Monitoring Weeder Robots and Anticipating Their Functioning by Using Advanced Topological Data Analysis

Tarek Frahi [1], Abel Sancarlos [1,2], Mathieu Galle [3], Xavier Beaulieu [3], Anne Chambard [2], Antonio Falco [4], Elias Cueto [5] and Francisco Chinesta [1,2]*

[1]PIMM Lab and ESI Group Chair, Arts et Metiers Institute of Technology, Paris, France, [2]ESI Group, Rungis, France, [3]VITIROVER, Saint-Emilion, France, [4]ESI-CEU International Chair CEU-UCH, Departamento de Matematicas, Fisica y Ciencias Tecnologicas, Universidad Cardenal Herrera-CEU, Valencia, Spain, [5]Aragon Institute of Engineering Research, Universidad de Zaragoza, Zaragoza, Spain

The present paper aims at analyzing the topological content of the complex trajectories that weeder-autonomous robots follow in operation. We will prove that the topological descriptors of these trajectories are affected by the robot environment as well as by the robot state, with respect to maintenance operations. Most of existing methodologies enabling efficient diagnosis are based on the data analysis, and in particular on some statistical quantities derived from the data. The present work explores the use of an original approach that instead of analyzing quantities derived from the data, analyzes the "shape" of the data, that is, the time series topology based on the homology persistence. We will prove that this procedure is able to extract valuable patterns able to discriminate the trajectories that the robot follows depending on the particular patch in which it operates, as well as to differentiate the robot behavior before and after undergoing a maintenance operation. Even if it is a preliminary work, and it does not pretend to compare its performances with respect to other existing technologies, this work opens new perspectives in considering quite natural and simple descriptors based on the intrinsic information that data contains, with the aim of performing efficient diagnosis and prognosis.

Keywords: autonomous robots, monitoring, topological data analysis, trajectory analysis, artificial intelligence, data classification

1 INTRODUCTION

Autonomous robots follow a number of rules introduced into their controllers (Alatise and Hancke, 2020; Shalal et al., 2013; Mohanty and Parhi, 2013). However, when they interact with the environment, small variations may result in long-time unpredictable motion. This behaviour is very usual in mechanics, characterizing systems exhibiting deterministic chaos (Avanço et al., 2016; Gupta et al., 2014).

In the practical case addressed in the present paper, a weeder robot (usually a float of them) is expected to cover a patch of a vineyard, in an optimal manner. Here, "optimal manner" refers to the path-line that allows covering the whole patch in a minimum time. However, the ground orography has a significant variability, as well as the location of the grapes. Robots are aimed at colliding the

grape foots in order to remove the grass around, and then numerous collisions following different directions are needed to ensure that all the grass around the grape foot is adequately removed. **Figure 1** depicts a VITIROVER MOWER ROBOT (https://www.vitirover.fr/en-robot for the technical specifications) considered in the present study under operational conditions.

The environmental variability (ground, grape location, grass distribution and size, and fixed and mobile obstacles) as well the intrinsic sensibility of the dynamics to small perturbations in the physical and operational conditions, provides an uncertain environment that makes useless the use of a deterministic framework for anticipating the robot trajectory. Thus, a random motion framework seems to be the most useful alternative.

In practice, to avoid under-performances characteristic of fully random motion, random motion operating at the local scale is combined with a more global deterministic planning that tries to better control the vineyard coverage by sequencing the operation at the different local patches covering the whole domain (Kavraki et al., 1998).

The present work does not aim at addressing such optimized operation conditions that will be addressed in a future publication under progress, but it aims at analyzing the data collected from a robot operating in different patches and under different conditions (with respect to the maintenance operations) in order to identify the existence of patterns able to identify the particular patch in which the robot operates, or to distinguish the different robot states with respect to the maintenance operations.

Having a sort of QR-code or identity card of each robot, when it operates within each patch, in a particular state (healthy or unhealthy), is of major relevance with respect to the predictive or operational maintenance of robots or floats of autonomous robots (Kavraki et al., 1998).

Most of existing methodologies enabling efficient diagnosis are based on the data analysis, and in particular on some statistical quantities derived from the data (Lhermitte et al., 2011) while the present paper aims at extracting information to be transformed into knowledge, from the data collected from each weeder robot, in particular the positions visited by the robot during its operation, and more concretely the topology contained in this data. Our goal is to extract the maximum information that could serve for differentiating them, enabling unsupervised clustering and/or supervised classification, prior to any action concerning diagnosis or modeling based on the use of adapted regressions. This first work aims at introducing a new methodology and does not pretend to compare its performances with respect to other existing technologies, comparison that will be addressed in a future work.

2 METHODS

Using data clustering is almost straightforward, as soon as data is homogeneous and quantitatively expressible using integer or real numbers, enabling boolean or algebraic operations (addition, multiplication, ...). The interest of organizing data in groups,

FIGURE 1 | Weeder robot from VITIROVER micro robotique viticole.

in a supervised or unsupervised manner, is that it is assumed that data belonging to a given group shares some qualities with the members of the group (Hastie et al., 2009; Murphy, 2012).

When proceeding in an unsupervised manner, the only information to group the data consists of the distance among them. Data that remain close to each other are expected to share some properties or behavior. This is the rationale considered in the very popular *k-means* technique (MacQueen, 1967; MacKay, 2003). However, the notion of proximity, leading to the derived concept of similarity, needs for the definition of a metric for comparison purposes. When data are well defined in a vector space, distances can be defined and data can be compared accordingly. In the case of supervised classification one is looking for the linear (or non-linear) Frontier separating the different groups on the basis of a quality or property that drives the data clustering. In this last case, the best Frontier separating two groups of data is the one maximizing the distance of the available data to the Frontier, in order to maximize the separation robustness. This is how support vector machine, SVM, works, for instance (Cristianini and Shawe-Taylor, 2000).

In both cases (supervised and unsupervised) the existence of a metric enabling data comparison is assumed. However, very often data could be much more complex, as for example when it concerns heterogeneous information, possibly categorial or qualitative. This is for example the case when a manufactured part is described by its identity card consisting of the name of the employee involved in the operation, the designation of the employed materials (some of them given by its commercial

name), the temperature of the oven in which the part was cured and the processing time. In that case, comparing two parts becomes quite controversial if the employed metric is not properly defined. In these circumstances, usually, metrics are learned from the existing training data, as is the case when using decision trees (or its random forest counterpart) (Kirkwood, 2022; Breiman, 2001), code-to-vector Martín et al. (2019) or neural networks Goodfellow et al. (2016).

The situation becomes even more extreme when data have a large and deep topology content. This is the case for example of time series or images of rich microstructures. These are usually encountered in material science when describing metamaterials (also called functional materials), or those exhibiting gradient of properties or mesoscopic architectures. Thus, even in nominal conditions, time series will differ if they are compared from their respective values at each time instant. That is, two time series, even when they describe the same system in similar conditions, never match perfectly. Thus, they differ even if they resemble in a certain metric that should be learned. For example, our electrocardiogram measured during two consecutive minutes will exhibit a resemblance, but certainly both of them are not identical, thus making a perfect match impossible. A small variation will create a misalignment needing for metrics less sensible to these effects. The same rationale applies when comparing two profiles of a rough surface, two images of a foam taken in two close locations, ... they exhibit a resemblance even if they do not perfectly match.

Thus, techniques aiming at aligning data were proposed. In the case of time-series, Dynamic Time Warping, DTW (Müller, 2007; Senin, 2008) has been successfully applied in many domains. The theory of optimal transport arose as a response to similar issues (Villani, 2006).

Another route consists of renouncing to *align* the data, and focussing on extracting the adequate, goal-oriented descriptors of these complex data, enabling comparison, clustering, classification and modelling (from non-linear regressions) (Lhermitte et al., 2011).

A first possibility consists of extracting the main statistical descriptors of time series or images (moments, correlations, covariograms, ...) (Torquato, 2002). Sometimes, data expressed in the usual space and time domains, are transformed into other spaces where their manipulation is expected to be simpler, like Fourier, Laplace, DCT, Wavelet, ... descriptions of data. The most valuable (in the sense given later) descriptions seem to be those maximizing sparsity. These are widely considered when using compressed sensing (Ibañez et al., 2019), because it represents a compact, concise and complete way of representing data that seemed much more complex in the usual physical space (space and time).

The present work considers this last route, but uses a description based on the topology of data, described later, and successfully considered in our former works for addressing complex mesostructures Yun et al. (2020), time-series Frahi et al. (2021a), rough surfaces Frahi et al. (2020) and shapes Frahi et al. (2021b), with the aim of classifying and also constructing robust regressions expressing properties or performance from the input data expressed from its topological description.
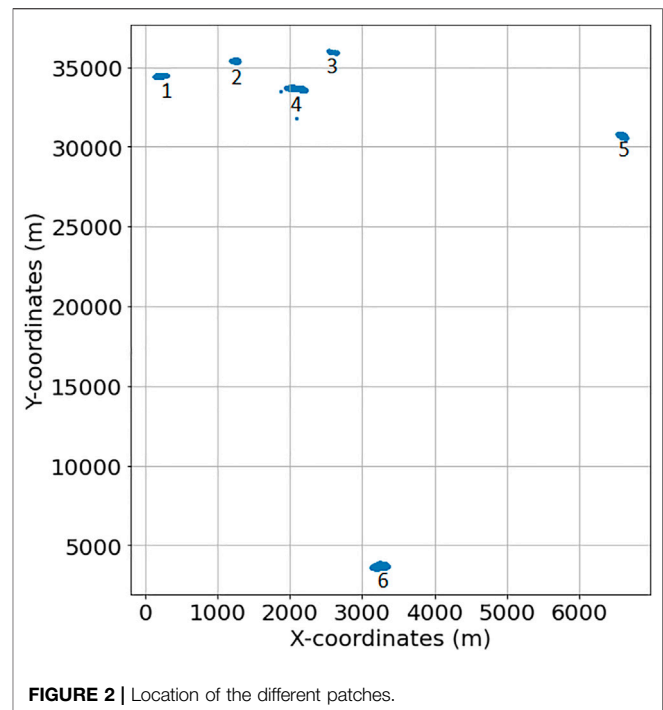

FIGURE 2 | Location of the different patches.

The present study, when compared with our former developments, addresses a new and complex purpose: how the topology contained in the trajectory that an autonomous robot follows in a cloudy environment (where interactions limits the predictability horizon) can inform on the robot location (which patch into the whole vineyard) or the robot state (with respect to maintenance operations).

2.1 Data Description

In the study that follows, we consider a dataset consisting of the x and y-coordinates, calculated from the GPS longitudes and latitudes, representing the recorded position of the robot at time t:

$$\mathcal{D} = \{(x(t), y(t), t), \ t \in \mathcal{T}\}.$$

These coordinates span six different disjoint geographical patches within the whole vineyard, as illustrated in **Figure 2**, that have been recorded in a period of time $\mathcal{T}$ leading to the maps reported in **Figure 3** that reflects the robot's trajectory.

Maintenance operations are also known and properly identified in the provided dataset. Thus, the dataset consists of a collection of n discrete, finite and compact two-dimensional trajectories $\mathbb{S}_1, \ldots, \mathbb{S}_n$.

2.2 Geometrical Features

We are interested in extracting the geometrical and topological features of the trajectories in $\mathcal{D}$ across different scales. For that purpose, we introduce the so-called *Rips filtration*. We construct a *Rips complex* from simplexes of varying dimensions that are generalizations of triangles of varying dimensions. More specifically, a d-simplex is the smallest convex set of $d + 1$ points, $x_0, \ldots, x_d$ where $x_1-x_0, \ldots, x_d-x_0$ are linearly

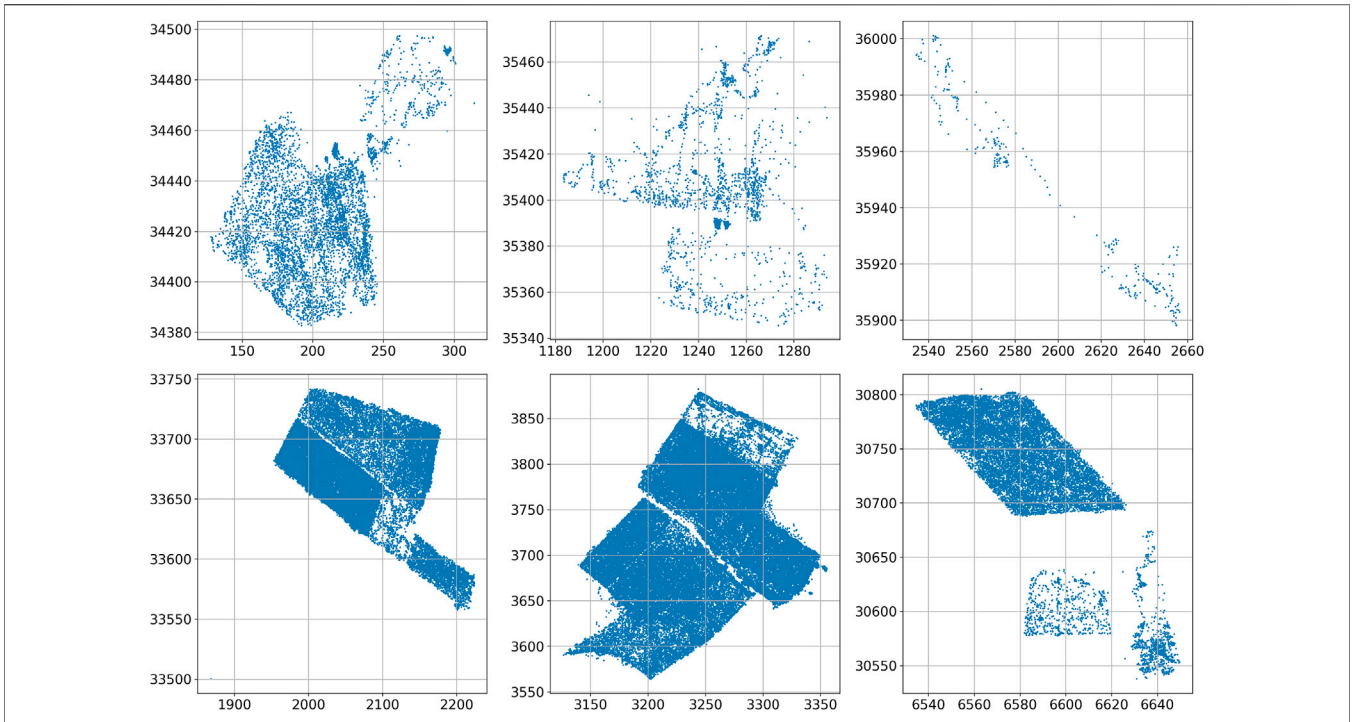**FIGURE 3** | Robot trajectories in the six considered vineyard patches, from right to left and top to bottom: 1, 2, 3, 4, 5 and 6 (units in meters).
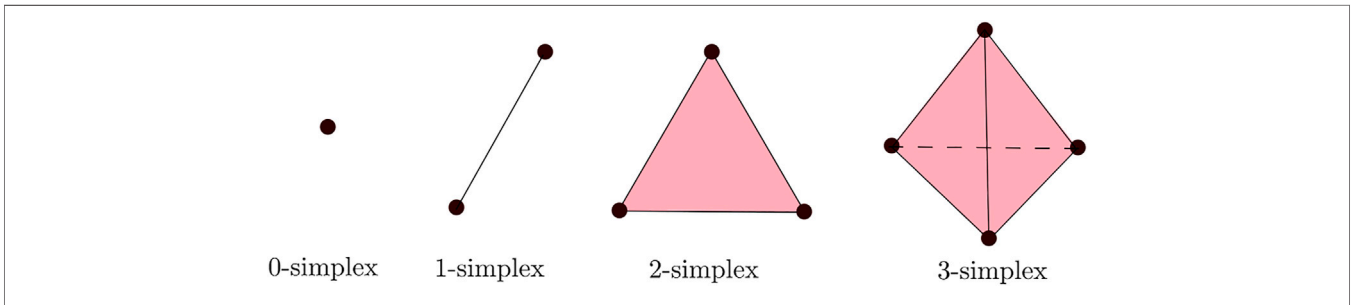

FIGURE 4 | Simplexes of different dimensions.

independent, as illustrated in **Figure 4**. The so-called *abstract simplicial complex* is a finite collection of sets that is closed under the subset relation, i.e., if $a \in A$ and $b \subset a$, then $b \in A$.

Let $\mathbb{S}$ be a trajectory, defined from a finite compact set of points in $\mathbb{R}^2$, and $\epsilon \geq 0$. The Rips complex of $\mathbb{S}$ at scale ϵ, $\mathcal{R}_\epsilon(\mathbb{S})$, is the abstract simplicial complex consisting of all subsets of diameter up to ϵ:

$$\mathcal{R}_\epsilon(\mathbb{S}) := \{\sigma \subset \mathbb{S} \mid \operatorname{diam}(\sigma) \leq \epsilon\},$$

where the diameter of a set of points is the maximum distance between any two points in the set.

Geometrically, we can construct the Rips complex by considering balls of radius $\frac{\epsilon}{2}$, centred at each point in $\mathbb{S}$. Whenever d balls have pairwise intersections, we add a $d-1$ dimensional simplex. An example of Rips complex is given in **Figure 5**.

A *filtration* of a simplicial complex $\mathcal{K}$ is a nested sequence of subcomplexes starting at the empty set and ending with the full simplicial complex

$$\emptyset \subset \mathcal{K}_0 \subset \cdots \subset \mathcal{K}.$$

By varying the value of the scale parameter ϵ, from $\epsilon_{\min} = 0$ to $\epsilon_{\max} = \operatorname{diam}(\mathbb{S})$ we get a family of nested Rips complexes known as the Rips filtration.

2.3 Persistent Homology

In order to have a more exhaustive view on how the features are changing across different scales, the appearance and disappearance of each feature within the filtration is tracked and coded into the homology groups $H_k(\mathbb{S})$, where k is the homology dimension. The elements of a *Homology Group* $H_k(\mathbb{S})$ are classes of chain of simplexes ("packets") in the Rips complex.
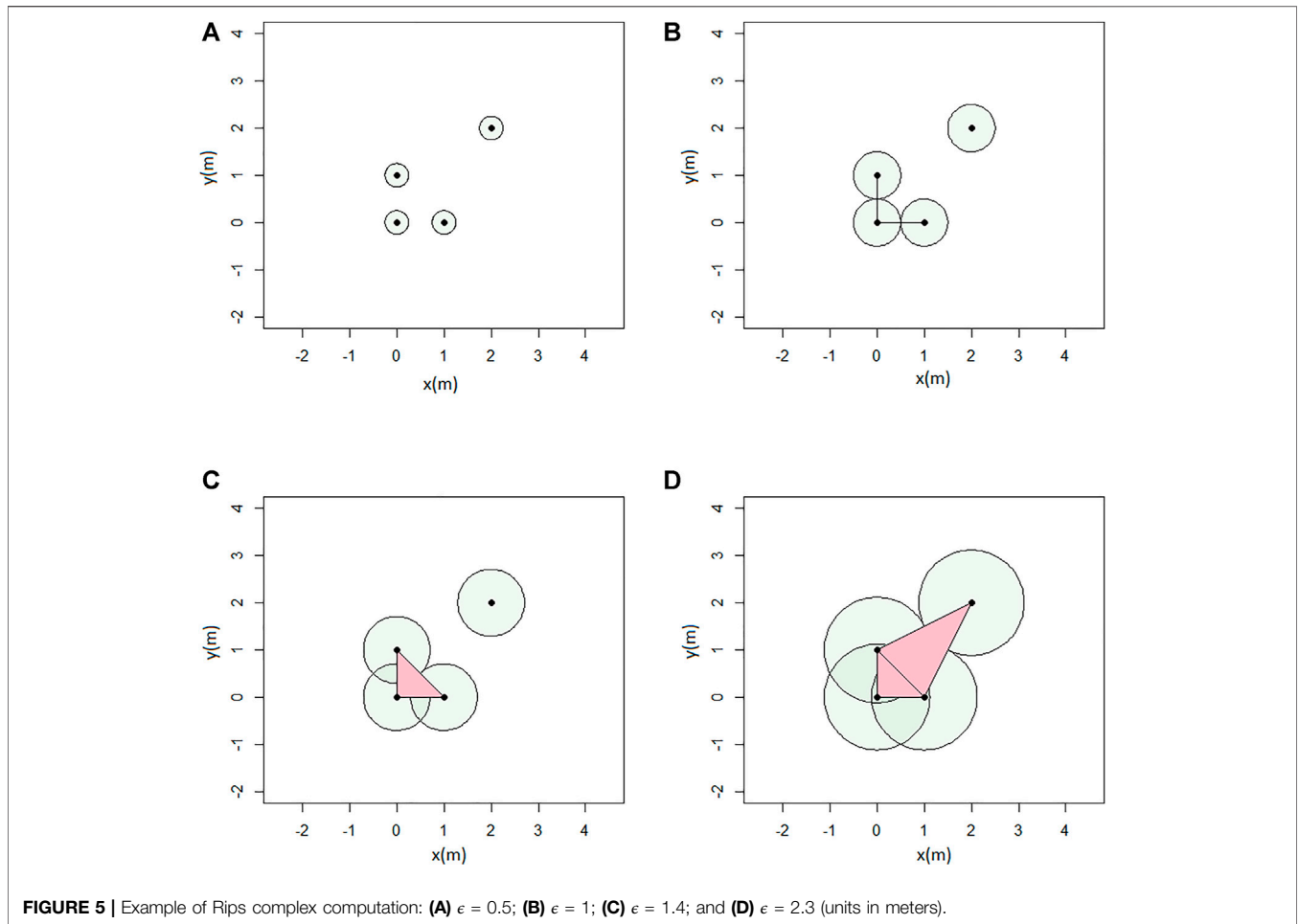
FIGURE 5 | Example of Rips complex computation: **(A)** $\epsilon = 0.5$; **(B)** $\epsilon = 1$; **(C)** $\epsilon = 1.4$; and **(D)** $\epsilon = 2.3$ (units in meters).

The use of homology groups allows us to perform algebraic operations over the simplicial elements. The homology group $H_0(\mathbb{S})$ represents the vertices, while the homology group $H_1(\mathbb{S})$ represents the cycles (loops) formed in the simplicial complex. Since our data is in $\mathbb{R}^2$ we are only interested in $k = 0$ and $k = 1$.

Given a homology group, we can now define how to track the appearance of the features across different scales, by defining the homology group at a scale ϵ, $H_k^\epsilon(\mathbb{S})$. It represents the classes of simplexes as described previously, but taken from $\mathcal{R}_\epsilon(\mathbb{S})$. That is, the elements of $\mathcal{R}_\epsilon(\mathbb{S})$ with a filtration value lower than ϵ. This approach is known as the *persistent homology*. It allows to quantify the appearance and disappearance of the features across the different scales (discretized by considering m values related to ϵ_j, $j = 0, \ldots, m$):

- For $H_0(\mathbb{S})$, the birth scale of all vertices is set to zero, while the death scale is the filtration value at which the vertex has been joined to another one by a segment.
- For $H_1(\mathbb{S})$, the birth scale of a cycle is the filtration value at which a loop has been formed, while the death scale is the filtration value at which the interior of the loop has been covered.

We can formalize this as follows:

- The birth scale b_γ of the feature γ

$$b_\gamma = \min_{0 \le j \le m}\left\{\epsilon_j: \gamma \in H_k^{\epsilon_j}\right\}$$

- The death scale d_γ of the feature γ

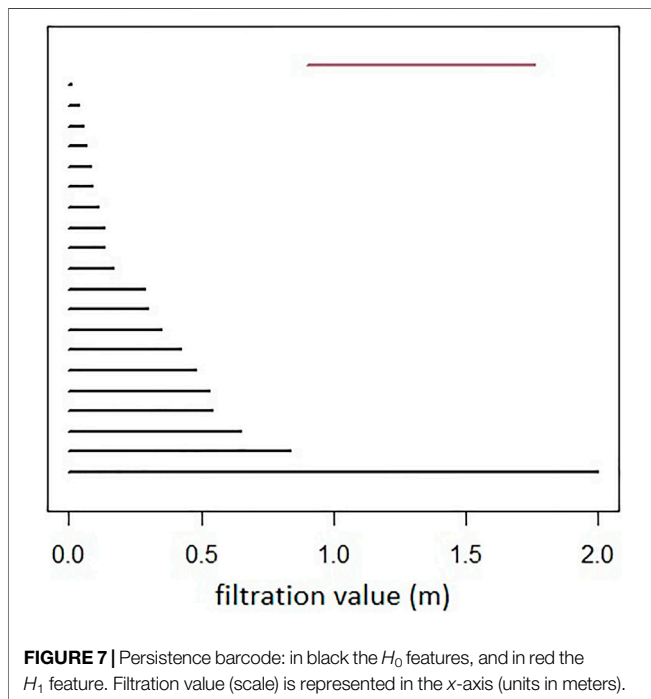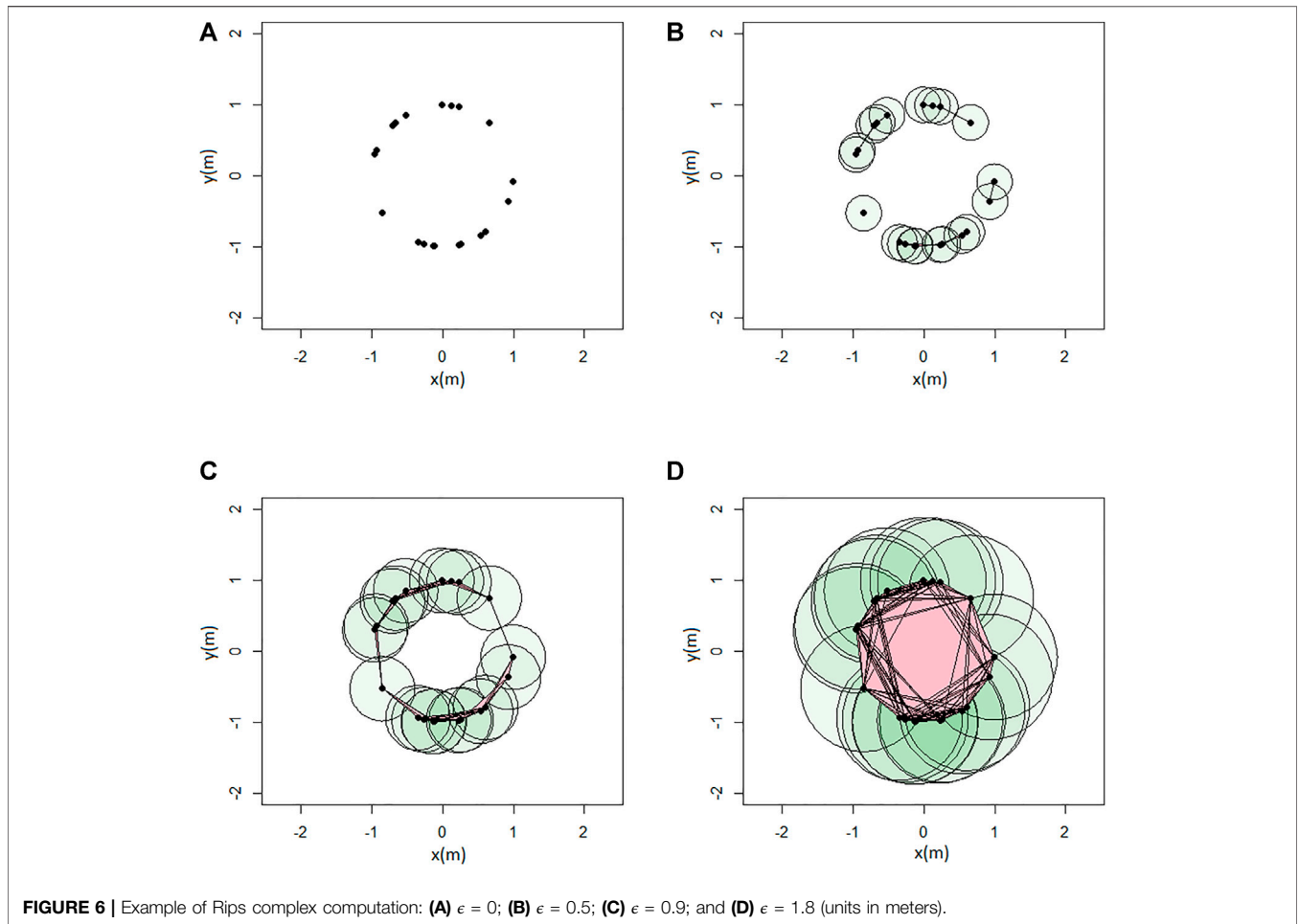$$d_\gamma = \max_{0 \le j \le m}\left\{\epsilon_j: \gamma \in H_k^{\epsilon_j}\right\}$$

The persistence of the features throughout the scales can then be represented by the so-called *persistence barcode* of $\mathbb{S}$. It is a histogram, where the bar associated to each feature starts at the birth scale and ends at the death scale.

An example of persistent homology computation is given with the rips complex in **Figure 6**, and the associated barcode in **Figure 7**. A loop is formed at $\epsilon = 0.9$ (birth) and then covered at $\epsilon = 1.8$ (death). It is represented by the red bar.

A more compact representation of the features persistence is the persistence diagram of $\mathbb{S}$, defined from

$$\mathcal{PD}(\mathbb{S}) = \left\{(b_\gamma, d_\gamma): \gamma \in H_k\right\},$$

where b_γ and d_γ are the birth and death scales associated to the feature γ. In what follows, in the trajectories analysis, we only consider one-dimensional features, i.e., $k = 1$.

FIGURE 6 | Example of Rips complex computation: **(A)** $\epsilon = 0$; **(B)** $\epsilon = 0.5$; **(C)** $\epsilon = 0.9$; and **(D)** $\epsilon = 1.8$ (units in meters).

FIGURE 7 | Persistence barcode: in black the H_0 features, and in red the H_1 feature. Filtration value (scale) is represented in the x-axis (units in meters).

The persistence diagram associated with the Rips complex shown in **Figure 6** is given in **Figure 8**. An equivalent representation of the persistence diagram consists in the so-called life-time diagram of $\mathbb{S}$, which is constructed by means of a bijective transformation $T(a, b) = (a, b-a)$, acting over $\mathcal{PD}(\mathbb{S})$, that is,

$$\mathcal{LT}(\mathbb{S}) := \{(a, b-a) \in \mathbb{R}^2 : (a, b) \in \mathcal{PD}(\mathbb{S})\}.$$

In order to use the persistence features in a machine learning approach, we construct the so-called *persistent image* of $\mathbb{S}$. First, observe that $\mathcal{LT}(\mathbb{S})$ is a finite set of p points,

$$\mathcal{LT}(\mathbb{S}) = \{(a_1, b_1 - a_1), \ldots, (a_p, b_p - a_p)\},$$

and such that $b_1 - a_1 \leq b_2 - a_2 \leq \ldots, \leq b_p - a_p$. Then, consider a non-negative weighting function given by

$$w: \mathcal{LT}(\mathbb{S}) \rightarrow [0, 1]$$

$$(a_i, b_i - a_i) \mapsto w(a_i, b_i - a_i) = \frac{b_i - a_i}{b_p - a_p}, \text{ for } 1 \leq i \leq p.$$

Finally, we fix M, a natural number, and take a bivariate normal distribution $g_u(x, y)$ centered at each point $u \in \mathcal{LT}(\mathbb{S})$ with a variance $\sigma \mathbf{I}_2 = \frac{b_p - a_p}{M} \mathbf{I}_2$ ($\mathbf{I}_2$ is the 2×2 identity matrix). A persistence kernel is then defined according to:

FIGURE 8 | Persistence Diagram: in black the H_0 features, and in red the H_1 feature.
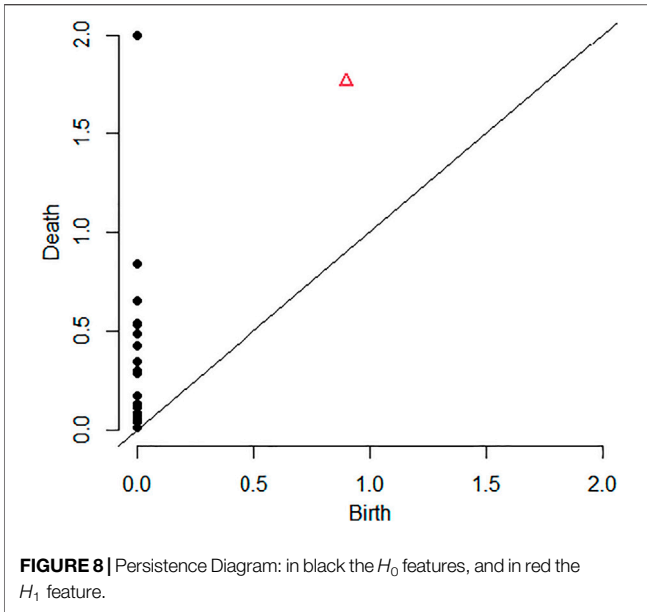
$$\rho_{\mathbb{S}}: \mathbb{R}^2 \to \mathbb{R}$$
$$(x, y) \mapsto \rho_{\mathbb{S}}(x, y) = \sum_{u \in \mathcal{LT}(\mathbb{S})} w(u) g_u(x, y).$$

We associate to a robot trajectory $\mathbb{S} \in \mathbb{R}^2$ a matrix in $\mathbb{R}^{M \times M}$ as follows: let $\delta > 0$ be a non-negative, small enough real number, and then consider a squared region $\Omega_{\mathbb{S},\delta} = [a, b] \times [c, d] \subset \mathbb{R}^2$, covering the support of $\rho_{\mathbb{S}}(x, y)$ up to a certain precision δ, such that

$$\iint_{\Omega_{\mathbb{S},\delta}} \rho_{\mathbb{S}}(x, y) dx dy \geq 1 - \delta.$$

Then, we consider two uniform partitions of the intervals

$$a = p_0 \leq p_1 \leq \ldots, \leq p_M = b \text{ and } c = q_0 \leq q_1 \leq \ldots, \leq q_M = d.$$

Finally, we express $\Omega_{\mathbb{S},\delta}$ from

$$\Omega_{\mathbb{S},\delta} = \bigcup_{i=0}^{M-1} \bigcup_{j=0}^{M-1} [p_i, p_{i+1}] \times [q_j, q_{j+1}] = \bigcup_{i=0}^{M-1} \bigcup_{j=0}^{M-1} P_{ij}.$$

The persistence image of $\mathbb{S}$ associated with the partition $\mathcal{P} = \{P_{ij}\}$ is then described by the $\mathbb{R}^{M \times M}$ matrix with elements:

$$PI(\mathbb{S}, M, \mathcal{P}, \delta)_{ij} = \left(\iint_{P_{ij}} \rho_{\mathbb{S}}(x, y) dx dy \right) \text{ for } 0 \leq i, j \leq (M - 1).$$

An example of persistence computation for a given trajectory is given in **Figure 9**.

2.4 Measuring Persistence Similarity

Consider two data sets $\mathbb{S}_u$ and $\mathbb{S}_v$ representing two trajectories. A matching between two persistence diagrams, $\mathcal{PD}(\mathbb{S}_u)$ and $\mathcal{PD}(\mathbb{S}_v)$, is a map ψ, that reads:

$$\psi: \mathcal{PD}(\mathbb{S}_u) \to \mathcal{PD}(\mathbb{S}_v),$$

such that $\forall \gamma = (b, d) \in \mathcal{PD}(\mathbb{S}_u)$,

$$\psi(\gamma) = (\psi_1(b), \psi_2(d))$$
$$= (b', d') \in \mathcal{PD}(\mathbb{S}_v).$$

The map ψ associates each feature from $\mathcal{PD}(\mathbb{S}_u)$ to a feature from $\mathcal{PD}(\mathbb{S}_v)$. The *optimal matching* between $\mathcal{PD}(\mathbb{S}_u)$ and $\mathcal{PD}(\mathbb{S}_v)$ is a matching $\hat{\psi}$

$$\hat{\psi}: \mathcal{PD}(\mathbb{S}_u) \to \mathcal{PD}(\mathbb{S}_v),$$

minimizing the transport cost $\mathcal{C}$ to move the features from $\mathcal{PD}(\mathbb{S}_u)$ to $\mathcal{PD}(\mathbb{S}_v)$:

$$\mathcal{C}_{\min} = \sum_{\gamma \in \mathcal{PD}(\mathbb{S}_u)} \|\gamma - \hat{\psi}(\gamma)\|_2$$
$$= \sum_{(b,d) \in \mathcal{PD}(\mathbb{S}_u)} \|(b - \hat{\psi}_1(b), d - \hat{\psi}_2(d))\|_2$$
$$= \sum_{(b,d) \in \mathcal{PD}_k(\mathbb{S}_u)} \sqrt{(b - \hat{\psi}_1(b))^2 + (d - \hat{\psi}_2(d))^2}.$$

Then, to measure the degree of similarity between two trajectories $\mathbb{S}_u$ and $\mathbb{S}_v$ we consider the *Wasserstein distance* (Villani, 2006; Peyré and Cuturi, 2019) between $\mathcal{PD}(\mathbb{S}_u)$ and $\mathcal{PD}(\mathbb{S}_v)$

$$W(\mathcal{PD}(\mathbb{S}_u), \mathcal{PD}(\mathbb{S}_v)) = \sum_{(b,d) \in \mathcal{PD}(\mathbb{S}_u)} \sqrt{(b - \hat{\psi}_1(b))^2 + (d - \hat{\psi}_2(d))^2},$$

where $\hat{\psi}$ is the optimal matching between $\mathcal{PD}(\mathbb{S}_u)$ and $\mathcal{PD}(\mathbb{S}_v)$.

An example of matching between the persistence diagrams of two trajectories is given in **Figure 10**.

2.5 Barycentres of Persistence Diagrams

Consider now a collection $\mathbb{S}_1 \ldots \mathbb{S}_n$ of trajectories with their associated diagrams $\mathcal{PD}_1 \ldots \mathcal{PD}_n$.

Since the space of persistence diagrams equipped with the Wasserstein distance, the *Wasserstein space*, is not a linear space, the notion of barycentres (Agueh and Carlier, 2011) can be extended for the persistence diagrams using the so-called *Frechet mean* (Turner et al., 2014), which always exists in the context of averaging finitely many diagrams.

The Frechet mean of $\mathcal{PD}_1 \ldots \mathcal{PD}_n$ is any diagram minimizing the map

$$\mathcal{E}: \mu \mapsto \sum_{i=1}^{n} W(\mu, \mathcal{PD}_i)^2.$$

The computation of the barycentre μ has proven to be challenging, and multiple approaches can be used, such as the Sinkhorn algorithm (Cuturi and Doucet, 2014). We will use the one based on the Hungarian algorithm presented in Turner et al. (2014) and consider Partial Optimal Matchings (Divol and Lacombe, 2020), as the diagrams may not be of the same size. In this case, points from the diagonal are matched with the remaining (exceeding) points.

In our case, we estimate the barycentres of a finite family of persistence diagrams, taking a Lagrangian approach by tracking the individual points of the diagrams. Given a collection $\mathcal{PD}_1 \ldots \mathcal{PD}_n$ of persistence diagrams, we proceed as follows:

1) Initialize the estimation μ of the barycenter at a certain diagram $\mu = \mathcal{PD}_{i_0}$.
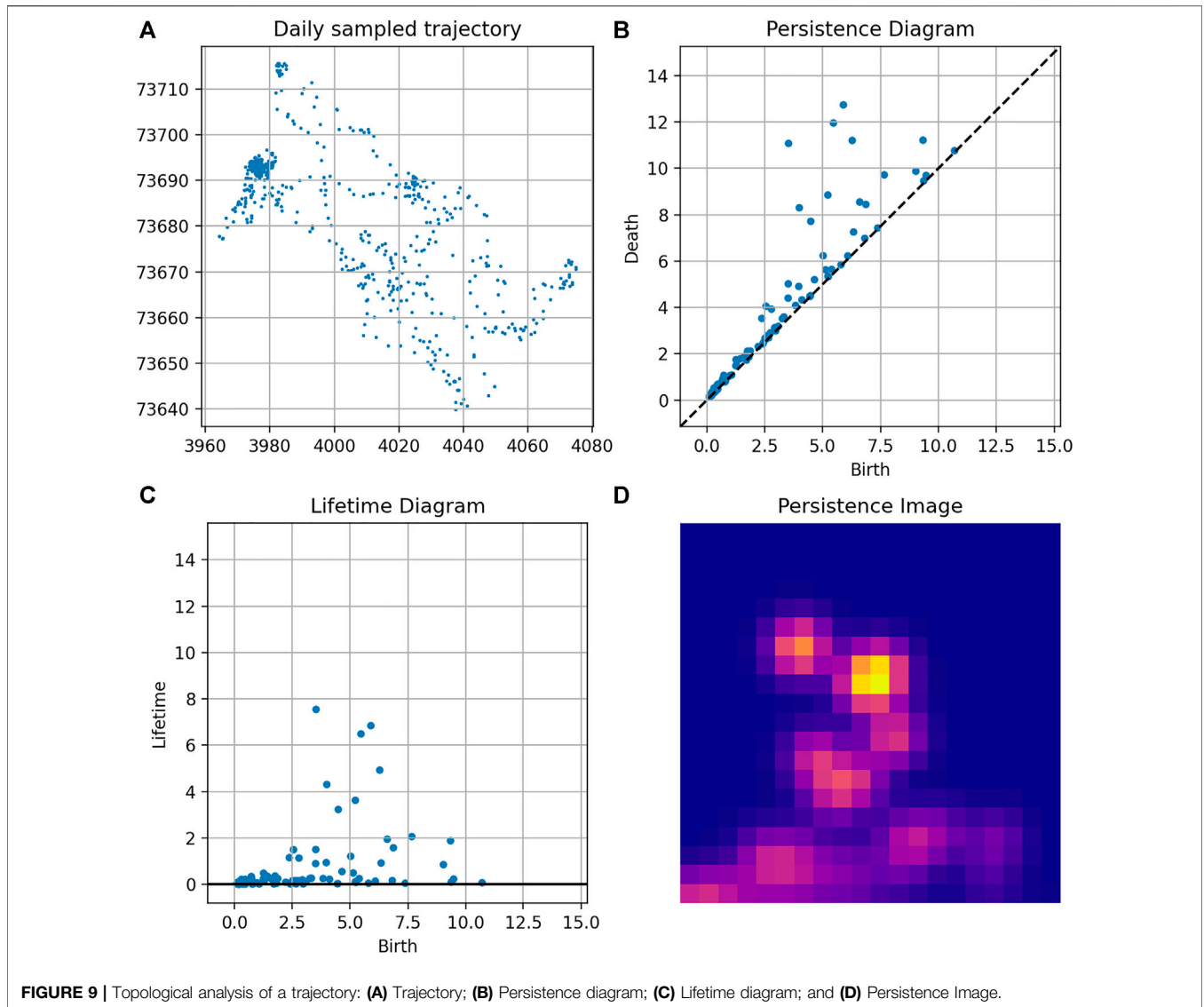
FIGURE 9 | Topological analysis of a trajectory: **(A)** Trajectory; **(B)** Persistence diagram; **(C)** Lifetime diagram; and **(D)** Persistence Image.

2) Compute the optimal partial matchings $\psi_1 \ldots \psi_n$, between μ and $\mathcal{PD}_1 \ldots \mathcal{PD}_n$ respectively.

3) Compute the updated barycentre $\hat{\mu}$, by averaging the transport of each point in the barycentre μ

$$\hat{\mu} = \left\{ y = \frac{1}{n} \sum_{i=1}^{n} \psi_i(x), x \in \mu \right\}.$$

4. If $\hat{\mu}$ minimizes $\mathcal{E}$, return $\hat{\mu}$. Otherwise, update $\mu = \hat{\mu}$ and go back to 2.

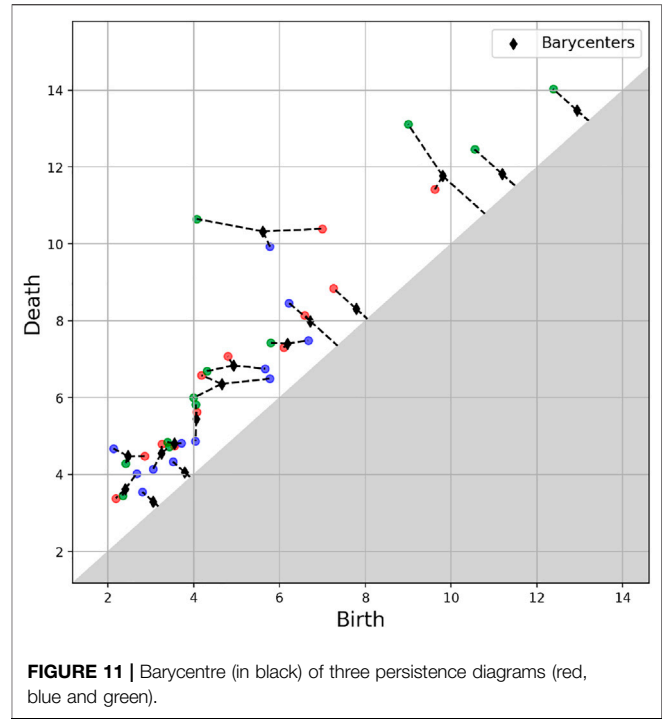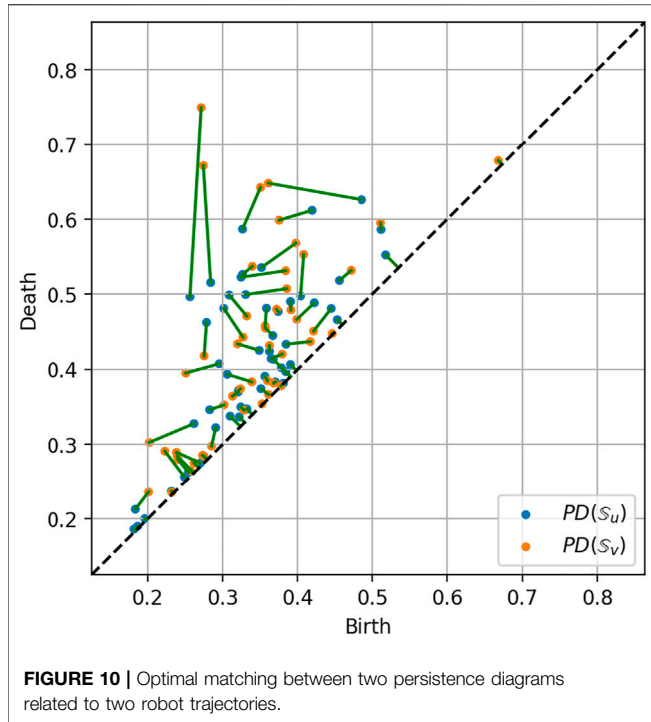An example of a barycentre of three persistence diagrams is given in **Figure 11**.

2.6 Classification

Image classification is a procedure that is used to automatically categorize images into classes by assigning to each image a label representative of its class. A supervised classification algorithm requires a training

sample for each class, that is, a collection of data points whose class of interest is known. Labels are assigned to each class of interest. The classification problem applied to a new observation (data) is thus based on how close a new point is to each training sample. The Euclidean distance is the most common metrics used in low-dimensional datasets. The training samples are representative of the known classes of interest to the analyst. In order to classify the persistence images, we considered the logistic regression algorithm.

Consider a training set $(\mathcal{X}_i)_{i=1}^{n}$ of flattened persistence images, i.e., $M \times M$-component vectors, computed from a set $(\mathbb{S}_i)_{i=1}^{n}$ of trajectories as described earlier. Associated is a list $(\mathcal{Y}_i)_{i=1}^{n}$ of binary labels $\{0, 1\}$, describing whether an image $\mathcal{X}_i$ is in the interest set or not.

The training of the $\mathcal{L}_2$-*penalized logistic regression binary classifier* is then the minimization of a cost function as described in the following optimization problem:

FIGURE 10 | Optimal matching between two persistence diagrams related to two robot trajectories.

FIGURE 11 | Barycentre (in black) of three persistence diagrams (red, blue and green).

$$\min_{\omega,c} \frac{1}{2}\omega^T\omega + C\sum_{i=0}^{p}\log\left(\exp\left(\mathcal{Y}_i\left(\mathcal{X}_i^T\omega + c\right)\right) + 1\right).$$

Here ω are the weights we optimize over, c a Bernoulli mean vector of the weights, and C an inverse regularization parameter. Once trained, the model is evaluated on a unseen set of flattened persistence images. The metric used for the model evaluation is the *Accuracy Score* defined in the next section.

2.7 Model Evaluation

Evaluating a classification model consists of determining how often labels are correctly or wrongly predicted for the testing samples. In other words, it is counting how many times a sample is correctly or wrongly labelled into a particular class. We distinguish four qualities:

- TP (True Positive): the correct prediction of a sample into a class;
- TN (True Negative): the correct prediction of a sample out of a class;
- FP (False Positive): the incorrect prediction of a sample into a class;
- FN (False Negative): the incorrect prediction of a sample out of class.

These quantities are involved in the definition of the model performances estimator, the AccuracyScore (A). It is giver by the ratio of the number of correct predictions over the number of all samples, expressed by

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

3 RESULTS

We recall the overall workflow of the proposed numerical procedure, summarized in **Figure 12**:

1) We start by preprocessing and cleaning the raw data gathered from the robot sensors
2) The data frequency is homogenized (minute data), and the pathways over each patch (or location) are daily sampled. We obtain 240 daily trajectories.
3) We compute the Rips filtration as described earlier, and then the persistence diagrams.
4) The persistence images are computed for each daily pathway, and used as inputs for the classification.
5) The persistence diagrams are also used to compute barycentres over given periods, and compute the Wasserstein distance between diagrams.

We also describe the two main classifications tasks at hand:

1) Predict the patch in which the robot is located every day, using the 240 daily persistence images as inputs. This is a binary classification task, associating to each daily persistent image either the label 1 if the robot is in the target patch, or the label 0 if the robot is in any other patch. The goal is to show the capacity to differentiate between pathways coming from different patches based on their topological signature.
2) Predict whether a maintenance has been operated on a robot or not, using the 50 daily persistence images associated to the same target patch as inputs. The maintenance dates are known. This is also a binary classification task, associating to each daily persistent image either the label 1 if the

FIGURE 12 | Workflow of the numerical procedure.
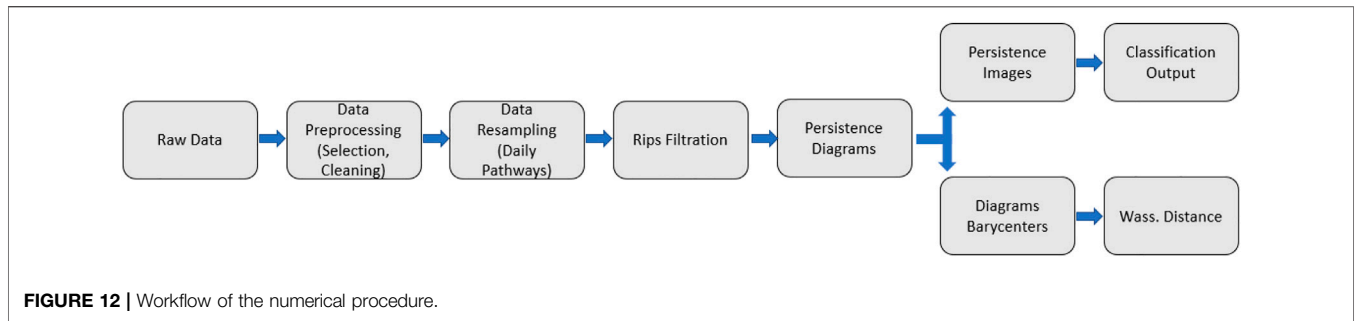
Classification task	Number of samples	0 − 1 labels frequencies	Train − Test split	Accuracy score
Predict if daily pathway is in target patch or not	240	52% - 48%	65% - 35%	80%
Predict if daily pathway is before or after maintenance operation	50	50% - 50%	65% - 35%	90%

FIGURE 13 | Classification tasks summary.

maintenance has occurred before that day, or the label 0 if not. The goal is to show the capacity to differentiate between functioning states of the robots, before and after a maintenance operation, while controlling the other factors (pathways sampled from the same patch).

The choice of the target patch #3 and has been motivated by two considerations:

- Have two equilibrated classes in the first classification task: the patch #3 is by far the one where the robot has spent most of time in operation, and it involves a very equilibrated 126/114 distribution of the two classes (0 and 1) after data cleaning.
- Also within the same patch it was possible to have 50 days time window around a maintenance operation where the robot stayed within the patch, with 25 days before and 25 after the maintenance operation on the robot, resulting in two perfectly balanced classes for the classification.
- We also note that the train-test split (65–35%) in both tasks has been done with stratification: the proportion of each class in the dataset is preserved when splitting the data (roughly 50–50%).

Both classification tasks and associated results are summarized in **Figure 13**.

3.1 Determination of the Patch in Which the Robot Is Located

We first want to predict whether a robot is in a certain patch. For that purpose we choose one parcel as a target, and train a classification model as described in **Section 2.6**. The complete dataset consists of daily trajectories for 240 days. For each day a persistence image is computed, which will then be used as input for the model (a sample is depicted in **Figure 9**). The samples are labelled according to the target patch (patch #3): 1 if the robot is in the target patch (114 samples), and 0 otherwise (126 samples). The dataset is split into 65% for training and 35% for testing. The proposed classifier achieves an 80% accuracy score in predicting the patch at which the robot is, based on the persistence images.

3.2 Maintenance Prediction

Then, we consider daily trajectories in the same patch (patch #3), consisting of 50 samples. For each day, a persistence image is computed, that will be used as input in the classifier. The periods considered here are the ones in between two consecutive maintenance operations of the robot. The samples are labelled 0 if they are associated to a day before the maintenance date (25 samples), 1 otherwise (25 samples). The dataset is split into 65% for training and 35% for testing. The model achieves a 90% accuracy score predicting the period associated to the sampled trajectories. The model high accuracy proves that the topological descriptors have enough information about the pathways to allow detecting patterns related to maintenance events, fact that could be used for predictive maintenance purposes.

3.3 A Time Varying Measure

Figure 14 depicts the Wasserstein distance between the persistence diagrams for consecutive daily trajectories, with the maintenance operation emphasized in red, whereas
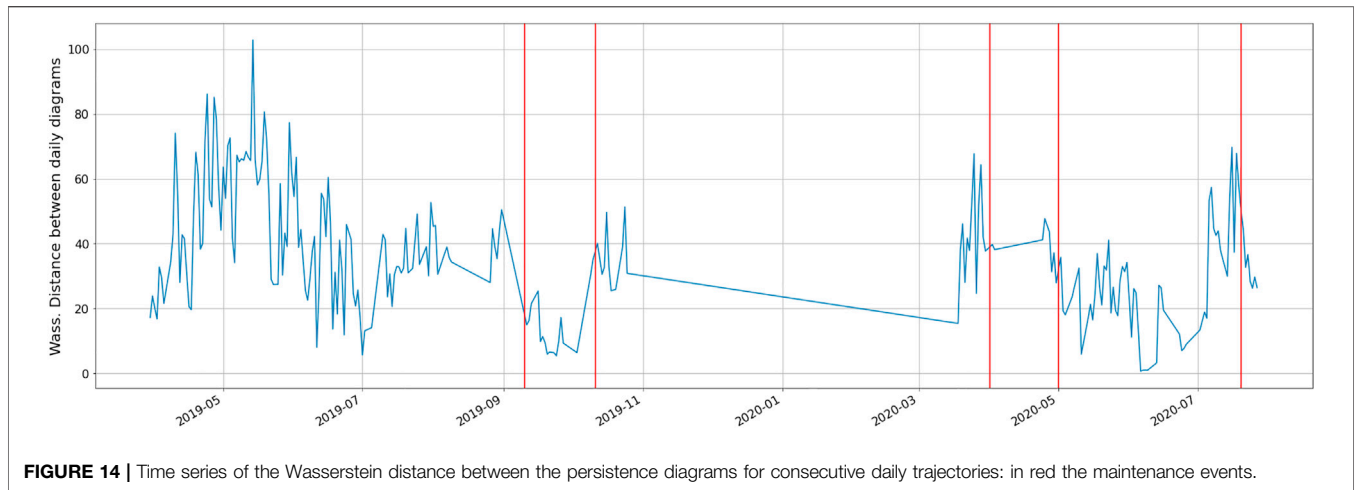
FIGURE 14 | Time series of the Wasserstein distance between the persistence diagrams for consecutive daily trajectories: in red the maintenance events.
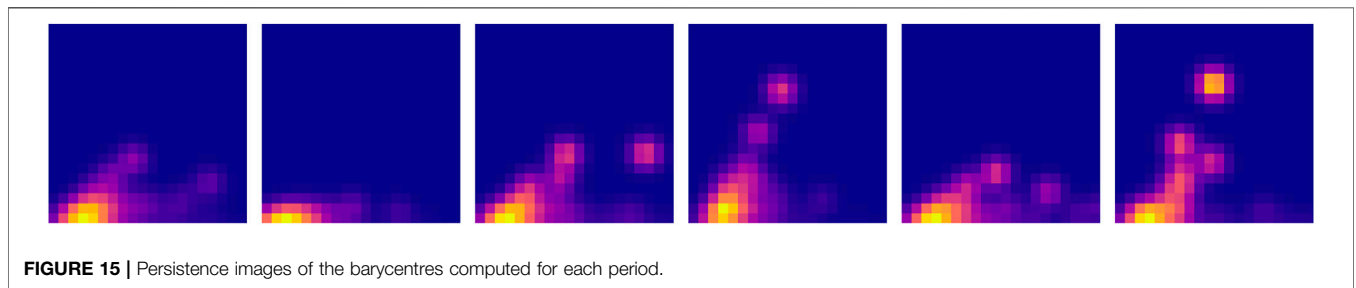

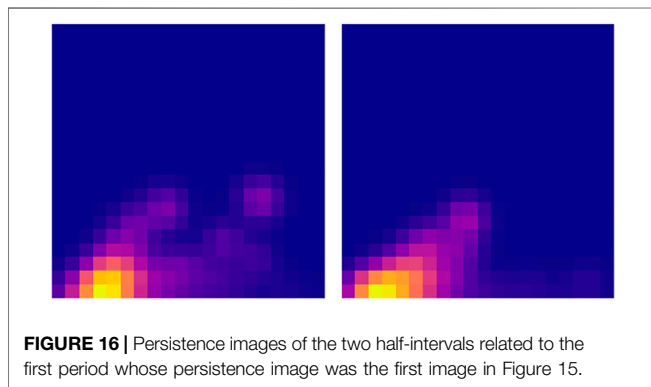**FIGURE 15 |** Persistence images of the barycentres computed for each period.

FIGURE 16 | Persistence images of the two half-intervals related to the first period whose persistence image was the first image in Figure 15.

Figure 15 shows the barycentres of each period between consecutive maintenance operations. As it can be noticed from the persistence images in **Figure 15**, maintenance operations affect the topology of the trajectory, as it was expected from the fact that classification performs successfully as just reported.

To better support our hypothesis about the effect of maintenance on the trajectory topology, we consider the first operation interval, the one before the first maintenance, that correspond to the first persistence image in **Figure 15** (left), and divide it in two parts with identical length. Then, the associated barycentres in both half intervals are obtained. Both are represented in **Figure 16**.

As it can be noticed, both of them resemble very much to the one associated to the whole interval (the first picture in **Figure 15**), with a Wasserstein distance of 23.5 and 24.1 (computed on the associated diagrams). Conversely, the distance of both to the second period is much higher (46.5 and 26.1).

Finally, we can compute the distance of the 5 latter periods to the first one (**Figure 15**) and we have: 36.7, 31.0, 34.2, 33.3, 35.2, so significantly higher than the first period compared to its own two halves.

These results support again our assumption on the effect of maintenance on the trajectory topology.

4 CONCLUSION

The characterization of the trajectories followed by the robot based on the geographical location proves to be a reliable method to differentiate between different environments affecting the robot motion. Then, over a single patch, the classification was proved being efficient to detect the changes in the robot signature related to maintenance events.

The proposed topology-based framework for sampled trajectories seems a very pertinent, powerful and intrinsic way of quantifying, characterizing and analysing the topological and geometrical nature of the robot's pathways. The strength of the framework relies on both the topology description of the

trajectory at multiple scales, and the use of metrics features that can be combined with machine learning.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

REFERENCES

Agueh, M., and Carlier, G. (2011). Barycenters in the Wasserstein Space. *SIAM J. Math. Anal.* 43, 904–924. doi:10.1137/100805741

Alatise, M. B., and Hancke, G. P. (2020). A Review on Challenges of Autonomous mobile Robot and Sensor Fusion Methods. *IEEE Access* 8, 39830–39846. doi:10.1109/access.2020.2975643

Avanço, R. H., Navarro, H. A., Nabarrete, A., Balthazar, J. M., and Tusset, A. M. (2016). *Chaotic Behavior in the Double Pendulum under Parametric Resonance.* American Society of Mechanical Engineers (ASME). doi:10.1115/imece2016-65711

Breiman, L. (2001). Random Forests. *Machine Learn.* 45, 5–32. doi:10.1023/a:1010933404324

Cristianini, N., and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* New York: Cambridge University Press.

Cuturi, M., and Doucet, A. (2014). "Fast Computation of Wasserstein Barycenters,". Editors E. P. Xing and T. Jebara (Bejing, China: of Proceedings of Machine Learning Research), 685–693.Proceedings of the 31st International Conference on Machine Learning

Divol, V., and Lacombe, T. (2020). Understanding the Topology and the Geometry of the Space of Persistence Diagrams via Optimal Partial Transport. *J. Appl. Comput. Topology* 5, 1–53. doi:10.1007/s41468-020-00061-z

Frahi, T., Argerich, C., Yun, M., Falco, A., and Barasinski, A. (2020). Tape Surfaces Characterization with Persistence Images. *AIMS Mater. Sci.* 7, 364–380. doi:10.3934/matersci.2020.4.364

Frahi, T., Chinesta, F., Falcó, A., Badias, A., Cueto, E., and Choi, H. Y. (2021a). Empowering Advanced Driver-Assistance Systems from Topological Data Analysis. *Mathematics* 9, 634. doi:10.3390/math9060634

Frahi, T., Falco, A., Mau, B. V., Duval, J. L., and Chinesta, F. (2021b). Empowering Advanced Parametric Modes Clustering from Topological Data Analysis. *Appl. Sci.* 11, 6554. doi:10.3390/app11146554

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning.* Cambridge: MIT Press.

Gupta, M. K., Bansal, K., and Singh, A. K. (2014). Mass and Length Dependent Chaotic Behavior of a Double Pendulum. *IFAC Proc. Volumes* 47, 297–301. doi:10.3182/20140313-3-in-3024.00071

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Berlin: Springer.

Ibañez, R., Abisset-Chavanne, E., Cueto, E., Ammar, A., Duval, J. L., and Chinesta, F. (2019). Some Applications of Compressed Sensing in Computational Mechanics: Model Order Reduction, Manifold Learning, Data-Driven Applications and Nonlinear Dimensionality Reduction. *Comput. Mech.* 64, 1259–1271. doi:10.1007/s00466-019-01703-5

Kavraki, L., Kolountzakis, M., and Latombe, J.-C. (1998). Analysis of Probabilistic Roadmaps for Path Planning. *IEEE Trans. Robotics Automation* 14, 166–171. doi:10.1109/70.660866

Kirkwood, C. (2022). Decision Tree Primer. Available at: https://www.public.asu.edu/kirkwood/DAStuff/refs/decisiontrees/index.html.

Lhermitte, S., Verbesselt, J., Verstraeten, W., and Coppin, P. (2011). A Comparison of Time Series Similarity Measures for Classification and Change Detection of Ecosystem Dynamics. *Remote Sensing Environ.* 115, 3129–3152. doi:10.1016/j.rse.2011.06.020

AUTHOR CONTRIBUTIONS

TF and AS performed research and coded the programs for the examples. MG, AC, and XB identified the problem, guided the research and revised the manuscript. AF and EC supervised the research and developed the methodology based on TDA. FC directed and coordinated the research and obtained the funding.

MacKay, D. J. (2003). *Information Theory, Inference, and Learning Algorithms.* Cambridge: Cambridge University Press.

MacQueen, J. (1967). "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability* (University of California Press), 281–297.

Martín, C. A., Pinillo, R. I., Barasinski, A., and Chinesta, F. (2019). Code2vect: An Efficient Heterogenous Data Classifier and Nonlinear Regression Technique. *Comptes Rendus Mÿcanique*, 347, 754–761. doi:10.1016/j.crme.2019.11.002

Mohanty, P., and Parhi, D. (2013). Controlling the Motion of an Autonomous mobile Robot Using Various Techniques: a Review. *J. Adv. Mech. Eng.* 1, 24–39. doi:10.7726/jame.2013.1003

Müller, M. (2007). *Information Retrieval for Music and Motion.* Berlin: Springer.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective.* Cambridge: MIT Press.

Peyré, G., and Cuturi, M. (2019). Computational Optimal Transport: With Applications to Data Science. *Foundations Trends® Machine Learn.* 11, 355–607. doi:10.1561/2200000073

Senin, P. (2008). *Dynamic Time Warping Algorithm reviewTech. Rep.* Honolulu, USA: University of Hawaii at Manoa.

Shalal, N., Low, T., Mccarthy, C., and Hancock, N. H. (2013). "A Review of Autonomous Navigation Systems in Agricultural Environments," in *SEAg 2013: Innovative Agricultural Technologies for a Sustainable Future* (Barton, Western Australia: Society for Engineering in Agriculture, 22–25. Sept 2013.

Torquato, S. (2002). Statistical Description of Microstructures. *Annu. Rev. Mater. Res.* 32, 77–111. doi:10.1146/annurev.matsci.32.110101.155324

Turner, K., Mileyko, Y., Mukherjee, S., and Harer, J. (2014). Fréchet Means for Distributions of Persistence Diagrams. *Discrete Comput. Geometry* 52, 44–70. doi:10.1007/s00454-014-9604-7

Villani, C. (2006). *Optimal Transport, Old and New.* Berlin: Springer.

Yun, M., Argerich, C., Cueto, E., Duval, J. L., and Chinesta, F. (2020). Nonlinear Regression Operating on Microstructures Described from Topological Data Analysis for the Real-Time Prediction of Effective Properties. *Materials* 13, 2335. doi:10.3390/ma13102335