed curve (also called Jordan curve), parameterized as

$$\partial\Omega_t = \{\gamma(t, s) \mid s \in [0, 1]\},$$

where the image of the function $\gamma : \mathbb{R}_{\geq t_0} \times [0, 1] \rightarrow \mathbb{R}^2$ is a curve in \mathbb{R}^2 , mapped by the parameter $s \in [0, 1]$, such that $\gamma(t, 0) = \gamma(t, 1)$ and the restriction that $\gamma(t, s)$ is an injective function of $s \in [0, 1)$ for a fixed time t . Hence, $\partial\Omega_t$ is a continuous loop with no self-intersecting points. Additionally, we assume that the function γ changes smoothly in time t .

To sample the dynamic boundary $\partial\Omega_t$, we use a robot whose configuration is represented by the vector $\mathbf{x} = [x, y, \theta]^T$, where x, y define the coordinates in the Euclidean space and θ is the robot's orientation.

At each time step t , the robot can observe its location (x, y) , and estimate the curve parameter s (more details about the s -estimation in Section III). Therefore, the robot data, in a finite time interval, is the set of k samples

$$\mathcal{D} = \{(t_i, x_i, y_i, s_i) \mid i = 1, \dots, k\}.$$

We wish to predict the shape of the boundary based on the data \mathcal{D} :

Problem 1: Given a set of samples $\mathcal{D} = \{(t_i, x_i, y_i, s_i) \mid i = 1, \dots, k\}$ by a robot moving along a dynamic boundary in counterclockwise manner, how to estimate the past boundary state $\partial\Omega_t$ for $t_0 \leq t \leq t_k$ and to predict the future $\partial\Omega_t$ for $t > t_k$.

We assume that the angular speed of the robot $\dot{\theta}$ is large enough to allow the robot to circulate the boundary, $\partial\Omega_t$, while moving with linear velocity v .

To solve this problem, we first propose in Section III an algorithm to estimate the arc-length parameter s and then, in Section IV, we propose a model to solve Problem 1.

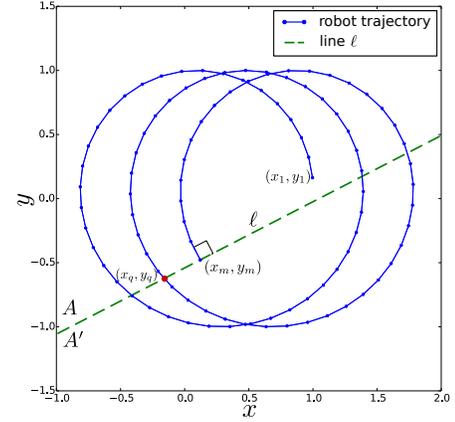


Figure 2: A trajectory for a robot with constant velocity $v = 2.6$ along the boundary $\gamma(t, s) = [0.1 t \cos(s), \sin(s)]^T$. The dots represent the sampled points.

III. ESTIMATING THE CURVE PARAMETER

During its trajectory, the robot takes m observations of its location (x, y) and sampling time t , producing the set $\mathcal{S} = \{(x_i, y_i, t_i) \mid i = 1, 2, \dots, m\}$. Figure 2 illustrates a trajectory for a robot moving with constant velocity along the boundary function $\gamma(t, s) = [0.1 t \cos(\pi s), \sin(\pi s)]^T$. The arc-length parameter s must be estimated for each sampled time in order to complete the robot data \mathcal{D} , which is used to analyze the behavior of the anomaly boundary $\partial\Omega_t$ (as described in Section IV).

The work described in [9] presents a technique to estimate the s parameter for slowly-varying boundaries with multiple robots and assuming that, at the initial time $t = 0$, the robots have an estimate of the boundary. In our approach, we avoid this assumption and use only the sampled points in the robot's trajectory.

A. Identifying cycles

As a requirement for estimating the s parameter, we need to identify the completed cycles in the robot trajectory. We assume that the boundary $\partial\Omega_t$ changes smoothly. If the robot leaves a mark in the boundary at its departure point, it can move along the boundary fast enough to reach the marked point in the evolving boundary at least once. Each of these laps will be called a *cycle*. The following algorithm identifies the completed cycles in the robot sampling data

- 1) Initially, we trace a line ℓ , which is perpendicular to the last line segment $\{(x_{m-1}, y_{m-1}), (x_m, y_m)\}$ of \mathcal{S} , and pass over the last point (x_m, y_m) (as exemplified in Figure 2 by the dashed line).
- 2) The line ℓ divides the space \mathbb{R}^2 into two sets, the set $A = \{p \in \mathbb{R}^2 \mid p \text{ is above } \ell\}$ and its complement A' .
- 3) We iterate along the samples \mathcal{S} backwards to find the first line segment $\{(x_{k-1}, y_{k-1}), (x_k, y_k)\}$ that satisfies one of following two conditions:

$$(x_{m-1}, y_{m-1}) \in A' \Rightarrow (x_k, y_k) \in A \wedge (x_{k-1}, y_{k-1}) \in A',$$

or

$$(x_{m-1}, y_{m-1}) \in A \Rightarrow (x_k, y_k) \in A' \wedge \\ ((x_{k-1}, y_{k-1}) \in A \vee (x_{k-1}, y_{k-1}) \in \ell).$$

- 4) As the segment $\{(x_{k-1}, y_{k-1}), (x_k, y_k)\}$ crosses from A to A' , from A' to A or from A' to ℓ , then there must be an intersecting point $(x_q, y_q) \in \ell$, which we use to represent the final point in the cycle.
- 5) We create a cycle $o_1 = \{(x_q, y_q, \hat{t}_q)\} \cup \{(x_i, y_i, t_i) | i = k, k+1, \dots, m\}$, where \hat{t}_q is interpolated.
- 6) We iteratively repeat the first five steps with the subset $S' = \{(x_i, y_i, t_i) | i = 1, 2, \dots, k\}$ until no more cycles are identified. In this way, we obtain the set of cycles \mathcal{O} .

In this algorithm, we discard the oldest points because they cannot complete a cycle. An important underline is that this algorithm works for boundaries changing slowly with respect to the robot's velocity. If the shape deforms too fast, it is possible that no cycle will be identified.

B. Estimating s

We use the set of cycles \mathcal{O} to estimate the curvature parameter s for each sampled point. For each cycle $o \in \mathcal{O}$ with an arc-length L and n points, we estimate the curve parameter as

$$\hat{s}_i = \frac{\text{arc-length}(\{(x_j, y_j, t_j) | j = 1, 2, \dots, i\})}{L}, \quad \forall i = 1, \dots, n.$$

Then, we can extend the cycle information by including the estimated \hat{s} , as $\hat{o} = \{(x_j, y_j, t_j, \hat{s}_i) | j = 1, 2, \dots, n\}$. The trajectory data \mathcal{D} is formed by the union of all cycles information (removing the repeated elements when $s = 0$ and $s = 1$). Hence, the complete dataset \mathcal{D} with k samples is given by $\mathcal{D} = \{(t_i, x_i, y_i, \hat{s}_i) | i = 1, \dots, k\}$.

IV. PREDICTING THE BOUNDARY BEHAVIOR

In this section, we propose an approach to solve Problem 1. We need to estimate the boundary $\partial\Omega_t$ for any $t > t_0$, using the robot's data \mathcal{D} .

A. Estimating the trajectory of point in the boundary

We analyze the trajectory of a single arbitrary point of the boundary during the time interval $t = [t_0, t_f]$. If we fix the curve parameter to an arbitrary value $s_0 \in [0, 1]$, the trajectory of this point can be represented by $\gamma(t, s_0)$, for $t \in [t_0, t_f]$. Figure 3 illustrates a trajectory $\gamma(t, s_0)$ in a time interval $t \in [t_0, t_f]$. The form of the trajectory depends on the dynamics of the boundary function. As we assume that it changes smoothly, we can approximate γ by a n -degree polynomial $\hat{\gamma}$:

$$\hat{\gamma}(t, s_0) = \begin{bmatrix} \beta_{0,0} & \beta_{0,1} & \dots & \beta_{0,n} \\ \beta_{1,0} & \beta_{1,1} & \dots & \beta_{1,n} \end{bmatrix} \begin{bmatrix} 1 \\ t \\ \vdots \\ t^n \end{bmatrix}.$$

Letting $\beta(s_0)$ be the $2 \times (n+1)$ constant matrix for s_0 , and $F = [t^0, t^1, \dots, t^n]^T$ the exponents of the variable t , we may represent the trajectory of the boundary point as

$$\hat{\gamma}(t, s_0) = \beta(s_0) F. \quad (1)$$

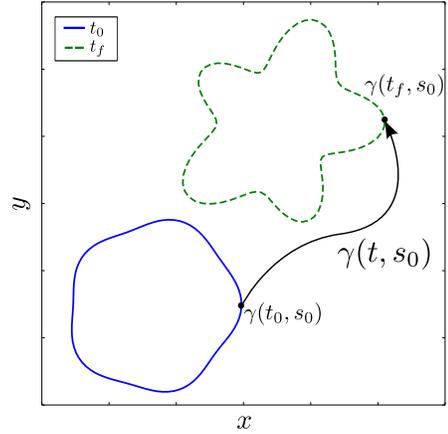


Figure 3: Trajectory of an arbitrary point in the boundary, that starts in $\gamma(t_0, s_0)$ and finishes in $\gamma(t_f, s_0)$.

The general case for every s , not only s_0 , is described in the following subsection.

B. Estimating the boundary behavior

As the matrix β changes continuously with respect to the parameter s (Definition 1), we define $\beta(s)$ as a general function that returns a matrix $2 \times (n+1)$, with values depending on the parameter s . Therefore, we can generalize Equation 1 for every $s \in [0, 1]$ as

$$\hat{\gamma}(t, s) = \beta(s) F. \quad (2)$$

Each value of the function $\beta(s)$ must satisfy the following properties.

Properties 1: The function $\beta(s)$ satisfies:

- 1) The function $\beta(s)$ is a closed loop with $\beta(0) = \beta(1)$.
- 2) The function $\beta(s)$ is continuous and differentiable.
- 3) The function $\beta(s)$ is injective for every $s \in [0, 1)$.

We approximate the function β with a periodic function $\hat{\beta}$ represented by a linear combination of sines and cosines. This approach satisfies the three mandatory properties for $\beta(s)$. We take the first $2m+1$ terms of the Fourier series for each β_{ij} ,

$$\hat{\beta}_{ij}(s) = a_0^{(ij)} + \sum_{k=1}^m a_k^{(ij)} \sin(2\pi ks) + \sum_{k=1}^m b_k^{(ij)} \cos(2\pi ks). \quad (3)$$

Separating the coordinates of γ into two independent functions, we can rewrite the γ as

$$\gamma(t, s) = \begin{bmatrix} x(t, s) \\ y(t, s) \end{bmatrix},$$

we rewrite (2) with $\hat{\beta}$,

$$\hat{\gamma}(t, s) = \begin{bmatrix} \hat{x}(t, s) \\ \hat{y}(t, s) \end{bmatrix} = \begin{bmatrix} \hat{\beta}_x(s)F \\ \hat{\beta}_y(s)F \end{bmatrix}, \quad (4)$$

where $\beta_x(s)$ and $\beta_y(s)$ are the first and second rows of $\beta(s)$, respectively. Taking $G = [1, \sin(s), \dots, \sin(ms), \cos(s), \dots, \cos(ms)]^T$, and C_x and

C_y as constant matrices of dimensions $(2m + 1) \times (n + 1)$, we have

$$\beta_x = G^T C_x. \quad (5)$$

$$\beta_y = G^T C_y. \quad (6)$$

Substituting the Equations (5) and (6) in (4),

$$\hat{\gamma}(t, s) = \begin{bmatrix} G^T C_x F \\ G^T C_y F \end{bmatrix}, \quad (7)$$

and multiplying the matrices, we have

$$\hat{\gamma}(t, s) = \begin{bmatrix} \sum_i^{n+1} \sum_j^{2m+1} c_{ij} F_i G_j \\ \sum_i^{n+1} \sum_j^{2m+1} d_{ij} F_i G_j \end{bmatrix},$$

where c_{ij} , and d_{ij} are the elements of the matrices C_x and C_y respectively. Finally, making C as the matrix of constants with dimension $(n + 1)(2m + 1) \times 2$ and H as the Kronecker product $H = F \otimes G = [F_1 G_1, F_1 G_2, \dots, F_{n+1} G_{2m+1}]^T$, we obtain

$$\hat{\gamma}(t, s) = C^T H. \quad (8)$$

Based on this model, we can frame this problem as a linear regression system, where we have to analyze the number of terms that should be used in the function vector H and attempt to infer the matrix of weights C .

It is important to highlight the influence of the parameters n and m . The parameter n determines the polynomial degree to approximate the trajectory of a single point in the closed curve. It can be chosen based on the number of identified cycles or depending on the order of the phenomena, *e.g.* in the plume experiment (Sec. VI-B), we used $n = 2$ as the Eq. 13 is a second order differential equation. As the parameter m , normally $m > 6$, determines the maximum frequency along the closed curve. For example, the simplest case $m = 1$ approximates the boundary by a circle. A large number for m can generate over-fitting in the presence of noisy measurements.

C. Modeling as a linear system

We use the time t and the curve parameter s as input variables, and the spatially located variables (x, y) as the output. For these reasons, we use the trajectory information $\mathcal{D} = \{(t_i, x_i, y_i, s_i) | i = 0, 1, \dots, k\}$ to predict the anomaly behavior by attempting to estimate the parameter matrix C of Equation (8).

We model the problem as a linear system with the form

$$A C = Y, \quad (9)$$

where Y is a matrix with dimension $k \times 2$ that contains each output location (x_i, y_i) , $i = 1, \dots, k$ in the robot's trajectory \mathcal{D} ; C is the weighted matrix (of Eq. 8) that we want to estimate; and A is the design matrix created using the input (t_i, s_i) , $i = 1, \dots, k$ and the functions of H , defined as

$$A = \begin{bmatrix} f_1(t_1, s_1) g_1(t_1, s_1) & \dots & f_{n+1}(t_1, s_1) g_{2m+1}(t_1, s_1) \\ f_1(t_2, s_2) g_1(t_2, s_2) & \dots & f_{n+1}(t_2, s_2) g_{2m+1}(t_2, s_2) \\ \vdots & \vdots & \vdots \\ f_1(t_k, s_k) g_1(t_k, s_k) & \dots & f_{n+1}(t_k, s_k) g_{2m+1}(t_k, s_k) \end{bmatrix}$$

Therefore, the independent regressors in matrix A are functions of t and s and the coordinates (x, y) are the dependent variables for the regression problem. The polynomial degree is associated with the number of cycles as $|\mathcal{O}| \geq n$ and the m terms in the finite Fourier series depends on the number of sample points k , because it must satisfy $k \geq (2m + 1)(n + 1)$.

D. Estimating by least-squares

The objective consists of adjusting the parameter matrix C (from Equation 9) of the model function $\hat{\gamma}$ to best fit the data set $\mathcal{D} = \{(t_i, x_i, y_i, s_i) | i = 0, 1, \dots, k\}$. Thus, we try to find the matrix C that minimizes the Euclidean norm

$$\|Y - A C\|^2.$$

The *least squares method* [15] finds its optimum by assuming that the errors $\epsilon_i = Y_i - A_i C_i$ are independent random variables for all $i = 1, 2, \dots, k$. Assuming that A is a full-rank matrix, we can estimate \tilde{C} by solving

$$\tilde{C} = (A^T A)^{-1} A^T Y. \quad (10)$$

Therefore, $\gamma(s, t)$ is approximated by $\tilde{\gamma}(s, t) = \tilde{C}^T H$. We present some experiments of the implementation of this method in Section VI.

V. CONVERGENCE ANALYSES

In this section, we show that our method produces accurate estimate $\tilde{\gamma}(s, t)$ of $\gamma(s, t)$ if the number k of sampling points is large enough and the parameters n and m are large enough to describe de phenomenon. By the repeated use of the triangle inequality, we have

$$|\tilde{\gamma}(s, t) - \gamma(s, t)| \leq |\tilde{\gamma}(s, t) - \hat{\gamma}(s, t)| + |\hat{\gamma}(s, t) - \gamma(s, t)|. \quad (11)$$

The Weierstrass approximation theorem [16] guarantees that the last term on the right hand side can be made arbitrarily small by taking the polynomial degree n to be large enough. The second term can also be made arbitrarily small. The reason is that, by the Fourier approximation theorem, there is an integer m such that, for $m > m_{ij}$, we have

$$|\hat{\beta}_{ij}(s) - \beta_{ij}(s)| < \frac{\epsilon}{2(n + 1)t_f^n}.$$

Taking $m > \max m_{ij}$ we have the inequality valid for all i, j and for all s . Therefore,

$$|\hat{x}(t, s) - \hat{x}(t, s)| \leq \sum_{j=0}^n |\hat{\beta}_{0j} - \beta_{0j}| |t^j| \leq \epsilon/2.$$

A similar calculation for $y(t, s)$ provides an arbitrarily small upper bound for the the second term in (11). That is, $|\hat{\gamma}(s, t) - \gamma(s, t)| \leq \epsilon$. Finally, the first term in (11) can also be made arbitrarily small assuming the classic linear regression model. Indeed, usual least squares regression asymptotic theory gives us that $\tilde{\gamma}(s, t)$ is a consistent estimator of $\hat{\gamma}(s, t)$. That is, $|\tilde{\gamma}(s, t) - \hat{\gamma}(s, t)| < \epsilon$ if k is large enough. Putting these three bounds together, we find that $|\tilde{\gamma}(s, t) - \gamma(s, t)| \leq 3\epsilon$.

VI. EXPERIMENTS

In our experiments, we initially work with a boundary based on an analytic function and subsequently with a simulated substance based on the advection-diffusion model. In order to quantify the difference between the real anomaly Ω_t and a prediction $\hat{\Omega}_t$, we use the Lebesgue measure of the symmetric difference between the two sets as a metric:

$$\delta(\Omega_t, \hat{\Omega}_t) := \mu(\Omega_t \setminus \hat{\Omega}_t) + \mu(\hat{\Omega}_t \setminus \Omega_t), \quad (12)$$

where μ is the Lebesgue measure, which computes the area of a set in \mathbb{R}^2 . The symmetric difference computes the error between the prediction and the real anomaly. It takes into account the union of the wrong covered area ($\Omega_t \setminus \hat{\Omega}_t$) and the uncovered area ($\hat{\Omega}_t \setminus \Omega_t$). This metric is used in the related works [9] and [13].

A. Simulating a boundary function

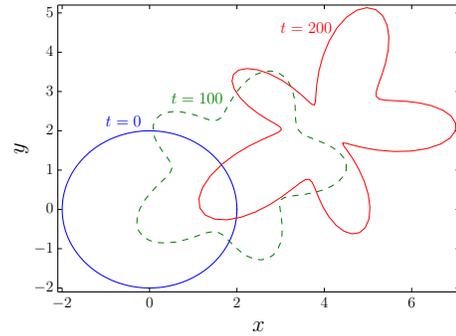
We initially simulate an anomaly which simultaneously translates, expands and shrinks its shape. The boundary satisfies the following equation,

$$\gamma_1 = \frac{1}{100} [4t + (t \sin(4\pi s) + 2t \cos(10\pi s) + 800) \cos(2\pi s)],$$

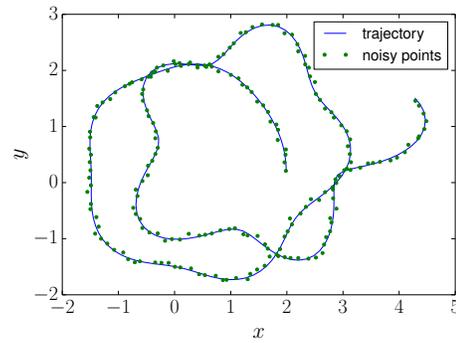
which is a variation of the boundary function in [9], but including a translation factor. In Figure 4a, it can be seen that the boundary γ_1 transforms its shape from a circle at $t = 0$ to a star at $t = 200$. Additionally, we can see that the boundary is also translating with constant velocity. The robot moves with velocity $v = 0.79$ and its trajectory is shown in Figure 4b. Another way to present the estimated trajectory is shown in Figure 1, where the z-axis represents the time variable. We can see the boundary at different times and how the robot trajectory has only one point in the boundary for each time.

Our objective is to predict the boundary state at time $t = 200$ by using the sampled data \mathcal{D} of the robot trajectory from time $t = 0$ to $t = 100$. We estimate the curve parameter \hat{s} as described in Section III. The estimation of the s parameter introduces an additional error that is minimized after applying the *least-squares*. The identified cycles are shown in Figure 5a and the resultant linear approximation for \hat{s} in contrast to the ground truth s is shown in Figure 5b. The model that we approximate is based on Equation 9. We use $n = 1$ for the polynomial and $m = 10$ for the Fourier series. Therefore, we estimated the 42 terms of the matrix A by using Equation 10. Using the exact points in the trajectory (represented by the continuous line in Figure 4b), the prediction error is $\delta = 4.27e - 12$ for all $t \in \mathbb{R}_{\geq 0}$, i.e. the approximation fits almost exactly to the real boundary.

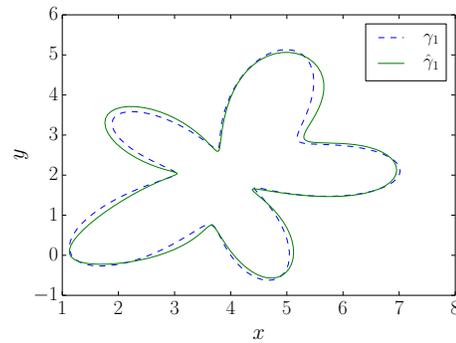
In order to show that our method also works with noisy measurements, we introduced a Gaussian error for each measurement along the trajectory (small dots around the continuous line in Figure 4b). The prediction is illustrated in Figure 4c by the continuous line, where the resultant error is $\delta = 1.53$. We can observe that in this kind of analytical boundaries, the extrapolation for prediction fits very close to the real one, even for long future predictions and with the presence of noisy measurements.



(a) Evolution of the boundary γ_1 at three different times $t = 0, 100, 200$.



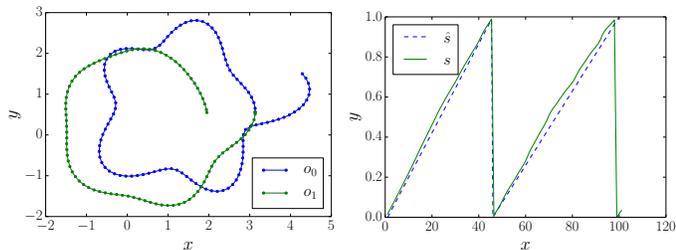
(b) Robot trajectory from time $t_0 = 0$ to $t_f = 100$, it is represented by the continuous line. The noisy measurements along the trajectory are represented by the dots.



(c) Prediction for $t = 200$ by using noisy measurements.

Figure 4: Dynamic boundary γ_1 , the robot trajectory and the resultant prediction.

In [9], the boundary is approximated by a polygon, where each vertex is updated as the robot moves close to it. Good approximations can be achieved when many robots are distributed along the boundary. In this case, every vertex is updated quickly by different robots. However, when a single robot is used, each vertex is only updated after a lap and therefore, in the case of time-varying boundaries, the error in this method will not converge to zero. Our technique is innovative in that, when the boundary shape is changing, our method is able to predict this motion for all the boundary points along the approximated closed-curve. However, our prediction method



(a) Identified cycles in the robot trajectory. (b) Comparison between the estimated \hat{s} and the real s .

Figure 5: Estimation of the curve parameter s .

works with phenomena that change smooth in time. Other behaviors that contain oscillations or random movements are difficult to estimate by the polynomial approximation.

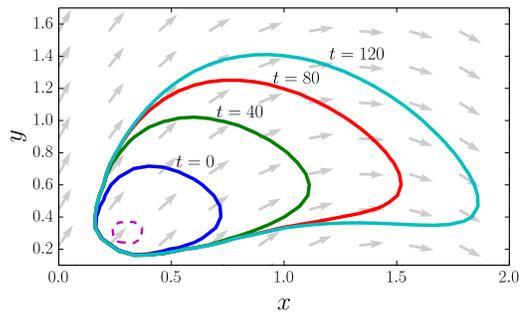
B. Estimating a plume-front behavior

The propagation of a polluting substance in a non-static liquid is widely studied in physics and engineering. This physical phenomena is governed by two processes: diffusion and advection. Diffusion is a smooth behavior that expands the concentration of the substance in the liquid and advection is the movement of the substance due to the dynamics of the liquid. The Navier-Stokes equations [17] describe the motion of viscous fluid substances and the equation for the advection-diffusion behavior is given by

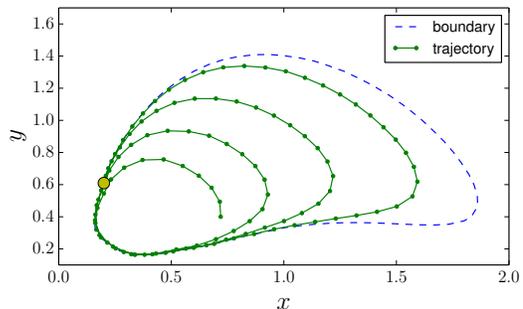
$$\frac{\partial u}{\partial t} + v_x \frac{\partial u}{\partial x} + v_y \frac{\partial u}{\partial y} = k \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (13)$$

where the variable $u \in \mathbb{R}_{\geq 0}$ is the substance concentration; v_x and v_y are the advection velocities in x and y respectively; and $k \in \mathbb{R}_{\geq 0}$ is a diffusion coefficient. The estimation and tracking of plumes with multiple robots is studied in [2], [3]. We apply our proposed model to show that a single robot also gives an accurate estimation and prediction of the behavior of a plume.

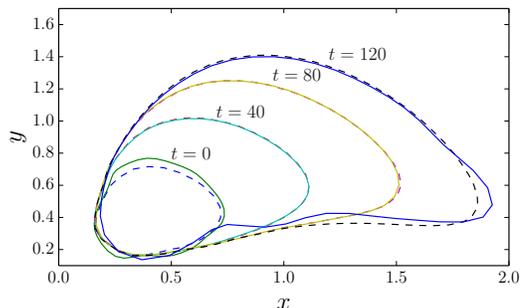
In our experiment, we simulate a point-source pollution by implementing Equation 13 in python. The evolution of a substance is shown in Figure 6a, where the gray arrows represent the vector field for the liquid's velocity $[v_x, v_y]^T$. The robot's trajectory around this substance with velocity $v = 0.09$ is shown in Figure 6b. As a result, we can see in Figure 6c that our approach can estimate the whole boundary for an arbitrary time $t \in [0, 110]$ or predict for $t > 110$, even when the robot takes either only a sample or no samples for each specific time. The metric δ for all the estimations along the time t (based on the data of Figure 6b) is plotted in Figure 7. If the robot's velocity increases from $v = 0.09$ to $v = 0.12$ or $v = 0.15$, the number of identified cycles increases from 3 to 4 and 5 respectively. Figure 7 also shows how the error δ is reduced because the same 110 samples are more evenly distributed in the rectangular (s, t) space. As our function interpolates the sample values with old and recent data, we can see that the estimated boundary in the time interval $t \in [20, 90]$ presents better accuracy. For the time $t = 110$, the error increases because there is data before



(a) Substance evolution (at $t = 0, 40, 80, 120$) in a dynamic liquid. The pollution source is represented by the dashed line.



(b) Robot trajectory from $t_0 = 0$ to $t_f = 110$ with velocity $v = 0.09$. The dashed line shows the boundary at time $t = 120$. The circle represents the last robot location and the dots represent the $k = 110$ sampled points.



(c) Liquid estimation for $t = 0, 40, 80$ and prediction for $t = 120$. The continuous lines represent the approximations and dashed lines are the real boundaries.

Figure 6: Viscous substance in a liquid and the robot trajectory around it.

but not after this moment. The method may predict the future boundaries for $t > 110$, but the error increases proportionally with time.

In real experiments, we used a differential robot with a Motion Capture System for localization in order to track the plume-front as in simulations. In the accompanying video [<https://www.youtube.com/watch?v=1bF9U3nbXJ8>], it can be seen how the robot moves around the boundary in order to sample it. As the actual robot is nonholonomic, the maximum curvature to be tracked is restricted by the steering capability and the floor friction. We can see that the robot cannot always track the boundary exactly because of the steering limitation during the tracking process. The resultant sampled points along

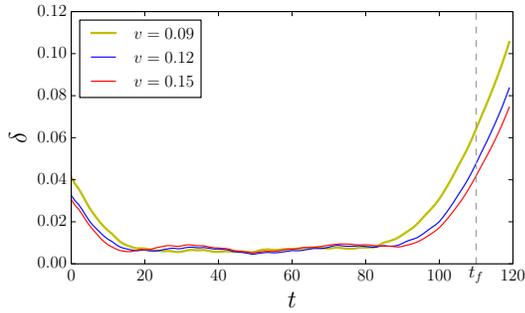
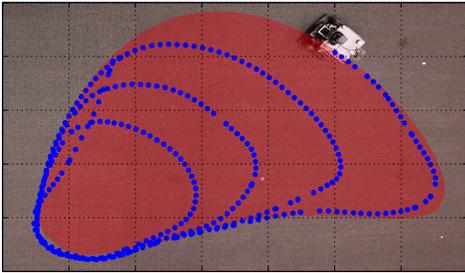
Figure 7: δ -error in time with different velocities.

Figure 8: Actual robot circulating around the dynamic boundary. The dots represent the sampled points along the robot trajectory.

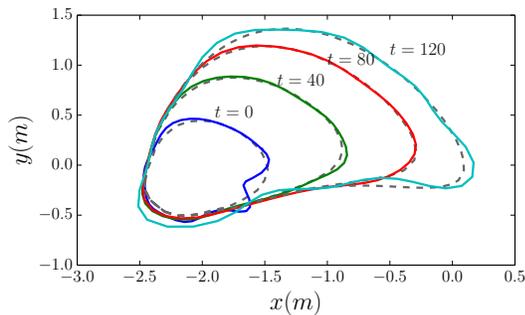


Figure 9: Liquid estimation by an actual robot. Continuous lines represent the estimations and dashed lines are the real boundaries.

the trajectory are presented in Figure 8. The error to move the robot along the boundary slightly affects the method, but the obtained prediction is very close to the real boundary as can be seen in Figure 9.

As a result, we can use a single robot to estimate and predict boundaries with an approximation accuracy $\delta < 0.15m^2$. The requirements to obtain a good estimation are: to complete the minimum number of cycles $|\mathcal{O}| \geq n$; to sample a number of points $k \geq (2m+1)(n+1)$; we also require that the robot move with high enough linear velocity v to obtain a good approximation \hat{s} of the curve parameter s . As we argued in Section V, the method works even in the presence of noisy location measurements, as the error $\delta \rightarrow 0$ when the number of sampled points increases.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a model that predicts the behavior of environmental boundaries by the parameters of an analytical function. We analyzed the convergence and demonstrated by simulations the efficiency of the proposed method. We showed how to estimate a boundary with high accuracy using a single robot without previous knowledge of the boundary dynamics.

The robot can predict the dynamics of the boundary by completing cycles around it. The estimation can be improved by increasing the sampling frequency and the velocity of the robot. When the robot is moving fast, it can identify more cycles with more accuracy and, as a consequence, the estimation is also improved, as we showed in the simulations.

We plan to extend the proposed model to use multiple robots in order to track fast-changing boundaries. We also desire to validate this model in outdoor environments in order to analyze its behavior in situations when the boundary is affected by several external variables.

REFERENCES

- [1] A. Bertozzi and M. Kemp, "Determining environmental boundaries: Asynchronous communication and physical scales," in *Proc. Block Island Workshop Cooperative Control*, 2004.
- [2] S. Li, Y. Guo, and B. Bingham, "Multi-robot cooperative control for monitoring and tracking dynamic plumes," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 67–73.
- [3] M. Fahad, N. Saul, Y. Guo, and B. Bingham, "Robotic simulation of dynamic plume tracking by unmanned surface vessels," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 2654–2659.
- [4] D. Casbeer, R. Beard, T. McLain, S.-M. Li, and R. Mehra, "Forest fire monitoring with multiple small uavs," in *American Control Conference, 2005. Proceedings of the 2005*, June 2005, pp. 3530–3535 vol. 5.
- [5] D. Bruemmer, "A robotic swarm for spill finding and perimeter formation," *Spectrum: International Conference on Nuclear and Hazardous Waste Management*, 2002.
- [6] A. L. Bertozzi, "Environmental boundary tracking and estimation using multiple autonomous vehicles," *2007 46th IEEE Conference on Decision and Control*, pp. 4918–4923, 2007.
- [7] D. Marthaler and A. Bertozzi, "Collective motion algorithms for determining environmental boundaries," in *SIAM Conference on Applications of Dynamical Systems*, 2003.
- [8] F. Zhang and N. Leonard, "Generating contour plots using multiple sensor platforms," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, June 2005, pp. 309–316.
- [9] S. Susca, F. Bullo, and S. Martinez, "Monitoring Environmental Boundaries With a Robotic Sensor Network," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 288–296, Mar. 2008.
- [10] J. Clark and R. Fierro, "Mobile robotic sensors for perimeter detection and tracking," *ISA transactions*, vol. 46, no. 1, pp. 3–13, Feb. 2007.
- [11] G. Zhang, G. K. Fricke, and D. P. Garg, "Spill detection and perimeter surveillance via distributed swarming agents," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 1, pp. 121–129, 2013.
- [12] S. Valli *et al.*, "An adaptive refresh distributed model for estimation and efficient tracking of dynamic boundaries," in *Communications (NCC), 2015 Twenty First National Conference on*. IEEE, 2015, pp. 1–6.
- [13] D. Saldana, R. Assuncao, and M. Campos, "A distributed multi-robot approach for the detection and tracking of multiple dynamic anomalies," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 1262–1267.
- [14] S. Duttagupta, K. Ramamritham, and P. Kulkarni, "Tracking dynamic boundaries using sensor network," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 10, pp. 1766–1774, 2011.
- [15] D. Borowiak, "Linear models, least squares and alternatives," *Technometrics*, vol. 43, no. 1, pp. 99–99, 2001.
- [16] G. F. Simmons, *Topology and modern analysis*. McGraw-Hill New York, 1963, vol. 3.
- [17] R. Temam, *Navier-Stokes equations: theory and numerical analysis*. American Mathematical Soc., 2001, vol. 343.