# analyse topologique des données dans la mécanique numérique

Tarek Frahi

HESAM UNIVERSITÉ

Arts et Métiers Sciences et Technologies

**ÉCOLE DOCTORALE SCIENCES DES MÉTIERS DE L'INGÉNIEUR**
**Laboratoire PIMM – Campus de Paris**

# THÈSE

présentée par : **Tarek FRAHI**

soutenue le : **19 octobre 2021**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée à : **École Nationale Supérieure d'Arts et Métiers**

Spécialité : **Mathématiques Appliquées**

# ANALYSE TOPOLOGIQUE DES DONNÉES DANS LA MÉCANIQUE NUMÉRIQUE

**THÈSE dirigée par :**
**M. Francisco CHINESTA**
**Et M. Antonio FALCO**

**Co-encadrée par :**
**M. Jean-Louis Duval**

**Jury**

| | | |
|---|---|---|
| **M. Aziz HADOUNI**, | Professeur, Université de La Rochelle | Président |
| **M. Mejdi AZAIEZ**, | Professeur, INP Bordeaux | Rapporteur |
| **M. Tomas CHACON**, | Professeur, Université de Séville | Rapporteur |
| **M. Bertrand MICHEL**, | Professeur, Ecole Centrale de Nantes | Examinateur |
| **Mme Susana FERREIRO**, | Docteure, Tekniker | Examinatrice |
| **M. Francisco CHINESTA**, | Professeur, ENSAM Paris | Examinateur |
| **M. Antonio FALCO**, | Professeur, Université CEU Valence | Examinateur |
| **M. Jean-Louis DUVAL**, | Docteur | Examinateur |
| **M. Yves TOURBIER**, | Docteur, Renault | Invité |

THÈSE

**Tarek FRAHI**

# ANALYSE TOPOLOGIQUE DES DONNÉES DANS LA MÉCANIQUE NUMÉRIQUE


## Résumé

La présente thèse à pour sujet la topologie numérique pour les systèmes mécaniques. Nous traitons de l'analyse, de la caractérisation et de l'exploitation des données à fort contenu topologique, tels que les déformations mécaniques, les microstructures, les séries temporelles et les trajectoires d'un système dynamique.

Ces données contiennent souvent des informations hétérogènes, difficiles à mesurer, et qui ne se prêtent pas aux approches et métriques classiques. D'où, la nécessité d'avoir une approche générale avec des propriétés d'invariance, et qui permet d'extraire l'information topologique et géométrique des données, de la mesurer, et de l'utiliser sous forme de descripteurs topologiques.

Ainsi, notre approche est d'adapter l'utilisation de l'homologie et de la persistance topologique aux problématiques physiques et d'ingénierie. Cette approche est purement basée sur les données, et consiste en l'extraction de descripteurs robustes, au moyen du transport optimal notamment, qui résument l'information contenue dans le système physique, dans un graphe, un diagramme, ou une image. Ces descripteurs sont ensuite utilisés dans des algorithmes d'apprentissage, pour le regroupement, la classification et la régression.

Nous présenterons quatre applications publiées de notre méthodologie. La première consiste à identifier les modes de déformations d'une structure métallique à partir de la déformation du maillage associé. La seconde est la caractérisation d'échantillons de surfaces rugueuses de polymères pour prédire des grandeurs d'intérêt. La troisième est la prédiction de l'état d'un conducteur de voiture à partir des séries temporelles associées au mouvement de la tête, et qui est dû aux vibrations induites par la route. La quatrième est la signature topologique extraite des données réelles de trajectoires d'un robot autonome pour améliorer la maintenance prédictive.


Mots-clés : Analyse Topologique des Données, Persistance Homologique, Transport Optimal, Mécanique Numérique, Apprentissage Automatique.

# TOPOLOGICAL DATA ANALYSIS IN COMPUTATIONAL MECHANICS


## Abstract

The present thesis focuses on the applications of numerical topology for mechanical systems. We will deal with the analysis, the characterization, and the exploration of data with high topological content, such as mechanical deformations, micro-structures, times series, and dynamical systems trajectories.

This data often contains heterogeneous information, difficult to measure, and is not suitable for classical approaches and metric. Hence, it is necessary to have a general approach with invariance properties, allowing to extract the topological and geometrical information of the data, measure it, then use it as topological descriptors.

Therefore, our approach is to adapt the use of homology and topological persistence to physics and engineering issues. This approach is purely data driven, and consists in the computation of robust descriptors, relying on optimal transport among others, to summarize the information contained within the physical system into a graph, a diagram, or an image. These descriptors are then used for machine learning, in clustering, classification and regression.

We will present four published applications of our methodology. The first one is the identification of deformation modes of a metallic structure, from the topology of the associated deformed mesh. The second is the characterization of rough polymers surfaces profiles in order to predict quantities of interest. The third is the prediction of a driver state from the time series associated to the head movement, which is induced by vibrations of the road. The fourth is the topological signature extracted from real data of an autonomous robot trajectories, to improve its predictive maintenance.



Keywords: Topological Data Analysis, Persistent Homology, Optimal Transport, Computational Mechanics, Machine Learning.

# Résumé de la thèse en français

L'analyse de données topologiques est l'étude des ensembles de données en fonction de leurs propriétés et invariants topologiques. Elle fournit un cadre robuste et général pour analyser les données sans avoir besoin d'une variété analytique ambiante. Elle s'appuie sur la caractérisation des formes de données, des boucles et des trous, à l'aide d'outils de topologie algébrique et de géométrie numérique. Pour analyser la forme des données sans avoir besoin d'un plongement dans un espace métrique, l'approche consiste à créer des objets géométriques appelés filtrations. En recouvrant les données de simplex et de sphères de dimensions et d'échelles différentes, une approximation du nuage de données est obtenue. L'idée est alors de voir comment les traits simpliciaux de différentes dimensions (points, segments, triangles, tétraèdres...) se répartissent sur les échelles, et comment ils apparaissent et disparaissent (formant des boucles et des trous). Pour cela, les groupes d'homologie sont introduits. C'est une représentation algébrique des relations entre les éléments simpliciaux. Ils sont regroupés en classes représentant des chaînes de simplexes, que nous appelons entités. L'homologie persistante permet alors de suivre comment ces entités sont distribuées sur les échelles et dimensions, associant à chacune une échelle de naissance et une échelle de mort, représentant respectivement les échelle d'apparition et de disparition dans la filtration.

Pour résumer les informations de l'homologie persistante, les entités sont regroupées selon leurs dimensions et leur durée de vie, suivant la représentation des groupes d'homologie. Ils peuvent ensuite être résumés à l'aide d'un ensemble de descripteurs tels que les diagrammes de persistance et les codes-barres. C'est une représentation en coordonnées 2D de la naissance et de la mort des entités. Des représentations plus avancées peuvent alors être calculées telles que les images de persistance, les noyaux, l'entropie et les paysages.

Le but de ces descripteurs de persistance est d'avoir des objets mesurables, qui peuvent être utilisés pour calculer des statistiques. Le transport optimal est un cadre idéal pour ce faire, permettant de définir des métriques robustes sur les représentations de persistance. Il est ensuite possible d'incorporer ces statistiques et représentations calculées dans des modèles d'apprentissage automatique décrivant le système physique sous-jacent.

Le besoin d'une approche générale axée sur les données est devenu plus aigu avec la multiplication des sources de données, le volume toujours croissant des données disponibles et la nature interdisciplinaire des nouveaux problèmes d'ingénierie.

Notre méthodologie présentée ici repose sur un cadre général de calcul d'homologie persistante. Nous considérons le nuage de données comme une discrétisation finie du système physique, et calculons la filtration associée la plus adaptée, prenant en compte la nature des données, la dimensionnalité et les quantités d'intérêt visées. Nous procédons ensuite au calcul des descripteurs de persistance, ainsi que des métriques, des statistiques et du transport optimal. Le résultat peut ensuite être intégré dans un processus d'apprentissage, pour le clustering, la classification et la régression.

Nous aborderons quatre applications de l'analyse de données topologiques :

- Regroupement expert de modes paramétriques [3] au chapitre 2
- Caractérisation des surfaces rugeuses [1] au chapitre 3
- Systèmes avancés d'aide à la conduite [2] au chapitre 4
- Surveillance et anticipation des états de fonctionnement de robots [4] au chapitre 5

## Regroupement expert de modes paramétriques

L'analyse modale est largement utilisée pour traiter le NVH - Bruit, Vibration et Dureté - dans l'ingénierie automobile. Les modes dits principaux constituent une base orthogonale, obtenue à partir des vecteurs propres liés au problème dynamique. Lorsque cette base est utilisée pour exprimer le champ de déplacement d'un problème dynamique, les équations du modèle deviennent découplées. De plus, une base réduite peut être définie en fonction de la grandeur des valeurs propres, conduisant à un modèle réduit non couplé, particulièrement intéressant lors de la résolution de grands systèmes dynamiques. Cependant, l'ingénierie recherche des conceptions optimales et se concentre donc sur des conceptions paramétriques nécessitant la solution efficace de modèles dynamiques paramétriques. La résolution de problèmes propres paramétrés reste une question délicate, et par conséquent, les approches non intrusives sont privilégiées. Dans ce cadre, une base réduite constituée des modes propres les plus significatifs est retenue pour chaque choix des paramètres du modèle considéré. Ensuite, on est tenté de créer une base réduite paramétrique, en exprimant simplement la base réduite de façon paramétrique en utilisant une technique de régression appropriée. Cependant, un problème demeure, qui limite l'application directe de l'approche qui vient d'être évoquée, qui est celui de l'ordonnancement des bases. Afin d'ordonner les modes avant de les interpoler, différentes techniques ont été proposées dans le passé, le critère d'assurance modale -MAC- étant l'un des plus utilisés. Nous proposons une technique alternative qui, au lieu de fonctionner au niveau des modes propres, classe les modes par rapport aux formes des structures déformées que les modes propres induisent.

La principale limitation de l'analyse modale est le manque de validité de la base réduite dans le cas des modèles paramétriques. Pour les systèmes dynamiques paramétrés, les matrices du modèle dépendront de ces paramètres, et la résolution des problèmes propres paramétriques reste un délicate.

Lorsque l'on ne s'intéresse pas vraiment au régime transitoire, mais bien plus au régime forcé, l'analyse harmonique représente une voie précieuse. La Décomposition Propre Généralisée—PGD— permet de considérer la fréquence comme un extra-paramètre du modèle ainsi que d'aborder l'amortissement général (non proportionnel) et la dynamique non-linéaire, sous la stricte contrainte de temps réel, avec même l'inclusion des paramètres du modèle comme coordonnées supplémentaires [18, 17, 20].

Cependant, certaines applications nécessitent des réponses transitoires précises, et dans ce cas, la formulation et la solution du problème dynamique dans le domaine temporel sont conservées. À cette fin, l'une des techniques les plus utilisées est l'analyse modale. Outre les avantages dans l'intégration temporelle, dus au découplage dynamique du système, les modes propres bénéficient d'une interprétation physique, d'un grand intérêt pour le concepteur ou l'analyste structurel. Néanmoins, lorsque l'on considère des modèles paramétriques comme c'est toujours le cas lors de la phase de conception, lorsque le matériau et la géométrie ne sont pas totalement définis, les modes dynamiques dépendent de ces paramètres comme discuté précédemment. Disposer d'un modèle de substitution exprimant l'évolution paramétrique des modes propres est d'un grand intérêt. La construction de ces modèles de substitution est aujourd'hui assez aboutie, en utilisant les régressions non linéaires usuelles et avancées [19], la dernière utilisant la parcimonie et la régularisation appropriée pour opérer dans des contextes de grande dimension, tout en gardant le plus réduit possible le nombre de données ( solution des problèmes propres), et conduisant à des régressions (non linéaires) suffisamment riches tout en évitant le surapprentissage. Ici, le plus délicat n'est pas la construction de la régression, mais le fait d'ordonner les différents modes propres impliqués dans la base modale pour chaque choix de paramètres, afin de créer des clusters $N$ (ou moins dans le cas réduit), et de mettre dans chacun un mode de chaque base modale, de sorte que les modes de chaque cluster restent proches (dans une certaine métrique). Le problème principal reste la métrique à utiliser pour accomplir avec succès et efficacité un tel clustering. En général un tel clustering est effectué en opérant au niveau des modes propres, dans l'espace vectoriel associé, en utilisant par exemple le critère d'assurance modale—MAC— [52] qui procède à la comparaison des modes résultant de chaque problème propre , en utilisant le produit scalaire habituel (les modes similaires à un mode donné doivent rester assez colinéaires).

Lorsqu'on opère dans des espaces paramétriques de grande dimension, peu échantillonnés, les matrices impliquées dans le problème propre résultant peuvent beaucoup varier d'un choix de paramètres à l'autre, et par conséquent le critère de produit scalaire à la base du MAC peut échouer. D'autre part, le fait de procéder dans un espace vectoriel nécessite d'aborder soigneusement l'expression des différents modes en considérant le même référentiel pour tous les systèmes mécaniques analysés.

Pour atténuer ces difficultés, nous proposons une technique alternative qui, au lieu de fonctionner au niveau des modes propres, classe les modes par rapport à la forme des structures déformées que les modes propres produisent, en tirant parti de la propriété d'invariance de la topologie. Ainsi, nous utilisons une métrique capable de comparer des formes, plus qu'une métrique pour comparer les vecteurs (modes propres) qui ont produit ces formes, la dernière étant plus intrinsèque et héritant des caractéristiques d'invariance. De plus, dans le cas présent, les modes propres sont hétérogènes dans le sens où ils impliquent des déplacements et des rotations, alors que les surfaces déformées associées sont purement géométriques.

Dans le chapitre 2, nous abordons la classification d'une série de bases modales liées aux modes propres d'une structure mince équipée d'un maillage constitué

d'éléments de coque, avec des degrés de liberté de déplacement et de rotation à chaque nœud du maillage. L'épaisseur de la partie structurelle varie, avec son effet conséquent sur les matrices de masse et de rigidité (l'amortissement est supposé proportionnel) et par conséquent sur les valeurs propres et vecteurs propres, les premiers définissant le nombre de modes à retenir dans la base réduite. Dans la présente étude, les six modes rigides représentant l'ensemble de la structure translation (trois modes) et rotation (trois modes) seront écartés et parmi les paires restantes valeur propre-vecteur propre, les six vecteurs propres les plus pertinents (correspondant aux six valeurs propres les plus élevées) retenus dans la base réduite liée à chaque choix du paramètre du modèle (l'épaisseur).

Ces six modes liés à chaque structure (liés à une valeur d'épaisseur) définissent une base réduite que l'on voudrait rendre paramétrique. Cependant, avant de construire une régression capable de définir la base réduite pour chaque choix possible du paramètre (épaisseur), il convient de classer les six modes propres de chaque base réduite associée à chaque structure, en six clusters.

Cette tâche est obligatoire pour faciliter l'interpolation dans l'espace paramétrique et aussi pour attacher un sens physique à ces modes. On pourrait imaginer que pour une épaisseur donnée le mode de déformation le plus pertinent pourrait être lié à l'extension alors que pour un autre choix d'épaisseur le mode de déformation le plus pertinent pourrait être la flexion. Dans un tel cas, on préfère créer un cluster regroupant des modes de déformation similaires, pour évaluer comment chacun d'eux dépend du paramètre d'un côté, et de l'autre pour faciliter la construction ultérieure de la base réduite modale paramétrique.

Pour effectuer un tel regroupement, nous devons utiliser une métrique appropriée pour comparer ces modes. En général, cette comparaison était traditionnellement effectuée en comparant les modes propres au sein de l'espace vectoriel auquel ils appartiennent. Dans le présent travail, comme annoncé précédemment, nous préférons appliquer le mode de déformation à la structure de référence (non déformée), c'est-à-dire appliquer le mode propre à l'emplacement des nœuds dans la structure de référence pour obtenir la structure déformée liée à chaque mode de chaque configuration de structure (épaisseur) puis regrouper les structures déformées résultantes par rapport à leur forme.

## Caractérisation des surfaces de rugeuses

Parmi les procédés de formage de composites pour la fabrication de pièces structurelles basés sur la consolidation de préformes pré-imprégnées, par exemple des feuilles, des rubans, .... le placement automatisé de ruban (ATP) apparaît comme l'une des techniques les plus intéressantes en raison de sa polyvalence et de sa - consolidation sur place, évitant ainsi l'utilisation d'autoclave. En particulier, pour obtenir la cohésion de deux couches thermoplastiques, deux conditions physiques spécifiques sont nécessaires (a) un contact quasi parfait (contact intime) et (b) une température permettant la diffusion moléculaire dans la fenêtre de temps du procédé, tout en évitant la dégradation thermique. Pour atteindre cet objectif, un ruban est

posé et progressivement collé sur le substrat constitué des rubans préalablement déposés. Du fait de la faible conductivité thermique des résines usuelles, un échauffement local intense est généralement envisagé (laser, torches à gaz...) en liaison avec une pression locale appliquée par un rouleau mobile. Ainsi, les deux principaux facteurs pour assurer le contact intime à la surface des plis sont la pression et la chaleur. Un contact intime est nécessaire pour favoriser la diffusion moléculaire. Dans ce processus, la chaleur joue un double rôle, d'une part elle améliore la mobilité moléculaire et d'autre part, la diminution de la viscosité du matériau avec l'augmentation de la température, facilite l'écoulement comprimé des aspérités chauffées situées sur les surfaces des plis sous la compression appliquée par le rouleau de consolidation.

Le modèle numérique de l'ATP a été introduit dans [6] en utilisant ce qu'on appelle la décomposition généralisée appropriée (PGD) [7, 8, 9, 10, 12]. La représentation séparée impliquée dans le PGD permet la solution 3D à haute résolution de modèles définis dans des domaines dégénérés où au moins une de leurs dimensions caractéristiques reste beaucoup plus petite que les autres et également la construction de solutions de modèles paramétriques où les paramètres du modèle sont considérés comme extra- coordonnées [11, 13].

La modélisation physique et la simulation pour le placement automatisé de bandes (ATP) ont été proposées dans [14] pour étudier l'influence des paramètres de matériaux et de processus, tandis que la modélisation de la consolidation et la régression non linéaire basée sur sPGD ont été utilisées dans [15] identifier les principaux descripteurs de surface pour une caractérisation complète des surfaces de la bande.

Dans le chapitre 3, nous revisitons d'abord la modélisation de consolidation et sa simulation haute résolution, permettant d'évaluer l'évolution temporelle du degré de contact intime –DIC– lorsque deux surfaces rugueuses sont mises en contact, chauffées et comprimées . L'écrasement des rugosités se produit principalement le long de la direction transversale (celle liée à la largeur de la bande) induite par la compression du rouleau. Ainsi, l'écoulement se produit dans la section transversale dans laquelle la surface se réduit à une courbe unidimensionnelle (le soi-disant profil de surface).

Il est bien connu au niveau expérimental que le degré de consolidation dépend fortement des caractéristiques de surface (rugosité). En particulier, les mêmes paramètres de processus appliqués à différentes surfaces produisent des degrés de contact intime très différents. Cela nous permet de penser que la topologie de surface joue un rôle important tout au long de ce processus. Cependant, la résolution des modèles basés sur la physique pour simuler la compression de la rugosité se produisant à l'interface des bandes représente un effort de calcul incompatible avec les objectifs de contrôle de processus en ligne. Une approche alternative consiste à prendre une population de différentes bandes, avec des surfaces différentes, et à simuler la consolidation pour évaluer pour chacune la progression du degré de contact intime –DIC– tout en comprimant les bandes chauffées, jusqu'à atteindre sa valeur finale à la fin. de la compression. L'objectif final est de créer une régression capable d'attribuer une valeur finale du DIC à n'importe quelle surface, permettant

un contrôle de processus en ligne. Le principal problème d'une telle approche est la description approximative de la surface, c'est-à-dire la manière la plus précise et la plus compacte de la décrire à partir de certains paramètres appropriés faciles à extraire expérimentalement, à inclure dans la régression qui vient d'être mentionnée.

Afin d'extraire une description concise et complète des surfaces rugueuses, nous utilisons une description topologique [27, 28, 29, 26] des profils de surface, pour construire des descripteurs tels que les diagrammes de persistance et les images. Ensuite, les images de persistance sont considérées pour classer les surfaces, ou comme descripteurs impliqués dans la régression les reliant au DIC final atteint dans le processus de consolidation, permettant une prise de décision en temps réel.

## Systèmes avancés d'aide à la conduite

Bien qu'il y ait eu récemment des progrès considérables dans la technologie des voitures autonomes, la conduite repose encore principalement sur des facteurs humains. Même en mode de conduite autonome, les conducteurs humains doivent souvent prendre une décision en une fraction de seconde pour éviter les accidents. Par conséquent, il est toujours de la plus haute importance de développer des systèmes capables de discerner si le conducteur humain est attentif ou non aux conditions de la route. En général, les systèmes avancés d'assistance à la conduite (ADAS) [55, 56] sont des systèmes capables d'améliorer les performances du conducteur, parmi lesquels, les limiteurs de vitesse adaptatifs, les détecteurs de piétons [57] et les régulateurs de vitesse Les contrôleurs sont parmi les systèmes les plus populaires. Les systèmes d'alerte de fatigue sont parmi les plus utiles parmi les systèmes ADAS, et le but de ce travail est de contribuer au développement d'un tel système basé sur une analyse systématique des conducteurs en conditions réelles de conduite. L'estimation de l'état du conducteur (degré d'attention à la route, fatigue, etc.) est un facteur très important pour assurer la sécurité de conduite [58, 59]. Une revue récente sur le sujet peut être trouvée dans [60].

Dans le chapitre 4, nous visons à extraire des modèles de comportement à partir des données des utilisateurs de voitures pour pouvoir estimer avec précision leur état. Nous utilisons des données expérimentales, recueillies en appliquant une stimulation mécanique à des personnes assises dans une automobile.

Notre objectif principal est d'extraire des modèles de comportement à partir des données pour nous permettre d'apprendre les facteurs les plus pertinents affectant l'attention du conducteur à la situation de la route.

Nous combinons certains outils de la théorie Morse [35] et de l'analyse de données topologiques (TDA) avec tous les concepts et méthodes associés (par exemple, nombres de Betti, persistance d'homologie, codes-barres, images de persistance, etc.) [34], la plupart d'entre eux ont été introduits et employés plus tard afin d'analyser et de classer les données expérimentales. Cela nous permet d'introduire des concepts sous forme de codes à barres, c'est-à-dire des diagrammes persistants et de durée de vie de la même manière qu'ils sont utilisés dans l'homologie persistante. Notre objectif principal est de prédire le comportement des automobilistes suivant une ap-

proche supervisée [27]. Au lieu de considérer un signal de capteur original comme la quantité d'intérêt, nous nous concentrons sur ses caractéristiques topologiques. En ce sens, le cadre proposé dans cet article nous permet de dévoiler la véritable dimensionnalité des données ou, en d'autres termes, le nombre réel de facteurs affectant la performance du conducteur. Ainsi, nous modélisons un signal de capteur en tant que système dynamique et, par conséquent, notre approche semble mieux décrire ses propriétés, ou plutôt ses variations, telles que les extrema, les modèles et l'auto-similitude, que d'autres approches.

Notons que notre approche est, dans certains sens, similaire à celle suivie par Milnor et Thurston [36] dans l'étude des propriétés combinatoires des systèmes dynamiques en combinant des outils de la théorie des automates.

## Surveillance et anticipation des états de fonctionnement de robots

Les robots autonomes suivent un certain nombre de règles introduites dans leurs contrôleurs [61, 62, 63]. Cependant, lorsqu'ils interagissent avec l'environnement, de petites variations peuvent entraîner des mouvements imprévisibles à long terme. Ce comportement est très courant en mécanique, caractérisant des systèmes présentant un chaos déterministe.

Dans le cas pratique abordé au chapitre 5, un robot désherbeur (généralement un flotteur) est censé couvrir une parcelle de vignoble, de manière optimale. Ici, la "manière optimale" fait référence à la ligne de chemin qui permet de couvrir l'ensemble du patch en un minimum de temps. Cependant, l'orographie du sol présente une variabilité importante, ainsi que la localisation des raisins. Les robots visent à heurter les pieds de raisin afin d'enlever l'herbe autour, puis de nombreuses collisions suivant différentes directions sont nécessaires pour s'assurer que toute l'herbe autour du pied de raisin est correctement enlevée.

Toute la variabilité pratique (sol, emplacement des raisins, répartition et taille de l'herbe, obstacles, ...) ainsi que la sensibilité intrinsèque de la dynamique à une faible variabilité des conditions physiques et opérationnelles, rend impossible la définition d'une trajectoire déterministe du robot. Dans ces conditions, un mouvement presque aléatoire semble être l'alternative la plus intéressante.

En pratique, pour éviter les contre-performances caractéristiques des mouvements totalement aléatoires, ce mouvement aléatoire opérant à l'échelle locale est combiné à une planification déterministe plus globale qui tente de mieux contrôler la couverture du vignoble en séquençant l'opération aux différentes parcelles locales couvrant l'ensemble domaine.

Nous visons à analyser les données collectées à partir d'un robot opérant dans différents patchs et dans différentes conditions (par rapport aux opérations de maintenance) afin d'identifier l'existence de motifs capables d'identifier le patch particulier dans lequel le robot opère, ou de distinguer les différents états du robot vis-à-vis des opérations de maintenance. Disposer d'une sorte de QR-code ou de carte d'identité de chaque robot, lorsqu'il opère au sein de chaque patch, dans un état particulier (sain ou malsain), est d'une importance majeure par rapport à la maintenance pré-

dictive ou opérationnelle des robots ou flotteurs de robots autonomes .

Nous analysons les données collectées afin d'en extraire le maximum d'informations pouvant servir à les différencier, permettant un clustering non supervisé et/ou une classification supervisée, préalablement à toute action concernant la modélisation à l'aide de régressions adaptées.

L'utilisation du clustering de données est presque simple, dès lors que les données sont homogènes et quantitativement exprimables à l'aide de nombres entiers ou réels, permettant des opérations booléennes ou algébriques (addition, multiplication, ...). L'intérêt d'organiser les données en groupes, de manière supervisée ou non, est qu'on suppose que les données appartenant à un groupe donné partagent certaines qualités avec les membres du groupe.

En procédant de manière non supervisée, la seule information pour regrouper les données est la distance entre elles. Les données qui restent proches les unes des autres sont censées partager certaines propriétés ou certains comportements. C'est le raisonnement pris en compte dans la très populaire technique *k-means* [64, 65]. Cependant, la notion de proximité, conduisant au concept dérivé de similarité, nécessite la définition d'une métrique à des fins de comparaison. Lorsque les données sont bien définies dans un espace vectoriel, les distances peuvent être définies et les données peuvent être comparées en conséquence. Dans le cas de la classification supervisée, on cherche la frontière linéaire (ou non linéaire) séparant les différents groupes sur la base d'une qualité ou d'une propriété qui pilote le clustering des données. Dans ce dernier cas, la meilleure frontière séparant deux groupes de données est celle maximisant la distance des données disponibles à la frontière, afin de maximiser la robustesse de séparation. C'est ainsi que fonctionne la machine à vecteurs de support, SVM, par exemple [66].

Dans les deux cas (supervisé et non supervisé) l'existence d'une métrique permettant la comparaison des données est supposée. Cependant, très souvent, les données peuvent être beaucoup plus complexes, comme par exemple lorsqu'il s'agit d'informations hétérogènes, éventuellement catégorielles ou qualitatives. C'est par exemple le cas lorsqu'une pièce fabriquée est décrite par sa carte d'identité constituée du nom de l'employé impliqué dans l'opération, la désignation des matériaux employés (certains d'entre eux étant donnés par son nom commercial), la température du four dans lequel la pièce a été durcie et le temps de traitement. Dans ce cas, la comparaison de deux parties devient assez controversée si la métrique employée n'est pas correctement définie. Dans ces circonstances, généralement, les métriques sont apprises à partir des données d'apprentissage existantes, comme c'est le cas lors de l'utilisation d'arbres de décision (ou de son homologue de forêt aléatoire) [67, 68], de code-to-vector [16] ou de neurones réseaux [69]. La situation devient encore plus extrême lorsque les données ont un contenu topologique important et profond. C'est le cas par exemple des séries temporelles ou des images de microstructures riches. Ceux-ci sont généralement rencontrés en science des matériaux lors de la description des métamatériaux (également appelés matériaux fonctionnels), ou de ceux présentant un gradient de propriétés ou des architectures mésoscopiques. Ainsi, même dans des conditions nominales, les séries temporelles différeront si elles sont comparées à

leurs valeurs respectives à chaque instant. C'est-à-dire que deux séries temporelles, même lorsqu'elles décrivent le même système dans des conditions similaires, ne correspondent jamais parfaitement. Ainsi, ils diffèrent même s'ils ressemblent à une certaine métrique qui devrait être apprise. Par exemple, notre électrocardiogramme mesuré pendant deux minutes consécutives présentera une ressemblance, mais certainement les deux ne sont pas identiques, rendant ainsi une correspondance parfaite impossible. Une petite variation créera un désalignement nécessitant des métriques moins sensibles à ces effets. Le même raisonnement s'applique lorsque l'on compare deux profils d'une surface rugueuse, deux images d'une mousse prises à deux endroits proches, ... ils présentent une ressemblance même s'ils ne correspondent pas parfaitement.

Ainsi, des techniques visant à aligner les données ont été proposées. Dans le cas des séries temporelles, le DTW [70, 71] a été appliqué avec succès dans de nombreux domaines. La théorie du transport optimal est née en réponse à des problèmes similaires [38].

Une autre voie consiste à renoncer à *aligner* les données, et à se concentrer sur l'extraction des descripteurs adéquats et orientés objectif de ces données complexes, permettant la comparaison, le clustering, la classification et la modélisation (à partir de régressions non linéaires).

Une première possibilité consiste à extraire les principaux descripteurs statistiques de séries temporelles ou d'images (moments, corrélations, covariogrammes, ...) [72]. Parfois, des données exprimées dans les domaines spatio-temporels habituels, sont transformées en d'autres espaces où leur manipulation est censée être plus simple, comme Fourier, Laplace, DCT, Wavelet, ... descriptions de données. Les descriptions les plus précieuses (au sens donné plus loin) semblent être celles qui maximisent la rareté. Celles-ci sont largement prises en compte lors de l'utilisation de la détection compressée [24], car elle représente une manière compacte, concise et complète de représenter des données qui semblaient beaucoup plus complexes dans l'espace physique habituel (espace et temps).

Nous considérons cette dernière voie, en utilisant une description basée sur la topologie des données, dans le but de classer et aussi de construire des régressions robustes exprimant des propriétés ou des performances à partir des données d'entrée exprimées à partir de sa description topologique.

Par rapport à nos développements antérieurs, cela répond à un objectif nouveau et complexe : comment la topologie contenue dans la trajectoire qu'un robot autonome suit dans un environnement nuageux (où les interactions limitent l'horizon de prévisibilité) peut renseigner sur l'emplacement du robot (qui s'intègre dans le vignoble entier) ou l'état du robot (vis-à-vis des opérations de maintenance).

## Conclusion

L'analyse des données topologiques et l'homologie persistante s'est avérée être une approche fiable et utile pour étudier les changements dans les systèmes observés sans une connaissance préalable du phénomène physique et de la modélisation.

Cette méthodologie est particulièrement adaptée à l'étude de jeux de données avec des informations topologiques et géométriques élevées telles que des formes, des signaux, des surfaces et des trajectoires. En extrayant la structure algébrique sous-jacente des données, il est possible de comparer et de détecter des changements dans les systèmes et la dynamique étudiés, d'extraire des descripteurs statistiques et de caractériser les systèmes physiques.

Il s'agit d'une méthodologie agnostique robuste et modèle, avec des possibilités de généralisation prometteuses. De plus, il ne nécessite pas d'hypothèse de continuité supplémentaire sur le collecteur de données, tout en étant sensible à l'échelle et à la dimension.

Compte tenu du cadre et des calculs décrits, il est essentiel d'avoir une spécification claire d'une dimensionnalité donnée d'un problème afin d'avoir une description géométrique adaptée : unidimensionnelle (comme les séries temporelles univariées), bidimensionnelle (comme les trajectoires planaires), tridimensionnel (comme la déformation de formes), multidimensionnel (séries chronologiques multivariées). La taille des données est également un paramètre crucial, car elle peut nécessiter un choix particulier de filtrage. De plus, certaines spécificités des données peuvent jouer un rôle, comme la filtration Alpha (triangulation) pour les surfaces maillées, la filtration Rips (sphères) pour la diffusion comme la dynamique et la filtration Sublevelset pour les données séquentielles. Enfin, des métriques spécifiques et personnalisées (transport optimal) permettent de tirer parti de la persistance calculée pour l'extraction de caractéristiques la plus pertinente. Les propriétés de ces caractéristiques (espace vectoriel, stabilité) affecteront largement le choix des procédures d'apprentissage ultérieures. Ce cadre affiche des capacités très prometteuses pour d'autres investigations et applications, comme dans les jumeaux numériques. Il pourrait permettre d'incorporer des ensembles de données de capteurs supplémentaires, d'améliorer la prédiction du comportement et du régime, tout en étant robuste au bruit et agnostique au modèle.

# Contents

# 1 Introduction

## 1.1 Motivations and Outlines

Topological Data Analysis is the study of datasets based on their topological properties and invariants. It provides a robust and general framework to analyse data without the need for an analytic ambient manifold. It relies on the characterization of the data shapes, loops and holes, using tools for algebraic topology and computational geometry.

To analyze how the data is shaped without the need for a metric space embedding, the approach is to create geometrical objects known as filtrations. By covering the data with simplices or spheres of different dimensions and scales, an approximation of the data cloud is obtained. The idea then is to see how the simplicial features of different dimensions (points, segments, triangles, tetrahedrons ...) are distributed across the scales, and how they appear and disappear (forming loops and holes). For that purpose, the homology groups are introduced. It is an algebraic representation of the relationships between the simplicial features. They are grouped into classes representing chains of simplices, that we refer to as features. The persistent homology allows then to track how these features are distributed scales and dimensions, associating to each a birth and a death scale, representing the scale at which the element has appeared and disappeared.

To summarize the information of the persistent homology, the features are grouped according to their dimensions and lifetime, following the homology groups representation. They can then be summarized using a collection of descriptors such as the persistence diagrams and barcodes. It is a representation in 2D coordinates of the birth and death of the features. More advanced representations can then be computed such as the persistence images, kernels, entropy and landscapes.

The purpose of these persistence descriptors is to have measurable objects, that can be used to compute statistics. The optimal transport is an ideal framework to do so, allowing to define robust metrics on the persistence representations. It is then possible to incorporate those computed statistics and representations into machine learning models describing the underlying physical system.

The need for a general data driven approach has become more acute with the multiplication of the data sources, the ever growing volume of big data available, and the interdisciplinary nature of the new engineering issues.

Our methodology presented here, relies on a general framework of persistent homology computation. We consider the data cloud as finite discretization of the physical system, and compute the associated and most adapted filtration, taking into account the nature of the data, the dimensionality and the targeted quantities of interest. We then proceed to compute the persistence descriptors, along with the adapted metrics, statistics and optimal transport. The output can finally be integrated in a learning process, for clustering, classification, and regression.

We will address four applications of topological data analysis:

- Advanced Parametric Modes Clustering [3] in Chapter 2

- Tape Surfaces Characterization [1] in Chapter 3
- Advanced Driver-Assistance Systems [2] in Chapter 4
- Monitoring and Anticipating Robots Functioning Behaviors [4] in Chapter 5

### 1.1.1 Advanced Parametric Modes Clustering

Modal analysis is widely used for addressing NVH –Noise, Vibration and Hardness– in automotive engineering. The so-called principal modes constitutes an orthogonal basis, obtained from the eigenvectors related to the dynamical problem. When this basis is used for expressing the displacement field of a dynamical problem, the model equations become uncoupled. Moreover, a reduced basis can be defined according to the eigenvalues magnitude, leading to an uncoupled reduced model, specially appealing when solving large dynamical systems. However, engineering looks for optimal designs and therefore it focuses on parametric designs needing the efficient solution of parametric dynamical models. Solving parametrized eigenproblems remains a tricky issue, and therefore, non-intrusive approaches are privileged. In that framework, a reduced basis consisting of the most significant eigenmodes is retained for each choice of the model parameters under consideration. Then, one is tempted to create a parametric reduced basis, by simply expressing the reduced basis parametrically by using an appropriate regression technique. However an issue remains, that limits the direct application of the just referred approach, the one related to the basis ordering. In order to order the modes before interpolating them, different techniques were proposed in the past, being the Modal Assurance Criterion –MAC– one of the most widely used. We propose an alternative technique that instead of operating at the eigenmodes level, classify the modes with respect to the deformed structure shapes that the eigenmodes induce.

The main limitation of modal analysis is the lack of validity the reduced basis in the case of parametric models. For parametrized dynamical systems, the model matrices will depends on those parameters, and solving parametric eigenproblems remains a tricky issue.

When one is not really interested in the transient regime, but much more in the forced regime, harmonic analysis represents a valuable route. The so-called Proper Generalized Decomposition—PGD— enables considering the frequency as a model extra-parameter as well as addressing general (non-proportional) damping and non-linear dynamics, under the stringent real-time constraint, with even the inclusion of model parameters as extra-coordinates [18, 17, 20].

However, certain applications need accurate transient responses, and in that case the formulation and solution of the dynamical problem in the time domain is retained.

For that purpose, one the most widely used techniques is modal analysis. Other than the benefits in the time integration, due to the dynamical system decoupling, the eigenmodes benefit from a physical interpretation, of great interest for the designer or structural analyst. Still, when considering parametric models as it is always

the case during the design stage, when the material and geometry are not totally defined, the dynamical modes depends on those parameters as previously discussed. Having a surrogate model expressing the parametric evolution of the eigenmodes is of great interest. Constructing those surrogate models is nowadays quite mature, by using usual and advanced nonlinear regressions [19], the last making use of sparsity and appropriate regularisation for operating in high-dimensional settings, while keeping as reduced as possible the number of data (eigenproblems solution), and leading to rich enough (nonlinear) regressions while avoiding overfitting. Here the trickiest issue is not the regression construction, but the fact of ordering the different eigenmodes involved in the modal basis for each parameters choice, in order to created $N$ clusters (or less in the reduced case), and putting in each one a mode of each modal basis, such that modes in each cluster remain close (in a certain metrics). The main issue remains the metric to be use to successfully and efficiently accomplishing such clustering. In general such clustering is performed by operating at the level of the eigenmodes, in the associated vector space, by using for example the Modal Assurance Criterion—MAC— [52] that proceed comparing the modes resulting from each eigenproblem, by using the usual scalar product (modes similar to a given one should remain quite collinear).

When operating in high dimensional parametric spaces, sparsely sampled, the matrices involved in the resulting eigenproblem can vary a lot from one choice of the parameters to another, and consequently the scalar product criterion at the basis of the MAC could fail. On the other hand, the fact of proceeding in a vector space needs to carefully address the expression of the different modes by considering the same frame for all the analyzed mechanical systems.

For alleviating those difficulties, we propose an alternative technique that instead of operating at the eigenmodes level, classify the modes with respect to the deformed structures shape that the eigenmodes produce, taking advantage from the invariance property of topology. Thus, we are employing a metric able to compare shapes, more that a metric for comparing the vectors (eigenmodes) that produced those shapes, the last being more intrinsic and inheriting invariance features. Moreover, in the present case study, eigenmodes are heterogeneous in the sense that they involve displacements and rotations, whereas the associated deformed surfaces are purely geometrical.

In Chapter 2, we address the classification of a series of modal basis related to the eigenmodes of a thin structure equipped with a mesh consisting of shell elements, with displacement and rotation degrees of freedom at each node of the mesh.

The thickness of the structural part varies, with its consequent effect on the mass and stiffness matrices (damping is assumed proportional) and consequently on the eigenvalues and eigenvectors, the former defining the number of modes to be retained in the reduced basis. In the present study, the six rigid modes representing the whole structure translation (three modes) and rotation (three modes) will be discarded and among the remaining pairs eigenvalue-eigenvector, the most relevant six eigenvectors (corresponding to the six highest eigenvalues) retained in the reduced basis related to each choice of the model parameter (the thickness).

These six modes related to each structure (related to a thickness value) define a reduced basis that one would like render parametric. However, prior to construct a regression able to define the reduced basis for each possible choice of the parameter (thickness) one should classify the six eigenmodes of each reduced basis associated to each structure, into six clusters.

This task is compulsory to facilitate the interpolation in the parametric space and also to attach a physical sense to those modes. One could imagine that for a given thickness the most relevant deformation mode could be related to the extension whereas for another choice of the thickness the most relevant deformation mode could be the bending. In such a case, one prefers create a cluster grouping similar deformation modes, to evaluate how each of them depends from the parameter from one side, and from the other to facilitate the subsequent construction of the parametric modal reduced basis.

To preform such a clustering, we must employ an appropriate metric to compare those modes. In general this comparison was traditionally performed by comparing the eigenmodes within the vector space to which they belongs. In the present work, as announced previously, we prefer applying the deformation mode to the reference (undeformed) structure, that is, applying the eigenmode at the nodes location in the reference structure for obtaining the deformed structure related to each mode of each structure configuration (thickness) and then cluster the resulting deformed structures with respect to their shape.

### 1.1.2 Tape Surfaces Characterization

Among composite forming processes for manufacturing structural parts based on the consolidation of pre-impregnated preforms, e.g., sheets, tapes, .... the automated tape placement (ATP) appears as one of the most interesting techniques due to its versatility and its in-situ consolidation, thus avoiding the use of autoclave. In particular, to obtain the cohesion of two thermoplastic layers two specific physical conditions are needed (a) an almost perfect contact (intimate contact) and (b) a temperature enabling molecular diffusion within the process time window, while avoiding thermal degradation. To reach this goal, a tape is placed and progressively bonded to the substrate consisting of the tapes previously laid-up. Due to the low thermal conductivity of usual resins, an intense local heating is usually considered (laser, gas torches, etc.) in conjunction with a local pressure applied by a moving roller. Thus, the two main factors to ensure the intimate contact at the plies surfaces are pressure and heat. Intimate contact is required to promote the molecular diffusion. In this process heat plays a double role, on one hand it enhances molecular mobility and on the other hand, the decrease of the material viscosity with the temperature increase, facilitates the squeeze flow of the heated asperities located on the ply surfaces under the compression applied by the consolidation roller.

The numerical model of ATP was introduced in [6] by using the so-called Proper Generalized Decomposition (PGD) [7, 8, 9, 10, 12]. The separated representation involved in the PGD enables the 3D high-resolution solution of models defined in degenerated domains where at least one of their characteristic dimensions remains

much smaller than the others and also constructing solutions of parametric models where the model parameters are considered as extra-coordinates [11, 13].

Physical modelling and simulation for Automated Tape Placement (ATP) have been proposed in [14] to study the influence of material and process parameters, while consolidation modelling and sPGD-based non-linear regression have been used in [15] to identify the main surface descriptors for a comprehensive characterization of the tape surfaces.

In Chapter 3, we first revisit the consolidation modeling and its high resolution simulation, enabling the evaluation of the time evolution of the degree of intimate contact –DIC– when two rough surfaces are put in contact, heated and compressed. The roughness squeezing mainly occurs along the transverse direction (the one related to the tape width) induced by the roller compression. Thus, the flow occurs in the transverse section in which the surface reduces to a one-dimensional curve (the so-called surface profile).

It is well-known at an experimental level that the consolidation degree strongly depends on the surface characteristics (roughness). In particular, same process parameters applied to different surfaces produce very different degrees of intimate contact. It allows us to think that the surface topology plays an important role along this process. However, solving the physics-based models for simulating the roughness squeezing occurring at the tapes interface represents a computational effort incompatible with online process control purposes. An alternative approach consists of taking a population of different tapes, with different surfaces, and simulating the consolidation for evaluating for each one the progression of the degree of intimate contact –DIC– while compressing the heated tapes, until reaching its final value at the end of the compression. The final goal is creating a regression able to assign a final value of the DIC to any surface, enabling online process control. The main issue of such an approach is the rough surface description, that is, the most precise and compact way of describing it from some appropriate parameters easy to extract experimentally, to be included in the just referred regression.

In order to extract a concise and complete description of the rough surfaces, we use a topological description [27, 28, 29, 26] of the surface profiles, to construct descriptors such as the persistence diagrams and images. Then, the persistence images are considered for classifying surfaces, or as descriptors involved in the regression relating them to the final DIC reached in the consolidation process, enabling real-time decision making.

### 1.1.3 Advanced Driver-Assistance Systems

While there have recently been considerable advances in self-driving car technology, driving still relies mainly on human factors. Even in self-driving mode, human drivers must often make decision in a fraction of a second to avoid accidents. Therefore, it is still of utmost importance to develop systems capable of discerning if the human driver is attentive or not to the road conditions. In general, the so-called advanced driver assistance systems (ADAS) [55, 56] are systems that are able to improve the driver's performance, among which, adaptive speed limiters, pedestrian

detectors [57], and cruise controllers are some of the most popular systems. Fatigue alerting systems are among the most useful among ADAS systems, and the aim of this work is to contribute to the development of such a system based on a systematic analysis of drivers in actual driving conditions.

The estimation of the driver's condition (degree of attention to the road, fatigue, etc.) is a very important factor to ensure safety in driving [58, 59]. A recent review on the topic can be found in [60].

In Chapter 4, we aim to extract behavior patterns from car user data to be able to accurately estimate their state. We use experimental data, gathered while applying mechanical stimulation to people seated in an automobile.

Our main goal is to extract patterns of behavior from the data to allow us to learn the most relevant factors affecting driver's attention to the situation of the road.

We combine some tools from Morse theory [35] and topological data analysis (TDA) with all of the associated concepts and methods (e.g., Betti numbers, homology persistence, barcodes, persistence images, etc.) [34], most of them introduced and employed later in order to analyze and classify the experimental data. This allows us to introduce concepts as barcodes, that is, persistent and life-time diagrams in a similar way to how they are used in persistent homology. Our main goal is to predict car user behavior following a supervised approach [27]. Instead of considering an original sensor signal as the quantity of interest, we focus on its topological features. In this sense, the framework proposed in this paper allows us to unveil the true dimensionality of data or, in other words, the actual number of factors affecting driver's performance. Thus, we model a sensor signal as a dynamical system, and, therefore, our approach seems to be better at describing its properties, or rather its variations, such as extrema, patterns, and self-similarity, than other approaches.

We note that our approach is, in some senses, similar to that followed by Milnor and Thurston [36] in the study of the combinatorial properties of dynamical systems by combining tools from automata theory.

### 1.1.4 Monitoring and Anticipating Robots Functioning Behaviors

Autonomous robots follow a number of rules introduced into their controllers [61, 62, 63]. However, when they interact with the environment, small variations may result in long-time unpredictable motion. This behavior is very usual in mechanics, characterizing systems exhibiting deterministic chaos.

In the practical case addressed in Chapter 5, 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.

All the practical variability (ground, grape location, grass distribution and size, obstacles, ...) as well as the intrinsic sensibility of the dynamics to small variability in the physical and operational conditions, makes it impossible to define a deterministic robot trajectory. In these conditions, an almost random motion seems to be the most valuable alternative.

In practice, to avoid under-performances characteristic of fully random motions, that 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.

We aim 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.

We analyze the collected data in order to extract the maximum information that could serve for differentiating them, enabling unsupervised clustering and/or supervised classification, prior to any action concerning modeling using adapted regressions.

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, 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.

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 [64, 65]. 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 nonlinear) 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 [66].

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 categorical 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) [67, 68], code-to-vector [16] or neural networks [69].

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 [70, 71] has been successfully applied in many domains. The theory of optimal transport arose as a response to similar issues [38].

Another route consists of renouncing to *align* the data, and focusing on extracting the adequate, goal-oriented descriptors of these complex data, enabling comparison, clustering, classification and modeling (from nonlinear regressions).

A first possibility consists of extracting the main statistical descriptors of time series or images (moments, correlations, covariograms, ...) [72]. 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 [24], 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).

We consider this last route, while using a description based on the topology of data, with the aim of classifying and also constructing robust regressions expressing properties or performance from the input data expressed from its topological description.

When compared with our former developments, this 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).

## 1.2 Preliminary Definitions and Results

Our approach is to define the tools and concepts needed for the topological data analysis in a concise and sufficient way, with an emphasis on a vector space representation of the homology operations.

In what follows we consider as data-set $\mathbb{M}$ a finite subset of points in $\mathbb{R}^3$.

### 1.2.1 Simplicial Filtration

In order to describe the geometry of a data-set $\mathbb{M}$ we first identify the geometrical features associated with $\mathbb{M}$:

- A vertex $[\mathbf{x}_m]$ is generated by an individual point $\mathbf{x}_m \in \mathbb{M}$;
- A segment $[\mathbf{x}_m, \mathbf{x}_n]$ joins two vertex $[\mathbf{x}_m],\ [\mathbf{x}_n] \in \mathbb{M}$

$$[\mathbf{x}_m, \mathbf{x}_n] := \left\{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \lambda \mathbf{x}_m + (1-\lambda)\mathbf{x}_n \text{ where } 0 \leq \lambda \leq 1\right\};$$

- A triangle is generated by three different vertex $[\mathbf{x}_m],\ [\mathbf{x}_n],\ [\mathbf{x}_l] \in \mathbb{M}$, such that $\mathbf{x}_m - \mathbf{x}_n$ and $\mathbf{x}_m - \mathbf{x}_l$ are linearly independent, and then:

$$[\mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_l] := \left\{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \lambda_m \mathbf{x}_m + \lambda_n \mathbf{x}_n + \lambda_l \mathbf{x}_l\right\},$$

  where $\lambda_m,\ \lambda_n$ and $\lambda_l$ are the barycentric coordinates, with $\lambda_m + \lambda_n + \lambda_l = 1$;

- A tetrahedron is generated by four different vertices $[\mathbf{x}_m],\ [\mathbf{x}_n],\ [\mathbf{x}_l],\ [\mathbf{x}_p] \in \mathbb{M}$, such that $\mathbf{x}_m - \mathbf{x}_n$, $\mathbf{x}_m - \mathbf{x}_l$ and $\mathbf{x}_m - \mathbf{x}_p$ are linearly independent, and then:

$$[\mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_l, \mathbf{x}_p] := \left\{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \lambda_m \mathbf{x}_m + \lambda_n \mathbf{x}_n + \lambda_l \mathbf{x}_l + \lambda_p \mathbf{x}_p\right\},$$

  where $\lambda_m,\ \lambda_n$, $\lambda_l$ and $\lambda_p$ are the barycentric coordinates, with $\lambda_m + \lambda_n + \lambda_l + \lambda_p = 1$.

More generally, 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 independent, as illustrated in Fig. 39.

The so-called *simplicial complex* $\mathcal{S}(\mathbb{M})$ is then a finite collection of sets that is closed under the subset relation, i.e., if $a \in A$ and $b \subset a$, then $b \in A$.

In order to perform algebraic operations on the elements of $\mathcal{S}(\mathbb{M})$, we define a *Removal Operator* for higher dimensional simplices.

**Definition 1.** *Given $k = 1, 2$ and for $0 \leq i \leq k$ the removal operator $R_i$ removes the $i$-th position elements of a $k$-simplex :*

$$R_i([x_0, x_1 \ldots, x_k]) := [x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k].$$

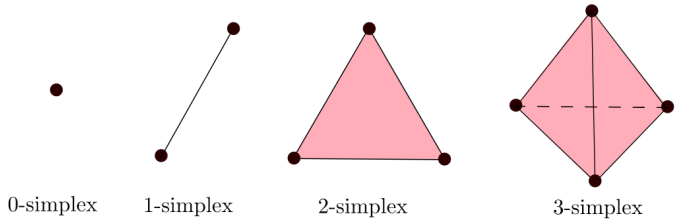



Figure 1: Simplices of different dimensions

In order to describe efficiently the elements of the filtration $\mathcal{S}(\mathbb{M})$ and map a set of $k$-simplices to a set of $k-1$-simplices, we construct a finite sequence of sets of $k$-simplices, denoted by $\mathcal{S}_k(\mathbb{M})$ $(k \geq 0)$, obtained as follows.

**Definition 2.** *Consider a finite three dimensional set* $\mathbb{M} = \{x_0, x_1, x_2 \ldots\}$.

- $\mathcal{S}_k(\mathbb{M}) = \emptyset$ *for every* $k \neq 0, 1, 2$; *otherwise*
- *Set* $\mathcal{S}_0(\mathbb{M}) := \{[x] : x \in \mathbb{M}\}$, *then the simplices in* $\mathcal{S}_k(\mathbb{M})$ *for each* $k = 1, 2$ *are obtained from the simplices in* $\mathcal{S}_{k-1}(\mathbb{M})$ *taking into account the following two properties:*

  *(P1) for every* $\sigma \in \mathcal{S}_k(\mathbb{M})$ *it holds that* $R_i(\sigma) \in \mathcal{S}_{k-1}(\mathbb{M})$ *for* $0 \leq i \leq k$, *and*

  *(P2) if* $\sigma, \gamma$ *are in* $\mathcal{S}_k(\mathbb{M})$ *and* $\sigma \cap \gamma \neq \emptyset$, *then there exists* $1 \leq \ell \leq k$ *such that* $\sigma \cap \gamma \in \mathcal{S}_{k-\ell}(\mathbb{M})$.

As consequence of the above inductive construction of the simplices we provide the following definition of the face of a simplex.

**Definition 3.** *Let be* $\sigma \in \mathcal{S}_k(\mathbb{M})$ *for* $k = 1, 2$. *Then we will say that* $\gamma \in \mathcal{S}_{k-\ell}(\mathbb{M})$ *for some* $1 \leq \ell \leq k$ *is a face of* $\sigma$ *if there exists a finite sequence* $\alpha_0\alpha_1 \cdots \alpha_{\ell-1} \in \{0, 1, \ldots, k\}^\ell$ *where* $\alpha_i \neq \alpha_j$ *such that*

$$\gamma = \left(R_{\alpha_0} \circ \cdots \circ R_{\alpha_{\ell-1}}\right)(\sigma).$$

Since we consider that $x_i \in \mathbb{R}^3$ for $1 \leq i \leq M$, and that $\mathbb{M}$ is a surface, we associate to our surface $\mathbb{M}$ three non-empty sets of simplices:

$$\mathcal{S}(\mathbb{M}) = \{\mathcal{S}_0(\mathbb{M}), \mathcal{S}_1(\mathbb{M}), \mathcal{S}_2(\mathbb{M})\},$$

recall that $\mathcal{S}_k(\mathbb{M}) = \emptyset \subset \mathcal{S}(\mathbb{M})$, for every integer number $k \in \mathbb{Z}$, $k \neq 0, 1, 2$. As we will see below it will have some consequences. From the construction of $\mathcal{S}(\mathbb{M})$ the following two properties holds.

(P3) Every face of a simplex $\sigma \in \mathcal{S}(\mathbb{M})$ is also in $\mathcal{S}(\mathbb{M})$.

(P4) Given $\sigma, \gamma \in \mathcal{S}(\mathbb{M})$ either $\sigma \cap \gamma = \emptyset$ or $\sigma \cap \gamma$ is a face of $\sigma$ and $\gamma$.

### 1.2.2 Simplicial Homology

In order to perform geometric operations (like unions of simplices) at each $k$-level and to describe the relationship between two consecutive levels (like cut the faces of a simplex) we endow to each $\mathcal{S}_k(\mathbb{M})$ with an algebraic structure of vector space over a finite field of scalars. To this end, we consider the finite field $\mathbb{Z}_2 = \{\mathbf{0}, \mathbf{1}\}$.

**Definition 4.** *For $0 \leq k \leq 3$ we introduce the vector space of formal series of $k$-simplices with coefficients over the finite field $\mathbb{Z}_2$ as*

$$\mathbb{Z}_2[\mathcal{S}_k(\mathbb{M})] := \left\{ \sigma : \sigma = \sum_{i=1}^{\ell} \eta_i \, \sigma_i \text{ where } \eta_1, \ldots, \eta_\ell \in \mathbb{Z}_2 \text{ and } \sigma_1, \ldots, \sigma_\ell \in \mathcal{S}_k(\mathbb{M}) \right\}.$$

Observe that if $\sigma \in \mathcal{S}_k(\mathbb{M})$ then we can identify this simplex with the formal series also denoted by $\sigma = \mathbf{1}\,\sigma \in \mathbb{Z}_2[\mathcal{S}_k(\mathbb{M})]$. In consequence,

$$\sigma + \sigma = \mathbf{1}\,\sigma + \mathbf{1}\,\sigma = 0,$$

because $\mathbf{1} + \mathbf{1} = \mathbf{0}$ in $\mathbb{Z}_2$. Thus, for a given $\sigma_1, \ldots, \sigma_\ell \in \mathcal{S}_k(\mathbb{M})$ the formal series $\sigma = \sum_{i=1}^{\ell} \eta_i \, \sigma_i$ represents a union or a "packet" of $k$-simplices where $\eta_i = \mathbf{1}$ if the $k$-simplex $\sigma_i$ is in $\sigma$ and $\eta_i = \mathbf{0}$ otherwise. For $k \in \mathbb{Z}$, $k \neq 0, 1, 2, 3$ we have $\mathbb{Z}_2[\mathcal{S}_k(\mathbb{M})] = \mathbb{Z}_2[\emptyset] = \{0\}$ is the trivial vector space.

**Example 1.** *Consider a surface $\mathbb{M} = (\mathbb{M}, \mathcal{S}(\mathbb{M}))$ given by the surface of a tetrahedron defined by a set of points $\mathbb{M} = \{x_0, x_1, x_2, x_3\} \subset \mathbb{R}^3$ (see Figure 2) and where*

$$\begin{aligned}
\mathcal{S}_2(\mathbb{M}) &= \{[x_1, x_2, x_3], [x_0, x_2, x_3], [x_0, x_1, x_3], [x_0, x_1, x_2]\}, \\
\mathcal{S}_1(\mathbb{M}) &= \{[x_2, x_3], [x_1, x_3], [x_1, x_2], [x_0, x_3], [x_0, x_2], [x_0, x_1]\}, \\
\mathcal{S}_0(\mathbb{M}) &= \{[x_0], [x_1], [x_2], [x_3]\}.
\end{aligned}$$

*Now, we can identify $\mathbb{Z}_2[\mathcal{S}_0(\mathbb{M})] \equiv \mathbb{Z}_2^4$, by using*

$$\begin{aligned}
[x_0] &\equiv (\mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}), \\
[x_1] &\equiv (\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}), \\
[x_2] &\equiv (\mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}), \\
[x_3] &\equiv (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}).
\end{aligned}$$

*Now, we can identify $\mathbb{Z}_2[\mathcal{S}_1(\mathbb{M})] \equiv \mathbb{Z}_2^6$, where we identify*

$$\begin{aligned}
[x_0, x_1] &\equiv (\mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}), \\
[x_0, x_2] &\equiv (\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}), \\
[x_0, x_3] &\equiv (\mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}), \\
[x_1, x_2] &\equiv (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}), \\
[x_1, x_3] &\equiv (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}), \\
[x_2, x_3] &\equiv (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}).
\end{aligned}$$

*Finally, we have $\mathbb{Z}_2[\mathcal{S}_2(\mathbb{M})] \equiv \mathbb{Z}_2^4$.*

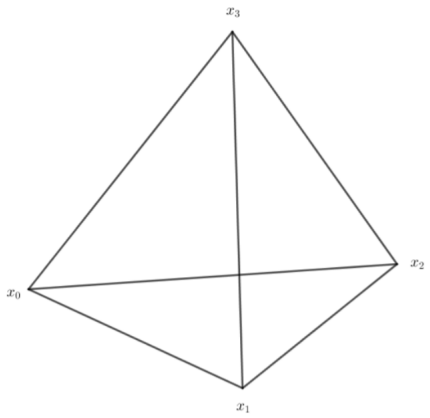



Figure 2: A tetrahedron defined by a set of points $\mathbb{M} = \{x_0, x_1, x_2, x_3\} \subset \mathbb{R}^3$.

From now on, we identify $\mathbb{Z}_2[\mathcal{S}_k(\mathbb{M})] \equiv \mathbb{Z}_2^{\ell_k(\mathbb{M})}$ where $\ell_k(\mathbb{M})$ is the number of elements in $\mathcal{S}_k(\mathbb{M})$ for each $0 \leq k \leq 2$. In particular, $\ell_0(\mathbb{M}) = M$. Moreover

- For any integer number $k$ we can endow each vector space $\mathbb{Z}_2^{\ell_k(\mathbb{M})}$ with a total ordering. To this end for a given $[x_{i_0}, x_{i_1}, \ldots, x_{i_k}] \in \mathbb{Z}_2^{\ell_k(\mathbb{M})}$ where $i_0 < i_1 < \ldots < i_k$ we consider the lexicographical order of its index set $i_0 i_1 \cdots i_k \in \{0, 1, \ldots, M\}^k$. For example, consider the surface of a tetrahedron $\mathbb{M} = \{x_0, x_1, x_2, x_3\} \subset \mathbb{R}^3$ as in Example 1, then in $\mathcal{S}_2(\mathbb{M})$ we have

$$[x_0, x_1] < [x_0, x_2] < [x_0, x_3] < [x_1, x_2] < [x_1, x_3] < [x_2, x_3].$$

- If $k = 1, 2$ then we can extend the map $R_i : \mathbb{Z}_2^{\ell_k(\mathbb{M})} \longrightarrow \mathbb{Z}_2^{\ell_{k-1}(\mathbb{M})}$ defined as $R_i\left(\sum_{j=1}^{\ell} \eta_j \, \sigma_j\right) = \sum_{j=1}^{\ell} \eta_j \, R_i(\sigma_j)$. Now, $R_i$ is a linear map between vectors spaces for $0 \leq i \leq k$.

- Assume that $\sigma, \gamma \in \mathbb{Z}_2^{\ell_k(\mathbb{M})}$ and $\sigma \cap \gamma = R_i(\sigma) = R_i(\gamma)$. Then

$$R_i(\mathbf{1}\,\sigma + \mathbf{1}\,\gamma) = \mathbf{1}\,R_i(\sigma) + \mathbf{1}\,R_i(\gamma) = 0,$$

  because $\mathbf{1} + \mathbf{1} = \mathbf{0}$ in $\mathbb{Z}_2$.

Next, we associate an incidence matrix (defined by a linear map between vector spaces) to each pair of consecutive levels $(k-1, k)$ as we show in the next example.

**Example 2.** *Consider the surface $\mathbb{M}$ given in the Example 1. We can relate the different homology levels by using matrices following the next strategy. In Table 1 the columns are in correspondence with the 1-simplices of $\mathbb{M}$, the rows are in correspondence with the 0-simplices and the entries are determined by incidence of a 1-simplex with its 0-simplex face. The result is a matrix $4 \times 6$ matrix with $\mathbb{Z}_2$ entries, that is, a $\mathbb{Z}_2^{4\times 6}$-matrix.*

*For the incidence matrix of 2-simplices against its 1-simplices considered as faces the construction of the corresponding $\mathbb{Z}_2^{6\times 4}$-matrix is explained in Table 2.*

| | $[x_0,x_1]$ | $[x_0,x_2]$ | $[x_0,x_3]$ | $[x_1,x_2]$ | $[x_1,x_3]$ | $[x_2,x_3]$ |
|---|---|---|---|---|---|---|
| $[x_0]$ | **1** | **1** | **1** | **0** | **0** | **0** |
| $[x_1]$ | **1** | **0** | **0** | **1** | **1** | **0** |
| $[x_2]$ | **0** | **1** | **0** | **1** | **0** | **1** |
| $[x_3]$ | **0** | **0** | **1** | **0** | **1** | **1** |

Table 1: The incidence matrix between the 1-level and the 0-level of $\mathbb{M}$.

| | $[x_0,x_1,x_2]$ | $[x_0,x_1,x_3]$ | $[x_0,x_2,x_3]$ | $[x_1,x_2,x_3]$ |
|---|---|---|---|---|
| $[x_0,x_1]$ | **1** | **1** | **0** | **0** |
| $[x_0,x_2]$ | **1** | **0** | **1** | **0** |
| $[x_0,x_3]$ | **0** | **1** | **1** | **0** |
| $[x_1,x_2]$ | **1** | **0** | **0** | **1** |
| $[x_1,x_3]$ | **0** | **1** | **0** | **1** |
| $[x_2,x_3]$ | **0** | **0** | **1** | **1** |

Table 2: The incidence matrix between the 2-level and the 1-level of $\mathbb{M}$.

The formal way to introduce the above matrices is the following.

**Definition 5.** *For $1 \leq k \leq 2$ we can define the following linear map between vector spaces:*

$$\partial_{k-1,k} : \mathbb{Z}_2^{\ell_k(\mathbb{M})} \longrightarrow \mathbb{Z}_2^{\ell_{k-1}(\mathbb{M})},$$
$$\sigma \mapsto \partial_{k-1,k}\left(\sum_{i=1}^{\ell} \eta_i\, \sigma_i\right) = \sum_{j=0}^{k} R_j\left(\sum_{i=1}^{\ell} \eta_i\, \sigma_i\right).$$

*This map uses the whole set $\{R_0, R_1, \ldots, R_k\}$ of remove the $i$-th position linear map for $0 \leq i \leq k$. Observe that for $k = 0$ we have*

$$\partial_{-1,0} : \mathbb{Z}_2^{M} \longrightarrow \mathbb{Z}_{-1}[\mathcal{S}_0(\mathbb{M})] = \{0\}, \quad [x] \mapsto 0,$$

*the zero map, and also for $k = 3$*

$$\partial_{2,3} : \mathbb{Z}_2[\mathcal{S}_3(\mathbb{M})] = \{0\} \longrightarrow \mathbb{Z}_2^{\ell_2(\mathbb{M})}, \quad 0 \mapsto 0,$$

*we obtain the 0-map. Finally, we consider that $\partial_{k-1,k} = 0$ for all integer $k$ such that $k \neq 0, 1, 2$.*

To better understand the role of these maps, observe that if we have two simplices $[x_0, x_1, x_2]$ and $[x_3, x_4, x_5]$ in $\mathbb{Z}_2^{\ell_2(\mathbb{M})}$ without common faces then the union of both is described under this algebraic framework by the sum $[x_0, x_1, x_2] + [x_3, x_4, x_5]$ (see Figure 3(a)). By using the map $\partial_{1,2}$ we obtain its description in $\mathbb{Z}_2^{\ell_0(\mathbb{M})}$ as

$$\begin{aligned}\partial_{1,2}([x_0,x_1,x_2] + [x_3,x_4,x_5]) = {} & [x_1,x_2] + [x_0,x_2] + [x_0,x_1] \\ & + [x_4,x_5] + [x_3,x_5] + [x_3,x_4]\end{aligned}$$

That is, the sum of the six faces of the two simplices (see Figure 3(a)). They represent the total number of faces in their union. However, if we consider the union of two simplices $[x_0, x_1, x_2]$ and $[x_1, x_2, x_4]$ with a common face, $R_0([x_0, x_1, x_2]) = R_2([x_1, x_2, x_3]) = [x_1, x_2]$ (see Figure 3(b)), then its description in $\mathbb{Z}_2^{\ell_0(\mathbb{M})}$ is now

$$\begin{aligned}\partial_{1,2}([x_0, x_1, x_2] + [x_1, x_2, x_3]) &= [x_1, x_2] + [x_0, x_2] + [x_0, x_1] \\ &\quad + [x_2, x_3] + [x_1, x_2] + [x_1, x_3] \\ &= [x_0, x_2] + [x_0, x_1] + [x_2, x_3] + [x_1, x_3],\end{aligned}$$

because $[x_1, x_2] + [x_1, x_2] = 0$. This fact implies that the union of both is now described by the non-common four faces by forgetting the inner common face (see Figure 3(b)).

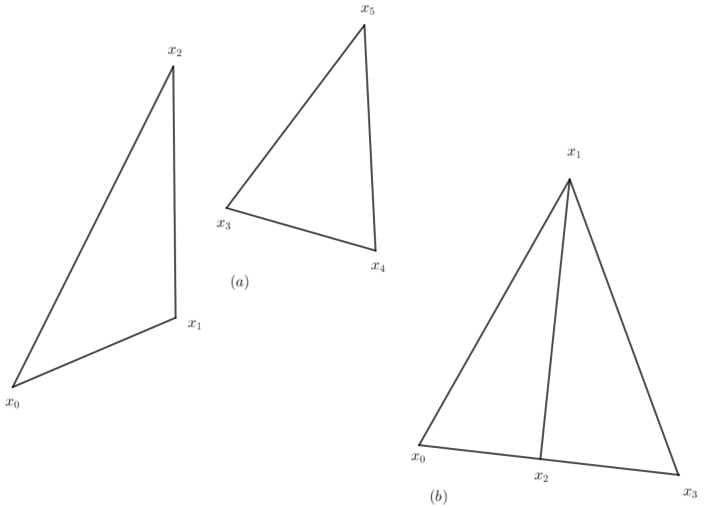



Figure 3: In (a) we have the union $[x_0, x_1, x_2] + [x_3, x_4, x_5]$ and in (b) $[x_0, x_1, x_2] + [x_1, x_2, x_3]$.

**Example 3.** *Consider the surface* $\mathbb{M}$ *given in the Example 1. Then the operators* $\partial_{k-1,k}$ *acts over the set of* $k$*-simplices* $(0 \leq k \leq 3)$ *as follows,*

$$\begin{aligned}\partial_{2,3} &= 0, \\ \partial_{1,2}([x_0, x_1, x_2]) &= [x_1, x_2] + [x_0, x_2] + [x_0, x_1], \\ \partial_{0,1}([x_0, x_1]) &= [x_0] + [x_1], \\ \partial_{-1,0} &= 0.\end{aligned}$$

*Moreover, the matrices associated to the linear maps $\partial_{0,1}$ and $\partial_{1,2}$ are*

$$\partial_{0,1} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix} \text{ and } \partial_{1,2} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix}.$$

*It is not difficult to see that the matrix product $\partial_{0,1} \cdot \partial_{1,2}$ is the zero matrix. Then the vector space generated by the columns of the matrix $\partial_{1,2}$, denoted by $\operatorname{Col} \partial_{1,2}$, is contained in the vector space, denoted by $\operatorname{Nul} \partial_{0,1}$, defined by the solutions of the homogeneous linear system*

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \nu \\ \eta \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

*Thus, we have the following subspaces $\operatorname{Col} \partial_{1,2} \subset \operatorname{Nul} \partial_{0,1} \subset \mathbb{Z}_2^6$. In a simlar way we have that $\operatorname{Col} \partial_{3,2} = \{\mathbf{0}\} \subset \operatorname{Nul} \partial_{1,2} \subset \mathbb{Z}_2^4$ and $\operatorname{Col} \partial_{0,1} \subset \operatorname{Nul} \partial_{-1,0} = \mathbb{Z}_2^4$.*

**Lemma 1.** *For all $0 \leq k \leq 3$:*

$$\partial_{k-1,k} \cdot \partial_{k,k+1} = 0$$

*Proof.* Indeed, it is true for all integer number $k$, if we introduce the 0 map as

$$\partial_{2,3} : \mathbb{Z}_2[\mathcal{S}_3(\mathbb{M})] = \{0\} \longrightarrow \mathbb{Z}_2[\mathcal{S}_2(\mathbb{M})]$$

This property means that the linear subspace

$$\operatorname{Col} \partial_{k,k+1} \subset \mathbb{Z}_2^{\ell_k(\mathbb{M})}$$

generated by the columns of the matrix $\partial_{k,k+1}$ is contained in the linear subspace

$$\operatorname{Nul} \partial_{k-1,k} := \left\{ \sigma \in \mathbb{Z}_2^{\ell_k(\mathbb{M})} : \partial_{k-1,k}\, \sigma = \mathbf{0} \right\}.$$

of the solution of the homogeneous linear system with matrix $\partial_{k-1,k}$. From the rank-nullity theorem, we known that

$$\ell_k(\mathbb{M}) = \dim \operatorname{Nul} \partial_{k-1,k} + \dim \operatorname{Col} \partial_{k-1,k}.$$

In particular, $\operatorname{Nul} \partial_{-1,0} = \mathbb{Z}_2^M$ and $\operatorname{Col} \partial_{2,3} = \operatorname{Col} 0 = \{\mathbf{0}\}$ is the trivial subspace. □

It allows us to introduce the vector space of $k$-features of $\mathbb{M}$.

**Definition 6** (Homology Groups)**.** *Consider a finite three dimensional set* $\mathbb{M} = \{x_0, x_1, x_2 \ldots\}$*. The* $k$-th Homology Group *of* $\mathbb{M}$*.*

$$H_k(\mathbb{M}) := \operatorname{Nul} \partial_{k-1,k} / \operatorname{Col} \partial_{k,k+1},$$

*where*

$$\begin{aligned} \dim H_k(\mathbb{M}) &= \dim \operatorname{Nul} \partial_{k-1,k} - \dim \operatorname{Col} \partial_{k,k+1} \\ &= \ell_k(\mathbb{M}) - \dim \operatorname{Col} \partial_{k-1,k} - \dim \operatorname{Col} \partial_{k,k+1}. \end{aligned}$$

*In particular,*

$$H_0(\mathbb{M}) = \operatorname{Nul} \partial_{-1,0} / \operatorname{Col} \partial_{0,1} = \mathbb{Z}_2^M / \operatorname{Col} \partial_{0,1},$$

*and*

$$H_2(\mathbb{M}) = \operatorname{Nul} \partial_{1,2} / \{\mathbf{0}\} = \operatorname{Nul} \partial_{1,2}.$$

*Moreover,* $H_k(\mathbb{M}) = \{0\}$ *for all integer* $k \neq 0, 1, 2$*.*

**Proposition 1.** *The elements in* $H_k(\mathbb{M})$ *are equivalence classes* $\boldsymbol{\sigma}$ *obtained from elements* $\sigma \in \mathbb{Z}_2^{\ell_k(\mathbb{M})}$ *satisfying that* $\partial_{k-1,k}\, \sigma = \mathbf{0}$ *and where each equivalence class is defined by a set*

$$\boldsymbol{\sigma} := \left\{ \gamma \in \operatorname{Nul} \partial_{k-1,k} : \gamma = \sigma + \partial_{k,k+1}(\delta) \text{ for some } \delta \in \mathbb{Z}_2^{\ell_{k+1}(\mathbb{M})} \right\}.$$

Observe, that a $k$-feature $\boldsymbol{\sigma} \in H_k(\mathbb{M})$ is a "packet" of simplices $\sigma$ that share its faces, that is $\partial_{k-1,k}(\sigma) = 0$ (and it can be seen as a connected component of the intersection of $\mathbb{M}$ with a $k$-dimensional linear subspace of $\mathbb{R}^3$), plus the vector subspace $\operatorname{Col} \partial_{k,k+1}$. Since $H_k(\mathbb{M})$ is a vector space it is possible to find a basis of $\beta_k(\mathbb{M}) := \dim H_k(\mathbb{M})$-vectors (or $k$-features).

**Proposition 2.** *There exists* $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{\beta_k(\mathbb{M})}$ *linear independent vectors in* $H_k(\mathbb{M})$ *such that it generates the whole vector space, that is,*

$$H_k(\mathbb{M}) = \operatorname{span}\{\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{\beta_k(\mathbb{M})}\}.$$

Moreover, we can easily extend the order relation of the vectors is $\mathbb{Z}_2^{\ell_k(\mathbb{M})}$ to the vectors in $H_k(\mathbb{M})$, and hence we will assume that the basis vector is ordered as follows $\boldsymbol{\sigma}_1 < \cdots < \boldsymbol{\sigma}_{\beta_k(\mathbb{M})}$. This basis vector (or $k$-features) characterizes the $k$-th homological group associated to $\mathbb{M}$.

**Proposition 3.** *Each* $\boldsymbol{\gamma} \in H_k(\mathbb{M})$ *can be written as* $\boldsymbol{\gamma} = \sum_{\ell=1}^{\beta_k(\mathbb{M})} \xi_\ell \boldsymbol{\sigma}_\ell$ *where* $\xi_\ell \in \mathbb{Z}_2$ *for* $1 \leq \ell \leq \beta_k(\mathbb{M})$*.*

### 1.2.3 Persistent Homology

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

$$\emptyset \subset \mathcal{K}_0 \subset \cdots \subset \mathcal{K} = \mathcal{S}(\mathbb{M}).$$

It is constructed as an approximation of $\mathcal{S}(\mathbb{M})$ which usually is computationally intractable.

Given a non decreasing finite sequence $(r_i)_{i=0}^n,\ n > 1$, we can construct the nested non decreasing sequence of sets representing the filtration of $\mathcal{S}(\mathbb{M})$. For a given $r = r_j,\ 0 \leq j \leq n$ we have

$$S_r(\mathbb{M}) := \{S_r^{(0)}(\mathbb{M}), S_r^{(1)}(\mathbb{M}), S_r^{(2)}(\mathbb{M})\} \subset \mathcal{S}(\mathbb{M})$$

Proceeding in a similar way as previously we can construct vector spaces over the finite field $\mathbb{Z}_2$ obtaining

$$\mathbb{Z}_2[S_r^{(0)}(\mathbb{M})] = \mathbb{Z}_2[S_0(\mathbb{M})] \equiv \mathbb{Z}_2^M,$$

and for $k \geq 1$ we have that $\mathbb{Z}_2[S_r^{(k)}(\mathbb{M})] \equiv \mathbb{Z}_2^{\ell_{k,r}(\mathbb{M})} \subset \mathbb{Z}_2[S_k(\mathbb{M})] \equiv \mathbb{Z}_2^{\ell_k(\mathbb{M})}$ is a linear subspace that depends on $\alpha > 0$, for $k \geq 1$. Moreover, we have a vector space

$$H_{k,r}(\mathbb{M}) := \mathrm{Nul}\,\partial_{k-1,k}^r / \mathrm{Col}\,\partial_{k,k+1}^r$$

of dimension $\beta_{k,r}(\mathbb{M}) := \dim H_{k,r}(\mathbb{M})$, for each integer number $k$. In particular, $H_{0,r}(\mathbb{M}) = \mathbb{Z}_2^M / \mathrm{Col}\,\partial_{0,1}^r$, and $H_{2,r}(\mathbb{M}) = \mathrm{Ker}\partial_{1,2}^r$. Also, $H_{k,r}(\mathbb{M}) = \{0\}$ for every integer number $k \neq 0, 1, 2, 3$. Thus, we only need to compute

$$\begin{aligned} H_{0,r}(\mathbb{M}) &= \mathbb{Z}_2^M / \mathrm{Col}\,\partial_{0,1}^r, \\ H_{1,r}(\mathbb{M}) &= \mathrm{Nul}\,\partial_{0,1}^r / \mathrm{Col}\,\partial_{1,2}^r, \\ H_{2,r}(\mathbb{M}) &= \mathrm{Ker}\partial_{1,2}^r \end{aligned}$$

for each $r > 0$. In a similar way as above, we have a basis for each of this three vector spaces (representative $k$-features at r-scale), namely

$$H_{k,r}(\mathbb{M}) = \mathrm{span}\left\{\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{\beta_{k,r}(\mathbb{M})}\right\},$$

for $k = 0, 1, 2$.

To determine if two surfaces are similarly homological at scale, we can study the behaviour of the basis functions of the vector spaces $H_{k,r}(\mathbb{M})$ depending on $r$. To this end, we introduce the notion of *birth* and *death* point of a $k$-basis vector at $\alpha$-scale $\boldsymbol{\gamma} \in H_{k,r}(\mathbb{M})$ as follows.

**Definition 7.** *The birth point $a_k(\boldsymbol{\gamma})$ and death point $b_k(\boldsymbol{\gamma})$ of a $k$-feature at $r$-scale $\boldsymbol{\gamma} \in H_{k,r}(\mathbb{M})$ are defined by*

$$\begin{aligned} a_k(\boldsymbol{\gamma}) &= \inf\{r > 0 : \boldsymbol{\gamma} \in H_{k,r}(\mathbb{M})\} \\ b_k(\boldsymbol{\gamma}) &= \sup\{r > 0 : \boldsymbol{\gamma} \in H_{k,r}(\mathbb{M})\} \end{aligned}$$

For each $k = 0, 1, \ldots$ and $j = 1, 2, \ldots, m$ fixed we have

$$H_{k,r_j}(\mathbb{M}) = \text{span}\left\{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{\beta_k^{(j)}}\right\},$$

Let $N_k$ $(0 \leq k \leq 2)$ be the number of vectors in the set $\left\{\boldsymbol{\gamma} \in H_{k,r_j}(\mathbb{M}) : \text{ for some } 1 \leq j \leq m\right\}$ that can be ordered as

$$\boldsymbol{\gamma}_1 < \boldsymbol{\gamma}_2 < \cdots < \boldsymbol{\gamma}_{N_k}.$$

Each $k$-feature $\boldsymbol{\gamma}_\upsilon$ $(1 \leq \upsilon \leq N_k)$ can be represented over the $\mathbb{R}^2$ plan as a single point with coordinates $(a_k^{(\upsilon)}, b_k^{(\upsilon)})$ where

$$a_k^{(\upsilon)} = a_k(\boldsymbol{\gamma}_\upsilon) = \min_{0 \leq j \leq m}\left\{r_j : \boldsymbol{\gamma}_\upsilon \in H_{k,r_j}(\mathbb{M})\right\},$$
$$b_k^{(\upsilon)} = b_k(\boldsymbol{\gamma}_\upsilon) = \max_{0 \leq j \leq m}\left\{r_j : \boldsymbol{\gamma}_\upsilon \in H_{k,r_j}(\mathbb{M})\right\}$$

The finite collection of points obtained from the $k$-features is called a *Persistence Diagram at $k$-level*. The $k$-level Persistence Diagram associated to the partition $\mathcal{P} := \{r_j\}_{j=1}^m$ is given by:

$$\mathcal{PD}_k(\mathbb{M}, \mathcal{P}) := \left\{(a_k^{(\upsilon)}, b_k^{(\upsilon)}) : 1 \leq \upsilon \leq N_k\right\}$$

for $k = 0, 1, 2$.

**Definition 8.** *Given a surface $\mathbb{M}$ and a partition $\mathcal{P}$, we define its set of persistence diagrams noted $\mathcal{PD}$ as:*

$$\mathcal{PD} = \{\mathcal{PD}_0(\mathbb{M}, \mathcal{P}), \mathcal{PD}_1(\mathbb{M}, \mathcal{P}), \mathcal{PD}_2(\mathbb{M}, \mathcal{P})\}$$

# 2 Advanced Parametric Modes Clustering

In this chapter, we propose the use of a topological metric based on the persistent homology, enabling efficient surfaces classification, and ordering the elastodynamics eigenmodes to construct parametric reduced bases.

## 2.1 Introduction

Modal analysis is widely used for addressing NVH –Noise, Vibration and Hardness– in automotive engineering. The so-called principal modes constitutes an orthogonal basis, obtained from the eigenvectors related to the dynamical problem. When this basis is used for expressing the displacement field of a dynamical problem, the model equations become uncoupled. Moreover, a reduced basis can be defined according to the eigenvalues magnitude, leading to an uncoupled reduced model, specially appealing when solving large dynamical systems. However, engineering looks for optimal designs and therefore it focuses on parametric designs needing the efficient solution of parametric dynamical models. Solving parametrized eigenproblems remains a tricky issue, and therefore, non-intrusive approaches are privileged. In that framework, a reduced basis consisting of the most significant eigenmodes is retained for each choice of the model parameters under consideration. Then, one is tempted to create a parametric reduced basis, by simply expressing the reduced basis parametrically by using an appropriate regression technique. However an issue remains, that limits the direct application of the just referred approach, the one related to the basis ordering. In order to order the modes before interpolating them, different techniques were proposed in the past, being the Modal Assurance Criterion –MAC– one of the most widely used. We propose an alternative technique that instead of operating at the eigenmodes level, classify the modes with respect to the deformed structure shapes that the eigenmodes induce.

Linear structural solid dynamics [50] expressed in the time domain results in the linear system of second order ordinary differential equations

$$\mathbf{M}\frac{d^2\mathbf{U}(t)}{dt^2} + \mathbf{C}\frac{d\mathbf{U}(t)}{dt} + \mathbf{K}\mathbf{U}(t) = \mathbf{F}(t), \tag{1}$$

with the mass, damping and stiffness matrices given by $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ respectively, $\mathbf{U}$ the vector that contains the nodal displacements and $\mathbf{F}$ the applied nodal forces. Its time integration can be performed by using any well experienced state of the art discretization technique, as [51] or [49].

In what follows we will omit the damping term, that results from the fact of assuming a proportional damping, that expresses it as a combination of the mass and stiffness contributions.

To enhance the integration efficiency, mass lumping is usually considered, leading to a mass diagonal matrix. Model analysis looks also for enhancing the solution efficiency by decoupling the motion equation. For that purpose, the last extracts the basis $\{\phi_1,\ \phi_2, \ldots,\ \phi_{\mathtt{N}}\}$ ($N$ being the problem size, i.e. the number of degrees of

freedom) by solving the eigenproblem

$$\left(-\omega^2\mathbf{M}+\mathbf{K}\right)\phi=\mathbf{0}, \tag{2}$$

associated with the dynamical problem expressed in the Fourier space,

$$\left(-\omega^2\mathbf{M}+\mathbf{K}\right)\mathcal{U}=\mathcal{F}, \tag{3}$$

where $\mathcal{U}$ and $\mathcal{F}$ refer to the Fourier transform of the nodal displacement $\mathbf{U}$ and forces $\mathbf{F}$.

The eigenmodes $\phi_i$, $i=1,\cdots,N$ define an orthogonal basis, normalized with respect to the mass matrix, i.e.

$$\phi_i^T\mathbf{M}\phi_j=\delta_{ij}, \tag{4}$$

with $\delta$ the Kroenecker delta, and

$$\phi_i^T\mathbf{K}\phi_j=\kappa_i\delta_{ij}. \tag{5}$$

With $\mathbf{P}$ the matrix composed by the eigenmodes, i.e. $\mathbf{P}=(\phi_1,\cdots,\phi_N)$, the matrix form of the previous expressions read $\mathbf{P}^T\mathbf{M}\mathbf{P}=\mathbf{I}$ and $\mathbf{P}^T\mathbf{K}\mathbf{P}=\mathbb{K}$, with $\mathbf{I}$ the identity matrix and $\mathbb{K}$ the diagonal matrix with entries $\mathbb{K}_{ii}=\kappa_i$.

In the modal basis $\mathbf{U}=\mathbf{P}\varphi$, and the dynamical problem reads

$$\mathbf{I}\frac{d^2\varphi(t)}{dt^2}+\mathbb{K}\varphi(t)=\mathbf{P}^T\mathbf{F}(t), \tag{6}$$

that constitutes a system of `N` uncoupled second order ordinary differential equations.

The main limitation of modal analysis is the lack of validity of such basis in the case of parametric models. In the case of parametrized dynamical systems, with the model parameters grouped in vector $\mu$, the model matrices will depends on those parameters, i.e. $\mathbf{M}(\mu)$ and $\mathbf{K}(\mu)$. The solution of parametric eigenproblems remains a tricky issue.

When one is not really interested in the transient regime, but much more in the forced regime, harmonic analysis represents a valuable route. The so-called Proper Generalized Decomposition—PGD— enables considering the frequency as a model extra-parameter as well as addressing general (non-proportional) damping and non-linear dynamics, under the stringent real-time constraint, with even the inclusion of model parameters as extra-coordinates [18, 17, 20].

However, certain applications need accurate transient responses, and in that case the formulation and solution of the dynamical problem in the time domain is retained. Three routes are usually considered:

1. The previously referred mass lumping that transforms the so-called consistent mass matrix into its diagonal counterpart, facilitating an explicit integration;

2. In the context of model order reduction—MOR—, a Proper Orthogonal Decomposition—POD— based reduced order modeling operating in the time domain has been proposed in [48]. Ladeveze and coworkers proposed an extension of their radial approximation [46] for addressing mid-frequency dynamics, the so called *variational theory of complex rays* [47]. A PGD formulation for constructing a parametric transfer function has been considered [21], allowing efficient solutions of transient dynamics operating in the time-domain. On the other hand, the separation of variables, at the heart of PGD [5], was extensively employed in the harmonic domain for solving multi-parametric dynamics, and was successfully extended to the non-linear case, and then combined with modal analysis [18, 17, 20].

3. Modal analysis is one the most widely used techniques for solving dynamical problems. Other than the benefits in the time integration, due to the dynamical system decoupling, the eigenmodes benefit of a physical interpretation, of great interest for the designer or structural analyst. However, when considering parametric models as it is always the case during the design stage, when the material and geometry are not totally defined, the dynamical modes depends on those parameters as previously discussed. Having a surrogate model expressing the parametric evolution of the eigenmodes is of great interest. Constructing those surrogate models is nowadays quite mature, by using usual and advanced nonlinear regressions [19], the last making use of sparsity and appropriate regularisation for operating in high-dimensional settings, while keeping as reduced as possible the number of data (eigenproblems solution), and leading to rich enough (nonlinear) regressions while avoiding overfitting. Here the trickiest issue is not the regression construction, but the fact of ordering the different eigenmodes involved in the modal basis for each parameters choice, in order to created $N$ clusters (or less in the reduced case), and putting in each one a mode of each modal basis, such that modes in each cluster remain close (in a certain metrics). The main issue remains the metric to be use to successfully and efficiently accomplishing such clustering. In general such clustering is performed by operating at the level of the eigenmodes, in the associated vector space, by using for example the Modal Assurance Criterion—MAC—[52] that proceed comparing the modes resulting from each eigenproblem, by using the usual scalar product (modes similar to a given one should remain quite collinear).

We will focuses on the techniques based on modal analysis. As just described, usual techniques operate at the eignemodes level, defined in a vector space. When operating in high dimensional parametric spaces, sparsely sampled, the matrices involved in the resulting eigenproblem can vary a lot from one choice of the parameters to another, and consequently the scalar product criterion at the basis of the MAC could fail. On the other hand, the fact of proceeding in a vector space needs to carefully address the expression of the different modes by considering the same frame for all the analyzed mechanical systems.

For alleviating those difficulties, we propose an alternative technique that instead of operating at the eigenmodes level, classify the modes with respect to the deformed structures shape that the eigenmodes produce, taking advantage from the invariance property of topology. Thus, we are employing a metric able to compare shapes, more that a metric for comparing the vectors (eigenmodes) that produced those shapes, the last being more intrinsic and inheriting invariance features. Moreover, in the present case study, eigenmodes are heterogeneous in the sense that they involve displacements and rotations, whereas the associated deformed surfaces are purely geometrical.

## 2.2 Methodology

Here, we address the classification of a series of modal basis related to the eigenmodes of a thin structure equipped with a mesh consisting of shell elements, with displacement and rotation degrees of freedom at each node of the mesh.

The thickness of the structural part varies, with its consequent effect on the mass and stiffness matrices (damping is assumed proportional) and consequently on the eigenvalues and eigenvectors, the former defining the number of modes to be retained in the reduced basis. In the present study, the six rigid modes representing the whole structure translation (three modes) and rotation (three modes) will be discarded and among the remaining pairs eigenvalue-eigenvector, the most relevant six eigenvectors (corresponding to the six highest eigenvalues) retained in the reduced basis related to each choice of the model parameter (the thickness).

These six modes related to each structure (related to a thickness value) define a reduced basis that one would like render parametric. However, prior to construct a regression able to define the reduced basis for each possible choice of the parameter (thickness) one should classify the six eigenmodes of each reduced basis associated to each structure, into six clusters.

This task is compulsory to facilitate the interpolation in the parametric space and also to attach a physical sense to those modes. One could imagine that for a given thickness the most relevant deformation mode could be related to the extension whereas for another choice of the thickness the most relevant deformation mode could be the bending. In such a case, one prefers create a cluster grouping similar deformation modes, to evaluate how each of them depends from the parameter from one side, and from the other to facilitate the subsequent construction of the parametric modal reduced basis.

To preform such a clustering, we must employ an appropriate metric to compare those modes. In general this comparison was traditionally performed by comparing the eigenmodes within the vector space to which they belongs. In the present work, as announced previously, we prefer applying the deformation mode to the reference (undeformed) structure, that is, applying the eigenmode at the nodes location in the reference structure for obtaining the deformed structure related to each mode of each structure configuration (thickness) and then cluster the resulting deformed structures with respect to their shape.

In the remaining part of the present section we will described the available data and its organization, and then, all the concepts enabling the use of a metric applying on the shapes, based on the employ of persistent homology, at the heart of the TDA.

## 2.3 Data description

As the different analyzed structures are shells of different thicknesses, from now on we will describe these structures by their surfaces, each equipped of a nodes distribution and the associated mesh.

In this study we consider a collection of $\mathcal{M} = 102$ surfaces corresponding to the effect of a given deformation mode on the reference undeformed surface, as Fig. 4 shows.

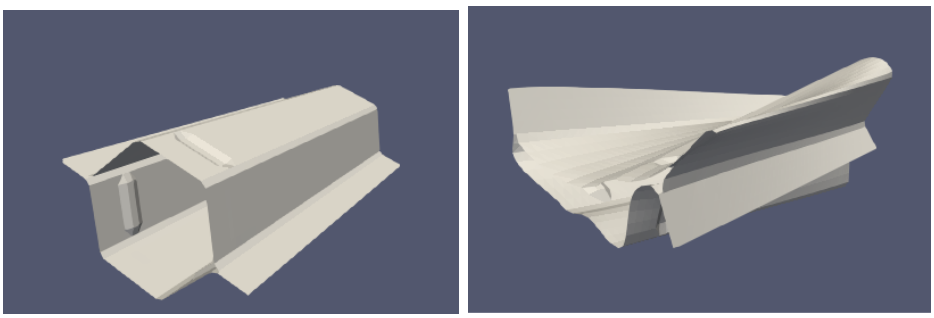


Figure 4: Reference undeformed structure (left) and its deformed counterpart (right) when a given deformation mode applied on the reference one

Each surface $\mathbb{M}_r$, $r = 1, \ldots, \mathcal{M}$, is equipped with a mesh associated with $\mathcal{N} = 3636$ nodes, each node described by $\mathbf{x}_n \equiv (x_n, y_n, z_n)$, $n = 1, \ldots, \mathcal{N}$ and $\mathbf{x}_n \in \mathbb{R}^3$, all them making user of the same common coordinates frame.

The deformed structures consists of the nodes and elements resulting from the reference one by applying the associated deformation mode. There is neither nodes redistribution nor refinement in the deformed surfaces. Figure 5 depicts the reference surface and the nodes distribution on it, from which all the deformed structures with their associated nodal distribution and deformed mesh will result. It is important to note that the undeformed and deformed meshes (elements connectivity) remain unchanged facilitating the use of the proposed metrics discussed later.

The $\mathcal{M} = 102$ surfaces are associated to 17 different structures, each one of them having a different thickness, with the consequent effect on the mass and stiffness matrices, and therefore on the resulting eigenfrequencies and eigenmodes. For each of the 17 structures, the 6 most significant deformation modes (related to the six highest eigenvalues with the rigid modes excluded) are retained. As mentioned, by applying this $17 \times 6$ deformation modes to the original undeformed reference surface, the $\mathcal{M} = 102$ deformed surfaces result.

The data, is structured as a table, with in each row the six deformation modes

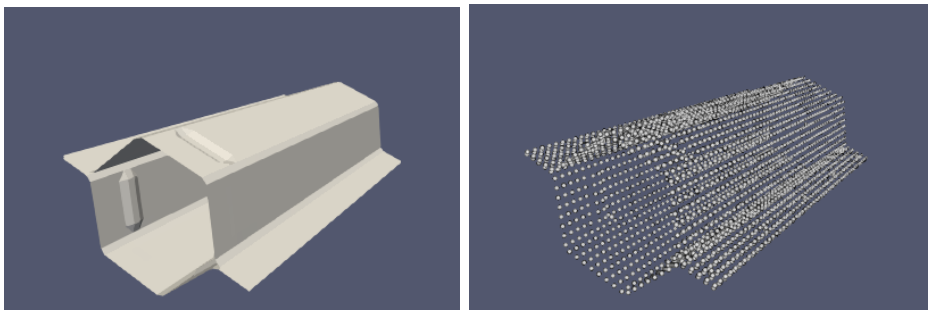


Figure 5: Reference surface (left) and nodes distribution on it (right)

related to a given structure (with its own thickness), as follows

$$\begin{array}{cccccc} \mathbb{M}_1 & \mathbb{M}_2 & \mathbb{M}_3 & \mathbb{M}_4 & \mathbb{M}_5 & \mathbb{M}_6 \\ \mathbb{M}_7 & \mathbb{M}_8 & \mathbb{M}_9 & \mathbb{M}_{10} & \mathbb{M}_{11} & \mathbb{M}_{12} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbb{M}_{97} & \mathbb{M}_{98} & \mathbb{M}_{99} & \mathbb{M}_{100} & \mathbb{M}_{101} & \mathbb{M}_{102} \end{array}$$

Our main aim in what follows is ordering the elements in the columns, such that each column will represent a similar deformation behavior.

## 2.4 On the surface topology

Consider a data-set $\mathbb{M}$ related to a given deformed surface defined from its $\mathcal{N}$ nodes, all them in $\mathbb{R}^3$. We are interested in extracting the geometric features of $\mathbb{M}$ and how they are distributed across the different spatial scales.

### 2.4.1 Geometric features

In order to describe the geometry of the data-set $\mathbb{M}$ we first identify four types of geometrical features associated with $\mathbb{M}$:

- A vertex $[\mathbf{x}_m]$ is generated by an individual point $\mathbf{x}_m \in \mathbb{M}$;
- A segment $[\mathbf{x}_m, \mathbf{x}_n]$ joins two vertex $[\mathbf{x}_m],\ [\mathbf{x}_n] \in \mathbb{M}$

$$[\mathbf{x}_m, \mathbf{x}_n] := \left\{ \mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \lambda \mathbf{x}_m + (1-\lambda)\mathbf{x}_n \text{ where } 0 \leq \lambda \leq 1 \right\};$$

- A triangle is generated by three different vertex $[\mathbf{x}_m],\ [\mathbf{x}_n],\ [\mathbf{x}_l] \in \mathbb{M}$, such that $\mathbf{x}_m - \mathbf{x}_n$ and $\mathbf{x}_m - \mathbf{x}_l$ are linearly independent, and then:

$$[\mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_l] := \left\{ \mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \lambda_m \mathbf{x}_m + \lambda_n \mathbf{x}_n + \lambda_l \mathbf{x}_l \right\},$$

  where $\lambda_m,\ \lambda_n$ and $\lambda_l$ are the barycentric coordinates, with $\lambda_m + \lambda_n + \lambda_l = 1$;

- A tetrahedron is generated by four different vertices $[\mathbf{x}_m]$, $[\mathbf{x}_n]$, $[\mathbf{x}_l]$, $[\mathbf{x}_p] \in \mathbb{M}$, such that $\mathbf{x}_m - \mathbf{x}_n$, $\mathbf{x}_m - \mathbf{x}_l$ and $\mathbf{x}_m - \mathbf{x}_p$ are linearly independent, and then:

$$[\mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_l, \mathbf{x}_p] := \left\{ \mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \lambda_m \mathbf{x}_m + \lambda_n \mathbf{x}_n + \lambda_l \mathbf{x}_l + \lambda_p \mathbf{x}_p \right\},$$

  where $\lambda_m$, $\lambda_n$, $\lambda_l$ and $\lambda_p$ are the barycentric coordinates, with $\lambda_m + \lambda_n + \lambda_l + \lambda_p = 1$.

The vertices represent the dimension-0 features, segments the dimension-1 features, triangles the dimension-2 features and tetrahedron dimension-3 features.

### 2.4.2 Features filtration

To describe the appearance and disappearance of the features of $\mathbb{M}$ across different scales, we consider the so-called *Alpha Filtration*. For that purpose an interval $[\alpha_{\min}, \alpha_{\max}]$ is considered. It reflects the smallest features scale, in our case $\alpha_{\min} = 0$ (the vertex) and the largest one $\alpha_{\max}$ representing the largest distance between points of $\mathbb{M}$.

The features of $\mathbb{M}$ considered here are elements of the *Simplicial complex* associated to $\mathbb{M}$, noted $S(\mathbb{M})$, and constructed from the finite cells of the the Delaunay Triangulation of the set of points in $\mathbb{M}$. The elements of $S(\mathbb{M})$ are the geometric features of $\mathbb{M}$ i.e. tetrahedrons, triangles, edges and vertices.

In order to describe efficiently the elements of the *Simplicial complex* $S(\mathbb{M})$, and map a set of $d$-dimensional simplices to a set of $(d-1)$-dimensional simplices, we construct a finite sequence of sets of $d$-dimensional simplices, denoted by $S_d(\mathbb{M})$, $d = 0, 1, 2, 3$.

Set $S_0(\mathbb{M}) := \{[x_m] : x_m \in \mathbb{M}\}$, then the simplices in $S_d(\mathbb{M})$ for each $d = 1, 2, 3$ are obtained from the simplices in $S_{d-1}(\mathbb{M})$ taking into account the following two properties:

1. For every simplex in $S_d(\mathbb{M})$, the $(d-1)$-dimensional simplices forming it are in $S_{d-1}(\mathbb{M})$ (e.g. a triangle is in $S_2(\mathbb{M})$ and its three edges are in $S_1(\mathbb{M})$);

2. If two simplices in $S_d(\mathbb{M})$ have a common element $\sigma$, then there exists $0 \leq l \leq (d-1)$ such that $\sigma \in S_l(\mathbb{M})$.

Given the scale values $(\alpha_j)_{j=0}^m$, the *Alpha Filtration* is then a non decreasing sequences describing the evolution of the features of the simplicial complex $S(\mathbb{M})$ at each scale $\alpha_j$, and computed as follows:

- First, the filtration value of each tetrahedron is computed as the circumradius of the tetrahedron if its circumsphere is empty, and as the minimum of the filtration values of the triangles that are within the circumsphere otherwise.

- Similarly, the filtration value of each triangle is computed as the circumradius of the triangle if the circumcircle is empty, and as the minimum of the filtration values of the segments that are within the circumcircle otherwise.

- Then, the filtration value of each segment is computed as its circumradius.
- Finally, the filtration value of the vertices is set to 0.

The discrete values used for the radii are the $\alpha_j$, and all simplices that have a filtration value larger than $\alpha_{\max}$ are discarded.

The time complexity of the algorithm is $\mathcal{O}(n^2)$. The choice of the *Alpha Filtration* was motivated by its relatively much smaller size compared to other filtrations. A detailed definition and implementation are provided in [26] and [75].

### 2.4.3 Persistence diagrams

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* $H_k(\mathbb{M})$, where $k = 0, 1, 2, 3$ is the homology dimension.

The elements of a *Homology Group* $H_k(\mathbb{M})$ are classes of chains, which are unions of simplices $\sigma \in S_k(\mathbb{M})$, that is, simplices sharing faces, edges or vertices. It can be seen as a connected component of the intersection of $\mathbb{M}$ with a $k$-dimensional linear subspace of $\mathbb{R}^3$. The use of *Homology Groups* allows to perform algebraic operations over their elements.

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* $\alpha$, $H_k^\alpha(\mathbb{M})$. It represents the classes of simplices as described previously, but taken from $S_k^\alpha(\mathbb{M})$. That is, the elements of $S_k(\mathbb{M})$ with a filtration value lesser that $\alpha$.

This approach is known as the *Persistent Homology*. It allows to quantify the appearance and disappearance of the features across the different scales and dimensions.

- The birth scale $b_\gamma$ of the feature $\gamma$ at homology $k$

$$b_\gamma = \min_{0 \leq j \leq m} \{\alpha_j : \gamma \in H_k^{\alpha_j}\}$$

- The death scale $d_\gamma$ of the feature $\gamma$ at homology $k$

$$d_\gamma = \max_{0 \leq j \leq m} \{\alpha_j : \gamma \in H_k^{\alpha_j}\}$$

The birth scale represents the value at which the feature appeared in the filtration by combining lower dimensional simplices to form it. Conversely, the death scale represents the value at which the feature disappeared in the filtration by being combined into a higher dimensional feature. For example, if a vertex is part of segment, then the death scale of the vertex is exactly the birth scale of the associated segment.

Note that, by definition, vertices always have a zero birth scale, while tetrahedrons always have an infinite death scale (in the numerical results, we removed the

infinite values for computation purposes). Given that our data points $\mathbb{M}$ are embedded in $\mathbb{R}^3$, we will only track up to the dimension-2 features, thus the definition of $S(\mathbb{M})$ with $k = 0, 1, 2$. More generally, if the data points are embedded in a $d$-dimensional manifold, the persistent homology can be computed up to dimension $d - 1$.

Finally, the persistence of the features throughout the scales can be represented by the so-called *Persistence Diagram* of $\mathbb{M}$, defined at dimension-$k$ from

$$\mathcal{PD}_k(\mathbb{M}) = \{(b_\gamma, d_\gamma) : \gamma \in H_k\},$$

where $b_\gamma$ and $d_\gamma$ are the birth and death scales of a feature $\gamma$ at homology $k$.

The surface $\mathbb{M}$ persistence diagrams $\mathcal{PD}(\mathbb{M})$ reads

$$\mathcal{PD}(\mathbb{M}) = \{\mathcal{PD}_0(\mathbb{M}), \mathcal{PD}_1(\mathbb{M}), \mathcal{PD}_2(\mathbb{M})\}.$$

### 2.4.4 Illustrating the concepts on an example

We illustrate the computational aspects of the *Alpha Filtration* on a simple example, consisting of six points in $\mathbb{R}^3$, as shown in Figure 6.

$$\begin{aligned}\mathbb{M} = \{&\mathbf{x}_0 = (1.1, 0.9, 0), \mathbf{x}_1 = (0.1, 0, 1), \mathbf{x}_2 = (0, 0, 0),\\ &\mathbf{x}_3 = (0, 0.1, 1), \mathbf{x}_4 = (0.9, 1.1, 0), \mathbf{x}_5 = (0, 1, 0)\}\end{aligned}$$

The filtration values are computed and presented below in Table 3:

| $\alpha$ | $S_0^\alpha$ | $S_1^\alpha$ | $S_2^\alpha$ | $S_3^\alpha$ |
|---|---|---|---|---|
| 0.00 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | - | - | - |
| 0.50 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3]$ | - | - |
| 2.00 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3], [x_0, x_4]$ | - | - |
| 20.50 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3], [x_0, x_4]$ $[x_0, x_2], [x_4, x_5]$ | - | - |
| 45.25 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3], [x_0, x_4]$ $[x_0, x_2], [x_4, x_5]$ $[x_1, x_2], [x_3, x_5]$ | - | - |
| 50.00 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3], [x_0, x_4]$ $[x_0, x_2], [x_4, x_5]$ $[x_1, x_2], [x_3, x_5]$ $[x_2, x_5]$ | - | - |
| 50.02 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3], [x_0, x_4]$ $[x_0, x_2], [x_4, x_5]$ $[x_1, x_2], [x_3, x_5]$ $[x_2, x_5], [x_2, x_4]$ | - | - |

| $\alpha$ | $S_0^\alpha$ | $S_1^\alpha$ | $S_2^\alpha$ | $S_3^\alpha$ |
|---|---|---|---|---|
| 64.73 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3], [x_0, x_4]$ $[x_0, x_2], [x_4, x_5]$ $[x_1, x_2], [x_3, x_5]$ $[x_2, x_5], [x_2, x_4]$ $[x_1, x_5]$ | $[x_1, x_2, x_5], [x_1, x_3, x_5]$ | - |
| 70.64 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3], [x_0, x_4]$ $[x_0, x_2], [x_4, x_5]$ $[x_1, x_2], [x_3, x_5]$ $[x_2, x_5], [x_2, x_4]$ $[x_1, x_5], [x_0, x_1]$ $[x_3, x_4]$ | $[x_1, x_2, x_5], [x_1, x_3, x_5]$ $[x_0, x_1, x_2], [x_3, x_4, x_5]$ | - |
| 71.38 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3], [x_0, x_4]$ $[x_0, x_2], [x_4, x_5]$ $[x_1, x_2], [x_3, x_5]$ $[x_2, x_5], [x_2, x_4]$ $[x_1, x_5], [x_0, x_1]$ $[x_3, x_4], [x_1, x_4]$ | $[x_1, x_2, x_5], [x_1, x_3, x_5]$ $[x_0, x_1, x_2], [x_3, x_4, x_5]$ $[x_0, x_1, x_4], [x_1, x_3, x_4]$ | - |
| 71.55 | $[x_0], [x_1], [x_2]$ $[x_3], [x_4], [x_5]$ | $[x_1, x_3], [x_0, x_4]$ $[x_0, x_2], [x_4, x_5]$ $[x_1, x_2], [x_3, x_5]$ $[x_2, x_5], [x_2, x_4]$ $[x_1, x_5], [x_0, x_1]$ $[x_3, x_4], [x_1, x_4]$ | $[x_1, x_2, x_5], [x_1, x_3, x_5]$ $[x_0, x_1, x_2], [x_3, x_4, x_5]$ $[x_0, x_1, x_4], [x_1, x_3, x_4]$ $[x_1, x_2, x_4], [x_1, x_4, x_5]$ | $[x_0, x_1, x_2, x_4]$ $[x_1, x_2, x_4, x_5]$ $[x_1, x_3, x_4, x_5]$ |

Table 3: *Alpha Filtration* of $\mathbb{M}$

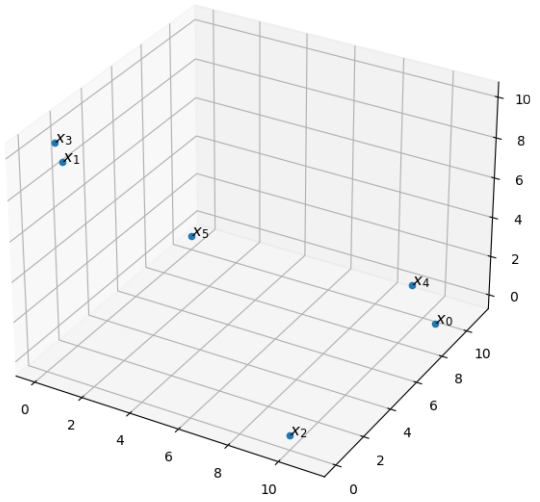



Figure 6: Example of data points $\mathbb{M}$

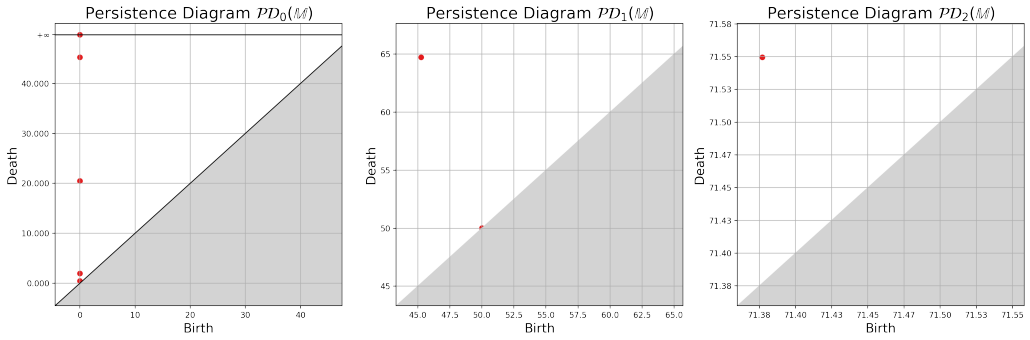



Figure 7: Persistence Diagrams $\mathcal{PD}(\mathbb{M})$

We can then track the *birth* and *death* of the features and compute the persistence diagrams $\mathcal{PD}(\mathbb{M})$, as shown in Fig. 7.

### 2.4.5 Matching persistence diagrams

Consider two data-sets $\mathbb{M}_r$ and $\mathbb{M}_s$ representing two deformed configurations of the same surface. A matching between two persistence diagrams with the same number of features, $\mathcal{PD}_k(\mathbb{M}_r)$ and $\mathcal{PD}_k(\mathbb{M}_s)$, for $k = 0, 1, 2$, is a bijective map $\psi^k$, that reads:

$$\psi^k : \mathcal{PD}_k(\mathbb{M}_r) \longrightarrow \mathcal{PD}_k(\mathbb{M}_s),$$

such that $\forall \gamma = (b, d) \in \mathcal{PD}_k(\mathbb{M}_r)$,

$$\begin{aligned} \psi^k(\gamma) &= (\psi_1^k(b), \psi_2^k(d)) \\ &= (b', d') \in \mathcal{PD}_k(\mathbb{M}_s). \end{aligned}$$

The map $\psi^k$ associates each feature from $\mathcal{PD}_k(\mathbb{M}_r)$ to a feature from $\mathcal{PD}_k(\mathbb{M}_s)$. The *Optimal Matching* between $\mathcal{PD}_k(\mathbb{M}_r)$ and $\mathcal{PD}_k(\mathbb{M}_s)$ is a matching $\hat{\psi}^k$

$$\hat{\psi}^k : \mathcal{PD}_k(\mathbb{M}_r) \longrightarrow \mathcal{PD}_k(\mathbb{M}_s),$$

minimizing the transport cost $\mathcal{C}^k$ to move the features from $\mathcal{PD}_k(\mathbb{M}_r)$ to $\mathcal{PD}_k(\mathbb{M}_s)$:

$$\begin{aligned} \mathcal{C}^k_{\texttt{min}} &= \sum_{\gamma \in \mathcal{PD}_k(\mathbb{M}_r)} \|\gamma - \hat{\psi}^k(\gamma)\|_2 \\ &= \sum_{(b,d) \in \mathcal{PD}_k(\mathbb{M}_r)} \left\| \left( b - \hat{\psi}_1^k(b), d - \hat{\psi}_2^k(d) \right) \right\|_2 \\ &= \sum_{(b,d) \in \mathcal{PD}_k(\mathbb{M}_r)} \sqrt{\left( b - \hat{\psi}_1^k(b) \right)^2 + \left( d - \hat{\psi}_2^k(d) \right)^2}. \end{aligned}$$

When $\mathbb{M}_r$ is the reference surface, and $\mathbb{M}_s$ any deformed surface resulting from $\mathbb{M}_r$, the optimal matching $\hat{\psi}^k$ represents and quantifies the deformation from a topological viewpoint, at each dimension $k = 0, 1, 2$.
We note that, in our case considered here, the diagrams have all been reduced to

their top 3000 persistence values, making the bijective matching possible. In the case of diagrams with different number of points, a partial matching is rather considered. The optimal matching is computed using a combinatorial optimization procedure, where the points in both diagrams are matched while minimizing the transport cost function defined above. A graphical representation of the matching is shown in figure 45.

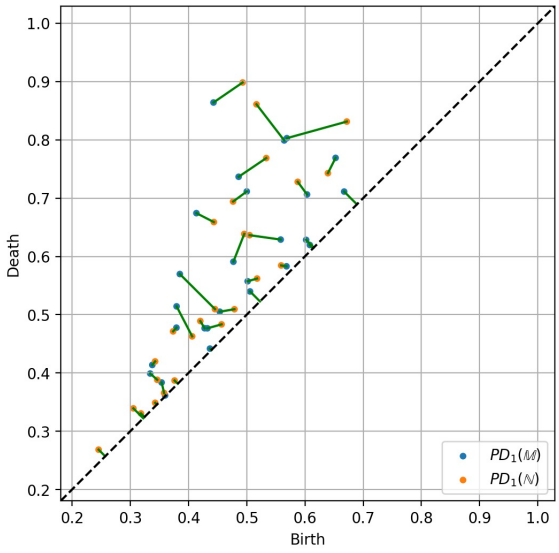



Figure 8: Optimal matching of two persistence diagrams $\mathcal{PD}_1(\mathbb{M})$ and $\mathcal{PD}_1(\mathbb{N})$

### 2.4.6 Multi-scale topological measure of surface deformation

It is now possible to measure the degree of deformation from one data-set to another. For that purpose consider two data-sets $\mathbb{M}_r$ and $\mathbb{M}_s$ representing two deformed states of the same surface, and a finite sequence of $(\alpha_j)_{j=0}^m$. Then, for $k = 0, 1, 2$, the $k$-distance between $\mathcal{PD}_k(\mathbb{M}_r)$ and $\mathcal{PD}_k(\mathbb{M}_s)$ reads

$$W_k\Big(\mathcal{PD}_k(\mathbb{M}_r), \mathcal{PD}_k(\mathbb{M}_s)\Big) = \sum_{(b,d)\in\mathcal{PD}_k(\mathbb{M}_r)} \sqrt{\Big(b - \hat{\psi}_1^k(b)\Big)^2 + \Big(d - \hat{\psi}_2^k(d)\Big)^2},$$

where $\hat{\psi}^k$ is the optimal matching between $\mathcal{PD}_k(\mathbb{M}_r)$ and $\mathcal{PD}_k(\mathbb{M}_s)$. An efficient computation of that distance $W_k$, known as the *Wasserstein Distance*, is performed using the kernel linearisation presented in the Algorithm 2 of reference [45].
The *Multi-Scale Topological Distance* between $\mathbb{M}_r$ and $\mathbb{M}_s$ reads

$$\Omega(\mathbb{M}_r, \mathbb{M}_s) = \sqrt{\omega_0^2 + \omega_1^2 + \omega_2^2},$$

where $\omega_k = W_k\Big(\mathcal{PD}_k(\mathbb{M}_r), \mathcal{PD}_k(\mathbb{M}_s)\Big),\ k = 0, 1, 2.$

### 2.4.7 Comparing topological descriptions of deformed surfaces

Consider now our collection $\{\mathbb{M}_0, \mathbb{M}_1, \ldots, \mathbb{M}_{\mathcal{M}}\}$ of data-sets, consisting of the reference surface $\mathbb{M}_0$ and $\mathcal{M} = 102$ deformed surfaces $\mathbb{M}_r$, $r = 1, \ldots, \mathcal{M}$.
By using the previously defined metric $\Omega$, we can measure each surface deformation with respect to the reference one, such that $\forall r \in \{1, \cdots, \mathcal{M}\}$ we have

$$\Omega_r = \Omega(\mathbb{M}_0, \mathbb{M}_r) = \sqrt{\omega_{0,r}^2 + \omega_{1,r}^2 + \omega_{2,r}^2} \ ,$$

where $\forall r \in \{1, \ldots, \mathcal{M}\}, \forall k \in \{0, 1, 2\}$, we denote

$$\omega_{k,r} = W_k \left( \mathcal{PD}_k(\mathbb{M}_0), \mathcal{PD}_k(\mathbb{M}_r) \right).$$

The measure $\Omega_r$ enables clustering the different surfaces ($6 \times 17$, 17 being the number of structural configurations, each one related to a particular value of the thickness) into 6 clusters.

## 2.5 Modal Assurance Criterion

One of the most popular tools for the quantitative comparison of modal vectors is the *Modal Assurance Criterion* (MAC) [52]. The MAC criterion is a statistical indicator quite sensitive to large differences of the eigenmodes.
In our case, each mode $\mathbb{M}_r$ ($r = 1, \ldots, \mathcal{M}$) is decomposed in its linear and angular components (with respect to the three coordinate axes) resulting in the six vectors $\{\Psi_c^r\}_{1 \le c \le 6}$.
The `MAC` of two surfaces $\mathbb{M}_r$ and $\mathbb{M}_s$, is then computed according to

$$\texttt{MAC}(\mathbb{M}_r, \mathbb{M}_s) = \frac{\sum_{c=1}^{6} \left( \Psi_c^r \cdot \Psi_c^s \right)^2}{\left( \sum_{c=1}^{6} \left( \Psi_c^r \right)^2 \right) \left( \sum_{c=1}^{6} \left( \Psi_c^r \right)^2 \right)}.$$

The `MAC` takes values between 0 (representing no consistent correspondence) and 1 (representing a consistent correspondence). Values larger than 0.9 indicate consistent correspondence whereas small values indicate poor resemblance of the two eigenmodes.
Thus, considering the six modes related to two different structures (with two different thicknesses), it is now possible to compute the so-called MAC matrix $\mathbf{M}$ to compare the modes and identify resemblances or discrepancies.
The MAC matrix $\mathbf{M}$ becomes diagonal dominant when modes are well ordered, whereas the loss of that diagonal dominance informs on eventual permutations. In order to apply the MAC criterion in the case study addressed here, the first reduced modal basis consisting of the six modes related to the first thickness will be compared with the six modes of all the other configurations, the remaining 16 thickness choices.

## 2.6 Topological modes identification

By applying the methodology described in Section 2.4 to the surface data-sets, by first computing the *Alpha Filtration*, we obtain the persistence diagrams $\{\mathcal{PD}_k(\mathbb{M}_r)\}_{r=0}^{\mathcal{M}=102}$, with the ones associated to the reference surface shown in Fig. 9. The multi-scale

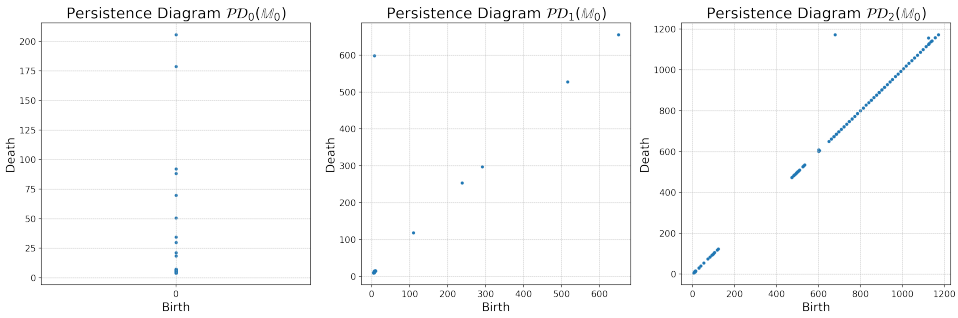



Figure 9: Persistence diagrams associated to the reference surface, $\mathcal{PD}_k(\mathbb{M}_0)$, for $k = 0$ (left), $k = 1$ (center) and $k = 2$ (right)

distance defined in Section 2.4.6 and here associated to each surface $\{\Omega_r\}_{r=1}^{\mathcal{M}=102}$ is then computed, and reported in Table 4.

| Case | 1st surf. | 2nd surf. | 3rd surf. | 4th surf. | 5th surf. | 6th surf. |
|---|---|---|---|---|---|---|
| 01 | 2828 | 3742 | 6012 | 6281 | 7070 | 7314 |
| 02 | 3839 | 4341 | 5160 | 5269 | 8299 | 9530 |
| 03 | 3540 | 4003 | 4702 | 5536 | 6436 | 8023 |
| 04 | 2971 | 3762 | 5883 | 7481 | 7429 | 9314 |
| 05 | 3062 | 3762 | 5437 | 6156 | 9865 | 10307 |
| 06 | 4852 | 5239 | 7294 | 8336 | 8237 | 9585 |
| 07 | 3482 | 4411 | 6095 | 6882 | 9319 | 10627 |
| 08 | 3392 | 3684 | 5903 | 6710 | 9273 | 9438 |
| 09 | 4648 | 5436 | 7986 | 7707 | 10415 | 9406 |
| 10 | 4425 | 4267 | 5583 | 5811 | 9620 | 9163 |
| 11 | 3256 | 3722 | 4782 | 4888 | 5840 | 8064 |
| 12 | 2993 | 3750 | 5551 | 6885 | 7135 | 8474 |
| 13 | 4281 | 4862 | 7127 | 8230 | 8170 | 9687 |
| 14 | 4367 | 5004 | 7140 | 7036 | 8285 | 8008 |
| 15 | 4937 | 5396 | 6484 | 6446 | 9323 | 10031 |
| 16 | 3184 | 3941 | 4907 | 4965 | 7205 | 8869 |
| 17 | 2957 | 3652 | 5652 | 6021 | 7659 | 7571 |

Table 4: Multi-scale topological distance of the six deformed surfaces related to the six most significant deformation modes, of the 17 choices of the structure thickness with respect to the reference undeformed surface

This multi-scale topological distance is then used to order the deformed surfaces to retain in each column of Table 5 those exhibiting a shape resemblance. From Table 4 to Table 5, the surfaces have been sorted using the values of Table 4, and their sorted order displayed in the Table 5. The goal is to have the surfaces labelled from the less to the most deformed, according to our measure of deformation. In Table 5, numbers in red indicate surface (modes) permutations that have been made in order to classify all the shapes.

| Case | 1st surf. | 2nd surf. | 3rd surf. | 4th surf. | 5th surf. | 6th surf. |
|---|---|---|---|---|---|---|
| 01 | 1 | 2 | 3 | 4 | 5 | 6 |
| 02 | 1 | 2 | 3 | 4 | 5 | 6 |
| 03 | 1 | 2 | 3 | 4 | 5 | 6 |
| 04 | 1 | 2 | 3 | 5 | 4 | 6 |
| 05 | 1 | 2 | 3 | 4 | 5 | 6 |
| 06 | 1 | 2 | 3 | 5 | 4 | 6 |
| 07 | 1 | 2 | 3 | 4 | 5 | 6 |
| 08 | 1 | 2 | 3 | 4 | 5 | 6 |
| 09 | 1 | 2 | 4 | 3 | 6 | 5 |
| 10 | 2 | 1 | 3 | 4 | 6 | 5 |
| 11 | 1 | 2 | 3 | 4 | 5 | 6 |
| 12 | 1 | 2 | 3 | 4 | 5 | 6 |
| 13 | 1 | 2 | 3 | 5 | 4 | 6 |
| 14 | 1 | 2 | 4 | 3 | 6 | 5 |
| 15 | 1 | 2 | 4 | 3 | 5 | 6 |
| 16 | 1 | 2 | 3 | 4 | 5 | 6 |
| 17 | 1 | 2 | 3 | 4 | 6 | 5 |

Table 5: Surface ordering. Number in red indicated a permutation that must be performed in order to align surfaces with respect to its shape

## 2.7 MAC Identification

Using the MAC criterion described in Section 2.5, we compute the MAC matrices comparing the model reduced basis (of the first thickness choice) with the remaining 16 reduced modal bases for the other thickness choices, and the results are reported in Fig. 10.

## 2.8 Discussion

Labelling the surfaces as reported in Table 5 aims at classifying them according their shape induced by their deformation. The greater the value of the topological metric $\Omega$, the more deformed the surfaces are. The surface discrepancies are quantified

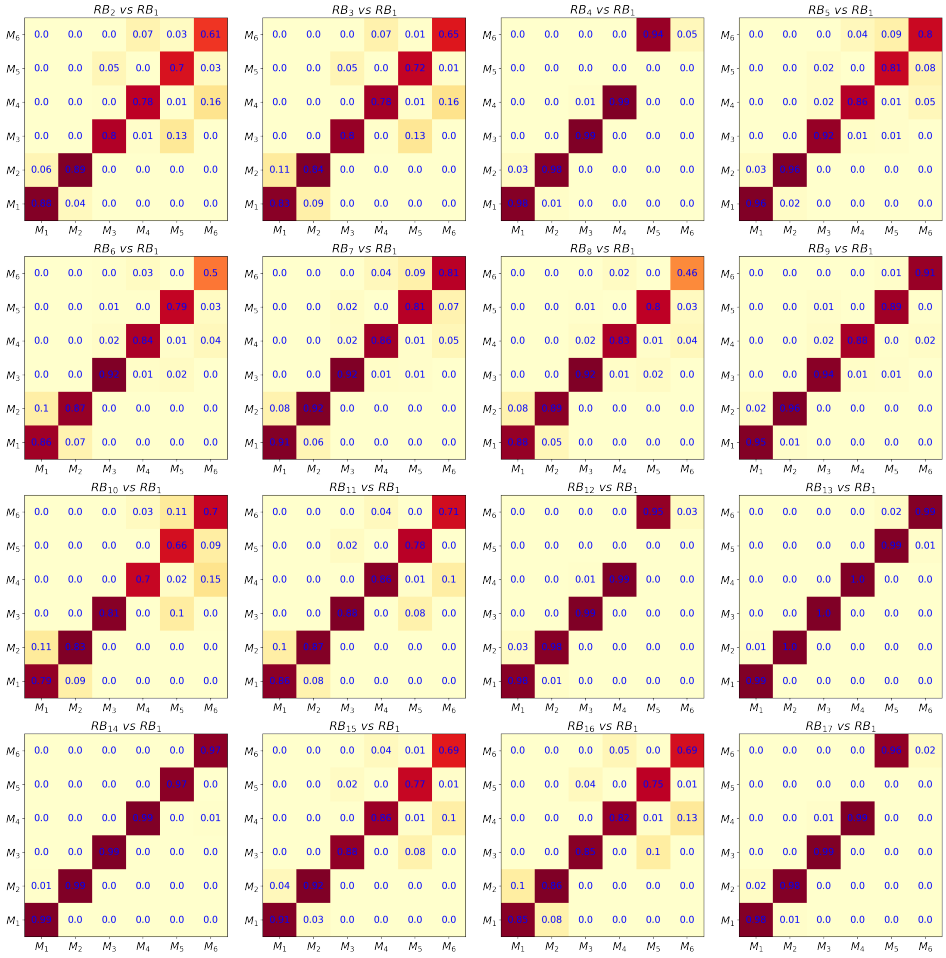



Figure 10: Modal Assurance Criterion matrices when comparing the model reduced basis—RB— (of the first thickness choice) with the remaining 16 reduced modal bases for the other thickness choices

from the transport cost related to the matching of the topological features that the deformed meshes express, through different scales and dimension.

The value of $\Omega$ can be interpreted as a level of topological deformation for a certain deformed mesh on the deformed surface compared with the non deformed mesh and surface. Thus, labels 1 to 6 in the case here addressed, express the magnitude of the surface deformation. Figure 11 depicts the six ordered deformed surfaces for one particular case (structure with a given thickness).

By inspecting Table 5, it can be noticed that the surface label usually match the order of the eigenmodes provided by the eigenproblem solution. However, when modifying the structure thickness, some shapes that were important for a given thickness can be now more or less significant and the order of apparition in the eigenproblem differs. Thus, a permutation must be performed for ordering the modes with respect to their intrinsic shapes, here evaluated by using a metric based on topological concepts.The MAC matrices displayed in Fig. 10 show similar tendencies, as the modes

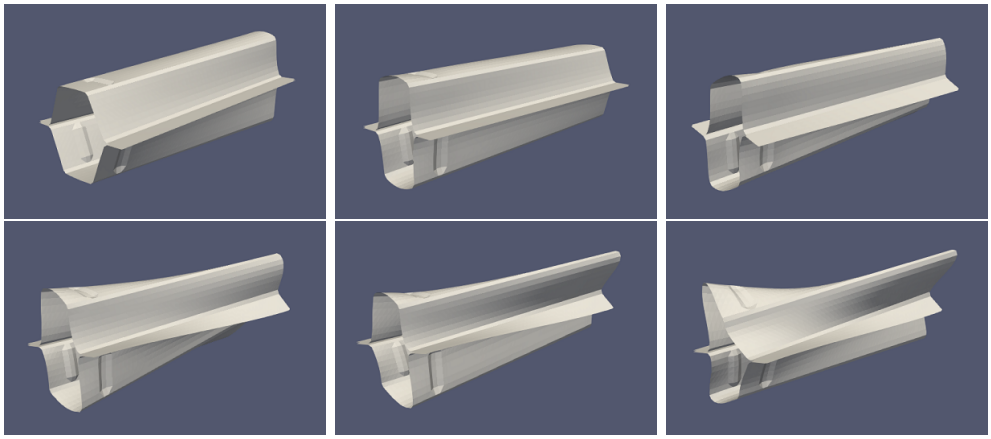


Figure 11: (top-left) First mode; (top-right) Second mode; (middle-left) Third mode; (middle-right) Fourth mode; (bottom- left) Fifth mode; and (bottom-right) Sixth mode

are globally consistent with their labels.

The presented topology-based framework for measuring surface deformations seems a very pertinent, powerful and intrinsic way of quantifying, characterizing and analyzing the deformation modes of structures. The strength of the framework relies on both the topology description of the surface at multiple scales, and the proposed measure based on the optimal matching of the features, to detect how each feature of the surface was deformed.

# 3 Tape Surfaces Characterization

In this chapter, we leverage the main surface topological descriptors to classify tape surface profiles, through the modelling of the evolution of the degree of intimate contact along the consolidation of pre-impregnated preforms associated to a composite forming process.

## 3.1 Introduction

Among composite forming processes for manufacturing structural parts based on the consolidation of pre-impregnated preforms, e.g., sheets, tapes, .... the automated tape placement (ATP) appears as one of the most interesting techniques due to its versatility and its in-situ consolidation, thus avoiding the use of autoclave. In particular, to obtain the cohesion of two thermoplastic layers two specific physical conditions are needed (a) an almost perfect contact (intimate contact) and (b) a temperature enabling molecular diffusion within the process time window, while avoiding thermal degradation. To reach this goal, a tape is placed and progressively bonded to the substrate consisting of the tapes previously laid-up. Due to the low thermal conductivity of usual resins, an intense local heating is usually considered (laser, gas torches, etc.) in conjunction with a local pressure applied by the consolidation roller moving with the heating head, as sketched in Figure 12. Thus, the two main factors to ensure the intimate contact at the plies surfaces are pressure and heat. Intimate contact is required to promote the molecular diffusion. In this process heat plays a double role, on one hand it enhances molecular mobility and on the other hand, the decrease of the material viscosity with the temperature increase, facilitates the squeeze flow of the heated asperities located on the ply surfaces under the compression applied by the consolidation roller.

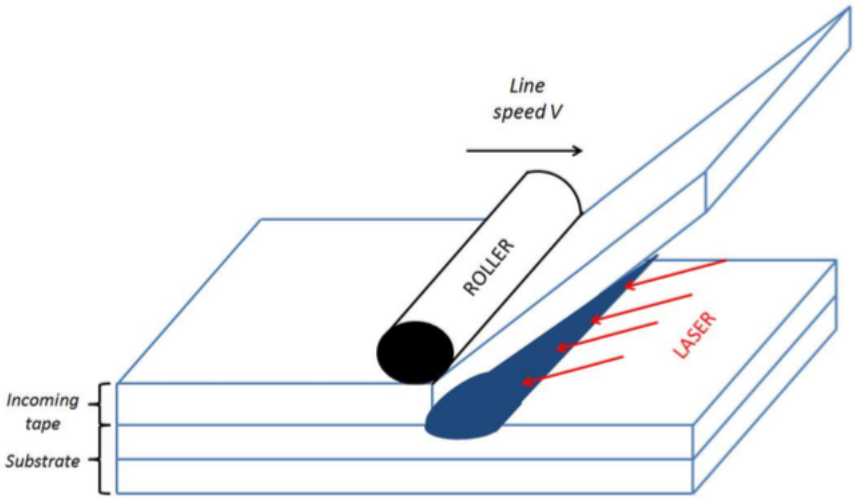



Figure 12: The automated tape placement (ATP).

The numerical model of ATP was introduced in [6] by using the so-called Proper Generalized Decomposition (PGD) [7, 8, 9, 10, 12]. The separated representation involved in the PGD enables the 3D high-resolution solution of models defined in degenerated domains where at least one of their characteristic dimensions remains much smaller than the others and also constructing solutions of parametric models where the model parameters are considered as extra-coordinates [11, 13].

Physical modelling and simulation for Automated Tape Placement (ATP) have been proposed in [14] to study the influence of material and process parameters, while consolidation modelling and sPGD-based non-linear regression have been used in [15] to identify the main surface descriptors for a comprehensive characterization of the tape surfaces.

In this chapter, we first revisit the consolidation modeling and its high resolution simulation, enabling the evaluation of the time evolution of the degree of intimate contact –DIC– when two rough surfaces are put in contact, heated and compressed.

As we are addressing tapes involved in the ATP process sketched in Fig. 12, the roughness squeezing mainly occurs along the transverse direction (the one related to the tape width) induced by the roller compression. Thus, the flow occurs in the transverse section in which the surface reduces to a one-dimensional curve (the so-called surface profile).

It is well-known at an experimental level that the consolidation degree strongly depends on the surface characteristics (roughness). In particular, same process parameters applied to different surfaces produce very different degrees of intimate contact. It allows us to think that the surface topology plays an important role along this process. However, solving the physics-based models for simulating the roughness squeezing occurring at the tapes interface represents a computational effort incompatible with online process control purposes. An alternative approach consists of taking a population of different tapes, with different surfaces, and simulating the consolidation for evaluating for each one the progression of the degree of intimate contact –DIC– while compressing the heated tapes, until reaching its final value at the end of the compression. The final goal is creating a regression able to assign a final value of the DIC to any surface, enabling online process control. The main issue of such an approach is the rough surface description, that is, the most precise and compact way of describing it from some appropriate parameters easy to extract experimentally, to be included in the just referred regression.

In order to extract a concise and complete description of the rough surfaces, we use a topological description [27, 28, 29, 26] of the surface profiles, to construct descriptors such as the persistence diagrams and images. Then, the persistence images are considered for classifying surfaces, or as descriptors involved in the regression relating them to the final DIC reached in the consolidation process, enabling real-time decision making.

## 3.2 Consolidation modelling

In our recent works [22, 23, 14] we proposed simulating the consolidation on the real surfaces instead of the, sometimes too crude, approximations of them based on the use of fractal representations or the ones based on the description of asperities from the use of rectangular elements [53, 54].

As sketched in Figure 13, a Haar-based wavelet representation [14] of a rough surface results in a multi-scale sequence of rectangular patterns, from the coarse scale (level 0) to the finest one (level 8) that constitutes a quite precise representation of the considered surface (the one illustrated in Figure 13). The smoother is the surface, the less levels in the description are required.

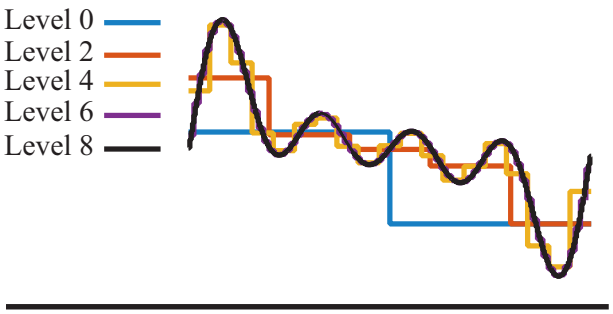



Figure 13: Simple surface representation using Haar wavelets.

The advantage of such a representation consisting of hierarchical rectangles is double: (i) from one side it facilitates the high-resolution of the thermal problem while accounting for all the interface details and their time evolution; and on the other (ii) it allows squeezing the rectangles of a certain level (from the finest level to the coarser one) while retaining the lubrication hypotheses, fact that simplifies significantly the flow modeling and the calculation of the interface evolution when squeezing the asperities. Both aspects are revisited below.

1. As soon as the rough surface profile is represented in a step-wise way consisting of $\mathtt{R}$ rectangular elements, with each rectangle $r$ having a length $l_r$ and a height $h_r$, assumed centered at $x_r$, each rectangle can be expressed by its characteristic function in a separated form $\chi_r(x,z) = \mathcal{L}_r(x)\mathcal{H}_r(z)$, with $\mathcal{L}_r(x)$ and $\mathcal{H}_r(z)$ given respectively by Eqs. (7) and (8),

$$\mathcal{L}_r(x) = \begin{cases} 1 & \text{if } x \in (x_r - l_r/2, x_r + l_r/2) \\ 0 & \text{elsewhere} \end{cases}, \tag{7}$$

and

$$\mathcal{H}_r(z) = \begin{cases} 1 & \text{if } z \in (0, h_r) \\ 0 & \text{elsewhere} \end{cases}, \tag{8}$$

that allows expressing the conductivity at the interface level according to Eq. (9),

$$\mathbf{K}(x,z) = \left(1 - \sum_{r=1}^{\mathtt{R}} \mathcal{L}_r(x)\mathcal{H}_r(z)\right)\mathbf{K}_c + \left(\sum_{r=1}^{\mathtt{R}} \mathcal{L}_r(x)\mathcal{H}_r(z)\right)\mathbf{K}_a, \tag{9}$$

where $\mathbf{K}_a$ and $\mathbf{K}_c$ represent the air and composites conductivities, with the former assumed isotropic and the last concerning the composite conductivity transverse components.

This separated representation of the thermal conductivity allows looking for a separated representation of the temperature field within the proper generalized decomposition –PGD– framework, according to Eq. (10)

$$T(x,z) \approx \sum_{i=1}^{M} X_i(x)\ Z_i(z), \tag{10}$$

that by decoupling the 2D heat equation solution into a sequence of 1D problems for computing the functions $X_i(x)$ and $Z_i(z)$ allows an extremely fine resolution as discussed in [14].

As the asperities squeezing progresses, the surface evolves and with it the height of the different rectangular elements. The conductivity separated representation must be updated and the thermal problem solved again to compute the updated temperature field (10).

2. As soon as the temperature field is available, the polymer viscosity can be evaluated and the asperities will flow under the applied pressure. As commented, the description of the surface by using rectangular elements, with their characteristic length $\overline{l}$ much larger that its characteristic height $\overline{h}$, i.e. $\overline{l} \gg \overline{h}$, makes possible the use of the lubrication hypotheses, widely addressed in our former works [23].

   The surface updating procedure is quite simple. We consider all the compressed rectangles, and solve in them the squeeze flow model, while assuming that the pressure in all the other elements vanishes. As soon as the pressures are available in all the rectangles that are being compressed, the velocity field and more precisely the flow rates can be obtained at the lateral boundaries. The fluid leaving each rectangular element that is being compressed is transferred to the neighbor rectangular element that increases its height accordingly in order to ensure the mass conservation.

As it can be noticed, this procedure allows unimaginable level of accuracy, however, despite of the speed-up that separated representation offers, its use online for predicting the thermal and flow coupled problem for any incoming rough tape is not an option.

### 3.3 Surface descriptors based on homology persistence

In this section we introduce the data and methods used, in particular the TDA and its related procedures (persistent diagrams and images), even if other approaches exist, e.g. [15, 31].

The proposed methodology proceeds in three main stages:

1. Processing the surface profiles data;

2. Compute persistent diagrams and images;

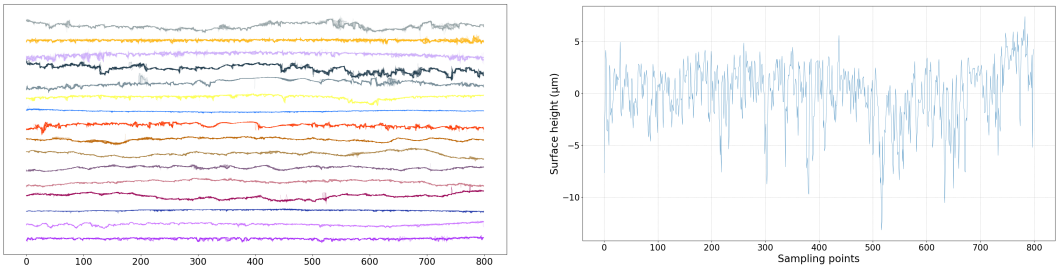
3. Construct the regressions relating the surface topological descriptors and the quantities of interest –QoI–, concretely the DIC.

### 3.3.1 Processing the surface profiles data

In order to classify the main surface descriptors of a tape surface, we will consider samples scanned with a 3D non contact profilometer, with a 3.5 um resolution and where each sample has a length of approximately 3 mm (along the tape width). A set of 1359 surface profiles were extracted from 16 different pre-impregnated composite tapes provided by different customers using different impregnation process, each one represented by 800 measured data points $\left\{S_\ell^{(k)} : 1 \leq \ell \leq 800,\, 1 \leq k \leq 1359\right\}$.

The main goal is to give a procedure to construct a classification $\mathcal{C}(S)$, that is, a map ensuring $\mathcal{C}(S^{(k)}) = i$ if and only if $S^{(k)}$ was extracted from the tape $i$, with $i = \{1, 2, \ldots, 16\}$.

In particular, to facilitate data comparison the profiles are corrected by subtracting the average height. Figure 14 depicts the different surfaces in each of the 16 classes, as well as normalized profile.




(a) (b)

Figure 14: Surface profiles data. (a) The 16 surface classes, (b) corrected surface profile when subtracting its averaged height.

### 3.3.2 Persistence diagrams and images

The persistence diagram consists of a one-to-one local-minimum-local-maxixum pairing. To illustrate the procedure we consider a simple case of a profile described by 9 heights, $S = \{11, 14, 9, 7, 9, 7, 8, 10, 9\}$, that corresponds to the 9 data points depicted in Figure 15: $\{(0, 11), (1, 14), (2, 9), (3, 7), (4, 9), (5, 7), (6, 8), (7, 10), (8, 9)\}$.

Now, we consider the 4 local minimum: $\{(0, 11), (3, 7), (5, 7), (8, 9)\}$ and the only 3 local maximum: $(1, 14), (4, 9), (7, 10)$. We associate $(0, 11)$ to $(1, 14)$, then $(8, 9)$ to $(7, 10)$ and finally $(5, 7)$ to $(4, 9)$. The remaining local minimum $(3, 7)$ can not be paired to any other local maximum because all of them have already been paired.

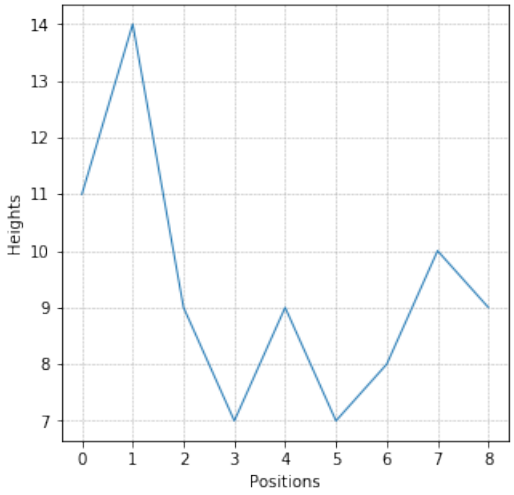



Figure 15: Profile consisting of 9 measured height at 9 positions.

The local-minimum-local-maximum paired heights constitutes the so-called persistence diagram $\mathcal{PD}(S)$, in our example $\mathcal{PD}(S) = \{(7, 9), (9, 10), (11, 14)\}$, with the minimum of the pair representing the topological occurrence birth, whereas the associated maximum its death.

In our example, consisting of the 9 data and the three topological occurrences composing, the associated persistence diagram is shown in Figure 16.

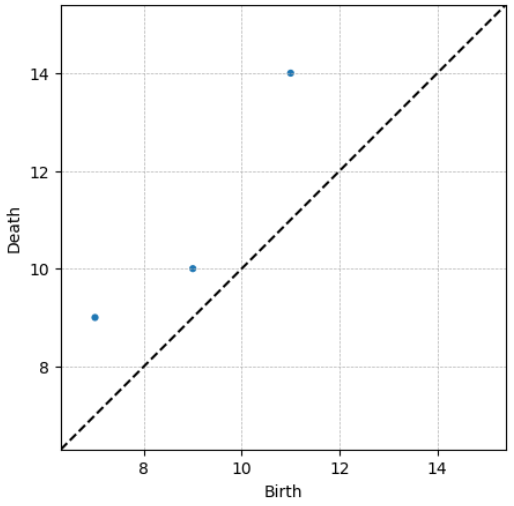



Figure 16: Persistence diagram $\mathcal{PD}(S)$.

In the two dimensional representation associated to the persistence diagram, each data point $(x, y)$ verifies the relationship $y \geq x$ (the topological occurrence birth precedes its death) and then points locate above the bisector $x = y$. The topological representation provided by the persistence diagram offers a very concise

description of any curve (e.g. surface profiles, time-series, etc.) and the change in their topological features as considered in [29, 26, 32, 74].

In order to use the persistence diagram and perform vectorial operations such as the ones required in classification, we must transform the persistence diagram into a vectorial representation of it, the so-called *persistence image* [33, 74].

For that purpose, we first introduce the so-called *lifetime diagram* $\mathcal{T}(S)$ associated to the function $\mathcal{PD}(S)$, defined in Eq.(11)

$$\mathcal{T}(S) = \{(x, y) \in \mathcal{PD}(S) \to (x, y - x) \in \mathbb{R}^2\}, \tag{11}$$

where $y - x$ represent the lifetime of the topological occurrence. In our example we have $\mathcal{T}(S) = \{(7, 2), (9, 1), (11, 3)\}$, that is illustrated in Figure 17.

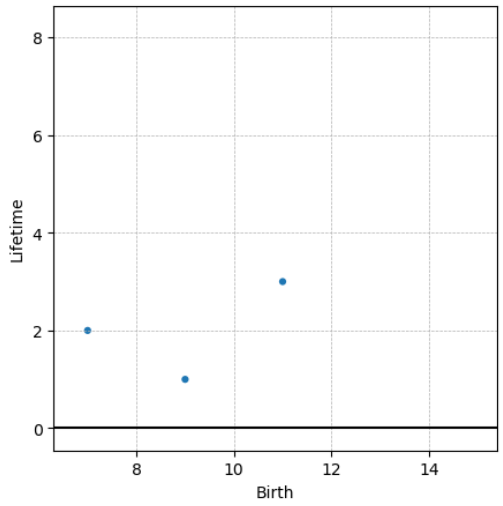



Figure 17: Lifetime diagram $\mathcal{T}(S)$ associated to $\mathcal{PD}(S)$.

Next, we will construct a persistent image as follows. We consider a continuous piecewise derivable non-negative weighting function (with $(x, y) \in \mathcal{T}(S)$, $w(x, 0) = 0$ and $w(x, y_{max}) = 1$, with $y_{max} = \max(y)$, that can be approximated by a linear function of the lifetime $y$, e.g. $w(x, y) = y/y_{max}$) and a bivariate normal distribution $g_{x,y}(u, v)$ centered at each point $(x, y) \in \mathcal{T}(S)$ and with its variance $\sigma$, $\sigma > 0$, scaling with the maximum of the lifetime diagram [33, 74], then we define the variable $\rho_S(u, v)$ expressed in Eq. (12)

$$\rho_S(u, v) = \sum_{(x,y)\in\mathcal{T}(S)} w(x, y)\ g_{(x,y)}(u, v), \tag{12}$$

with $(u, v) \in \mathcal{D}$, with $\mathcal{D}$ a compact domain (for example the domain in which $\mathcal{T}(S)$ is defined).

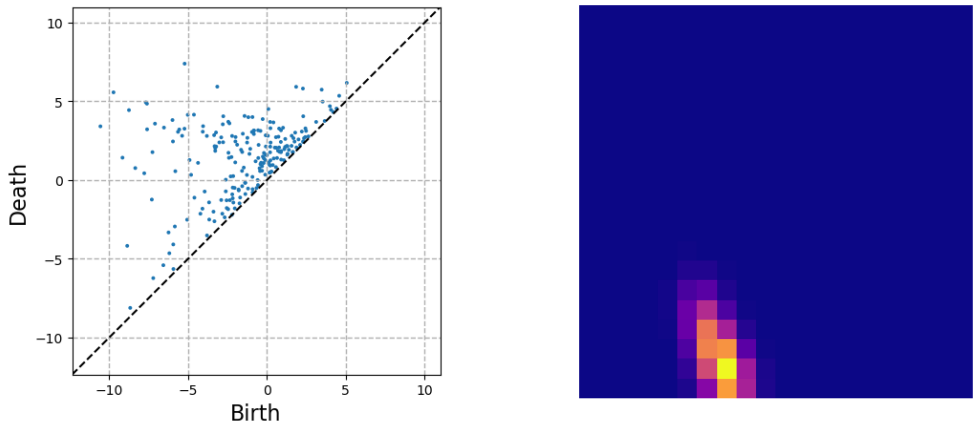
Now, the domain $\mathcal{D}$ is partitioned in a series of non-overlapping subdomains covering it, the so-called pixels $P_i$, with $\mathcal{D} = \cup_{i=1}^{\mathsf{P}} P_i$, and function $\rho_S(u, v)$ averaged

in each of those pixels, that will define the *persistence image* $\mathcal{PI}(S)$. Thus each of the $\mathtt{P}$ pixels of the persistence image $\mathcal{PI}(S)$ takes the value given by Eq. (13)

$$\mathcal{PI}_{P_i}(S) = \iint_{P_i} \rho_S(u,v)\ du\ dv. \tag{13}$$

As the profile that served to illustrate the different concepts contains too few topological occurrences, to illustrate what a persistence image resembles to, we consider a profile related to one of the measured rough surfaces $S$, compute the persistence diagram $\mathcal{PD}(S)$, then its associated lifetime diagram $\mathcal{T}(S)$, and finally its persistence image $\mathcal{PI}(S)$. Figure 18 shows $\mathcal{PD}(S)$ and $\mathcal{PI}(S)$, the last employing $20 \times 20$ pixels, i.e. $\mathtt{P} = 400$ with a variance $\sigma$ in the normal distribution $g_{x,y}(u,v)$ given by Eq. (14)

$$\sigma = \frac{max_{(x,y)\in\mathcal{T}(S)}\ \{y\}}{20}. \tag{14}$$




(a) (b)

Figure 18: TDA analysis of a real rough surface. (a) Persistence diagram $\mathcal{PD}(S)$, (b) persistence image $\mathcal{PI}(S)$.

### 3.3.3 Images classification

When applying the rationale just described to the 1359 rough surfaces $S^{(k)}$, $k = 1, \ldots, 1359$, we will obtain the associated 1359 persistence diagrams $\mathcal{PD}(S^{(k)})$, lifetime diagrams $\mathcal{T}(S^{(k)})$ and persistence images $\mathcal{PI}(S^{(k)})$, $k = 1, \ldots, 1359$.

Thus each surface produced a persistence image composed of $\mathtt{P} = 400$ pixels. These images are expected belonging to 16 different classes, the 16 families of composite tapes. Obviously, trying to proceed to that classification directly from the surface raw data $S^{(k)}$ seems a tricky issue because the proximity is not well defined when using a standard Euclidean metric. The same surface taken with a small shift will induce a significant difference. Metrics based on the topology seem more robust because the appealing associated invariance properties. Thus, more than trying to classify from the raw data, persistence images seem to be the right starting point.

Image classification is a procedure 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 is thus based on how close a new point is to each training sample. The Euclidean distance is the most common distance metric used in low dimensional data sets. The training samples are representative of the known classes of interest to the analyst.

In order to classify the persistence images we can use any state of the art technique. In our case we considered the Random Forest classification [73]. We train the random forest (consisting of 400 trees) by using 65% of the the persistence images (the remaining 35% serving to evaluate the classification performances), where a label was attached to each one, a label precisely specifying the family, among the 16 composites considered, to which it belongs.

With the trained random forest one expects, from a given persistence image, obtaining in almost real-time the family to which it belongs, of major interest in process control.

### 3.3.4 Images clustering

Unsupervised learning algorithms aim at finding unknown patterns in data sets without pre-existing labels. Clustering is used in unsupervised learning to group, or segment, data that has not been labelled, classified or categorized. It is based on the presence or absence of commonalities in each new piece of data. This approach also helps detect anomalous data points that do not fit into either group.

One of the most popular clustering techniques, the $k$-means, aims at partitioning the observations into $k$ clusters in which each observation belongs to the cluster with the nearest mean or center [73]. The cluster center serves as a prototype of the cluster population. The observations are then allocated according to the criterion of minimizing the within-cluster variances, which is a squared Euclidean distance. The data can be then labelled according to their respective clusters (arbitrarily numbered).

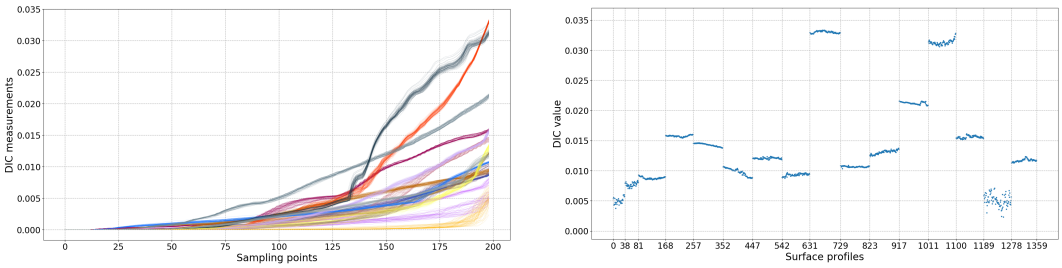
To determine the optimal number of clusters we proceed as follows. For different values of $k$, $k$-means is trained with the whole data-set, and the data-labelled depending on the cluster to which each data belongs. Then, $k$-means is applied again but now with only 65% of the data. Then, for each data the cluster to which it belongs is compared to the label (cluster to which it belonged when all the data was employed in the $k$-means). A parametric variance analysis allows determining the optimal value of $k$, that in our case resulted as expected $k = 16$.

As soon as the best number of cluster is determined, $k = 16$ in our case, $k$-means proceeds with the whole data to generate the reference labels (cluster to which each data belongs). Then, the process repeats but now employing only 65% of the data. Finally we estimate for the remaining 35% of the data to which cluster it is associated and compare with its label to have an estimation of the clustering performances.

### 3.3.5 Predicting the degree of intimate contact

The consolidation process of all the available surfaces (1359) was simulated by using the PGD-based high-resolution solver described in Section 3.2. The evolution of the DIC, that is, the fraction of the surface in perfect contact, was evaluated at the different time steps. Figure 19a depicts the DIC evolution for the 1359 surfaces during the first 200 time steps of the consolidation process. As it can be noticed the dispersion of the DIC is quite small within each one of the 16 composite tapes (classes), however it exhibits large differences from one composite to another.

In what follows we are interested in the DIC prediction at the last time step (the number 200), that will consist of our quantify of interest –QoI– $\mathcal{O}$, that for each surface results in the values $\mathcal{O}^{(k)}$, $k = 1, \ldots, 1359$ depicted in Figure 19b.




(a) (b)

Figure 19: Simulated degree of intimate contact. (a) Simulated time evolution of the DIC, (b) DIC reached at time step 200 for each surface.

Now, we are interested in constructing a regression for expressing the QoI, $\mathcal{O}$, as a function of the considered surface, the geometry of the last expressed through its persistence image.

For that purpose we are considering two regression techniques: (i) the so-called *Code2Vect* [16] summarized in the Appendix, and (ii) the random forest.

*Code2Vect* maps the surfaces described by the 400 values related to the pixels of their associated persistence images into another low dimensional vector space where the distance between any two points (representing two surfaces) scales with the QoI difference, that is, with respect to the difference of their DIC. However, as for usual nonlinear regression techniques, the complexity scales with the number of parameters involved in the regression, and here 400 seems a bit excessive with respect to the available data.

For this reason, and prior to the use of *Code2Vect* regression, the 1359 persistence images, each represented by $20 \times 20$ pixels, are first analyzed by using the principal component analysis –PCA– to remove linear correlations [73] where the two most significant modes were retained, and each persistence image described by its projection on both models. Thus, the reduction is impressive, each persistence image, and in consequence each surface, is now described from only two parameters. Then, the *Code2Vect* was employed to establish the regression between these two parameters and the quantity of Interest $\mathcal{O}$, the final DIC [16].

Again, to evaluate the regression performances, *Code2Vect* was trained by using 80% of the data, and the remaining 20% served for evaluating the prediction performances.

As previously indicated a regression based on the use of Random Forest [73] (using 400 trees) was considered, with 65% of the data used in the training and 35% for evaluating the prediction performances.

### 3.3.6 Models evaluation

For evaluating the model performances we consider different procedures:

- Confusion matrix

  The component $(i, j)$ of the confusion matrix contains de number of surfaces that belonging to a class $i$ are predicted belonging to class $j$. Obviously the classification is perfect when this matrix becomes diagonal.

- Classification scoring. Evaluating a classification model is 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 different estimators of the model performances:

  - The Precision (P) is the number of correct positive results divided by the number of all positive results, expressed by (15)

$$P = \frac{TP}{TP + FP} \tag{15}$$

  - The Recall (R) is the number of correct positive results divided by the number of all relevant samples, expressed by (16)

$$R = \frac{TP}{TP + FN} \tag{16}$$

  - The F1 score is the harmonic mean of precision and recall, expressed by (17)

$$F1 = 2\frac{P \cdot R}{P + R} \tag{17}$$

– The Accuracy (A) is the number of correct predictions over the number of all samples, expressed by (18)

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

- Regression scoring

  We evaluate our regression prediction using the $\mathcal{R}^2$ coefficient, defined in Eq. (19)

$$\mathcal{R}^2 = 1 - \frac{\sum_i^n \left(\mathcal{O}_i^{true} - \mathcal{O}_i^{pred}\right)^2}{\sum_i^n \left(\mathcal{O}_i^{true} - \overline{\mathcal{O}}^{true})\right)^2}. \tag{19}$$

  We also use the mean absolute percentage error $MAPE$, defined in (20)

$$MAPE = \frac{100\%}{n} \sum_i^n \left| \frac{\mathcal{O}_i^{true} - \mathcal{O}_i^{pred}}{\mathcal{O}_i^{true}} \right|, \tag{20}$$

  with best model having the closest MAPE to 0%.

- Features importance. In decision trees, every node is a condition on how to split values for a single feature, so that similar values of the dependent variable end up in the same set after the split. The condition is based on impurity, which in the case of classification problems is the Gini impurity or the information gain (entropy), while for regression trees it is the variance. So when training a tree, we can compute how much each feature contributes to decreasing the weighted impurity, and in the case of Random Forest, we are talking about averaging the decrease in impurity over all the trees [73]. Although this method is known to be statistically biased for categorical variables, it should not be affected in our case, as we only have homogeneous and continuous variables, $20 \times 20$ pixels images.

### 3.3.7 Code2Vect

*Code2Vect* maps data, eventually heterogeneous, discrete, categorical, ... into a vector space equipped of an euclidean metric allowing computing distances, and in which points with similar outputs $\mathcal{O}$ remain close one to other as sketched in Figure 20.

We assume that points in the origin space (space of representation) consists of `P` arrays composed on $D$ entries, noted by $\mathbf{y}_i$. Theirs images in the vector space are noted by $\mathbf{x}_i \in \mathbb{R}^d$, with $d \ll D$. The mapping is described by the $d \times D$ matrix $\mathbf{W}$, according to (21)

$$\mathbf{x} = \mathbf{W}\mathbf{y}, \tag{21}$$

where both, the components of $\mathbf{W}$ and the images $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \cdots, \mathtt{P}$, must be calculated. Each point $\mathbf{x}_i$ keep the label (value of the output of interest) associated with is origin point $\mathbf{y}_i$, denoted by $\mathcal{O}_i$.

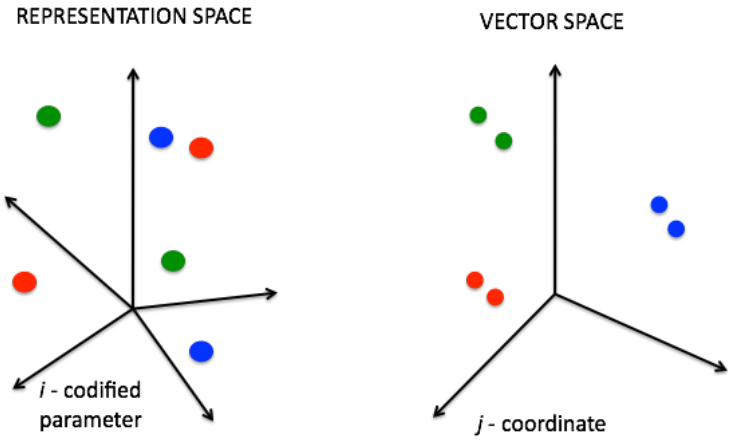



Figure 20: Input space (a) and target vector space (b).

We would like placing points $\mathbf{x}_i$, such that the Euclidian distance with each other point $\mathbf{x}_j$ scales with their outputs difference, as expressed in Eq. (22)

$$(\mathbf{W}(\mathbf{y}_i - \mathbf{y}_j)) \cdot (\mathbf{W}(\mathbf{y}_i - \mathbf{y}_j)) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 = |\mathcal{O}_i - \mathcal{O}_j|, \tag{22}$$

where the coordinates of one of the points can be arbitrarily chosen. Thus, there are $\frac{\mathtt{P}^2}{2} - \mathtt{P}$ relations to determine the $d \times D + \mathtt{P} \times d$ unknowns.

Linear mappings are limited and do not allow proceeding in nonlinear settings. Thus, a better choice consists of the nonlinear mapping $\mathbf{W}(\mathbf{y})$ [16].

## 3.4 Results

In this section we provide the numerical results and evaluations associated to each of the previously introduced models: Random Forest classification, $k$-means clustering, *Code2Vect* and Random Forest regression.

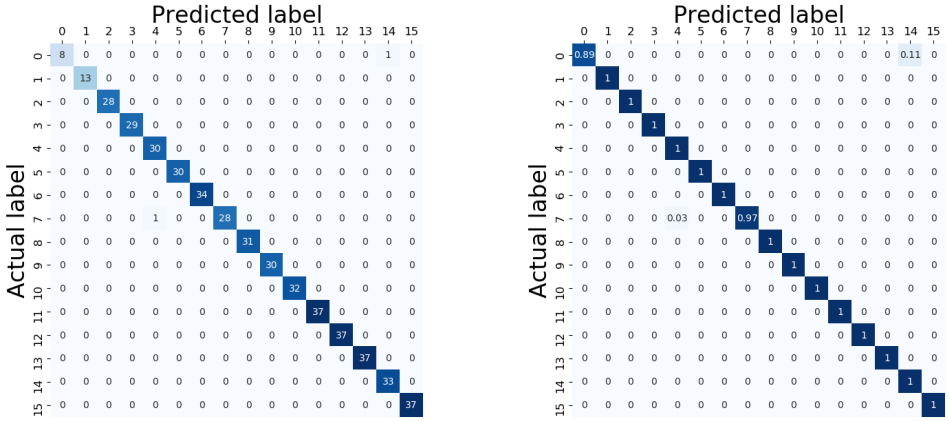
### 3.4.1 Classification results

The trained random forest classifier for the persistence images shows high accuracy scores (over 99%), suggesting a strong differentiation of the images with respect to their generating surface profiles. The classification performance report shown in Figure 21 summarizes the precision, recall, f1-score estimators over each of the 16 classes (surface labels) from the test dataset. The number of samples for each class is also provided. The accuracy score estimator is computed over the complete test dataset, along with the macro and weighted averages of the previously cited estimators.

The confusion matrix given in Figure 22 shows that images are accurately labelled across all classes, reporting also the normalized scores. It was proved that these results are quite insensible to randomizing and changing the ratio between the training and testing samples.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Surface 01 | 1.00 | 0.89 | 0.94 | 9 |
| Surface 02 | 1.00 | 1.00 | 1.00 | 13 |
| Surface 03 | 1.00 | 1.00 | 1.00 | 28 |
| Surface 04 | 1.00 | 1.00 | 1.00 | 29 |
| Surface 05 | 1.00 | 1.00 | 1.00 | 30 |
| Surface 06 | 1.00 | 1.00 | 1.00 | 30 |
| Surface 07 | 1.00 | 1.00 | 1.00 | 34 |
| Surface 08 | 1.00 | 1.00 | 1.00 | 29 |
| Surface 09 | 1.00 | 1.00 | 1.00 | 31 |
| Surface 10 | 1.00 | 1.00 | 1.00 | 30 |
| Surface 11 | 1.00 | 1.00 | 1.00 | 32 |
| Surface 12 | 1.00 | 1.00 | 1.00 | 37 |
| Surface 13 | 1.00 | 1.00 | 1.00 | 37 |
| Surface 14 | 1.00 | 1.00 | 1.00 | 37 |
| Surface 15 | 0.97 | 1.00 | 0.99 | 33 |
| Surface 16 | 1.00 | 1.00 | 1.00 | 37 |
| accuracy | | | 1.00 | 476 |
| macro avg | 1.00 | 0.99 | 1.00 | 476 |
| weighted avg | 1.00 | 1.00 | 1.00 | 476 |

Figure 21: Classification performance report



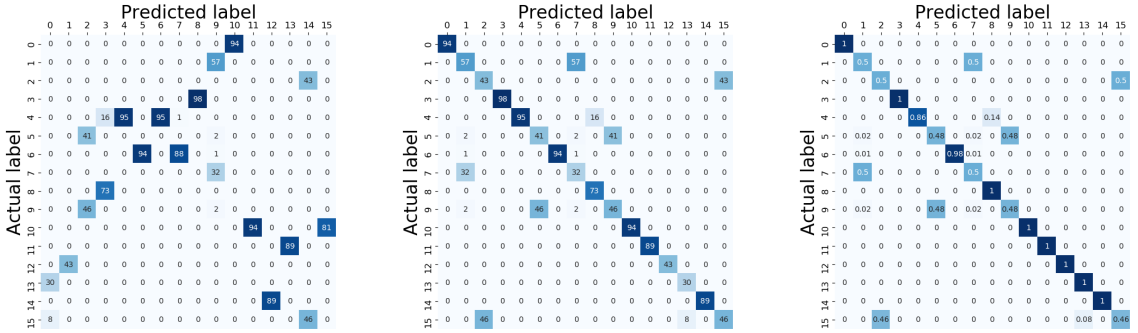

(a) (b)

Figure 22: Confusion matrix for the random forest classifier. (a) Original, (b) normalized.

### 3.4.2 Clustering results

Given the disparity between clusters labels and original labels ($k$-means algorithm assigns clusters labels arbitrarily), the confusion matrix is the best way to evaluate the model performance. It shows a majority of one-to-one classes correspondence, meaning that given a certain permutation of the columns (clusters labels), we can obtain a rearranged matrix. The permuted confusion matrix given in Figure 23b

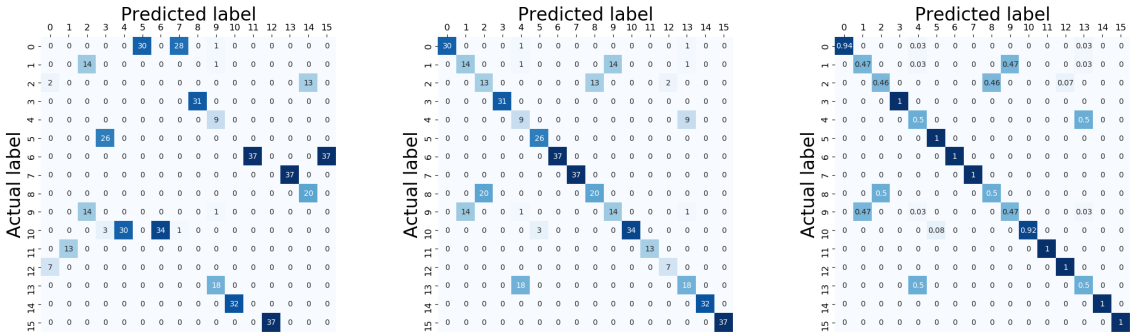
shows a good accuracy (80%) for the clustering compared to the original profiles labels.




(a) (b) (c) d

Figure 23: Confusion matrix for $k$-means clustering of the complete dataset. (a) Original, (b) permuted, and (c) normalized.

In order to evaluate the predictive performances of the trained model, we compare the predicted labels (clusters) of the test data against their actual labels. The labelling disparity still remains, with a majority of one-to-one classes correspondence. After reordering the confusion matrix, depicted in Figure 24b, we can observe a good enough accuracy of the clustering (77%) for predicting labels. Thus, the model allows to identify the surface of new incoming profiles, when proceeding in an unsupervised way.



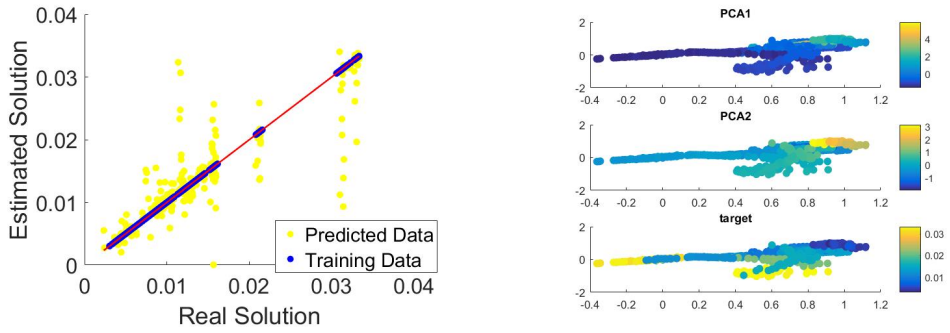

(a) (b) (c)

Figure 24: Confusion matrix for $k$-means predictions over the test dataset. (a) Original, (b) permuted, and (c) normalized.

### 3.4.3 DIC prediction by regression

*Code2Vect* performs an accurate regression of the DIC, with a MAPE of 2.3% when considering all the data and a MAPE of 12.86% when applied on the points that were not used in training, as shown in Figure 25. Thus, it can be concluded that the reduction of the persistence images to only two quantities (the weights of the two

most relevant modes extracted from the PCA applied on the persistence images) has not a significant impact in the regression performances, proving that the combination of *Code2Vect* and PCA constitutes an excellent nonlinear dimensionality reduction technique. The correlation between these two parameteres (PCA weights) and the QoI (DIC) is also shown in Figure 25b.




(a) (b)

Figure 25: *Code2Vect* regression performance. (a) Prediction error, (b) projected space.

Similarly, the random forest regression shows a high reliability to accurately predict our quantity of interest, with an $R^2$ score over 96%.

## 3.5 Discussion

Composite tapes have been successfully classified using the persistence images related to their rough surfaces. Topological Data Analysis seems a very valuable way of describing accurately and concisely those surfaces, in particular its roughness that constitutes the main factor when evaluating the consolidation performances, from the time evolution of the DIC (degree of intimate contact).

Different classification (supervised) and clustering (unsupervised) were successfully applied for associating the different surfaces to the composites from which they were extracted. On the other hand, by using advanced regression techniques, the degree of intimate contact was associated to the surface topological content, with excellent and fast predictions of the expected DIC for a given surface.

These procedures open unimaginable possibilities in process control and the online adaptation of processing parameters for ensuring the adequate DIC at the end of the process.

# 4 Advanced Driver-Assistance Systems

In this chapter, we aim at evaluating the state of drivers to determine whether they are attentive to the road or not by using motion sensor data collected from car driving experiments. That is, our goal is to design a predictive model that can estimate the state of drivers given the data collected from motion sensors. For that purpose, we analyze and transform the data coming from sensor time series and build a machine learning model based on the topological features extracted with the TDA. We provide some experiments showing that our model proves to be accurate in the identification of the state of the user, predicting whether they are relaxed or tense.

## 4.1 Introduction

While there have recently been considerable advances in self-driving car technology, driving still relies mainly on human factors. Even in self-driving mode, human drivers must often make decision in a fraction of a second to avoid accidents. Therefore, it is still of utmost importance to develop systems capable of discerning if the human driver is attentive or not to the road conditions. In general, the so-called advanced driver assistance systems (ADAS) [55, 56] are systems that are able to improve the driver's performance, among which, adaptive speed limiters, pedestrian detectors [57], and cruise controllers are some of the most popular systems. Fatigue alerting systems are among the most useful among ADAS systems, and the aim of this work is to contribute to the development of such a system based on a systematic analysis of drivers in actual driving conditions.

The estimation of the driver's condition (degree of attention to the road, fatigue, etc.) is a very important factor to ensure safety in driving [58, 59]. A recent review on the topic can be found in [60]. The goal of this work is to extract behavior patterns from car user data to be able to accurately estimate their state. We used data obtained by the laboratory of prof. Hyung Yun Choi at Hongik University in Seoul. His experiment involved the application of mechanical stimulation to people seated in an automobile.

Our main goal is to extract patterns of behavior from experimental data so as to allow us to learn the most relevant factors affecting driver's attention to the situation of the road.

In the present work, we combine some tools from Morse theory [35] and topological data analysis (TDA) with all of the associated concepts and methods (e.g., Betti numbers, homology persistence, barcodes, persistence images, etc.) [34], most of them introduced and employed later in order to analyze and classify the experimental data. This allows us to introduce concepts as barcodes, that is, persistent and life-time diagrams in a similar way to how they are used in persistent homology. Our main goal is to predict car user behavior following a supervised approach [27]. Instead of considering an original sensor signal as the quantity of interest, we focus on its topological features. In this sense, the framework proposed in this paper allows us to unveil the true dimensionality of data or, in other words, the actual

number of factors affecting driver's performance. Thus, we model a sensor signal as a dynamical system, and, therefore, our approach seems to be better at describing its properties, or rather its variations, such as extrema, patterns, and self-similarity, than other approaches. We note that our approach is, in some senses, similar to that followed by Milnor and Thurston [36] in the study of the combinatorial properties of dynamical systems by combining tools from automata theory.

In Section **??**, we describe the material and methods employed in this work. Particular attention is paid to the process of data acquisition and the description of time series and data curating. In Section 4.7, we present the main results of this work, and we discuss the main consequences in Section 4.8. In Section 4.2, we describe the data acquisition, and in Section 4.3, we provide a description of the time series. Section 4.4 is devoted to data preprocessing. The mathematical tools used to describe the times series at a topological level are explained in Section 4.5. Finally, the image classification methodology is given in Section 4.6.

## 4.2 Data Acquisition

Our proposed predictor directly uses the data collected from the experiments. The data acquisition process involves measuring the response of human behavior when an excitation is applied to the seat. Figure 26 shows the location of the sensors in the experiments.

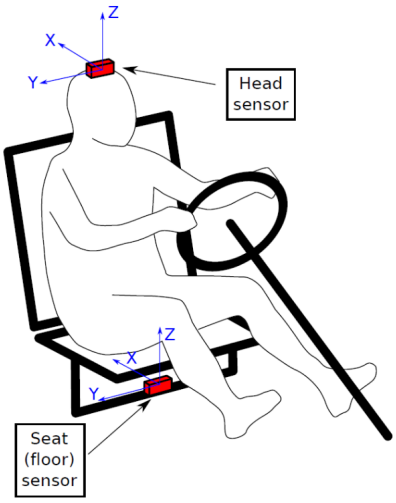



Figure 26: Scheme of the data acquisition process showing the location of the sensors.

The excitation signal is an angular acceleration imposed on the seat of the user. This acceleration is an oscillating chirp function with a frequency range of 1 to 7.5 Hz on the X axis in rotation. The linear acceleration $\mathbf{a} = (a_x, a_y, a_z)$ and angular velocity $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ were measured in both the head and the seat by two IMU (Shimmer inertia measurement unit (IMU) sensors) at 256 Hz. By observing the floor excitation signals, we noted that the excitation is purely rotational around the X-axis—see Figure 27.

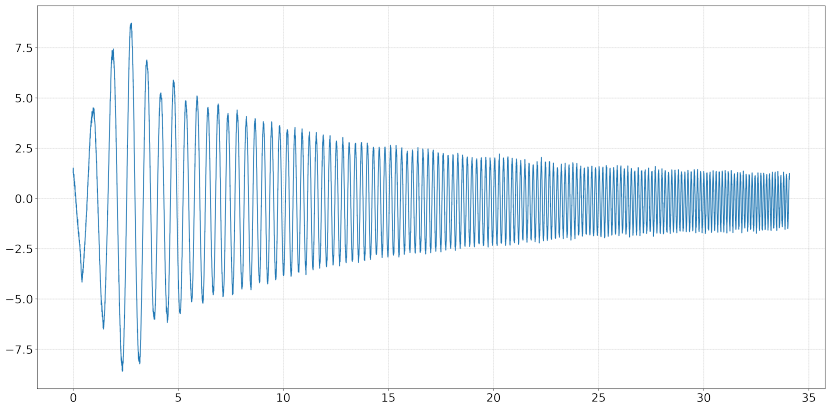



Figure 27: Floor excitation: X-axis angular velocity time series.

Several experiences were conducted by nine people by taking into account a set of six fixed states: driver, passenger, tense person, relaxed person, rigid seat, and SAV (sport activity vehicle seat). In particular, for each individual, eight experiments for the six available states were performed:

| **Class** | **Label** |
|---|---|
| 1 | SAVRelaxedPassager |
| 2 | SAVTensePassager |
| 3 | SAVRelaxedDriver |
| 4 | SAVTenseDriver |
| 5 | RigidRelaxedPassager |
| 6 | RigidTensePassager |
| 7 | RigidRelaxedDriver |
| 8 | RigidTenseDriver |

As a consequence, we worked with a sample of 72 experiences, each of them encoded in a time series (as we explain later). Our goal is to classify the behavior of a generic driver, assigning one of the two states (tense or relaxed) by using the sensor data.

## 4.3 Time Series Description

The data acquired from sensors (see Figures 28 and 29) were stored into six-dimensional time series, for both linear acceleration and angular velocity of the head movement. The sampling frequency of the data was 256 Hz, and the duration of the experiment was 34 s; hence, the resulting data dimensionality is $256 \times 34 = 8704$. For each times series, where $1 \leq t \leq 8704$, we constructed three new times series called the sliding window, embedding a length of 5800. The first one is given by the times values from $t = 1$ to $t = 5800$, the second is given by the times values from $t = 1450$ to $t = 7250$, and, to conclude, the third time window is defined as from $t = 2904$ to $t = 8704$. Each element in the sample ($1 \leq i \leq 72$) was encoded by means of three

six-dimensional time series representing each of the three sliding windows that we represent in matrix form as follows:

$$TS_{3(i-1)+1} = \begin{bmatrix} a_x^\ell(1) & a_x^\ell(2) & \cdots & a_x^\ell(5800) \\ a_y^\ell(1) & a_y^\ell(2) & \cdots & a_y^\ell(5800) \\ a_z^\ell(1) & a_z^\ell(2) & \cdots & a_z^\ell(5800) \\ \omega_x^\ell(1) & \omega_x^\ell(2) & \cdots & \omega_x^\ell(5800) \\ \omega_y^\ell(1) & \omega_y^\ell(2) & \cdots & \omega_y^\ell(5800) \\ \omega_z^\ell(1) & \omega_z^\ell(2) & \cdots & \omega_z^\ell(5800) \end{bmatrix}$$

$$TS_{3(i-1)+2} = \begin{bmatrix} a_x^\ell(1450) & a_x^\ell(1451) & \cdots & a_x^\ell(7251) \\ a_y^\ell(1450) & a_y^\ell(1451) & \cdots & a_y^\ell(7251) \\ a_z^\ell(1450) & a_z^\ell(1451) & \cdots & a_z^\ell(7251) \\ \omega_x^\ell(1450) & \omega_x^\ell(1451) & \cdots & \omega_x^\ell(7251) \\ \omega_y^\ell(1450) & \omega_y^\ell(1451) & \cdots & \omega_y^\ell(7251) \\ \omega_z^\ell(1450) & \omega_z^\ell(1451) & \cdots & \omega_z^\ell(7251) \end{bmatrix}$$

$$TS_{3i} = \begin{bmatrix} a_x^\ell(2903) & a_x^\ell(2905) & \cdots & a_x^\ell(8704) \\ a_y^\ell(2903) & a_y^\ell(2905) & \cdots & a_y^\ell(8704) \\ a_z^\ell(2903) & a_z^\ell(2905) & \cdots & a_z^\ell(8704) \\ \omega_x^\ell(2903) & \omega_x^\ell(2905) & \cdots & \omega_x^\ell(8704) \\ \omega_y^\ell(2903) & \omega_y^\ell(2905) & \cdots & \omega_y^\ell(8704) \\ \omega_z^\ell(2903) & \omega_z^\ell(2905) & \cdots & \omega_z^\ell(8704) \end{bmatrix}$$

Here, the matrices have a size of $6 \times 5800$ and $1 \leq i \leq 72$. This allows us to represent the information by using a third-order tensor, namely, $\mathcal{Z} \in \mathbb{R}^{216\times 6\times 5800}$ defined by

$$\mathcal{Z}_{i,j,k} := (TS_i)_{j,k}$$

for $1 \leq i \leq 216$, $1 \leq j \leq 6$ and $1 \leq k \leq 5800$. We can identify $Z_i = TS_i$ for $1 \leq i \leq 216$.

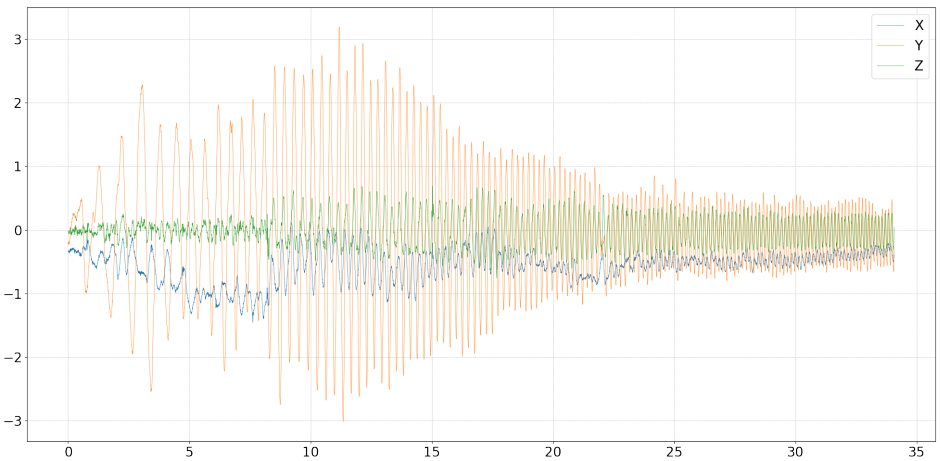



Figure 28: Sensor data: linear acceleration time series.

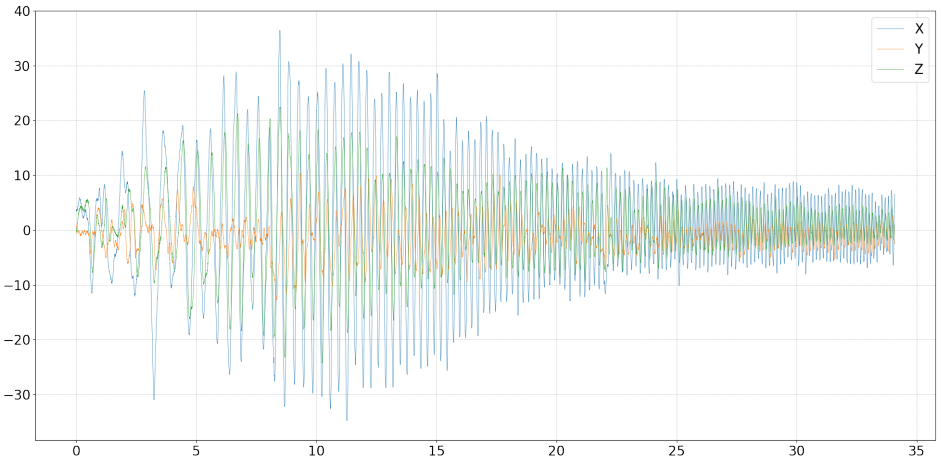



Figure 29: Sensor data: angular velocity time series.

## 4.4 Data Preprocessing

In order to obtain a single series for each observation, we concatenated all of the 6 time series (linear accelerations and angular velocities) for each observation horizontally and then created a data frame by stacking the 216 in sample observations.

The concatenation operation on the multidimensional time series collapsed the last two dimensions into one dimensional arrays with a length of $5800 \times 6 = 34{,}800$. The result is the two-dimensional table of concatenated time series

$$D = \begin{bmatrix} vec(\mathcal{Z}_{1,:,:}) \\ \dots \\ vec(\mathcal{Z}_{216,:,:}) \end{bmatrix} \in \mathbb{R}^{216 \times 34800}.$$

We chose not to filter the signals because the topological sub-level set method should filter the high-frequency features naturally. We also chose to keep working on acceleration signals in order to avoid signal deviations after two integrations in time so as to obtain positions, the sensors not always keeping a zero mean height. Thus, the approach is completely (topologically) data-based.

The six time series $\mathcal{Z}_i$ of each observation were collapsed into a single concatenated time series with a size of 34,800—see Figure 30. The concatenated time series for the 216 observations were then stacked to create the dataset $D$ with a size of 216× 34,800. We also used binary labels in the chained time series $\mathcal{Z}_i$ on the two target classes that we were interested in. In particular, we wrote $\mathcal{Z}_i^{(\alpha)}$ where $\alpha$ is "0" for a relaxed driver and "1" for a tense one.

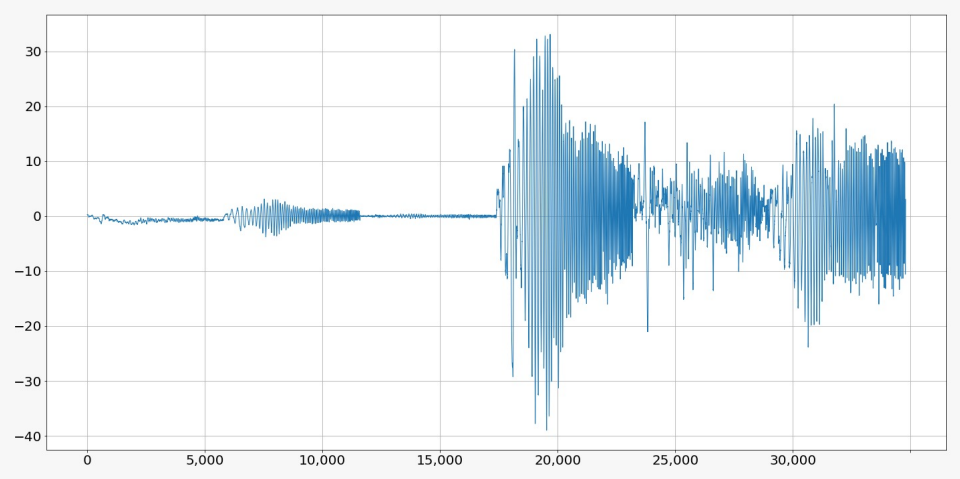



Figure 30: Tensor reduction of a sensor time series.

## 4.5 Extracting Topological Features from a Time Series

The idea to extract the topological information regarding the times series is to consider each sample observation as a piece-wise linear continuous map from a closed interval to the real line. Therefore, we used a construction closely related to the Reeb graph [37] used in Morse theory to describe the times series at the topological level.

To this end, we consider the time series $x_t$ for $0 \leq t \leq N-1$ ($N \geq 3$) given by a vector

$$\mathbf{X} = (x_0, x_1, \ldots, x_{N-1}) \in \mathbb{R}^N.$$

we can view $\mathbf{X}$ as a function also denoted by $\mathbf{X} : \{0, 1, \ldots, N-1\} \longrightarrow \mathbb{R}$ defined by $\mathbf{X}(i) = x_i$ for $0 \leq i \leq N-1$. Here, to study the topological features of $\mathbf{X}$ we use the sub-level set of a piece-wise linear function $f_{\mathbf{X}} : \mathbb{R} \longrightarrow \mathbb{R}$ associated with $\mathbf{X}$ satisfying that $f_{\mathbf{X}}(i) = \mathbf{X}(i) = x_i$ for $0 \leq i \leq N-1$.

To construct this function, we consider the basis functions $\{\varphi_0, \ldots, \varphi_{N-1}\}$ of continuous functions $\varphi_i : \mathbb{R} \longrightarrow \mathbb{R}$ defined by

$$\varphi_i(s) := \begin{cases} s - i + 1 & \text{if} \quad i - 1 \leq s \leq i \\ i + 1 - s & \text{if} \quad i \leq s \leq i + 1 \\ 0 & \text{if} \quad s \notin ]i-1, i+1[ \end{cases}$$

where $i = 1, \ldots, N-2$ and

$$\varphi_0(s) := \begin{cases} 1 - s & \text{if} \quad 0 \leq s \leq 1 \\ 0 & \text{if} \quad s \in [0, 1] \end{cases}$$

$$\varphi_{N-1}(s) := \begin{cases} s - N + 2 & \text{if} \quad N-2 \leq s \leq N-1 \\ 0 & \text{if} \quad s \notin [N-2, N-1[ \end{cases}$$

This allows us to construct a piece-wise continuous map $f_{\mathbf{X}} : \mathbb{R} \longrightarrow \mathbb{R}$ by

$$f_{\mathbf{X}}(s) = \sum_{j=0}^{N-1} x_j \varphi_j(s),$$

and also to endow $\mathbb{R}^N$ with a norm given by

$$\|\mathbf{X}\| := \|f_{\mathbf{X}}\|_{L^2(\mathbb{R})} = \left(\int_{-\infty}^{\infty} |f_{\mathbf{X}}(s)|^2 ds\right)^{1/2}.$$

In particular, we prove the following result, which helps us to identify the time series given by the vector $\mathbf{X}$ in $\mathbb{R}^N$ with the function $f_{\mathbf{X}}$ in $L^2(\mathbb{R})$.

**Proposition 4.** *The linear map $\Phi : (\mathbb{R}^N, \|\cdot\|) \longrightarrow (L^2(\mathbb{R}), \|\cdot\|_{L^2(\mathbb{R})})$ given by $\Phi(\mathbf{X}) = f_{\mathbf{X}}$ is an injective isometry between Hilbert spaces. Furthermore, $\Phi(\mathbb{R}^N)$ is a closed subspace in $L^2(\mathbb{R}^N)$.*

*Proof.* The map is clearly isometric and injective because $\{\varphi_0, \ldots, \varphi_{N-1}\}$ is a set of linear independent functions in $L^2(\mathbb{R})$. □

Here, we describe the maps $f_{\mathbf{X}} \in \Phi(\mathbb{R}^N)$ at the combinatorial level using the connected components (intervals) associated with its $\lambda$ sub-level sets

$$\mathcal{LS}_\lambda(f_{\mathbf{X}}) := \{x \in [0, N-1] : f_{\mathbf{X}}(x) \leq \lambda\}$$

for $\lambda \in \mathbb{R}$. For this purpose, we introduce the following distinguished objects related to the $\text{supp}(f_{\mathbf{X}}) = [0, N-1] \subset \mathbb{R}$ of $f_{\mathbf{X}}$ :

- The nodes or vertices denoted by

  $$\mathcal{V} := \{[0], [1], \ldots, [N-1]\}$$

  that represent the components of the vector $\mathbf{X}$,;

- The faces denoted by

  $$\mathcal{F} := \{[0,1][1,2], \ldots, [N-2, N-1]\}$$

  that represent the intervals used to construct the connected components of the sub-level sets of the map $f_{\mathbf{X}}$. Recall that we consider

  $$[i, i+1] := \{z \in \mathbb{R} : z = \mu\, x_{i+1} + (1-\mu)x_i,\ 0 \leq \mu \leq 1\} \subset \mathbb{R}.$$

Let

$$\lambda_{\max} = \max_{s \in [0, N-1]} f_{\mathbf{X}}(s) = \max_{0 \leq i \leq N-1} \mathbf{X}(i),$$

and

$$\lambda_{\min} = \min_{s \in [0, N-1]} f_{\mathbf{X}}(s) = \min_{0 \leq i \leq N-1} \mathbf{X}(i).$$

For each $\lambda_{\min} \leq \lambda \leq \lambda_{\max}$, we introduce the following symbolic $\lambda$ sub-level set for the map $f_{\mathbf{X}}$ :

$$LS_\lambda(f_{\mathbf{X}}) := \{\sigma \in \mathcal{F} : f(\sigma) \leq \lambda\}$$

For $\lambda_{\min} \leq \lambda \leq \lambda' \leq \lambda_{\max}$, it holds

$$LS_\lambda(f_{\mathbf{X}}) \subset LS_{\lambda'}(f_{\mathbf{X}}).$$

Our next goal was to quantify the evolution of the above symbolic $\lambda$ sub-level with. To this end, we introduce the notion of feature associated with the $\lambda$ sub-level set $LS_\lambda(f_{\mathbf{X}})$.

We define the set of features for functions in $\Phi(\mathbb{R}^N)$ as

$$\mathfrak{F}(\Phi(\mathbb{R}^N)) := \{[i,j] \subset \mathbb{R} : 0 \leq i < j \leq N-1\}.$$

We note that $LS_\lambda(f_{\mathbf{X}}) \subset \mathcal{F} \subset \mathfrak{F}(\Phi(\mathbb{R}^N))$. Then next definition introduces the notion of features for a symbolic $\lambda$ sub-level set as the interval of $\mathfrak{F}(\Phi(\mathbb{R}^N))$ constructed by a maximal union of faces of $LS_\lambda(f_{\mathbf{X}})$.

**Definition 9.** *We suggest that $\mathbb{I} \in \mathfrak{F}(\Phi(\mathbb{R}^N))$ is a feature for the symbolic $\lambda$ sub-level set $LS_\lambda(f_{\mathbf{X}})$ if there exists $\mathbb{I}_1, \ldots, \mathbb{I}_k \in LS_\lambda(f_{\mathbf{X}})$ such that $\mathbb{I} = \bigcup_{j=1}^k \mathbb{I}_k$ and for every $\mathbb{J} \in LS_\lambda(f_{\mathbf{X}})$ such that $\mathbb{J} \neq \mathbb{I}_i$ for $1 \leq i \leq k$ it holds that $\mathbb{I} \cap \mathbb{J} = \emptyset$. We denote by $\mathfrak{F}(LS_\lambda(f_{\mathbf{X}}))$ the set of features for the $\lambda$ sub-level set $LS_\lambda(f_{\mathbf{X}})$.*

A feature for a $\lambda$ sub-level set $LS_\lambda(f_{\mathbf{X}})$ is the maximal interval of $\mathfrak{F}(\Phi(\mathbb{R}^N))$ that we can construct by unions of intervals in $LS_\lambda(f_{\mathbf{X}})$. To illustrate this definition, we give the following example:

**Example 4.** *Let us consider the time series*

$$\mathbf{X} = (11, 14, 9, 7, 9, 7, 8, 10, 9).$$

*This allows us to construct the map $f_{\mathbf{X}}$ as shown in Figure 31. Then, $\lambda_{min} = 7$ and $\lambda_{\max} = 14$, and we have the following symbolic $\lambda$ sub-level sets.*

$$\begin{aligned}
LS_{\lambda=7}(f_{\mathbf{X}}) &= \emptyset \\
LS_{\lambda=8}(f_{\mathbf{X}}) &= LS_{\lambda=7}(f_{\mathbf{X}}) \cup \{[5,6]\} \\
LS_{\lambda=9}(f_{\mathbf{X}}) &= LS_{\lambda=8}(f_{\mathbf{X}}) \cup \{[2,3],[3,4],[4,5]\} \\
LS_{\lambda=10}(f_{\mathbf{X}}) &= LS_{\lambda=9}(f_{\mathbf{X}}) \cup \{[6,7],[7,8]\} \\
LS_{\lambda=11}(f_{\mathbf{X}}) &= LS_{\lambda=10}(f_{\mathbf{X}}) \\
LS_{\lambda=12}(f_{\mathbf{X}}) &= LS_{\lambda=11}(f_{\mathbf{X}}) \\
LS_{\lambda=13}(f_{\mathbf{X}}) &= LS_{\lambda=11}(f_{\mathbf{X}}) \\
LS_{\lambda=14}(f_{\mathbf{X}}) &= LS_{\lambda=11}(f_{\mathbf{X}}) \cup \{[0,1]\}.
\end{aligned}$$

*This allows us to compute the available features for each $\lambda$-value:*

| | $\lambda=7$ | $\lambda=8$ | $\lambda=9$ | $\lambda=10$ | $\lambda=11$ | $\lambda=12$ | $\lambda=13$ | $\lambda=14$ |
|---|---|---|---|---|---|---|---|---|
| $\mathfrak{F}(LS_\lambda(f_{\mathbf{X}}))$ | $\emptyset$ | $[5,6]$ | $[2,6]$ | $[2,8]$ | $[2,8]$ | $[2,8]$ | $[2,8]$ | $[0,8]$ |

Let $\mathfrak{F}(f_{\mathbf{X}})$ be the whole set of features for $f_{\mathbf{X}}$, that is,

$$\mathfrak{F}(f_{\mathbf{X}}) = \{\mathbb{I} : \mathbb{I} \in \mathfrak{F}(LS_\lambda(f_{\mathbf{X}})) \text{ for some } \lambda_{\min} \leq \lambda \leq \lambda_{\max}\}.$$

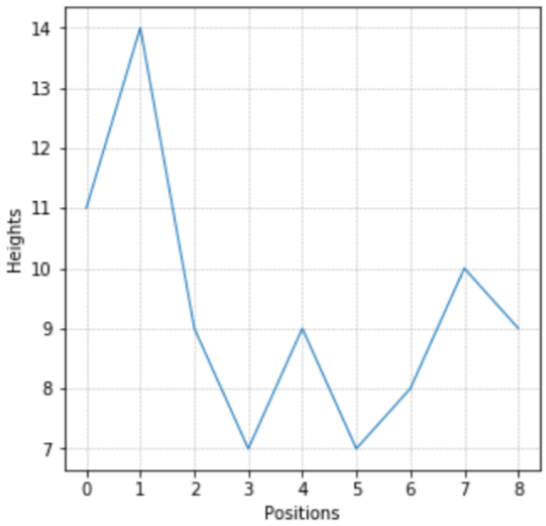



Figure 31: The map $f_{\mathbf{X}}$ for $\mathbf{X} = (11, 14, 9, 7, 9, 7, 8, 10, 9)$.

**Example 5.** *From Example 4, we obtain*

$$\mathfrak{F}(f_{\mathbf{X}}) = \{[5, 6], [2, 6], [2, 8], [0, 8]\}.$$

*We can represent the map* $\lambda \mapsto LS_\lambda(f_{\mathbf{X}})$ *from* $[\lambda_{\min}, \lambda_{\max}]$ *to* $\mathfrak{F}(f_{\mathbf{X}})$ *as shown in Figure 32.*

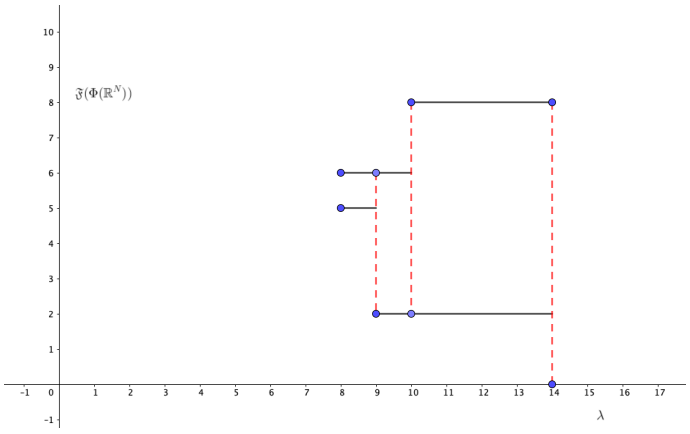



Figure 32: The map $\lambda \mapsto LS_\lambda(f_{\mathbf{X}})$ for $\mathbf{X} = (11, 14, 9, 7, 9, 7, 8, 10, 9)$.

Let $\mathbb{I} \in \mathfrak{F}(f_{\mathbf{X}})$; in order to quantify the persistence of this particular feature for the map $f_{\mathbf{X}}$, we use the map $\lambda \mapsto LS_\lambda(f_{\mathbf{X}})$ from $[\lambda_{\min}, \lambda_{\max}]$ to $\mathfrak{F}(f_{\mathbf{X}})$. To this end,

we introduce the following definition: the birth point of the feature $\mathbb{I}$ is defined by

$$a(\mathbb{I}) = \inf\{\lambda : \mathbb{I} \in \mathfrak{F}(LS_\lambda(f_{\mathbf{X}}))\}$$

and the corresponding death point by

$$b(\mathbb{I}) = \sup\{\lambda : \mathbb{I} \in \mathfrak{F}(LS_\lambda(f_{\mathbf{X}}))\}.$$

In particular, we note that $a([0, N-1]) = \lambda_{\max}$ (see Figure 32). Since $a(\mathbb{I}) \leq b(\mathbb{I}) < \infty$ holds for all $\mathbb{I} \in \mathfrak{F}(f_{\mathbf{X}})$, $\mathbb{I} \neq [0, N-1]$, we call the finite interval $[a(\mathbb{I}), b(\mathbb{I})]$ *the barcode of the feature* $\mathbb{I} \in \mathfrak{F}(f_{\mathbf{X}}) \setminus \{[0, N-1]\}$.

**Example 6.** *From Example 4 we consider the features* $[5,6] \in LS_{\lambda=8}(f_{\mathbf{X}})$, $[2,6] \in LS_{\lambda=9}(f_{\mathbf{X}})$, *and* $[2,8] \in LS_{\lambda=10}(f_{\mathbf{X}})$. *Then, the feature* $[5,6]$ *has its birth point at* $a([5,6]) = 8$ *and its death point at* $b([5,6]) = 9$; *the feature* $[2,6]$ *has its birth point at* $a([2,6]) = 9$ *and its death point at* $b([2,6]) = 10$. *Finally, the feature* $[2,8]$ *has its birth point at* $a([2,8]) = 10$ *and its death point at* $b([2,8]) = 14$. *As a consequence, the set*

$$\mathcal{B}(f_{\mathbf{X}}) := \{([5,6]; 8, 9), ([2,6]; 9, 10), ([2,8]; 10, 14)\}$$

*of features and its corresponding barcodes contain the relevant information of the shape of* $f_{\mathbf{X}}$ *(see Figure 32).*

Thus, we define the set of barcodes for $f_{\mathbf{X}}$ by

$$\mathcal{B}(f_{\mathbf{X}}) = \{(\mathbb{I}; a(\mathbb{I}), b(\mathbb{I})) : \mathbb{I} \in \mathfrak{F}(f_{\mathbf{X}}) \setminus \{[0, N-1]\}\}$$

and its persistence diagram as

$$\mathcal{PD}(f_{\mathbf{X}}) := \left\{(a(\mathbb{I}), b(\mathbb{I})) \in \mathbb{R}^2 : \mathbb{I} \in \mathfrak{F}(f_{\mathbf{X}}) \setminus \{[0, N-1]\}\right\}$$

(see Figure 33). An equivalent representation of the persistence diagram is the lifetime diagram for $f_{\mathbf{X}}$, which is constructed by means of a bijective transformation $T(a, b) = (a, b - a)$, acting over $\mathcal{PD}(f_{\mathbf{X}})$, that is,

$$\mathcal{LT}(f_{\mathbf{X}}) := \left\{(a(\mathbb{I}), b(\mathbb{I}) - a(\mathbb{I})) \in \mathbb{R}^2 : \mathbb{I} \in \mathfrak{F}(f_{\mathbf{X}})) \setminus \{[0, N-1]\}\right\};$$

see Figure 34.

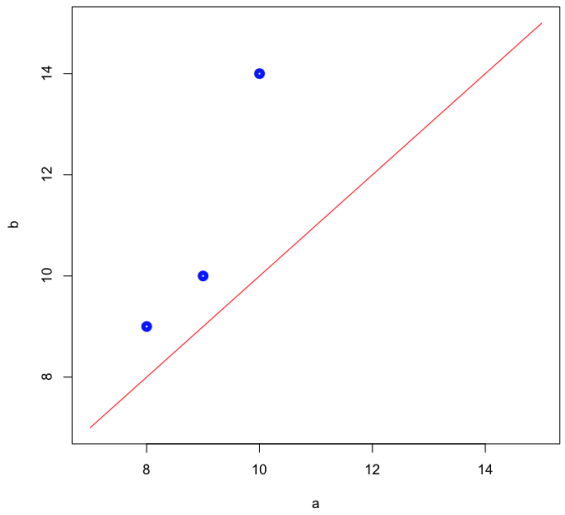



Figure 33: Persistence diagram for the map $f_{\mathbf{X}}$ when $\mathbf{X} = (11, 14, 9, 7, 9, 7, 8, 10, 9)$.

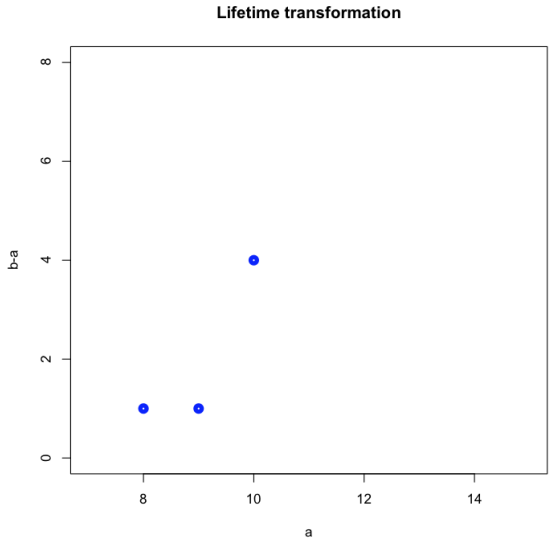



Figure 34: Life-time diagram for the map $f_{\mathbf{X}}$ when $\mathbf{X} = (11, 14, 9, 7, 9, 7, 8, 10, 9)$.

In order to determine the grade of similarity between two barcodes from two different time series, we need to set a similarity metric. To this end, we construct the persistent image for $f_{\mathbf{X}}$ as follows: we observe that $\mathcal{LT}(f_{\mathbf{X}})$ is a finite set of points, namely,

$$\mathcal{LT}(f_{\mathbf{X}}) = \{(a_1, b_1 - a_1), \ldots, (a_k, b_k - a_k)\}$$

for some natural numbers $k \geq 1$ and such that $b_1 - a_1 \leq b_2 - a_2 \ldots \leq b_k - a_k$. Then, we consider a non-negative weighting function $w : \mathcal{LT}(f_{\mathbf{X}}) \longrightarrow [0, 1]$ given by

$$w(a_i, b_i - a_i) = \frac{b_i - a_i}{b_k - a_k} \text{ for } 1 \leq i \leq k.$$

Finally, we fix $M$, a natural number, and take a bivariate normal distribution $g_u(x, y)$ centered at each point $u \in \mathcal{LT}(f_{\mathbf{X}})$ and with its variance $\sigma\, id = \frac{1}{M} \max_{1 \leq i \leq k}(b_i - a_i)\, id$, where $id$ is the $2 \times 2$ identity matrix. A persistence kernel is then defined by means of a function $\rho_{\mathbf{X}} : \mathbb{R}^2 \to \mathbb{R}$, where

$$\rho_{\mathbf{X}}(x, y) = \sum_{u \in \mathcal{LT}(f_{\mathbf{X}})} w(u) g_u(x, y). \tag{23}$$

We associate with each $\mathbf{X} \in \mathbb{R}$ a matrix in $\mathbb{R}^{M \times M}$ as follows: let $\varepsilon > 0$ be a non-negative real number that is sufficiently small, and then consider a square region $\Omega_{\mathbf{X},\varepsilon} = [\alpha, \beta] \times [\alpha^*, \beta^*] \subset \mathbb{R}^2$, covering the support of $\rho_{\mathbf{X}}(x, y)$ (up to a certain precision), such that

$$\iint_{\Omega_{\mathbf{X},\varepsilon}} \rho_{\mathbf{X}}(x, y)\, dx\, dy \geq 1 - \varepsilon$$

holds. Next, we consider two equispaced partitions of the intervals

$$\alpha = p_0 \leq p_1 \ldots \leq p_M = \beta \text{ and } \alpha^* = p_0^* \leq p_1^* \ldots \leq p_M^* = \beta^*.$$

Now, we put

$$\Omega_{\mathbf{X},\varepsilon} = \bigcup_{i=0}^{M-1} \bigcup_{j=0}^{M-1} [p_i, p_{i+1}] \times [p_j^*, p_{j+1}^*] = \bigcup_{i=0}^{M-1} \bigcup_{j=0}^{M-1} P_{i,j}$$

The persistence image of $\mathbf{X}$ associated with the partition $\mathcal{P} = \{P_{i,j}\}$ is then described by the matrix given by the following equation:

$$PI(\mathbf{X}, M, \mathcal{P}, \varepsilon) = \left( \iint_{P_{i,j}} \rho_{\mathbf{X}}(x, y) dx dy \right)_{i=0,j=0}^{i=M-1,j=M-1} \in \mathbb{R}^{M \times M}. \tag{24}$$

## 4.6 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 is thus based on how close a new point is to each training sample. The Euclidean distance is the most common distance metric 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 diagrams, we can use any state-of-the-art technique. In our case, we considered the random forest classification.

Recall that we conducted 9 different experiments, with 24 samples associated with each one of them corresponding to 3 samples for each of the different experimental conditions: relaxed rigid driver, relaxed rigid passenger, relaxed SAV driver, relaxed SAV passenger, tense rigid driver, tense rigid passenger, tense SAV driver,

and tense SAV passenger. Their respective labels are $\{0, 0, 0, 0, 1, 1, 1, 1\}$. Therefore, we designed the following training validation process: The model is trained over 144 samples and evaluated over the remaining unseen 72 experiments (two-to-one training-to-testing ratio). The split between training and sampling is achieved using random shuffling and stratification to ensure balance between the classes. In order to improve the evaluation of the model generalizability, we also performed a cross-validation procedure following a *leave-one-out* strategy, consisting of iteratively training over the full dataset except one sample that was left out and used to test and score the model. We used the *accuracy* metric to evaluate the classification model. We can represent the performance of the model using the so-called confusion matrix: a 2D entries table where elements account for the number of samples in each category, with the first axis representing the true labels and the second axis the predicted labels. We also computed the different classification metrics to obtain a more detailed reporting of the model performances.

## 4.7 Results

The trained random forest classifier model for the persistence images has a notably high accuracy score on the training dataset (144) for both approaches and high accuracy for the testing dataset (72 samples). This suggests strong differentiation of the images with the respect to their generating signals, see Figure 35. The scores on the training and testing are 93 and 83%, respectively. The leave-one-out cross-validation achieved a score of 81%, indicating a good variance–bias trade-off and good generalization potential of the model.

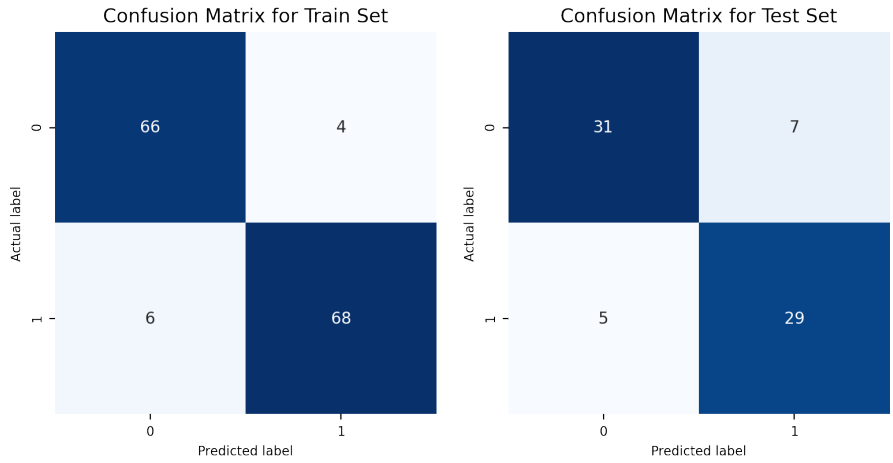



Figure 35: Model performance for predicting the attention state.

## 4.8 Discussion

The combination of Morse theory and topological data analysis allows us to extract information from real data without the need for smoothness or regularity assumption on the time series. In our case, input data for each experiment were reduced from

six-sensor time series of measurements to one single image containing the persistent pattern for attention to the road. Using the obtained persistence images as the new inputs, supervised learning proved to successfully predict the attention state of the driver or passenger.

The procedure used and described does not involve any additional pre-processing of the sensor data; is robust to noise and degraded signals; and supports large quantities of data, which makes it efficient and scalable.

# 5 Monitoring and Anticipating Robots Functioning Behaviors

In this chapter we aim 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. Topological Data Analysis will be used for extracting the trajectory descriptors. Then, appropriate metrics will be applied in order to compare that topological representation of the trajectories, for classifying them or for making efficient pattern recognition.

## 5.1 Introduction

Autonomous robots follow a number of rules introduced into their controllers [61, 62, 63]. However, when they interact with the environment, small variations may result in long-time unpredictable motion. This behavior is very usual in mechanics, characterizing systems exhibiting deterministic chaos.

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 36 depicts one of these robots considered in the present study in operational conditions.

Figure 36: Weeder robot from VITIROVER *micro robotique viticole*

All the practical variability (ground, grape location, grass distribution and size,

obstacles, ...) as well as the intrinsic sensibility of the dynamics to small variabilities in the physical and operational conditions, makes it impossible to define a deterministic robot trajectory. In these conditions, an almost random motion seems to be the most valuable alternative.

In practice, to avoid under-performances characteristic of fully random motions, that 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.

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.

The present paper aims at analyzing the collected data in order to extract the maximum information that could serve for differentiating them, enabling unsupervised clustering and/or supervised classification, prior to any action concerning modeling using adapted regressions.

## 5.2 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, 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.

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 [64, 65]. 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 nonlinear) 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 [66].

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) [67, 68], code-to-vector [16] or neural networks [69].

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 [70, 71] has been successfully applied in many domains. The theory of optimal transport arose as a response to similar issues [38].

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 modeling (from nonlinear regressions).

A first possibility consists of extracting the main statistical descriptors of time series or images (moments, correlations, covariograms, ...) [72]. 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 [24], 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 [25], time-series [2], rough surfaces [1] and shapes [3], with the aim of classifying and also constructing robust regressions ex-

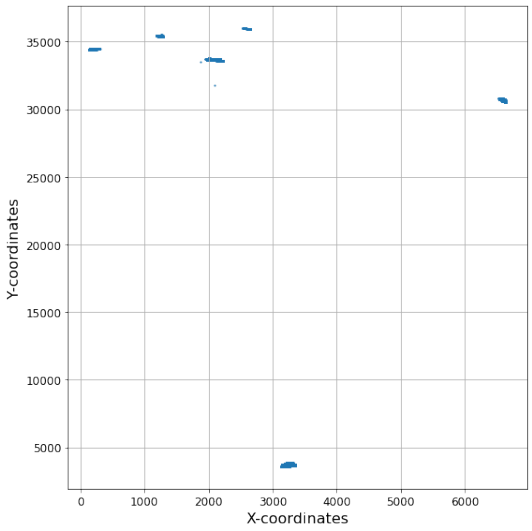



Figure 37: Location of the different patches

pressing properties or performance from the input data expressed from its topological description.

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).

### 5.2.1 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 37, that have been recorded in a period of time $\mathcal{T}$ leading to the maps reported in Figure 38 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 $\mathbb{S}_1, \ldots, \mathbb{S}_n$.

### 5.2.2 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 simplices 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 $x_1 - x_0, \ldots, x_d - x_0$

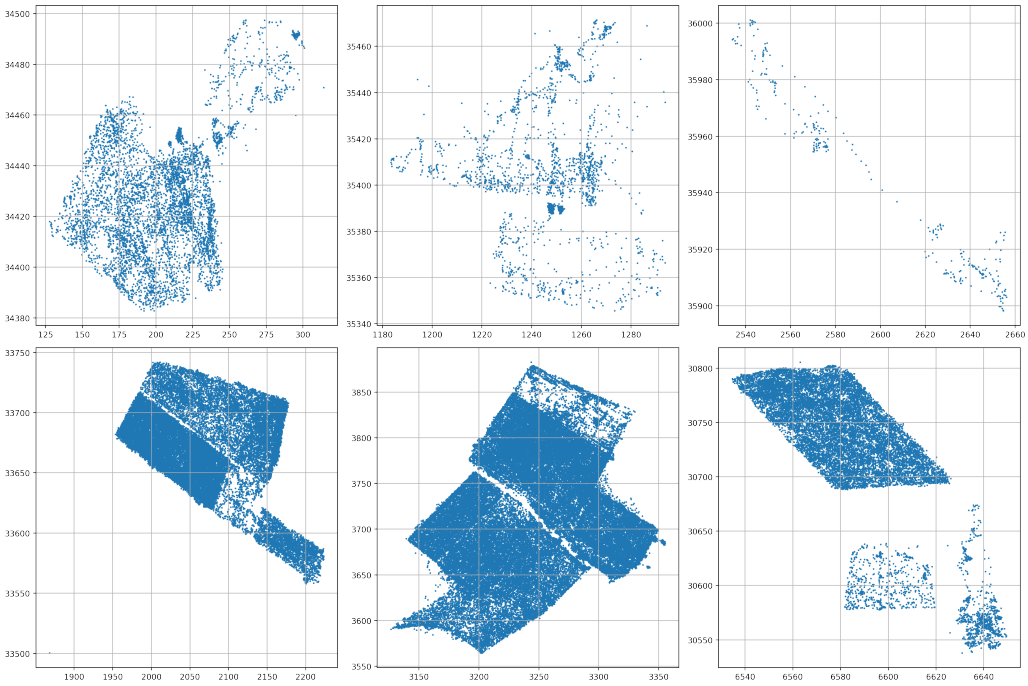


Figure 38: Robot trajectories in the six considered vineyard patches (units in meters)

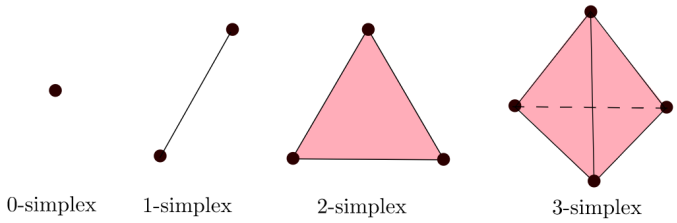



Figure 39: Simplices of different dimensions

are linearly independent, as illustrated in Fig. 39. 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 $\epsilon$, $\mathcal{R}_\epsilon(\mathbb{S})$, is the abstract simplicial complex consisting of all subsets of diameter up to $\epsilon$:

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

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 $\frac{\epsilon}{2}$, centered 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 Fig. 40.

A *filtration* of a simplicial complex $\mathcal{K}$ is a nested sequence of subcomplexes

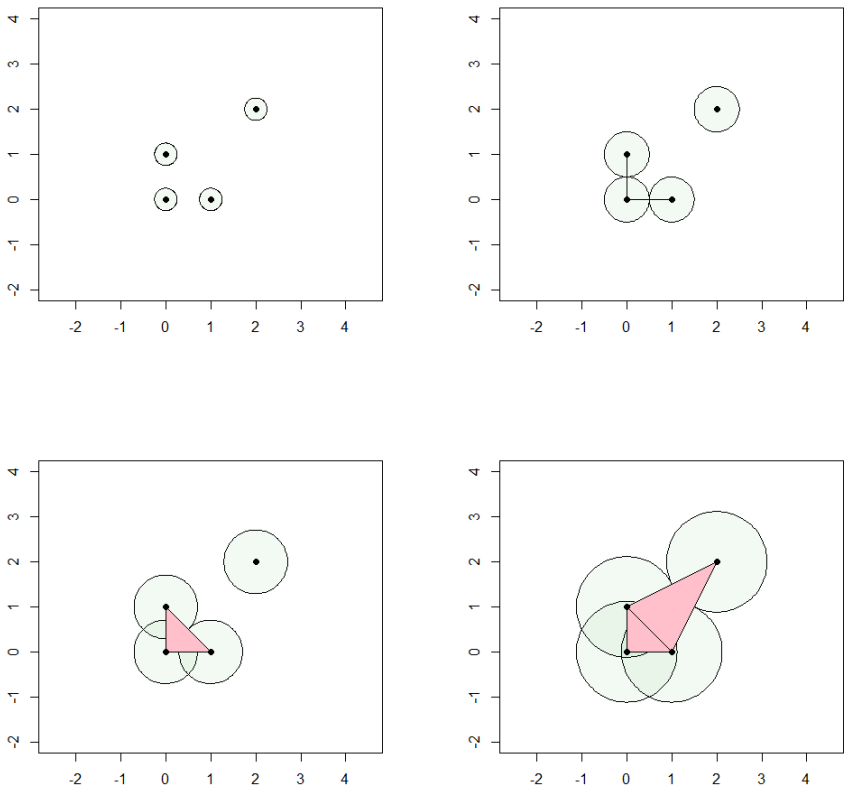



Figure 40: Example of Rips complex computation: (top-left) $\epsilon = 0.5$; (top-right) $\epsilon = 1$; (bottom-left) $\epsilon = 1.4$; and (bottom-right) $\epsilon = 2.3$.

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 $\epsilon$, from $\epsilon_{\min} = 0$ to $\epsilon_{\max} = \text{diam}(\mathbb{S})$ we get a family of nested Rips complexes known as the Rips filtration.

### 5.2.3 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 simplices ("packets") in the Rips complex. 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 $\epsilon$, $H_k^{\epsilon}(\mathbb{S})$. It represents the classes of simplices 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 $\epsilon$. 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 $\epsilon_j$, $j = 0, ..., m$) :

- 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 $b_\gamma$ of the feature $\gamma$

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

- The death scale $d_\gamma$ of the feature $\gamma$

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

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 Fig. 41, and the associated barcode in Fig. 42. 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

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

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

The persistence diagram associated with the Rips complex shown in Fig. 41 is given in Fig. 43. 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,

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

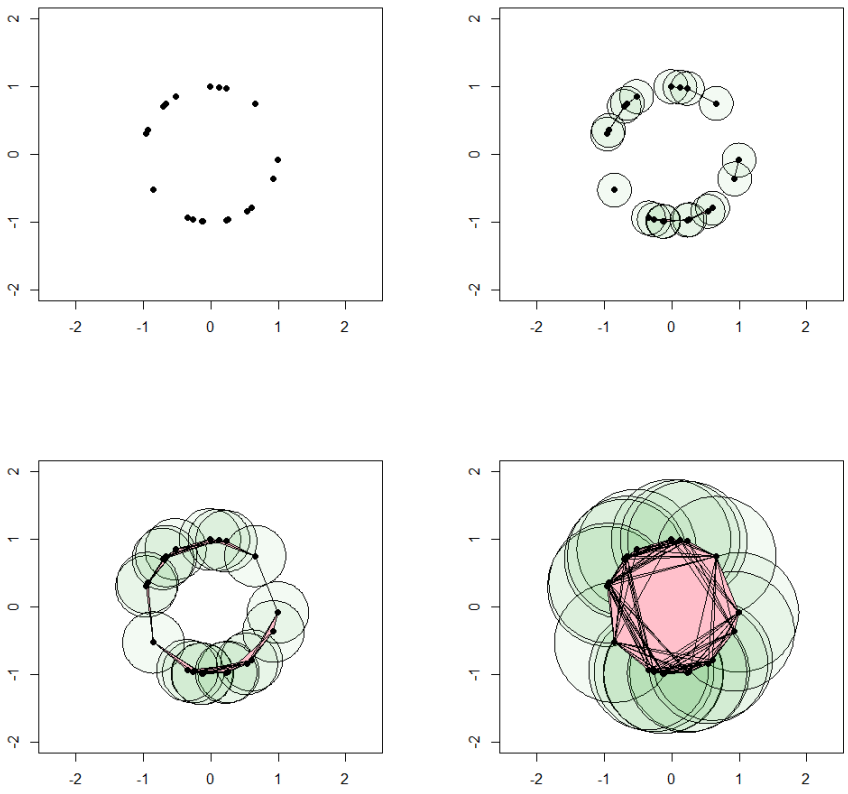


Figure 41: Example of Rips complex computation: (top-left) $\epsilon = 0$; (top-right) $\epsilon = 0.5$; (bottom-left) $\epsilon = 0.9$; and (bottom-right) $\epsilon = 1.8$.

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

$$\begin{aligned} w : \mathcal{LT}(\mathbb{S}) &\to [0, 1] \\ (a_i, b_i - a_i) &\mapsto w(a_i, b_i - a_i) = \frac{b_i - a_i}{b_p - a_p}, \text{ for } 1 \leq i \leq p. \end{aligned}$$

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 $2 \times 2$ identity matrix). A persistence kernel is then defined according to:

$$\begin{aligned} \rho_{\mathbb{S}} : \mathbb{R}^2 &\to \mathbb{R} \\ (x, y) &\mapsto \rho_{\mathbb{S}}(x, y) = \sum_{u \in \mathcal{LT}(\mathbb{S})} w(u) g_u(x, y). \end{aligned}$$

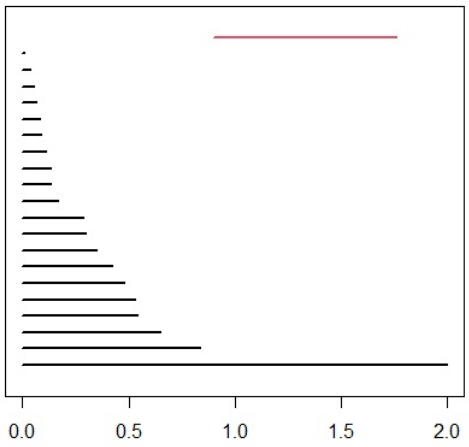



Figure 42: Persistence barcode: in black the $H_0$ features, and in red the $H_1$ feature. Filtration value (scale) is represented in the $x$-axis.

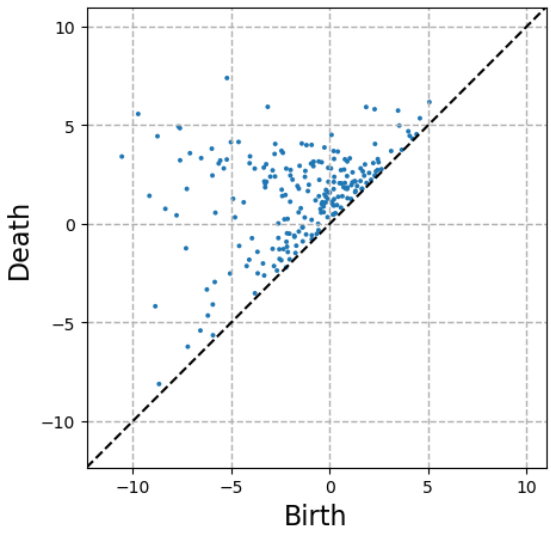



Figure 43: Persistence Diagram: in black the $H_0$ features, and in red the $H_1$ feature.

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 $\delta$, 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

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

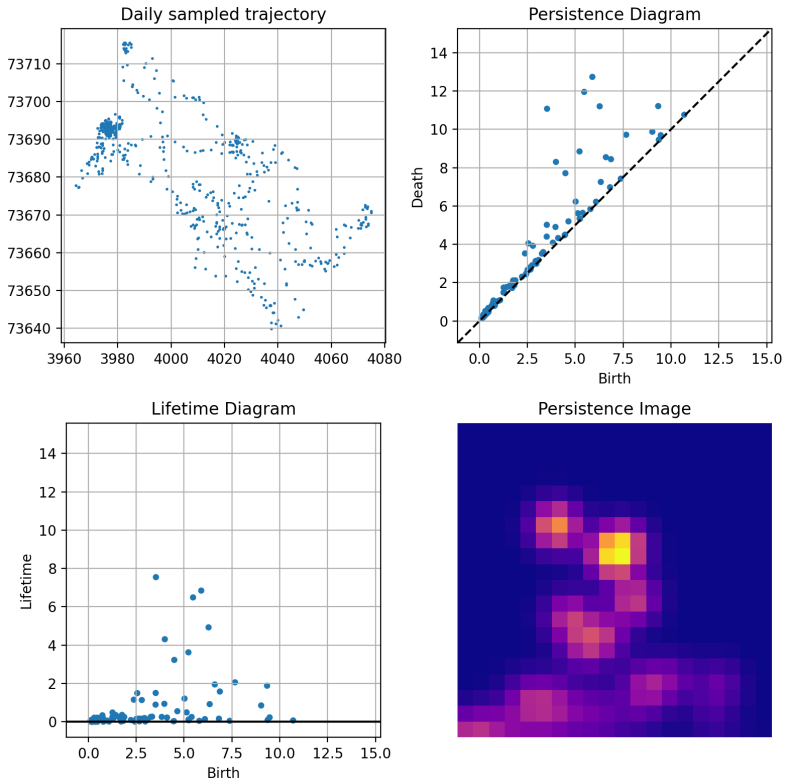



Figure 44: Topological analysis of a trajectory: (top-left) Trajectory; (top-right) Persistence diagram; (bottom-left) Lifetime diagram; and (bottom-right) Persistence Image.

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 Fig.44.

### 5.2.4 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 $\psi$, that reads:

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

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

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

The map $\psi$ 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 $\hat{\psi}$

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

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

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

Then, to measure the degree of similarity between two trajectories $\mathbb{S}_u$ and $\mathbb{S}_v$ we consider the *Wasserstein distance* [38, 39] between $\mathcal{PD}(\mathbb{S}_u)$ and $\mathcal{PD}(\mathbb{S}_v)$

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

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 Fig. 45.

### 5.2.5 Barycenters of persistence diagrams

Consider now a collection $\mathbb{S}_1 \dots \mathbb{S}_n$ of trajectories with their associated diagrams $\mathcal{PD}_1 \dots \mathcal{PD}_n$.
Since the space of persistence diagrams equipped with the Wasserstein distance, the *Wasserstein space*, is not a linear space, the notion of barycenters [40] can be extended for the persistence diagrams using the so-called *Frechet mean* [41], which always exists in the context of averaging finitely many diagrams.

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

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

The computation of the barycenter $\mu$ has proven to be challenging, and multiple approaches can be used, such as the Sinkhorn algorithm [42]. We will use the one based on the Hungarian algorithm presented in [41] and consider Partial Optimal

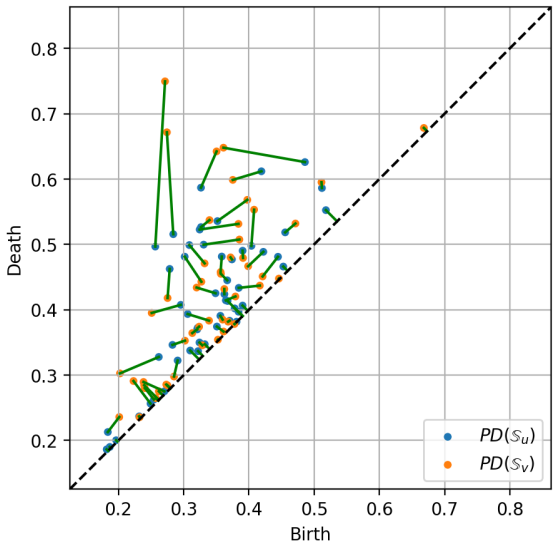



Figure 45: Optimal matching between two persistence diagrams related to two robot trajectories

Matchings [43], 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 barycenters of a finite family of persistence diagrams, taking a Lagrangian approach by tracking the individual points of the diagrams. Given a collection $\mathcal{PD}_1 \dots \mathcal{PD}_n$ of persistence diagrams, we proceed as follows:

1. Intialize the estimation $\mu$ of the barycenter at a certain diagram $\mu = \mathcal{PD}_{i_0}$.
2. Compute the optimal partial matchings $\psi_1 \dots \psi_n$, between $\mu$ and $\mathcal{PD}_1 \dots \mathcal{PD}_n$ respectively.
3. Compute the updated barycenter $\hat{\mu}$, by averaging the transport of each point in the barycenter $\mu$
$$\hat{\mu} = \{y = \frac{1}{n} \sum_{i=1}^{n} \psi_i(x), x \in \mu\}.$$
4. If $\hat{\mu}$ minimizes $\mathcal{E}$, return $\hat{\mu}$. Otherwise, update $\mu = \hat{\mu}$ and go back to 2.

An example of a barycenter of three persistence diagrams is given in Fig. 46.

### 5.2.6 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

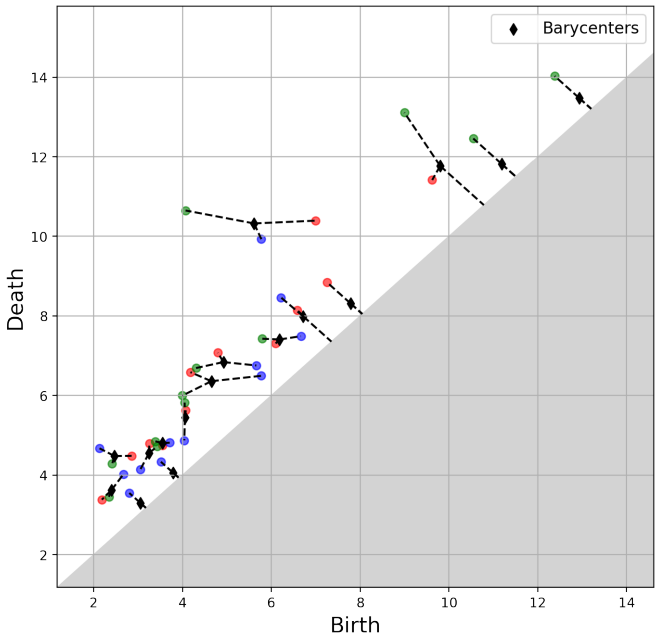



Figure 46: Barycenter (in black) of three persistence diagrams (red, blue and green)

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 $\left(\mathcal{X}_i\right)_{i=1}^{n}$ of flattened persistence images, i.e., $M \times M$-component vectors, computed from a set $\left(\mathbb{S}_i\right)_{i=1}^{n}$ 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:

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

Here $\omega$ are the weights we optimize over, $c$ a Bernouilli 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 metrics used for the model evaluation is the *Accuracy Score* defined as the number of correct predictions over the number of samples.

## 5.3 Results

### 5.3.1 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, that will be used as input for the model (a sample is depicted in Figure 44).

The samples are labelled according to the considered patch, 1 if the robot is in the target patch, and 0 otherwise. 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.

### 5.3.2 Maintenance prediction

Then, we consider daily trajectories in the same patch, 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, 1 otherwise. 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 the sampled trajectories, proving that robot trajectories exhibit a topological pattern when maintenance applies, fact that could be used for predictive maintenance purposes.

Figure 47 depicts the Wasserstein distance between the persistence diagrams for consecutive daily trajectories, with the maintenance operation emphasized in red, whereas Fig. 48 shows the barycenters of each period between consecutive maintenance operations. As it can be noticed from the persistence images in Fig. 48, 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 Fig. 48 (left), and divide it in two parts with identical length. Then, the associated barycenters in both half intervals are obtained. Both are represented in Fig. 49. As it can be noticed, both of them resemble very much to the one associated to the whole interval (the first picture in Fig. 48), all them (both in Fig. 49 and the first in Fig. 48) are significantly different to the second image in Fig. 48 that represents the trajectory topology after the first maintenance operation. These results support again our assumption on the effect of maintenance on the trajectory topology.

## 5.4 Discussion

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 analyzing

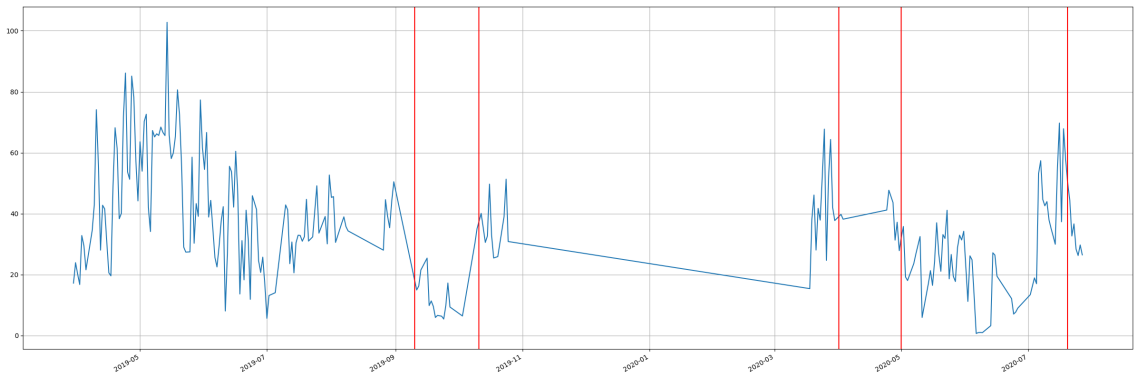



Figure 47: Time series of the Wasserstein distance between the persistence diagrams for consecutive daily trajectories: in red the maintenance events.

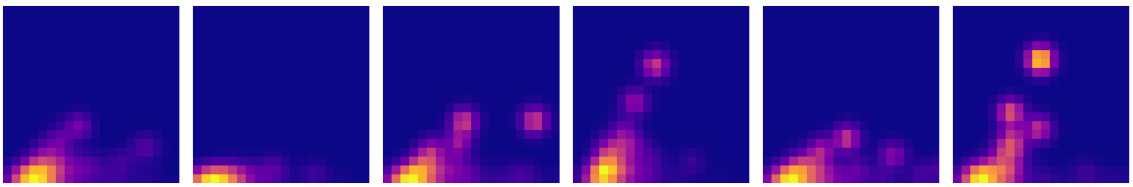


Figure 48: Persistence images of the barycenters computed for each period

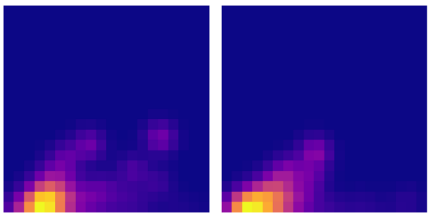


Figure 49: Persistence images of the two half-intervals related to the first period whose persistence image was the first image in Fig. 48

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.

# 6 Conclusions

The topological data analysis and persistent homology has proven to be a reliable and useful approach to study the changes in the observed systems without a prior knowledge of the physical phenomenon and modeling.

This methodology is particularly adapted for studying data sets with high topological and geometrical information such as shapes, signals, surfaces and trajectories. By extracting the underlying algebraic structure of the data it is possible to compare and detect changes in the studied systems and dynamics, extract statistical descriptors and characterize the physical systems.

It is a robust and model agnostic methodology, with promising generalization possibilities. Moreover, it does not require further continuity hypothesis on the data manifold, while being scale and dimension sensitive.

Given the described framework and computations, it is essential to have a clear specification of a given problem dimensionality in order to have an adapted geometrical description: one-dimensional (such as univariate time series), two-dimensional (such as planar trajectories), three-dimensional (such as deforming shapes), multi-dimensional (multivariate time-series). The data size is also a crucial parameter, as it may requires a particular choice of filtration. Additionally, some data specificity can play a role such as Alpha filtration (triangulation) for meshed surfaces, Rips filtration (spheres) for diffusion like dynamics, and Sublevelset filtration for sequential data. Finally, specific and custom metrics (optimal transport) allow to leverage the computed persistence for the most relevant feature extraction. The properties of these features (vector space, stability) will largely affect the choice of further learning procedures.

This framework displays very promising capabilities for further investigations and applications, such as in digital twins. It could allow to incorporate additional sensors data sets, improve behavior and regime prediction, while being robust to noise and model agnostic.

# List of Figures

# List of Tables

# References


[1] T. Frahi, C. Argerich, M. Yun, A. Falco, A. Barasinski, F. Chinesta. Tape surfaces characterization with persistence images. *AIMS Materials Science* **2020**, Volume 7, Issue 4: 364-380.

[2] T. Frahi, F. Chinesta, A. Falco, A. Badias, E. Cueto, H.Y. Choi, M. Han, J.L. Duval. Empowering Advanced Driver-Assistance Systems from Topological Data Analysis. *Mathematics* **2021**, 9, 634.

[3] T. Frahi , A. Falco , B. Vinh Mau , J.L. Duval, F. Chinesta. Empowering Advanced Parametric Modes Clustering from Topological Data Analysis. *Applied Sciences* **2021**, 11, 6554.

[4] T. Frahi , A. Sancarlos , M. Galle , X. Beaulieu, A. Chambard, A. Falco, E. Cueto, F. Chinesta. Monitoring weeder robots and anticipating their functioning by using advanced topological data analysis. *Frontiers in AI* **2021**, Accepted.

[5] F. Chinesta, A. Leygue, F. Bordeux, J.V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Huerta. Parametric PGD based computational vademecum for efficient design, optimization and control. *Archives of Computational Methods in Engineering* **2013**, 20/1, 31-59.

[6] Chinesta F, Leygue A, Bognet B, et al. First steps towards an advanced simulation of composites manufacturing by automated tape placement. *Int J Mater Form* **2014**, 7: 81–92.

[7] Chinesta F, Ammar A, Cueto E. Recent advances and new challenges in the use of the Proper Generalized Decomposition for solving multidimensional models. *Arch Comput Method Eng* **2010**, 17: 327–350.

[8] Chinesta F, Ladeveze P, Cueto E. A short review in model order reduction based on Proper Generalized Decomposition. *Arch Comput Method Eng* **2011**, 18: 395–404.

[9] Chinesta F, Keunings R, Leygue A. The Proper Generalized Decomposition for advanced numerical simulations: A primer, *Heidelberg, New York, Dordrecht, London: Springer-Cham* **2014**.

[10] Chinesta F, Ladeveze P. Separated Representations and PGD Based Model Reduction: Fundamentals and Applications, *Springer-Verlag* **2014**.

[11] Chinesta F, Leygue A, Bordeu F, et al. Parametric PGD based computational vademecum for efficient design, optimization and control. *Arch Comput Method Eng* **2013**, 20: 31–59.

[12] Falcó A, Nouy. A Proper generalized decomposition for nonlinear convex problems in tensor banach spaces. *Numer Math* **2013**, 121: 503–530.

[13] Falcó A, Montés N, Chinesta F, et al. On the existence of a progressive variational vademecum based in the proper generalized decomposition for a class of elliptic parametrised problems. *J Comput Appl Math* **2018**, 330: 1093–1107.

[14] Leon A, Argerich C, Barasinski A, et al. Effects of material and process parameters on in-situ consolidation. *Int J Mater Form* **2018**, 12: 491–503.

[15] Argerich C, Ruben I, Leon A, et al. Tape surface characterization and classification in automated tape placement processability: Modeling and numerical analysis. *AIMS Mater Sci*, **2018**, 5: 870–888.

[16] Argerich C, Ruben I, Leon A, et al. *Code2Vect*: An efficient heterogenous data classifier and nonlinear regression technique. *CR Mecanique* **2019**, 347: 754–761.

[17] G. Quaranta, C. Argerich, R. Ibañez, J. L. Duval, E. Cueto and F. Chinesta. From linear to nonlinear PGD-based parametric structural dynamics. *Comptes Rendus Mécanique* **2019**, 347(5), 445-454.

[18] M. H. Malik, D. Borzacchiello, J. V. Aguado and F. Chinesta. Advanced parametric space-frequency separated representations in structural dynamics: A harmonic–modal hybrid approach. *Comptes Rendus Mécanique* **2018**, 346(7), 590-602.

[19] A. Sancarlos, V. Champaney, J.L. Duval, E. Cueto, F. Chinesta. PGD-based advanced nonlinear multiparametric regressions for constructing metamodels at the scarce-data limit. Available online (arxiv).

[20] C. Germoso, J.L. Duval, F. Chinesta. Harmonic-Modal Hybrid Reduced Order Model for the Efficient Integration of Non-Linear Soil Dynamics. *Applied Sciences* **2020**, 10(19), 6778.

[21] D. Gonzalez, E. Cueto and F. Chinesta. Real-Time Direct Integration of Reduced Solid Dynamics Equations. *International Journal for Numerical Methods in Engineering* **2014**, 99/9, 633-653.

[22] Leon A, Barasinski A, Nadal E, et al. High-resolution thermal analysis at thermoplastic pre-impregnated composite interfaces. *Compos Interface* **2015**, 22: 767–777.

[23] A. Leon, A. Barasinski, F. Chinesta Microstructural analysis of pre-impreganted tapes consolidation. *Int J Mater Form* **2017**, 10: 369–378.

[24] R. Ibanez, E. Abisset-Chavanne, E. Cueto, A. Ammar, J.L. Duval, F. Chinesta. Some applications of compressed sensing in computational mechanics. Model order reduction, manifold learning, data-driven applications and nonlinear dimensionality reduction. *Computational Mechanics* **2019**, 64, 1259-1271.

[25] M. Yun, C. Argerich, E. Cueto, J.L. Duval, F. Chinesta. Nonlinear regression operating on microstructures described from Topological Data Analysis for the real-time prediction of effective properties. *Materials* **2020** 13/10, 2335.

[26] H. Edelsbrunner, J.L. Harer. Computational Topology: An Introduction. *American Mathematical Society* **2009**

[27] Rabadan R, Blumberg AJ. Topological Data Analysis For Genomics And Evolution *Cambridge University Press* **2020**.

[28] Oudot SY. Persistence Theory: From Quiver Representation to Data Analysis, *American Mathematical Society, Mathematical Surveys and Monographs* **2010** 209.

[29] Chazal F, Michel B (2017) An introduction to topological data analysis: fundamental and practical aspects for data scientists. *arXiv* **2017**, 1710.04019.

[30] Carlsson G. Topology and data. *Bull Amer Math Soc* **2009** 46: 255–308.

[31] Krishnapriyan AS, Haranczyk M, Morozov D. Robust topological descriptors for machine learning prediction of guest adsorption in nanoporous materials. *arXiv* **2020**, 2001.05972.

[32] Carlsson G, Zomorodian A, Colling A, et al. Persistence Barcodes for Shapes, *Eurographics Symposium on Geometry Processing* **2004**, 127-138

[33] Adams H, Chepushtanova S, et al. Persistence images: A stable vector representation of persistent homology. *J Mach Learn Res* **2016**, 18: 218–252.

[34] Epstein, C.; Carlsson, G.; Edelsbrunner, H. Topological data analysis. *Inverse Problems* **2011**, 27, 120201.

[35] Milnor, J. Morse Theory. *Princeton University Press* **1963**.

[36] Milnor, J.W.; Thurston, W. On iterated maps of the interval, Dynamical systems. *Lecture Notes in Mathematics, Springer* **1988**, pp. 465–563.

[37] Reeb, G. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *C. R. Acad. Sci. Paris* **1946**, 222, 847–849.

[38] C. Villani, Optimal transport, old and new. *Springer* **2006**.

[39] G. Peyre, M. Cuturi.Computational Optimal Transport. *Foundations and Trends in Machine Learning* **2019**, 11/5-6, 355-607.

[40] M. Agueh, G. Carlier. Barycenters in the Wasserstein Space. *SIAM J. Math. Anal.* **2011**, 43/2, 904-924.

[41] K. Turner, Y. Mileyko, S. Mukherjee, J. Harer. Frechet Means for Distributions of Persistence Diagrams. *Discrete and Computational Geometry*, **2014** 52, 44-70.

[42] M. Cuturi, A. Doucet. Fast Computation of Wasserstein Barycenters. *Proceedings of the 31st International Conference on Machine Learning* **2014**, PMLR 32(2), 685-693.

[43] V. Divol, T. Lacombe. Understanding the topology and the geometry of the space of persistence diagrams via optimal partial transport. *Journal of Applied and Computational Topology* **2021**, 5, 1-53.

[44] M. Louis, A. Bône, B. Charlier, S. Durrleman. Parallel Transport in Shape Analysis: A Scalable Numerical Scheme. *Lecture Notes in Computer Science* **2017**, vol 10589.

[45] M. Carrière, M. Cuturi and S. Oudot. Sliced Wasserstein Kernel for Persistence Diagrams. *Proceedings of the 34th International Conference on Machine Learning* **2017**, PMLR 70:664-673.

[46] P. Ladeveze. The large time increment method for the analyze of structures with nonlinear constitutive relation described by internal variables. *CR Acad. Sci. Paris* **1989**, 309, 1095-1099.

[47] P. Ladeveze, L. Arnaud, P. Rouch and C. Blanze, The variational theory of complex rays for the calculation of medium-frequency vibrations. *Engrg. Comp.* **1999**, 18, 193-214.

[48] L. Boucinha, A. Gravouil, A. Ammar. Space-time proper generalized decompositions for the resolution of transient elastodynamic models. *Computer Methods in Applied Mechanics and Engineering* **2013**, 255(0), 67-88.

[49] J. Bathe. Conserving energy and momentum in nonlinear dynamics: A simple implicit time integration scheme. *Computers and Structures* **2007**, 85, 437-445.

[50] R.W. Clough and J. Penzien. Dynamics of structures. *Civil engineering series McGraw-Hill* **1993**.

[51] N.M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division* **1959**, 85(EM3), 67-94.

[52] M. Pastora, M. Bindaa, T. Harcarika. Modal Assurance Criterion. *Procedia Engineering* **2012**, 48, 543–548.

[53] Yang F, Pitchumani R. A fractal cantor set based description of interlaminar contact evolution during thermoplastic composites processing. *J Mater Sci* **2001**, 36: 4661–4671.

[54] Levy A, Heider D, Tierney J, et al. Inter-layer thermal contact resistance evolution with the degree of intimate contact in the processing of thermoplastic composite laminates. *J Compos Mater* **2014**, 48: 491–503.

[55] Paul, A., Chauhan, R., Srivastava, R., Baruah, M. Advanced Driver Assistance Systems. *SAE Technical Paper SAE International: Warrendale* **2016**, 28-0223

[56] Shaout, A., Colella, D., Awad, S.S. Advanced Driver Assistance Systems-Past, Present and Future. *In Proceedings of the 2011 Seventh International Computer Engineering Conference (ICENCO'2011), Cairo, Egypt* **2016**, 72–82.

[57] Geronimo, D., Lopez, A.M., Sappa, A.D., Graf, T. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, 32, 1239–1258.

[58] Lindgren, A., Chen, F. State of the art analysis: An overview of advanced driver assistance systems (adas) and possible human factors issues. *Hum. Factors Econ. Asp. Saf.* **2006**, 38, 50.

[59] Izquierdo-Reyes, J., Ramirez-Mendoza, R.A., Bustamante-Bello, M.R. A study of the effects of advanced driver assistance systems alerts on driver performance. *Int. J. Interact. Des. Manuf.* **2018**, 12, 263–272.

[60] Sikander, G.; Anwar, S. Driver fatigue detection systems: A review. *IEEE Trans. Intell. Transp. Syst.* **2018**, 20, 2339–2352.

[61] M.B. Alatise, G.P. Hancke, A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods, *IEEE Access* **2020**, 8, 39830-39846.

[62] N. Shalal, T. Low, C. McCarthy, N. Hancock.A review of autonomous navigation systems in agricultural environments. *Innovative Agricultural Technologies for a Sustainable Future* **2013**, Barton, Western Australia.

[63] K.P. Mohanty, D.R. Parhi. Controlling the Motion of an Autonomous Mobile Robot Using Various Techniques: a Review. *Journal of Advance Mechanical Engineering* **2013**, 1, 24-39.

[64] J. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press* **1967**, 281-297.

[65] D. MacKay, Chapter 20 - An Example Inference Task: Clustering. *Information Theory, Inference and Learning Algorithms. Cambridge University Press* **2003**, 284-292.

[66] N. Cristianini, J. Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. *Cambridge University Press, New York* **2000**.

[67] C.W. Kirkwood, Decision Tree primer **2002**.

[68] L. Breiman, Random Forests. *Machine Learning* **2001**, 45, 5-32.

[69] I. Goodfellow, Y. Bengio, A. Courville. Deep learning. *MIT Press, Cambridge* **2016**.

[70] M. Muller. Information retrieval for music and motion. *Springer-Verlag Berlin Heidelberg*.

[71] P. Senin, Dynamic time warping algorithm review. *Technical report* **2008**.

[72] S. Torquato. Statistical description of microstructures. *Annu. Rev. Mater. Res.* **2002**, 32, 77-111.

[73] Pedregosa F, et al. Scikit-learn: Machine Learning in Python. *arXiv* **2011** 1201.0490v4.

[74] Saul N, Tralie C. Scikit-TDA: Topological Data Analysis for Python **2019**.

[75] GUDHI Project. GUDHI User and Reference Manual. *GUDHI Editorial Board* **2021**.