Functional and Shape Data Analysis under the Frenet-Serret Framework : Application to Sign Language Motion Trajectories Analysis

Perrine Chassat

université PARIS-SACLAY

Functional and Shape Data Analysis under the Frenet-Serret Framework: Application to Sign Language Motion Trajectories Analysis

Analyse de Données Fonctionnelles et Analyse de Formes dans le cadre de Frenet-Serret : Application à l'Analyse de Trajectoires de Mouvement de Langue des Signes

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°574 : mathématiques Hadamard (EDMH)
Spécialité de doctorat: Mathématiques appliquées
Graduate School : Mathématiques. Référent : Université d'Évry Val d'Essonne

Thèse préparée dans l'unité de recherche **Laboratoire de Mathématiques et Modélisation d'Évry (Université Paris-Saclay, CNRS, Univ Évry)**, sous la direction de **Nicolas BRUNEL**, la co-direction de **Juhyun PARK** et la co-supervision de **Rémi BRUN**.

Thèse soutenue à Paris-Saclay, le 16 Mai 2024, par

Perrine CHASSAT

Composition du jury

Membres du jury avec voix délibérative

Frédéric FERRATY Professeur, Université Toulouse Jean Jaurès - IMT	Président
Silvère BONNABEL Professeur, Mines Paris - PSL	Rapporteur & Examinateur
Anuj SRIVASTAVA Professeur, Florida State University	Rapporteur & Examinateur
Ian DRYDEN Professeur, University of South Carolina	Examinateur
Michèle GOUIFFES Maître de Conférences, Université Paris-Saclay - LISN - CNRS	Examinatrice
Alice LE BRIGANT Maître de Conférences, Université Paris 1 - SAMM	Examinatrice

THESE DE DOCTORAT
NNT : 2024UPASM005


université
évry
val-d'essonne
université
PARIS-SACLAY


Laboratoire de
Mathématiques
et Modélisation
d'Évry
LaMME


Fondation mathématique
FMJH
Jacques Hadamard


MATH
INNOV


îledeFrance


MOCAPLAB

Remerciements

Je souhaite débuter cette thèse en remerciant les personnes qui m'ont aidé, accompagné et soutenu tout au long de ces années de doctorat.

First of all, I would like to thank Silvère Bonnabel and Anuj Srivastava for having reviewed this manuscript, as well as Ian Dryden, Frédéric Ferraty, Michèle Gouffiès, and Alice Le Brigant for being part of my defense jury.

Je suis profondément reconnaissante envers mes deux encadrants, Nicolas Brunel et Rémi Brun, pour avoir créé la collaboration dont ma thèse est née, pour avoir inventé ce beau sujet et pour m'avoir choisie pour en relever les défis. Je remercie Nicolas pour ses idées, ses conseils, ses encouragements et son enthousiasme tout au long de ces années. Je remercie Rémi de m'avoir accueillie dans les locaux de Mocaplab pendant cette thèse, à une époque où le Covid-19 en avait rendu d'autres bien déserts, et pour m'avoir partagé et communiqué sa passion pour le mouvement !
Je tiens à remercier particulièrement mon encadrante Juhyun Park, pour sa présence et son encadrement régulier qui ont été d'une grande importance pour moi pendant ces années. Je la remercie également pour sa confiance et sa bienveillance, pour avoir toujours abordé nos échanges comme une collaboration et pour tout ce que j'ai appris à ses côtés.
Je voudrais aussi remercier chaleureusement Boris Dauriac pour son aide précieuse sur les données, et surtout pour sa présence et son soutien moral tout au long de cette thèse.

Je remercie toute l'équipe de Mocaplab pour ces belles données sur lesquelles j'ai pu travailler et pour m'avoir tous très bien accueillie dans les locaux.
Je remercie également mes collègues du LaMME et de l'ENSIIE, dont les doctorants avec qui j'ai eu la chance de partager cette expérience particulière qu'est la thèse. Merci à Salim pour ces conseils avisés en matière de choix d'ordinateur et pour ces grandes discussions dans le bureau, et merci à Claire pour son soutien et pour avoir su dynamiser le labo !

Je remercie grandement mes amis de m'avoir accompagnée pendant ces années de thèse, d'avoir toujours très bien su me changer les idées et me faire déconnecter.
Je remercie ma famille. Merci à mes deux soeurs, Lucile et Adèle, pour leur présence, leur soutien et pour avoir été des piliers durant ces années trop mouvementées. Un grand merci à ma mère pour son soutien et son amour, et dont l'influence et la carrière de chercheuse en mathématiques ont été une source d'inspiration, me montrant que cette discipline peut aussi être féminine et élégante. Sans son influence, je n'aurais, j'en suis certaine, jamais entrepris une thèse dans ce domaine.
Enfin, merci à Romain pour son immense soutien depuis le début, sa patience et son calme face aux aléas de ces années, qui m'ont donné confiance au cours de cette aventure.

Résumé

Cette thèse, réalisée dans le cadre d'une collaboration avec MocapLab, une entreprise spécialisée en motion capture, vise à déterminer le cadre mathématique le plus adapté et des descripteurs pertinents pour l'analyse des trajectoires de mouvement en langue des signes. En nous appuyant sur les principes du contrôle moteur, nous avons identifié le cadre défini par les formules de Frenet-Serret, incluant les paramètres de courbure, torsion et vitesse, comme particulièrement pertinent pour cette tâche. Ainsi, en introduisant de nouvelles approches d'analyse de courbes basées sur le cadre de Frenet, cette thèse contribue au développement de nouvelles méthodes dans les domaines de l'analyse de données fonctionnelles et de l'analyse de forme. La première partie de ce travail aborde le défi de l'estimation lisse des paramètres de courbures de Frenet, en traitant le problème comme une estimation de paramètres d'une équation différentielle dans $SO(d), (d \geq 1)$. Nous introduisons un algorithme Expectation-Maximization fonctionnel qui définit une méthode d'estimation unifiée des variables dans le groupe $SE(3)$, fournissant des estimateurs lisses, plus fiables et robustes que les méthodes existantes. Dans la deuxième partie, deux nouvelles représentations des courbes sont introduites : les courbures de Frenet non paramétrisées et la Square Root Curvatures (SRC) transform, établissant de nouveaux cadres géométriques riemanniens pour les courbes lisses dans $\mathbb{R}^d, (d \geq 1)$. En utilisant les informations géométriques d'ordre supérieur et dépendant de la paramétrisation, la Square Root Curvatures transform surpasse la représentation state-of-the-art Square-Root Velocity Function (SRVF) sur des résultats synthétiques. Étant donné une collection de courbes, ce type de géométrie nous permet de définir des critères statistiques efficaces pour estimer les formes moyennes de Karcher sur les espaces de formes riemanniens associés, qui se révèlent particulièrement performants sur des données bruitées. Enfin, ce cadre développé ouvre la voie à des applications plus pratiques dans le traitement de la langue des signes, comprenant l'étude des lois puissances sur nos données et le développement d'un modèle génératif pour le mouvement d'un point en langue des signes.

Mots clés : Analyse de Données Fonctionnelles, Analyse de Forme, Cadre de Frenet-Serret, Analyse de Trajectoires de Mouvement, Langues des Signes, Groupes de Lie

Abstract

This thesis, conducted in collaboration with MocapLab, a company specializing in motion capture, aims to determine the optimal mathematical framework and relevant descriptors for analyzing sign language motion trajectories. Drawing on principles of motor control, we identified the framework defined by the Frenet-Serret formulas, including curvature, torsion, and velocity parameters, as particularly suitable for this task. By introducing new curve analysis approaches based on the Frenet framework, this thesis contributes to developing novel methods in functional data analysis and shape analysis. The first part of this thesis addresses the challenge of smoothly estimating Frenet curvature parameters, treating the problem as parameter estimation of differential equation in $SO(d), (d \geq 1)$. We introduce a functional Expectation-Maximization algorithm that defines a unified variable estimation method in the $SE(3)$ group, providing smoother estimators that are more reliable and robust than existing methods. In the second part, two new curve representations are introduced: unparametrized Frenet curvatures and the Square Root Curvatures (SRC) transform, establishing new Riemannian geometric frameworks for smooth curves in $\mathbb{R}^d, (d \geq 1)$. Leveraging higher-order geometric information and parametrization dependence, the Square Root Curvatures transform outperforms the state-of-the-art Square-Root Velocity Function (SRVF) representation on synthetic results. Given a collection of curves, this type of geometry allows us to define efficient statistical criteria for estimating Karcher mean shapes on the associated Riemannian shape spaces, proving particularly effective on noisy data. Finally, this developed framework opens the door to more practical applications in sign language processing, including the study of power laws on our data and the development of a generative model for a point motion in sign language.

Keywords : Functional Data Analysis, Shape Analysis, Frenet-Serret Framework, Analysis of Motion Trajectories, Sign Language, Lie groups

Contents

1

Introduction

Contents

MocapLab (https://www.mocaplab.com/fr), a Parisian company specializing in motion capture, stands as one of the leading European entities in this field. Their diverse portfolio encompasses various research activities, particularly focusing on sign language explored through several projects (SignEveil, StorySign, Rosetta LIMSI and LISN (2022)). The data generated by MocapLab can be considered a collection of multivariate Euclidean curves. This type of data, corresponding to the recording of behavior or phenomena over time or space, can be seen as observations of a function. It has become increasingly common in recent years across various domains (aircraft trajectories, movement studies, biology, continuous temperature monitoring, etc.), notably due to the progress made in continuous data acquisition technologies (sensors, embedded devices, motion capture systems, etc.). The analysis of such complex data requires sophisticated and specialized mathematical tools, given their inherent high dimensionality and continuous nature. These challenges have motivated research in *functional data analysis* (Ramsay and Silverman, 2005), aiming to establish a dedicated framework for data modeled as functions, and *shape analysis* (Srivastava and Klassen, 2016), which seeks to develop statistical tools able to capture the complex shapes inherent in this type of data, using appropriate mathematical spaces. MocapLab's commitment to advancing sign language research, combined with the expertise of the Mathematics and Modeling Laboratory of Evry (LaMME), has resulted in their collaboration through this present thesis, generously supported by grants from Région Ile-de-France as part of the "Paris Region PhD 2020" project. The thesis aims to contribute significantly to the evolution of the functional data analysis and shape analysis landscape within a context motivated by sign language motion analysis.

1 Sign Language Context

Sign Languages (SLs) constitute the primary means of communication for approximately 70 million deaf individuals worldwide, according to World Federation of the Deaf (https://wfdeaf.org/our-work/). These languages have more than 200 variations globally, depending on the country. In France, French Sign Language (LSF) is predominant and considered as the first language for many deaf individuals. Historically, sign languages faced challenges in gaining recognition as independent natural languages. During antiquity, the Aristotelian notion that those who cannot speak cannot think hindered the education of deaf individuals. It was not until the 16th century that educators began working with deaf individuals, and in 1760, the Abbé de L'Épée established the first school for the deaf in Paris. The 1880 International Congress on Deaf Education marked a turning point, favoring oral methods over sign language in education. This ban on sign languages persisted until the end of the 20th century. The groundbreaking work of Stokoe (1960) led to a shift in perception and stimulated further research on sign languages. This led France to recognize LSF as a full-fledged language in the 2005 French law on disability. Subsequently, linguistic research, university programs, and professional training for sign language-related professions have proliferated, emphasizing the importance of accessibility in public spaces for individuals with disabilities. Today, despite progress, outdated beliefs persist, contributing to linguistic neglect and impacting the cognitive, socio-emotional, and academic development of deaf individuals.

1.1 Sign Language Linguistics

As with spoken languages, sign languages are governed by a linguistic system encompassing syntactic, morphological, semantic and phonological structures (Emmorey, 2001). SLs are oral languages as involving face-to-face communication: a signer conveys a message in SL, and an observer perceives this message (Figure B.1). SLs use the visual-gestural modality, involving continuous movements of the face, hands, signer's body, and surrounding space to construct a discourse. In addition, signers incorporate iconicity, characterized by a strong resemblance between the form of signs and their meanings (Sallandre and Cuxac, 2002). This enables the expression of complex ideas without resorting to a standard lexicon.

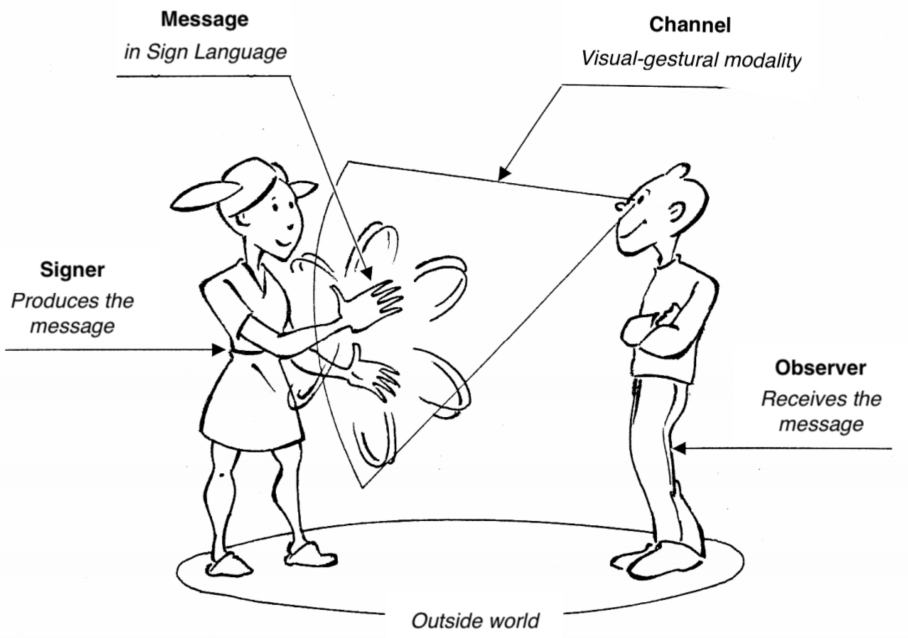


Figure 1.1: Sign Language schematic (illustration by Laurent Verlaine, Guitteny and Verlaine (2018)).

Recently, Yin et al. (2021) has proposed a detailed description of the linguistic features of SL. Among these features, *simultaneity* allows transmitting as much information as spoken language within a similar timeframe, using multiple visual elements to convey different information simultaneously (Sandler, 2012). While manual components are the primary source of emitted information, facial expressions are crucial in disambiguating certain signs and/or providing qualifications (e.g., big, small). Another distinctive aspect of SL involves placing discourse elements in specific locations in the signing space. To recall these elements later, signers designate their position with a gaze or a pointing gesture, a process known as *referencing*, which may induce a modification in sign movement due to changes in direction (e.g., directional verbs) (Dudis, 2004). Furthermore, signers use *fingerspelling* to indicate names, places, or new concepts from spoken language that lack a dedicated sign (Wilcox, 1992). This involves a set of manual gestures corresponding to written spelling. Finally, the model proposed by Brentari (1998) breaks down the articulatory properties of SL into four main phonemic classes: handshape, location, movement, and palm orientation. Non-manual features, such as eye gaze, head movement, and torso positioning, can also be included (Johnson and Liddell, 2011). However, there is still no consensus among linguists regarding the description of SLs, and even the concept of signs can vary according to linguistic theories.

1.2 Current Challenges in Sign Language Processing

The late recognition of sign languages as official languages, coupled with the complexity of SL linguistics and significant conceptual differences from spoken languages, has led to a considerable lag in understanding underlying mechanisms and processing methods in SL. This lag is evident compared to spoken and written language knowledge and processing capabilities. Today, most new communication or processing tools are designed for spoken languages. Significant progress, with considerable impact, have been made in these areas, such as speech recognition and voice assistants (*Siri*, *Google Assistant*), automatic translation between languages (*DeepL*, *Google Translate*), and more recently, in the automatic generation of text (*ChatGPT*, *Bard*). However, despite the recent development of promising applications for SL, like *Signily*, *Spread Signs*, *SignEveil* (by MocapLab), and others, substantial additional efforts are needed to develop tools for SLs comparable to those available today for spoken languages. There is a real and urgent social challenge in the development of automatic processing tools for SLs to reduce the communication barriers faced by signers and the discrimination caused by this delay in technological advancements. Research on SLs is relatively recent and sparse, primarily focusing on linguistics. However, innovation in SL technologies requires the involvement of other fields, such as computer science, statistics, and mathematics.

Today, many challenges still hinder progress in the automatic processing of SL. One primary challenge lies in representation methods. The absence of a widely adopted written form for SLs leads to the use of various modalities to address the sign language processing issue. Most research works in SL, particularly those based on deep learning methods, use video data of signers (Rastgoo et al., 2021). While being the most straightforward representation method, its high dimensionality makes storing, transmitting, and encoding costly. Moreover, anonymizing videos (where the signer's face appears) remains an open issue (Isard, 2020). Other research studies aim to develop written notation systems representing signs as discrete visual features linearly or in two dimensions. Universal notation systems like SignWriting (Sutton, 2014) and HamNoSys (Prillwitz and Zienert, 1990) exist, but none have gained widespread adoption in the sign language community. Additionally, among the main phonetic classes of sign language (handshape, location, movement, and palm orientation), representing the movement component in writing remains an open problem. Another representation, often referred to as *poses*, consists of a skeleton, mesh, or set of points in space representing the signer's articulation locations. These positions over time can be acquired with a motion capture system, typically expensive, or estimated directly from videos (Cao et al., 2019; Wang et al., 2023) with reduced quality and a potential loss of information. The pose representation is generally more reliable and natural than videos, enabling a more precise movement

analysis. However, it involves a continuous and multidimensional format that is non-trivial to analyze and not well-suited for language processing algorithms such as NLP. Moreover, even with this representation, the question of data anonymization arises. Like spoken languages, where voice characteristics can reveal the speaker's identity, a signer's identity might be discernible through their movements, but little is known about the motion features that characterize him and how they can be manipulated (Bigand, 2022).

The automatic processing of SLs involves developing methods for typical machine learning tasks: segmentation, classification, recognition, translation, and generation (Bragg et al., 2019; Yin et al., 2021). Segmentation in sign language involves detecting meaningful units or identifying sub-entities composing the signal (similar to phonemes in spoken languages) (Santemiz et al., 2009; Farag and Brock, 2019; Bull et al., 2020). Recognition involves associating a label with a sign and differs from translation, which refers to the transition from sign to spoken language. With recent progress in deep learning, numerous research efforts have been conducted for these tasks, primarily using video data (Camgöz et al., 2020; Wadhawan and Kumar, 2020). For sign language generation, on the other hand, most works use poses as representations. Here, the goal is to automatically animate avatars, which can serve as equivalents to voice assistants for SLs. In existing methods for this task, some generate and concatenate isolated signs (Stoll et al., 2020) using neural machine translation and generative adversarial networks, while others (Saunders et al., 2020b,a; Xiao et al., 2020) autonomously decode a sequence using transformers or recurrent neural networks. However, evaluating the quality of these generative models remains challenging. Moreover, one main limitation of further development of more efficient methods is the lack of available data across all the different sign language representations. Rastgoo et al. (2021) present a comprehensive list of existing datasets for SL research, with most designed for sign classification. For translation purposes, available continuous sign corpora contain significantly fewer sentence pairs than similar spoken language datasets (Arivazhagan et al., 2019). Furthermore, many SL datasets discussed in the literature are either not accessible or subject to strict restrictions and licenses, primarily due to the ongoing challenge of anonymizing this type of data. Therefore, developing processing methods that can operate effectively with limited data is particularly interesting in this domain. Finally, a common limitation in developing models for each of these tasks, highlighted by Yin et al. (2021), is the lack of an effective and standardized tokenization method for sign language. It is, therefore, crucial, first and foremost, to gain a better understanding of the properties and complex structure of sign language.

2 Analysis of Sign Language Movement Trajectories

At the core of this thesis, our exploration focuses on the analysis of movement within the four phonemic classes in SL (hand shape, position, movement, and palm orientation). Our central objective is to enable a thorough analysis of what could be likened to a "movement signal" in sign language by analogy with the auditory signal of spoken languages. We seek to define a relevant mathematical framework to identify primitives, descriptors, and other "kinetic features" of movement in SL, with a particular emphasis on hand movements, which constitute the primary source of information in SL, although not exhaustive. The overall goal of our research is to unveil the underlying mechanisms of human movement in SL, thereby enhancing our current understanding of this gestural language. This approach is crucial in the context of SL processing. For example, in the creation of authentic avatar animations, given that humans are particularly sensitive to biological movement and can naturally distinguish between human and synthesized movement. Additionally, in the context of the written transcription of SL, it is imperative to decompose the movement component into primitives or descriptive elements. Hence, the central role of this thesis is to develop mathematical tools and an analysis framework specifically tailored to this task, contributing to filling current gaps in the understanding of sign language movement. In this regard, we are fortunate to work with data acquired through motion capture, which is particularly well-suited to our objective among the various types of SL representation mentioned earlier.

2.1 Motion Capture Data

Motion capture, or mocap, is a revolutionary technology for recording and reproducing human movements or other objects using various sensors and/or camera systems. For human movements, mocap involves tracking trajectories of key points on the moving body over time, directly translating these movements into an interpretive digital representation (temporal vectors in three dimensions). For over a century, mocap has evolved from techniques based on images such as chronophotography or rotoscoping to now focus on advanced approaches like optical, mechanical, and inertial systems.

Figure 1.2: MocapLab's motion capture system and field for sign language motion acquisition.

The mocap system used by MocapLab is a passive optical system with 50 Vicon cameras in a volume of 20 m^3 to 60 m^3 (Figure B.2). Marker-based optical motion capture systems (Vicon, Inc, Optitrack, Inc) are considered the gold standard for human motion analysis. They use retroreflective markers placed on objects or the moving body. Infrared cameras then record the position of these markers by reflecting the infrared light emitted by them. Triangulation between the cameras subsequently reconstructs three-dimensional trajectories with extremely high precision - theoretically with sub-millimeter precision using MocapLab's system. Due to this high precision, this type of mocap system is a powerful tool for creating realistic virtual content and analyzing human movement. In particular, this has allowed MocapLab to engage in projects across various domains, including avatar animation for the film industry and video games (see https://www.mocaplab.com/fr/gallerie), research in biomechanics and medicine, and sign language linguistics (LIMSI and LISN, 2022; Gibet et al., 2015). They are notably capable of simultaneously measuring the body, face, gaze direction, and fingers, all while minimally impeding the signer's gestures.

The primary goal of mocap is to capture movements precisely to reproduce them realistically in virtual or study contexts (Figure B.3). However, while passive optical systems offer exceptional spatial precision, they also pose technical challenges and limitations. Issues such as occlusions, light interference, and marker confusion can lead to missing or inaccurate data, often requiring a controlled environment with a specific camera setup to ensure optimal performance. Moreover, the purchase and use of mocap systems are expensive, limiting the possibility of obtaining a large amount of data for applications such as deep learning model training. Although there are less costly markerless alternatives, such as Microsoft Kinect, which uses depth sensors to track movements without physical markers, their precision may not always be sufficient for many applications. Thus, this type of data is particularly suitable for the detailed study of human movement but requires analysis methods that do not necessitate a large amount of data.

Figure 1.3: The signer in motion capture and her avatar (wrist's drawn trajectory highlighted in yellow).

2.2 Background on Human Movement Trajectories Analysis

Sign language movements are governed not only by linguistic rules but also by principles inherent to human motion (Benchiheub, 2018). It has been demonstrated that human motion is subject to biomechanical constraints and motor control laws. The study of motor control is a complex problem addressed by various disciplines such as psychology, cognitive science, biomechanics, and neuroscience, and more recently extended by the application of machine learning techniques to motion data, providing additional perspectives for analyzing human motion characteristics. A significant portion of research in human motion analysis focuses on specific movements, where the goal is uniquely defined by the motion of a well-identified endpoint, referred to as the "end-effector" movement (Polyakov et al., 2009; Carreno-Medrano et al., 2015). These studies explore these movements' geometric and kinematic characteristics and the constraints governing them. In the context of the sign language movement, the movement of the wrist or hand can be considered an end-effector movement (Endres et al., 2013).

In principle, the human musculoskeletal system allows numerous potential trajectories for a hand movement towards a goal. Researchers in motor control hypothesize that the human system has developed optimal control strategies to select between these possibilities through evolution, learning, and adaptation. Thus, human motion is assumed to be governed by internal and external constraints ensuring the movement's success. Then, various optimality principles have been proposed in the literature (Wochner et al. (2020), Oguz et al. (2018)). One of the early principles highlighted by Flash and Hogan (1985) is the jerk minimization principle. The jerk represents the third derivative of position with respect to time, i.e., the variation of acceleration. Minimizing jerk aims to reduce abrupt changes in acceleration, making the movement smoother and more energy-efficient. This principle has led to other motor control principles, such as the isochrony principle and the two-thirds power law (Viviani and Flash, 1995). Viviani and McCollum (1983) demonstrated the isochrony principle, showing that the speed of drawing movements increased proportionally to the trajectory distance, maintaining the execution time of these complex trajectories independent of movement size. This linear relationship between speed and trajectory extent has been demonstrated for various actions such as writing or hand and arm movements (Freund and Btidingen, 1978). The kinematic analysis of circular movements revealed another well-known law of motion, the two-thirds power law (also known as the one-third power law), suggesting a nonlinear correlation between movement speed and curvature. Originally observed for planar movements, another version of this law for space movements also involves the torsion parameter of the movement trajectory. The curvature and torsion of a curve are parameters that fully describe its geometry. These power laws are formulated as follows:

$$\dot{s}(t) = C_1 \kappa(s(t))^{-1/3} \qquad \text{(1/3 power law)}, \tag{1.1}$$

$$\dot{s}(t) = C_2 \left(\kappa(s(t))^2 |\tau(s(t))|\right)^{-1/6} \qquad \text{(1/6 power law)}, \tag{1.2}$$

where C_1 and C_2 are constants, $\dot{s}$ is the curvilinear speed and κ and τ the curvature and torsion of the motion trajectory. They assumed that speed increases in less curved portions of the trajectory and decreases inversely in more curved portions. Endres et al. (2013) uses the two-thirds power law to segment sign language movements. Finally, other kinematic theories of movements are found in the literature, such as the Sigma-Lognormal model, suggesting that complex human movements consist of overlapping strokes, each with a lognormal-shaped velocity profile. This theory has been particularly demonstrated for movements in the signature plane (Ferrer et al., 2020) and extended to three dimensions by Schindler et al. (2018).

These laws and principles can serve as starting points for the mathematical and statistical analysis of movement trajectories. Ramsay et al. (1995) uses the Sigma-Lognormal model to propose a functional data analysis of human finger pinch force data. In a similar functional data analysis framework, Raket et al. (2016) conducted a biomedical experiment on hand movements. Many other works in this framework do not only study end-effector movement but the entire skeleton movement, mostly using data acquired by motion capture. Among many others, some seek to align movements temporally (Olsen et al., 2018; Raket et al., 2016), others to model the spatial and temporal variability of a set of movements (Park et al., 2022), or to analyze their shapes (Celledoni et al., 2016). In most of these works, movement trajectories are mathematically represented by their three Cartesian coordinates. However, this representation is known not to be suitable for the analysis of trajectory shapes, as it depends on all rigid transformations (translations, rotations, scaling, parameterization). Consequently, others use a more sophisticated representation involving the derivative of the movement trajectory, called the square-root velocity function (SRVF) (Srivastava et al., 2011), directly on Euclidean curves (Devanne et al., 2015; Park et al., 2022) or on curves with values in rotations that appear in character animation (Celledoni et al., 2016). Finally, more recently, Celledoni et al. (2019) and Yang et al. (2022) proposed using signatures (Fermanian, 2021) to describe human movements.

3 Approach and Contributions

This thesis aims to propose a mathematical framework tailored to analyze sign language motion trajectories. We restrict the analysis to the movement of a point corresponding to the right wrist. This choice is justified by the fact that a significant part of sign language communication relies on wrist movements and by the decomposition into four phonemic classes proposed by Brentari (1998), which isolates the movement component from the others. Therefore, our analysis falls within the scope of end-effector movement. Drawing from the previously discussed principles of motor control, we can identify descriptors that appear particularly pertinent for the analysis of human movement. Among these, we are interested in the motion trajectory's speed, curvature, and torsion. The curvature and torsion parameters in a three-dimensional space are the components of the Frenet-Serret formulas, initially defined by two French mathematicians Jean Frédéric Frenet (1852) and Joseph Alfred Serret (1851). The framework defined by the Frenet-Serret formulas is particularly powerful for describing the geometry of a Euclidean curve. This inspired our approach to the problem but raises two main challenging questions:

- How can we derive reliable estimates of curvature and torsion parameters from discrete and potentially noisy numerical data corresponding to wrist motion capture?
- How can we comprehensively understand and analyze the variability inherent in a set of motion trajectories based on these parameters?

These questions extend beyond their application in sign language, finding relevance in any collection of Euclidean curves. They embody classical challenges in Euclidean curve shape analysis, where the initial obstacle lies in identifying a suitable curve representation.

3.1 Datasets

Throughout the thesis, we worked with two main datasets produced by MocapLab. The first dataset is from the Rosetta project (LIMSI and LISN, 2022), which includes the translation in LSF of 194 newspaper headlines, descriptions of 28 photographs, and 1200 isolated signs corresponding to the most common notions in newspaper headlines. All these movements were performed by the same deaf signer. In this dataset, we isolated 143 well-segmented signs repeated at least 2 times, but only 14 have at least 4 repetitions. The second dataset was captured by MocapLab as part of this thesis. Using LSF dictionaries LSFPlus, Inc and LeDicoElix, Inc, we selected 60 mono-manual (i.e., involving the movement of a single hand) signs in LSF. Two signers, a deaf woman (Aliza), and a hearing man (Thomas), performed approximately 5 repetitions of each sign, returning to an initial position between each repetition, resulting in a set of 648 movements. Both datasets include the movements of several key points on the signer's body, but we consider only the movement of the right wrist in this thesis.

3.2 Contributions

We now list the contributions of this thesis to the aforementioned problems, followed by the description of the organization of the manuscript. In particular, Chapter 2 is not included in these contributions as it is a background chapter on mathematical concepts used throughout the manuscript that includes introductions to differential and Riemannian geometry, matrix Lie groups, functional and shape data analysis, and the Frenet-Serret framework.

Part I focuses on estimating Frenet curvatures.

- Chapter 3 gathers existing methods for estimating Frenet curvatures of a Euclidean curve from its discrete and noisy numerical observations. These methods involve estimating latent variables corresponding to either the derivatives of the curve or the Frenet path in a preprocessing step. They include using the extrinsic formulas of Frenet curvatures and the methods proposed by Park and Brunel (2019), treating Frenet curvature estimation as a parameter estimation problem of a differential equation. Our distinctive contribution lies in introducing an alternative approach for smooth estimation of the Frenet path based on a tracking algorithm. This method proves more computationally efficient and yields results comparable to existing methods. As a contribution, we also provide full theoretical descriptions of these different methods and a comprehensive comparison of them through studies on simulated data.

- Chapter 4 tackles the challenge posed by existing methods for Frenet curvatures estimation, all dependent on estimating latent variables in a distinct preprocessing stage. We introduce a new method that addresses the estimation of latent variables and parameters in a unified manner by formulating the problem within the $SE(3)$ group. This method involves the development of a functional Expectation-Maximization algorithm, where the expectation step uses an invariant extended Kalman filter on the Lie group $SE(3)$, and parameter estimation is formulated within a functional data analysis framework using penalized expected log-likelihood maximization. The improved accuracy of the estimators obtained with this approach is demonstrated through experiments on synthetic data and by applying the different methods to wrist movement data in sign language.

Part II is devoted to developing a shape analysis method based on the Frenet-Serret framework.

- Chapter 5 establishes a Riemannian geometry framework for smooth curves in $\mathbb{R}^d, d > 1$, based on Frenet curvatures. We introduce two new representations of a smooth Euclidean curve: the unparametrized Frenet Curvatures (FC) directly and the Square Root Curvature Transform (SRC), inspired by the square root velocity transform of the $SO(d)$-valued Frenet path. Both representations consider all geometric features of the shape, providing likely geodesics that avoid artifacts

encountered by representations using only first-order geometric information, such as the state-of-the-art Square-Root Velocity Function (SRVF). The Square Root Curvature Transform combines the strengths of the other two, depending on parametrization like the SRVF and therefore involving a registration step necessary to obtain consistent geodesics in certain cases. Our analysis, supported by simulated data, demonstrates its relevance to analyzing human motions. This chapter is published as a paper Chassat et al. (2023) for the ICCV conference.

- Chapter 6 addresses the challenge of defining a mean shape of a set of Euclidean curves in $\mathbb{R}^d, d > 1$. The classical approach to shape analysis considers the Karcher mean in shape spaces defined by the SRVF, SRC, or FC representations. We compare two distinct approaches for the practical estimation of Karcher means based on Frenet curvatures, considering their sensitivity to observational noise. The first relies on a robust preprocessing step involving efficient FDA-based estimation of each Frenet curvatures individually. As an alternative, we introduce a regularization-based approach inspired by the work of Park and Brunel (2019). In a functional data modeling framework, they explicitly formulate the problem by providing a statistical characterization of the mean Frenet curvature parameter through the concept of the mean vector field. Bridging their work with shape analysis, we demonstrate that their proposed mean parameter estimation regularized criterion approximates the FC Karcher mean. Leveraging their idea, we define two alternative criteria involving a step of alignment of individual curvatures. The strengths of estimated means obtained through a regularization-based approach are demonstrated on noisy simulated data. In the case of less noisy data, such as our sign language movement trajectories data, the two approaches show marginal differences, but methods based on Frenet curvatures and considering parametrization appear more effective. Some of this work has been submitted through the paper Park et al. (2022) as a first revision of the paper Park and Brunel (2019) but still requires further review.

- Chapter 7 stands as an independent chapter demonstrating more practical applications of the developed tools and framework for motion trajectory analysis in sign language processing based on the Frenet framework. Using our robust estimators for curvature and torsion of motion trajectories, we assess the validity of power laws on wrist movement data in LSF. Additionally, we introduce a generative model for wrist movement in sign language, based on the decomposition of the variability of a set of trajectories through its mean geometry and non-linear warping functions derived from the Frenet curvatures parameter. Our discussion extends to future research directions in sign language processing within the established framework, particularly focusing on motion trajectory segmentation and clustering in sign language.

The implementation of all the methods defined in this thesis has contributed to the development of the Python package `FrenetFDA` available on GitHub (https://github.com/perrinechassat/FrenetFDA) and detailed in the appendix, which gathers all the necessary tools for performing curve analysis in dimension less than or equal to 3 based on the Frenet framework.

3.3 Publications

These contributions have resulted in the following peer-reviewed publications and pre-prints. We are also writing other papers, which we will submit soon.

Conferences Articles

- **Chassat Perrine**, Juhyun Park, Nicolas Brunel. "*Shape Analysis of Euclidean Curves under Frenet-Serret Framework*". International Conference on Computer Vision (ICCV), Paris, 2023 (accepted for oral presentation), https://openaccess.thecvf.com/content/ICCV2023/papers/.

- **Chassat Perrine**, Juhyun Park, Nicolas Brunel. "*Analysis of variability in sign language hand trajectories: development of generative model*". Proceedings of the 8th International Conference on Movement and Computing (MOCO '22), Chicago, 2022.
  DOI: https://doi.org/10.1145/3537972.3537999

Article Under Review

- Park Juhyun, Nicolas Brunel, **Perrine Chassat**. "*Curvature and Torsion estimation of 3D functional data: A geometric approach to build the mean shape under the Frenet Serret framework*". Preprint arXiv:2203.02398, 2021.

Articles in Preparation

- **Chassat Perrine**, Juhyun Park, Nicolas Brunel. "*Functional Expectation-Maximization Algorithm on SE(3) for Frenet Curvatures Estimation*". (*to come*).

2

Literature Review and Background Material

Abstract

In this chapter, we present the mathematical concepts essential for understanding the methods developed in the thesis. This includes an introduction to differential and Riemannian geometry. We particularly delve into properties related to Riemannian manifolds and matrix Lie groups. We review existing methods for analyzing the type of data of interest, namely Euclidean trajectories. This encompasses techniques from functional and shape data analysis. Finally, this chapter introduces the Frenet-Serret framework, forming the basis for the methods developed in the subsequent chapters.

Contents

1 Differential and Riemannian Geometry

The motion trajectories we consider in this thesis are curves in the Euclidean space $\mathbb{R}^3$. To compare their shapes, define distances between them, and describe their geometries, we will see that it is essential to represent them mathematically in more complex spaces that cannot adequately be described using traditional Euclidean geometry, which deals with flat and straight-line geometries. Then, we introduce

differential geometry, a branch of mathematics that focuses on studying smooth, curved, and higher-dimensional spaces. Differential geometry deals with the properties of these spaces (metrics, distance, etc.) and provides a mathematical framework for understanding and describing curved geometries. Moreover, these spaces describe certain invariances that need to be considered when analyzing the shapes of these trajectories. We present the main spaces, their main structures, and properties that will be useful in the following. We refer to Boothby (2003); Lang (2006); Tu (2008); Sommer et al. (2020) for a deeper understanding of these concepts.

1.1 Manifolds, Tangent Spaces, and Submanifolds

A *manifold* is a mathematical structure that allows one to study and understand spaces with complex and varying geometries by approximating them with simpler, locally Euclidean structures. Essentially, a m-dimensional manifold is a topological space M which is locally homeomorphic to $\mathbb{R}^m$. This means that for each point $p \in M$, there exists an open neighborhood $U \subset M$ of p and a homeomorphism $\phi : U \to \phi(U)$ such that $\phi(U)$ is an open set of $\mathbb{R}^m$. The core idea behind this locally Euclidean property of M is to perform calculations in a more convenient Euclidean space through a *coordinate chart* (U, ϕ).

Then, one can define a *differentiable manifold*, which is equipped with a smooth structure defined from charts and collections of charts referred to as atlases. This structure allows the definition of smooth or differentiable functions on the differentiable manifold M and the definition of tangent spaces representing the behavior of functions near points on M. For that, we consider a chart on M, $\phi : U \to \mathbb{R}^m$, where U is an open subset of M containing a point p. Let $\gamma_1 :]-\epsilon_1, \epsilon_1[\to M$ and $\gamma_2 :]-\epsilon_2, \epsilon_2[\to M$ be two $\mathcal{C}^1$-curves passing through p, that is $\gamma_1(0) = \gamma_2(0) = p$. We define an equivalence relation on such curves; that is: the curves are called equivalent at p if and only if $(\phi \circ \gamma_1)'(0) = (\phi \circ \gamma_2)'(0)$. Then, for any $p \in M$, a *tangent vector* to M at p is any equivalence class of $\mathcal{C}^1$-curves through p at M modulo this equivalence relation. The space of tangent vectors at p is called the *tangent space* of M at p and is denoted T_pM. The tangent space T_pM is a vector space of dimension $dim(M)$. The set of all tangent spaces at every point on M constitutes the tangent bundle of M, denoted TM. For more theoretical and abstract mathematical definitions of these concepts, we refer to do Carmo (1992); Lang (2006); Sommer et al. (2020) and references therein.

In contrast to the abstract definition of a manifold, an alternative and more intuitive point of view, is to envisage manifolds as subsets, like surfaces or shapes, embedded in a larger manifold or in a larger Euclidean space $\mathbb{R}^n$, which is itself an n-dimensional differentiable manifold. This is referred to as *submanifolds*. In practice, many sets are often regarded as submanifolds of larger manifolds, even though they can be studied intrinsically as manifolds. This is partly due to the following theorem, which is often used to establish that a set is indeed a manifold. To state this theorem, we first define the concept of the differential of a map between two manifolds.

Definition 2.1 (Differential of a map between manifolds)**.** *Suppose M and N are manifolds and let $f : M \to N$ be a smooth mapping. The differential of f at $p \in M$ is the linear map $df_p : T_pM \to T_{f(p)}N$ such that for any $v \in T_pM$ and $g : N \to \mathbb{R}$ a smooth function:*

$$df_p(v)(g) = v(g \circ f)\,.$$

Theorem 2.1 (Srivastava and Klassen (2016))**.** *Suppose M and N are manifolds of dimensions m and n, respectively, and let $f : M \to N$ be a smooth map, and $y \in N$ a regular value of f (i.e. for any point $p \in f^{-1}(y)$, df_p is onto). Then, $f^{-1}(y)$ is a submanifold of M of dimension $m - n$. Furthermore, the tangent space of $f^{-1}(y)$ at a point p is given by the kernel of df_p.*

As we are particularly interested in function spaces in this thesis, we introduce *infinite-dimensional manifolds*, which are mathematical spaces that extend the concept of finite-dimensional manifolds to infinite-dimensional spaces. These are locally homeomorphic to an open subset of an infinite-dimensional

Banach or Hilbert space (rather than to $\mathbb{R}^n, n \geq 1$ as in finite dimension). Similarly to the finite-dimensional case, an infinite-dimensional manifold is commonly seen as a submanifold of a known Hilbert or Banach manifold. Then, a similar result as Theorem 2.1 can be stated for the general submanifolds supposing that M and N are two smooth Banach manifolds and with the additional assumption that f is transversal over $y \in N$, which means that the differential of f at every point of $f^{-1}(y)$ is onto (Lang, 2006). In addition, we note here that it is also possible to establish that a set is a manifold by demonstrating that it constitutes the Cartesian product of two differentiable manifolds.

1.2 Riemannian Structure on Manifolds

So far, the manifold M has only been equipped with a differentiable structure. However, to define distances between points on a manifold (shapes), we must introduce an additional structure called a *Riemannian structure*. These distances will then be determined by constructing the shortest paths between the shapes and measuring the length of these paths, using a *Riemannian metric* defined infinitesimally from the elements of the tangent space at a point.

Definition 2.2. *A **Riemannian metric** on a differential manifold M is a smooth inner product ϕ (symmetric, bilinear, positive definite form) on the tangent space T_pM of M. A differential manifold with a Riemannian metric is called a **Riemannian manifold**.*

Let $\alpha : [0,1] \to M$ represent a path on a Riemannian manifold M. Using the Riemannian metric ϕ, we can define the length of the path α. Assuming that α is differentiable everywhere on $[0,1]$ we consider the velocity vector at t, $\frac{d\alpha}{dt} \in T_{\alpha(t)}M$, and the length of α is defined as

$$L[\alpha] = \int_0^1 \sqrt{\phi\left(\frac{d\alpha}{dt}, \frac{d\alpha}{dt}\right)} dt\,. \tag{2.1}$$

Then, the distance between any two points $p, q \in M$, a Riemannian manifold, is the minimum length among the smooth paths on M joining these points. In particular, the path with the shortest length is called the *geodesic* between them, and its length defines a proper metric on M.

Recall that the tangent space at a point T_pM is a vector space that can be viewed as a linear approximation of the manifold M in a neighborhood of the point p. Using geodesics, we can then define a mapping, called the *exponential map*, which allows us to transfer points between M and T_pM and thus exploit this approximation. Let $v \in T_pM$ be a tangent vector to M at p. There is a unique geodesic $\alpha_v : [0,1] \to M$ satisfying $\alpha_v(0) = p$ with initial tangent vector $\dot{\alpha}_v(0) = v$. Then, the exponential map $\exp_p : U \subset T_pM \to M$ is defined by $\exp_p(v) = \alpha_v(1)$. Where the inverse is defined, locally around $0 \in T_pM$, we define locally the logarithm map $\log_p(q) : M \to T_pM$ as the inverse of the exponential map. Then, the logarithm $\log_p(q)$ is the smallest vector between p and q in M as measured by the Riemannian metric and thus we have

$$\operatorname{dist}(p,q) = \|\log_p(q)\|. \tag{2.2}$$

Note that in an Euclidean space, these maps are simply defined as $\exp_p(v) = p + v$ and $\log_p(q) = q - p$.

Finally, an important point for our application is defining a Riemannian structure and geodesics in infinite-dimensional spaces of curves on manifolds. We denote $\mathcal{M}$ the space of measurable functions from $I \to M$ where M is a manifold. The space $\mathcal{M}$ is also a manifold of infinite dimension. Using the Riemannian metric on M denoted ϕ_M, we can define a Riemannian metric on $\mathcal{M}$, $\phi_{\mathcal{M}}$ as follows: let $f \in \mathcal{M}$, for $w_1, w_2 \in T_f\mathcal{M}$

$$\phi_{\mathcal{M}}(w_1, w_2) = \int_I \phi_M(w_1(s), w_2(s))_{f(s)} ds, \tag{2.3}$$

where the tangent space $T_f\mathcal{M}$ is the set

$$T_f\mathcal{M} = \left\{ w : I \to TM \,|\, \forall t \in I, w(t) \in T_{f(t)}M \text{ and } \int_I \phi_M(w(t), w(t))dt < \infty \right\}.$$

Then, a map $\alpha : I \times [0,1] \to M$ is a geodesic on $\mathcal{M}$ if for all $t \in I$, $\alpha_t = \alpha(t, \cdot) : [0,1] \to M$ is a geodesic on M (see proof in Srivastava and Klassen (2016)).

1.3 Lie Groups

We now introduce *Lie groups*, which are fundamental entities in differential geometry as they merge manifold and group properties. In this thesis, we will use Lie groups from two distinct perspectives. First, we will regard matrix Lie groups as the ambient spaces for our objects. Secondly, we will consider Lie groups for their actions on manifolds, such as translations, rotations, affine transforms, etc.

Definition 2.3. *A **Lie group** G is a group and a smooth manifold such that the group operations, combination $G \times G \to G : (g,h) \mapsto gh$ and inversion operation $G \to G : g \mapsto g^{-1}$, are smooth mappings.*

Then, the group G satisfies under these operations the four group axioms: closure, associativity, identity, and invertibility. We refer to Stillwell (2008); Chirikjian (2009); Hall (2003); Moskowitz and Abbaspour (2007) for detailed introductions to Lie theory.

1.3.1 Matrix Lie Group and Lie Algebra

A *matrix Lie group* is a Lie group whose elements are matrices and whose combination and inversion operations are, respectively, matrix multiplication and matrix inversion. As a matrix Lie group G is also a manifold, we can define its tangent space at any point $X \in G$, representing the space of all possible tangent vectors at that point: T_XG. For $X, Y \in G$ and $A \in T_XG$, we define the left action of the matrix Lie group onto itself and its associated tangent mapping as

$$L_XY = XY, \qquad (L_X)_* A = XA\,. \tag{2.4}$$

The right action can be defined in a similar way. Then, this left (or right) -translation map defines a left (or right) -invariant Riemannian metric on any matrix Lie group G, between $A, B \in T_XG$, by

$$\langle A, B\rangle_X = \langle (L_{X^{-1}})_* A, (L_{X^{-1}})_* B\rangle_I = \langle X^{-1}A, X^{-1}B\rangle_I = \mathrm{Tr}\left((X^{-1}A)^T X^{-1}B\right), \tag{2.5}$$

and its associated norm is called the Frobenius norm and denoted $\|\cdot\|_F$. In particular, at the identity element of the Lie group, the tangent space is called the Lie algebra. The *Lie algebra*, denoted $\mathfrak{g}$, is a p-dimensional vector space over a specific field (here $\mathbb{R}$) that captures the infinitesimal behavior and algebraic properties of the Lie group. The Lie algebra can be defined by a basis of real matrices E_i for $i = 1, \ldots, p$. We introduce the Lie algebra isomorphisms,

$$(\cdot)^\vee : \mathfrak{g} \to \mathbb{R}^p \quad \textit{(“vee”)} \tag{2.6}$$

$$(\cdot)^\wedge : \mathbb{R}^p \to \mathfrak{g} \quad \textit{(“wedge”)}\,, \tag{2.7}$$

such that $E_i^\vee = \boldsymbol{e}_i$ where $\{\boldsymbol{e}_i\}_{i=1,\ldots,p}$ is the natural basis of $\mathbb{R}^p$. The Lie algebra is equipped with a binary operation called the *Lie bracket*, denoted as $[\cdot,\cdot]$, and defined as the map $A, B \in \mathfrak{g} \times \mathfrak{g} \mapsto [A,B] = AB - BA \in \mathfrak{g}$. The Lie bracket satisfies the following properties:

1. $\mathbb{R}$-bilinearity: for any elements $A, B \in \mathfrak{g}$, $[A,B] \in \mathfrak{g}$.
2. Anti-symmetry: for any elements $A, B \in \mathfrak{g}$, $[A,B] = -[B,A]$.
3. Jacobi identity: for any elements $A, B, C \in \mathfrak{g}$, $[A,[B,C]] + [C,[A,B]] + [B,[C,A]] = 0$.

The Lie bracket operation on the Lie algebra reflects the group multiplication in the Lie group. In fact, the Lie group and its Lie algebra are interconnected by some key relations, detailed below, which are essential for manipulating and working with Lie group objects.

Exponential and Logarithm map. The Lie algebra being geometrically the tangent space to the Lie group at the identity there are linked by the exponential map, which relates the tangent space to its corresponding manifold. The *exponential map* is a mapping from the Lie algebra to the Lie group: $\exp_G : \mathfrak{g} \to G$, which provides a way to "undo" the linearization. One can also define the inverse mapping, the *logarithm map* from the Lie group to the Lie algebra: $\log_G : G \to \mathfrak{g}$. Fortunately, if we consider a matrix Lie group, the exponential and logarithmic maps are simply the classical matrix exponential and matrix logarithm given by

$$X = \exp(A) = \sum_{n=0}^{\infty} \frac{1}{n!} A^n \,, \qquad A = \log(X) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} (X - I)^n \,. \tag{2.8}$$

Note that, in the logarithm expression, the series is convergent only if $\|X - I\| < 1$. Moreover, as in the general case of any manifold, the exponential and logarithmic mappings are usually only subjective and are locally defined around an open neighborhood of 0 in $\mathfrak{g}$ and an open neighborhood of the identity element I_d in G.

Adjoint representation. The *adjoint representation* of a Lie group is a mapping that represents a group element as a linear transformation on the group's Lie algebra. More specifically, the adjoint representation of G on $\mathfrak{g}$ is a mapping $Ad_G : G \to GL(\mathfrak{g})$ that assigns to each group element $X \in G$ a linear transformation $Ad_G(X)$ on the Lie algebra $\mathfrak{g}$, defined, using the notations (2.6) and (2.7), as

$$Ad_G(X)\boldsymbol{b} = \left(X(\boldsymbol{b})^\wedge X^{-1}\right)^\vee \,, \tag{2.9}$$

where $\boldsymbol{b}^\wedge$ is an element of the Lie algebra $\mathfrak{g}$, and thus $\boldsymbol{b} \in \mathbb{R}^p$. The differential of the adjoint representation Ad_G of the Lie group G on the Lie algebra $\mathfrak{g}$ matches the adjoint representation ad_G of the Lie algebra $\mathfrak{g}$ onto itself, also denoted by "$\cdot^\curlywedge$" and defined for $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^p$ as

$$ad_G(\boldsymbol{a}^\wedge)\boldsymbol{b} = \boldsymbol{a}^\curlywedge \boldsymbol{b} = [\boldsymbol{a}^\wedge, \boldsymbol{b}^\wedge]^\vee \,. \tag{2.10}$$

In the following, we will denote by "$\cdot^\curlyvee$" the operation that undoes the adjoint representation of the Lie algebra, "$\cdot^\curlywedge$". The adjoint representation has several properties related to commutation; for $X \in G$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^p$ we have,

$$X \exp(\boldsymbol{b}^\wedge) = \exp\left((Ad_G(X)\boldsymbol{b})^\wedge\right) X \,, \tag{2.11}$$

$$ad_G(\boldsymbol{a}^\wedge)\boldsymbol{b} = -ad_G(\boldsymbol{b}^\wedge)\boldsymbol{a} \,, \tag{2.12}$$

$$Ad_G\left(\exp(\boldsymbol{a}^\wedge)\right) = \exp(ad_G(\boldsymbol{a}^\wedge)) = \exp(\boldsymbol{a}^\curlywedge) \,. \tag{2.13}$$

Jacobian matrix. The *Jacobian matrix* of a Lie group G captures the exponential map's derivative at the Lie group's identity element. It serves as a fundamental tool for differentiating and integrating functions on Lie groups that are expressed in coordinates. It can be defined as a function of the adjoint representation as

$$J_G(\boldsymbol{a}) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)!} ad_G(\boldsymbol{a}^\wedge)^n \,. \tag{2.14}$$

Baker-Campbell-Hausdorff formula. The *Baker-Campbell-Hausdorff (BCH) formula* allows one to approximate the group product in terms of the Lie algebra operations.

Theorem 2.2 (Baker-Campbell-Hausdorff formula)**.** *If $\mathfrak{g}$ is a Lie algebra defined over a field of characteristic 0 and $A, B \in \mathfrak{g}$, then $\exp(A)\exp(B) = \exp(C)$ for some formal infinite sum C of elements in $\mathfrak{g}$. In particular,*

$$C = \log(\exp(A)\exp(B)) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} \sum_{r_i + s_i > 0} \frac{[A^{(r_1)}, B^{(s_1)}, \ldots, A^{(r_n)}, B^{(s_n)}]}{(\sum_{i=1}^{n}(r_i + s_i)) \prod_{i=1}^{n} r_i! s_i!}$$

where $[C_1, \ldots, C_m]$ *denotes the iterated Lie bracket* $[C_1, [\ldots, [C_{m-2}, [C_{m-1}, C_m]]]\ldots]$*, and* $C^{(r)}$ *denotes the iteration of* r *copies of* C*, i.e* $[C_1^{(2)}, C_2] = [C_1, [C_1, C_2]]$*.*

Proof. See Hall (2003). ■

Concretely, keeping only the first terms, the BCH formula is given by

$$\log\left(\exp(A)\exp(B)\right) = A + B + \frac{1}{2}[A,B] + \frac{1}{12}([A,[A,B]] + [B,[B,A]]) - \frac{1}{24}[B,[A,[A,B]]] + \ldots \quad (2.15)$$

where $A, B \in \mathfrak{g}$. The BCH formula can also be expressed in $\mathbb{R}^p$ by taking the $\vee$ operator on the equation (2.15). Let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^p$ such that $A = \boldsymbol{a}^\wedge$ and $B = \boldsymbol{b}^\wedge$, given the relation (2.10) we have

$$\begin{aligned} \log\left(\exp(\boldsymbol{a}^\wedge)\exp(\boldsymbol{b}^\wedge)\right)^\vee &= \boldsymbol{a} + \boldsymbol{b} + \tfrac{1}{2} ad_G(\boldsymbol{a}^\wedge)\boldsymbol{b} + \tfrac{1}{12}\left(ad_G(\boldsymbol{a}^\wedge)ad_G(\boldsymbol{a}^\wedge)\boldsymbol{b} + ad_G(\boldsymbol{b}^\wedge)ad_G(\boldsymbol{b}^\wedge)\boldsymbol{a}\right) + \ldots \\ &= \boldsymbol{a} + \boldsymbol{b} + \tfrac{1}{2}\boldsymbol{a}^\curlywedge\boldsymbol{b} + \tfrac{1}{12}\left(\boldsymbol{a}^\curlywedge\boldsymbol{a}^\curlywedge\boldsymbol{b} + \boldsymbol{b}^\curlywedge\boldsymbol{b}^\curlywedge\boldsymbol{a}\right) + \ldots \end{aligned} \quad (2.16)$$

The BCH formula is guaranteed to converge in a neighborhood of the identity element of any Lie group. As results, we have $\exp(A)\exp(B) = \exp(A+B)$ if and only if the matrix A and B commute.

1.3.2 Lie Group Actions on Manifolds

In the context of shape analysis, we will focus on specific variations of shapes, which are points on a manifold, such as rotations, translations, and others. These transformations are mathematically defined as actions of Lie group elements on the manifold elements. Let M be a manifold and G be a Lie group. We define the actions of G on M as follows.

Definition 2.4. *A left group action of a Lie group* G *on a manifold* M *is a map* $G \times M \to M$*, denoted by* $(g, p) \mapsto g * p$*, satisfying:*
1. $g_1 * (g_2 * p) = (g_1 g_2) * p, \forall g_1, g_2 \in G$ *and* $p \in M$*,*
2. $e * p = p, \forall p \in M$*.*

The right group action can be defined similarly. We say that G *acts on* M*.*

If M is a Riemannian manifold, the Lie group G also acts on distances between points on M.

Definition 2.5. *A group action of* G *on a Riemannian manifold* M *is called isometric if it preserves the Riemannian metric on* M*, that is, for* $p, q \in M$*,* $d(p,q) = d(g*p, g*q)$ *for all* $g \in G$*. In this case, we also say that* G *acts on* M *by isometries.*

From this group action of G on M we can then define the set of all possible points one can reach in M using the action of G on a particular point of M. This is called the *orbit* of $p \in M$ under the action of G and mathematically defined as the set

$$[p] = G * p = \{g * p \,|\, g \in G\}.$$

Using the orbit, we can define an equivalence relation in M by p is equivalent to q, $p \sim q$, if and only if $p \in [q]$. In particular, we say that the action of G is *transitive* if for of any $p \in M, G*p = M$. Otherwise, we can consider the set of all orbits of G in M, that is, the *quotient space* M/G defined as

$$M/G = \{[p] \,|\, p \in M\}. \quad (2.17)$$

The concept of quotient spaces is especially useful in shape analysis to define space where shapes are invariant to some group transformations. In particular, we can define an inherited distance on the quotient space from the Riemannian metric on M as follows.

Definition 2.6. *If a group action of G on a Riemannian manifold M is isometric and the orbits under G are closed, then we can define a distance on the quotient space M/G using the distance on M as*

$$d_{M/G}([p],[q]) = \min_{g\in G} d_M(p, g*q) = \min_{g\in G} d_M(g*p, q),$$

which is a proper distance satisfying the properties of symmetry, positive definiteness, and triangle inequality (Srivastava and Klassen, 2016).

The main actions of finite-dimensional Lie groups that we will consider in this thesis are the action of the translation group $\mathbb{R}^d$, the rotation group $SO(d)$, and the scaling group $\mathbb{R}^\times$.

2 Functional and Shape Data Analysis

The core objective of this thesis focuses on the statistical analysis of motion trajectories obtained through a motion capture system. Motion capture data, in practice, is acquired discretely, with minimal time intervals between successive position measurements of specific points on the body. As a result, we are dealing with data that are theoretically continuous, but for which we have only discrete observations. These are known as functional data. The analysis, representation, and explanation of this type of data is the aim of the mathematical field of *functional data analysis* (FDA). This field has developed by extending tools for multivariate statistics through various works of Ramsay and Dalzell (1991); James (2002); Müller and Stadtmüller (2005); Reiss and Ogden (2007); Ramsay et al. (2007). Moreover, the statistical analysis of a collection of functions involves examining their geometries or shapes. This constitutes a significant part of the large field of *shape analysis* (Srivastava and Klassen, 2016; Younes, 2010; Srivastava et al., 2011; Pennec et al., 2020; Bauer et al., 2022). In this section, we introduce the main existing approaches of these two domains that we will use later in the functional and shape analysis of our data.

2.1 Functional Data Analysis

2.1.1 Smoothing Functional Data

The data we are interested in are discretely observed or recorded as n pairs (t_i, y_i), where y_i represents the measured value at the specific instant or position t_i. Despite the discrete nature of these data points, their contextual information suggests an inherent continuous structure, represented as a function $t \mapsto x(t)$. Consequently, the initial step in analyzing this type of data involves transforming these discrete values into a continuous function $x(t)$ that can be evaluated across the entire domain $t \in \mathcal{T}$. In cases where the data points are sufficiently dense and noiseless, straightforward interpolation methods can be applied. However, when the data lacks density or is affected by noise, a smoothing method becomes essential (Green and Silverman, 1993). In this context, the underlying model for the observations can be expressed as follows:

$$y_i = x(t_i) + \epsilon_i\,,$$

where ϵ_i represents the observation error or noise. The objective of smoothing is to estimate the latent variable x within this model. To achieve this, the main approach in functional data analysis involves considering a set of mathematically independent basis functions. These basis functions are chosen to relatively approximate any function through a linear combination of a sufficient number K of these functions. In mathematical terms, the function $x(t)$ is represented as:

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t) \tag{2.18}$$

where $\phi_k(t)$ are known basis functions. Note that the number of basis functions considered influences the degree of smoothness of the function. Indeed, when $K = n$, it essentially amounts to interpolation. Therefore, the parameter K must be optimized or selected based on the considered dataset.

B-spline Basis Functions. The choice of the basis functions also depends on the data. Among the commonly used basis functions in the literature are Fourier basis, polynomial basis (monomials), wavelet functions, and B-spline basis. In this thesis, we use the B-spline basis functions known for their flexibility in approximating and modeling complex shapes (de Boor, 1978; Silverman, 1985). They are defined as piecewise polynomials characterized by their local support. A B-spline of order p is a collection of piecewise polynomial functions $B_{i,p}(t)$ of degree $p-1$ that are non-zero on a few adjacent subintervals defined by a sequence of knots $\tau_0, \ldots, \tau_m$. They can be constructed with the Cox-de Boor recursion formula:

$$B_{i,0}(t) := \begin{cases} 1 \text{ if } \tau_i \leq t < \tau_{i+1}, \\ 0 \text{ otherwise.} \end{cases}$$

$$B_{i,p}(t) := \frac{t - \tau_i}{\tau_{i+p} - \tau_i} B_{i,p-1}(t) + \frac{\tau_{i,p+1} - t}{\tau_{i+p+1} - \tau_{i+1}} B_{i+1,p-1}(t).$$

The degree of the B-spline basis, as well as the placement of knots, must be adjusted to adapt to the specific characteristics of the data. A common practice is to use equally spaced knots, which works well when the data points are uniformly distributed. However, if this is not the case, another approach consists of placing interior knots at the quantiles of the argument distribution, i.e., at every j-th data point, where j is a fixed number.

Least Square Fitting. Given the observations (t_i, y_i) and the K basis functions ϕ_k, we must determine the value of the coefficients of the basis expansion c_k. This is achieved by minimizing the least squares criterion

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}} \sum_{i=1}^{n} \left(y_i - \sum_{k}^{K} c_k \phi_k(t_i) \right)^2 \tag{2.19}$$

where $\boldsymbol{c} = (c_1, \ldots, c_k)$ is the vector notation of the K coefficients. By considering the vector $\boldsymbol{y}$ with the observed values y_i and the matrix $\boldsymbol{\Phi}$ whose columns are the basis functions ϕ_k, this criterion can be expressed in matrix notations as

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}} (\boldsymbol{y} - \boldsymbol{\Phi c})^T (\boldsymbol{y} - \boldsymbol{\Phi c}). \tag{2.20}$$

Then, taking the derivative of the criterion with respect to $\boldsymbol{c}$ yields the solution

$$\hat{\boldsymbol{c}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{y}, \tag{2.21}$$

and the vector of fitted values is $\hat{\boldsymbol{y}} = \boldsymbol{\Phi}\hat{\boldsymbol{c}}$. If we are in the case of a weighted smoothing problem, with a weights symmetric positive definite matrix $\boldsymbol{W}$, the solution becomes

$$\hat{\boldsymbol{c}} = (\boldsymbol{\Phi}^T \boldsymbol{W} \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{W} \boldsymbol{y}. \tag{2.22}$$

In particular, this appears when we consider localized basis function estimators. Basically, the idea is to estimate the function at a point t using a weighted combination of the function observation values at neighbor points. This extends the previous least squares criterion as

$$\boldsymbol{c}(t) = \arg\min_{\boldsymbol{c}} \sum_{i=1}^{n} w_i(t) \left(y_i - \sum_{k}^{K} c_k \phi_k(t_i) \right)^2 \tag{2.23}$$

where the weight functions w_i are defined by a kernel function as $w_i = \text{Kern}((t_i - t)/h)$, and the parameter h is the bandwidth of the kernel.

Roughness Penalty. When dealing with noisy or irregularly sampled functional data, it is difficult to control the degree of smoothness using smoothing methods such as basis smoothing or localized basis smoothing. To overcome this, the roughness penalty approach consists of adding a penalty term to the objective function being optimized to balance between fitting the data closely and maintaining a certain level of smoothness in the estimated function (Green and Silverman, 1993; Chong, 2002). In this way, the smoothness property is expressed explicitly at the criterion level rather than implicitly in terms of the number of basis functions used. The penalty term discourages the estimated function from being too oscillatory, leading to a smoother and more interpretable result. In practice, a popular way to measure the concept of "roughness" in a function is by considering the square of its second derivative $[x''(t)]^2$. Then, the criterion for smoothing with a roughness penalty can be formulated as follows:

$$\text{minimize} \quad \sum_{i=1}^{n}(y_i - x(t_i))^2 + \lambda \int [x''(t)]^2 dt \tag{2.24}$$

In this equation, the first term represents the data fitting term, ensuring that the estimated function $x(t)$ closely matches the observed data points, and the second term, $\int [x''(t)]^2 dt$, represents the roughness penalty. The parameter λ controls the trade-off between data fitting and smoothness. Therefore, the choice of this parameter is crucial and often requires careful tuning based on the characteristics of the data and the desired level of smoothness in the estimated function. The roughness penalty term can also be expressed as a function of the basis decomposition of x as

$$\int [x''(t)]^2 dt = \boldsymbol{c}^T \left[\int \boldsymbol{\Phi}''(t)(\boldsymbol{\Phi}''(t))^T dt \right] \boldsymbol{c} = \boldsymbol{c}^T \boldsymbol{R} \boldsymbol{c}. \tag{2.25}$$

Then, the optimal coefficients are obtained by minimizing the criterion

$$(\boldsymbol{y} - \boldsymbol{\Phi c})^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{\Phi c}) + \lambda \boldsymbol{c}^T \boldsymbol{R} \boldsymbol{c}, \tag{2.26}$$

leading to the solution

$$\hat{\boldsymbol{c}} = (\boldsymbol{\Phi}^T \boldsymbol{W} \boldsymbol{\Phi} + \lambda \boldsymbol{R})^{-1} \boldsymbol{\Phi}^T \boldsymbol{W} \boldsymbol{y}. \tag{2.27}$$

2.1.2 Summary Statistics for Functional Data

We now assume that we have a collection of functions $x_i(t), i = 1, \ldots, N$. We can define basic statistics on this set, such as the mean and standard deviations in a functional version, by analogy with the multivariate statistical analysis. The mean function is simply the point-wise average across the replications defined as

$$\bar{x}(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(t)\,. \tag{2.28}$$

Similarly, the variance function is defined as

$$\text{var}(x(t)) = \frac{1}{N-1} \sum_{i=1}^{N} (x_i(t) - \bar{x}(t))^2 \tag{2.29}$$

and the standard deviation function is the square root of the variance function. Beyond these basic statistics, covariance and correlation functions summarize the interdependence of functions across different time points. The covariance function is defined for all t_1, t_2 as

$$\text{cov}_x(t_1, t_2) = \frac{1}{N-1} \sum_{i=1}^{N} (x_i(t_1) - \bar{x}(t_1))\,(x_i(t_2) - \bar{x}(t_2)) \tag{2.30}$$

and similarly, the correlation function is defined as

$$\mathrm{corr}_x(t_1, t_2) = \frac{\mathrm{cov}_x(t_1, t_2)}{\sqrt{\mathrm{var}(x(t_1))\mathrm{var}(x(t_2))}} \tag{2.31}$$

These covariance and correlation functions provide insights into how the functions in the collection vary together across different values of t. If we consider two different collections of functions x_i and f_i for $i = 1, \ldots, N$, this is given by the cross-covariance and cross-correlation functions defined as

$$\mathrm{cov}_{f,x}(t_1, t_2) = \frac{1}{N-1} \sum_{i=1}^{N} \left(f_i(t_1) - \bar{f}(t_1) \right) \left(x_i(t_2) - \bar{x}(t_2) \right), \tag{2.32}$$

$$\mathrm{corr}_{f,x}(t_1, t_2) = \frac{\mathrm{cov}_{f,x}(t_1, t_2)}{\sqrt{\mathrm{var}(f(t_1))\mathrm{var}(x(t_2))}}. \tag{2.33}$$

Functional Principal Components Analysis. Exploring variability within a dataset can also be achieved using Principal Components Analysis (PCA). Building upon the principles of PCA for multivariate data, an extension tailored for functional data is known as functional principal components analysis (fPCA) (Yao et al., 2005; Reiss and Ogden, 2007). The basic idea is to capture the main modes of variability in a set of functions by identifying a set of K orthonormal functions ϕ_k so that the expansion of each curve in terms of these basis functions provides the best possible approximation of the curve. Then, the FPCA involves expressing each function as a sum of principal components:

$$x_i(t) = \bar{x}(t) + \sum_{k=1}^{K} \xi_{i,k} \phi_k(t) \tag{2.34}$$

where $\bar{x}(t)$ is the mean function, ξ_{ik} are the functional principal component scores for the i-th observation and k-th component, and $\phi_k(t)$ are the functional principal components, which are the functions capturing the main modes of variability. In practice, we compute the full sample covariance matrix $\boldsymbol{K}$ using the sample covariance function $\mathrm{cov}_x(t_i, t_j)$ on a discrete grid $t_1 < \ldots < t_n$. Then, by taking its singular value decomposition (SVD) $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, the principal directions are defined as the K first columns of the matrix $\boldsymbol{U}$ and the observed principal coefficients as $\xi_{i,k} = \langle x_i, \boldsymbol{U}_k \rangle$.

2.1.3 Phase and Amplitude Variability

A common problem in functional data analysis, rendering these sample statistics less relevant, is the presence of variations in the temporality of functions within the dataset, known as *phase variations* (Marron et al., 2015). The phase of a functional curve refers to its temporal alignment or positioning in the domain, such as time or space. Phase variability occurs when observed functions exhibit shifts or misalignments in their temporal patterns. A typical example for scalar functions in $\mathbb{R}$ is depicted in Figure 2.1, where the peaks of each curve are not present at the same time. Consequently, computing a mean or performing fPCA directly on this set is intuitively not meaningful. Therefore, prior alignment of functions is necessary to obtain a more comprehensible and coherent data analysis. Various methods for function alignment have been proposed (Kneip and Gasser, 1992; Liu and Müller, 2004; Tang and Müller, 2008). In general, alignment is modeled by a *warping function* γ that modifies the distribution of points along the curve. Mathematically, this is expressed as the right composition of x by γ, denoted as $x \circ \gamma$. From the aligned curves (Figure 2.1), we can clearly observe residual variations corresponding to so-called *amplitude variations*. These reflect variations in the height or size of observed patterns and must also be taken into account. Assuming we have a phase alignment algorithm capable of identifying the set of warping functions $\{\gamma_i\}$ and the set of aligned curves $\{\tilde{x}_i\}$, in this case, the mean of the collection of functions x_i is defined as the previously defined sample mean of the aligned functions. Subsequently,

fPCA can be performed separately on the aligned functions to identify the main modes of amplitude variations and on the warping functions to identify the main modes of phase variations (Kneip and Ramsay, 2008; Happ et al., 2019; Tucker et al., 2019).

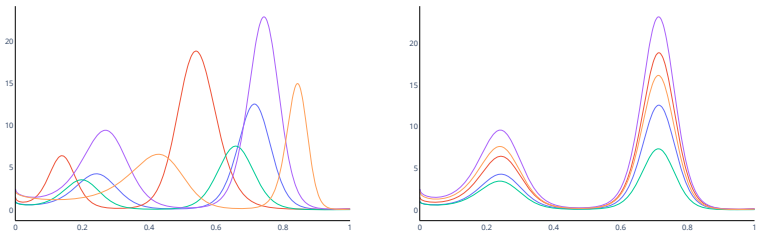


Figure 2.1: Illustration of phase and amplitude variations in a set of scalar functions (left) and corresponding aligned functions with residual amplitude variations (right).

These observations, introduced here in one dimension within the framework of functional data analysis, highlight the general problem in curve shape analysis, which involves properly identifying and modeling different types of variations to obtain meaningful statistics across the entire set of curves. This necessitates, among other things, a method for aligning or registering functions or curves.

2.2 Shape Analysis of Functional Data

As previously introduced, analyzing a collection of curves and computing basic statistics on them first requires identifying the various types of variations present in these curves. The idea is to isolate what comes under the inherent shape of the curves or specific patterns from variations in parameterization (phase) or rigid transformations, known as "shape-preserving", such as translation, rotation, or scaling. Shapes are defined as what remains invariant under the action of the reparameterization group, and each mentioned transformation (Grenander, 1993). From there, the question becomes how to quantify the similarities and dissimilarities of shapes within a set of curves and how to model the variability of these shapes. This falls within the broad field of mathematics known as shape analysis (Dryden and Mardia, 1998; Small, 1996; Kendall et al., 1999; Kendall and Le, 2010). Within this field, several methods exist and differ based on the considered representation of the geometric object defining the shape. Staying within the case of Euclidean curves in $\mathbb{R}^d$, Glaunès et al. (2008); Younes (2010) proposes shape analysis methods, such as Large Deformation Diffeomorphic Metric Mapping (LDDMM), in which curves are represented as a sequence of discrete points. These methods rely on the deformation of points from one curve to another induced by the action of a diffeomorphism on the ambient space. On the other hand, Srivastava et al. (2011); Srivastava and Klassen (2016); Bauer et al. (2022) models the geometric object constituting the shape continuously as a function, even if the actual observations of the curves are generally discrete. This allows, in some applications, to retain the inherently continuous nature of the original data and simultaneously model temporal variations if necessary. In this thesis, given our interest in motion trajectories, time is a significant variable, and we thus consider the case of continuous objects. Mathematically, the arc-length parameterization uniquely defines the shape of an absolutely continuous curve x. The arc-length function is defined for all $t \in [0, 1]$ as

$$s(t) = \int_0^t \left\| \frac{dx(u)}{du} \right\|^2 du \tag{2.35}$$

and defines a natural parameterization of the curve. The shape of the curve is then the image X of the function x under this parameterization, $X(s(t)) = x(t)$.

As illustrated previously in the case of scalar functions, a major step in shape analysis is to align or register curves with respect to their geometric features. The advantage of continuous modeling of curves is to use parameterization as a means to align points along the curves. For this purpose, we need to consider the reparameterization group, $\mathrm{Diff}_+([0,1])$, which represents all boundary-preserving and orientation-preserving diffeomorphisms of $[0,1]$ to itself. $\mathrm{Diff}_+([0,1])$ is a Lie group with the group operation given by composition, the identity element given by the identity function from $[0,1]$ to $[0,1]$ and inverse function well defined as: for $\gamma, \gamma_1, \gamma_2 \in \mathrm{Diff}_+([0,1])$

$$(\gamma_1 \circ \gamma_2)(t) = \gamma_1(\gamma_2(t)), \qquad \gamma_{id}(t) = t, \qquad \gamma \circ \gamma^{-1} = \gamma^{-1} \circ \gamma = \gamma_{id}. \tag{2.36}$$

We refer to these elements as "warping" functions or "re-parametrization" functions. The action of warping functions on the curve space is by right composition: $(x, \gamma) = x \circ \gamma$. For more detail on the geometry of this Lie group we refer to Kac (1990).

2.2.1 Registration of Functional Data with the $\mathbb{L}^2$ Norm

We consider two absolutely continuous curves x_1 and x_2 defined from $[0,1]$ to $\mathbb{R}^d$. The reparametrization of x_1 by a warping function $\gamma \in \mathrm{Diff}_+([0,1])$ results in the function $x_1 \circ \gamma$ that maps the same points in $\mathbb{R}^d$ as x_1 and thus has the same shape. Registering x_2 to x_1 involves finding the optimal warping function γ such that for all $t \in [0,1]$, $x_2 \circ \gamma(t)$ and $x_1(t)$ are as close as possible. Therefore, it requires an objective function quantifying the distance between $x_2 \circ \gamma(t)$ and $x_1(t)$. As x_1 and x_2 lies in $\mathbb{L}^2([0,1], \mathbb{R}^d)$, a natural choice will be the $\mathbb{L}^2$ norm, as used in Ramsay and Li (1998); Tang and Müller (2008). We recall that the $\mathbb{L}^2$ norm of x is defined as $\|x\|_{\mathbb{L}^2} = \sqrt{\int_0^1 \|f(t)\|_2^2 dt}$. Then, the registration problem amounts to finding the optimal warping function γ^* such that

$$\gamma^* = \underset{\gamma \in \mathrm{Diff}_+([0,1])}{\arg\min} \|x_1 - x_2 \circ \gamma\|_{\mathbb{L}^2}^2 . \tag{2.37}$$

However, this objective function has two main limitations: the *lack of isometry or symmetry* and the *pinching effect*. The isometry property defined in Definition 2.4 is not verified by the $\mathbb{L}^2$ metric, $\|x_1 \circ \gamma - x_2 \circ \gamma\|_{\mathbb{L}^2} \neq \|x_1 - x_2\|_{\mathbb{L}^2}$. This means that the alignment of x_2 to x_1 will not necessarily be the inverse of the alignment of x_1 to x_2. Therefore, this objective function is not well defined for a registration problem. The second main problem is a degeneracy of the $\mathbb{L}^2$ norm, called the "pinching effect". Basically, the quantity $\|x_1 - x_2 \circ \gamma\|_{\mathbb{L}^2}^2$ can be made infinitesimally small by squeezing or pinching a large part of x_2. Figure 2.2 of Pennec et al. (2020) illustrate this effect with the example of two scalar functions on $[0,1]$.

A possible solution to limit this effect is to penalize large warpings using an additional roughness penalty term in the objective function like

$$\gamma^* = \underset{\gamma \in \mathrm{Diff}_+([0,1])}{\arg\min} \|x_1 - x_2 \circ \gamma\|_{\mathbb{L}^2}^2 + \lambda \mathcal{P}(\gamma). \tag{2.38}$$

Common choices for $\mathcal{P}(\gamma)$ are $\int \dot{\gamma}(t)^2 dt$, $\int \ddot{\gamma}(t)^2 dt$ or $\int \|1 - \dot{\gamma}(t)\|_2^2 dt$. Although this method limits the pinching effect, it also restricts the space of solutions for warping functions. Additional illustrations are available in Pennec et al. (2020); Srivastava and Klassen (2016). The $\mathbb{L}^2$ metric does not, therefore, appear to be suitable for shape analysis of Euclidean curves, being blind to "spikes".

2.2.2 Pre-Shape and Shape Space

Given these limitations, a common approach in shape analysis is to consider an intermediate mathematical representation of curves with properties appropriate to shape analysis. Introduced first by Kendall et al. (1999) with landmarks and then used to define the *elastic shape analysis* approach, the idea is

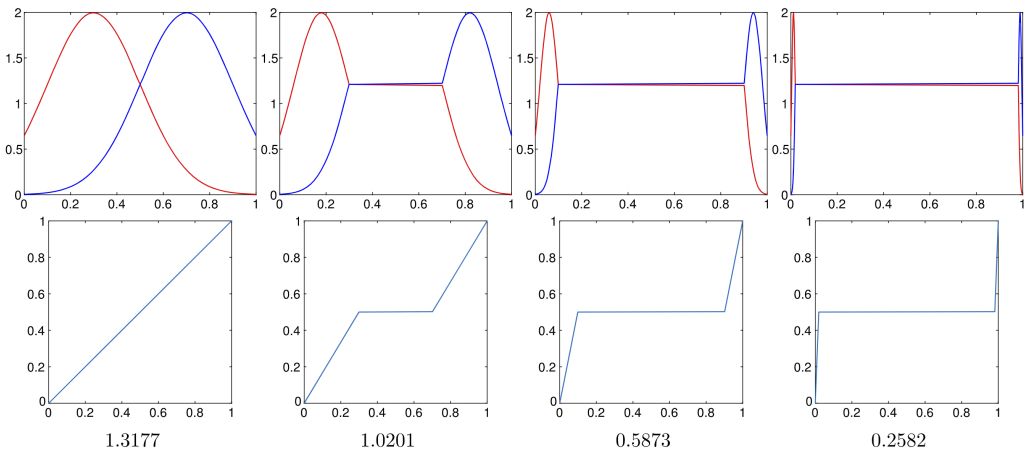


Figure 2.2: Figure taken from Pennec et al. (2020) illustrating the pinching effect that appears when registering the red and blue functions with the $\mathbb{L}^2$ norm as an objective function. In each column they show $x_1 \circ \gamma$ and $x_2 \circ \gamma$ (top panel), γ (bottom panel), and the value of $\|x_1 \circ \gamma - x_2 \circ \gamma\|^2_{\mathbb{L}^2}$ below them.

to represent a curve x with a mathematical representation $\mathcal{R}(x)$ which is invariant under certain transformations. In addition, one of the stakes of such representation is to offer an (infinite-dimensional) Riemannian manifold structure that brings, as introduced in Section 1.2, powerful and flexible tools defining metrics and studying the geometry of shapes (Lang, 2006; Sommer et al., 2020). The set of these representations is called the *pre-shape space*. The remaining shape-preserving transformations in the pre-shape space are then removed by using the notion of equivalence class and quotient spaces. This leads to the definition of the *shape space*. In this setting, the distance between two curves is defined through the distance between the mathematical representations modulo any remaining shape-preserving transformations. This leads to a unified framework for registration, shape comparison, and analysis.

The method most commonly used today for shape analysis of Euclidean curves modeled as continuous functions involves the representation of curves known as the *Square-Root Velocity Function* (SRVF) defined first in Srivastava et al. (2011) and further in Srivastava and Klassen (2016); Su et al. (2017); Bauer et al. (2021). The framework related to this representation is called the *elastic shape analysis*. The term "elastic" underlines the fact that the registration of curves is part of their shape comparisons.

Definition 2.7. *The square-root velocity function (SRVF) of x absolutely continuous curve from $[0,1]$ to $\mathbb{R}^d$ is the mathematical representation of x defined by*

$$q(t) = \begin{cases} \frac{\dot{x}(t)}{\sqrt{\|\dot{x}(t)\|_2}}, & \|\dot{x}\|_2 \neq 0, \\ 0, & \|\dot{x}\|_2 = 0. \end{cases} \tag{2.39}$$

In one dimensional case $d = 1$, this representation is referred to as *square-root slope function* (SRSF) and reduces to $q(t) = \text{sign}(\dot{x}(t))\sqrt{\|\dot{x}\|_2}$. Under this representation, the pre-shape space is defined as the set of all square-root velocity representations of curves in $\mathbb{R}^d$, which is the unit Hilbert sphere inside $\mathbb{L}^2([0,1], \mathbb{R}^d)$, and have a structure of Riemannian manifold equipped with the $\mathbb{L}^2$ metric. Moreover, this representation offers several advantages for the shape analysis of curves. The transformation from x to q is bijective. The reparameterization of x by a warping function γ modifies its SRVF as $q * \gamma = (q \circ \gamma)\sqrt{\dot{\gamma}}$, and this group action preserves the $\mathbb{L}^2$ norm, $\|q\| = \|q * \gamma\|$. Thus, the action is isometric, and there is no issue with the pinching effect. Additionally, the $\mathbb{L}^2$ norm between two SRVFs is already invariant to translations of the corresponding curves. Therefore, the shape space $\mathcal{S}$ under this representation is

defined as the quotient space of the pre-shape space with the group $\text{Diff}_+([0,1])$ and $SO(d)$. Then, the distance between two shapes of curves x_1 and x_2 is defined under the SRVF representation as

$$d_{\mathcal{S}}(x_1, x_2) = \inf_{\gamma \in \text{Diff}_+([0,1]), O \in SO(d)} \|q_1(t) - O(q_2 * \gamma)(t)\| \ . \tag{2.40}$$

This metric is called an *elastic metric* and is shown to be a Sobolev-type metric (Srivastava and Klassen, 2016), which also considers geometric information of derivatives. More details on this representation for Euclidean curves and the associated Riemannian shape analysis framework (geodesic, mean, etc.) will be given in Chapter 5. Extensions of this elastic shape analysis framework have been developed for curves with values in manifolds or Lie groups. In particular, we will use the SRV transform of $SO(d)$-valued curves in Chapter 5.

The case of this specific representation illustrates the method employed in the field of shape analysis with continuous objects to establish a suitable framework for statistical analysis of Euclidean curves in $\mathbb{R}^d$. This method relies on the pre-shape space's Riemannian framework and the shape space's definition as the quotient space by the reparameterization group, allowing for simultaneous curve registration. Therefore, the choice of representation is a pivotal step that defines the expression of the shape space, the considered metrics, and consequently, the obtained statistics.

3 The Frenet-Serret Framework

In this thesis, we are particularly interested in the Frenet-Serret framework, which provides a geometric representation of a curve in $\mathbb{R}^d$. The Frenet-Serret formulas were initially defined in three-dimensional Euclidean space by two French mathematicians who independently discovered them: Jean Frédéric Frenet (1852), and Joseph Alfred Serret (1851). They were later extended to arbitrary dimensions d by Camille Jordan (1874). These formulas describe the kinematic properties of a point moving along a differentiable curve in Euclidean space. Furthermore, they introduce local descriptors of a curve at each point, capturing essential shape information and exhibiting invariance properties under rigid transformations. Consequently, the Frenet-Serret framework focuses on the intrinsic geometry of a curve rather than its global position and represents a powerful tool for analyzing and understanding the geometric properties of curves, making it a central topic of investigation in this thesis.

We introduce the Frenet-Serret framework for curves of any dimension d. Let $F([0,1], \mathbb{R}^d)$ be the set of curves $x \in AC_0\left([0,1], \mathbb{R}^d\right)$ d-times continuously differentiable, and with the first d derivatives linearly independent. $F([0,1], \mathbb{R}^d)$ is called the set of *Frenet curves*. In the following, we will restrict the shape space to be the set

$$\mathcal{S}([0,1], \mathbb{R}^d) = F([0,1], \mathbb{R}^d)/\text{Diff}_+([0,1]) \, . \tag{2.41}$$

The *Frenet frame* $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_d$ associated with $X \in \mathcal{S}([0,1], \mathbb{R}^d)$ is uniquely defined by applying the Gram-Schmidt process to the first d derivatives of X. The unit tangent vector is $\boldsymbol{e}_1(s) = X'(s)$, the $(d-2)$ next unit vectors are defined by

$$\boldsymbol{e}_j(s) = \frac{X^{(j)}(s) - \sum_{i=1}^{j-1} \langle X^{(j)}(s), \boldsymbol{e}_i(s) \rangle \boldsymbol{e}_i(s)}{\left\| X^{(j)}(s) - \sum_{i=1}^{j-1} \langle X^{(j)}(s), \boldsymbol{e}_i(s) \rangle \boldsymbol{e}_i(s) \right\|} \tag{2.42}$$

and the last one is the cross-product of the first $d-1$ vectors. We define the function Q that maps to $s \in [0,1]$ along the curve the corresponding Frenet frame

$$Q(s) = [\boldsymbol{e}_1(s) \,|\, \boldsymbol{e}_2(s) \,|\, \ldots \,|\, \boldsymbol{e}_d(s)] \, . \tag{2.43}$$

The function Q is a measurable curve from $[0,1]$ to the Lie group of rotation matrices $SO(d)$ called the *Frenet path.*

Theorem 2.3 (Frenet-Serret equation Kühnel (2002))**.** *Let $X \in \mathcal{S}([0,1],\mathbb{R}^d)$ and $Q(s)$ its associated Frenet path. Then there are functions $\theta_1,\dots,\theta_{d-1}$ defined on that curve with $\theta_1,\dots,\theta_{d-2} > 0$, so that every θ_i is $(d-1-i)$-times continuously differentiable and*

$$Q'(s) = Q(s)A_{\boldsymbol{\theta}}(s) \tag{2.44}$$

where $\boldsymbol{\theta}(s) = (\theta_1(s),\dots,\theta_{d-1}(s))^T$ and

$$A_{\boldsymbol{\theta}}(s) = \begin{bmatrix} 0 & -\theta_1(s) & 0 & \dots & 0 \\ \theta_1(s) & 0 & -\theta_2(s) & \ddots & \vdots \\ 0 & \theta_2(s) & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & -\theta_{d-1}(s) \\ 0 & \dots & 0 & \theta_{d-1}(s) & 0 \end{bmatrix}$$

and θ_i is called the i-th Frenet curvature, and the equation is called the Frenet-Serret equation.

The Frenet-Serret equation with an initial condition $Q(0) = Q_0$ defines an ordinary differential equation on the Lie group $SO(d)$ where the function $s \mapsto A_\theta(s)$ has values in the Lie algebra of skew-symmetric matrices. This equation can also be expressed in function of the time variable t as

$$\frac{dQ(s(t))}{dt} = \dot{s}(t)Q(s(t))A_{\boldsymbol{\theta}}(s(t))\,. \tag{2.45}$$

Lemma 2.1. *The action of the translation, rotation, scaling, and reparametrization group on the Frenet curvatures and the Frenet path are defined as follows. Let $x \in F([0,1],\mathbb{R}^d), O \in SO(d), a \in \mathbb{R}^d, L \in \mathbb{R}^\times$ and $h \in Diff_+([0,1])$. The curve defined by $\tilde{x}(t) = a + \frac{1}{L}Ox(h(t))$ has a Frenet path $\tilde{Q}(s) = OQ(Ls)$ and Frenet curvatures $\tilde{\boldsymbol{\theta}}(s) = L\boldsymbol{\theta}(Ls)$. Therefore, the Frenet curvatures are invariant under rotations, translations, and reparametrization.*

Theorem 2.4 (Fundamental theorem of the local theory of curves Kühnel (2002))**.** *Let $\theta_1,...,\theta_{d-1} \in C^\infty([0,1],\mathbb{R})$ such that $\theta_1,\dots,\theta_{d-2} > 0$. For a given $X_0 \in \mathbb{R}^d$ and $Q_0 \in SO(d)$ there is a unique $X \in \mathcal{S}([0,1],\mathbb{R}^d)$ parametrized by arc length and satisfying the following three conditions:*

- *$X(0) = X_0$,*
- *Q_0 is the Frenet frame of X at point $s = 0$,*
- *$\theta_1,...,\theta_{d-1}$ are the Frenet curvatures of X.*

Theorems 2.3 and 2.4 state that there is a bijection, up to a translation and a rotation, between the shape space $\mathcal{S}([0,1],\mathbb{R}^d)$, the set of admissible Frenet curvatures $\mathcal{H}$ and the set of corresponding Frenet paths $\mathcal{F}_0$, where

$$\mathcal{H} = \left\{ \boldsymbol{\theta} \in \mathbb{L}^2\left([0,1],\mathbb{R}^{d-1}\right) \mid \theta_1,\dots,\theta_{d-2} > 0 \right\}, \tag{2.46}$$

$$\mathcal{F}_0 = \left\{ Q \in \mathbb{L}^2([0,1],SO(d)) \text{ such that } Q'(s) = Q(s)A_{\boldsymbol{\theta}}(s), Q(0) = I_d, \boldsymbol{\theta} \in \mathcal{H} \right\}. \tag{2.47}$$

Three-dimensional case. Given that the motion trajectories we are interested in are in three dimensions, we specify the Frenet frame in this case. Note that up to now, it is primarily in this dimension that the Frenet-Serret frame is mainly used, and thus, for which most estimation methods are developed. For a curve $\mathbb{R}^3$ with shape function $X : [0, 1] \to \mathbb{R}^3$, the three vectors of the associated Frenet frame are called the *tangent vector* T, the *normal vector* N, and the *bi-normal vector* B, such that $Q(s) = [T(s)|N(s)|B(s)]$. The two Frenet curvatures are known as the *curvature* function, $s \mapsto \kappa(s)$, and the *torsion* function, $s \mapsto \tau(s)$. They have interpretable physical meanings. The curvature function measures how sharply the curve changes direction at a given point, and the torsion function measures the degree to which the curve twists and turns as it moves along its path (Carmo, 1976).

Part I

Statistical Estimation of Frenet Curvatures

3

Profiled Frenet-Serret ODE Inference for Frenet Curvatures Estimation

Abstract

In this chapter, we provide an overview of existing methods used to estimate Frenet curvatures from discrete and noisy numerical data. This includes direct approaches using the extrinsic formulas of Frenet curvatures, as well as methods proposed by Park and Brunel (2019), treating curvatures estimation as a parameter estimation problem of a differential equation. These methods involve smooth estimation of the Frenet path, followed by a least squares minimization adapted to the geometry of the problem to estimate the curvatures as parameters of the Frenet-Serret differential equation. We delve into these methods in detail. As a contribution, we introduce an alternative approach for obtaining a smooth estimation of the Frenet path inspired by smoothing techniques using tracking algorithms (Wahba, 1978; Brunel and Clairon, 2015). This method proves to be more computationally efficient and yields results similar to the existing method. Finally, we present examples using simulated data to compare these methods and highlight their limitations.

Contents

1 Introduction

We consider the problem of estimating the Frenet curvatures defined in Theorem 2.3. In a d-dimensional ambient space $\mathbb{R}^d$, these Frenet curvatures are $d-1$ scalar functions defined on $[0,1]$ that can be grouped in one vector-valued function $\boldsymbol{\theta} : [0,1] \to \mathbb{R}^{d-1}$. Assuming that we have potentially noisy observations of Frenet curvatures, estimating them would correspond to a classical problem of multivariate functional regression (Ramsay and Silverman, 2005). However, in practice, the Frenet curvatures are not directly observable. Instead, the available experimental data typically consist of noisy and discrete observations of the Euclidean trajectory in $\mathbb{R}^d$. Therefore, in this case, intermediate estimates of additional geometric components of the curve are necessary for estimating the Frenet curvatures.

Methods commonly found in literature (Kim et al., 2013; Sangalli et al., 2009; Lewiner et al., 2005) are based on the extrinsic formulas of the Frenet curvatures. These ones involve combinations of the d first derivatives of the Euclidean curve. Hence, these methods rely on nonparametric estimation of the derivatives from noisy and discrete observations of the trajectory and application of these extrinsic formulas.

More recent works introduce the idea of regarding the estimation of Frenet curvatures as an ODE inference problem (Park and Brunel, 2019; Hirsh et al., 2021). Indeed, the vector-valued function $\boldsymbol{\theta}$ is by definition (Theorem 2.3) a *structural* parameter of the Frenet-Serret differential equation (2.44) on $SO(d)$: $Q'(s) = Q(s)A_{\boldsymbol{\theta}}(s)$. Compared with the previous approach, this one is more focused on the functional parameter $\boldsymbol{\theta}$ within the geometric Frenet-Serret framework. Basically, the main objective of an ODE inference problem is to estimate the unknown parameters of the differential equation based on experimentally observed data of state variables, which may sometimes be only partially observed. Due to the wide use of differential equations to describe and quantify phenomena in many scientific fields, the inverse problem of differential equation models is a significant challenge, and numerous different methods have been proposed to solve it (Ramsay and Hooker, 2017). Compared to the usual setting in which this problem is studied, our framework introduces two additional constraints: the state variable Q takes values in the Lie group $SO(d)$, and the parameter $\boldsymbol{\theta}$ to estimate is time-varying.

In this context, a classic method to solve this ODE estimation problem is the penalized non-linear least squares (NLS) procedure (Cao et al., 2012; Chen and Wu, 2008). This method extends the classic NLS one (Biegler et al., 1986) to the time-varying parameter estimation case by considering a basis expansion of the parameter $\boldsymbol{\theta}(s) = \boldsymbol{\Phi}(s)\boldsymbol{c}$, where $\boldsymbol{\Phi}(s)$ represents the basis function expansion, and $\boldsymbol{c}$ denotes the parameter coefficients. The estimation process involves numerical methods to approximate the ODE solution and optimization of coefficients $\boldsymbol{c}$ by minimizing the penalized residual sum of squares from the noisy state observations. This is typically achieved using a Gauss-Newton algorithm. However, the NLS method, while widely used, is known to face several limitations. It is computationally intensive, requiring numerical approximations, repeated ODE integrations, and nonlinear optimizations with possible multiple local minima in the residual sum of squares function (Miao et al., 2011).

Facing these limitations, a family of alternative methods, called *collocation* methods, has been introduced. They are based on a two-stage estimation procedure, including an intermediate stage of non-parametric approximation of the state variable before parameters estimation by minimization of a data fitting criterion, e.g. the residual sum of squares. Consequently, these methods involve the estimation of an additional parameter, the state variable, defined by Ramsay et al. (2007) as a *nuisance* parameter, which is often not of primary concern but essential to the further estimation of the structural parameters of interest. A direct version of these two-stage collocation procedures is to obtain a nonparametric proxy of the state variable by observed data preprocessing without considering the ODE model, and then use it for estimating the ODE parameters, as proposed by Varah (1982). The extension of this version, proposed by Ramsay and Silverman (2005), consists of iterating these two steps, enabling the use of a nonparametric model-based smoother for the state variable approximation that depends on the previous estimated ODE parameters. This process is commonly referred to as *profiling*. Different methods are used to obtain the nonparametric proxy of the state variable; some original collocation methods use polynomial bases decomposition, Ramsay et al. (2007) define the Generalized Smoothing method based on a splines basis expansion of the state variable (Qi and Zhao, 2010; Campbell and Chkrebtii, 2013), and alternatively Brunel and Clairon (2015) introduce a tracking approach. However, it should be noted that these methods are all initially defined within an Euclidean framework.

Park and Brunel (2019) propose Frenet curvatures estimation procedures inspired by these two-stages ODE estimation methods, which they extend to $SO(d)$-valued state variable and time-varying ODE parameter. Their approach relies on the estimation of a nonparametric proxy of the Frenet path and the use of Lie group ODE approximation for the estimation of Frenet curvatures. Obtaining a smooth nonparametric estimator of the Frenet path is already not straightforward as it involves estimating a manifold-valued curve (Kim et al., 2021; Patrangenaru and Ellingson, 2015; Su et al., 2012). In particular, different methods can be considered depending on the model used for state observations. In their main algorithm, they assume having noisy discrete observations of the entire Frenet path. In this context, they adapt the classic local smoothing methods to Lie group data to better handle the inherent geometric structure. As a contribution, we propose an alternative method for computing this nonparametric proxy from noisy observations of the Frenet path inspired by the tracking approach developed by Brunel and Clairon (2015). Both of these smoothing techniques are model-based and thus require a candidate parameter $\tilde{\boldsymbol{\theta}}$. Therefore, they are used in a profiled iterative estimation algorithm, such as the Generalized Smoothing method of Ramsay et al. (2007). However, in practice, the assumption of having noisy discrete observations of the entire Frenet path is rarely true. Instead, one typically only has noisy discrete observations of the Euclidean curve. Hence, we also introduce methods for deriving a smooth estimator of the Frenet path by preprocessing the Euclidean curve observations. As a result, this estimator can be used directly to estimate Frenet curvatures in a non-iterative algorithm, like Varah's one.

The chapter is structured as follows. Firstly, in Section 2, we introduce the most popular state-of-the-art method for computing Frenet curvatures based on extrinsic formulas. Next, in Section 3, we recall some fundamental mathematical concepts related to the Lie group $SO(d)$ and ODE resolution. Then, in Section 4, we present methods for obtaining a non-parametric estimator of the Frenet path, depending on the observation model considered. These ones include the methods proposed by Park and Brunel (2019) and a proposed new alternative that proves more efficient. Finally, we derive the estimation algorithm of Frenet curvatures in Section 5. We propose some simulation studies in Section 6 to compare the different methods.

2 Related Work: Extrinsic Formulas of Frenet Curvatures

In practice and in most applications, we observe a noisy and discrete version of the Euclidean trajectory. Suppose that we have n noisy observations $\boldsymbol{y}_i \in \mathbb{R}^d$, $i = 1, \dots n$ on a grid $t_1 < \dots < t_n$, which satisfy

$$\boldsymbol{y}_i = x(t_i) + \boldsymbol{\epsilon}_i \tag{3.1}$$

where $\boldsymbol{\epsilon}_i$'s are independent random variables with mean zero and covariance matrix $\boldsymbol{\Gamma}$, and $t \mapsto x(t)$ is the time-parameterized curve function. We assume that the arc-length parametrization can be done relatively easily by a simple estimate of the first time-derivative of the curve x given the observations $(t_1, y_1), \dots, (t_n, y_n)$. The equation (3.1) can then be rewritten using the estimated arc-length grid $s_1 < \dots < s_n$ as

$$\boldsymbol{y}_i = X(s_i) + \boldsymbol{\epsilon}_i \tag{3.2}$$

where the function X is the shape of the curve with a corresponding Frenet path Q solution of the Frenet-Serret equation $Q'(s) = Q(s)A_{\boldsymbol{\theta}}(s)$ (Section 3). The most commonly used method in the literature (Kim et al., 2013; Sangalli et al., 2009; Lewiner et al., 2005) for estimating the Frenet curvatures from this observations model (3.2) is to employ their extrinsic formulas directly. These formulas are defined in terms of the curve derivatives of order 1 to d. Therefore, a straightforward method involves using local polynomial regression to estimate the derivatives first and then computing the Frenet curvatures based on these extrinsic formulas.

2.1 Derivatives Estimation using Local Polynomial Regression

The direct application of the Frenet equations requires accurate estimates of curve derivatives. Several non-parametric regression estimators, such as spline, kernel, or wavelet smoother, are available for this purpose. However, most standard non-parametric regression techniques are primarily designed for one-dimensional curves. To extend these methods to multi-dimensional curves, the problem can be reformulated as a simultaneous estimation problem. Here we present the estimation of the derivatives using a multi-dimensional local polynomial regression. Using local neighborhoods and weighting functions, local polynomial regression is well adapted to unequally spaced argument values and captures localized features by avoiding the excessive influence of distant outliers (Fan and Gijbels, 1996).

Let $s_0 \in [0, 1]$, suppose the $(d+1)$ first derivatives of X at the point s_0 exist. Then, the unknown regression function X can be locally approximated by a polynomial of order d using the Taylor expansion at s in a neighborhood of s_0. This polynomial can be fitted locally by a multi-dimensional local polynomial regression defined by the following weighted least squares criterion

$$\min_{\boldsymbol{\beta}_j \in \mathbb{R}^d} \sum_{i=1}^{n} K_h(s_i - s_0) \left\{ \boldsymbol{y}_i - \sum_{j=0}^{d} \boldsymbol{\beta}_j (s_i - s_0)^j \right\}^T \left\{ \boldsymbol{y}_i - \sum_{j=0}^{d} \boldsymbol{\beta}_j (s_i - s_0)^j \right\} \tag{3.3}$$

where $\boldsymbol{\beta}_j \in \mathbb{R}^d$ are the unknown coefficients and $K_h(\cdot) = K(\cdot/h)/h$ is a kernel function that assigns a weight to each point, parameterized by h the bandwidth controlling the size of the local neighborhood. The criterion (3.3) can be rewritten in matrix notation by denoting $\boldsymbol{Y} = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_n)^T$ the full matrix of observations of size $n \times d$, $\boldsymbol{W}(s_0) = diag\{K_h(s_j - s_0)\}$ the $n \times n$ diagonal weight matrix, $B = (\boldsymbol{\beta}_0, \dots, \boldsymbol{\beta}_d)$ the $d \times (d+1)$ matrix of coefficients, and the $n \times (d+1)$ design matrix $\boldsymbol{U}(s_0) = \left(\mathbb{1}_{\mathbb{R}^n}, (\boldsymbol{s} - s_0 \mathbb{1}_{\mathbb{R}^n}), \dots, (\boldsymbol{s} - s_0 \mathbb{1}_{\mathbb{R}^n})^d \right)$, with $\boldsymbol{s} = (s_1, \dots, s_n)^T$. Then, we have

$$\min_B \left\{ (\boldsymbol{Y} - \boldsymbol{U}(s_0)B)^T \boldsymbol{W}(s_0) (\boldsymbol{Y} - \boldsymbol{U}(s_0)B) \right\}. \tag{3.4}$$

This problem is a classic least squares minimization problem, Ramsay and Silverman (2005); by taking the derivatives with respect to the parameter B, we obtain a normal equation for simultaneous estimation,

$$\boldsymbol{U}(s_0)^T\boldsymbol{W}(s_0)\boldsymbol{U}(s_0)\hat{B}(s_0) = \boldsymbol{U}(s_0)^T\boldsymbol{W}(s_0)\boldsymbol{Y}\,. \tag{3.5}$$

Note that this equation can be equivalently written as the system of the normal equations for each marginal variable with the same design matrix $\boldsymbol{U}$ and weight matrix $\boldsymbol{W}$. Then, for $q \leq d$, $X^{(q)}(s) = q!\hat{\boldsymbol{\beta}}_q(s)$.

When fitting multi-dimensional curves, selecting appropriate smoothing parameters is an additional challenge. We control the overall smoothness of the multidimensional curve, assuming that the degree of smoothness does not vary significantly across variables. Typically, a 10-fold cross-validation method is used to select the optimal bandwidth parameter. It consists of dividing the data into 10 folds and iteratively training and evaluating the model on different combinations of these folds. We minimize the mean square error between the observations test set and the estimated zero-order derivative evaluated on the test grid:

$$\sum_{k=1}^{K}\frac{1}{|T_k|}\sum_{i\in T_k}\left(\boldsymbol{y}_i - X(s_i;h)\right)^T\left(\boldsymbol{y}_i - X(s_i;h)\right), \tag{3.6}$$

where T_k is the kth index set based on $K = 10$ random partitions of the observations $\{\boldsymbol{y}_i\}_{i=1,\dots,n}$. Typically, this problem is solved by a grid search minimization over different values of h.

Remark 3.1. *To consider the equation (3.2) as the observations model instead of (3.1), we assume that we are able to obtain a good estimate of the arc-length parametrization. This requires estimating the first time-derivative of the curve, which can typically be done by applying a local polynomial regression of that kind using the time parametrization.*

2.2 Extrinsic Formulas of Frenet Curvatures from Derivatives Estimators

The extrinsic formulas of the Frenet curvatures $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{d-1})$ are, for all $i = 1, \dots, n$,

$$\theta_i(s) = \left\langle \boldsymbol{e}_i'(s), \boldsymbol{e}_{i+1}(s)\right\rangle, \tag{3.7}$$

where $\boldsymbol{e}_i(s)$ is the ith unit vector of the Frenet frame, which is defined as a function of the i first derivatives $\{X^{(1)}, \dots, X^{(i)}\}$ (2.42). For example, in dimension 3, these formulas can be expressed for the curvature κ and torsion τ directly in function of the Euclidean curve derivatives as

$$\kappa(s(t)) = \frac{\|\dot{x}(t)\times\ddot{x}(t)\|}{\|\dot{x}(t)\|^3} = \|X'(s)\times X''(s)\|\,, \tag{3.8}$$

$$\tau(s(t)) = \frac{\langle \dot{x}(t)\times\ddot{x}(t),\, \dddot{x}(t)\rangle}{||\dot{x}(t)\times\ddot{x}(t)||^2} = \frac{\langle X'(s)\times X''(s), X'''(s)\rangle}{||X'(s)\times X''(s)||^2}. \tag{3.9}$$

Based on the derivatives estimates obtained through the local polynomial regression presented previously in Section 2.1, one can derive initial estimations of the curvature and torsion using the given formulas. However, these formulas exhibit high instability and unreliability in practical applications due to their reliance on derivative ratios and combinations, which are subject to significant numerical errors. To address this issue, a potential approach for obtaining more likely estimates of the continuous Frenet curvatures functions is to apply an additional step of penalized B-spline regression to further smooth the raw data. Despite this, careful consideration should be given to the potential limitations and challenges associated with these estimators to ensure accurate and reliable results (see Table 3.2).

3 Mathematical Preliminaries

The Frenet-Serret framework is characterized by the Frenet-Serret differential equation (2.44), written in the matrix Lie group $SO(d)$. Consequently, we recall some preliminary notions about the group $SO(d)$ and the resolution of differential equations in a Lie group. In addition, we introduce a few mathematical concepts in linear quadratic control theory that will be useful later on.

3.1 The Special Orthogonal Group

The *special orthogonal group* is a particular matrix Lie group, also called rotations group, corresponding to the set of valid rotation matrices. In dimension $d \geq 2$, it is defined as

$$SO(d) = \left\{ R \in \mathbb{R}^{d\times d} \,\middle|\, RR^T = I_d \text{ and } \det(R) = 1 \right\}. \tag{3.10}$$

This set is not a vector space as it is not closed under addition, and the null matrix is not a valid element of $SO(d)$, but it can be shown to be a non-abelian matrix Lie group. The inverse element in $SO(d)$ is simply $R^{-1} = R^T$ as $RR^T = I_d$. The Lie algebra of the special orthogonal group is the vector space of dimension $\frac{d(d-1)}{2}$, denoted by $\mathfrak{so}(d)$ and corresponds to the set of infinitesimal rotations. Their matrix representations correspond exactly to the set of skew-symmetric matrices. Then, for $R \in SO(d)$ there is $\boldsymbol{a} \in \mathbb{R}^{d(d-1)/2}$ such that $R = \exp(\boldsymbol{a}^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} (\boldsymbol{a}^\wedge)^n$ (2.7).

Lemma 3.1. *The geodesic distance between two points R_1 and R_2 on $SO(d)$ induced by the scale-dependent left invariant metric (2.5) is given by*

$$d(R_1, R_2) = \left\| \log\left(R_1^T R_2\right) \right\|_F. \tag{3.11}$$

Three-dimensional case: In the following, we will use in particular the rotations group in the 3-dimensional space, $SO(3)$, which has specific properties. The Lie algebra $\mathfrak{so}(3)$ is the 3-dimensional vector space defined as

$$\mathfrak{so}(3) = \left\{ \boldsymbol{a}^\wedge = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}^\wedge = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \in \mathbb{R}^{3\times 3} \,\middle|\, \boldsymbol{a} \in \mathbb{R}^3 \right\}. \tag{3.12}$$

In the case of $\mathfrak{so}(3)$ the adjoint representation is particularly simple, we have $ad_{SO(3)}(\boldsymbol{a}^\wedge) = \boldsymbol{a}^\wedge$. As a result, in $\mathfrak{so}(3)$, the Lie bracket operation corresponds to the cross product of the two vectors,

$$[\boldsymbol{a}^\wedge, \boldsymbol{b}^\wedge]^\vee = ad_{SO(3)}(\boldsymbol{a}^\wedge)\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}. \tag{3.13}$$

In other dimensions $d > 3$, the expressions for the adjoint and the Lie bracket do not have known simple forms. In addition, the Jacobian of $SO(3)$ is then given by $J_{SO(3)}(\boldsymbol{a}) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)!} (\boldsymbol{a}^\wedge)^n$, which can also be expressed and computed by the useful formulas:

$$J_{SO(3)}(\boldsymbol{a}) = \frac{\sin(\|\boldsymbol{a}\|)}{\|\boldsymbol{a}\|} I + \frac{1}{\|\boldsymbol{a}\|^2} \left(1 - \frac{\sin(\|\boldsymbol{a}\|)}{\|\boldsymbol{a}\|}\right) \boldsymbol{a}^\wedge (\boldsymbol{a}^\wedge)^T + \frac{1 - \cos(\|\boldsymbol{a}\|)}{\|\boldsymbol{a}\|^2} \boldsymbol{a}^\wedge, \tag{3.14}$$

$$J_{SO(3)}(\boldsymbol{a})^{-1} = \frac{\|\boldsymbol{a}\|}{2} \cot\left(\frac{\|\boldsymbol{a}\|}{2}\right) I + \frac{1}{\|\boldsymbol{a}\|^2} \left(1 - \frac{\|\boldsymbol{a}\|}{2} \cot\left(\frac{\|\boldsymbol{a}\|}{2}\right)\right) \boldsymbol{a}^\wedge (\boldsymbol{a}^\wedge)^T - \frac{1}{2} \boldsymbol{a}^\wedge. \tag{3.15}$$

3.2 Solving Linear Matrix ODE on Lie Group

Since the Frenet-Serret ODE is a linear matrix differential equation on the Lie group $SO(d)$, we focus on this specific case. Consider the linear matrix ODE defined on the Lie group G as

$$\dot{Y}(t) = Y(t)A(t) \tag{3.16}$$

where $Y(t) \in G$ and $t \mapsto A(t)$ is a continuous matrix function. First, a fundamental concept for solving an ODE is the flow over time t, denoted by $\varphi(t, \cdot)$ (Hairer et al., 2006). It is the mapping that, to any point $Y_0 \in G$, associates the value $Y(t)$ of the solution with initial value $Y(0) = Y_0$. That is, $\varphi(t, Y_0) = Y(t)$ if $Y(0) = Y_0$. To express the dependence on the initial time, we can extend the definition of the flow as $\varphi(t, s, Y_0) = Y(s + t)$ if $Y(s) = Y_0$ so $\varphi(t - s, s, Y_0) = Y(t)$. The essential property of the flow is the group property, i.e for all $s, u, t \in [0, 1]$ and $Y_0 \in G$, $\varphi(t - s, s, Y_0) = \varphi(t - u, u, \varphi(u - s, s, Y_0))$, allowing us to express any localized solution coherently to the global solution. This group property of the flow can be interpreted as a *self-prediction property*: if $s \mapsto Y(s)$ is a solution to equation (3.16), then for all t, $s \in [0, 1]$ such that $|t - s| \leq 1$, we have

$$Y(t) = \varphi(t - s, s, Y(s)). \tag{3.17}$$

To express the flow $\varphi(t, \cdot)$ of the equation (3.16), Magnus (1954) proposes to search for a solution of the form

$$Y(t) = Y_0 \exp(\Omega(t)) \tag{3.18}$$

where $\exp(\cdot)$ is the matrix exponential map (2.8) and $t \mapsto \Omega(t)$ is a matrix function defined in the Lie algebra $\mathfrak{g}$.

Theorem 3.1 (Magnus (1954))**.** *The solution of the differential equation (3.16) can be written as* $Y(t) = Y_0 \exp(\Omega(t))$ *with* $\Omega(t)$ *defined by*

$$\dot{\Omega} = d\exp_{\Omega}^{-1}(A(t)), \quad \Omega(0) = 0. \tag{3.19}$$

where the inverse of the derivative of the matrix exponential is given by

$$d\exp_{\Omega}^{-1}(H) = \sum_{k \geq 0} \frac{B_k}{k!} ad(\Omega)^k H,$$

with B_k *the Bernoulli numbers and* $ad(\Omega)A = [\Omega, A] = \Omega A - A\Omega$ *the adjoint operator defined in (2.10). As long as* $\|\Omega(t)\| < \pi$*, the convergence of the* $d\exp_{\Omega}^{-1}$ *expansion is assured.*

Proof. (Hairer et al., 2006) Comparing the derivatives of $Y(t) = Y_0 \exp(\Omega(t))$ (see Lemma 4.1 in Hairer et al. (2006)),

$$\dot{Y}(t) = Y_0 \dot{\Omega}(t) \left(\frac{d}{d\Omega} \exp \Omega(t) \right) = Y_0 \exp(\Omega(t)) \left(d\exp_{\Omega(t)} \left(\dot{\Omega}(t) \right) \right),$$

with (3.16) we obtain $A(t) = d\exp_{\Omega(t)} \left(\dot{\Omega}(t) \right)$. Applying the inverse operator $d\exp_{\Omega}^{-1}$ to this relation yields the differential equation (3.19) for $\Omega(t)$. ∎

The function $\Omega(t)$ is shown to admit the so-called Magnus expansion (Hairer et al., 2006; Iserles et al., 2000)

$$\Omega(t) = \int_0^t A(s)ds - \frac{1}{2}\int_0^t \left[\int_0^\tau A(s)ds, A(\tau) \right] d\tau + \frac{1}{4}\int_0^t \left[\int_0^\tau \left[\int_0^\sigma A(\mu)d\mu, A(\sigma) \right] d\sigma, A(\tau) \right] d\tau + \dots \tag{3.20}$$

In practice, to numerically compute the solution of equation (3.16), the Magnus expansion has to be truncated, and the integrals have to be approximated by numerical quadrature (Blanes et al., 2009). One

possibility is to use the following second-order approximation based on a midpoint rule and truncating after the first term, which is valid for small enough intervals $|t-s|$,

$$Y(t) = Y(s)\exp\left((t-s)A\left(\frac{t+s}{2}\right)\right) + O\left((t-s)^2\right). \tag{3.21}$$

It defines a numerical approximation of the flow $\varphi(t,\cdot)$ in (3.17), which can be seen as an Euler-Lie method that possesses several interesting features: it respects the $SO(d)$ constraint, has an explicit and pointwise dependence in A, and the approximation is uniform on $SO(d)$.

3.3 Linear Quadratic Optimal Control

Optimal control is a field of study that deals with finding the best possible control strategy for a dynamic system, such that a certain objective function is optimized. It is widely used in engineering, economics, and various other disciplines to optimize the performance of systems. One important class of optimal control problems is linear quadratic control (LQC), which focuses on linear systems with quadratic cost functions (Whittle, 1990; Chow, 1986). LQC provides a powerful and tractable framework for analyzing and designing optimal control strategies. Consider a linear time-variant system described by the state-space model:

$$\begin{cases} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= Cx(t) + Du(t) \end{cases} \tag{3.22}$$

where $x(t) \in \mathbb{R}^d$ represents the state vector, $u(t) \in \mathbb{R}^d$ denotes the control input, and $y(t)$ represents the output of the system. The matrix functions $A \in L^2([0,T],\mathbb{R}^{d\times d})$ and $B \in L^2([0,T],\mathbb{R}^{d\times d})$ define the dynamics, and the matrices C and D the input-output relationship of the system. The goal of linear quadratic control is to design a control input $u(t)$ that minimizes a quadratic cost function of the form:

$$J(u) = \int_0^T \left(x(t)^T W(t)x(t) + u(t)^T R(t)u(t)\right)dt + x(T)^T Px(T) \tag{3.23}$$

where P a positive matrix, $W \in L^2([0,T],\mathbb{R}^{d\times d})$ a positive matrix for all $t \in [0,T]$, and R a definite positive matrix for all $t \in [0,T]$ respecting the coercivity condition:

$$\exists \alpha > 0 : \forall u \in L^2([0,1],\mathbb{R}^d), \int_0^T u(t)^T R(t)u(t)dt \geq \alpha \int_0^T \|u(t)\|_2^2 dt.$$

The term $x(T)^T Px(T)$ is known as the terminal cost and is sometimes (Linear Quadratic Regulator) supposed to be zero.

Theorem 3.2. *Given the system (3.22) and the cost function (3.23), it exists a unique optimal trajectory x_u associated with the unique optimal control u that has the feedback form $u(t) = R(t)^{-1}E(t)B(t)x_u(t)$ where E is the matrix solution of the Riccati ODE:*

$$\begin{cases} \dot{E}(t) &= W(t) - A(t)^T E(t) - E(t)A(t) - E(t)B(t)U(t)^{-1}B(t)^T E(t) \\ E(T) &= -P \end{cases}$$

and the minimal cost is equal to: $J(u) = -x_0^T E(0)x_0$.

Linear quadratic control offers a trade-off between stability and performance. The choice of the weighting matrices W and R influences the trade-off between regulating the state variables and minimizing the control effort. Increasing the weight on the control input improves performance but may result in less stable behavior.

4 Non-parametric Proxy of Frenet-Serret ODE Solution Path

As part of the two-stage ODE estimation procedure, we develop non-parametric methods for estimating the Frenet-Serret ODE solution path, known as the Frenet path. Firstly, we demonstrate how to pre-process the Euclidean curve observations to obtain a smooth Frenet path estimator without any prior knowledge of the parameter $\boldsymbol{\theta}$. Secondly, given a parameter candidate $\tilde{\boldsymbol{\theta}}$, we introduce two model-based smoothers of the noisy Frenet path observations. The first smoother is defined and detailed in Park and Brunel (2019), while the second one is a novel alternative we propose that utilizes a tracking approach.

4.1 Frenet Path Estimation by Preprocessing of Noisy Euclidean Curve Observations

The Frenet frame is defined (2.42) as the orthonormalization of the d first arc-length-derivatives of the curve $\{X'(s), \dots, X^{(d)}\}$. Therefore, given the proposed local polynomial regression for estimating the derivatives in Section 2.1, the Frenet frame $Q(s)$ can be computed for all $s \in [0,1]$ by applying the Gram-Schmidt orthonormalization procedure to these smooth derivatives estimators (LaPlace, 1820).

However, being primarily interested in estimating the Frenet path and not the derivatives, we can imagine a version of local polynomial regression that takes directly into account orthogonality constraints. We define this method here in dimension $d = 3$. In this case, the Frenet frame is composed of the tangent, normal and bi-normal vectors, $Q(s) = [T(s)|N(s)|B(s)]$, and is solution of the Frenet-Serret ODE with the Frenet curvatures parameter $\boldsymbol{\theta}(s) = (\kappa(s), \tau(s))^T$. The Taylor expansion of $X(s)$, in a neighborhood of s_0, of order 3 can be rewritten in functions of these Frenet components given

$$\begin{cases} X'(s_0) &= T(s_0) \\ X''(s_0) &= \kappa(s_0)N(s_0) \\ X'''(s_0) &= -\kappa(s_0)^2 T(s_0) + \kappa'(s_0)N(s_0) + \kappa(s_0)\tau(s_0)B(s_0)\,. \end{cases} \tag{3.24}$$

This clearly shows that the orthogonality constraints of the Frenet frame are translated into

$$\|X'\|^2 = 1, \qquad \langle X', X''\rangle = 0, \qquad \langle X', X'''\rangle + \langle X'', X''\rangle = 0\,. \tag{3.25}$$

Then, by incorporating these constraints, the local polynomial regression (3.4) can be turned into a constrained local polynomial estimation that can be solved by the method of Lagrange multipliers. To express these constraints, we define

$$f_1(\boldsymbol{\beta}) = \|\beta_1\|^2 - 1, \qquad f_2(\boldsymbol{\beta}) = 2\beta_1^T\beta_2, \qquad f_3(\boldsymbol{\beta}) = \beta_1^T\beta_3 + \beta_2^T\beta_2\,. \tag{3.26}$$

Then, the gradients of the Lagrangian function with respect to the parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$ and the Lagrange multipliers $\lambda_1, \lambda_2, \lambda_3$ must be zeros,

$$\boldsymbol{U}(s_0)^T\boldsymbol{W}(s_0)\boldsymbol{U}(s_0)\boldsymbol{\beta} = \boldsymbol{U}(s_0)^T\boldsymbol{W}(s_0)\boldsymbol{Y} + \lambda_1\nabla f_1(\boldsymbol{\beta}) + \lambda_2\nabla f_2(\boldsymbol{\beta}) + \lambda_3\nabla f_3(\boldsymbol{\beta}), \tag{3.27}$$

$$f_1(\boldsymbol{\beta}) = 0, \quad f_2(\boldsymbol{\beta}) = 0, \quad f_3(\boldsymbol{\beta}) = 0. \tag{3.28}$$

Since the parameters $\lambda_1, \lambda_2, \lambda_3$ depend on each other in a rather complicated way, we solve the problem iteratively for each parameter, in which case an explicit solution exists. In practice, because they are linked, the problem can be solved only for λ_1 and λ_2. The optimal selection of the bandwidth parameter h is then performed in the same way as for the classic local polynomial regression (Section 2.1).

These methods derive a non-parametric estimator of the Frenet path $\hat{Q}$ directly from noisy observations of the Euclidean curve (3.2). Assuming the Frenet path estimator is sufficiently smooth and reliable, Frenet curvatures can be computed directly from it in a second stage with the method detailed in Section 5.

4.2 Model-based Smoother of Noisy Frenet Path Observations

In this section, we assume to have noisy observations of the Frenet path as inputs. These Frenet frame observations are unlikely to correspond to experimental data from real applications but, more likely, to the outputs of a first pre-processing stage of Euclidean curve observations, as one of those considered in the previous section, which may still be too noisy for parameter estimation. Hence, we denote $\{U_i\}_{i=0\dots n}$ the noisy and discrete observations of the Frenet path Q at positions $s_0 < \ldots < s_n$, which can be expressed as:

$$U_i = Q(s_i)\exp(W_i), \tag{3.29}$$

where W_i are random matrices in $\mathfrak{so}(d)$ (i.e $\exp(W_i)$ are random rotations). To simplify the analysis, we suppose that the matrices W_i are independent and identically distributed, satisfying $\mathbb{E}[W_i] = 0$, $\mathbb{E}[\exp(W_i)] = \mathbf{I}_d$, and $\mathbb{E}\left[\|W_i\|_F^2\right] = \sigma_W^2 < \infty$. The density of the noise is assumed to be unknown but fixed. For the further estimation of Frenet curvatures, we aim to recover the Frenet path Q from the noisy observations $\{U_i\}_{i=0\dots n}$. Inspired by the profiling procedure of the Generalized Smoothing (Ramsay et al., 2007) approach, Park and Brunel (2019) propose a model-based smoother of the Frenet path observations given a parameter candidate $\tilde{\boldsymbol{\theta}}$. They are looking for a solution path $\tilde{Q}(\cdot, \tilde{\boldsymbol{\theta}})$ that solves approximately the original ODE model with the estimate $\tilde{\boldsymbol{\theta}}$ while being close to the data $\{U_i\}_{i=0\dots n}$. Their approach relies on local smoothing of the approximated ODE flow $\varphi_{\tilde{\boldsymbol{\theta}}}$. We propose an alternative method inspired by the tracking approach defined in Brunel and Clairon (2015), which looks for the solution path $\tilde{Q}(\cdot, \tilde{\boldsymbol{\theta}})$ of a perturbed Frenet-Serret ODE parametrized by $\tilde{\boldsymbol{\theta}}$ and closed to the data $\{U_i\}_{i=0\dots n}$.

4.2.1 Local Karcher Mean Smoother

To approximate the solution path Q from the noisy observations $\{U_i\}_{i=0\dots n}$, Park and Brunel (2019) introduces a model-based smoother defined by local smoothing of ODE flows originating from various initial conditions. They assume that the flow $\varphi_{\boldsymbol{\theta}}$, associated with the previously estimated parameter $\boldsymbol{\theta}$, is known. Instead of relying on a Frenet path, their approach aims to find a path denoted as $t \mapsto M(t)$ that closely matches the given data and approximately satisfies the self-prediction property: for all i, and for all $t \in [0, 1]$, we should have approximately $\varphi_{\boldsymbol{\theta}}(t - s_i, s_i, U_i) \approx Q(t)$. By combining and averaging all these predictions, we can effectively reduce the prediction error caused by the noise propagation through the flow $\varphi_{\boldsymbol{\theta}}$. Obviously, the typical arithmetic average is not suitable for this purpose, as it is implicitly based on Euclidean assumptions. Therefore, they use instead a Karcher mean to determine the optimal prediction at time t,

$$M_h(t) = \underset{M \in SO(3)}{\arg\min} \frac{1}{n+1} \sum_{i=0}^{n} K_h(t - s_i) \left\| \log\left(M^T \varphi_{\boldsymbol{\theta}}(t - s_i, s_i, U_i) \right) \right\|_F^2 \tag{3.30}$$

where the kernel K_h takes into consideration the increasing uncertainty for distant points. This approach bears similarities to local polynomial smoothing techniques and aligns with the general idea of adapting smoothing methods to manifolds and other non-Euclidean spaces. In these settings, conventional least squares estimates are replaced by Fréchet or Karcher means, allowing for more appropriate handling of the inherent geometric structure and improvement of the smoothing process (Jakubiak et al., 2006; Samir et al., 2012; Thomas Fletcher, 2013). The Karcher mean in $SO(d)$ does not benefit from a closed-form expression. However, the following property guarantees its existence and uniqueness and provides a simple and efficient gradient algorithm for computing it.

Proposition 3.1 (Park and Brunel (2019))**.** *Let $R_0, \ldots, R_n$ observations in $SO(3)$, with positive weights w_i, such that they are all in a ball of radius $\min\left(inj(SO(3)), \pi/\sqrt{k}\right)/2$ where $inj(SO(3))$ is the injectivity radius and k is the sectional curvature of $SO(3)$. Then there exists a unique Karcher mean defined as*

$$\bar{R} = \underset{M \in SO(3)}{\arg\min} \frac{1}{2} \sum_{i=0}^{n} w_i \left\| \log(M^T R_i) \right\|_F^2$$

The gradient of $f(M) = \frac{1}{2}\sum_{i=0}^{n} w_i \left\|\log(M^T R_i)\right\|_F^2$ *is* $grad(f(M)) = \sum_{i=0}^{n} w_i \log(M^T R_i)$*, and the sequence defined for* $k \geq 1$*,* $R_{k+1} = R_k \exp(-grad(f(R_k)))$ *converges to* $\bar{R}$ *for any initial guess* R_0*.*

Proof. See Le (2004); Rentmeesters and Absil (2011). ■

Given $h > 0$ and based on the flow approximation (3.21), the smooth nonparametric proxy of the Frenet path is then estimated at any point $t \in [0, 1]$ by,

$$M_h(t) = \underset{M \in SO(3)}{\arg\min} \frac{1}{n+1} \sum_{i=0}^{n} K_h(t - s_i) \left\| \log\left(M^T U_i \exp\left((t - s_i) A_{\boldsymbol{\theta}} \left(\frac{t + s_i}{2} \right) \right) \right) \right\|_F^2. \tag{3.31}$$

This method is particularly interesting because it utilizes a formulation specific to the Lie group $SO(d)$. As a result, the obtained estimator $M_h(t)$ satisfies the orthogonality constraint of the Frenet frame. However, the drawback of this method is its high computational cost, requiring a large number of matrix exponential calculations.

4.2.2 Tracking Smoother

To define a more efficient smoother algorithm, we draw inspiration from the fast spline smoothing method of Wahba (1978) based on Kalman filtering. Building upon this approach, we propose here, as a contribution, a new model-based algorithm for smoothing the Frenet path, which proves to be faster and equally effective as the one described above.

Main Idea. The relation between spline smoothing and tracking of a state-space model was made very early by Wahba (1978) and exploited in a series of papers (Wecker and Ansley, 1983; Ansley and Kohn, 1985). The main idea is to express the classic smoothing spline problem in a state space model formulation. Given some data $\{Y(t_1), \ldots, Y(t_n)\}$, the Lg-smoothing spline of degree $2m-1$ to the data is the solution of the following minimization problem. Find g such that the first $2m-2$ derivatives are continuous and g minimize:

$$\sum_{i=1}^{n} (g(t_i) - Y_i)^2 + \lambda \int_0^1 [Lg(u)]^2 \, du, \tag{3.32}$$

where $Y_i = Y(t_i)$, λ must be chosen, and the penalty is defined by a higher-order differential operator $L : g \mapsto Lg = g^{(m)} - \sum_{k=0}^{m-1} a_k(t) g^{(k)}$, and where $t \mapsto a_k(t)$ are appropriate smooth functions. With a complex penalization of this form, although we can still obtain a basis expansion by solving a convex quadratic problem, this can be computationally challenging, in particular when we have a lot of observations. Therefore Wahba (1978) did introduce a stochastic differential equation model

$$Lg(t) = \sigma \lambda^{1/2} \frac{dW_t}{dt} \tag{3.33}$$

where dW_t/dt is a white noise process. We consider the extended state $G_t = (g(t)\, g'(t) \ldots g^{(m-1)}(t))^T$ in $\mathbb{R}^m$ and the time-dependent $m \times m$ matrix

$$A_t = \begin{bmatrix} 0 & & & \\ \vdots & & I_{m-1} & \\ 0 & & & \\ a_{m-1}(t) & a_{m-2}(t) & \ldots & a_0(t) \end{bmatrix} \tag{3.34}$$

and we introduce the stochastic differential equation

$$dG_t = A_t G_t dt + \sigma \lambda^{-1/2} \boldsymbol{e}_m dW_t. \tag{3.35}$$

where $\boldsymbol{e}_m = (0 \dots 0\, 1)^T \in \mathbb{R}^m$. The trajectory of G_t is then very close to the behavior of the function g solution of (3.33), as the last dimension of G_t satisfies this equation. If $\phi(t,s)$ is the fundamental solution to the linear deterministic ODE $\dot{G}(t) = A_t G(t)$, then the solution to the general linear SDE (3.35) starting from G_s, $s \leq t$ is (Sarkka, 2006)

$$G(t) = \phi(t-s,s)G(s) + \int_s^t \phi(t,\tau)\sigma\lambda^{-1/2}\boldsymbol{e}_m dW_\tau. \tag{3.36}$$

From the continuous time formulation, we derive an exact discrete-time state-space process on the grid $t_1 < \cdots < t_n$

$$\begin{cases} G(t_{k+1}) & = \phi(t_{k+1}-t_k, t_k)G(t_k) + u_k \\ Y_k & = (1\,0\,\dots 0)G(t_k) + \epsilon_k \end{cases} \tag{3.37}$$

with $u_k = \int_{t_k}^{t_{k+1}} \phi(t,s)\,[0\,0\,\cdots 1]^T\, dW_s \in \mathbb{R}^m$ being a Gaussian variable with zero mean and variance $(\sigma^2/\lambda)U(t_{k+1},t_k)$. Then, given the parameters λ and σ, the Kalman filtering and smoothing algorithms can be implemented to perform estimation of the state G_t and, therefore, estimation of the smooth function g.

Application to the Frenet-Serret Differential Equation. Our objective is to estimate the smooth Frenet path function $s \mapsto Q(s) = [\boldsymbol{e}_1(s)|\boldsymbol{e}_2(s)|\dots|\boldsymbol{e}_d(s)] \in SO(d)$ solution of the Lie group Frenet-Serret differential equation (2.44), from the noisy and discrete observations $\{U_i\}_{i=0\dots n}$ (3.29). This problem bears similarities to the smoothing spline problem (3.32), but it is not defined in a Euclidean ambient space, unlike Wahba's method framework. Therefore, we consider the alternative expression of the Frenet-Serret differential equation in the Euclidean space $\mathbb{R}^{d^2}$ by considering the linear representation of the state variable $\boldsymbol{q}(s) = (\boldsymbol{e}_1(s), \dots, \boldsymbol{e}_d(s))^T$ in $\mathbb{R}^{d^2}$. Then, the Frenet-Serret equations take the form of a linear ordinary differential equation defined in $\mathbb{R}^{d^2}$,

$$\boldsymbol{q}'(s) = (-A_{\boldsymbol{\theta}}(s) \otimes \mathbf{I}_d)\boldsymbol{q}(s)\,, \tag{3.38}$$

subjects to the additional constraint that the d vectors in $\mathbb{R}^d$ composing $\boldsymbol{q}(s)$ are mutually orthonormal. The data themselves can also be reformulated in a linear manner as $\mathbf{u}_i = (U_i^{(0)}, \dots, U_i^{(d-1)})^T \in \mathbb{R}^{d^2}$, where $U_i^{(j)}$ is the vector corresponding to the jth column of the matrix U_i. Following the approach of Wahba (1978) and Brunel and Clairon (2015), we relax the ODE constraint with the introduction of the following penalized scheme:

$$\boldsymbol{q}'(s) = (-A_{\boldsymbol{\theta}}(s) \otimes \mathbf{I}_d)\boldsymbol{q}(s) + \mathbf{v}(s) \tag{3.39}$$

where the function $s \mapsto \mathbf{v}(s)$ can be any function in L^2. This perturbed ODE model accounts for several sources of model misspecification, including parameter uncertainty and random measurement errors. The objective is then to minimize the norm of the forcing function $\|\mathbf{v}\|_{L^2}^2$ while being close to the observations data $\{\mathbf{u}_i\}_{i=0,\dots,n}$.

We are looking for the solution path of (3.39), given the known initial condition $\boldsymbol{q}(0) = (Q_0^{(0)}, \dots, Q_0^{(d-1)})^T$. The general discrete form of the solution of the linear time-varying differential equation (3.39) is (Sarkka, 2006)

$$\boldsymbol{q}(s_{i+1}) = \phi(s_{i+1}, s_i)\boldsymbol{q}(s_i) + \int_{s_i}^{s_{i+1}} \phi(s_{i+1},\tau)\mathbf{v}(s_i)d\tau \tag{3.40}$$

where $\phi(s,t)$ is the transition function of equation (3.38) defined as:

$$\begin{cases} \phi(s,s) & = I \\ \phi'(s,t) & = (-A_{\boldsymbol{\theta}}(s) \otimes \mathbf{I}_d)\phi(s,t) \\ \phi(s,t) & = \phi(s,r)\phi(r,t). \end{cases} \tag{3.41}$$

We denote $u_i = s_{i+1} - s_i$, $v_i = (s_{i+1} + s_i)/2$ and consider the mid-point approximation of the transition matrix function $\phi(s_{i+1}, s_i) \approx \exp(u_i(-A_{\boldsymbol{\theta}}(v_i) \otimes \mathbf{I}_d))$. Thus, we obtain the approximated discrete solution

$$\boldsymbol{q}_{i+1} = \exp(u_i(-A_{\boldsymbol{\theta}}(v_i) \otimes \mathbf{I}_d))\boldsymbol{q}_i + \int_{s_i}^{s_{i+1}} \exp((s_{i+1} - \tau)(-A_{\boldsymbol{\theta}}((s_{i+1} + \tau)/2) \otimes \mathbf{I}_d))\mathbf{v}_i d\tau \tag{3.42}$$

$$= \exp(u_i(-A_{\boldsymbol{\theta}}(v_i) \otimes \mathbf{I}_d))\boldsymbol{q}_i + u_i f(u_i(-A_{\boldsymbol{\theta}}(v_i) \otimes \mathbf{I}_d))\mathbf{v}_i \tag{3.43}$$

where the function $f(\cdot)$ of any $\mathbb{R}^{d^2 \times d^2}$ matrix A can be computed remarking that (Higham, 2008)

$$\exp\left(\begin{pmatrix} A & \mathbf{I}_d \\ 0 & 0 \end{pmatrix}\right) = \begin{pmatrix} \exp(A) & f(A) \\ 0 & \mathbf{I}_d \end{pmatrix}. \tag{3.44}$$

The objective of smoothing is then to find a sequence of controls vectors $\{\mathbf{v}_i\}_{i=1,\dots,n-1}$ in $\mathbb{R}^{d^2}$ such that

$$\begin{cases} \min_{(\mathbf{v}_i)_{i=1,\dots,n-1}} \sum_{i=1}^{n} \|\mathbf{u}_i - \boldsymbol{q}_i\|^2 + \lambda_{\boldsymbol{q}} \sum_{i=1}^{n-1} u_i \|\mathbf{v}_i\|^2 \\ \boldsymbol{q}_{i+1} = \exp(u_i(-A_{\boldsymbol{\theta}}(v_i) \otimes \mathbf{I}_d))\boldsymbol{q}_i + u_i f(u_i(-A_{\boldsymbol{\theta}}(v_i) \otimes \mathbf{I}_d))\mathbf{v}_i \end{cases} \tag{3.45}$$

where $\lambda_{\boldsymbol{q}} > 0$ is a hyperparameter that controls the amounts of smoothing. In the following we will denote $\mathbf{A}_i = \exp(u_i(-A_{\boldsymbol{\theta}}(v_i) \otimes \mathbf{I}_d))$ and $\mathbf{B}_i = u_i f(u_i(-A_{\boldsymbol{\theta}}(v_i) \otimes \mathbf{I}_d))$.

Theorem 3.3. *For any $\boldsymbol{\theta}, \lambda_{\boldsymbol{q}}$, there exists a unique optimal control sequence $(\mathbf{v}_i)_{i=1,\dots,n-1}$ solution of the minimization problem (3.45), and its exact form is given in the following proof.*

Proof. The problem (3.45) can be expressed as a classic Linear Quadratic (LQ) regulator problem. For that, we introduce the following extended state $\tilde{\boldsymbol{q}}_i \in \mathbb{R}^{d^2+1}$, the extended ODE matrices $\tilde{\mathbf{A}}_i$ and $\tilde{\mathbf{B}}_i$, and the observations matrix $\boldsymbol{Y}_i$:

$$\tilde{\boldsymbol{q}}_i = \begin{pmatrix} \boldsymbol{q}_i \\ 1 \end{pmatrix}, \quad \tilde{\mathbf{A}}_i = \begin{pmatrix} \mathbf{A}_i & 0 \\ 0 & 1 \end{pmatrix}, \quad \tilde{\mathbf{B}}_i = \begin{pmatrix} \mathbf{B}_i \\ 0 \end{pmatrix}, \quad \boldsymbol{Y}_i = \begin{pmatrix} \mathbf{I}_{d^2} & -\mathbf{u}_i \\ -\mathbf{u}_i^T & \|\mathbf{u}_i\|^2 \end{pmatrix}. \tag{3.46}$$

Solving the problem 3.45 is then equivalent to solving the LQ problem

$$\begin{cases} \min_{(\mathbf{v}_i)_{i=1,\dots,n-1}} \tilde{\boldsymbol{q}}_n^T \mathbf{Y}_n \tilde{\boldsymbol{q}}_n + \sum_{i=1}^{n-1} \left\{ \tilde{\boldsymbol{q}}_i^T \mathbf{Y}_i \tilde{\boldsymbol{q}}_i + \mathbf{v}_i^T (\lambda_{\boldsymbol{q}} u_i \mathbf{I}_{d^2}) \mathbf{v}_i \right\} \\ \tilde{\boldsymbol{q}}_{i+1} = \tilde{\mathbf{A}}_i \tilde{\boldsymbol{q}}_i + \tilde{\mathbf{B}}_i \mathbf{v}_i. \end{cases} \tag{3.47}$$

Then, as a result of the existence and uniqueness theorem for the solution to the classic LQ regulator (Section 3.3, Theorem 3.2), the optimal sequence $(\mathbf{v}_i)_{i=1,\dots,n-1}$ is given by

$$\mathbf{v}_i = -F_i \tilde{\boldsymbol{q}}_i \tag{3.48}$$

$$F_i = (\lambda_{\boldsymbol{q}} u_i \mathbf{I}_{d^2} + \tilde{\mathbf{B}}_i^T P_{i+1} \tilde{\mathbf{B}}_i)^{-1} \tilde{\mathbf{B}}_i^T P_{i+1} \tilde{\mathbf{A}}_i \tag{3.49}$$

where P_i is solution of the dynamical Riccati equation integrated backward from $P_n = \mathbf{Y}_n$,

$$P_i = \tilde{\mathbf{A}}_i^T P_{i+1} \tilde{\mathbf{A}}_i + \tilde{\mathbf{A}}_i^T P_{i+1} \tilde{\mathbf{B}}_i (\lambda_{\boldsymbol{q}} u_i \mathbf{I}_{d^2} + \tilde{\mathbf{B}}_i^T P_{i+1} \tilde{\mathbf{B}}_i)^{-1} \tilde{\mathbf{B}}_i^T P_{i+1} \tilde{\mathbf{A}}_i + \mathbf{Y}_i. \tag{3.50}$$

■

Finally, the Linear Quadratic theory provides an efficient algorithm to obtain a smooth approximation of the linear Frenet path from noisy observations at the points $\{s_0 \dots, s_n\}$. However, there is no guarantee in this formulation that the estimated linear state $\hat{\boldsymbol{q}}_i$ is composed of d orthonormal vectors. To ensure this constraint, one can project on $SO(d)$ the $d \times d$ matrix reconstructed from the estimated $\hat{\boldsymbol{q}}_i$ by minimizing the Frobenius norm. Then, the nonparametric proxy of the Frenet path at point s_i is finally given as

$$M_{\lambda_{\boldsymbol{q}}}(s_i) = \text{proj}_{SO(d)} \hat{\boldsymbol{q}}_i. \tag{3.51}$$

5 Profiled Estimation of Frenet Curvatures

Given a smooth and reliable estimate of the Frenet path $\hat{Q}$, the estimation of the Frenet curvatures amounts to estimate the time-varying parameters of the corresponding Frenet-Serret differential equation: $\hat{Q}'(s) = \hat{Q}(s)A_{\boldsymbol{\theta}}(s)$. The estimation of the functional and infinite-dimensional parameter $\boldsymbol{\theta}$ implies using appropriate functional data analysis estimation methods (Ramsay and Silverman, 2005). Basically, the parameter estimation is done by minimization of a least squares measure of the fit of $d\hat{Q}(s)/ds$ to $\hat{Q}(s)A_{\boldsymbol{\theta}}(s)$ with respect to $\boldsymbol{\theta}$. In our context, Park and Brunel (2019) adapt the usual least squares criterion to the $SO(d)$-valued data constraint and to the estimation of a time-varying parameter.

5.1 Penalized Least Squares Criterion Definition

We consider the flow of the Frenet-Serret ODE that depends on the Frenet curvatures, denoted $\varphi_{\boldsymbol{\theta}}$. From the self-prediction property of the flow (3.17), the curve $t \mapsto Q(t)$ is a solution to $Q'(s) = Q(s)A_{\boldsymbol{\theta}}(s)$ on $[0, 1]$, if and only if

$$\int_0^1 \int_0^1 d(Q(t), \varphi_{\boldsymbol{\theta}}(t - s, s, Q(s)))dtds = 0 \tag{3.52}$$

where d is a distance on $SO(d)$. Therefore, using the geodesic distance on $SO(d)$ (3.11) and the flow approximation based on the Magnus expansion (3.21), the error can be measured with the following least squares criterion

$$\mathcal{V}(Q, \varphi_{\boldsymbol{\theta}}) = \int_0^1 \int_0^1 \left\| \log \left(Q(t)^T Q(s) \exp \left((t - s) A_{\boldsymbol{\theta}} \left(\frac{t + s}{2} \right) \right) \right) \right\|_F^2 dtds. \tag{3.53}$$

Remark 3.2. *If we assume having only discrete estimations of the Frenet path on a grid $\{s_0, \ldots, s_n\}$, we can define the following discrete criterion instead*

$$\sum_{i=0}^{n-1} \left\| \log \left(Q(s_{i+1})^T Q(s_i) \exp \left((s_{i+1} - s_i) A_{\boldsymbol{\theta}} \left(\frac{s_{i+1} + s_i}{2} \right) \right) \right) \right\|_F^2 \tag{3.54}$$

which uses a simple discrete second-order scheme of the Frenet-Serret differential equation (3.21).

Regarding our problem, two additional constraints must be taken into account in the definition of the criterion in a continuous framework. Firstly, as the flow's approximation is locally valid and to allow for variation in the prediction error, we incorporate weights according to the distance to initial values s in evaluating the solution at t. Secondly, as we consider a non-parametric estimation of the functional parameter $\boldsymbol{\theta}$, we introduce a smooth regularization with the addition of a penalty term. We define then the weighted criterion

$$\mathcal{J}_{h,\lambda}(\boldsymbol{\theta}) = \int_0^1 \int_0^1 K_h(t - s) \left\| \log \left(Q(t)^T Q(s) \exp \left((t - s) A_{\boldsymbol{\theta}} \left(\frac{t + s}{2} \right) \right) \right) \right\|_F^2 dtds + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt \tag{3.55}$$

where $K(\cdot)$ is a kernel function with compact support, e.g. $K(u) = \frac{3}{4}(1 - u)^2 \mathbb{1}_{[-1,1]}(u)$ and $K_h(u) = (1/h)K(u/h)$. The kernel $K(\cdot)$ and the bandwidth h define a prediction horizon for the flow. The regularization parameters λ is a hyperparameter that controls the amounts of smoothing of $\boldsymbol{\theta}$. Based on this statistical criterion, the estimation of $\boldsymbol{\theta}$ given the Frenet path Q is done by solving

$$\hat{\boldsymbol{\theta}}_{h,\lambda} = \arg\min_{\boldsymbol{\theta}} \mathcal{J}_{h,\lambda}(\boldsymbol{\theta}|Q). \tag{3.56}$$

5.2 Frenet Curvatures Estimation Algorithm

In practice, we do not know the exact Frenet path Q, but we are able to estimate a non-parametric proxy $M : [0,1] \to SO(d)$, based on the smoother algorithms detailed in Section 4 and observations $\{U_i\}_{i=1,\dots,n}$. The path M approximately satisfies the self-prediction property of the flow: $M(t) \approx \varphi_{\boldsymbol{\theta}}(t - s_i, s_i, U_i)$. Therefore, with a discretization of the integral on a grid $\{t_0, \dots, t_m\}$, we consider the effective criterion:

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{nm} \sum_{i,j=0}^{n,m} K_h(t_j - s_i) \left\| \log\left(M(t_j)^T U_i \exp\left((t_j - s_i) A_{\boldsymbol{\theta}}\left(\frac{t_j + s_i}{2} \right) \right) \right) \right\|_F^2 + \lambda \int_0^1 \|\theta''(t)\|^2 dt. \quad (3.57)$$

5.2.1 Tractable Criterion

The presence of the exponential makes the actual estimation of the parameter $\boldsymbol{\theta}$ from the criterion (3.57) difficult. Therefore, we consider an additional approximation of the term

$$L_{ij} = \log\left(M(t_j)^T U_i \exp\left((t_j - s_i) A_{\boldsymbol{\theta}}\left(\frac{t_j + s_i}{2} \right) \right) \right). \quad (3.58)$$

We denote $u_{ij} = t_j - s_i$ and $v_{ij} = (t_j + s_i)/2$ and define the matrix $R_{ij} = -\frac{1}{u_{ij}} \log(M(t_j)^T U_i) \in \mathbb{R}^{d \times d}$. Then, using the BCH formula (2.15) we obtain the approximation

$$L_{ij} = \log\left(\exp(-u_{ij} R_{ij}) \exp(u_{ij} A_{\boldsymbol{\theta}}(v_{ij})) \right) = u_{ij}(A_{\boldsymbol{\theta}}(v_{ij}) - R_{ij}) - \frac{u_{ij}^2}{2} [R_{ij}, A_{\boldsymbol{\theta}}(v_{ij})] + O(u_{ij}^3). \quad (3.59)$$

Proposition 3.2. *Given the approximation (3.59) of the term L_{ij}, we have*

$$\frac{1}{nm} \sum_{i,j=0}^{n,m} K_h(u_{ij}) \|L_{ij}\|_F^2 + \lambda \int_0^1 \|\theta''(t)\|^2 dt = \tilde{\mathcal{J}}_{h,\lambda}(\boldsymbol{\theta}; \boldsymbol{R}) + O(h^3)$$

where the tractable criterion $\tilde{\mathcal{J}}_{h,\lambda}(\boldsymbol{\theta}; \boldsymbol{R})$ is defined as

$$\tilde{\mathcal{J}}_{h,\lambda}(\boldsymbol{\theta}; \boldsymbol{R}) = \frac{1}{nm} \sum_{i,j=0}^{n,m} K_h(u_{ij}) u_{ij}^2 \|A_\theta(v_{ij}) - R_{ij}\|_F^2 + \lambda \int_0^1 \|\theta''(t)\|^2 dt\,. \quad (3.60)$$

Proof. From (3.59), we make the following expansion of the norm of the matrices L_{ij} (if $u_{ij}^2 \neq 0$)

$$\begin{aligned} \frac{1}{u_{ij}^2} \|L_{ij}\|_F^2 &= \left\| (A_{\boldsymbol{\theta}}(v_{ij}) - R_{ij}) - \frac{u_{ij}}{2} [R_{ij}, A_{\boldsymbol{\theta}}(v_{ij})] + O(u_{ij}^2) \right\|_F^2 \\ &= \|A_{\boldsymbol{\theta}}(v_{ij}) - R_{ij}\|_F^2 - u_{ij} \operatorname{Tr}\left((A_{\boldsymbol{\theta}}(v_{ij}) - R_{ij})^T [R_{ij}, A_{\boldsymbol{\theta}}(v_{ij})] \right) + O(u_{ij}^2) \end{aligned}$$

Then, using this second-order expansion, the criterion is approximated as

$$\begin{aligned} &\frac{1}{nm} \sum_{i,j=0}^{n,m} K_h(u_{ij}) u_{ij}^2 \|L_{ij}\|_F^2 \\ &= \sum_{i,j=0}^{n,m} \frac{u_{ij}^2}{nm} K_h(u_{ij}) \left\{ \|A_{\boldsymbol{\theta}}(v_{ij}) - R_{ij}\|_F^2 - u_{ij} \operatorname{Tr}\left((A_{\boldsymbol{\theta}}(v_{ij}) - R_{ij})^T [R_{ij}, A_{\boldsymbol{\theta}}(v_{ij})] \right) + O(u_{ij}^2) \right\} \\ &= \sum_{i,j=0}^{n,m} \frac{u_{ij}^2}{nm} K_h(u_{ij}) \|A_{\boldsymbol{\theta}}(v_{ij}) - R_{ij}\|_F^2 - \sum_{i,j=0}^{n,m} \frac{u_{ij}^3}{nm} K_h(u_{ij}) \left\{ \operatorname{Tr}\left((A_{\boldsymbol{\theta}}(v_{ij}) - R_{ij})^T [R_{ij}, A_{\boldsymbol{\theta}}(v_{ij})] \right) + O(u_{ij}) \right\}. \end{aligned}$$

The second term of the right-hand side can be controlled as a $O(h^3)$. Indeed, its behavior can be understood by considering the discretization of the double integral $\iint_{[0,1]^2} (t-s)^3 K_h(t-s) F(t,s) dt ds$, where $(t,s) \mapsto F(t,s)$ is a continuous and bounded matrix function. ∎

5.2.2 Effective Optimization in case $d = 3$

The last derived criterion $\tilde{\mathcal{J}}_{h,\lambda}(\boldsymbol{\theta}; \boldsymbol{R})$ (3.60) can be reformulated, using an elementwise expansion of the Frobenius norm, as $d-1$ independent univariate functional regressions for each of the $d-1$ Frenet curvature. In a three-dimensional space ($d = 3$), the matrix $R_{ij} = -\frac{1}{u_{ij}} \log(M(t_j)^T U_i)$ is an element of the Lie algebra $\mathfrak{so}(3)$ and thus can be written as $R_{ij} = \left[(r^1_{ij}, r^2_{ij}, r^3_{ij})^T\right]^\wedge$. Consequently, in case $d = 3$, we obtain the two independent optimization problems

$$\hat{\kappa}_{h,\lambda_1} = \underset{\kappa \in \mathcal{H}}{\arg\min} \sum_{i,j=0}^{n,m} \omega_{ij} \left(\kappa(v_{ij}) - r^3_{ij}\right)^2 + \lambda_1 \int_0^1 \|\kappa''(t)\|^2 dt \tag{3.61}$$

$$\hat{\tau}_{h,\lambda_2} = \underset{\tau \in \mathcal{H}}{\arg\min} \sum_{i,j=0}^{n,m} \omega_{ij} \left(\tau(v_{ij}) - r^1_{ij}\right)^2 + \lambda_2 \int_0^1 \|\tau''(t)\|^2 dt. \tag{3.62}$$

where the weights are defined as $\omega_{ij} = \frac{1}{nm} K_h(u_{ij}) u^2_{ij}$. It gives rise to the computation of two independent weighted smoothing splines (with splines of third order), defined at the knots v_{ij}, with the pseudo-observations r^1_{ij}, r^3_{ij} (see Section 2.1.1).

5.3 Two-Stage Frenet-Serret ODE Inference Algorithms

In light of the previously discussed methods for obtaining a smooth estimator of the Frenet path, we have introduced two different two-stage procedures for Frenet curvatures estimation. Firstly, if we have noisy observations of the Euclidean curve (3.2), we can employ the non-iterative two-stage estimation Algorithm 1. Alternatively, if the available data consist of noisy observations of the Frenet path (such as experimental data or the outcomes of a noisy preprocessing step on Euclidean curve observations), the smoothers developed in Section 4 give rise to the iterative two-stage Frenet curvatures estimation Algorithm 2.

Algorithm 1 Non-iterative two-stage Frenet curvatures estimation algorithm

Require: Euclidean curve noisy observations $\{\boldsymbol{y}_i\}_{i=0,\dots,n}$ (3.2)

1: Compute the nonparametric estimator of the Frenet path $\hat{Q}$ using a local polynomial regression method defined in Section 4.1,
2: Using the smooth obtained estimator $\hat{Q}$, estimate the Frenet curvatures as

$$\hat{\boldsymbol{\theta}}_{h,\lambda} = \underset{\boldsymbol{\theta} \in \mathcal{H}}{\arg\min}\, \tilde{\mathcal{J}}_{h,\lambda}\left(\boldsymbol{\theta} \,|\, \hat{Q}, \{\hat{Q}(s_i)\}_{i=1,\dots,n}\right).$$

Algorithm 2 Iterative two-stage Frenet curvatures estimation algorithm

Require: Frenet path noisy observations $\{U_i\}_{i=0,\dots,n}$ (3.29), first initial guess of either smooth Frenet path $\hat{M}^{(0)}$ or smooth Frenet curvatures $\hat{\boldsymbol{\theta}}^{(0)}$, and fixed $h > 0$ and $\lambda > 0$ (and eventually $\lambda_{\boldsymbol{q}} > 0$).

1: **while** convergence **do**
2: Given the observations $\{U_i\}_{i=0,\dots,n}$ and the estimator $\hat{\boldsymbol{\theta}}^{(l)}$, compute the nonparametric proxy of the Frenet path $\hat{M}^{(l+1)}$ using one of the model-based smoothers proposed in Section 4.2,
3: Using the smooth obtained proxy $\hat{M}^{(l+1)}$, estimate the Frenet curvatures as

$$\hat{\boldsymbol{\theta}}^{(l+1)}_{h,\lambda} = \underset{\boldsymbol{\theta} \in \mathcal{H}}{\arg\min}\, \tilde{\mathcal{J}}_{h,\lambda}\left(\boldsymbol{\theta} \,|\, \hat{M}^{(l+1)}, \{U_i\}_{i=0,\dots,n}\right).$$

In both algorithms the prediction error depends on $h, \lambda = (\lambda_1, \ldots, \lambda_{d-1})$, and eventually the additional hyperparameter $\lambda_{\boldsymbol{q}}$ required for the tracking-based approach (Section 4.2.2). If the value of h is too large, integrating across the entire interval can lead to error accumulation, making it more advantageous to limit the integration to a smaller interval, like 10% when the total length of the curve is 1 ($h \approx 0.1$). The optimal selection of these hyperparameters is made by performing a 10-fold cross-validation by minimizing the following criterion:

$$\sum_{k=1}^{K} \sum_{i \in T_k} \left\| \log\left(U_i^T \hat{Q}^{-(k)}(s_i; h, \lambda) \right) \right\|_F^2 \tag{3.63}$$

where T_k is the kth index set based on $K = 10$ random partition of the observations $\{U_i\}_{i=1,\ldots,n}$, and $\hat{Q}^{-(k)}(s_i; h, \lambda)$ is the predicted Frenet path reconstructed with parameters $\hat{\theta}^{-(k)}$, estimated without the kth partition dataset, using hyperparameters h, λ and the initial value of Q_0.

6 Simulations Studies

We conduct limited simulation studies in dimension $d = 3$ to assess and compare the performance of the proposed methods. We define a reference shape with the parameter $s \mapsto \boldsymbol{\theta}(s) = (\kappa(s), \tau(s))^T$ such that

$$\begin{cases} \kappa(s) = & 2\cos(2\pi s) + 5 \\ \tau(s) = & 2\sin(2\pi s) + 1 \end{cases} \tag{3.64}$$

and the arc-length function defined from $[0,1]$ to $[0,1]$ as $s : t \mapsto \frac{\exp(at)-1}{\exp(a)-1}$ if $a \neq 0$ and $s : t \mapsto t$ otherwise. We choose $a \sim \mathcal{N}(0,1)$. All simulation models are repeated 100 times. To measure the error between true and estimated components, we use the L^2 norm for Euclidean curves in $\mathbb{R}^3$, the $SO(3)$ geodesic distance for $SO(3)$-valued curves, denoted $\mathbf{d}$ and defined in equation (3.11), and the $\mathbb{L}^2$ relative error norm for scalar functions like Frenet curvatures, denoted $\mathbf{re}$ and defined as

$$\mathbf{re}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \frac{\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_{\mathbb{L}^2}}{\|\boldsymbol{\theta}\|_{\mathbb{L}^2}}. \tag{3.65}$$

6.1 Scenario 1: Algorithm 2 with Frenet path observations

Given the reference shape, $s \mapsto \boldsymbol{\theta}(s)$ and $t \mapsto s(t)$, we generate noisy observations of the Frenet path Q solution of $Q'(s) = Q(s)A_{\boldsymbol{\theta}}(s)$, with assuming known initial condition Q_0, using the model

$$U_i = Q(s_i)R_i, \quad i = 1, \ldots, n, \tag{3.66}$$

where R_i follows a Fisher-Langevin matrix distribution with mean $\mathbf{I}_3$ and concentration α. We recall the density of the Fisher-Langevin matrix distribution with mean M and concentration parameter α (Mardia and Jupp, 1999)

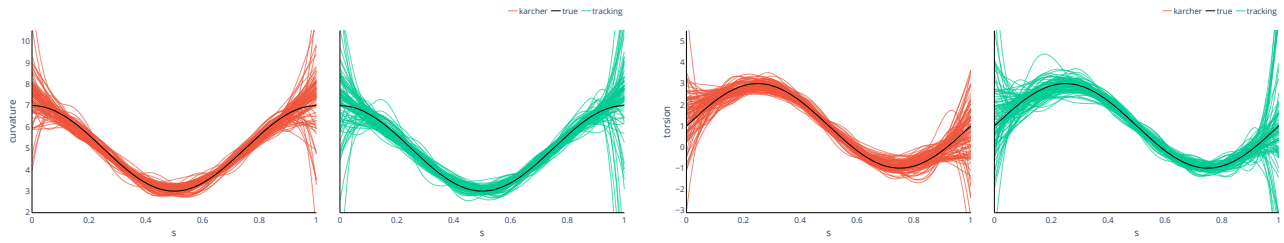
$$f(R; M, \alpha) = \left\{ {}_0F_1\left(\frac{p}{2}; \frac{\alpha^2}{4}\right) \right\}^{-1} \exp(\alpha \operatorname{Tr}(M^T R)).$$

From the obtained noisy observations $\{U_i\}_{i=1,\ldots,n}$, we apply the iterative Algorithm 2 to simultaneously recover the Frenet path Q and estimate the Frenet curvatures $\boldsymbol{\theta}$. We consider different sample sizes $n = 100, 200$ and noise levels $\alpha = 10, 20$. Each of the two model-based smoothing methods (Karcher mean or tracking) is compared for each combination of parameters. Using the cross-validation criterion (3.63), we run a Bayesian optimization of the hyperparameters in $h \in [0.05, 0.35]$, $\lambda_1, \lambda_2 \in [10^{-9}, 10^{-3}]$

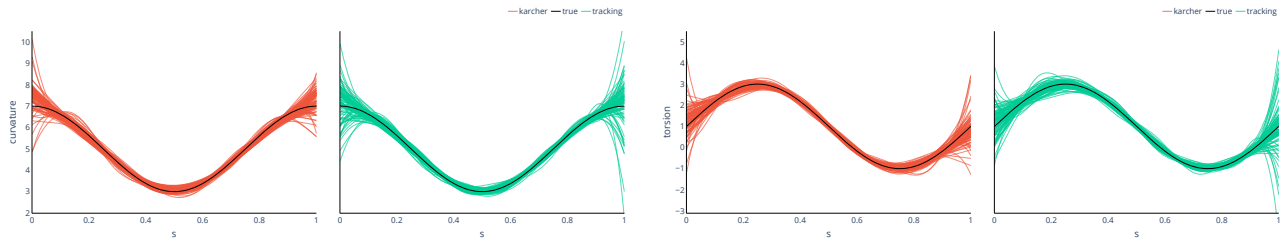
and $\lambda_{\boldsymbol{q}} \in [0.0001, 1]$. We fix the number of basis functions in the B-spline basis expansion of κ and τ to $\lceil \sqrt{n} \rceil = 10, 15$, where n is the sample size. Results are shown in Table 3.1, where, for each simulation, Q is the true Frenet path, U is the simulated noisy Frenet path, $\hat{Q}$ is the smooth estimated Frenet path at the end of the algorithm, $(\hat{\kappa}, \hat{\tau})$ are the estimated curvature and torsion, and $\hat{Q}^{(\hat{\theta})}$ is the reconstructed Frenet path by solving the Frenet-Serret ODE with the estimated Frenet curvatures.

n	α	Smoother	$\mathbf{d}(Q, U)$	$\mathbf{d}(Q, \hat{Q})$	$\mathbf{re}(\kappa, \hat{\kappa})$	$\mathbf{re}(\tau, \hat{\tau})$	$\mathbf{d}(Q, \hat{Q}^{(\hat{\theta})})$
100	10	Karcher	0.113 (0.005)	0.038 (0.011)	**0.069** (0.037)	**0.247** (0.109)	0.043 (0.013)
		Tracking		**0.035** (0.013)	0.088 (0.080)	0.3 (0.209)	**0.037** (0.010)
	20	Karcher	0.056 (0.002)	0.020 (0.006)	**0.038** (0.014)	**0.124** (0.059)	0.022 (0.007)
		Tracking		**0.017** (0.004)	0.045 (0.028)	0.143 (0.090)	**0.019** (0.005)
200	10	Karcher	0.113 (0.003)	0.031 (0.013)	0.061 (0.041)	**0.224** (0.119)	0.039 (0.015)
		Tracking		**0.025** (0.006)	**0.060** (0.058)	0.233 (0.186)	**0.031** (0.008)
	20	Karcher	0.056 (0.002)	0.016 (0.005)	**0.034** (0.018)	0.119 (0.060)	0.019 (0.006)
		Tracking		**0.013** (0.003)	0.036 (0.028)	**0.117** (0.074)	**0.016** (0.005)

Table 3.1: Estimation errors of scenario 1: Algorithm 2 with Frenet path observations. Means over the 100 repetitions of the simulation and standard deviations in parenthesis.



(a) $n = 100, \alpha = 10$



(b) $n = 100, \alpha = 20$

Figure 3.1: Estimated curvature (left) and torsion (right) in scenario 1 (Algorithm 2 with Frenet path observations): comparison between true parameters (black), estimates using Karcher mean smoother (red) or using tracking smoother (green) for two simulation cases.

In summary, these results indicate that the tracking algorithm excels in smoothing noisy Frenet frame observations, consistently yielding smaller geodesic distances between the smoothed Frenet path and the true path compared to the Karcher mean method. The sample size does not significantly impact the performance of both methods, although a slight reduction in error is observed with larger sample sizes. Furthermore, both methods exhibit robustness in the presence of high noise levels, performing even better under such conditions. While the Karcher mean method provides slightly more accurate estimates of Frenet curvatures, the tracking method yields smaller errors when reconstructing Frenet paths using the estimated curvatures. This discrepancy in performance may be attributed to the influence of initial conditions for the reconstruction chosen as $\hat{Q}(0)$. Figure 3.1 shows that Frenet curvatures

estimation results by the two methods are very similar (case $n = 100$). Finally, the tracking approach is approximately 20% faster than the Karcher mean method.

6.2 Scenario 2: Algorithm 1 with Euclidean curve observations

Given the functions $s \mapsto \boldsymbol{\theta}(s)$ and $t \mapsto s(t)$, we consider the observation model

$$\boldsymbol{y}_i = X(s_i) + \boldsymbol{\epsilon}_i, \quad i = 1, \dots, n$$

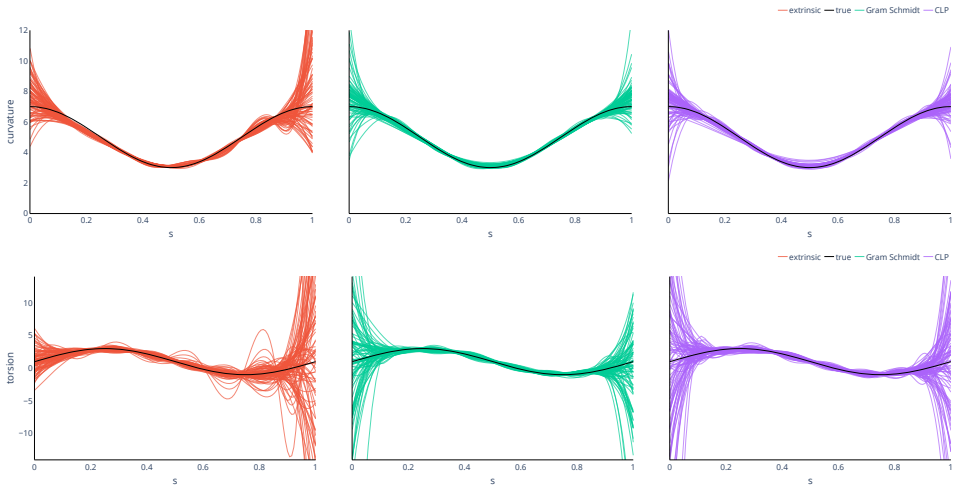
where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \gamma^2 \boldsymbol{I}_3)$ and the shape function $s \mapsto X(s)$ has a Frenet path $Q(s)$ solution of $Q'(s) = Q(s)A_{\boldsymbol{\theta}}(s)$. We consider different sample sizes $n = 100, 200$ and realistic noise levels $\gamma = 0.001, 0.005$. Again, the number of basis functions is fixed to $\lceil \sqrt{n} \rceil = 10, 15$. Then, we compare the following Frenet curvatures estimation procedures:

- **S2.1** Derivatives estimation based on local polynomial regression and use of extrinsic formulas of Frenet curvatures.
- **S2.2** Application of Algorithm 1 from an estimator of the Frenet path $\hat{Q}$ obtained by Gram-Schmidt orthonormalization of estimated derivatives.
- **S2.3** Application of Algorithm 1 from an estimator of the Frenet path $\hat{Q}$ obtained with constrained local polynomial regression.

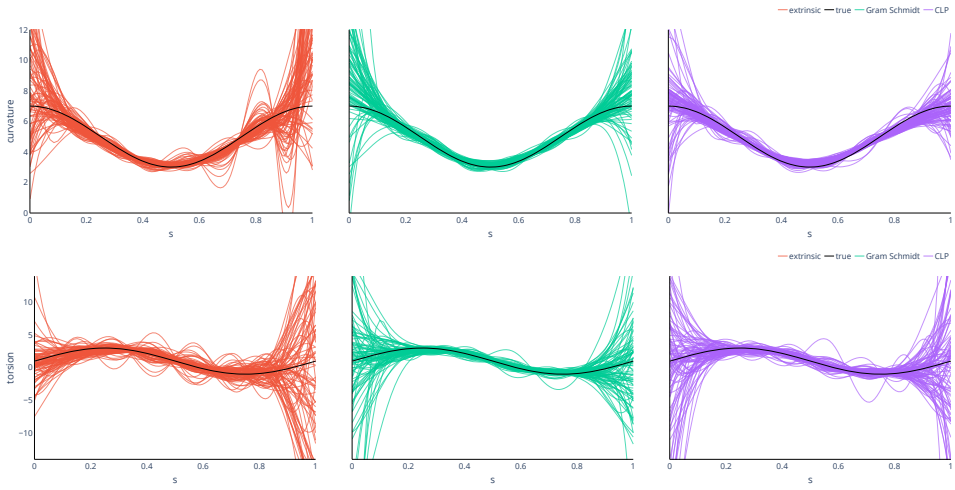
n	γ		$\mathbf{d}(Q, \hat{Q})$	$\|X - \hat{X}\|_2$	$\mathbf{re}(\kappa, \hat{\kappa})$	$\mathbf{re}(\tau, \hat{\tau})$	$\mathbf{d}(Q, \hat{Q}^{(\hat{\theta})})$	$\|X - \hat{X}^{(\hat{\theta})}\|_2$
100	0.001	**S2.1**	-	0.007 (0.001)	0.118 (0.078)	1.807 (1.824)	0.125 (0.050)	0.392 (0.206)
		S2.2	**0.034** (0.014)	0.007 (0.001)	0.053 (0.032)	**0.943** (0.923)	**0.045** (0.020)	**0.075** (0.044)
		S2.3	0.040 (0.016)	**0.006** (0.001)	**0.049** (0.029)	1.353 (1.148)	0.050 (0.024)	0.087 (0.067)
	0.005	**S2.1**	-	0.028 (0.004)	0.359 (0.312)	1.934 (1.236)	0.276 (0.118)	0.867 (0.473)
		S2.2	**0.085** (0.049)	0.028 (0.004)	0.148 (0.129)	**1.305** (1.136)	0.107 (0.049)	0.184 (0.119)
		S2.3	0.088 (0.038)	**0.026** (0.004)	**0.093** (0.053)	1.555 (1.221)	**0.101** (0.034)	**0.161** (0.069)
200	0.001	**S2.1**	-	0.007 (0.001)	0.127 (0.099)	1.740 (1.516)	0.119 (0.081)	0.559 (0.454)
		S2.2	**0.030** (0.010)	0.007 (0.001)	0.052 (0.031)	**1.144** (1.104)	**0.045** (0.020)	**0.122** (0.093)
		S2.3	0.033 (0.009)	**0.006** (0.001)	**0.044** (0.027)	1.445 (1.101)	0.047 (0.021)	0.141 (0.115)
	0.005	**S2.1**	-	0.029 (0.004)	0.359 (0.291)	2.334 (1.631)	0.272 (0.197)	1.235 (1.091)
		S2.2	**0.070** (0.033)	0.029 (0.004)	0.157 (0.141)	**1.567** (1.989)	0.118 (0.204)	0.344 (0.903)
		S2.3	0.075 (0.035)	**0.028** (0.004)	**0.080** (0.040)	1.777 (1.994)	**0.088** (0.039)	**0.204** (0.121)

Table 3.2: Estimation errors of scenario 2: Algorithm 1 with Euclidean curve observations. Means over the 100 repetitions of the simulation and standard deviations in parenthesis.

Results are displayed in Table 3.2 and Figure 3.2. In summary, they reveal that the Gram-Schmidt method outperforms the constrained local polynomial regression in estimating the Frenet path, although both methods yield fairly similar results in terms of estimating Frenet curvatures. Conversely, the extrinsic formulas method demonstrates significantly less effectiveness, as evidenced by the significant errors observed in both the estimated curvatures ($\hat{\theta}$) and the corresponding Frenet frames ($\hat{Q}^{(\hat{\theta})}$). In addition, the results highlight the cumulative effect of errors: the combination of preprocessing errors in Q and errors in Frenet curvatures estimation leads to a higher overall error in the reconstructed Frenet path compared to the error between the true and preprocessed Q. This suggests a potential inconsistency in the final estimate with the true observations of the Euclidean curve, or at least an increasing error stemming from preprocessing.



(a) $n = 100, \gamma = 0.001$



(b) $n = 100, \gamma = 0.005$

Figure 3.2: Estimated curvature (first row) and torsion (second row) in scenario 2 (Algorithm 1 with Euclidean curve observations): comparison between true parameters (black), estimates using extrinsic formulas **S2.1** (red), method **S2.2** (green) and method **S2.3** (purple) for two simulation cases.

6.3 Scenario 3: Algorithm 2 with Euclidean curve observations

In this last scenario, we test the performance of the Algorithm 2 starting from noisy observations of a Euclidean curve as in the previous scenario. In this case, to apply the algorithm, we need to go through a data pre-processing stage to obtain pseudo-observations of the Frenet path. For that, as in the previous scenario, we can use either (**S3.1**) the Gram-Schmidt orthonormalization of estimated derivatives (Section 2.1) or (**S3.2**) the constrained local polynomial regression (Section 4.1). From the obtained Frenet path pseudo-observations $\{\hat{Q}^{(\text{init})}(s_1), \ldots, \hat{Q}^{(\text{init})}(s_n)\}$, we apply the iterative Algorithm 2 to simultaneously recover the Frenet path $\hat{Q}^{(\text{smooth})}$ and estimate the Frenet curvatures $\hat{\boldsymbol{\theta}}$. We test each of the two pre-processing methods with each of the two smoother.

Results obtained using the Karcher mean method for smoothing of Frenet path noisy observations are printed in Table 3.3 and the ones obtained using the tracking method in Table 3.4. Plots of the estimated Frenet curvatures are available on Figure 3.3 for the case $n = 100$. Since, in this scenario, the

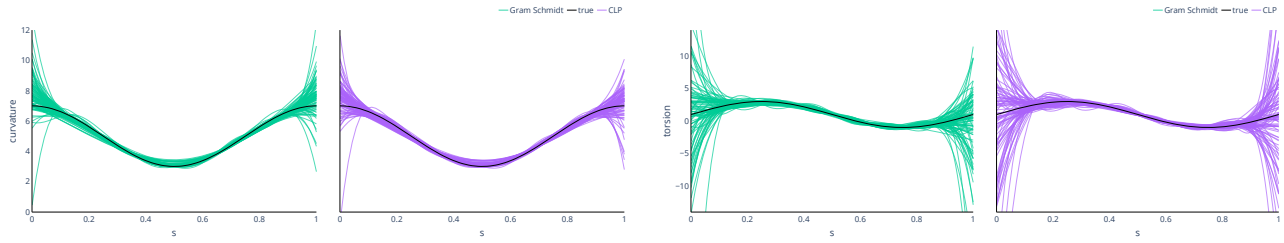
n	γ		$\mathbf{d}(Q, \hat{Q}^{(\text{init})})$	$\mathbf{d}(Q, \hat{Q}^{(\text{smooth})})$	$\mathbf{re}(\kappa, \hat{\kappa})$	$\mathbf{re}(\tau, \hat{\tau})$	$\mathbf{d}(Q, \hat{Q}^{(\hat{\theta})})$	$\mathbf{d}(\hat{Q}^{(\text{init})}, \hat{Q}^{(\hat{\theta})})$
100	0.001	**S3.1**	**0.034** (0.014)	**0.041** (0.016)	0.061 (0.036)	**0.836** (0.849)	**0.046** (0.020)	0.019 (0.010)
		S3.2	0.040 (0.016)	0.046 (0.016)	**0.055** (0.031)	1.134 (0.929)	0.051 (0.022)	0.019 (0.013)
	0.005	**S3.1**	**0.085** (0.049)	**0.086** (0.051)	0.165 (0.100)	**1.238** (1.022)	0.096 (0.101)	0.033 (0.048)
		S3.2	0.088 (0.038)	0.087 (0.029)	**0.094** (0.043)	1.431 (1.088)	**0.090** (0.033)	**0.022** (0.014)
200	0.001	**S3.1**	**0.030** (0.010)	**0.037** (0.012)	0.052 (0.028)	**1.087** (1.077)	**0.042** (0.016)	**0.016** (0.011)
		S3.2	0.033 (0.009)	0.040 (0.012)	**0.045** (0.026)	1.323 (1.037)	0.045 (0.017)	0.017 (0.011)
	0.005	**S3.1**	**0.070** (0.033)	0.079 (0.088)	0.166 (0.152)	**1.449** (1.768)	0.104 (0.214)	0.043 (0.104)
		S3.2	0.075 (0.035)	**0.077** (0.033)	**0.085** (0.063)	1.642 (1.694)	**0.086** (0.082)	**0.029** (0.068)

Table 3.3: Estimation errors of scenario 3 with **Karcher mean smoother**: Algorithm 2 with Euclidean curve observations. Means over the 100 repetitions of the simulation and standard deviations in parenthesis.

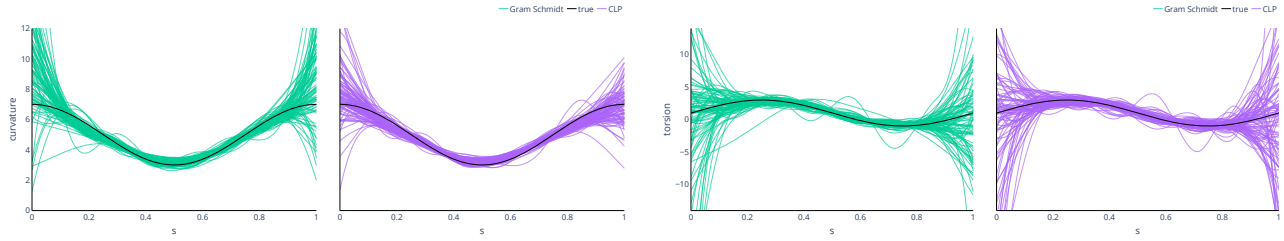
n	γ		$\mathbf{d}(Q, \hat{Q}^{(\text{init})})$	$\mathbf{d}(Q, \hat{Q}^{(\text{smooth})})$	$\mathbf{re}(\kappa, \hat{\kappa})$	$\mathbf{re}(\tau, \hat{\tau})$	$\mathbf{d}(Q, \hat{Q}^{(\hat{\theta})})$	$\mathbf{d}(\hat{Q}^{(\text{init})}, \hat{Q}^{(\hat{\theta})})$
100	0.001	**S3.1**	**0.034** (0.014)	**0.040** (0.015)	0.072 (0.044)	**1.016** (1.023)	**0.043** (0.020)	**0.009** (0.012)
		S3.2	0.040 (0.016)	0.045 (0.019)	**0.065** (0.044)	1.358 (1.188)	0.051 (0.024)	0.014 (0.016)
	0.005	**S3.1**	**0.085** (0.049)	**0.082** (0.037)	0.193 (0.146)	**1.478** (1.605)	**0.084** (0.040)	0.012 (0.025)
		S3.2	0.088 (0.038)	0.086 (0.033)	**0.110** (0.074)	1.685 (1.606)	0.089 (0.033)	0.012 (0.012)
200	0.001	**S3.1**	**0.030** (0.010)	**0.037** (0.015)	0.056 (0.037)	**1.054** (1.030)	**0.042** (0.020)	**0.014** (0.013)
		S3.2	0.033 (0.009)	0.038 (0.013)	**0.049** (0.046)	1.324 (1.148)	0.046 (0.020)	0.021 (0.015)
	0.005	**S3.1**	**0.070** (0.033)	**0.073** (0.040)	0.189 (0.255)	1.787 (2.609)	0.080 (0.052)	0.023 (0.031)
		S3.2	0.075 (0.035)	0.075 (0.035)	**0.098** (0.080)	**1.544** (1.410)	0.080 (0.040)	**0.021** (0.017)

Table 3.4: Estimation errors of scenario 3 with **tracking smoother**: Algorithm 2 with Euclidean curve observations. Means over the 100 repetitions of the simulation and standard deviations in parenthesis.

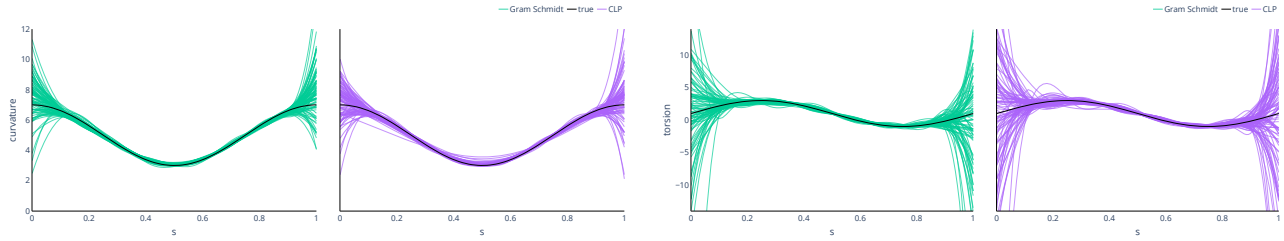
data are noisy observations of the Euclidean curve and not of the Frenet path, a preprocessing step is used to compute $\hat{Q}^{(\text{init})}$. Then, $\hat{Q}^{(\text{smooth})}$ is the smooth Frenet path after iterations and $(\hat{\kappa}, \hat{\tau})$ the estimated Frenet curvatures after iterations. The results align with those observed in scenario 1, confirming that the tracking method generally outperforms the Karcher mean method in Frenet path smoothing. However, when compared to scenario 2, we observe that the iterative algorithm yields improved results in terms of Frenet curvatures estimates, especially when employing the Karcher mean method. This highlights the importance of using an iterative algorithm instead of solely relying on the Frenet path preprocessing step when working with observed Euclidean curve data. Nonetheless, as already observed in the previous scenarios, the algorithm minimizes errors concerning the data $\hat{Q}^{(\text{init})}$, which may contain preprocessing errors, in contrast to the actual observed data $\boldsymbol{y}_i$. Similarly, the smoothing of $\hat{Q}^{(\text{init})}$ into $\hat{Q}^{(\text{smooth})}$ tends to increase the error concerning the true parameter Q, primarily due to the presence of additional errors in $\hat{Q}^{(\text{init})}$.



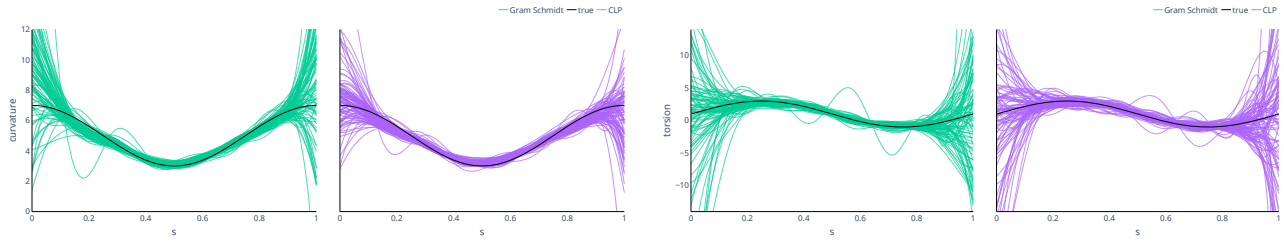
(a) Karcher mean smoother with $n = 100, \gamma = 0.001$



(b) Karcher mean smoother with $n = 100, \gamma = 0.005$



(c) Tracking smoother with $n = 100, \gamma = 0.001$



(d) Tracking smoother with $n = 100, \gamma = 0.005$

Figure 3.3: Estimated curvature (left) and torsion (right) in scenario 3 (Algorithm 2 with Euclidean curve observations): comparison between true parameters (black), estimates using method **S3.1** (green) and method **S3.2** (purple) for two simulation cases and with the two different smoothers.

7 Conclusion and Limitations

In this chapter, we mainly present the method proposed by Park and Brunel (2019) for estimating Frenet curvatures from noisy observations of a single Euclidean curve. Viewing this problem as an ordinary differential equation inference task within a Lie group, they introduce an iterative algorithm involving state variable smoothing and optimization of the differential equation parameters. By formulating a local polynomial regression method within the $SO(d)$ Lie group and minimizing the least squares error with the Frenet-Serret ODE flow, they derive an efficient and reliable algorithm that significantly outperforms classical methods based on extrinsic formulas. In an effort to enhance the state variable estimation step, which is the Frenet path, we propose an alternative tracking-based approach. This approach proves to be faster and more efficient but does not fully address the major issues of the method. Indeed, as highlighted in our simulations, a significant portion of the error in the final estimates of Frenet curvatures arises from the preprocessing step required in this problem formulation. Since the method requires observations of the Frenet frame, it must be estimated in an initial step, introducing a non-negligible error and potentially leading to estimators inconsistent with real observations. However, this new approach suggests the idea of rewriting the problem in the form of an expectation-maximization algorithm, which is the subject of the next chapter.

4

Functional EM Algorithm on SE(3) for Frenet Curvatures Estimation

Abstract

We aim to develop a statistical estimation method for Frenet curvatures of a Euclidean curve in $\mathbb{R}^3$ directly from its discrete and possibly noisy observations. This problem corresponds to real-world scenarios where such observations are the only available data. Existing methods for estimating Frenet curvatures rely on latent variable observations (derivatives or Frenet paths). Thus, the preprocessing step needed in this case to estimate these variables introduces inherent errors in the "pseudo" observations used for curvatures parameters estimation. The goal of this work is to establish a statistical framework addressing the estimation of latent variables and parameters in a unified manner. By expressing the Frenet-Serret equation and trajectory observations as a state-space model in the Lie group $SE(3)$, we propose an expectation-maximization algorithm for this purpose. The expectation step employs an invariant extended Kalman filter on $SE(3)$. We then solve the problem of determining parameters by formulating it as a nonparametric function estimation problem using penalized expected log-likelihood maximization. We demonstrate the improved accuracy of the estimators obtained with this approach through experiments on synthetic data. Finally, we compare the proposed approach with existing methods or methods developed earlier in this thesis for estimating Frenet curvatures on real data corresponding to motion trajectories.

Contents

1 Introduction

As introduced in Chapter 3, our objective is to estimate the continuous function of Frenet curvatures from discrete and noisy observations of a Euclidean curve. So far, this problem has been tackled using two main approaches: either relying on extrinsic formulas of Frenet curvatures (Kim et al., 2013; Sangalli et al., 2009; Lewiner et al., 2005) or assuming that the Frenet path can be accurately estimated from these observations through a pre-processing step (Park and Brunel, 2019). Although the latter approach improves results compared to extrinsic formulas, the pre-processing step to obtain "pseudo" observations of the Frenet path from real Euclidean curve observations poses a major limitation to these methods. In this case, the estimation algorithms of the Frenet curvatures seek to minimize errors based on these "pseudo" Frenet path data. Consequently, if the pre-processing error is significant, the estimator of the underlying parameter $\boldsymbol{\theta}$ may not correspond to the initial and real Euclidean curve observations. This chapter addresses this inconsistency issue by redefining the problem in a unified framework that allows for the estimation of Frenet curvatures, still as an ODE parameter estimation, but with a direct tracking of the error with respect to the actually available Euclidean data.

To effectively unify the different estimation steps of the existing algorithms, we need to model the link between the discrete and noisy Euclidean observations of the curve X, detoned by $\{\boldsymbol{y}_i\}_{i=1,\dots,n}$, and the vector-valued functional parameter of Frenet curvatures $\boldsymbol{\theta}$. These two entities are connected by the observation model (3.2) and by the two Frenet-Serret differential equations, one on the curve X in $\mathbb{R}^d$ ($\frac{dX(s)}{ds} = T(s)$) and the other on the Frenet path Q in $SO(d)$ (2.3) due to the orthogonality constraints. As they are either imperfectly or not observed at all, the variables $(X(s), Q(s))$ are hidden variables of the model. Using the idea from Pilté et al. (2017), we can embed these hidden variables $(X(s), Q(s))$ into a single entity defined on $SE(d)$, see (4.1) for the formal definition. Subsequently, the model is described by an observation model and a single differential equation on $SE(d)$ parameterized by the Frenet curvatures. We aim to estimate the infinite-dimensional parameter $\boldsymbol{\theta}$ of this latent variable differential equation, but we only have incomplete observations of the solution and lack any prior information on the functional parameter $\boldsymbol{\theta}$. Hence, we need to account for potential model misspecification arising from this structural uncertainty and the limited information in the available data (Brynjarsdóttir and O'Hagan, 2014; Clairon and Brunel, 2019). To address this, we introduce stochasticity to capture the uncertainty by perturbing the latent variable equation into a stochastic differential equation on the $SE(d)$ group. This finally defines a state space model having a propagation equation defined on a Lie group parameterized by the infinite-dimensional Frenet curvatures parameter and a Euclidean observation model. Therefore, our modeling differs from that proposed by Pilté et al. (2017) for tracking the Frenet-Serret frame, where

they assume nearly constant Frenet curvatures and velocity. In contrast to their formulation, as our primary focus lies on Frenet curvatures estimation, we consider a functional data modeling framework to estimate them as smooth functions.

In general, inference and learning of a state-space model is done with the well-known Expectation-Maximization (EM) algorithm (Dempster et al., 1977; Meng and van Dyk, 1997; Dwivedi et al., 2018). This algorithm iterates between an Expectation step (E-step), which infers the values of the hidden states based on the current parameter estimates and observed data, and a Maximization step (M-step), which maximizes the expected log-likelihood to update the parameter estimates. However, the classical framework for using the EM algorithm differs from the one defined by our model in two main aspects.

Firstly, the EM algorithm is typically employed to infer a Euclidean latent process, often assumed to follow a Gaussian distribution. Under these circumstances, inference is usually performed using a classic Kalman filter (KF) (Kalman, 1960) or an extended Kalman filter (EKF) in the case of a nonlinear propagation equation (Baras et al., 1988). Subsequently, one might consider the straightforward idea of firstly linearizing our $SE(d)$ latent process and its associated stochastic equation into a Euclidean framework, as proposed in Chapter 3. However, this approach leads to an ill-defined model that does not guarantee the estimated state to be a valid element of the Lie group. To directly incorporate the inherent geometry of the problem in the inference step, we use instead the Invariant Extended Kalman Filter (IEKF) defined by Bonnabel (2007). This generalization of the EKF for cases where the state evolves on a connected unimodular matrix Lie group is derived under different settings in several papers (Barrau and Bonnabel, 2015, 2017, 2018; Bourmaud et al., 2013, 2015; Zhu et al., 2022; Pilté et al., 2017). It typically approximates the Lie group latent process as a concentrated Gaussian process on the Lie group and allows for tracking and smoothing of the hidden variables directly in the Lie group.

Secondly, the classical EM framework is typically used to estimate low-dimensional parameters in discrete-time settings. In our model, the parameters to estimate include the vector-valued function of Frenet curvatures $\boldsymbol{\theta}$, resulting in an infinite-dimensional parameter that requires additional constraints for reliable estimation. In a continuous-time setting, the EM algorithm has been used in the work by Wang and Weinan (2021) for continuous-time Hidden Markov Models inference or by Liu et al. (2009) for a mixing density functional parameter estimation using a nonparametric maximum penalized likelihood method. The penalized likelihood is a statistical approach commonly used for functional parameter estimation that introduces a regularization term in the likelihood function to handle the complexity and overfitting issues that may arise in functional parameter estimation and leads to more stable and well-behaved estimates (Cardot and Sarda, 2005; Du and Wang, 2014; Cole et al., 2013).

Hence, we consider in this work the general problem of the continuous-discrete state space model inference and learning in the case of a propagation equation defined on a Lie group and parameterized by a functional parameter, with an incomplete Euclidean observation model. This chapter focuses on the specific case of estimating Frenet curvatures in a space of dimension 3. We then propose a general framework and an efficient algorithm, as a variant of the EM algorithm, for inferring the values of the Frenet path and estimating the Frenet curvatures function. The proposed functional EM algorithm on $SE(3)$ utilizes the IEKF for filtering and smoothing of the Frenet path and the Euclidean curve, and compute the maximum penalized likelihood estimate of $\boldsymbol{\theta}$ over the function space $\mathcal{H}$ (defined in Section 3 of Chapter 2) using a B-spline basis decomposition method (see Section 2.1.1 of Chapter 2). This unified method is evaluated using experiments on synthetic data. In addition, we apply the proposed algorithm to estimate the curvature and torsion functions of real human motion trajectories of the wrist, acquired with a motion capture system, and compare with methods that are available and/or described in Chapter 3.

2 Preliminaries

2.1 The Special Euclidean Group

The *special Euclidean group* is a particular non-abelian matrix Lie group that represents the set of rigid transformations, i.e., all possible transformations that preserve distances and angles, which are translations and rotations. We refer to Section 1.3.1 of Chapter 2 for an introduction to matrix Lie group and Lie algebra and their main properties and to Section 3.1 of Chapter 3 for the preliminaries on the Special Orthogonal group $SO(d)$, which represents the rotations. The special Euclidean group corresponds to the set of valid transformation matrices, sometimes called *poses*, defined for $d \geq 2$ as

$$SE(d) = \left\{ T = \begin{pmatrix} R & X \\ \mathbf{0}^T & 1 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \;\middle|\; R \in SO(d), X \in \mathbb{R}^d \right\}. \tag{4.1}$$

In $SE(d)$, the combination and inverse operations correspond to

$$T_1 T_2 = \begin{pmatrix} R_1 & X_1 \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} R_2 & X_2 \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} R_1 R_2 & R_1 X_2 + X_1 \\ \mathbf{0}^T & 1 \end{pmatrix} \in SE(d) \qquad \textit{(combination)} \tag{4.2}$$

$$T^{-1} = \begin{pmatrix} R^T & -R^T X \\ \mathbf{0}^T & 1 \end{pmatrix} \in SE(d) \qquad \textit{(inversion)} \tag{4.3}$$

The Lie algebra of the special Euclidean group $SE(d)$ is the vector space of infinitesimal transformations of dimension $\frac{d(d+1)}{2}$ denoted by $\mathfrak{se}(d)$ and defined as

$$\mathfrak{se}(d) = \left\{ \boldsymbol{\xi}^\wedge = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{\rho} \end{pmatrix}^\wedge = \begin{pmatrix} \boldsymbol{a}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \;\middle|\; \boldsymbol{\xi} \in \mathbb{R}^{d(d+1)/2}, \boldsymbol{a} \in \mathbb{R}^{d(d-1)/2}, \boldsymbol{\rho} \in \mathbb{R}^d \right\}. \tag{4.4}$$

Lemma 4.1. *(Duan et al., 2013) The geodesic distance between two points T_1, T_2 on $SE(d)$, having as rotation and translation respectively (R_1, X_1) and (R_2, X_2), induced by the scale-dependent left invariant metric (2.5) is given by*

$$d(T_1, T_2) = \left(\| \log(R_1^T R_2) \|_F^2 + \| X_2 - X_1 \|_2^2 \right)^{1/2}. \tag{4.5}$$

Three-dimensional case: In this chapter, we focus on deriving our estimation algorithm specifically for the case $d = 3$. This choice is motivated by the specific properties of the Special Euclidean group in a three-dimensional space. The exponential and logarithm maps have particular representations in $SE(3)$. For $T = \begin{pmatrix} R & X \\ \mathbf{0}^T & 1 \end{pmatrix} \in SE(3)$, there is $\boldsymbol{\xi} = (\boldsymbol{a}, \boldsymbol{\rho})^T \in \mathbb{R}^6$, with $\boldsymbol{a}, \boldsymbol{\rho} \in \mathbb{R}^3$, such that

$$T = \exp(\boldsymbol{\xi}^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} \begin{pmatrix} \boldsymbol{a}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{pmatrix}^n = \begin{pmatrix} \exp(\boldsymbol{a}^\wedge) & J_{SO(3)}(\boldsymbol{a})\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{pmatrix}, \tag{4.6}$$

$$\boldsymbol{\xi}^\wedge = \log(T) = \log\left(\begin{pmatrix} R & X \\ \mathbf{0}^T & 1 \end{pmatrix} \right) = \begin{pmatrix} \log(R)^\vee & J_{SO(3)}^{-1}(\log(R)^\vee) X \\ \mathbf{0}^T & 0 \end{pmatrix}, \tag{4.7}$$

where $J_{SO(3)}$ and $J^{-1}_{SO(3)}$ are the Jacobian and inverse Jacobian matrix of $SO(3)$ defined in (3.14) and (3.15). The adjoint representations on $SE(3)$ and $\mathfrak{se}(3)$ are given by

$$\mathcal{T} = Ad_{SE(3)}(T) = \begin{pmatrix} R & 0_{3\times 3} \\ X^\wedge R & R \end{pmatrix} \in \mathbb{R}^{6\times 6}, \tag{4.8}$$

$$\boldsymbol{\xi}^\curlywedge = ad_{SE(3)}(\boldsymbol{\xi}^\wedge) = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{\rho} \end{pmatrix}^\curlywedge = \begin{pmatrix} \boldsymbol{a}^\wedge & 0_{3\times 3} \\ \boldsymbol{\rho}^\wedge & \boldsymbol{a}^\wedge \end{pmatrix} \in \mathbb{R}^{6\times 6}, \tag{4.9}$$

and we have $\mathcal{T}^{-1} = Ad_{SE(3)}(T^{-1})$.

Barfoot (2017) gives a detailed and complete description of the two Lie groups $SO(3)$ and $SE(3)$ (Chapter 7), and we refer to it for additional properties relating to them.

2.2 Concentrated Gaussian Distribution on Lie Groups

In this section, we introduce the concept of *concentrated Gaussian distribution on Lie groups*, a generalization to Lie groups of the Gaussian distribution in Euclidean space (Wang and Chirikjian, 2006, 2008). Let G be a connected unimodular (i.e. $\forall g \in G, |\det Ad_G(g)| = 1$) matrix Lie group and $\mathfrak{g}$ its corresponding Lie algebra. The idea for defining a concentrated Gaussian random variable T on G is to consider the following parametrization:

$$T = M \exp\left(-\boldsymbol{\xi}^\wedge\right), \tag{4.10}$$

where $M \in G$ is a "large" noise-free mean value, and $\boldsymbol{\xi}^\wedge \in \mathfrak{g}$ is a "small" noisy perturbation. Defining a Gaussian density on the vector space $\mathfrak{g}$ will induce a probability density on G. We define the small perturbation variable $\boldsymbol{\xi}$ as zero-mean multivariate Gaussian random variable on $\mathbb{R}^p$, $p(\boldsymbol{\xi}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$, where $\boldsymbol{\Lambda}$ is a $p \times p$ covariance matrix. As $\boldsymbol{\xi} = \log\left(T^{-1}M\right)^\vee$, we can apply this change of variable in the Gaussian probability density of $\boldsymbol{\xi}$, and assume $dT \approx d\boldsymbol{\xi}$ in cases where $\boldsymbol{\xi}$ is sufficiently small (Wang and Chirikjian, 2006; Barfoot and Furgale, 2014). Then, in this case, the concentrated Gaussian probability density is defined as

$$1 = \int_G \frac{1}{\sqrt{(2\pi)^p \det(\boldsymbol{\Lambda})}} \exp\left(-\frac{1}{2}\log\left(T^{-1}M\right)^{\vee T} \boldsymbol{\Lambda}^{-1} \log\left(T^{-1}M\right)^\vee\right) dT = \int_G p(T) dT. \tag{4.11}$$

We say that T is a concentrated Gaussian random variable on G centered at M, and we denote

$$T \sim \mathcal{N}_G(M, \boldsymbol{\Lambda}), \tag{4.12}$$

where M is the *mean* and $\boldsymbol{\Lambda}$ the *covariance* matrix of T. In practice, we never need to compute $p(T)$ as we can work directly with $p(\boldsymbol{\xi})$. For example, the expected value of some nonlinear function of T, $f(T)$, can be computed given the following,

$$\mathbb{E}[f(T)] = \int_G f(T)p(T)dT = \int_{\mathbb{R}^p} f(M\exp(-\boldsymbol{\xi}^\wedge))p(\boldsymbol{\xi})d\boldsymbol{\xi}. \tag{4.13}$$

Remark 4.1. *In the related literature, the random variable T in G is generally defined as $T = M\exp(\boldsymbol{\xi}^\wedge)$ rather than (4.10), given in this case $\boldsymbol{\xi} = \log\left(M^{-1}T\right)^\vee$. However, as $\boldsymbol{\xi}$ is centered on zero, our formulation is equivalent and used in this chapter for convenience.*

Given this definition, we can also extend the concept of the Gaussian process to the concentrated Gaussian distribution on a Lie group G. A *Gaussian process* is a stochastic process defined over a

continuous domain, typically represented as a collection of random variables, such that any finite subset of those variables has a joint Gaussian distribution. We define the following stochastic process on G,

$$T(t) = M(t)\exp(-\boldsymbol{\xi}(t)^{\wedge}), \tag{4.14}$$

where $t \mapsto M(t)$ is a continuous function on G and $t \mapsto \boldsymbol{\xi}(t)$ is a Gaussian process of $\mathbb{R}^p$ defined as $\boldsymbol{\xi}(t) \sim \mathcal{GP}_{\mathbb{R}^p}(\mathbf{0}, P(t,t'))$. Then, we can say that $T(t)$ is a *concentrated Gaussian process on* G, and we denote

$$T(t) \sim \mathcal{GP}_G(M(t), P(t,t')). \tag{4.15}$$

3 Frenet-Serret State Space Model in SE(3)

Recall the main Frenet-Serret equations in the 3-dimensional space defined in Section 3 of Chapter 2. Given a curve $X : [0,1] \to \mathbb{R}^3$ and its corresponding Frenet path $Q : [0,1] \to SO(3)$ such that $s \mapsto Q(s) = [T(s)|N(s)|B(s)]$, they are defined through the following dynamic equations parameterized by scalar functions on $[0,1]$, which are the curvature $s \mapsto \kappa(s)$ and the torsion $s \mapsto \tau(s)$:

$$\frac{dX(s)}{ds} = T(s), \qquad \frac{dQ(s)}{ds} = Q(s)\begin{pmatrix} 0 & -\kappa(s) & 0 \\ \kappa(s) & 0 & -\tau(s) \\ 0 & \tau(s) & 0 \end{pmatrix}, \tag{4.16}$$

with the additional initial conditions, $X(0) = X_0$ and $Q(0) = Q_0$, corresponding respectively to the initial position and the initial orientation of the trajectory in the space. In this chapter, we solely consider the realistic observation model, which assumes that we have only noisy and discrete observations of the Euclidean curve. We suppose having n noisy observations $\boldsymbol{y}_i \in \mathbb{R}^3$, $i = 1, \ldots, n$ on the discretized grid $s_1 < \ldots < s_n$ on $[0,1]$, which satisfy

$$\boldsymbol{y}_i = X(s_i) + \boldsymbol{\epsilon}_i, \tag{4.17}$$

where $\boldsymbol{\epsilon}_i$'s are independent random variables with mean zero and constant covariance matrix $\boldsymbol{\Gamma}$. In this case, estimating the Frenet-Serret differential equations in a unified way means being able to obtain an estimate of the latent variables X and Q and the functional parameters of the differential equation $\boldsymbol{\theta}$ simultaneously from the discrete Euclidean observations $\{\boldsymbol{y}_i\}_{i=1,\ldots,n}$ only. To address this problem, we follow the idea of Pilté et al. (2017), who suggest expressing this problem in a more uniform way by embedding the latent variables into a single variable defined on $SE(3)$ and considering the associated dynamic equation.

3.1 Embedding of Frenet-Serret Equations in SE(3)

From the shape function of the Euclidean curve X and the corresponding Frenet path Q, we define the variable Z from $[0,1]$ to the special Euclidean group $SE(3)$ as

$$Z(s) = \begin{pmatrix} Q(s) & X(s) \\ \mathbf{0}^T & 1 \end{pmatrix}. \tag{4.18}$$

The curve $s \mapsto Z(s)$ is given in terms of the arc-length parameter and is well defined in $SE(3)$ as for all $s \in [0,1]$, $X(s) \in \mathbb{R}^3$ and $Q(s) \in SO(3)$. It defines an embedded representation of the position, orientation, and dynamics in the Euclidean space at each point along the curve. The dynamic equation of the variable Z inherited from equations (4.16) is

$$\frac{dZ(s)}{ds} = \begin{pmatrix} \frac{dQ(s)}{ds} & \frac{dX(s)}{ds} \\ \mathbf{0}^T & 0 \end{pmatrix} = \begin{pmatrix} Q(s) & X(s) \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} 0 & -\kappa(s) & 0 & 1 \\ \kappa(s) & 0 & -\tau(s) & 0 \\ 0 & \tau(s) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.19}$$

By defining the vector $\boldsymbol{\omega}_{\boldsymbol{\theta}}(s) = (\tau(s),\, 0,\, \kappa(s),\, 1, 0,\, 0)^T$ and using the notation (2.7) the Frenet-Serret equation on $SE(3)$ can be written as

$$\frac{dZ(s)}{ds} = Z(s)\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\wedge}, \qquad Z(0) = Z_0 = \begin{pmatrix} Q_0 & X_0 \\ \mathbf{0}^T & 1 \end{pmatrix}. \tag{4.20}$$

The noisy observations of the Euclidean curve $\{\boldsymbol{y}_i\}_{i=1,\dots,n}$ can also be expressed as a function of Z by

$$\tilde{\boldsymbol{y}}_i = Z(s_i)\boldsymbol{u} + \tilde{\boldsymbol{\epsilon}}_i \tag{4.21}$$

where $\boldsymbol{u} = (0,0,0,1)^T$ and $\tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{\epsilon}}_i \in \mathbb{R}^4$ are the augmented vectors $\tilde{\boldsymbol{y}}_i = \begin{pmatrix} \boldsymbol{y}_i \\ 1 \end{pmatrix}$ and $\tilde{\boldsymbol{\epsilon}}_i = \begin{pmatrix} \boldsymbol{\epsilon}_i \\ 0 \end{pmatrix}$.

In this model, rewritten in $SE(3)$ by the equations (4.20) and (4.21), the latent variable of the problem is the state $Z(s) \in SE(3), \forall s \in [0,1]$. This state variable is non-Euclidean, and its propagation is described by a differential equation on the Lie group $SE(3)$ parameterized by the infinite-dimensional time-varying parameter $\boldsymbol{\theta}$ of interest. In particular, and contrary to the assumptions made by Park and Brunel (2019) introduced in Chapter 3, in this setting (4.20, 4.21), the state is highly partially observed. Therefore, the usual methods for ODE estimation discussed in the introduction of Chapter 3 (Ramsay and Hooker, 2017; Ramsay et al., 2007; Brunel and Clairon, 2015) may lead to difficulties in obtaining accurate estimates due to limited information content in the observations.

3.2 Definition of the Propagation Model

The equation (4.20) that defines the dynamics of our model is completely deterministic (i.e., does not inherently model stochasticity), but the structural parameters of the model are infinite-dimensional and, as we mention, a large proportion of the model's components are unknown and unobserved. In this setting, we must take into account the potential misspecification of the model due to its structural uncertainty. Otherwise, as shown in Brynjarsdóttir and O'Hagan (2014), this can lead to biased parameter estimates and inaccurate state predictions.

3.2.1 Accounting for Model's Misspecification

Brynjarsdóttir and O'Hagan (2014) emphasizes the importance of carefully modeling the available prior information related to the model when accounting for the model discrepancy to obtain realistic parameter estimates. In our case, the specific form of the matrix $\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\wedge}$ must be preserved to ensure that the solution effectively corresponds to a Euclidean trajectory and its corresponding Frenet path. In other words, we must maintain the form of a Frenet-Serret equation by keeping the constant terms in the matrix $\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\wedge}$ at their specific values (0 or 1). Different methods could be considered to incorporate the uncertainty of the parameters in the model based on this prior information. For example, Clairon and Brunel (2019) consider adding a forcing function that represents the potential model's misspecification. We choose here to apply stochasticity to capture the model's uncertainties due to the unknown structural parameters, although the system is typically not inherently stochastic (Pilté et al., 2017). As our dynamic equation is defined over a matrix Lie group, a process noise cannot simply be added as in the Euclidean setting, but standard addition may be replaced with group multiplication as well explained in detail in Barrau and Bonnabel (2018). We then consider the following perturbed model, defined by a stochastic differential equation on the matrix Lie group $SE(3)$:

$$\frac{dZ(s)}{ds} = Z(s)\left(\boldsymbol{\omega}_{\boldsymbol{\theta}}(s) + L\mathbf{w}(s)\right)^{\wedge} \tag{4.22}$$

where $\mathbf{w}(s)$ is a continuous centred white noise process in $\mathbb{R}^2$ with covariance matrix $\boldsymbol{\Sigma}$, and $L \in \mathbb{R}^{6\times 2}$ is a time-invariant dispersion matrix maintaining the form of the Frenet-Serret equation, such that $L_{1,2} = 1$,

$L_{3,1} = 1$, and all other coefficients are null. Basically, the only variables in the vector $\boldsymbol{\omega_\theta}(s)$ that are assumed to be noisy in this formulation are the components of the vector $\boldsymbol{\theta}(s) = (\kappa(s), \tau(s))^T$, and the considered process noise $\mathbf{w}(s)$ directly models the noise on these Frenet curvatures. We recall that the white noise $\mathbf{w}(s)$ is mathematically defined through its integral, $\int_{s_1}^{s_2} \mathbf{w}(s)ds = B(s_1) - B(s_2)$, with $B(s)$ a 2-dimensional Brownian motion such that for $s < t$, $B(t) - B(s) \sim \mathcal{N}(0, (t-s)\boldsymbol{\Sigma})$. Based on this, the continuous white noise process $\mathbf{w}(s)$ is sometimes expressed as a Gaussian process with a zero constant mean function and a specific form of the covariance function defined with the Dirac delta function δ as

$$\mathbf{w}(s) \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\Sigma}\delta(s - s')). \tag{4.23}$$

The proposed observation and propagation models defined by equations (4.21) and (4.22), represent a partially observed stochastic differential equation on $SE(3)$, considering potential uncertainties in the parameters. In addition, we assume a prior initial state distribution $Z(0) = Z_0 \sim \mathcal{N}_{SE(3)}(\boldsymbol{\mu_0}, \boldsymbol{P_0})$. Then, the model (4.24) defines our Frenet-Serret continuous-discrete state space model in $SE(3)$

$$\begin{cases} \frac{dZ(s)}{ds} &= Z(s)\left(\boldsymbol{\omega_\theta}(s) + L\mathbf{w}(s)\right)^\wedge & \mathbf{w}(s) \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\Sigma}\delta(s-s')) \\ \tilde{\boldsymbol{y}}_i &= Z(s_i)\boldsymbol{u} + \tilde{\boldsymbol{\epsilon}}_i & \tilde{\boldsymbol{\epsilon}}_i^T = (\boldsymbol{\epsilon}_i^T, 0),\ \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})\,. \end{cases} \tag{4.24}$$

3.2.2 Induced State Prior Distribution

In the case of an SDE on $\mathbb{R}^d$ that is linear and driven by a Gaussian noise process, the solution will also have a Gaussian distribution (Sarkka, 2006). However, if the SDE is nonlinear or defined in a more complex space than $\mathbb{R}^d$, as in our setting, the distribution of the solution may not be Gaussian and can be more complex to determine analytically. In such cases, numerical methods or approximation techniques may be employed to estimate the distribution of the solution, as proposed in Barrau and Bonnabel (2018) through discretization schemes of the SDE or in Barfoot (2017) through approximations in a continuous formulation. These approaches rely on the idea of linearizing the Lie group equation. Let $\mu(s)$ denote the solution of the deterministic Frenet-Serret ODE (4.20) and $Z(s)$ the solution of the SDE (4.22), which are both defined from $[0, 1]$ to $SE(3)$. We introduce the variable $\boldsymbol{\xi} : [0, 1] \to \mathbb{R}^6$ as

$$\boldsymbol{\xi}(s) = \log\left(Z(s)^{-1}\mu(s)\right)^\vee. \tag{4.25}$$

The vector $\boldsymbol{\xi}(s) \in \mathbb{R}^6$ represents the small deviation or perturbation in the Lie algebra between the solution obtained from the deterministic model's ODE and the solution obtained from the perturbed model's SDE. From this modeling, the variable $Z(s)$ is then expressed as $Z(s) = \mu(s)\exp(-\boldsymbol{\xi}(s)^\wedge)$. Assuming by hypothesis that the error variable $\boldsymbol{\xi}(s)$ remains small, we can derive the linearization of the SDE (4.22) over $Z(s)$ proposed in the following Lemma (Barrau and Bonnabel, 2017).

Lemma 4.2. *Consider the stochastic differential equation (4.22) with the initial condition $Z(0) = Z_0$ such that $Z_0 \sim \mathcal{N}_{SE(3)}(\boldsymbol{\mu_0}, \boldsymbol{P_0})$, where $\boldsymbol{\mu_0} \in SE(3)$ and $\boldsymbol{P_0} \in \mathbb{R}^{6\times6}$. This SDE can be linearized into the two following dynamical equations: a mean dynamics over the function $\mu(s) \in SE(3)$ and a perturbation dynamics over the matrix function $P(s) \in \mathbb{R}^{6\times6}$:*

$$\frac{d\mu(s)}{ds} = \mu(s)\boldsymbol{\omega_\theta}(s)^\wedge, \tag{4.26}$$

$$\frac{dP(s)}{ds} = -\boldsymbol{\omega_\theta}(s)^\curlywedge P(s) - P(s)\boldsymbol{\omega_\theta}(s)^{\curlywedge T} + L\boldsymbol{\Sigma}L^T, \tag{4.27}$$

with initial conditions $\mu(0) = \boldsymbol{\mu_0}$ and $P(0) = \boldsymbol{P_0}$, assuming that the left-invariant error $\log\left(Z(s)^{-1}\mu(s)\right)^\vee$ remains small for all $s \in [0, 1]$.

Proof. Consider the previously defined left-invariant error between $Z(s)$ and $\mu(s)$, which is $\boldsymbol{\xi}(s) = \log\left(Z(s)^{-1}\mu(s)\right)^{\vee}$. This error variable is assumed by hypothesis to be relatively small so that we can consider the following approximation,

$$Z(s) = \mu(s)\exp(-\boldsymbol{\xi}(s)^{\wedge}) = \mu(s)\left(I - \boldsymbol{\xi}(s)^{\wedge}\right) + O\left(\|\boldsymbol{\xi}(s)^{\wedge}\|^2\right). \tag{4.28}$$

From Ito's product rule (Jacod and Shiryaev, 2013), we can also obtain an approximation for $dZ(s)$ as

$$d\left(\mu(s) - \mu(s)\boldsymbol{\xi}(s)^{\wedge}\right) = d\mu(s) - (d\mu(s))\boldsymbol{\xi}(s)^{\wedge} - \mu(s)d\boldsymbol{\xi}(s)^{\wedge} - \underbrace{d\mu(s)d\boldsymbol{\xi}(s)}_{=0}, \tag{4.29}$$

where the last term is null because μ is non-stochastic. By replacing, in the SDE on $Z(s)$ (4.22), the terms by these two approximations (4.28) and (4.29) we obtain

$$\frac{d\mu(s)}{ds} - \frac{d\mu(s)}{ds}\boldsymbol{\xi}(s)^{\wedge} - \mu(s)\frac{d\boldsymbol{\xi}(s)^{\wedge}}{ds} = \mu(s)(I - \boldsymbol{\xi}(s)^{\wedge})(\boldsymbol{\omega_\theta}(s) + L\mathbf{w}(s))^{\wedge} + O\left(\|\boldsymbol{\xi}(s)\|^2\right).$$

As $\mu(s)$ is defined as the solution of (4.26), we can manipulate the perturbed system (4.22), using the definitions of the adjoint representation and the Lie bracket (2.10), into the following pair of equations:

$$\begin{aligned}
\text{(Mean)} \quad & \frac{d\mu(s)}{ds} = \mu(s)\boldsymbol{\omega_\theta}(s)^{\wedge} \\
\text{(Perturbation)} \quad & \frac{d\boldsymbol{\xi}(s)}{ds} = -\left(\boldsymbol{\omega_\theta}(s)^{\wedge}\boldsymbol{\xi}(s)^{\wedge} - \boldsymbol{\xi}(s)^{\wedge}\boldsymbol{\omega_\theta}(s)^{\wedge}\right)^{\vee} - L\mathbf{w}(s) + \left(\boldsymbol{\xi}(s)^{\wedge}L\mathbf{w}(s)^{\wedge}\right)^{\vee} + O\left(\|\boldsymbol{\xi}(s)\|^2\right) \\
& \qquad = -\boldsymbol{\omega_\theta}(s)^{\curlywedge}\boldsymbol{\xi}(s) - L\mathbf{w}(s) + O\left(\|\boldsymbol{\xi}(s)\|^2\|\mathbf{w}(s)\|^2\right) + O\left(\|\boldsymbol{\xi}(s)\|^2\right)
\end{aligned}$$

The mean dynamic equation is deterministic given the parameter $\boldsymbol{\theta}$. For the perturbation dynamics, by neglecting terms of order $O\left(\|\boldsymbol{\xi}(s)\|^2\|\mathbf{w}(s)\|^2\right) + O\left(\|\boldsymbol{\xi}(s)\|^2\right)$ (see Barrau and Bonnabel (2017)) the remaining pertubation dynamical equation is a linear time-varying SDE driven by a Gaussian noise process. From Theorem 2.6 in Sarkka (2006), its solution is a Gaussian process with mean function $\boldsymbol{m}(s) \in \mathbb{R}^6$ and covariance function $P(s) \in \mathbb{R}^{6\times 6}$ defined as solutions of

$$\frac{d\boldsymbol{m}(s)}{ds} = -\boldsymbol{\omega_\theta}(s)^{\curlywedge}\boldsymbol{m}(s) \tag{4.30}$$

$$\frac{dP(s)}{ds} = -\boldsymbol{\omega_\theta}(s)^{\curlywedge}P(s) - P(s)\boldsymbol{\omega_\theta}(s)^{\curlywedge^T} + L\boldsymbol{\Sigma}L^T. \tag{4.31}$$

Therefore, by neglecting terms of order $O\left(\|\boldsymbol{\xi}(s)\|^2\|\mathbf{w}(s)\|^2\right) + O\left(\|\boldsymbol{\xi}(s)\|^2\right)$, the solution $\boldsymbol{\xi}(s)$ can be approximated by a Gaussian process $\mathcal{GP}(\boldsymbol{m}(s), P(s,s'))$. As we suppose $Z(0) \sim \mathcal{GP}_{SE}(\mu(0) = \boldsymbol{\mu_0}, P(0) = \boldsymbol{P_0})$ and $\boldsymbol{\xi}(0) = \log\left(Z(0)^{-1}\mu(0)\right)^{\vee}$, we have $\boldsymbol{\xi}(0) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{P_0})$ (see Section 2.2). Therefore, $\boldsymbol{m}(0) = \mathbf{0}$ and thus the mean function $\boldsymbol{m}(s)$ remains null for all $s \in [0,1]$. Hence, the error variable $\boldsymbol{\xi}(s)$ is approximated by the Gaussian process $\mathcal{GP}\left(\mathbf{0}, P(s,s')\right)$ and so the perturbation dynamics is fully described by the equation (4.31). ■

From the linearization of (4.22) derived in Lemma 4.2, we have obtained an approximation of the variable $\boldsymbol{\xi}(s) = \log\left(Z(s)^{-1}\mu(s)\right)^{\vee}$ by the Gaussian process $\mathcal{GP}\left(\mathbf{0}, P(s,s')\right)$. Therefore, given the definition of the concentrated Gaussian distribution on Lie group (Section 2.2), we approximate also the distribution of the solution $Z(s)$ of the SDE (4.22) by a concentrated Gaussian process on $SE(3)$ (Section 2.2) with mean $\mu(s)$ and covariance function $P(s,s')$. We denote

$$Z(s) \overset{\text{approx}}{\sim} \mathcal{GP}_{SE}(\mu(s), P(s,s')). \tag{4.32}$$

From (4.31), the cross-covariance function is given as (Sarkka, 2006)

$$P(s,s') = \phi(s,0)P_0\phi(s',0)^T + \int_0^{min(s,s')} \phi(s,t)L\boldsymbol{\Sigma}L^T\phi(s',t)^T dt, \tag{4.33}$$

where $\phi(s,t)$ is the transition function defined as:

$$\begin{cases} \phi(s,s) & = I \\ \phi'(s,t) & = -\boldsymbol{\omega_\theta}(s)^\curlywedge \phi(s,t) \\ \phi(s,t) & = \phi(s,r)\phi(r,t). \end{cases} \tag{4.34}$$

In this section, we propose to model the estimation problem of the Frenet-Serret framework's parameters through the inference of the state space model defined in (4.24). We aim to estimate the hidden state $Z(s)$ and learn the parameter values $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\Sigma}, \boldsymbol{\Gamma}, \boldsymbol{P_0}, \boldsymbol{\mu_0}\}$ of this proposed model. In a full Euclidean setting, this problem is typically addressed using an Expectation-Maximization (EM) algorithm (Dempster et al., 1977; Meng and van Dyk, 1997; Dwivedi et al., 2018), which involves an Expectation step (E-step) and a Maximization step (M-step). The E-step infers the posterior distribution of the hidden states based on the current parameter estimates and observed data, while the M-step maximizes the expected log-likelihood to update the parameter estimates. The algorithm iteratively repeats these steps until convergence, determined by a predefined threshold for the change in parameter estimates or likelihood values. However, since our continuous dynamics operate on the Lie group $SE(3)$ while our discrete observations are in Euclidean space $\mathbb{R}^3$, and since some of our parameters are infinite-dimensional, we propose an adapted version of the EM framework accordingly in the following sections.

4 Latent Variables Estimation using an Invariant Extended Kalman Filter/Smoother on SE(3)

The classic algorithm for inferring the hidden state in a state space model is the well-known Kalman Filter (KF), originally introduced by Kalman (1960), which sequentially updates the state estimate based on the measurements and system dynamics. Its direct use is adapted to the restricted case of a linear Gaussian state space model. To handle the case of a state space model governed by a non-linear dynamic equation, an extension known as the Extended Kalman Filter (EKF) uses a linearization of the dynamics at each time step around the current estimated state. It assumes a Gaussian approximation of the true state's posterior distribution at each time point given the previous observations and linearizes the error system through a first-order Taylor expansion of the non-linear functions (Särkkä, 2010; Särkkä and Sarmavuori, 2013). This version is well-adapted to the non-linear system defined in Euclidean space, as the error corresponds to the simple difference between the true and estimated state. For a system defined in a matrix Lie group, Barrau and Bonnabel (2017) propose a version of the EKF that inherently respects the group structure, named the Invariant Extended Kalman Filter (IEKF). The idea, compared with the classic version of the EKF, is to use a representation of uncertainty that is consistent with the Lie group structure in the approximation of the state's posterior distribution and the error's definition. The IEKF algorithm is, therefore, well-suited to our setting (4.24). We describe it more precisely in this section and apply it for the estimation of the posterior distribution of our model hidden state $Z(s_i)$ given the i previous observations $\boldsymbol{y}_{1:i} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_i\}$ and a set of parameter $\boldsymbol{\Theta}$. Moreover, updating the optimal parameters in the maximization step of the EM algorithm requires having the posterior distribution of the hidden state given all the observations $\boldsymbol{y}_{1:n} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$. To achieve this, we propose also, in this thesis, an Invariant Extended Kalman Smoother (IEKS).

4.1 Invariant Extended Kalman Filter

The EKF supposes first to make an approximation of the posterior state distribution (Särkkä, 2010). In the IEKF algorithm, we assume the state posterior filtering distribution to be a concentrated Gaussian distribution on Lie groups (Section 2.2) (Bourmaud et al., 2015; Barrau and Bonnabel, 2018). We denote,

$$p(Z(s)|\boldsymbol{y}_1,\dots,\boldsymbol{y}_i) \approx \mathcal{N}_{SE(3)}(\boldsymbol{\mu}^{\boldsymbol{t}}(s), \boldsymbol{P}^{\boldsymbol{t}}(s)), \qquad s \in [s_{i-1}, s_i)\,, \tag{4.35}$$

where $\boldsymbol{\mu}^{\boldsymbol{t}} : [0,1] \to SE(3)$ and $\boldsymbol{P}^{\boldsymbol{t}} : [0,1] \to \mathbb{R}^{6\times 6}$ are the mean and covariance functions of the tracking distribution. At $s_{i-1} \in [0,1]$, we write the conditional random variable $Z(s_{i-1})|\boldsymbol{y}_{1:i-1} = Z_{i-1|i-1}$. Given the distribution parameters at time step s_{i-1}, ${\boldsymbol{\mu}^{\boldsymbol{t}}}_{i-1|i-1}$ and ${\boldsymbol{P}^{\boldsymbol{t}}}_{i-1|i-1}$, and the observation $\boldsymbol{y}_i$ at time step s_i, the IEKF provides the values of distribution parameters at time step s_i, which are ${\boldsymbol{\mu}^{\boldsymbol{t}}}_{i|i}$ and ${\boldsymbol{P}^{\boldsymbol{t}}}_{i|i}$. It consists of a propagation step defined by the equations (4.37) and (4.38) derived in Section 4.1.1 and an updating step (4.43) defined in Section 4.1.2.

The essential ingredient for defining the IEKF's equations is to approximate the system error through the left-invariant error defined in a vector space. By definition of the concentrated Gaussian distribution on $SE(3)$, we have from (4.35)

$$Z_{i-1|i-1} \approx {\boldsymbol{\mu}^{\boldsymbol{t}}}_{i-1|i-1} \exp\left(-{\boldsymbol{\xi}^{\boldsymbol{t}}}^{\wedge}_{i-1|i-1}\right)\,, \tag{4.36}$$

where the Lie algebraic error ${\boldsymbol{\xi}^{\boldsymbol{t}}}_{i-1|i-1} = \log\left((Z_{i-1|i-1})^{-1}{\boldsymbol{\mu}^{\boldsymbol{t}}}_{i-1|i-1}\right)^{\vee} \sim \mathcal{N}({\boldsymbol{m}^{\boldsymbol{t}}}_{i-1|i-1} = \boldsymbol{0}, {\boldsymbol{P}^{\boldsymbol{t}}}_{i-1|i-1})$ locally approximates the error between true and estimated states as the left-invariant error on $SE(3)$.

4.1.1 Propagation

This section aims to propagate the parameters $\boldsymbol{\mu}^{\boldsymbol{t}}$ and $\boldsymbol{P}^{\boldsymbol{t}}$ between two consecutive measurement points s_{i-1} and s_i. Along the lines of the EKF, our aim is to linearize the error system by employing a first-order approximation of the non-linear functions around the estimated state. Precisely, Lemma 4.2 shows that by considering the first-order approximation of the system error, $\exp(-{\boldsymbol{\xi}^{\boldsymbol{t}}}^{\wedge}_{i-1|i-1}) = I - {\boldsymbol{\xi}^{\boldsymbol{t}}}^{\wedge}_{i-1|i-1} + O(\|{\boldsymbol{\xi}^{\boldsymbol{t}}}^{\wedge}_{i-1|i-1}\|^2)$, we can linearize the system between two differential equations: one defining the mean propagation and the other the covariance propagation. Then, the propagation step consists of integrating the differential equations

$$\frac{d\boldsymbol{\mu}^{\boldsymbol{t}}(s)}{ds} = \boldsymbol{\mu}^{\boldsymbol{t}}(s)\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\wedge}\,, \tag{4.37}$$

$$\frac{d\boldsymbol{P}^{\boldsymbol{t}}(s)}{ds} = -\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\curlywedge}\boldsymbol{P}^{\boldsymbol{t}}(s) - \boldsymbol{P}^{\boldsymbol{t}}(s)\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\curlywedge^T} + L\boldsymbol{\Sigma}L^T\,, \tag{4.38}$$

from initial conditions $\boldsymbol{\mu}^{\boldsymbol{t}}(s_{i-1}) = {\boldsymbol{\mu}^{\boldsymbol{t}}}_{i-1|i-1}$ and $\boldsymbol{P}^{\boldsymbol{t}}(s_{i-1}) = {\boldsymbol{P}^{\boldsymbol{t}}}_{i-1|i-1}$ to the time instance s_i. The propagated mean and covariance are given as ${\boldsymbol{\mu}^{\boldsymbol{t}}}_{i|i-1} = \boldsymbol{\mu}^{\boldsymbol{t}}(s_i)$ and ${\boldsymbol{P}^{\boldsymbol{t}}}_{i|i-1} = \boldsymbol{P}^{\boldsymbol{t}}(s_i)$, respectively. In the same way as in (4.30), the mean Lie algebraic error is propagated by the equation (Sarkka, 2006)

$$\frac{d\boldsymbol{m}^{\boldsymbol{t}}(s)}{ds} = -\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\curlywedge}\boldsymbol{m}^{\boldsymbol{t}}(s). \tag{4.39}$$

However, as the initial condition is ${\boldsymbol{m}^{\boldsymbol{t}}}_{i-1|i-1} = \boldsymbol{0}_{\mathbb{R}^6}$, to the first order we will have ${\boldsymbol{m}^{\boldsymbol{t}}}_{i|i-1} = \boldsymbol{0}_{\mathbb{R}^6}$.

4.1.2 Update

To update the propagated mean ${\boldsymbol{\mu}^{\boldsymbol{t}}}_{i|i-1}$ and covariance ${\boldsymbol{P}^{\boldsymbol{t}}}_{i|i-1}$ by incorporating the new measurement $\tilde{\boldsymbol{y}}_i$, we begin by examining the linearization of the observation model with respect to ${\boldsymbol{\xi}^{\boldsymbol{t}}}_{i|i-1}$, as detailed in the following proposition.

Proposition 4.1. *The observation model in (4.24) can be expressed as a function of* $\boldsymbol{\xi}(s) = \log\left(Z(s)^{-1}\mu(s)\right)^{\vee}$, *where* $\mu(s)$ *is a solution of (4.26), which is given by*

$$\hat{\boldsymbol{y}}_i = -H\boldsymbol{\xi}(s) + \hat{\boldsymbol{\epsilon}}_i\,, \tag{4.40}$$

where $H = \begin{pmatrix} 0_{3\times 3} & I_3 \end{pmatrix} \in \mathbb{R}^{3\times 6}$, $\hat{\boldsymbol{y}}_i = \mu_Q(s)^T(\boldsymbol{y}_i - \mu_X(s)) \in \mathbb{R}^3$ *and* $\hat{\boldsymbol{\epsilon}}_i = \mu_Q(s)^T\boldsymbol{\epsilon}_i \in \mathbb{R}^3$ *such that* $\hat{\boldsymbol{\epsilon}}_i \sim \mathcal{N}(\mathbf{0}, \mu_Q(s)^T\boldsymbol{\Gamma}\mu_Q(s))$. *The parameters* $\mu_Q(s) \in SO(3)$ *and* $\mu_X(s) \in \mathbb{R}^3$ *are respectively the rotation matrix and translation vector of* $\mu(s) \in SE(3)$ *as defined in (4.1).*

Proof. We linearize the observation model using the first-order approximation of the exponential map,

$$\tilde{\boldsymbol{y}}_i = Z(s)\boldsymbol{u} + \tilde{\boldsymbol{\epsilon}}_i = \mu(s)\exp\left(-\boldsymbol{\xi}(s)^{\wedge}\right)\boldsymbol{u} + \tilde{\boldsymbol{\epsilon}}_i = \mu(s)\left(I - \boldsymbol{\xi}(s)^{\wedge}\right)\boldsymbol{u} + \tilde{\boldsymbol{\epsilon}}_i + O\left(\|\boldsymbol{\xi}(s)^{\wedge}\|^2\right).$$

This equation can be rearranged as

$$\mu(s)^{-1}\tilde{\boldsymbol{y}}_i - \boldsymbol{u} = -\boldsymbol{\xi}(s)^{\wedge}\boldsymbol{u} + \mu(s)^{-1}\tilde{\boldsymbol{\epsilon}}_i + O\left(\|\boldsymbol{\xi}(s)^{\wedge}\|^2\right),$$

and the equation (4.40) is then obtained by identifying the vectors' first 3 components in this equality. ■

From Proposition 4.1 we have

$$\hat{\boldsymbol{y}}_i = -H{\boldsymbol{\xi}^{\boldsymbol{t}}}_{i|i-1} + \hat{\boldsymbol{\epsilon}}_i\,. \tag{4.41}$$

Since (4.41) is a linear equation with respect to ${\boldsymbol{\xi}^{\boldsymbol{t}}}_{i|i-1} \in \mathbb{R}^6$, we can use the conventional update equations of the Kalman filter to update the Lie algebraic error into the posterior distribution. Consequently, we obtain ${\boldsymbol{\xi}^{\boldsymbol{t}}}^{-}_{i|i} \sim \mathcal{N}({\boldsymbol{m}^{\boldsymbol{t}}}^{-}_{i|i}, {\boldsymbol{P}^{\boldsymbol{t}}}^{-}_{i|i})$, where the values of ${\boldsymbol{m}^{\boldsymbol{t}}}^{-}_{i|i}$ and ${\boldsymbol{P}^{\boldsymbol{t}}}^{-}_{i|i}$ are computed as follows:

$$\begin{cases} S_i & = H{\boldsymbol{P}^{\boldsymbol{t}}}_{i|i-1}H^T + {\boldsymbol{\mu}^{\boldsymbol{t}}_{\boldsymbol{Q}}}^{T}_{i|i-1}\boldsymbol{\Gamma}\boldsymbol{\mu}^{\boldsymbol{t}}_{\boldsymbol{Q}i|i-1} \\ K_i & = {\boldsymbol{P}^{\boldsymbol{t}}}_{i|i-1}H^T S_i^{-1} \\ {\boldsymbol{m}^{\boldsymbol{t}}}^{-}_{i|i} & = \mathbf{0}_{\mathbb{R}^6} + K_i(\hat{\boldsymbol{y}}_i + H\mathbf{0}_{\mathbb{R}^6}) \\ {\boldsymbol{P}^{\boldsymbol{t}}}^{-}_{i|i} & = (\mathbf{I} - K_iH){\boldsymbol{P}^{\boldsymbol{t}}}_{i|i-1} \end{cases} \tag{4.42}$$

To satisfy the concentrated Gaussian distribution assumption $Z_{i|i} = {\boldsymbol{\mu}^{\boldsymbol{t}}}_{i|i}\exp(-{\boldsymbol{\xi}^{\boldsymbol{t}}}^{\wedge}_{i|i}) \sim \mathcal{N}_{SE(3)}({\boldsymbol{\mu}^{\boldsymbol{t}}}_{i|i}, {\boldsymbol{P}^{\boldsymbol{t}}}_{i|i})$ we must have $\mathbb{E}[{\boldsymbol{\xi}^{\boldsymbol{t}}}_{i|i}] = \mathbf{0}_{\mathbb{R}^6}$ at the end of the updating step. However, we have ${\boldsymbol{m}^{\boldsymbol{t}}}^{-}_{i|i} \neq \mathbf{0}_{\mathbb{R}^6}$ after the update (4.42). Hence, we set ${\boldsymbol{m}^{\boldsymbol{t}}}_{i|i} = \mathbf{0}_{\mathbb{R}^6}$ and we perform the following reparametrization,

$${\boldsymbol{\mu}^{\boldsymbol{t}}}_{i|i} = {\boldsymbol{\mu}^{\boldsymbol{t}}}_{i|i-1}\exp\left(-\left[{\boldsymbol{m}^{\boldsymbol{t}}}^{-}_{i|i}\right]^{\wedge}\right) = {\boldsymbol{\mu}^{\boldsymbol{t}}}_{i|i-1}\exp\left(-\left[K_i\hat{\boldsymbol{y}}_i\right]^{\wedge}\right) = {\boldsymbol{\mu}^{\boldsymbol{t}}}_{i|i-1}\exp\left(\left[K_i{\boldsymbol{\mu}^{\boldsymbol{t}}_{\boldsymbol{Q}}}^{T}_{i|i-1}(\boldsymbol{\mu}^{\boldsymbol{t}}_{\boldsymbol{X}i|i-1} - \boldsymbol{y}_i)\right]^{\wedge}\right), \tag{4.43}$$

which gives the update equation of $\boldsymbol{\mu}^{\boldsymbol{t}}$. For the covariance matrix, we consider ${\boldsymbol{P}^{\boldsymbol{t}}}_{i|i} \approx {\boldsymbol{P}^{\boldsymbol{t}}}^{-}_{i|i}$. Note that filtering functions, $\boldsymbol{\mu}^{\boldsymbol{t}}$ and $\boldsymbol{P}^{\boldsymbol{t}}$, are therefore not continuous at the measurement times.

4.2 Invariant Extended Kalman Smoother

We also propose to extend the Extended Kalman Smoother on $SE(3)$ as an Invariant Extended Kalman Smoother (IEKS) to obtain an approximate smoothing solution to the filtering model. Again, we make the assumption that the smoothing solution follows a concentrated Gaussian distribution on $SE(3)$ parametrized by the mean $\boldsymbol{\mu}^{\boldsymbol{s}}$ and covariance function $\boldsymbol{P}^{\boldsymbol{s}}$,

$$p(Z(s)|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) \approx \mathcal{N}_{SE(3)}(\boldsymbol{\mu}^{\boldsymbol{s}}(s), \boldsymbol{P}^{\boldsymbol{s}}(s)), \qquad s_0 \leq s \leq s_n. \tag{4.44}$$

From Särkkä and Sarmavuori (2013), the Kalman smoothing solutions, $\boldsymbol{\xi}^{\boldsymbol{s}}$ and $\boldsymbol{P}^{\boldsymbol{s}}$, of the continuous-discrete linearized Gaussian model of the variable $\boldsymbol{\xi}^{\boldsymbol{t}}$, are given by the equations for $\tau \in [s_{i-1}, s_i]$

$$D(\tau) = \boldsymbol{P}^{\boldsymbol{t}}(\tau) C(\tau)^T \boldsymbol{P}^{\boldsymbol{t}}(s_i)^{-1} \tag{4.45}$$

$$\boldsymbol{m}^{\boldsymbol{s}}(\tau) = \boldsymbol{m}^{\boldsymbol{t}}(\tau) + D(\tau)\left(\boldsymbol{m}^{\boldsymbol{s}}(s_i) - \boldsymbol{m}^{\boldsymbol{t}}(s_i)\right) = D(\tau)\boldsymbol{m}^{\boldsymbol{s}}(s_i) \tag{4.46}$$

$$\boldsymbol{P}^{\boldsymbol{s}}(\tau) = \boldsymbol{P}^{\boldsymbol{t}}(\tau) + D(\tau)\left(\boldsymbol{P}^{\boldsymbol{s}}(s_i) - \boldsymbol{P}^{\boldsymbol{t}}(s_i)\right) D(\tau)^T , \tag{4.47}$$

where D is the smoothing gain parameterized by the solution $C(s)$ of

$$\frac{dC(s)}{ds} = -\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\curlywedge} C(s) , \tag{4.48}$$

integrated on $[s_{i-1}, s_i]$, from the initial condition $C(s_{i-1}) = I$. These smoothing parameters are computed backward with the terminal conditions $\boldsymbol{m}^{\boldsymbol{s}}(s_n) = \boldsymbol{m}^{\boldsymbol{t}}{}_{n|n} = \mathbf{0}_{\mathbb{R}^6}$ and $\boldsymbol{P}^{\boldsymbol{s}}(s_n) = \boldsymbol{P}^{\boldsymbol{t}}{}_{n|n}$. Therefore, given (4.46), we have for all $\tau \in [s_0, s_n]$, $\boldsymbol{m}^{\boldsymbol{s}}(\tau) = \mathbf{0}_{\mathbb{R}^6}$. By analogy with the IEKF, we define the smoothing mean $\boldsymbol{\mu}^{\boldsymbol{s}}$ by using the Lie group structure to express the error between $\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)$ and $\boldsymbol{\mu}^{\boldsymbol{t}}(s_i)$ through the left-invariant error. Therefore, we define the following equation for the smoothing solution. For $\tau \in [s_{i-1}, s_i]$,

$$\boldsymbol{\mu}^{\boldsymbol{s}}(\tau) = \boldsymbol{\mu}^{\boldsymbol{t}}(\tau) \exp\left(D(\tau) \log\left(\boldsymbol{\mu}^{\boldsymbol{t}}(s_i)^{-1} \boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\right)\right) , \tag{4.49}$$

which is computed backward from $\boldsymbol{\mu}^{\boldsymbol{s}}(s_n) = \boldsymbol{\mu}^{\boldsymbol{t}}{}_{n|n}$.
The maximization step of the EM algorithm also requires knowledge of the cross-covariance function $\boldsymbol{P}^{\boldsymbol{s}}(s, s')$. This one can be computed by considering a discretization scheme of the smoothing covariance differential equation of a Gaussian linear model (Särkkä and Sarmavuori, 2013), verified by $\boldsymbol{P}^{\boldsymbol{s}}(s)$,

$$\frac{d\boldsymbol{P}^{\boldsymbol{s}}(s)}{ds} = \left[-\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\curlywedge} + L\boldsymbol{\Sigma}L^T \boldsymbol{P}^{\boldsymbol{t}}(s)^{-1}\right] \boldsymbol{P}^{\boldsymbol{s}}(s) + \boldsymbol{P}^{\boldsymbol{s}}(s) \left[-\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^{\curlywedge} + L\boldsymbol{\Sigma}L^T \boldsymbol{P}^{\boldsymbol{t}}(s)^{-1}\right]^T - L\boldsymbol{\Sigma}L^T . \tag{4.50}$$

In practice, we must run the tracking algorithm first and store the estimated values so that we can use the smoothing algorithm thereafter. The computational complexity and memory requirements are then of the order of $O(6^3 n)$ and $O(6^2 n)$ (Hartikainen and Särkkä, 2010).

5 Parameter Inference by Maximization of the Penalized Expected Log-Likelihood

In this section, we aim to update the estimation of our model's parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\Sigma}, \boldsymbol{\Gamma}, \boldsymbol{P_0}, \boldsymbol{\mu_0}\}$, given the posterior distribution of our model's hidden state (4.24), $p(Z(s)|\boldsymbol{y}_{1:n})$, obtain through the IEKF and IEKS. For this, first note that this work deals with the parameter estimation of a single curve. Therefore, the total amount of information involved is insufficient to obtain a reliable estimate of the full covariance matrix $\boldsymbol{\Sigma}$, and we can only consider estimating a single constant variance σ^2, assuming the standard model for the error (Ramsay and Silverman, 2005). Then, in the following, we are looking for this simpler form of the white noise covariance matrix, $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$, which will also simplify the derivation of the parameter optimization criterion.

5.1 Complete Data Log-Likelihood

The complete data log-likelihood of the Frenet-Serret state space model (4.24) is defined by

$$\mathcal{L}(\boldsymbol{\Theta}) = \log(p(Z_{0:n}, \boldsymbol{y}_{1:n}; \boldsymbol{\Theta})) = \log\left(p(Z_{0:n}; \boldsymbol{\Theta}) \prod_{i=1}^{n} p(\boldsymbol{y}_i|Z_i; \boldsymbol{\Theta})\right) = \log\left(p(Z_{0:n}; \boldsymbol{\Theta})\right) + \sum_{i=1}^{n} \log\left(p(\boldsymbol{y}_i|Z_i; \boldsymbol{\Theta})\right) , \tag{4.51}$$

where $Z_{0:n} = (Z(s_0), \dots, Z(s_n))$ and $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \sigma, \boldsymbol{\Gamma}, \boldsymbol{P_0}, \boldsymbol{\mu_0}\}$ the set of parameters to learn. The first term accounts for the error in the propagation model defined by the SDE (4.22), and the second one for the error in the measurements model (4.21).

5.1.1 Error in Propagation Model

From Lemma 4.2, by denoting by $\mathbf{P}$ the full covariance matrix such that $\mathbf{P}_{ij} = P(s_i, s_j) \in \mathbb{R}^{6\times 6}$, and the error vector by $\boldsymbol{\xi}_{0:n} = (\boldsymbol{\xi}(s_0), \dots, \boldsymbol{\xi}(s_n))^T$, where $\boldsymbol{\xi}(s_i) = \log(Z(s_i)^{-1}\mu(s_i))^\vee$ is assumed to remain sufficiently small, we have $p(Z_{0:n}; \boldsymbol{\Theta}) \overset{\text{approx}}{\sim} \mathcal{N}_{SE(3)}(\mu_{0:n}, \mathbf{P})$ and thus

$$p\left(Z_{0:n}; \boldsymbol{\Theta}\right) \approx \frac{1}{\sqrt{(2\pi)^{6n}\det\left(\mathbf{P}\right)}} \exp\left(-\frac{1}{2}\boldsymbol{\xi}_{0:n}^T \mathbf{P}^{-1} \boldsymbol{\xi}_{0:n}\right). \tag{4.52}$$

Proposition 4.2. *The distribution of $Z_{0:n}$ can be re-expressed as*

$$p(Z_{0:n}; \boldsymbol{\Theta}) \approx \frac{1}{\sqrt{(2\pi)^{6n}\det(\mathbf{W})}} \exp\left(-\frac{1}{2}\boldsymbol{e}_{0:n}^T \mathbf{W}^{-1} \boldsymbol{e}_{0:n}\right), \tag{4.53}$$

where $\boldsymbol{e}_0 = \boldsymbol{\xi}(s_0) = \log\left(Z(s_0)^{-1}\mu(s_0)\right)^\vee$, $\mathbf{W} = diag(P_0, W_1, \dots, W_n)$, and for $i > 0$,

$$W_i = \int_{s_{i-1}}^{s_i} \phi(s_i, t)\sigma^2 L L^T \phi(s_i, t)^T dt\,, \tag{4.54}$$

$$\boldsymbol{e}_i = \boldsymbol{\xi}(s_i) - \phi(s_i, s_{i-1})\boldsymbol{\xi}(s_{i-1}) = \log\left(Z(s_i)^{-1}\mu(s_i)\right)^\vee - \phi(s_i, s_{i-1}) \log\left(Z(s_{i-1})^{-1}\mu(s_{i-1})\right)^\vee. \tag{4.55}$$

Proof. From the covariance differential equation (4.27) one can show that $\mathbf{P} = \mathbf{C}\mathbf{W}\mathbf{C}^T$ (Barfoot (2017) *section 3.4*), where the matrix $\mathbf{C}$ and its inverse are defined by

$$\mathbf{C} = \begin{pmatrix} I & 0 & \dots & & 0 \\ \phi(s_1, s_0) & I & & & \\ \phi(s_2, s_0) & \phi(s_2, s_1) & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ \phi(s_n, s_0) & \phi(s_n, s_1) & \dots & \phi(s_n, s_{n-1}) & I \end{pmatrix}, \quad \mathbf{C}^{-1} = \begin{pmatrix} I & 0 & & \dots & 0 \\ -\phi(s_1, s_0) & I & & & \\ 0 & -\phi(s_2, s_1) & \ddots & & \\ \vdots & & \ddots & \ddots & \\ 0 & \dots & 0 & -\phi(s_n, s_{n-1}) & I \end{pmatrix}$$

given the transition matrix function $\phi(\cdot, \cdot)$ defined by the equations (4.34). Then $\mathbf{P}^{-1} = \mathbf{C}^{T^{-1}}\mathbf{W}^{-1}\mathbf{C}^{-1}$ is block-tridiagonal and the 6×6 dimensional block matrices are

$$\begin{cases} \mathbf{P}_{i,i+1}^{-1} &= -\phi(s_{i+1}, s_i)^T W_{i+1}^{-1} \\ \mathbf{P}_{i,i}^{-1} &= W_i^{-1} + \phi(s_{i+1}, s_i)^T W_{i+1}^{-1}\phi(s_{i+1}, s_i) \\ \mathbf{P}_{i,i-1}^{-1} &= -W_i^{-1}\phi(s_i, s_{i-1}) \end{cases}$$

and $\mathbf{P}_{0,0}^{-1} = \boldsymbol{P_0}^{-1}$. Then, by denoting $\boldsymbol{e}_{0:n} = \mathbf{C}^{-1}\boldsymbol{\xi}_{0:n} \in \mathbb{R}^{6n}$ the distribution of $Z_{0:n}$ can be re-expressed as (4.53). ∎

In the Proposition 4.2, the variable $\boldsymbol{e}_i$ is expressed in terms of $\mu(s_i), \mu(s_{i-1})$ and $\phi(s_i, s_{i-1})$, which depend on the unknown parameter $\boldsymbol{\theta}$. Therefore, to obtain a convenient expression for the estimation of $\boldsymbol{\theta}$, we derive an approximation of the error $\boldsymbol{e}_i$ that isolates this parameter in the following proposition.

Proposition 4.3. *Given the notation $u_i = s_i - s_{i-1}$ and $v_i = \frac{s_i + s_{i-1}}{2}$, the error of the propagation model $\boldsymbol{e}_i \in \mathbb{R}^6$ at a time step $i > 0$, defined in (4.55), is approximated as*

$$\boldsymbol{e}_i = u_i\left(\boldsymbol{\omega}_{\boldsymbol{\theta}}\left(v_i\right) - \boldsymbol{r}_i\right) + O\left(u_i^2\right), \tag{4.56}$$

where $\boldsymbol{r}_i = -\frac{1}{u_i}\log\left(Z(s_i)^{-1}Z(s_{i-1})\right)^\vee \in \mathbb{R}^6$ represents the raw observation of $\boldsymbol{\omega}_{\boldsymbol{\theta}}\left(v_i\right)$.

Proof. First, we can approximate the transition matrix $\phi(s_i, s_{i-1})$ with the Magnus approximation (3.21) of the associated differential equation:

$$\phi(s,t) = \exp\left(-(s-t)\boldsymbol{\omega_\theta}\left(\frac{s+t}{2}\right)^\curlywedge\right) + O\left((s-t)^2\right) = Ad\left(\exp\left(-(s-t)\boldsymbol{\omega_\theta}\left(\frac{s+t}{2}\right)^\wedge\right)\right) + O\left((s-t)^2\right).$$

Using the notation $u_i = s_i - s_{i-1}$ and $v_i = \frac{s_i + s_{i-1}}{2}$ we derive an approximation of the propagation error,

$$\begin{aligned}
\boldsymbol{e}_i &= \log\left(Z(s_i)^{-1}\mu(s_i)\right)^\vee - Ad\left(\exp\left(-u_i\boldsymbol{\omega_\theta}\left(v_i\right)^\wedge\right)\right)\log\left(Z(s_{i-1})^{-1}\mu(s_{i-1})\right)^\vee + O\left(u_i^2\right) \\
&= \log\left(Z(s_i)^{-1}\mu(s_i)\right)^\vee + \log\left(\exp\left(-\left(Ad\left(\exp\left(-u_i\boldsymbol{\omega_\theta}\left(v_i\right)^\wedge\right)\right)\log\left(Z(s_{i-1})^{-1}\mu(s_{i-1})\right)^\vee\right)^\wedge\right)\right)^\vee + O\left(u_i^2\right) \\
&= \log\left(Z(s_i)^{-1}\mu(s_i)\right)^\vee + \log\left(\exp\left(-u_i\boldsymbol{\omega_\theta}\left(v_i\right)^\wedge\right)\mu(s_{i-1})^{-1}Z(s_{i-1})\exp\left(u_i\boldsymbol{\omega_\theta}\left(v_i\right)^\wedge\right)\right)^\vee + O\left(u_i^2\right) \\
&= \log\left(Z(s_i)^{-1}\underbrace{\mu(s_i)\exp\left(-u_i\boldsymbol{\omega_\theta}\left(v_i\right)^\wedge\right)\mu(s_{i-1})^{-1}}_{=I+O(u_i^2)}Z(s_{i-1})\exp\left(u_i\boldsymbol{\omega_\theta}\left(v_i\right)^\wedge\right)\right)^\vee + O\left(u_i^2\right) \\
&= \log\left(Z(s_i)^{-1}Z(s_{i-1})\right)^\vee + u_i\boldsymbol{\omega_\theta}\left(v_i\right) + O\left(u_i^2\right) \quad \text{(using the BCH formula (2.16))} \\
&= u_i\left(\boldsymbol{\omega_\theta}\left(v_i\right) - \boldsymbol{r}_i\right) + O\left(u_i^2\right)
\end{aligned}$$

where $\boldsymbol{r}_i = -\frac{1}{u_i}\log\left(Z(s_i)^{-1}Z(s_{i-1})\right)^\vee$. ∎

In the model introduced in Section 3.2.1, we assume that only the components of $\boldsymbol{\omega_\theta}(v_i)$ which correspond to $\boldsymbol{\theta}(v_i)$ are noisy. Therefore, given the formulation of $\boldsymbol{e}_i$ derived in Proposition 4.3, only these corresponding components in $\boldsymbol{e}_i$ are supposed to be noisy and the other are assumed to be constants equal to zero. Once more, this allows the form of the Frenet-Serret differential equation to be preserved. As a result, we can reduce the error to the 2-dimensional vector $\tilde{\boldsymbol{e}}_i = L\boldsymbol{e}_i \in \mathbb{R}^2$ and its covariance to the 2-dimensional matrix $L^T W_i L = \sigma^2 \tilde{W}_i$ by keeping only the supposed non-zero and random components:

$$\tilde{\boldsymbol{e}}_i = L^T\boldsymbol{e}_i = u_i\left(\boldsymbol{\theta}(v_i) - L^T\boldsymbol{r}_i\right) + O\left(u_i^2\right), \qquad \tilde{W}_i = L^T\int_{s_{i-1}}^{s_i}\phi(s_i,t)LL^T\phi(s_i,t)^T dt L. \tag{4.57}$$

The distribution of $Z_{0:n}$ can then be written using $\tilde{\boldsymbol{e}}_i$ and $\tilde{W}_i$ as

$$p(Z_{0:n};\boldsymbol{\Theta}) \approx \frac{1}{\sqrt{(2\pi)^{6(n+1)}|\boldsymbol{P_0}|\prod_{i=1}^n|\sigma^2\tilde{W}_i|}}\exp\left(-\frac{1}{2}\boldsymbol{e}_0^T{\boldsymbol{P_0}}^{-1}\boldsymbol{e}_0 - \frac{1}{2\sigma^2}\sum_{i=1}^n\tilde{\boldsymbol{e}}_i^T\tilde{W}_i^{-1}\tilde{\boldsymbol{e}}_i\right). \tag{4.58}$$

5.1.2 Error in Observation Model

From the observation model (4.21), the distribution of $\boldsymbol{y}_i$ given $Z(s_i)$ is a multidimensional Gaussian distribution of mean $X(s_i)$ and covariance $\boldsymbol{\Gamma}$,

$$p\left(\boldsymbol{y}_i|X(s_i);\boldsymbol{\Theta}\right) = \frac{1}{\sqrt{(2\pi)^3|\boldsymbol{\Gamma}|}}\exp\left(-\frac{1}{2}\left(\boldsymbol{y}_i - X(s_i)\right)^T\boldsymbol{\Gamma}^{-1}\left(\boldsymbol{y}_i - X(s_i)\right)\right). \tag{4.59}$$

Finally, the complete data log-likelihood is written

$$\begin{aligned}\mathcal{L}(\boldsymbol{\Theta}) = &-\frac{6(n+1)}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{P_0}|) - \frac{1}{2}\log\left(Z(s_0)^{-1}\mu_0\right)^{\vee T}\boldsymbol{P_0}\log\left(Z(s_0)^{-1}\mu_0\right)^{\vee} \\ &- 2n\log(\sigma) - \frac{1}{2}\sum_{i=1}^{n}\log(|\tilde{W}_i|) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}u_i^2(\boldsymbol{\theta}(v_i)-\tilde{\boldsymbol{r}}_i)^T\tilde{W}_i^{-1}(\boldsymbol{\theta}(v_i)-\tilde{\boldsymbol{r}}_i) \\ &- \frac{3n}{2}\log(2\pi) - \frac{n}{2}\log(|\boldsymbol{\Gamma}|) - \frac{1}{2}\sum_{i=1}^{n}\left(\boldsymbol{y}_i - X(s_i)\right)^T\boldsymbol{\Gamma}^{-1}\left(\boldsymbol{y}_i - X(s_i)\right). \quad (4.60)\end{aligned}$$

5.2 Expected Complete Data Log-Likelihood

As the state Z is unknown, we cannot directly use the previously derived log-likelihood function to learn the parameters. We must consider the expected complete data log-likelihood given the observations $\boldsymbol{y}_{1:n}$, and compute the optimal parameter $\boldsymbol{\Theta}$ as solutions of the maximization problem

$$\boldsymbol{\Theta}^{k+1} = \arg\max_{\boldsymbol{\Theta}} \mathbb{E}\left[\mathcal{L}(\boldsymbol{\Theta}) \,\middle|\, \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}^k\right]. \quad (4.61)$$

By considering only the terms of $\mathcal{L}(\boldsymbol{\Theta})$ in (4.60) that depend on at least one parameter of $\boldsymbol{\Theta}$, we can turn the problem into the following minimization problem

$$\boldsymbol{\Theta}^{k+1} = \arg\min_{\boldsymbol{\Theta}} \mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^k) \quad (4.62)$$

where

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^k) = &\log\left(|\boldsymbol{P_0}|\right) + \mathbb{E}\left[\log\left(Z(s_0)^{-1}\boldsymbol{\mu_0}\right)^{\vee T}\boldsymbol{P_0}^{-1}\log\left(Z(s_0)^{-1}\boldsymbol{\mu_0}\right)^{\vee} \,\middle|\, \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}^k\right] \\ &+ 4n\log(\sigma) + \sum_{i=1}^{n}\log(|\tilde{W}_i|) + \frac{1}{\sigma^2}\sum_{i=1}^{n}\mathbb{E}\left[\tilde{\boldsymbol{e}}_i^T\tilde{W}_i^{-1}\tilde{\boldsymbol{e}}_i \,\middle|\, \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}^k\right] \\ &+ n\log(|\boldsymbol{\Gamma}|) + \sum_{i=1}^{n}\mathbb{E}\left[\left(\boldsymbol{y}_i - X(s_i)\right)^T\boldsymbol{\Gamma}^{-1}\left(\boldsymbol{y}_i - X(s_i)\right) \,\middle|\, \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}^k\right]. \quad (4.63)\end{aligned}$$

Remark 4.2. *In order to compute these conditional expectations, we recall the following general formula of the expectation of a quadratic form using the trace application (Kendrick, 1981), which will be useful in the following:*

$$\mathbb{E}\left[X^T W^{-1} X \,|Y\right] = \mathbb{E}\left[X\,|Y\right]^T W^{-1}\mathbb{E}\left[X\,|Y\right] + Tr\left(W^{-1}\mathbb{V}\left[X\,|Y\right]\right). \quad (4.64)$$

Given the formula (4.64) we must be able to compute the terms $\mathbb{E}\left[\log\left(Z(s_0)^{-1}\boldsymbol{\mu_0}\right)^{\vee}|\boldsymbol{y}_{1:n}\right]$, $\mathbb{E}\left[\tilde{\boldsymbol{e}}_i\,|\boldsymbol{y}_{1:n}\right]$, and $\mathbb{E}\left[\left(\boldsymbol{y}_i - X(s_i)\right)|\boldsymbol{y}_{1:n}\right]$ in order to derive a convenient expression of $\mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^k)$ for the estimation of optimal parameters.

5.2.1 Conditional Expectation on Propagation Model

Let's look at the conditional distribution of the propagation model error $\tilde{\boldsymbol{e}}_i$. From (4.57) we have,

$$\mathbb{E}\left[\tilde{\boldsymbol{e}}_i\,|\boldsymbol{y}_{1:n}\right] = u_i\left(\boldsymbol{\theta}(v_i) - L^T\mathbb{E}\left[\boldsymbol{r}_i\,|\boldsymbol{y}_{1:n}\right]\right), \quad (4.65)$$

where $\boldsymbol{r}_i = -\frac{1}{u_i}\log\left(Z(s_i)^{-1}Z(s_{i-1})\right)^{\vee}$. From Section 4 we have obtained $p(Z(s)|\boldsymbol{y}_{1:n}) = \mathcal{N}_{SE}(\boldsymbol{\mu}^{\boldsymbol{s}}(s), \boldsymbol{P}^{\boldsymbol{s}}(s))$. However, $\boldsymbol{r}_i$ is a non-linear function of the compounded variables $Z(s_i)^{-1}Z(s_{i-1})$, and approximate the

distribution of the compound of two concentrated Gaussian random variables on a Lie group is a complex problem investigated by Wang and Chirikjian (2008); Barfoot and Furgale (2014). We propose in this work to derive an approximation of the variable $\boldsymbol{r}_i$ as a function of the smoothing mean parameter $\boldsymbol{\mu}^{\boldsymbol{s}}$ that is more convenient for computing the conditional expected value.

Proposition 4.4. *The conditional distribution of the propagation error $\tilde{\boldsymbol{e}}_i$ defined in (4.57) can be approximated as*

$$\mathbb{E}\left[\tilde{\boldsymbol{e}}_i \,|\boldsymbol{y}_{1:n}\right] \approx u_i \left(\boldsymbol{\theta}(s_i) - L^T \bar{\boldsymbol{r}}_i\right), \quad \mathbb{V}\left[\tilde{\boldsymbol{e}}_i \,|\boldsymbol{y}_{1:n}\right] \approx L^T \mathbb{V}\left[\boldsymbol{\eta}_i \,|\boldsymbol{y}_{1:n}\right] L, \tag{4.66}$$

where $\bar{\boldsymbol{r}}_i := -\frac{1}{u_i}\log\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^{\vee} \in \mathbb{R}^6$ is a deterministic parameter and $\boldsymbol{\eta}_i \in \mathbb{R}^6$ is a random variable which depends on the hidden states $Z(s_{i-1}), Z(s_i)$, defined as

$$\boldsymbol{\eta}_i := \log\left(Z(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\right)^{\vee} - Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\log\left(Z(s_{i-1})^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^{\vee}, \tag{4.67}$$

with mean zero and conditional variance

$$\begin{aligned}\mathbb{V}\left[\boldsymbol{\eta}_i \,|\boldsymbol{y}_{1:n}\right] = \boldsymbol{P}^{\boldsymbol{s}}(s_i) - Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\boldsymbol{P}^{\boldsymbol{s}}(s_{i-1}, s_i) - \boldsymbol{P}^{\boldsymbol{s}}(s_{i-1}, s_i)^T Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^T \\ + Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\boldsymbol{P}^{\boldsymbol{s}}(s_{i-1})Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^T.\end{aligned} \tag{4.68}$$

Proof. We have,

$$\begin{aligned}\boldsymbol{r}_i &= -\frac{1}{u_i}\log\left(Z(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})^{-1}Z(s_{i-1})\right)^{\vee} \\ &\approx -\frac{1}{u_i}\log\left(\left(Z(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\right)\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\exp\left(-\log\left(Z(s_{i-1})^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\right)\right)^{\vee} \\ &= -\frac{1}{u_i}\log\left(\left(Z(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\right)\exp\left(-Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\log\left(Z(s_{i-1})^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\right)\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\right)^{\vee} \\ &= -\frac{1}{u_i}\log\left(\exp\left(\hat{\boldsymbol{\eta}}_i^{\wedge}\right)\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\right)^{\vee}\end{aligned}$$

where

$$\exp\left(\hat{\boldsymbol{\eta}}_i^{\wedge}\right) = \exp\left(\log\left(Z(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\right)\right)\exp\left(-Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\log\left(Z(s_{i-1})^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\right). \tag{4.69}$$

We can apply the BCH formula to find

$$\begin{aligned}\hat{\boldsymbol{\eta}}_i = \log\left(Z(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\right)^{\vee} - Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\log\left(Z(s_{i-1})^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^{\vee} \\ - \frac{1}{2}\left(\log\left(Z(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\right)^{\vee}\right)^{\curlyvee} Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\log\left(Z(s_{i-1})^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^{\vee} + ...\end{aligned}$$

By considering only the first-order term we have

$$\hat{\boldsymbol{\eta}}_i = \boldsymbol{\eta}_i + O\left(\|\log\left(Z(s)\boldsymbol{\mu}^{\boldsymbol{s}}(s)\right)^{\vee}\|^2\right), \tag{4.70}$$

where the random variable $\boldsymbol{\eta}_i$ is defined by

$$\boldsymbol{\eta}_i := \log\left(Z(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\right)^{\vee} - Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\log\left(Z(s_{i-1})^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^{\vee}.$$

The random variable $\boldsymbol{\eta}_i$ is a linear function of the random variable $\log\left(Z(s)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s)\right)^{\vee}$, for which we have the complete continuous posterior distribution given the observations $\boldsymbol{y}_{1:n}$, obtained in the inference step (Section 4). Its expected mean and covariance are given by the following equations:

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{\eta}_i \,|\boldsymbol{y}_{1:n}\right] &= \mathbb{E}\left[\log\left(Z(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\right)^{\vee}|\boldsymbol{y}_{1:n}\right] - Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\mathbb{E}\left[\log\left(Z(s_{i-1})^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^{\vee}|\boldsymbol{y}_{1:n}\right] \\ &= \mathbf{0}_{\mathbb{R}^6}\end{aligned}$$

$$\begin{aligned}\mathbb{V}\left[\boldsymbol{\eta}_i \,|\boldsymbol{y}_{1:n}\right] = \boldsymbol{P}^{\boldsymbol{s}}(s_i) - Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\boldsymbol{P}^{\boldsymbol{s}}(s_{i-1},s_i) - \boldsymbol{P}^{\boldsymbol{s}}(s_{i-1},s_i)^T Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^T \\ + Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)\boldsymbol{P}^{\boldsymbol{s}}(s_{i-1})Ad\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^T.\end{aligned}$$

Then, by applying again the BCH formula on the expression of $\boldsymbol{r_i}$ we obtain the following approximation by keeping only the first-order terms,

$$\boldsymbol{r}_i = -\frac{1}{u_i}\log\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^{\vee} - \frac{1}{u_i}\boldsymbol{\eta}_i + O\left(\|\log\left(Z(s)\boldsymbol{\mu}^{\boldsymbol{s}}(s)\right)^{\vee}\|^2\right),$$

corresponding to a deterministic mean term $\bar{\boldsymbol{r}}_i := -\frac{1}{u_i}\log\left(\boldsymbol{\mu}^{\boldsymbol{s}}(s_i)^{-1}\boldsymbol{\mu}^{\boldsymbol{s}}(s_{i-1})\right)^{\vee} \in \mathbb{R}^6$ plus the random error term $-\frac{1}{u_i}\boldsymbol{\eta}_i \in \mathbb{R}^6$. Therefore, the propagation error $\tilde{\boldsymbol{e}}_i$ defined in (4.57) can be approximated as

$$\tilde{\boldsymbol{e}}_i = u_i\left(\boldsymbol{\theta}(s_i) - L^T\bar{\boldsymbol{r}}_i\right) + L^T\boldsymbol{\eta}_i + O\left(\|\log\left(Z(s)\boldsymbol{\mu}^{\boldsymbol{s}}(s)\right)^{\vee}\|^2\right), \tag{4.71}$$

which gives the approximated expectation and variance formulas stated in the proposition. ■

Remark 4.3. *Considering the variable $\hat{\boldsymbol{\eta}}_i$ defined in (4.69), one could explore deriving a Monte Carlo approximation for its conditional distribution to directly estimate the conditional distribution of the propagation error. In practice, we find that this method yields results quite comparable to the analytical approach presented in the proof of the Proposition 4.4. Nevertheless, for computational efficiency, we opt for the analytical approximation in our experiments.*

5.2.2 Conditional Expectation on the Observation Model

The approximation used for the conditional distribution of the observation error in (4.63) is derived in the following proposition.

Proposition 4.5. *The conditional distribution of the observation error $\boldsymbol{y}_i - X(s_i) \in \mathbb{R}^3$ of the model (4.24), given the observations $\boldsymbol{y}_{1:n}$, can be approximated as*

$$\mathbb{E}\left[\boldsymbol{y}_i - X(s_i)\,|\boldsymbol{y}_{1:n}\right] = \boldsymbol{y}_i - \boldsymbol{\mu}^{\boldsymbol{s}}_{\boldsymbol{X}}(s_i), \qquad \mathbb{V}\left[\boldsymbol{y}_i - X(s_i)\,|\boldsymbol{y}_{1:n}\right] = \boldsymbol{\mu}^{\boldsymbol{s}}_{\boldsymbol{Q}}(s_i)H\boldsymbol{P}^{\boldsymbol{s}}(s_i)H^T\boldsymbol{\mu}^{\boldsymbol{s}}_{\boldsymbol{Q}}(s_i)^T. \tag{4.72}$$

Proof. We first consider the error in $\mathbb{R}^4$ as a function of $Z(s_i)|\boldsymbol{y}_{1:n}$ and a first-order approximation of the exponential map as done previously in Section 4,

$$\begin{aligned}\tilde{\boldsymbol{y}}_i - Z_{i|n}\boldsymbol{u} &= \tilde{\boldsymbol{y}}_i - \boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\exp\left(-\boldsymbol{\xi}^{\boldsymbol{s}}(s_i)^{\wedge}\right)\boldsymbol{u} = \tilde{\boldsymbol{y}}_i - \boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\left(I - \boldsymbol{\xi}^{\boldsymbol{s}}(s_i)^{\wedge}\right)\boldsymbol{u} + O\left(\|\boldsymbol{\xi}^{\boldsymbol{s}}(s_i)^{\wedge}\|^2\right) \\ &= \tilde{\boldsymbol{y}}_i - \boldsymbol{\mu}^{\boldsymbol{s}}(s_i) + \boldsymbol{\mu}^{\boldsymbol{s}}(s_i)\boldsymbol{\xi}^{\boldsymbol{s}}(s_i)^{\wedge}\boldsymbol{u} + O\left(\|\boldsymbol{\xi}^{\boldsymbol{s}}(s_i)^{\wedge}\|^2\right).\end{aligned}$$

The first 3 components of the error are then

$$\boldsymbol{y}_i - X_{i|n} = \boldsymbol{y}_i - \boldsymbol{\mu}^{\boldsymbol{s}}_{\boldsymbol{X}}(s_i) + \boldsymbol{\mu}^{\boldsymbol{s}}_{\boldsymbol{Q}}(s_i)H\boldsymbol{\xi}^{\boldsymbol{s}}(s_i) + O\left(\|\boldsymbol{\xi}^{\boldsymbol{s}}(s_i)^{\wedge}\|^2\right), \tag{4.73}$$

where $H = \begin{pmatrix} 0_{3\times 3} & I_3 \end{pmatrix}$ and $\boldsymbol{\xi}^{\boldsymbol{s}}(s_i)|\boldsymbol{y}_{1:n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{P}^{\boldsymbol{s}}(s_i))$. ■

Finally, given the equations (4.64), (4.66), (4.68) and (4.72), the function $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^k)$, which corresponds to the relevant terms of the expected complete data log-likelihood, is written as

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^k) &= \log\left(|\boldsymbol{P_0}|\right) + \log\left(\boldsymbol{\mu^s}(s_0)^{-1}\boldsymbol{\mu_0}\right)^{\vee T} \boldsymbol{P_0}^{-1} \log\left(\boldsymbol{\mu^s}(s_0)^{-1}\boldsymbol{\mu_0}\right)^{\vee} + \operatorname{Tr}\left(\boldsymbol{P_0}^{-1}\boldsymbol{P^s}(s_0)\right) \\ &+ 4n\log(\sigma) + \sum_{i=1}^{n}\log(|\tilde{W}_i|) + \frac{1}{\sigma^2}\sum_{i=1}^{n} u_i^2(\boldsymbol{\theta}(v_i) - L^T\bar{\boldsymbol{r}}_i)^T\tilde{W}_i^{-1}(\boldsymbol{\theta}(v_i) - L^T\bar{\boldsymbol{r}}_i) + \operatorname{Tr}\left(\tilde{W}_i^{-1}L^T\mathbb{V}\left[\boldsymbol{\eta}_i \,|\boldsymbol{y}_{1:n}\right]L\right) \\ &+ n\log(|\boldsymbol{\Gamma}|) + \sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{\mu^s_X}(s_i))^T\boldsymbol{\Gamma}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu^s_X}(s_i)) + \operatorname{Tr}\left(\boldsymbol{\Gamma}^{-1}\boldsymbol{\mu^s_Q}(s_i)H\boldsymbol{P^s}(s_i)H^T\boldsymbol{\mu^s_Q}(s_i)^T\right). \quad (4.74)\end{aligned}$$

5.3 Learning Optimal Parameters

In the set of parameters to optimize, we have the Frenet curvatures functional parameter $\boldsymbol{\theta}$ that is infinite-dimensional and assumed to be smooth. To impose the desired smoothness constraint on this specific parameter, a commonly used approach is to introduce a penalty term in the likelihood function, known as penalized likelihood (Liu et al., 2009; Cardot and Sarda, 2005; Du and Wang, 2014; Cole et al., 2013). By incorporating this penalty into the objective function, the estimation of parameters is influenced to align with the desired properties, leading to improved accuracy and reliability in inference. This allows for control over the complexity of the model and helps prevent overfitting by encouraging more appropriate parameter estimates (Ramsay and Silverman, 2005). We choose as the roughness function the integrated squared second derivative of $\boldsymbol{\theta}$ scaled by σ^2,

$$\mathcal{R}(\boldsymbol{\Theta}) = \frac{1}{\sigma^2}\int_0^1 \|D^2\boldsymbol{\theta}(s)\|^2 ds = \frac{1}{\sigma^2}\int_0^1 \left([D^2\kappa(s)]^2 + [D^2\tau(s)]^2\right) ds. \quad (4.75)$$

Note that this is equivalent to penalizing the two components of $\boldsymbol{\theta}(s) = (\kappa(s), \tau(s))^T$ separately. We then consider two regularization parameters to allow for adjusting the smoothing rate of each Frenet curvature independently,

$$\mathcal{R}_\lambda(\boldsymbol{\Theta}) = \frac{\lambda_1}{\sigma^2}\int_0^1 [D^2\kappa(s)]^2 ds + \frac{\lambda_2}{\sigma^2}\int_0^1 [D^2\tau(s)]^2 ds\,, \quad (4.76)$$

where $\lambda = (\lambda_1, \lambda_2) \in {\mathbb{R}_+^*}^2$. Our minimization problem is therefore relaxed to the minimization of the following penalized criterion

$$\tilde{\mathcal{Q}}_\lambda(\boldsymbol{\Theta}; \boldsymbol{\Theta}^k) = \mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^k) + \mathcal{R}_\lambda(\boldsymbol{\Theta}). \quad (4.77)$$

5.3.1 Learning Propagation Model Parameters

The propagation model parameters are $\{\boldsymbol{\mu_0}, \boldsymbol{P_0}, \boldsymbol{\theta}, \sigma\}$. First, setting the gradients of $\tilde{\mathcal{Q}}_\lambda(\boldsymbol{\mu_0}, \boldsymbol{P_0}; \boldsymbol{\Theta}^k)$ with respect to $\boldsymbol{\mu_0}$ and $\boldsymbol{P_0}$ to zero respectively, gives the optimal values

$$\boldsymbol{\mu_0}^{k+1} = \boldsymbol{\mu^s}(s_0), \qquad \boldsymbol{P_0}^{k+1} = \boldsymbol{P^s}(s_0). \quad (4.78)$$

Next, the computation of the optimal variance parameter σ relies on the determination of the unknown mean parameter $\boldsymbol{\theta}$, necessitating its estimation as a preliminary step. This last one is supposed to be a smooth vector function in $\mathbb{R}^2$. Therefore, we decomposed each scalar function Frenet curvature composing $\boldsymbol{\theta}$ using a B-spline basis function expansion (Section 2.1.1),

$$\kappa(s) = \sum_{k=1}^{K_1} c_k^{(1)}\Phi_k^{(1)}(s), \qquad \tau(s) = \sum_{k=1}^{K_2} c_k^{(2)}\Phi_k^{(2)}(s). \quad (4.79)$$

In practice, we choose the same number of basis functions for each component, $K_1 = K_2 = K$, as we have the same number of observations and have already considered two different smoothing parameters. Then the vector-valued function $\boldsymbol{\theta}$ can be expressed in a matrix form as

$$\boldsymbol{\theta}(s) = \Phi^T(s)\boldsymbol{c}\,, \tag{4.80}$$

where $\Phi(s)$ is the block diagonal matrix $\Phi(s) = diag\left(\Phi^{(1)}(s), \Phi^{(2)}(s)\right) \in \mathbb{R}^{2K\times 2}$, with for $i = 1, 2$, $\Phi^{(i)}(s) = \left(\Phi_1^{(i)}(s), \ldots, \Phi_K^{(i)}(s)\right)^T \in \mathbb{R}^K$ are the known B-spline basis functions vectors of each component, and the coefficient vector $\boldsymbol{c} = \left(c_1^{(1)}, \ldots, c_K^{(1)}, c_1^{(2)}, \ldots, c_K^{(2)}\right)^T \in \mathbb{R}^{2K}$ is the concatenation of the unknown basis coefficients. The roughness penalty term (4.76) can then be written also in matrix form as $\boldsymbol{c}^T\boldsymbol{R}(\lambda)\boldsymbol{c}$, where $\boldsymbol{R}(\lambda)$ is $2K \times 2K$ matrix defined as

$$\boldsymbol{R}(\lambda) = \begin{pmatrix} \lambda_1 \int_0^1 D^2\Phi^{(1)}(t)D^2\Phi^{(1)}(t)^T dt & 0 \\ 0 & \lambda_2 \int_0^1 D^2\Phi^{(2)}(t)D^2\Phi^{(2)}(t)^T dt \end{pmatrix}. \tag{4.81}$$

The minimization problem of $\boldsymbol{\theta}$ and σ is now a minimization problem over $\boldsymbol{c}$ and σ of the criterion

$$\begin{aligned}\tilde{\mathcal{Q}}_\lambda(\boldsymbol{c}, \sigma; \boldsymbol{\Theta}^k) &= 4n\log(\sigma) + \sum_{i=1}^n \log(|\tilde{W}_i|) \\ &+ \frac{1}{\sigma^2}\left[\sum_{i=1}^n (\Phi(v_i)^T\boldsymbol{c} - L^T\bar{\boldsymbol{r}}_i)^T u_i^2 \tilde{W}_i^{-1}(\Phi(v_i)^T\boldsymbol{c} - L^T\bar{\boldsymbol{r}}_i) + \mathrm{Tr}\left(\tilde{W}_i^{-1}L^T\mathbb{V}\left[\boldsymbol{\eta}_i \,|\boldsymbol{y}_{1:n}\right]L\right) + \boldsymbol{c}^T\boldsymbol{R}(\lambda)\boldsymbol{c}\right].\end{aligned} \tag{4.82}$$

It is crucial to highlight that the weights $\tilde{W}_i$, defined in (4.57) as

$$\tilde{W}_i = L^T \int_{s_{i-1}}^{s_i} \phi(s_i, t)LL^T\phi(s_i, t)^T dt L\,,$$

are dependent on the mean parameter $\boldsymbol{\theta}$ (due to the transition function $\phi(s_i, t)$), and consequently, on the coefficients $\boldsymbol{c}$. This places us within the context of a generalized linear model (McCullagh and Nelder, 1960). In this scenario, the prevailing approach to optimizing the mean parameter involves an iterative algorithm that alternates between computing the weights and optimizing the coefficients, assuming the weights to be fixed. Given the estimation of coefficients, the weights $\tilde{W}_i$ can be computed using their formulas (4.57). In practice, we observe that a simple middle point approximation of the integral in the expression of $\tilde{W}_i$ is already sufficient. Then, given the weights and fixed smoothing parameters λ, the optimal coefficients $\hat{\boldsymbol{c}}_\lambda$ are computed by

$$\hat{\boldsymbol{c}}_\lambda = \left(\boldsymbol{\Phi}^T\tilde{\boldsymbol{W}}\boldsymbol{\Phi} + \boldsymbol{R}(\lambda)\right)^{-1}\boldsymbol{\Phi}^T\tilde{\boldsymbol{W}}\hat{\boldsymbol{r}}\,, \tag{4.83}$$

where $\tilde{\boldsymbol{W}} = blockdiag\left(u_1^2\tilde{W}_1^{-1}, \ldots, u_n^2\tilde{W}_n^{-1}\right)$ and $\hat{\boldsymbol{r}} = \left(\bar{\boldsymbol{r}}_1^T L, \ldots, \bar{\boldsymbol{r}}_n^T L\right)^T \in \mathbb{R}^{2n}$. Then, given the optimal coefficients $\hat{\boldsymbol{c}}_\lambda$, we minimize (4.82) with respect to σ, and we obtain $\hat{\sigma}_\lambda$ as an implicit function of λ and $\hat{\boldsymbol{c}}_\lambda$,

$$\hat{\sigma}_\lambda^2 = \frac{1}{2n}\left[\sum_{i=1}^n (\Phi(v_i)^T\hat{\boldsymbol{c}}_\lambda - L^T\bar{\boldsymbol{r}}_i)^T u_i^2\tilde{W}_i^{-1}(\Phi(v_i)^T\hat{\boldsymbol{c}}_\lambda - L^T\bar{\boldsymbol{r}}_i) + \mathrm{Tr}\left(\tilde{W}_i^{-1}L^T\mathbb{V}\left[\boldsymbol{\eta}_i \,|\boldsymbol{y}_{1:n}\right]L\right) + \hat{\boldsymbol{c}}_\lambda^T\boldsymbol{R}(\lambda)\hat{\boldsymbol{c}}_\lambda\right]. \tag{4.84}$$

The next optimal parameters are then

$$\boldsymbol{\theta}^{k+1}(s) = \Phi(s)^T\hat{\boldsymbol{c}}_\lambda, \qquad \boldsymbol{\Sigma}^{k+1} = \hat{\sigma}_\lambda I. \tag{4.85}$$

5.3.2 Learning Observation Model Parameter

We still have to compute the optimal observations covariance matrix $\mathbf{\Gamma}$. Keeping only the terms of $\tilde{\mathcal{Q}}_\lambda(\mathbf{\Theta}; \mathbf{\Theta}^k)$ that depend on it, we have

$$\tilde{\mathcal{Q}}_\lambda(\mathbf{\Gamma}; \mathbf{\Theta}^k) = n \log(|\mathbf{\Gamma}|) + \sum_{i=1}^{n} (\boldsymbol{y}_i - \boldsymbol{\mu}_{\boldsymbol{X}}^{\boldsymbol{s}}(s_i))^T \mathbf{\Gamma}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_{\boldsymbol{X}}^{\boldsymbol{s}}(s_i)) + \operatorname{Tr}\left(\mathbf{\Gamma}^{-1} \boldsymbol{\mu}_{\boldsymbol{Q}}^{\boldsymbol{s}}(s_i) H \boldsymbol{P}^{\boldsymbol{s}}(s_i) H^T \boldsymbol{\mu}_{\boldsymbol{Q}}^{\boldsymbol{s}}(s_i)^T \right). \tag{4.86}$$

Therefore, by setting its gradient with respect to $\mathbf{\Gamma}$ to zero, the optimal observations covariance matrix is

$$\mathbf{\Gamma}^{k+1} = \frac{1}{n} \sum_{i=1}^{n} \left[(\boldsymbol{y}_i - \boldsymbol{\mu}_{\boldsymbol{X}}^{\boldsymbol{s}}(s_i))(\boldsymbol{y}_i - \boldsymbol{\mu}_{\boldsymbol{X}}^{\boldsymbol{s}}(s_i))^T + \boldsymbol{\mu}_{\boldsymbol{Q}}^{\boldsymbol{s}}(s_i) H \boldsymbol{P}^{\boldsymbol{s}}(s_i) H^T \boldsymbol{\mu}_{\boldsymbol{Q}}^{\boldsymbol{s}}(s_i)^T \right]. \tag{4.87}$$

5.4 Optimization of the Smoothing Parameter

The optimal selection of the hyperparameter $\lambda = (\lambda_1, \lambda_2)$ is crucial in minimizing our prediction error. To achieve this, the commonly used criterion for smoothing parameter optimization is the *generalized cross-validation* (GCV). However, in our specific case, the GCV score or its extensions tailored for dependent noise tend to favor excessively small values of λ due to the inherent smoothness of our observations. Consequently, in our numerical studies, we have employed k-fold cross-validation as an alternative, aiming to minimize

$$\sum_{j=1}^{k} \sum_{i \in T_j} \left\| \boldsymbol{y}_i - \hat{X}^{-(j)}(s_i; \lambda) \right\|^2, \tag{4.88}$$

where T_j is the jth index set based on $k = 5$ random partition of the observations $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$, and $\hat{X}^{-(j)}(s_i; \lambda)$ is the predicted curve, defined as the Euclidean part of the solution of the ODE (4.20), solved with parameters $\hat{\boldsymbol{\theta}}^{-(j)}$ and the initial value $\hat{\boldsymbol{\mu}}_{\mathbf{0}}{}^{-(j)}$, which are estimated by the EM algorithm without the jth partition data set and using the hyperparameters λ. Furthermore, this criterion is more consistent with the theoretical foundations of our problem, as it corresponds to the definition provided by the initial ordinary differential equation (4.20). However, with this parameter optimization method for λ, for each candidate value, the EM algorithm must be run k times (the number of partitions in the cross-validation). Thus, this significantly increases the computation time.

6 Simulation Studies

To assess the performance of the proposed algorithm and the influence of the individual parameters, we have run simulations on synthetic data. For comparison, we use the same reference shape as in the simulations of Chapter 3, which is $s \mapsto \boldsymbol{\theta}(s) = (\kappa(s), \tau(s))$, where

$$\begin{cases} \kappa(s) = & 2\cos(2\pi s) + 5\,, \\ \tau(s) = & 2\sin(2\pi s) + 1\,, \end{cases} \tag{4.89}$$

and the arc-length function defined from $[0,1]$ to $[0,1]$ as $s : t \mapsto \frac{\exp(at)-1}{\exp(a)-1}$, where $a \sim \mathcal{N}(0,1), a \neq 0$.

An important point with the EM algorithm is that it requires an initial value for the various parameters. Moreover, the initialization of parameters must be carefully handled, as it can considerably speed up the convergence of the EM. Here the parameters to initialize from available observations of the Euclidean curve $\{\boldsymbol{y}_i\}_{i=1,\ldots,n}$ are $\mathbf{\Theta} = \{\boldsymbol{\theta}, \mathbf{\Sigma}, \mathbf{\Gamma}, \boldsymbol{P}_{\mathbf{0}}, \boldsymbol{\mu}_{\mathbf{0}}\}$. In addition, we still have assumed, in this algorithm, that the arc-length grid is known, so it must be estimated as a preprocessing step as well.

Using a local polynomial regression of $\{\boldsymbol{y}_i\}_{i=1,\dots,n}$, introduced in Section 2.1, one can obtain a reasonably reliable and smooth estimate of the arc-length function and the shape of the curve $\hat{X}^{(0)}$. Then, the covariance $\boldsymbol{\Gamma}$ of the observation noise can be initialized as

$$\boldsymbol{\Gamma}^{(0)} = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{y}_i - \hat{X}^{(0)}(s_i)\right)\left(\boldsymbol{y}_i - \hat{X}^{(0)}(s_i)\right)^T. \tag{4.90}$$

To initialize the initial value $\boldsymbol{\mu}_0$, we can simply use the Gram-Schmidt orthonormalization process of the estimated derivatives at $s = 0$. The initial covariance matrix $\boldsymbol{P}_0$, on the other hand, cannot be properly estimated from the data and will be thus set to a chosen value. The same applies to the variance of the propagation model σ as it accounts for the misspecification of the model due to the unknown parameter $\boldsymbol{\theta}$. However, if we know the performance of the algorithm chosen to initialize $\boldsymbol{\theta}$, it can serve as a reference for selecting the associated initial value $\sigma^{(0)}$. Specifically, the initialization of Frenet curvatures $\boldsymbol{\theta}$ can typically be done by using one of the algorithms proposed in Chapter 3.

All simulation models are repeated 80 times. We select the optimal hyperparameters $\lambda_1, \lambda_2 \in [1e^{-09}, 1e^{-03}]$, with 30 iterations of Bayesian optimization based on the cross-validation criterion (4.90). The 80 simulations are run in parallel. The computation time, using a machine equipped with 96 CPUs and 376 GiB of RAM, is around 27 hours for a curve with 100 sample points. To evaluate the performance of the proposed EM algorithm we use similar metrics as in Chapter 3, that is, the geodesic distance $\mathbf{d}$ on $SE(3)$ defined in (4.5) and the $\mathbb{L}^2$ relative error $\mathbf{re}$ defined in equation (3.65). We compare the true state variable Z with the estimated one by tracking and smoothing $\hat{Z}$. Then, we compare the true Frenet curvatures κ and τ with the estimates $\hat{\kappa}$ and $\hat{\tau}$. As an additional measure of the accuracy of Frenet curvatures estimation, we assess the discrepancy between the reconstructed state using estimated parameters $\hat{Z}^{(\hat{\theta})}$ and the others.

6.1 Simulation Model with Deterministic State Variable

Given the reference shape, $s \mapsto \boldsymbol{\theta}(s)$ and $t \mapsto s(t)$, we generate noisy observations of the Euclidean curve as

$$\boldsymbol{y}_i = X(s_i) + \boldsymbol{\epsilon}_i, \quad i = 1, \dots, n\,,$$

where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \gamma^2 \boldsymbol{I}_3)$ and the shape function $s \mapsto X(s)$ is part of the state variable $Z(s)$ solution of the deterministic ODE (4.20), $Z'(s) = Z(s)\boldsymbol{\omega}_{\boldsymbol{\theta}}(s)^\wedge$.

6.1.1 Scenario 1: Influence of Variance Parameter σ Initialization

In this scenario, we want to test the influence of the chosen initial value for the variance parameter σ when the model to initialize $\boldsymbol{\theta}$ is fixed. Hence, given the true parameter $\boldsymbol{\theta}$, we simulate a smooth noisy initial guess $\hat{\boldsymbol{\theta}}^{(0)}$ using the model:

$$\hat{\boldsymbol{\theta}}^{(0)} = \underset{\hat{\boldsymbol{\theta}} \in \mathcal{H}}{\arg\min} \sum_{i=0}^{n} \|\boldsymbol{\theta}(s_i) + \boldsymbol{\zeta}_i - \hat{\boldsymbol{\theta}}(s_i)\|_2^2 + \lambda \int_0^1 \|\hat{\boldsymbol{\theta}}''(t)\|^2 dt\,, \tag{4.91}$$

where $\boldsymbol{\zeta}_i \sim \mathcal{N}(0, \mathbf{I}_2)$ and $\lambda = 10^{-8}$. We fix $\hat{\boldsymbol{P}}_0^{(0)} = (0.01)^2\mathbf{I}_6$ and $\hat{\boldsymbol{\mu}}_0^{(0)}$ and $\hat{\boldsymbol{\Gamma}}^{(0)}$ are initialized as proposed above. We fix $n = 100$, $\gamma = 0.001$, and the number of basis functions to 10. Then, we run the proposed EM algorithm for the values of $\sigma^{(0)} = 0.001, 0.01, 0.1$. Results are displayed in Table 4.1 and Figure 4.1. We observe quite similar errors as a function of the value of $\sigma^{(0)}$, which, therefore, does not seem to have a strong influence on the final estimate in this case.

$\sigma^{(0)}$	$\mathbf{d}(Z, \hat{Z})$	$\mathbf{re}(\kappa, \hat{\kappa})$	$\mathbf{re}(\tau, \hat{\tau})$	$\mathbf{d}(Z, \hat{Z}^{(\hat{\theta})})$	$\mathbf{d}(\hat{Z}, \hat{Z}^{(\hat{\theta})})$
0.001	0.021 (0.008)	0.031 (0.013)	0.132 (0.058)	0.038 (0.010)	0.0012 (0.0006)
0.01	**0.017** (0.005)	**0.025** (0.010)	**0.111** (0.041)	**0.035** (0.008)	0.0012 (0.0004)
0.1	**0.017** (0.005)	0.026 (0.010)	0.118 (0.052)	**0.035** (0.009)	**0.0011** (0.0003)

Table 4.1: Estimation errors of scenario 1: influence of variance parameter initialization.

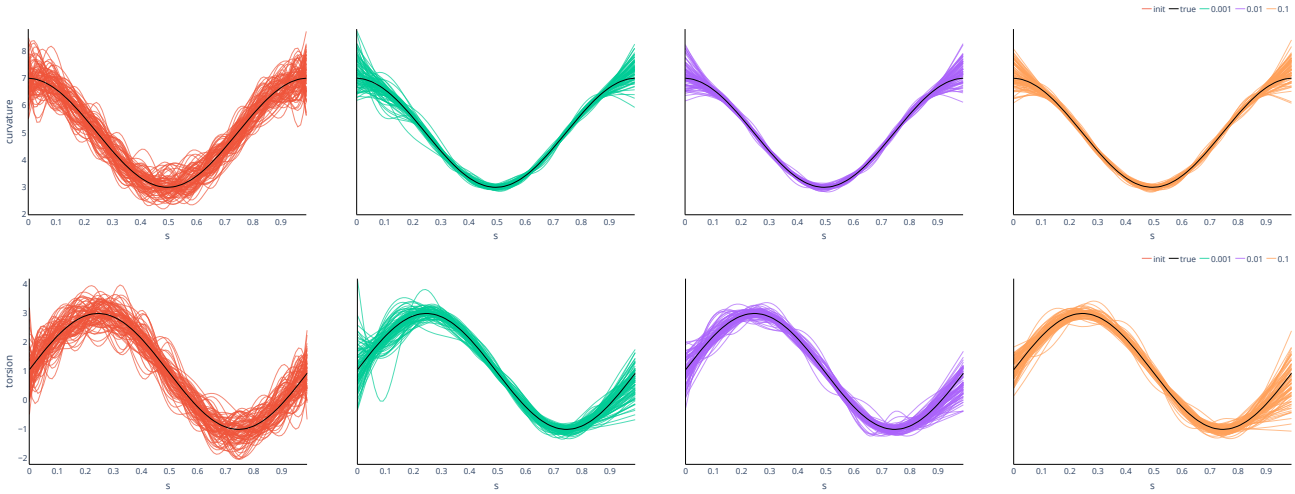


Figure 4.1: Estimated curvature (first row) and torsion (second row) in scenario 1 (Influence of variance parameter σ initialization): comparison between true parameter (black), initial guess (red), estimates using $\sigma^{(0)} = 0.001$ (green), $\sigma^{(0)} = 0.01$ (purple) and $\sigma^{(0)} = 0.1$ (orange).

6.1.2 Scenario 2: Influence of Sample Size and Noise Level

Let's now test the performance of the proposed EM algorithm as a function of the sample size and the noise level. For that, we consider the same generative model for $\hat{\boldsymbol{\theta}}^{(0)}$ as before (4.91). We fix $\sigma^{(0)} = 0.05$, $\hat{\boldsymbol{P}}_0^{(0)} = (0.01)^2 \mathbf{I}_6$ and again $\hat{\boldsymbol{\mu}}_0^{(0)}$ and $\hat{\boldsymbol{\Gamma}}^{(0)}$ are initialized as proposed above. We run the algorithm for the different combinations of sample size $n = 100, 200$ and noise level $\gamma = 0.001, 0.005$. The number of basis function for $\hat{\boldsymbol{\theta}}$ is $\lceil \sqrt{n} \rceil = 10, 15$. Results are printed in Table 4.2 and plotted in Figure 4.2. We observe that varying the number of points has no significant impact on the results. On the other hand, the higher the initial noise level of the trajectory, the less satisfactory the final result, which is coherent. Additionally, we have set a maximum of 200 iterations for the EM algorithm due to computational time constraints, which is almost always reached. Therefore, presumably, with more iterations, these errors could be further reduced.

n	γ	$\mathbf{d}(Z, \hat{Z})$	$\mathbf{re}(\kappa, \hat{\kappa})$	$\mathbf{re}(\tau, \hat{\tau})$	$\mathbf{d}(Z, \hat{Z}^{(\hat{\theta})})$	$\mathbf{d}(\hat{Z}, \hat{Z}^{(\hat{\theta})})$
100	0.001	0.017 (0.005)	0.023 (0.010)	0.120 (0.053)	0.035 (0.008)	0.001 (0.0003)
	0.005	0.042 (0.012)	0.043 (0.019)	0.163 (0.066)	0.053 (0.011)	0.001 (0.0003)
200	0.001	0.014 (0.004)	0.021 (0.008)	0.109 (0.048)	0.034 (0.008)	0.001 (0.0002)
	0.005	0.038 (0.011)	0.040 (0.015)	0.171 (0.083)	0.049 (0.012)	0.001 (0.0002)

Table 4.2: Estimation errors of scenario 2: influence of sample size and noise level.

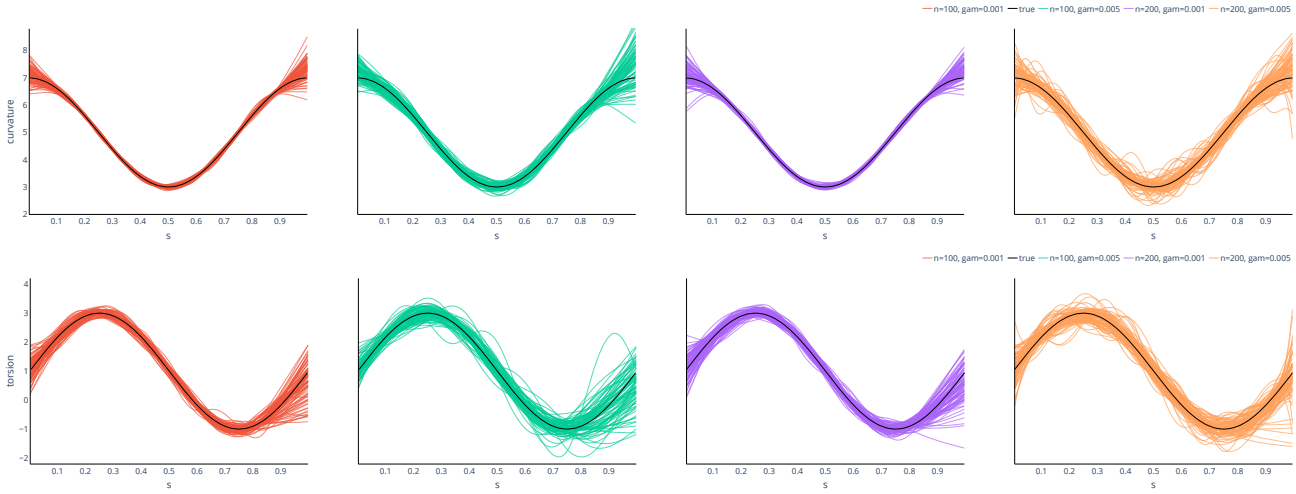


Figure 4.2: Estimated curvature (first row) and torsion (second row) in scenario 2 (Influence of sample size and noise): comparison between true parameter (black), estimates using $n = 100, \gamma = 0.001$ (red), $n = 100, \gamma = 0.005$ (green), $n = 200, \gamma = 0.001$ (purple) and $n = 200, \gamma = 0.005$ (orange). Initial guesses for $\boldsymbol{\theta}$ are plotted in red on Figure 4.1.

6.1.3 Scenario 3: Influence of Frenet Curvatures Initialization

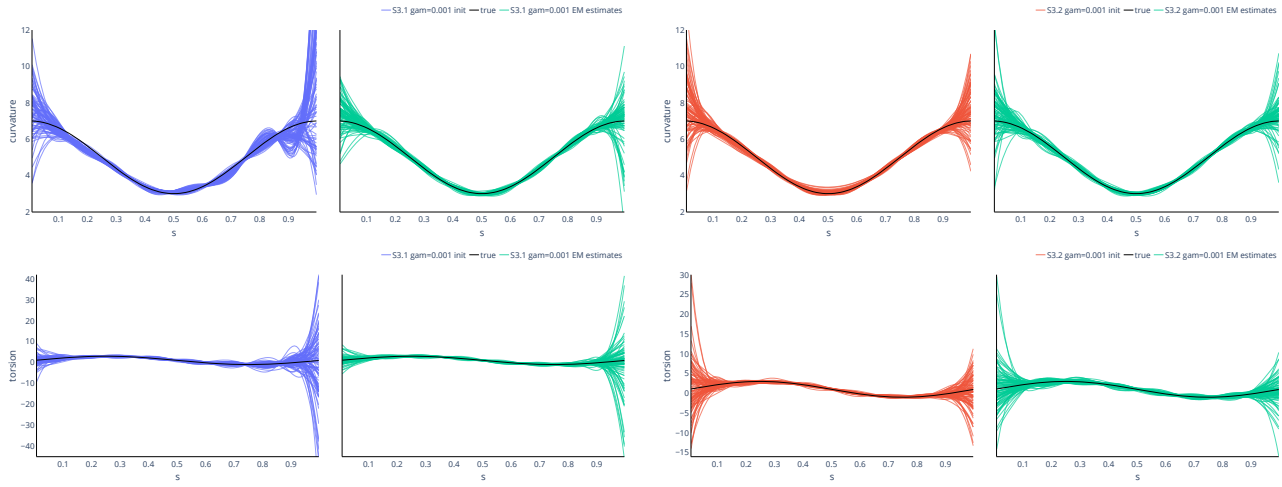
As in practice, we do not know the true parameter $\boldsymbol{\theta}$, we must initialize the Frenet curvatures parameter directly from the observations of the Euclidean curve $\{\boldsymbol{y}_i\}_{i=1,\dots,n}$. Hence, in this simulation scenario, we test the performance of the EM algorithm using various available methods for computing the first guess $\hat{\boldsymbol{\theta}}^{(0)}$ introduced in Chapter 3. Results of Table 3.2 show that the penalized least squares method gives nearly similar results regardless of the method used to compute the Frenet path. Therefore here, we compare only the method based on extrinsic formulas and the least squares method based on Gram-Schmidt orthonormalization for computing $\hat{\boldsymbol{\theta}}^{(0)}$ from $\{\boldsymbol{y}_i\}_{i=1,\dots,n}$:

- **S3.1** Derivatives estimation based on local polynomial regression and use of extrinsic formulas of Frenet curvatures to estimate $\hat{\boldsymbol{\theta}}^{(0)}$.
- **S3.2** Frenet path estimation using Gram-Schmidt orthonormalization of estimated derivatives and use of the penalized least squares criterion to estimate $\hat{\boldsymbol{\theta}}^{(0)}$.

We fix $\sigma^{(0)} = 0.1$, $\hat{\boldsymbol{P}}_0^{(0)} = (0.01)^2\mathbf{I}_6$, $n = 100$ and $\gamma = 0.001$. All results are displayed on Table 4.3. For the case $\gamma = 0.001$, they are also plotted on Figure 4.3. We observe that the EM algorithm improves errors in the estimates of Frenet curvatures and the Frenet path in all cases in comparison with the methods studied in Chapter 3. However, this improvement is quite significant when starting with estimates from extrinsic formulas but much less so when starting with estimates from the least squares criterion minimization given Gram-Schmidt Frenet path estimates. This shows that the criterion proposed in Chapter 3 as an improvement to the method based on extrinsic formulas can be sufficiently good in certain cases to be used directly as an estimation method for $\boldsymbol{\theta}$. Furthermore, we note that the error obtained with EM often depends on the initialization method in this case. However, once again, we have chosen a maximum of 200 iterations for EM for computational time reasons. However, EM requires more iterations if the initial guess parameter is less accurate. We can, therefore, imagine obtaining errors in case **S3.1** similar to those in case **S3.2** by increasing the number of possible iterations for the EM.

	γ	$\mathbf{d}(Z, \hat{Z})$	$\mathbf{re}(\kappa, \hat{\kappa}^{(0)})$	$\mathbf{re}(\tau, \hat{\tau}^{(0)})$	$\mathbf{d}(Z, \hat{Z}^{\hat{\theta}^{(0)}})$	$\mathbf{re}(\kappa, \hat{\kappa})$	$\mathbf{re}(\tau, \hat{\tau})$	$\mathbf{d}(Z, \hat{Z}^{(\hat{\theta})})$
S3.1	0.001	0.049 (0.019)	0.129 (0.087)	1.601 (1.243)	0.257 (0.229)	0.049 (0.034)	1.127 (1.195)	0.058 (0.021)
	0.005	0.117 (0.048)	0.308 (0.224)	1.936 (1.701)	0.620 (0.728)	0.219 (0.213)	1.671 (1.588)	0.127 (0.045)
S3.2	0.001	0.033 (0.008)	0.061 (0.031)	0.704 (0.449)	0.063 (0.014)	0.046 (0.020)	0.608 (0.456)	0.050 (0.014)
	0.005	0.075 (0.029)	0.139 (0.079)	1.131 (1.173)	0.153 (0.080)	0.131 (0.090)	1.125 (1.123)	0.107 (0.048)

Table 4.3: Estimation errors of scenario 3: influence of Frenet curvatures initialization.



(a) **S3.1** (b) **S3.2**

Figure 4.3: Estimated curvature (first row) and torsion (second row) in scenario 3 in the case $\gamma = 0.001$. **S3.1**, comparison between true parameter (black), initial guesses estimated using extrinsic formulas (blue), and estimates by the EM (green). **S3.2**: comparison between true parameter (black), initial guesses estimated using penalized least squares (red), and estimates by the EM (green).

6.2 Simulation Model with Random State Variable

With the ultimate aim of estimating the Frenet curvatures of motion trajectories, we have introduced the propagation model (4.22) as a relaxation of the true model (4.20) that accounts for misspecification. However, the proposed EM algorithm for inferring the parameters of the state space model (4.24) can also be used in theory for state variables generated by a random model like the SDE (4.22). In order to test the ability of the proposed algorithm to correctly estimate the parameters in this case, we consider the following data generative model:

$$\boldsymbol{y}_i = X(s_i) + \boldsymbol{\epsilon}_i, \quad i = 1, \dots, n\,,$$

where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \gamma^2 \boldsymbol{I}_3)$, and the shape function $s \mapsto X(s)$ is part of the random state variable $Z(s)$ solution of the stochastic differential equation

$$dZ(s) = Z(s)\left(\boldsymbol{\omega}_{\boldsymbol{\theta}}(s) + L\boldsymbol{\zeta}(s)\right)^{\wedge} ds\,,$$

where $s \mapsto \boldsymbol{\theta}(s)$ and $t \mapsto s(t)$ defined the reference shape (4.89) and $\boldsymbol{\zeta}(s) \sim \mathcal{GP}(0, k(s, s'))$ where $k(s, s')$ is a Matern kernel with a length scale $l = 0.5$ and a parameter $\nu = 1.5, 2.5$. We fix the sample size $n = 100$ and noise level $\gamma = 0.001$. The number of basis functions for $\hat{\boldsymbol{\theta}}$ is 15. We denote μ the solution of the ODE (4.26) with the parameter $\boldsymbol{\theta}$ and $Z^{(init)}$ the solution of the SDE. The results displayed in Table 4.4 clearly show that the proposed model does not work in this case. We observe an identification

issue between the introduced smooth noise $\boldsymbol{\zeta}(s)$ and the mean parameter $\boldsymbol{\theta}(s)$ in this context. As we have only one noisy observation of the parameter, the model is unable to differentiate between the mean parameter and the smooth noise. The estimated mean parameter corresponds to the sum of the true mean parameter and the smooth noise defined by a Gaussian process. Furthermore, the propagation model, as defined in this chapter (4.24), is supposed to be perturbed by white noise rather than a smooth perturbation defined by a Gaussian process. Consequently, the filtering and smoothing algorithms are not suitable for this scenario.

ν	$\mathbf{d}(Z^{(\text{init})}, \hat{Z})$	$\mathbf{d}(\mu, \hat{Z})$	$\mathbf{re}(\kappa,\hat{\kappa})$	$\mathbf{re}(\tau,\hat{\tau})$	$\mathbf{d}(Z^{(\text{init})}, \hat{\mu}^{(\hat{\theta})})$	$\mathbf{d}(\mu, \hat{\mu}^{(\hat{\theta})})$	$\mathbf{d}(\hat{Z}, \hat{\mu}^{(\hat{\theta})})$
1.5	1.482 (1.339)	2.859 (0.610)	1.244 (0.753)	3.568 (1.705)	1.573 (1.344)	2.755 (0.675)	0.311 (0.499)
2.5	1.389 (1.324)	2.797 (0.564)	1.230 (0.718)	3.761 (1.767)	1.469 (1.336)	2.755 (0.553)	0.251 (0.370)

Table 4.4: Estimation errors of scenario: simulation model with random state variable.

7 Application to Sign Language Motion Trajectories

Frenet curvatures appear in the literature as particularly interesting parameters for studying and analyzing human motion (Lacquaniti et al., 1983; Maoz et al., 2009; Pollick et al., 2009; Flash and Berthoz, 2021). Different laws emphasize specific relationships among these parameters, depending on the type of human movement considered. Consequently, when testing the applicability of these laws for a specific type of motion, it becomes crucial to have precise, realistic, robust, and minimally biased estimators for these Frenet curvatures. In this context, we evaluate the performance of the different methods proposed in this chapter and the previous one for estimating Frenet curvatures of real data corresponding to wrist motion trajectories in sign language. Specifically, we focus on 6 French sign language gestures being "mono-manual", i.e. executed with a single hand (in this case, the right hand), which correspond to words "dimanche", "toujours", "autrefois", "avril", "train", and "avoir l'air". The signer repeated each of these 6 signs approximately 5 times, leading to a set of 32 trajectories (part of the second datasets detailed in Section 3.1). Motion data was captured using Mocaplab's advanced motion capture system at intervals of 0.01 seconds. Manual segmentation was applied to identify individual signs and repetitions. Note that we consider only isolated signs, where the signer returns to a "zero" position between each repetition, which significantly simplifies the segmentation process. Compared to experiments conducted with synthetic data, the wrist motion trajectories of sign movement are generally much more complex. Consequently, variations in curvature and torsion are more pronounced. Moreover, despite the high efficiency of Mocaplab's motion capture system and techniques, resulting in minimally noisy data, the arc length functions often deviate significantly from the identity function. Consequently, the corresponding discretization grids are irregular. These elements pose additional challenges in estimating Frenet curvatures within a functional data analysis framework. We compare the Frenet curvatures estimation results obtained with each of the following 4 estimation methods.

1. Use of extrinsic formulas based on derivatives estimated with a local polynomial regression (Chapter 3 Section 2).

2. Use of the penalized least squares criterion based on Frenet path "pseudo" observations (Chapter 3 Section 5, Algorithm 1).

3. Use of the iterative Algorithm 2 with tracking based smoother of Frenet path and the penalized least squares criterion (Chapter 3 Section 4.2.2 and Section 5)

4. Use of the EM algorithm developed in this chapter based on an initial guess of $\boldsymbol{\theta}$ obtained with the method 2.

When necessary, the "pseudo" observations of the Frenet path are estimated as a preprocessing step by the Gram-Schmidt orthonormalization of the curve's derivatives (Chapter 3 Section 4). In the synthetic data experiments, we select a number of B-spline basis functions and place the knots regularly on the grid $[0, 1]$. In this case, to deal with the irregular observations grid, one out of every 4 points in the irregular grid corresponding to the observations is selected as a knot. Consequently, using this method, we place more knots in areas with a higher density of points, which are generally regions displaying a more complex shape in the observations. For all methods, the hyperparameters are selected with 30 iterations of a Bayesian optimization based on the corresponding cross-validation criteria in the ranges $h \in [0.05, 0.15]$ and $\lambda_1, \lambda_2 \in [10^{-30}, 10^{-5}]$.

To evaluate the quality of the estimates of the Frenet path and Frenet curvatures by each of the considered methods, we compare the $\mathbb{L}^2$ and SRVF (2.7) distances between the true movement trajectory captured with the system and the trajectories reconstructed from the estimators. In Figure 4.4, the trajectories are reconstructed by solving the Frenet-Serret equation based on the obtained $\hat{\boldsymbol{\theta}}$. In Figure 4.5, the trajectories are reconstructed by integrating the first vector (tangent vector) of the estimated Frenet path (in the case of method 2 by Gram-Schmidt, in the case of method 3 by the tracking-based smoother, and in the case of method 4 by the IEKF). Additionally, for one trajectory of each of the 6 signs, we display the 4 different estimates of $\hat{\boldsymbol{\theta}}$ and the corresponding reconstructed 3D curves based on these estimators.

From Figure 4.5, we observe that the EM algorithm significantly reduces the error made in estimating the tangent vector. This suggests that the Frenet path used for estimating $\boldsymbol{\theta}$ is more reliable in this case. However, the error on the curves reconstructed from $\hat{\boldsymbol{\theta}}$ is not reduced as much by this method (on Figure 4.4, 1st quartile and median lower than other methods but 3rd quartile higher than that of methods 2 and 3). This can partly be explained by the phenomenon visible in Figure 4.8 for the sign "dimanche". Indeed, we observe that the initial trajectory contains a very small loop in the middle that the other methods have completely smoothed out. The method based on the EM algorithm seems to attempt to fit this small loop but does not fully succeed. At this point, even though the algorithm accurately estimates the Frenet path, the local estimation error of curvature and torsion, which are too smooth, affects the overall shape of the reconstructed curve and thus the $\mathbb{L}^2$ distance. This may be due to the number of knots placed at that location to estimate $\hat{\boldsymbol{\theta}}$. Indeed, because of this small loop, there is a sharp curvature peak, which may be too smoothed. Therefore, it can be assumed that adding more knots could reduce these different errors. Furthermore, we see in Figure 4.8 that the torsion estimates from the EM algorithm seem to have much more variability than the torsion estimates from the other methods. This is potentially due in part to a much better estimation of the other vectors of the Frenet frame (normal and binormal), the quality of which significantly influences that of the torsion. These vectors are generally not always well estimated by the Gram-Schmidt method. Finally, we note that on these data, there is very little difference between methods 2 and 3. Indeed, since the data are already very noiseless, the iterative algorithm with intermediate Frenet path smoothings is unnecessary. Furthermore, comparing the method based on extrinsic formulas with the others confirms the results obtained on simulated data in Chapter 3.

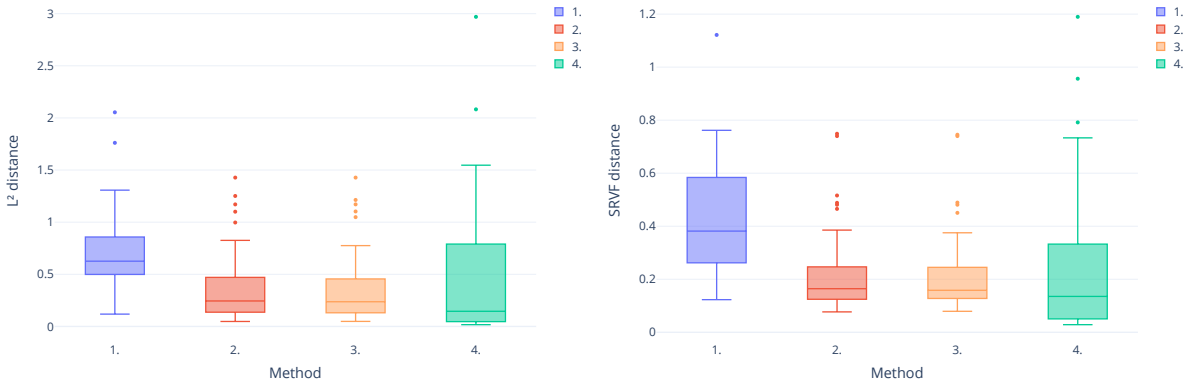


Figure 4.4: $\mathbb{L}^2$ (left boxplot) and SRVF (right boxplot) distances between the true recorded trajectory and the reconstructed one from Frenet curvatures estimates by each of the four methods.

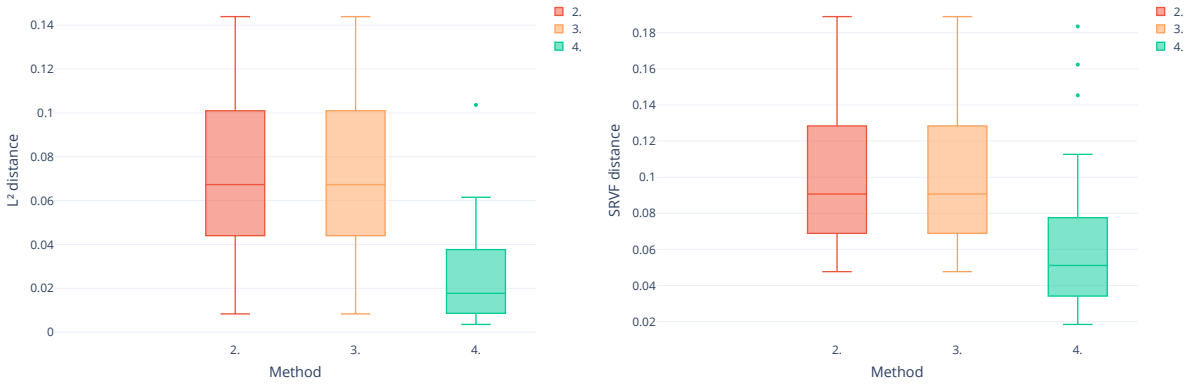
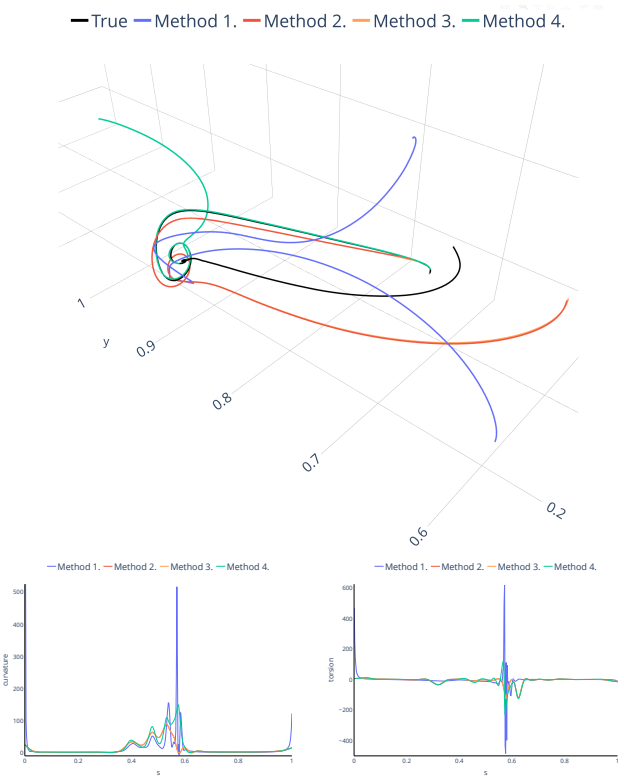
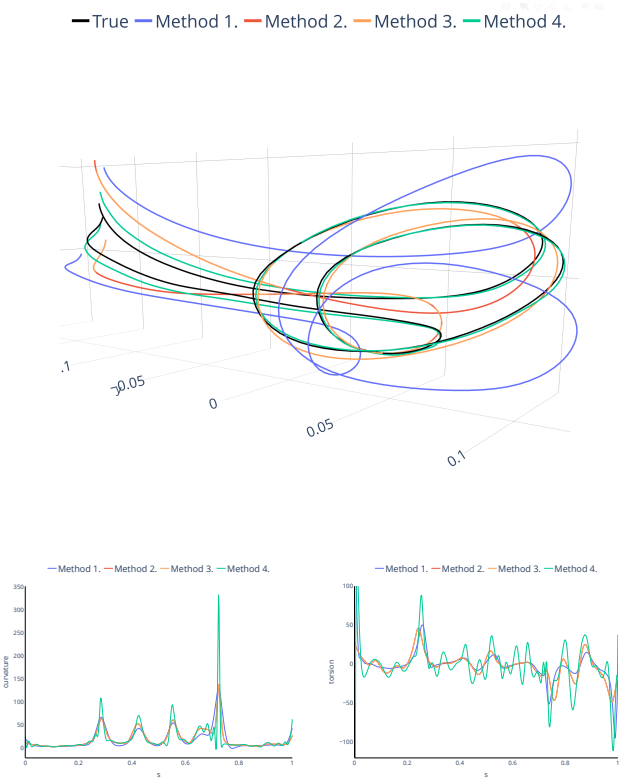


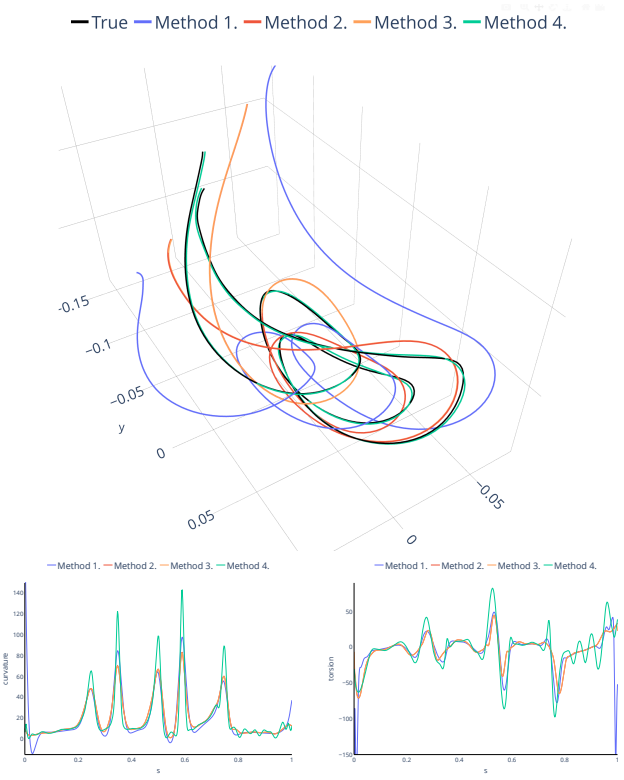
Figure 4.5: $\mathbb{L}^2$ (left boxplot) and SRVF (right boxplot) distances between the true recorded trajectory and the integrated tangent vector estimated by the methods.



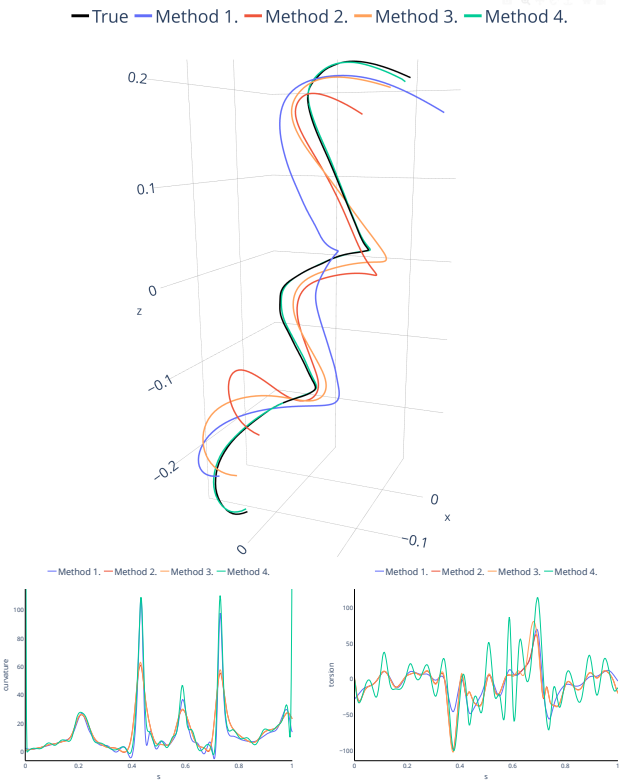
Wrist motion of sign "dimanche".



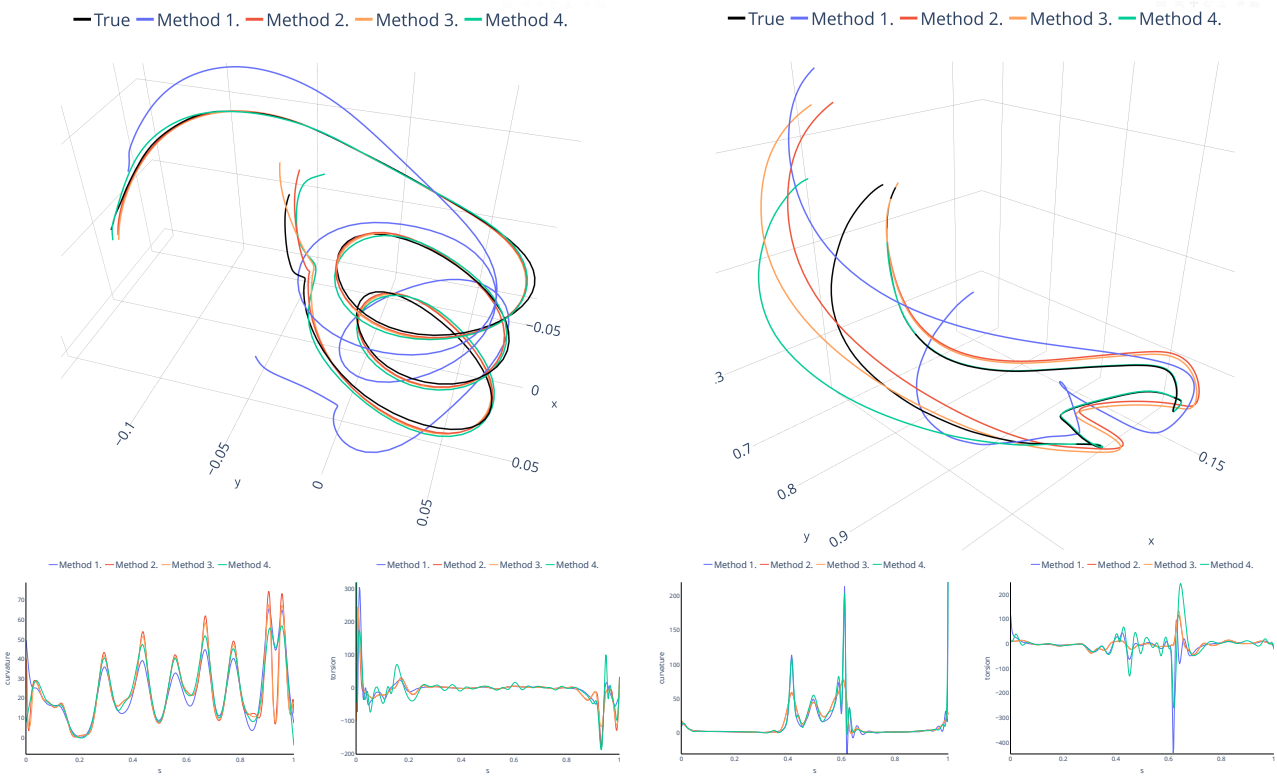
Wrist motion of sign "toujours".



Wrist motion of sign "autrefois".



Wrist motion of sign "avril".



Wrist motion of sign "train".

Wrist motion of sign "avoir l'air".

Figure 4.8: Estimation results for one trajectory of each of the 6 signs. Top: Reconstructed trajectories from the Frenet curvature estimates by each method (1: blue, 2: red, 3: orange, 4: green) and the true captured trajectory in black. Bottom: Each method estimated curvatures (left) and torsions (right).

8 Conclusion and Perspectives

We have developed a Frenet curvatures estimation method that eliminates the need for a preprocessing step and and allows for the estimation of the corresponding trajectory and Frenet path in a unified manner. This is achieved through an original Expectation-Maximization algorithm. Specifically, starting with the problem of estimating the parameters of the Frenet framework, we propose a novel formulation of the classic EM algorithm that adapts to the estimation of Lie group-valued latent variables and time-varying parameter estimation. The results of the proposed experiments demonstrate that this unified formulation generally outperforms other methods from Chapter 3 in terms of Frenet curvatures estimation. However, we have also shown that the method initially proposed by Park and Brunel (2019) proves sufficient to obtain good estimates in cases where the trajectory geometry is not too complicated, and there is not too much noise. Furthermore, the results highlight a clear improvement in the estimation of the Frenet path provided by this new method. It appears that the major remaining limitations lie more in the $\boldsymbol{\theta}$ estimation method. Due to the functional framework, the estimation of this parameter depends on many hyperparameters that require proper adjustment (number of basis functions, knot placement, smoothing parameters). Currently, a more extensive optimization of these hyperparameters is limited by the computational time required for the method, which is largely due to the search for the optimal smoothing parameter λ by cross-validation. Therefore, there is room for improvement by accelerating and optimizing the code.

Moreover, in the formulation of the perturbed model (4.22), the perturbation accounting for the possible uncertainty of the parameter $\boldsymbol{\theta}$ is modeled by a white noise Gaussian process, representing thus independent noise. This approach may not be well-suited for estimating a parameter known to be functional and exhibiting smoothing properties. Consequently, the works of Álvarez et al. (2009, 2011) suggest considering a non-parametric model for the perturbation, such as a Gaussian process modeling an unknown perturbation function. Typically, the covariance function is defined by a kernel function and encodes prior knowledge about the smoothness of the parameter. Hartikainen and Särkkä (2011) and Hartikainen et al. (2012) then demonstrate how these non-linear latent force models can be reduced to state-space models driven by white noise and then estimated in a similar manner to the approach proposed in this chapter. This method could be applied to our problem and would likely be more relevant to the physical model it represents. However, it would need to be extended to the case of non-Euclidean variables. Nevertheless, this extension would further complicate the derivation of formulas for optimizing the parameters. Thus, the complexity of the model and equations, limiting the extension of the method to any dimension d and requiring significant computation time, prompts consideration. This suggests that alternative approaches for estimating Frenet curvatures, such as deep learning-based methods, might be more efficient and effective for practical use. However, these approaches remain to be explored and have not yet been proposed in the literature to our knowledge.

Part II

Shape Analysis of Curves under Frenet-Serret Framework

5

Riemannian Geometry of Curves under Frenet Curvatures Representation

Abstract

Geometric frameworks for analyzing curves are common in applications as they focus on invariant features and provide visually satisfying solutions to standard problems such as computing invariant distances, averaging curves, or registering curves. We show that for any smooth curve in $\mathbb{R}^d, d > 1$, the generalized curvatures associated with the Frenet-Serret equation can be used to define a Riemannian geometry that takes into account all the geometric features of the shape. This geometry is based on a Square Root Curvature Transform that extends the square root-velocity transform for Euclidean curves (in any dimensions) and provides likely geodesics that avoid artefacts encountered by representations using only first-order geometric information. Our analysis is supported by simulated data and is especially relevant for analyzing human motions. We consider trajectories acquired from sign language, and show the interest of considering curvature and also torsion in their analysis, both being physically meaningful.

Contents

1 Introduction

Identifying and comparing different types of visual objects is a fundamental task in machine learning and computer vision problems (Lähner et al., 2016; Eisenberger and Cremers, 2020; Koestler et al., 2022). The shape is one of the essential features of objects that allow us to understand and characterize them. Nowadays, it is much easier to obtain data in the form of shapes, typically as dense point clouds or landmarks. The main task in shape analysis is to define a proper framework to compare and quantify the variation of the shapes. However, the shape space is generally nonlinear, and extracting meaningful information or features is complex. One of the successful approaches to shape analysis utilizes a Riemannian framework of differential geometry, where a metric can be defined between the shapes, which is invariant with respect to shape-preserving transformations such as translation and rotation. For instance, this gives rise to geodesic distances that are naturally invariant to smooth and optimal deformations through geodesic paths between the shapes (Lang, 2006). This approach is very versatile as it can be adapted to various kinds of manifold-value data and can be designed to emphasize important geometric information to be preserved. As a consequence, several choices of metrics are possible, such as the class of invariant Sobolev metrics, often called *elastic*, for the analysis of curves (Bauer et al., 2021). In this work, we are concerned with curves that often arise in the application as trajectories (function of time) or motions (animation, activity recognition) and with the definition of a framework to compare their shapes. The differential geometry of Euclidean curves is among the simplest (with respect to higher dimensional manifolds), and relatively simple Riemannian metrics are available with different mathematical representations of curves (Younes, 1998). Quite remarkably, the introduction of the Square Root Velocity (SRV) transform (Srivastava et al., 2011) that consists of a particular representation of the shape of a curve enables us to define a so-called elastic Riemannian distance, which has proven to be useful for the statistical shape analysis of 2D and 3D curves in applications. The SRV possesses interesting properties such as a principled theoretical framework, efficient computation, and generalization to higher dimensions (Bauer et al., 2022). Nevertheless, a limitation of the SRV transform and the corresponding elastic distance is the restrictive use of the first-order derivative, while the geometry of 3-D (or d-D) curves depends on the derivatives until order d. Indeed, it is well-known that a 3D curve is characterized by its curvature and its torsion: this is particularly critical when we consider trajectories or human movements, where the curvature and torsion can have a physical meaning.

2 Related works and contributions

As we will recall in Section 3, the full geometry of a curve can be given either by the Frenet curvatures (standard curvature and torsion in 3D) or by the path of Frenet frames. There have been few attempts to directly deal with the Frenet curvatures: most of the works have been produced in 2D curves as an alternative representation, Srivastava and Klassen (2016). Nevertheless, the potential for applications has not been investigated. In Needham (2019), the elastic shape analysis framework has been considered for 3D curves based on the Frenet frames, but the link to the physical parameters has been overlooked. We can also mention the shape analysis of curves on Lie groups with application in computer animation, Celledoni et al. (2016). Outside the Riemannian framework, an attempt has been made to use a direct curvature-based interpolation of curves (Saba, 2012; Surazhsky and Elber, 2002).

In this work, we introduce two representations of Euclidean curves for their shape analysis that use their complete geometry through Frenet curvatures. We provide the full development of the Riemannian frameworks associated with these two representations. As a consequence and in comparison with existing

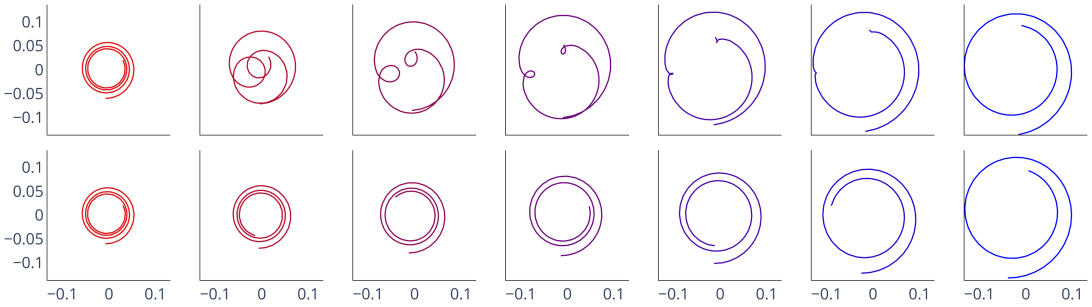


Figure 5.1: Geodesic paths between two 2D scaled spirals with different number of spins: SRVF (1^{st} row) and SRC (2^{nd} row).

methods, our approaches also give explicit formulas for geodesics and geodesic distances. The first representation considered is based directly on unparametrized Frenet curvatures. We show through experiments that it defines a shape analysis framework that lacks elasticity. As the main contribution, we propose the definition of a second representation, called the Square-Root Curvature (SRC) Transform, which takes into account reparameterization and defines a metric on the space of shapes through the quotient space with the group of diffeomorphisms. One can imagine that the classical method associated with the SRVF, defining a Sobolev elastic metric (Bauer et al., 2022), already implicitly uses all the geometric information necessary for a relevant curve analysis. We show here with simple examples that this is not the case. Through experiments on synthetic data, we compare the methods, and illustrate the limitations of the SRVF one, due to its lack of use of geometric information. To be able to judge and compare the quality of these metrics, we compare consistent sets of curves characterized by specific features. The SRC method shows a special strength in defining a framework that remains consistent with these sets. The straightforward example of geodesics between helices with different numbers of spins (Figure 5.1 in 2D and Figure 5.2 in 3D) shows that, in contrast, this is not the case for the SRVF method. In addition, we highlight the interest of these Frenet curvatures-based representations in the real application case of human motion trajectory analysis.

3 Riemannian Geometry on Shape Space

We introduce useful notations and we review the main approach for constructing tractable representations of the shape of a curve and deriving a Riemannian geometry.

3.1 Shape Analysis of Euclidean Curves

We consider absolutely continuous curves that are smooth, open, and with values in some Euclidean space $\mathbb{R}^d$, we denote this set as $AC\left([0,1],\mathbb{R}^d\right)$. These curves are typically parametrized by a variable t that can usually be interpreted as time. Nevertheless, from a (statistical) shape analysis point of view, we focus on the *geometric shape* of curves that do not depend on a specific parametrization or standard transformations such as translations, rotations, scaling, or reparametrizations. To distinguish between parametrized curves that differ only by translation, we consider the set of absolutely continuous curves where $x(0) = 0$, denoted by $AC_0\left([0,1],\mathbb{R}^d\right)$. The natural and intrinsic parametrization that uniquely defines the *shape* of a curve x is the arc-length parametrization, defined with the arc length function $s(t) = \int_0^t \|\dot{x}(u)\|du$, for $t \in [0,1]$. In order to remove the scaling variability, the total length of the curve $s(1)$ is set to 1. Under this parametrization, the *shape* $X : [0,1] \mapsto \mathbb{R}^d$ of the curve is the image of the function x such that $x(t) = X(s(t))$. As we want to study shapes independently of

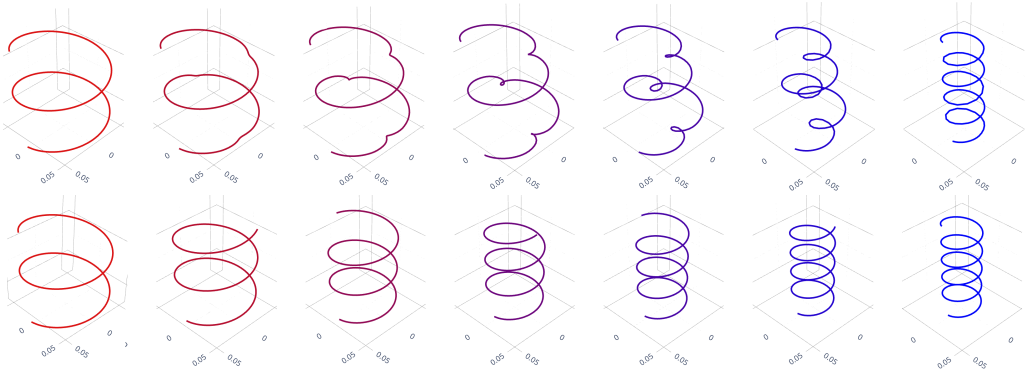


Figure 5.2: Geodesic paths between two scaled 3D circular helices with different number of spins: SRVF (1^{st} row) and SRC (2^{nd} row).

their parameterizations, we introduce the reparametrization group $\mathrm{Diff}_+([0,1])$, of smooth orientation preserving diffeomorphisms of the interval $[0,1]$ onto itself. This group acts on the space of absolutely continuous curves by right composition, and this action only alters the parametrization of the curve, not the inherent shape X. Therefore, X represents the equivalent class $[x]$ that contains all the possible time reparametrization of x. The space of such shapes (or unparametrized curves) is often mathematically defined as the quotient space

$$\mathcal{S}([0,1],\mathbb{R}^d) = AC_0\left([0,1],\mathbb{R}^d\right)/\mathrm{Diff}_+([0,1]). \tag{5.1}$$

The purpose of shape analysis of curves is to define a distance function $d_{\mathcal{S}}$ on $\mathcal{S}$ and a framework to perform a complete statistical analysis on a set of curves in $\mathcal{S}$ (e.g. mean, classification, or Principal Component Analysis etc.). One of the main challenges in defining this distance is to choose an appropriate mathematical representation of the curves that can be made invariant to all shape-preserving transformations - translation, rotation, scaling, and reparametrization. Moreover, one of the stakes of such representation is to offer an (infinite-dimensional) Riemannian manifold structure that brings powerful and flexible tools for studying the geometry of shapes or statistical properties notably thanks to the tangent space of the manifold (Lang, 2006; Sommer et al., 2020). In Srivastava and Klassen (2016), a list of the few possible representations is given - coordinate functions, curvatures, angle function, and square-root velocity function (SRVF) - and a framework for curve analysis is derived for the last two ones. While the angle representation is unparameterized, the SRVF representation depends on the parametrization, which is shown to be very useful as a tool for the registration of points across curves. As a consequence, the parameterization group $\mathrm{Diff}_+([0,1])$ must be eliminated by using a quotient space. The classical approach is to define the Riemannian metric on the shape space through a metric on the space of parametrized representations that is invariant to reparametrization: $\forall h \in \mathrm{Diff}_+([0,1])$

$$d_{AC_0}(x_0,x_1) = d_{AC_0}(x_0\circ h, x_1\circ h). \tag{5.2}$$

In that case, the distance on $\mathcal{S}$ is defined as the infimum over all possible reparametrization. For $X_0, X_1 \in \mathcal{S}$,

$$d_{\mathcal{S}}(X_0,X_1) := \inf_{h\in\mathrm{Diff}([0,1])} d_{AC_0}(x_0, x_1\circ h). \tag{5.3}$$

In the following, we will denote with a dot the derivation with respect to the time variable, and with a prime the one with respect to the arc-length parameter.

3.2 Square Root Velocity Framework

The square-root velocity function framework is the most commonly used representation for curve shape analysis in $\mathbb{R}^d$, (Srivastava et al., 2011; Srivastava and Klassen, 2016). The square-root velocity function (SRVF) of $x \in AC_0\left([0,1],\mathbb{R}^d\right)$, denoted by $\mathcal{R}_{\text{SRVF}}(x) = q$, is defined as

$$q(t) = \frac{\dot{x}(t)}{\sqrt{\|\dot{x}(t)\|}}\,. \tag{5.4}$$

Noting that the unit tangent vector of the curve x is equal to $T(s(t)) = \frac{\dot{x}(t)}{\|\dot{x}(t)\|} = X'(s(t))$, the SRVF can also be expressed in terms of this tangent vector and the curvilinear speed as

$$q(t) = \sqrt{\dot{s}(t)}T(s(t))\,. \tag{5.5}$$

The SRV transform is a bijection with $AC_0\left([0,1],\mathbb{R}^d\right)$ and its explicit inverse is $x(t) = \int_0^t q(u)|q(u)|du$. As we consider length-normalized curves, the SRVFs have a unit $\mathbb{L}^2$ norm, and their set is the convenient unit Hilbert sphere, a Riemannian submanifold of $\mathbb{L}^2([0,1],\mathbb{R}^d)$ (with the $\mathbb{L}^2$ inner product). Then, the $\mathbb{L}^2$ metric on SRVF induces a Riemannian metric on $AC_0\left([0,1],\mathbb{R}^d\right)$ where geodesics are given by the shorter arcs on great circles between SRV functions. The action of $\text{Diff}_+([0,1])$ on $AC_0\left([0,1],\mathbb{R}^d\right)$ is reflected on q by the group action denoted by $*$ and defined as

$$(q * h)(t) = \sqrt{\dot{h}(t)}q(h(t)) \tag{5.6}$$

and if the curve is rotated by a matrix $O \in SO(d)$, its SRVF gets rotated by the same matrix. The key property of this representation is the invariance of its associated distance under the action of $\text{Diff}_+([0,1])$ and $SO(d)$:

$$\|O(q_0 * h) - O(q_1 * h)\|_{\mathbb{L}^2} = \|q_0 - q_1\|_{\mathbb{L}^2}. \tag{5.7}$$

The metric can be used to define a proper distance on the shape space $\mathcal{S}([0,1],\mathbb{R}^d)$

$$d_{\mathcal{S}}^{(\text{SRVF})}(X_0, X_1) := \inf_{\substack{O\in SO(d)\\ h\in\text{Diff}([0,1])}} \cos^{-1}\langle q_0, O(q_1 * h)\rangle\,. \tag{5.8}$$

Then, given the optimal warping function h^*, the geodesic path on the shape space is taken between q_0 and $q_1 * h^*$ and given by

$$\alpha_{\mathcal{S}}^{(q)}(\tau) = \frac{1}{\sin(\vartheta)}\left(\sin((1-\tau)\vartheta)q_0 + \sin(\tau\vartheta)O(q_1 * h^*)\right)\,, \tag{5.9}$$

where $\vartheta = \cos^{-1}\langle q_0, O(q_1 * h^*)\rangle$.

The definition of this distance on the shape space under the SRVF representation can be interpreted as the following registration problem

$$h^*, O^* = \underset{\substack{O\in SO(d)\\ h\in\text{Diff}([0,1])}}{\arg\min} \int_0^1 \left\| q_0(t) - Oq_1(h(t))\sqrt{\dot{h}(t)}\right\|_2^2 dt\,. \tag{5.10}$$

In Brunel and Park (2019), this registration problem has been reformulated with the unit tangent vector and the arc length functions. By defining $\gamma = s_1 \circ h \circ s_0^{-1} \in \text{Diff}_+([0,1])$ the optimization problem (5.10) amounts to finding the optimal diffeomorphism of $\text{Diff}_+([0,1])$ that acts on the arc-length parameter s and solves the minimization problem:

$$\gamma^*, O^* = \underset{\substack{O\in SO(d)\\ \gamma\in\text{Diff}([0,1])}}{\arg\min} \int_0^1 \|T_0(s) - OT_1(\gamma(s))\sqrt{\gamma'(s)}\|_2^2 ds\,. \tag{5.11}$$

It should be noted, in this reformulation, that the object $OT_1(\gamma(s))\sqrt{\gamma'(s)}$ does not represent the same shape as $X_1(s)$ in the shape space. Here the element γ of $\mathrm{Diff}_+([0,1])$ is not used as a reparametrization of the curve but to deform the element of $\mathcal{S}([0,1],\mathbb{R}^d)$. Under this point of view, the set of unit tangent vectors that can be reached by deforming the vector $T(s)$, with the group action $T*\gamma$ defines an equivalence class of shapes associated with that one, as in the setting of deformable templates of Grenander's theory (Younes, 2010, 2018).

Finally, the choice of a parametrized curve representation for shape analysis, discussed in Srivastava and Klassen (2016), can be seen as the problem of choosing a good geometric representative of the shape as a template and defining an associated registration problem. Hence, an appropriate choice may be seen as a matter of modeling and should be done in interaction with the type of data analyzed and the dimension of the space. In the next sections, we use h to refer to functions of $\mathrm{Diff}_+([0,1])$ that act on the time variable t and γ for ones that act on the arc-length variable s.

4 Exhaustive Geometric Information with Frenet Representation

Based on these previous observations and with the intention of developing a more suitable framework for the analysis of three-dimensional curves, Brunel and Park (2019) proposed a direct extension of the SRVF method, which considers not only the tangent vector as the geometric representation of the curve but the whole Frenet-Serret frame in three dimensions. Their idea is to use an exhaustive description of the geometry of curves by incorporating higher-order information about the geometry in the representation. To exploit this idea, we propose to study suitable representations based on this Frenet-Serret framework.

4.1 Direct Extension of the SRVF with the Frenet Frame

From the detailed Frenet-Serret framework in Section 3, one can think of the direct extension of the square-root velocity function (5.5) that simply consists in replacing the tangent vector with the entire Frenet frame. The representation of a parametrized curve $x \in F([0,1],\mathbb{R}^d)$ will be then

$$\mathcal{R}_Q(x)(t) = \sqrt{\dot{s}(t)}Q(s(t)). \tag{5.12}$$

This representation is used in Brunel and Park (2019) to define a new alignment method on $\mathcal{S}$. They extend the SRVF registration problem (5.11) by using the Frobenius distance between the Frenet frames instead of only the $\mathbb{L}^2$ distance of the unit tangent vectors. They show to have obtained more precise results with their method than the SRVF one. From the previous theorems, it is clear that this representation uniquely defines a parametrized curve $x \in F([0,1],\mathbb{R}^d)$. Indeed, for all $t \in [0,1]$, $\mathcal{R}_Q(x)(t) \in \mathcal{GL}_d(\mathbb{R})$ and if we consider the Frobenius norm on $\mathcal{GL}_d(\mathbb{R})$, we have $\frac{1}{d}\|\mathcal{R}_Q(x)(t)\| = \sqrt{\dot{s}(t)}$, as $Q(s(t)) \in SO(d)$. Then the Euclidean curve x can be reconstructed from $\mathcal{R}_Q(x)$ by integration along the time variable of the first vector of $\frac{1}{d}\|\mathcal{R}_Q(x)(t)\|\mathcal{R}_Q(x)(t) = \dot{s}(t)Q(s(t))$.

The set of such mathematical representations $\mathcal{R}_Q$ is a subset of the set of measurable curves from $[0,1]$ to $\mathcal{GL}_d(\mathbb{R})$ such that the curve $t \mapsto \frac{\mathcal{R}_Q(x)(t)}{\frac{1}{d}\|\mathcal{R}_Q(x)(t)\|} = Q(s(t))$ is a Frenet path and therefore an element of $\mathcal{F}_0$. One could attempt to show that the set of representations $\mathcal{R}_Q$ is a manifold if the set of Frenet path $\mathcal{F}_0$ is one. However, we can show from the Frobenius theorem that $\mathcal{F}_0$ is not a manifold. The set $\mathcal{F}_0$ corresponds to the set of solutions of the first-order ordinary differential equations on $SO(d)$ defined for $s \in [0,1]$, with fixed initial conditions $Q(0) = I_d$,

$$Q'(s) = V_{\boldsymbol{\theta}}(s, Q(s))\,,$$

where $V_{\boldsymbol{\theta}} : [0,1]\times SO(d) \to TSO(d)$ is a time-dependent vector field, defined by $V_{\boldsymbol{\theta}}(s,Q(s)) = Q(s)A_{\boldsymbol{\theta}}(s) \in T_{Q(s)}(SO(d))$ with $\boldsymbol{\theta} \in \mathcal{H}$. We denote by $\mathcal{D}$ the subbundle of the tangent bundle $TSO(d)$, defined by

$\mathcal{D} = \text{span}\{V_{\boldsymbol{\theta}} : [0,1] \times SO(d) \,|\, \boldsymbol{\theta} \in \mathcal{H}\}$. Then, the Frobenius theorem (Warner, 1983; Lyons, 2016; Lang, 2006) states that the subbundle $\mathcal{D}$ is integrable if and only if it is involutive. By definition $\mathcal{D}$ is involutive if, for any two vector fields $V^1_{\boldsymbol{\theta}}$ and $V^2_{\boldsymbol{\theta}}$ in $\mathcal{D}$, the Lie bracket $[V^1_{\boldsymbol{\theta}}, V^2_{\boldsymbol{\theta}}]$ is also in $\mathcal{D}$. However, for any $Q \in \mathbb{L}^2([0,1], SO(d))$, we have

$$[V^1_{\boldsymbol{\theta}}, V^2_{\boldsymbol{\theta}}](Q(s)) = V^1_{\boldsymbol{\theta}}(V^2_{\boldsymbol{\theta}}(Q(s))) - V^2_{\boldsymbol{\theta}}(V^1_{\boldsymbol{\theta}}(Q(s))) = Q(s)\left(A_{\boldsymbol{\theta}^2}(s)A_{\boldsymbol{\theta}^1}(s) - A_{\boldsymbol{\theta}^1}(s)A_{\boldsymbol{\theta}^2}(s)\right) ,$$

and we can easily show that the matrix $B = A_{\boldsymbol{\theta}^2}(s)A_{\boldsymbol{\theta}^1}(s) - A_{\boldsymbol{\theta}^1}(s)A_{\boldsymbol{\theta}^2}(s)$ no longer has the form a matrix $A_{\boldsymbol{\theta}}$ with $\boldsymbol{\theta} \in \mathcal{H}$. Indeed if we look at the coefficient $B_{i,i+2}$, for $1 < i < d-1$, we have

$$B_{i,i+2} = \sum_{j=1}^{d} a^1_{i,j} a^2_{j,i+2} = \theta^1_i \theta^2_{i+1} ,$$

which is not equal to zero a priori. Then $[V^1_{\boldsymbol{\theta}}, V^2_{\boldsymbol{\theta}}]$ does not take values in $\mathcal{D}$, and $\mathcal{D}$ is not involutive by definition. Therefore, from the Frobenius theorem, $\mathcal{D}$ is not integrable. This means that $\mathcal{F}_0$, the set of integral curves defined from $\mathcal{D}$, is not a manifold. Given this last result and since we are looking, in this work, for a mathematical representation of a Euclidean curve that allows us to define a complete Riemannian framework for shape analysis (geodesic, distance, mean), we do not consider further the representation $\mathcal{R}_Q$.

4.2 Unparametrized Frenet curvatures

A possible representation of a parametrized curve, already suggested in Srivastava and Klassen (2016); Surazhsky and Elber (2002); Saba (2012), which keeps the idea of encoding more geometric information, is the unparametrized Frenet curvatures and the arc-length function pair

$$\mathcal{R}_{\boldsymbol{\theta}}(x)(t) = \left(\sqrt{\dot{s}(t)},\, \boldsymbol{\theta}(s(t))\right) . \tag{5.13}$$

We denote by $\Psi([0,1])$ the set of square root velocity functions of length-normalized arc-length functions. This set is well-studied in the literature (Marron et al., 2015; Tucker et al., 2013). It is the unit sphere of the Hilbert space $\mathbb{L}^2([0,1], \mathbb{R})$ and therefore a Riemannian manifold equipped with the $\mathbb{L}^2$ metric. Then, the geodesic distance between two elements in $\Psi([0,1])$ is

$$d_\Psi\left(\sqrt{\dot{s}_0}, \sqrt{\dot{s}_1}\right) = \cos^{-1}\left(\left\langle \sqrt{\dot{s}_0}, \sqrt{\dot{s}_1} \right\rangle\right) \tag{5.14}$$

and the geodesic path connecting them is given by

$$\alpha_\Psi(\tau) = \frac{\sin((1-\tau)\vartheta)}{\sin(\vartheta)}\sqrt{\dot{s}_0} + \frac{\sin(\tau\vartheta)}{\sin(\vartheta)}\sqrt{\dot{s}_1} , \tag{5.15}$$

where $\vartheta = d_\Psi(\sqrt{\dot{s}_0}, \sqrt{\dot{s}_1})$. Moreover any element of $\text{Diff}_+([0,1])$ is uniquely represented by an element of $\Psi([0,1])$.

Proposition 5.1. *The set of Frenet curvatures $\mathcal{H}$ is a Riemannian submanifold of $\mathbb{L}^2([0,1], \mathbb{R}^{d-1})$.*

Proof. The set $M = \{x \in \mathbb{R}^{d-1} | x_1, \ldots, x_{d-2} > 0\}$ is an open subset of the Riemannian manifold $\mathbb{R}^{d-1}$. Then it is itself a differentiable Riemannian manifold with the standard inner product of $\mathbb{R}^{d-1}$, and for any point $p \in M$ the tangent space $T_p(M)$ is $\mathbb{R}^{d-1}$. The set of Frenet curvatures $\mathcal{H}$ is the set of measurable curves from $[0,1]$ to the Riemannian manifold M and thus also a manifold (Tixiang (1989)). Its tangent space is $\mathbb{L}^2([0,1], \mathbb{R}^{d-1})$ and it can be equipped with the $\mathbb{L}^2$ Riemannian metric. ■

Consequently, the geodesic distance on $\mathcal{H}$ is simply the $\mathbb{L}^2$ norm

$$d_{\mathcal{H}}(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1) = \|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_1\|_{\mathbb{L}^2} \tag{5.16}$$

and the geodesic path is the straight line connecting them

$$\alpha_{\mathcal{H}}(\tau) = (1-\tau)\boldsymbol{\theta}_0 + \tau\boldsymbol{\theta}_1. \tag{5.17}$$

Proposition 5.2. *The map $\mathcal{R}_{\boldsymbol{\theta}} : F([0,1], \mathbb{R}^d) \to \Psi([0,1]) \times \mathcal{H}$, defined above, is a bijection.*

Proof. The element of $\Psi([0,1])$ uniquely defines the arc-length function by $s(t) = \int_0^t (\sqrt{\dot{s}(u)})^2 du$. As mentioned before, we have a bijection between the unparametrized Frenet curvatures in $\mathcal{H}$ and the unparametrized curve in the shape space $\mathcal{S}$ (Theorem 2.3, Theorem 2.4). Then, from $X \in \mathcal{S}$, the initial parametrized curve is simply $x(t) = X(s(t))$. ∎

The set of such $\mathcal{R}_{\boldsymbol{\theta}}$ is the Cartesian product of $\Psi([0,1])$ and $\mathcal{H}$ and, therefore, is also a Riemannian manifold equipped with the product metric $d_\Psi \oplus d_{\mathcal{H}}$ (Mehmet, 2005). The induced metric on $F([0,1], \mathbb{R}^d)$ under the representation $\mathcal{R}_{\boldsymbol{\theta}}$ is

$$d_{\boldsymbol{\theta}}(x_0, x_1) = d_\Psi\left(\sqrt{\dot{s}_0}, \sqrt{\dot{s}_1}\right) + d_{\mathcal{H}}(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)\,. \tag{5.18}$$

In order to define a distance on the shape space $\mathcal{S}$ from that one, we must quotient out the space $\mathrm{Diff}_+([0,1])$. The action of $\mathrm{Diff}_+([0,1])$ on $\Psi([0,1])$ is the same as (5.6), and the Frenet curvatures are invariant under reparametrization of the corresponding parametrized curve (Lemma 2.1). Moreover, by simply taking $h^* = s_1^{-1} \circ s_0 \in \mathrm{Diff}_+([0,1])$, we have $\sqrt{\dot{s}_1} * h^* = \sqrt{\dot{s}_0}$, and thus the distance on $\Psi([0,1])/\mathrm{Diff}_+([0,1])$ between s_0 and s_1 is zero. Hence, the induced distance on the shape space, between $X_0, X_1 \in \mathcal{S}$, under the representation $\mathcal{R}_{\boldsymbol{\theta}}$ is defined as

$$d_{\mathcal{S}}^{(\boldsymbol{\theta})}(X_0, X_1) := d_{\mathcal{H}}(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1) = \|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_1\|_{\mathbb{L}^2} \tag{5.19}$$

and the geodesic path connecting them is

$$\alpha_{\mathcal{S}}^{(\boldsymbol{\theta})}(\tau) = \left(\sqrt{\dot{s}_0},\, \alpha_{\mathcal{H}}(\tau)\right)\,, \tag{5.20}$$

where $\alpha_{\mathcal{H}}$ is defined in equation (5.17). This immediate representation by the Frenet curvatures appears in the experiments not to be sufficiently elastic (Figure 5.3). It has somewhat the same weakness as the angle representation proposed in Srivastava and Klassen (2016), the Frenet curvatures being already independent of the parametrization.

4.3 Square Root Curvatures Transform

To overcome the "non-elasticity" issue of the representation defined above, we propose a second framework for shape analysis based on Frenet curvatures which uses, like the square root velocity function, the parametrization as a tool to register the curves and define a more "elastic" method. The latter is inspired by the square-root velocity transform of $SO(d)$-valued curves.

Definition 5.1 (SRV Transform for curves on $SO(d)$)**.** *Let $P \in C^\infty([0,1], SO(d))$. The Square Root Velocity transform of P is the map*

$$q(P)(t) = \frac{L_{P(t)^{-1}}\dot{P}(t)}{\sqrt{\|\dot{P}(t)\|_F}} = \frac{P(t)^T\dot{P}(t)}{\sqrt{\|\dot{P}(t)\|_F}}\,, \tag{5.21}$$

where $\|.\|_F$ is the Frobenius norm associated with the scalar product on the Lie Algebra of skew-symmetric matrices $\langle A, B\rangle = \frac{1}{2}tr(A^TB) = -\frac{1}{2}tr(AB)$, and $L_{P(t)^{-1}}$ is the left action of $SO(d)$ onto itself as defined in equation (2.4).

Let $x \in F([0,1],\mathbb{R}^d)$ and $Q(t) \in C^\infty([0,1], SO(d))$ be its associated Frenet path. Using the Frenet-Serret differential equation of Q, $Q'(s) = Q(s)A_{\boldsymbol{\theta}}(s)$, the SRV Transform of the Frenet path is

$$q(Q)(t) = \sqrt{\dot{s}(t)}\frac{A_{\boldsymbol{\theta}}(s(t))}{\sqrt{\|A_{\boldsymbol{\theta}}(s(t))\|_F}}\,. \tag{5.22}$$

Proposition 5.3. *For $\boldsymbol{\theta} \in \mathcal{H}$, we have*

$$\|A_{\boldsymbol{\theta}}(s(t))\|_F = \|\boldsymbol{\theta}(s(t))\|_2\,. \tag{5.23}$$

Proof. We recall that the vector $\boldsymbol{\theta}(s) = (\theta_1(s), \ldots, \theta_{d-1}(s))$ is a $(d-1)$-dimensional vector of $\mathbb{R}$. The matrix $A_{\boldsymbol{\theta}}(s)$ is skew-symmetric with the superdiagonal equal to $-\boldsymbol{\theta}(s)$, the subdiagonal equal to $\boldsymbol{\theta}(s)$, and all the other coefficients null. Observe that

$$\|A_{\boldsymbol{\theta}}(s(t))\|_F^2 = \frac{1}{2}\mathrm{tr}\left(A_{\boldsymbol{\theta}}(s)^T A_{\boldsymbol{\theta}}(s)\right) = -\frac{1}{2}\mathrm{tr}\left(A_{\boldsymbol{\theta}}(s)^2\right) = -\frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} a(s)_{ij}a(s)_{ji}\,.$$

We differentiate three cases depending on i:

$$\begin{cases} \text{if } i = 1: & \sum_{j=1}^{d} a(s)_{1,j}a(s)_{j,1} = a(s)_{1,2}a(s)_{2,1} = -\theta_1(s)^2 \\ \text{if } i = d: & \sum_{j=1}^{d} a(s)_{d,j}a(s)_{j,d} = a(s)_{d,d-1}a(s)_{d-1,d} = -\theta_{d-1}(s)^2 \\ \text{otherwise:} & \sum_{j=1}^{d} a(s)_{i,j}a(s)_{j,i} = a(s)_{i,i-1}a(s)_{i-1,i} + a(s)_{i,i+1}a(s)_{i+1,i} = -\theta_{i-1}(s)^2 - \theta_i(s)^2 \end{cases}$$

and then we have,

$$\begin{aligned} \|A_{\boldsymbol{\theta}}(s(t))\|_F^2 &= -\frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} a(s)_{ij}a(s)_{ji} = \frac{1}{2}\sum_{i=2}^{d-1}\theta_{i-1}(s)^2 + \frac{1}{2}\theta_{d-1}(s)^2 + \frac{1}{2}\sum_{i=2}^{d-1}\theta_i(s)^2 + \frac{1}{2}\theta_1(s)^2 \\ &= \sum_{i=1}^{d-1}\theta_i(s)^2 = \boldsymbol{\theta}(s)^T\boldsymbol{\theta}(s) = \|\boldsymbol{\theta}(s)\|_2^2\,. \end{aligned}$$

■

Based on the SRV Transform of a Frenet path and Proposition 5.3, we propose a new transformation of a parametrized curve, which we have called the *Square-Root Curvatures (SRC)* transform.

Definition 5.2 (Square-Root Curvatures Transform)**.** *Let $x \in F([0,1],\mathbb{R}^d)$. We consider its associated arc-length function $s(t)$ and Frenet curvatures $\boldsymbol{\theta}(s(t))$ defined as in Theorem 2.3. Then we define its square-root curvatures transform to be the map*

$$c(t) = \sqrt{\dot{s}(t)}\frac{\boldsymbol{\theta}(s(t))}{\sqrt{\|\boldsymbol{\theta}(s(t))\|}}\,. \tag{5.24}$$

The set of such square-root curvatures transforms is

$$\mathcal{C} = \left\{c \in \mathbb{L}([0,1],\mathbb{R}^{d-1}) \,|\, c_1, \ldots, c_{d-2} > 0\right\}\,, \tag{5.25}$$

which is the same as the set of admissible Frenet curvatures $\mathcal{H}$. We have already shown in the previous section that this set is a Riemannian manifold equipped with the $\mathbb{L}^2$ metric. Therefore, the geodesic distance between $c_0, c_1 \in \mathcal{C}$ is the $\mathbb{L}^2$ distance between them, and the geodesic path is a straight line. We define the following representation of a parametrized curve $x \in F([0,1],\mathbb{R}^d)$, from its Square-Root Curvatures transform, by

$$\mathcal{R}_{\mathrm{SRC}}(x)(t) = \left(\sqrt{\dot{s}(t)},\, c(t)\right)\,. \tag{5.26}$$

Proposition 5.4. *The map $\mathcal{R}_{SRC} : F([0,1],\mathbb{R}^d) \to \Psi([0,1]) \times \mathcal{C}$, defined above, is a bijection.*

Proof. This is again a result of Theorem 2.3 and Theorem 2.4. To get x from $\mathcal{R}_{\text{SRC}}(x)$, it should be noted firstly that $c(t)\|c(t)\| = \dot{s}(t)\boldsymbol{\theta}(s(t))$. From that, the skew-symmetric matrix function of the Frenet-Serret ODE can be reconstructed. By solving the corresponding Frenet-Serret ODE one gets the associated time parametrized Frenet path $Q(t)$. Then, using the first component of $\mathcal{R}_{\text{SRC}}(x)$, we get $x(t) = X(s(t)) = \int_0^t \dot{s}(u)T(s(u))du$. ■

The set of such square root curvature representations $\mathcal{R}_{\text{SRC}}$ is the Cartesian product $\Psi([0,1]) \times \mathcal{C}$ and therefore a Riemannian manifold with the product metric $d_\Psi \oplus d_\mathcal{C}$. This representation is, by definition, invariant under the action of $SO(d)$. Then, the corresponding shape space is the quotient space $\Psi([0,1]) \times \mathcal{C}/\text{Diff}([0,1])$. Let's $x \in F([0,1],\mathbb{R}^d)$ and $h \in \text{Diff}([0,1])$. The SRC representation of $\tilde{x} = x \circ h$ is

$$\mathcal{R}_{\text{SRC}}(\tilde{x}) = \left(\sqrt{\dot{s}} * h,\, c * h\right) = \mathcal{R}_{\text{SRC}}(x) * h \tag{5.27}$$

where $*$ is the group action defined in (5.6).

Proposition 5.5. *The metric on $F([0,1],\mathbb{R}^d)$ induced by the Riemannian metric on $\Psi([0,1]) \times \mathcal{C}$ defined by $d_{SRC} := d_\Psi \oplus d_\mathcal{C}$ is invariant under the action of $Diff_+([0,1])$.*

Proof. Let $h \in \text{Diff}_+([0,1])$, $x_0, x_1 \in F([0,1],\mathbb{R}^d)$ with $(\sqrt{\dot{s}_0}, c_0), (\sqrt{\dot{s}_1}, c_1) \in \Psi([0,1]) \times \mathcal{C}$ their corresponding square-root curvature transform. Under this representation the induced distance on $F([0,1],\mathbb{R}^d)$ is

$$d_{\text{SRC}}(x_0 \circ h, x_1 \circ h) = d_\Psi(\sqrt{\dot{s}_0} * h, \sqrt{\dot{s}_1} * h) + d_\mathcal{C}(c_0 * h, c_1 * h)\,.$$

Let us show that the action of $\text{Diff}_+([0,1])$ on $\Psi([0,1])$ and on $\mathcal{C}$ is by isometry with respect to the $\mathbb{L}^2$ metric. For $\Psi([0,1])$ we have

$$\left\langle \sqrt{\dot{s}_0} * h, \sqrt{\dot{s}_1} * h \right\rangle = \int_0^1 \left\langle \sqrt{\dot{s}_0(h(t))\dot{h}(t)}, \sqrt{\dot{s}_1(h(t))\dot{h}(t)} \right\rangle dt = \int_0^1 \left\langle \sqrt{\dot{s}_0(u)}, \sqrt{\dot{s}_1(u)} \right\rangle du\,, \quad u = h(t)$$

and with the same change of variable, we have for $\mathcal{C}$,

$$\begin{aligned}
\|c_0 * h - c_1 * h\|^2_{\mathbb{L}^2} &= \int_0^1 \left| \sqrt{\dot{h}(t)}\sqrt{\dot{s}_0(h(t))}\frac{\boldsymbol{\theta}_0(s_0(h(t)))}{\sqrt{\|\boldsymbol{\theta}_0(s_0(h(t)))\|}} - \sqrt{\dot{h}(t)}\sqrt{\dot{s}_1(h(t))}\frac{\boldsymbol{\theta}_1(s_1(h(t)))}{\sqrt{\|\boldsymbol{\theta}_1(s_1(h(t)))\|}} \right|_2 dt \\
&= \int_0^1 \left| \sqrt{\dot{s}_0(u)}\frac{\boldsymbol{\theta}_0(s_0(u))}{\sqrt{\|\boldsymbol{\theta}_0(s_0(u))\|}} - \sqrt{\dot{s}_1(u)}\frac{\boldsymbol{\theta}_1(s_1(u))}{\sqrt{\|\boldsymbol{\theta}_1(s_1(u))\|}} \right|_2 du\,, \quad u = h(t) \\
&= \int_0^1 |c_0(u) - c_1(u)|_2\, du\,.
\end{aligned}$$

Then d_{SRC} is invariant under the action of $\text{Diff}_+([0,1])$

$$\begin{aligned}
d_{\text{SRC}}(x_0 \circ h, x_1 \circ h) &= d_\Psi(\sqrt{\dot{s}_0} * h, \sqrt{\dot{s}_1} * h) + d_\mathcal{C}(c_0 * h, c_1 * h) \\
&= d_\Psi(\sqrt{\dot{s}_0}, \sqrt{\dot{s}_1}) + d_\mathcal{C}(c_0, c_1) = d_{\text{SRC}}(x_0, x_1)\,.
\end{aligned}$$

■

The distance on the shape space $\mathcal{S}$ under the representation $\mathcal{R}_{\text{SRC}}$, between two elements $X_0, X_1 \in \mathcal{S}$, is defined as

$$d_\mathcal{S}^{(\text{SRC})}(X_0, X_1) := \inf_{h \in \text{Diff}_+([0,1])} d_{\text{SRC}}(x_0, x_1 \circ h). \tag{5.28}$$

From the optimal wrapping function h^* the geodesic path on $\mathcal{S}$ between them is

$$\alpha_\mathcal{S}^{(\text{SRC})}(\tau) = \left(\tfrac{\sin((1-\tau)\vartheta)}{\sin(\vartheta)}\sqrt{\dot{s}_0} + \tfrac{\sin(\tau\vartheta)}{\sin(\vartheta)}(\sqrt{\dot{s}_1} * h^*)\,,\quad (1-\tau)c_0 + \tau(c_1 * h^*)\right)\,, \tag{5.29}$$

where $\vartheta = d_\Psi(\sqrt{\dot{s}_0}, \sqrt{\dot{s}_1} * h^*)$. The registration problem consider here is to find the minimizer h^* over $\mathrm{Diff}_+([0,1])$ such that

$$h^* = \underset{h \in \mathrm{Diff}_+([0,1])}{\arg\min} \int_0^1 \|c_0(t) - (c_1 * h)(t)\|^2 + \|\sqrt{\dot{s}_0(t)} - (\sqrt{\dot{s}_1} * h)(t)\|^2 dt \,. \tag{5.30}$$

Using the reformulation principle of Brunel and Park (2019), this registration problem is shown to be equivalent to finding $\gamma^* \in \mathrm{Diff}_+([0,1])$ such that

$$\gamma^* = \underset{\gamma \in \mathrm{Diff}_+([0,1])}{\arg\min} \int_0^1 \left\| \frac{\boldsymbol{\theta}_0(s)}{\sqrt{\|\boldsymbol{\theta}_0(s)\|}} - \sqrt{\gamma'(s)} \frac{\boldsymbol{\theta}_1(\gamma(s))}{\sqrt{\|\boldsymbol{\theta}_1(\gamma(s))\|}} \right\|^2 + \|1 - \gamma'(s)\|^2 ds \,. \tag{5.31}$$

Note that this reformulation has the form of a penalized registration problem. The second term represents a penalty term on γ and ensures a certain smoothness of the warping function. In this framework, the deformable templates are the square-root normalized curvatures which encode more geometric information than the unit tangent vector.

Remark 5.1. *One can consider the equivalent weighted penalized registration problem, where the second term is weighted by a certain $\sigma > 0$, without changing the global shape analysis framework under this representation.*

Remark 5.2. *In Park et al. (2022), we have used the following distance*

$$\int_0^1 \|\theta_0(s) - \gamma'(s)\theta_1(\gamma(s))\|^2 + \sigma \|1 - \gamma'(s)\|^2 ds \,. \tag{5.32}$$

This distance corresponds to the representation $\dot{s}(t)\theta(s(t))$ instead of the square root curvature transform $c(t)$. This is another possibility that has more or less the same properties, except that in this formulation, the action of $Diff_+([0,1])$ on $\mathcal{H}$ is not by isometry with the $\mathbb{L}^2$ metric. Although the penalty term helps avoid the pinching effect that may result from this, the corresponding metric will not be symmetric. This is why we prefer to consider here what we have defined as the square root curvature transform. However, it may be more stable for numerical applications, as the reconstruction map of the curve is more direct without the normalization part.

5 Experiments

In this section, we report the experimental results of the proposed methods, comparing them with the SRVF method. We use both synthetic and real data.

Remark 5.3. *In this work, we assume that the Frenet curvatures are perfectly known; however, this is rarely the case in practice. This is the main limitation of shape analysis methods that rely on Frenet curvatures, as estimation of the Frenet curvatures is not straightforward. We refer here to the methods proposed in Chapter 3 and Chapter 4 for smooth estimation of Frenet curvatures from various types of observations, as well as the references contained therein.*

5.1 Experiments with synthetic curves

We use synthetic data to highlight the differences between the methods discussed above (SRVF, SRC, and Frenet curvatures). The computations related to the SRVF method are made with the package fdasrsf. The registration problems are solved with a dynamic programming algorithm (Bertsekas, 1995). We consider three types of synthetic curves: spirals in $\mathbb{R}^2$ (Section 5.1.2), circular helices in $\mathbb{R}^3$ (Section 5.1.3), and loops in $\mathbb{R}^2$ characterized by a curvature function with one large peak (Section 5.1.1). For each one,

we propose to visualize the geodesic paths under each method (SRVF, SRC, Frenet curvatures) between two particular curves chosen to have either significantly different characteristics (significantly different numbers of spins for the two spirals and helices, and peaks of curvature quite far apart for the two loops), or conversely slightly closer ones. We also display, in both cases, the evolution of the curvatures of the curves along the geodesic. In addition, for each type of the curve, we generate a set of 20 random curves from which we compute the matrices of pairwise distances utilizing the three different distances again.

The following results highlight the advantages of the SRC representation of Euclidean curves in defining a curve shape analysis framework: being sufficiently "elastic" to avoid the occurrence of artefacts along the geodesic or in the mean (Section 5.1.1), and being consistent when analyzing a set of curves sharing a particular geometry (Section 5.1.1, 5.1.2 and 5.1.3).

5.1.1 Loop in $\mathbb{R}^2$

We consider the simple case of a set of 20 curves in $\mathbb{R}^2$ with a single large peak of curvature. This one is created by generating curvature functions on $[0, 1]$ with one peak of maximum value 60.5, width 0.15, and location chosen randomly between 0.1 and 0.9. These curves have the shape of a loop made with a wire, where the loop is more or less close to the right or left wire end, depending on the location of the curvature peak. We compare the three methods through the pairwise distance matrices in Figure 5.5, through the geodesic paths computed between two of these curves with close peak locations in Figure 5.4 and between two more distant peaks in Figure 5.3. The corresponding deformations through the variations of the curvature along the different geodesic paths on Euclidean curves are shown in these figures and highlight the strengths and weaknesses of each method. First, it reveals "non-elasticity" of the unparametrized Frenet curvatures method: in the middle of the geodesic path in Figure 5.3 and 5.4, we have two peaks of curvature and, therefore, a completely different shape without any loop. This explains the inconsistency of the heatmap under this method (Figure 5.5). Therefore, we clearly see with this result that the simple Frenet curvature representation is not sufficient and that considering a representation that allows for registration of the points along the curve seems to be a more than necessary criterion to have a coherent shape analysis framework. Conversely, there is an elastic deformation of the curvature with the SRC transform, and shapes along the geodesic are consistent in both cases with the set of curves considered, which is well summarized on the corresponding heatmap where all distances are rather close to zero. For the SRVF method, the chosen example with distant peaks of curvature (Figure 5.3) exhibits artifacts along the geodesic, with small loops at the edges of the middle curve. However, when considering closer peaks of curvature (Figure 5.4), the SRVF geodesic paths show more coherent curves, each having a single peak of curvature. This significant difference in curve shapes along SRVF geodesic paths between both cases may explain the observed inconsistency in the SRVF heatmap (Figure 5.5), where distances are not monotonous with respect to the spacing between curvature peaks, leading to unreliable results.

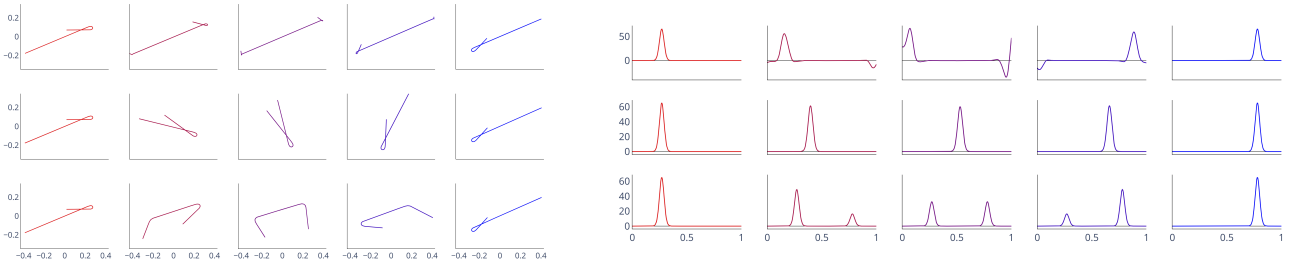


Figure 5.3: Geodesic paths (left) between two loops with curvature peaks located at 0.27 (red) and 0.78 (blue) and corresponding curvatures (right) of each curve along the geodesics: SRVF (1^{st} row), SRC (2^{nd} row), and Frenet curvatures (3^{rd} row).

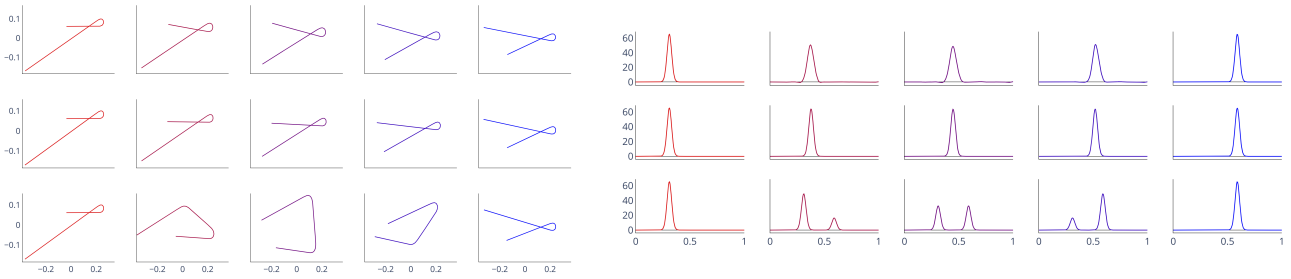


Figure 5.4: Geodesic paths (left) between two loops with curvature peaks located at 0.31 (red) and 0.59 (blue) and corresponding curvatures (right) of each curve along the geodesics: SRVF (1^{st} row), SRC (2^{nd} row), and Frenet curvatures (3^{rd} row).

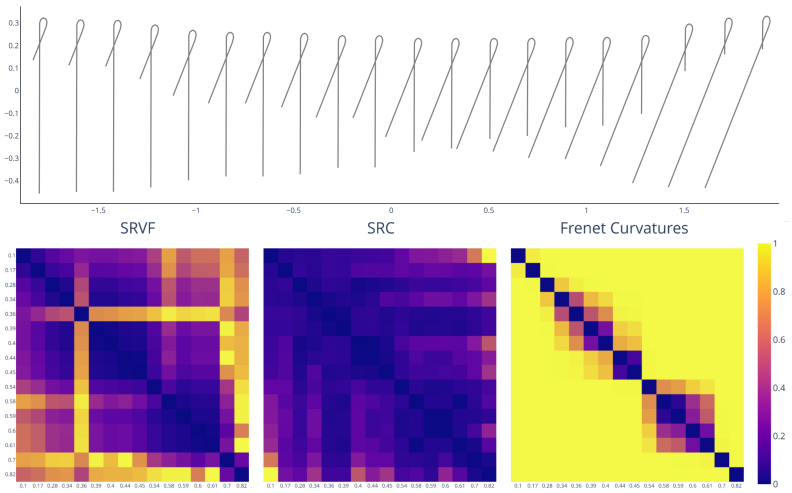


Figure 5.5: Matrices of pairwise distances computed under each method (SRVF, SRC, and Frenet curvatures) from a set of 20 loops, with random locations of curvature peak, plotted on the top.

5.1.2 Spiral in $\mathbb{R}^2$

We consider a curvature function of the type $t \mapsto a \log(t+2)$, where $a \in \mathbb{R}_+^*$. This curvature defines a curve having a shape of a helix with more or less spins depending on the value of the coefficient a. We show in Figure 5.7 the geodesic paths computed under each framework between two spirals with a similar number of spins and in Figure 5.6 with a significantly different number of spins. In this case of spirals, the SRC and Frenet curvatures methods give really similar results and appear to be really consistent in the shape of the curves along the geodesic. On the contrary, the curves along the SRVF geodesic are no longer spirals in the case of Figure 5.6. They are more coherent along the SRVF geodesic in Figure 5.7 between two spirals with a less different number of spins, but the curvatures still show a peak at the end that is not present in the two considered curves. In both figures, we see that methods based on a representation using the Frenet curvatures are more suitable and consistent in this case. Again, we generate 20 spirals by selecting a random coefficient a between 7 and 37. The corresponding heatmaps under each framework are displayed in Figure 5.8. The distance matrices show that the Frenet curvatures-based methods calculate distances globally close to zero within this set. They, therefore, seem to capture that the curves share a common geometry. This is less the case for the SRVF distance matrix, which seems to well partition the spirals according to the number of spins, but the distances are not always monotone as a function of these ones.

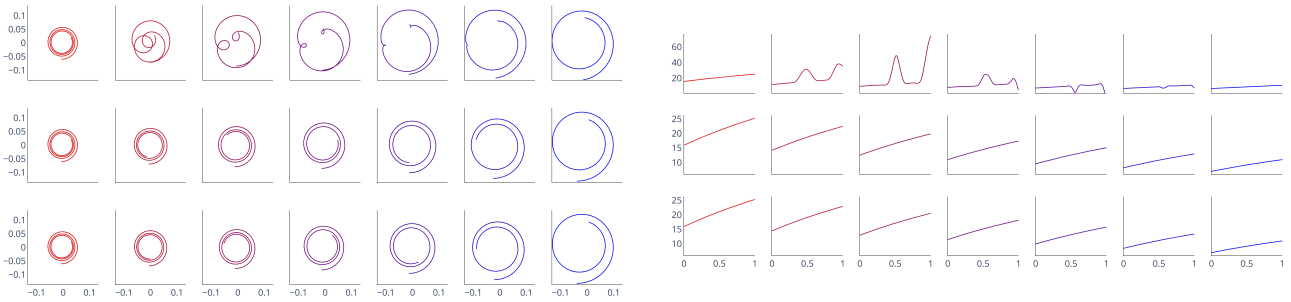


Figure 5.6: Geodesic paths (left) between two spirals with a different number of spins and corresponding curvatures (right) of each curve along the geodesics: SRVF (1^{st} row), SRC (2^{nd} row), and Frenet curvatures (3^{rd} row).

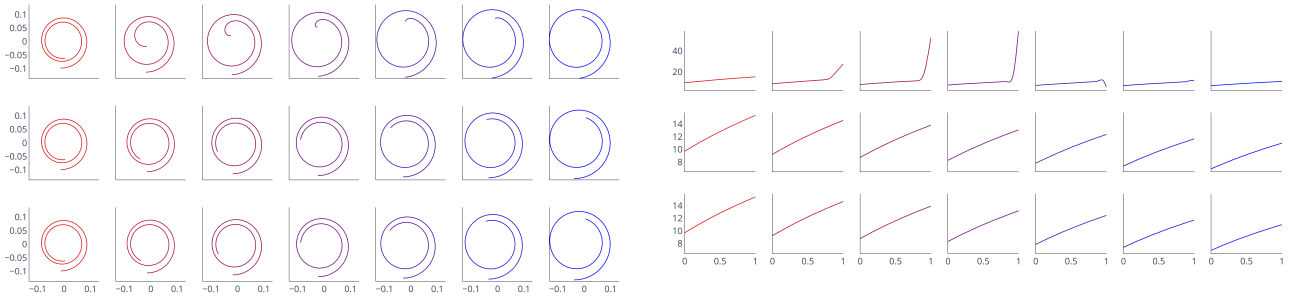


Figure 5.7: Geodesic paths (left) between two spirals with a similar number of spins and corresponding curvatures (right) of each curve along the geodesics: SRVF (1^{st} row), SRC (2^{nd} row), and Frenet curvatures (3^{rd} row).

5.1.3 Circular helix in $\mathbb{R}^3$

We also propose an example in 3-dimension with the special case of circular helices. A three-dimensional circular helix is characterized by having a constant curvature and torsion. Again, the three geodesic paths are compared in the case of similar (Figure 5.9) or different numbers of spins (Figure 5.10). The analysis and comparison of the three methods for helices in $\mathbb{R}^3$ are almost identical as in the case of spirals in $\mathbb{R}^2$. Again, the geodesic under the Frenet curvature representation is very similar to SRC, and both methods conserve the specific geometry of a helix along the geodesic. Conversely, within both geodesic paths under SRVF method, the curves lose the characteristic geometry of the helix as curves no longer have a constant curvature and torsion. In that case, the 20 random circular helices, Figure 5.11, are generated from a random number of spins, and they are all of unit length. We observe again that the SRVF framework is a bit more coherent between helices with a similar number of spins in Figure 5.9. Here also, in three dimensions, the Frenet curvatures-based methods define a shape analysis framework well adapted to this particular geometry.

5.2 Application to sign language motion data

It appears that curvilinear velocity and Frenet curvatures are particularly relevant parameters for the analysis of human motion. Several laws involving these parameters can be found in the literature (Lacquaniti et al., 1983; Maoz et al., 2009; Pollick et al., 2009; Flash and Berthoz, 2021); among others, the power laws state a special relationship between the curvature, the torsion, and the velocity of a point trajectory representing human motion. Therefore, using a method that conserves the shape of

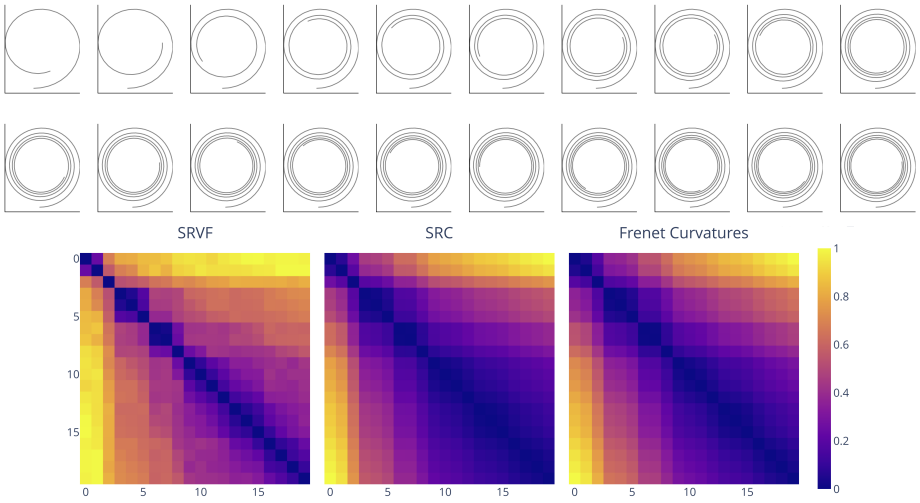
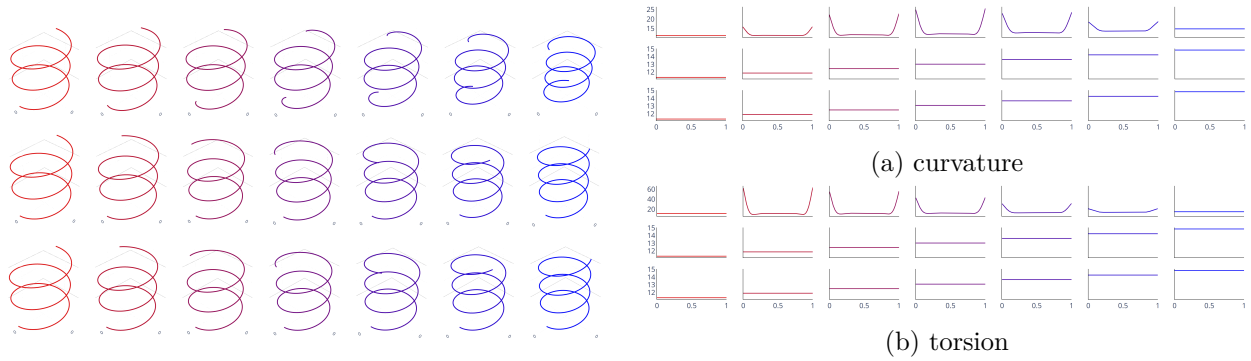


Figure 5.8: Matrices of pairwise distances computed under each method (SRVF, SRC, and Frenet curvatures) from a set of 20 random spirals plotted on the top.

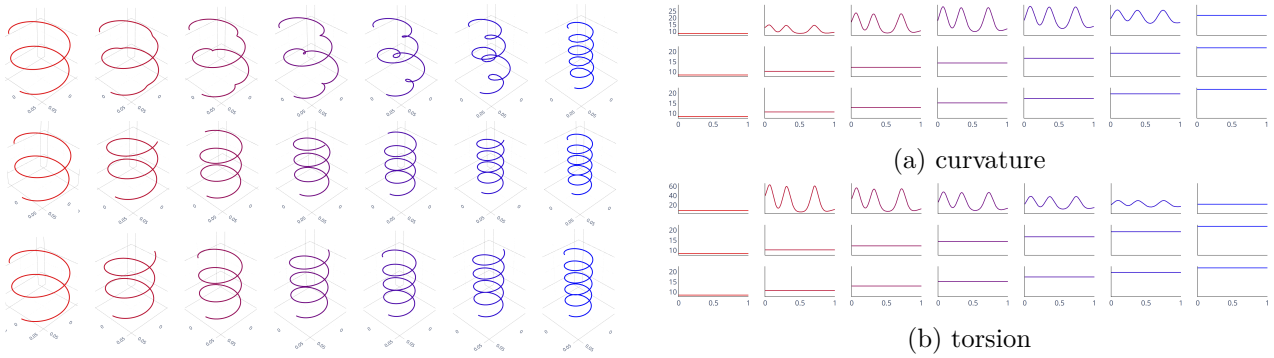


(a) curvature

(b) torsion

Figure 5.9: Geodesic paths (left) between two circular helices in 3D with a similar number of spins and corresponding curvatures (right) of each curve along the geodesics: SRVF (1^{st} row), SRC (2^{nd} row), and Frenet curvatures (3^{rd} row).

these parameters is of particular interest in this application. We demonstrate here with the case of wrist trajectories in sign language, acquired with a motion capture system by the company MocapLab. We compute the geodesic paths, under each of the frameworks, between the arbitrarily chosen red and blue curves within the set of several repetitions of the sign "Femme" and "Europe", shown respectively in Figure 5.12 and Figure 5.15, with the corresponding time-parameterized Frenet curvatures. Figure 5.13 and 5.16 highlight the benefits of considering a representation that depends on the parameterization, enabling registration before geodesic computation. However, solely relying on the tangent vector as a representative object (SRVF) proves inadequate in finding the optimal reparametrization to align torsions accurately. This limitation could adversely affect subsequent SRVF method analyses, such as computing the mean, leading to the emergence of new minimums, maximums, and zeros in the torsion functions along the SRVF geodesic. These characteristic points are vital in observing the laws of motion. Consequently, it is preferable to use a method, like the proposed SRC method, which directly optimizes the optimal alignment from these parameters while preserving these characteristics. In Figure 5.16, we observe that the alignment obtained by the SRC method better preserves the curves' geometry, compared to the SRVF geodesic path. Additionally, the artefacts present in the synthetic curve experiments, resembling peaks of Frenet curvatures at the boundaries, are also noticeable in this case.



(a) curvature

(b) torsion

Figure 5.10: Geodesic paths (left) between two circular helices with a different number of spins and corresponding curvatures (right) of each curve along the geodesics: SRVF (1^{st} row), SRC (2^{nd} row), and Frenet curvatures (3^{rd} row).

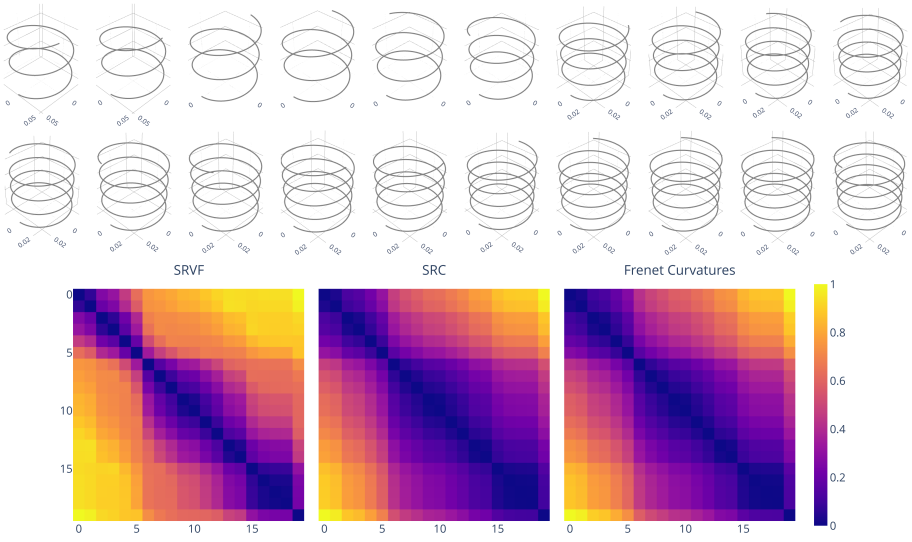


Figure 5.11: Matrices of pairwise distances computed under each method (SRVF, SRC, and Frenet curvatures) from a set of 20 random circular helices plotted on the top.

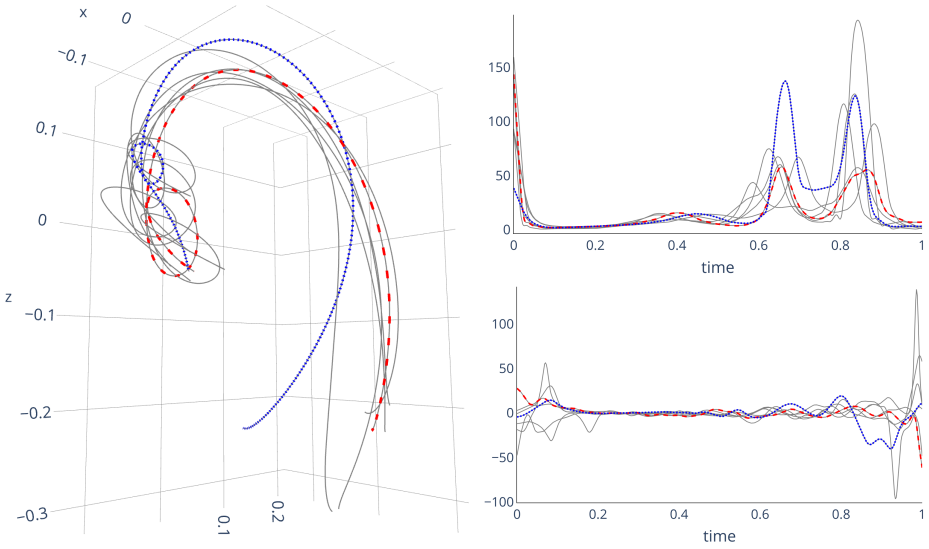


Figure 5.12: Trajectories of right wrist while signing "Femme" in sign language: 3D curves (left), arc-length parametrized curvatures (top right), torsions (bottom right). The blue and red ones are used to compute the geodesic in Figure 5.13.

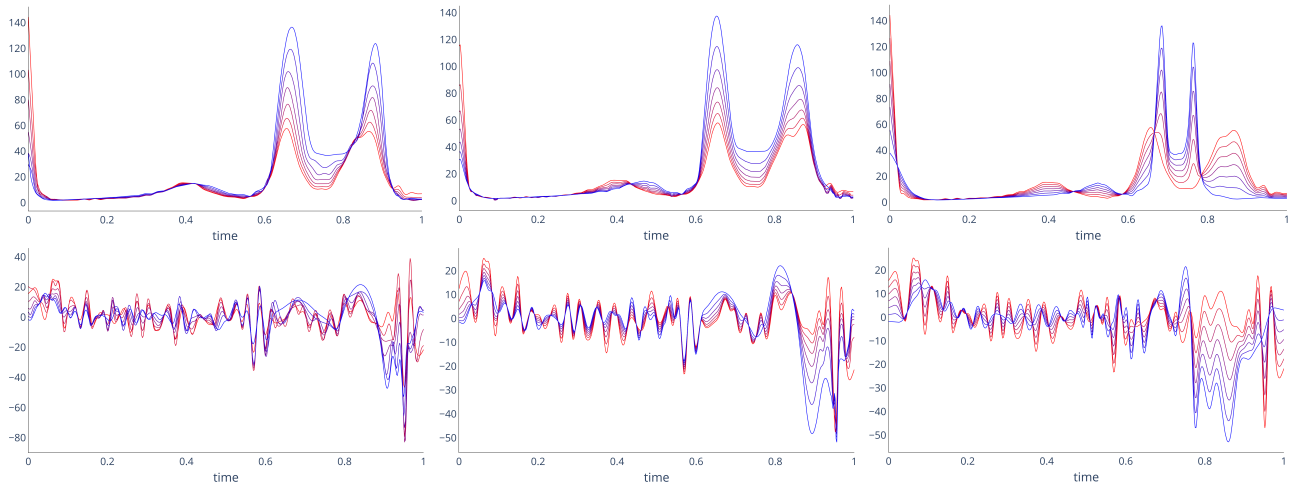


Figure 5.13: Comparison between time-parametrized curvature and torsion along the geodesic path under SRVF (left), SRC (middle), and Frenet curvatures (right).

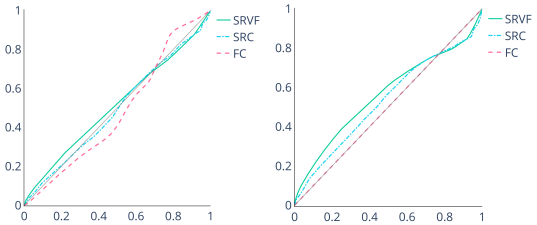


Figure 5.14: Comparison of estimated warping functions h (left) and γ (right) to compute Figure 5.13.

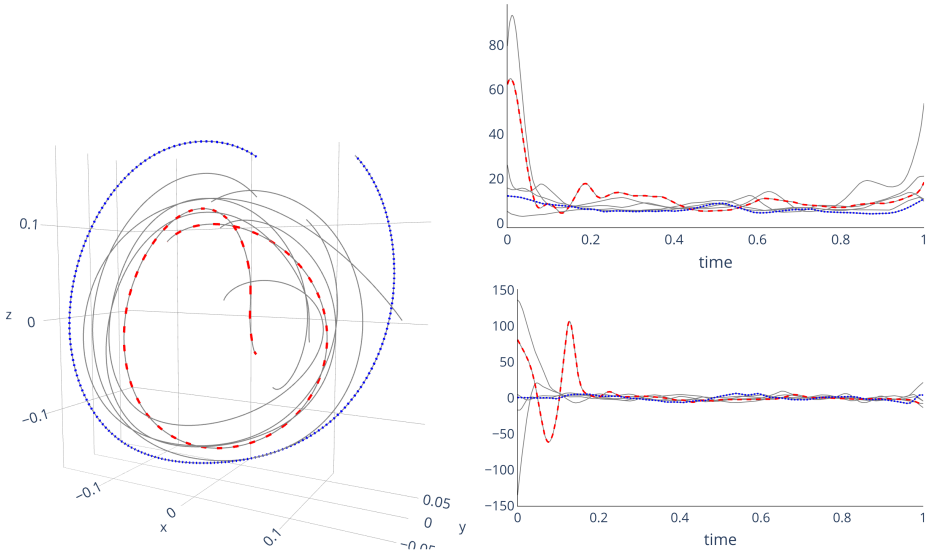


Figure 5.15: Trajectories of the right wrist while signing "Europe" in sign language: 3D curves (left), arc-length parametrized curvatures (top right), torsions (bottom right). The blue and red ones are used to compute the geodesic in Figure 5.16.

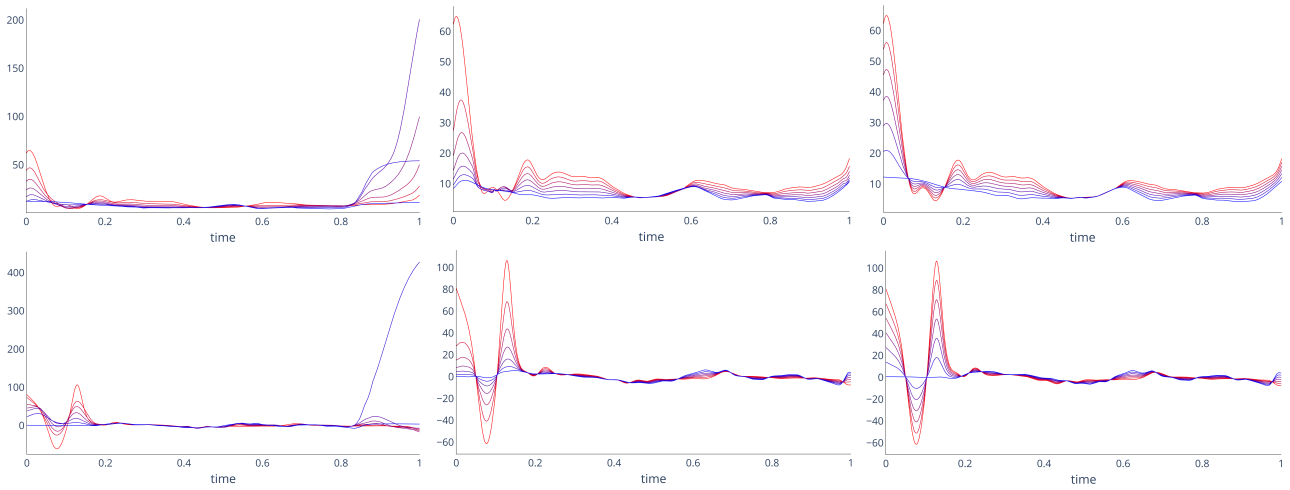


Figure 5.16: Comparison between time-parametrized curvature and torsion along the geodesic path under SRVF (left), SRC (middle), and Frenet curvatures (right).

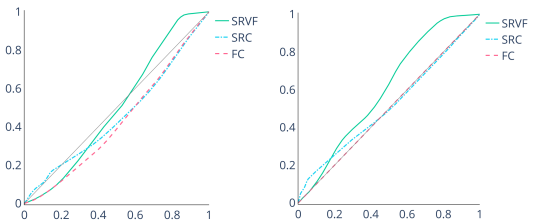


Figure 5.17: Comparison of estimated warping functions h (left) and γ (right) to compute Figure 5.16.

6 Conclusion

We have developed a new geometric framework for shape analysis of Euclidean curves in $\mathbb{R}^d$. This one is based on the square-root curvature transform of a Euclidean curve and the associated Riemannian framework. This representation encodes more geometric information of the curves, giving results easier to interpret than existing methods. Our experiments show that, even in dimension 2, this representation gives results more consistent with the underlying geometry of the curves. The main limitation lies in the estimation of the Frenet curvatures from real and noisy data which is much more computationally expensive and more unstable than the estimations required by the SRVF-based method. Nevertheless, recent smooth statistical estimators can be used for computing the SRC (Sangalli et al., 2009; Park et al., 2022), as the ones we detailed in Chapter 3 and Chapter 4. We believe our method is particularly interesting for motion trajectory analysis and could be developed further in the future as a tool for the generation, segmentation, and classification of complex trajectories.

6

Statistical Frenet-Serret Mean Shape

Abstract

The statistical analysis of a population of curves in $\mathbb{R}^d$ involves defining a mean shape and modeling variations around it. The classical approach to shape analysis considers the Karcher mean in the shape space, defined as a Riemannian manifold associated with a specific representation of curves. In this framework, three Karcher means in the space of SRVF, square-root curvature (SRC) transforms, and Frenet curvatures (FC) can be considered. We highlight the statistical challenge inherent in computing these Karcher means based on unknown parameters in a typical application scenario where only noisy observations of curves are available. We compare an approach based on preprocessing, involving efficient FDA-based estimation of each Frenet curvatures individually, with a regularization-based approach derived from the works of Park and Brunel (2019), utilizing a unique criterion. Drawing on their statistical characterization of the mean Frenet curvatures parameter, we define three optimization criteria leading to three distinct mean shapes, two of which approximate the FC and SRC Karcher means. Finally, we assess and compare the different means and estimation methods using simulated data and examples of sign language movement trajectories.

Contents

1 Introduction

In this chapter, we delve into the challenge of analyzing a population of curves in $\mathbb{R}^d$ under the Frenet-Serret framework, inspired by the principles of functional data analysis. Over recent years, there has been a growing interest in multivariate functional data analysis due to the surge in data represented as multivariate curves. Various domains have highlighted this interest, ranging from registering multivariate growth curves (Carroll et al., 2020), employing multivariate functional mixed modeling for spectroscopy data (Zhu et al., 2017), to solving complex issues like cross-correlating multivariate human gesture tracking (Zhang et al., 2020). Our motivating example deals with the sign language movement signals automatically captured by a motion capture system. These types of movement are challenging to study as they are meaningful but are difficult to characterize without specific knowledge in the field. Our aim is to develop statistical tools to extract "primitives" or templates specific to the nature of the signals studied. Achieving this involves identifying commonalities among curves and quantifying significant variations within the curve population. Hence, like any statistical analysis, establishing a representative mean for the population and assessing deviations from this mean is crucial.

Statistically analyzing a curve population often involves examining their geometric variations through shape analysis. Shapes, treated as points on a manifold, represent what remains invariant under rigid transformations of Euclidean space and then are formally equivalence classes under appropriate group actions. Diverse methods for constructing infinite-dimensional shape spaces have been proposed, as discussed in Chapter 5 and in existing literature (Younes, 2010; Srivastava et al., 2011). Shape analysis focuses on finding a curve representation better suited for comparing distinct curves, transforming the statistical analysis into that of these new objects. The shape space is established as a Riemannian manifold, with a proper Riemannian distance and geodesic for direct curve comparisons based on these representative objects. Adapting classical statistical methodologies is essential due to the non-Euclidean properties of shape spaces, where the natural extension of the usual mean is defined as a Karcher mean (Karcher, 1977). Some of these concepts, particularly under the square-root velocity function (SRVF) representation, have been proposed for analyzing functional data variations, especially in one-dimensional curve registration (Kurtek et al., 2012). Its extension to multi-dimensional curves is discussed in Srivastava and Klassen (2016). In Chapter 5, we introduce two new representations of Euclidean curves using the Frenet-Serret framework: the unparametrized Frenet curvatures (FC) and the Square-Root Curvature (SRC) transform. Both representations define a Riemannian geometry on shape space. Within the classical shape analysis framework, this chapter introduces the associated theoretical Karcher means for both representations, defining two new mean shapes that fully exploit curve geometry through the Frenet-Serret framework. Therefore, these mean shapes can be effectively compared with the state-of-the-art SRVF mean shape.

However, the primary focus of this chapter is to address statistical challenges associated with estimating these means in practical scenarios. In a typical application setting, only Euclidean curves are observed discretely and potentially with noise. Consequently, none of these representations (SRVF, SRC,

Frenet curvatures) are directly accessible and must be accurately estimated beforehand. While this might not be a central concern in the literature that only involves the SRVF representation, it becomes crucial when dealing with representations based on Frenet curvatures due to their estimation complexity and sensitivity to observation noise. Indeed, computing the SRVF only requires estimating the first curve's derivative, a task efficiently accomplished with classical FDA techniques. In contrast, as discussed in Srivastava and Klassen (2016) and Part I of this thesis, estimating Frenet curvatures necessitates estimating the d first derivatives, a process prone to instability even in scenarios with minimal observation noise. Therefore, this chapter introduces and compares two approaches to accurately approximate these means while accounting for potential noise. The first approach, commonly employed in shape analysis, involves estimating Frenet curvatures in an efficient preprocessing step. This implies individually estimating Frenet curvatures for each curve in the population using an efficient criterion that adequately models constraints associated with estimating a functional parameter in infinite dimensions. If the preprocessing method proves effective or the data exhibits minimal noise, applying Karcher mean formulas to these estimators is expected to yield satisfactory results. However, in other cases, independently estimating each representation can introduce bias, thereby impacting the quality of means.

Consequently, in this chapter, we present an alternative approach based on directly modeling global noise through the definition of a unique criterion with regularization for estimating the mean parameter. This alternative approach is introduced in Park and Brunel (2019). They treat multidimensional curves as a special case of multivariate functional data, drawing inspiration from standard assumptions in FDA (Ramsay and Silverman, 2005; Ferraty and Vieu, 2006; Wang et al., 2016). These approaches assume a common structure, often defined by a mean function and a variance function, enabling variability decomposition through functional principal component analysis. Successful applications of these ideas to motion data have been demonstrated (Goldsmith and Kitago, 2016; Backenroth et al., 2018). While this linear approximation is powerful for extending univariate tools to multivariate ones (Chiou et al., 2014; Happ and Greven, 2018), it can sometimes obscure key features in such data (Dai and Genton, 2018). Recent developments in FDA focus on generalizations beyond Euclidean data by allowing for non-standard features, such as data on a manifold or in a general metric space (Lin and Yao, 2019; Dubey and Müller, 2019; Petersen and Müller, 2019). Thus, Park and Brunel (2019) propose an alternative statistical characterization of the common structure based on the geometric features of curves, namely Frenet paths and curvatures. Specifically, they introduce the notion of the mean vector field associated with the Frenet-Serret differential equation, thereby statistically characterizing the mean Frenet curvature parameter. By placing their approach within a shape analysis framework, we demonstrate that the optimization criterion they propose approximates a regularized optimization criterion for the mean Karcher in the space of unparametrized Frenet curvatures. Exploiting this approach, we derive two new optimization criteria with regularization, having "elastic" properties and thus, defining more relevant mean Frenet curvatures and associated mean shapes. In particular, we show that one of the proposed new criteria approximates the Karcher mean in the space of SRCs through this "regularization-based" approach.

Subsequently, we develop an efficient estimation algorithm in each case. Finally, as an extension, we demonstrate that this last approach can be adapted to spherical curves through a straightforward reformulation of the Frenet framework, providing a mean curve lying on the sphere. We provide a detailed comparison of these different means and approaches for estimating them based on experiments conducted with simulated curves, and we demonstrate the significance of these Frenet-based mean shapes and methods in the study of a set of sign language motion trajectories.

2 Mean Characterization within a Shape Analysis Framework

In this section, we present the classical framework used in shape analysis to define the mean shape of a population of curves in $\mathbb{R}^d$. We consider a population of N curves in $\mathbb{R}^d$ defined by the set of functions $\mathcal{E} = \{x_1, \ldots, x_N\}$, such that for all $k = 1, \ldots, N$, $x_k \in \mathbb{L}^2([0,1], \mathbb{R}^d)$.

2.1 Shape Variations in a Population of Curves

Defining the mean shape of such a set of functions $\mathcal{E}$, first requires properly identifying the distinct types of variations. This is a fundamental concept in functional data analysis. Typically, variability in a set of functional data is decomposed into two types of variations: *phase* variations and *amplitude* variations (Ramsay and Silverman, 2005). This is easily understandable and visible in the case of scalar functions from $[0,1]$ to $\mathbb{R}$ (Figure 2.1). Amplitude variability involves changes in the magnitudes of specific features, like a peak, while disregarding their timing. On the other hand, phase variability refers to variations in the timing of these features without accounting for their magnitudes. We refer to Section 2.1.3 of Chapter 2 for more details on this point in the case of scalar functions. For a set of multidimensional functions, this interesting separation of variability can be interpreted more as a separation between *phase* variations and *shape* variations. Indeed, we recall here that the *shape* of the curve is defined as what is left invariant under reparametrization. For regular curves x_k, the shape function X_k is identified with the arc-length parameterized curves as $x_k(t) = X_k(s_k(t)), k = 1, \ldots, N$. The arc-length parametrization should not be confused with the standard representation of phase variation in the functional data. Suppose that x_k is given as $x_k(t) = x(h_k(t))$, where h_k are time-warping functions. As $\dot{x}_k(t) = \dot{x}(h_k(t))\dot{h}_k(t)$, by change of variables, the corresponding arc-length function can be expressed as

$$s_k(t) = \int_0^t \|\dot{x}(h_k(t))\dot{h}_k(t)\|\, dt = \int_0^{h_k(t)} \|\dot{x}(u)\|\, du = s(h_k(t))\,.$$

It follows that $x_k(t) = x(h_k(t)) = X(s(h_k(t)) = X(s_k(t))$, that is, the *shape* of the curve is preserved under time warping. In this work, we distinguish between phase (h_k) and shape (X_k) variation. An additional source of variability in $\mathcal{E}$ can arise from the varying lengths of curves $\{l_1, \ldots, l_N\}$, where $l_k = s_k(1)$. This work assumes that the curves within $\mathcal{E}$ all have a comparable meaningful start and end. Therefore this source of variability is not meaningful for computing the mean shape, and thus we consider instead the set of length normalized curves $\tilde{\mathcal{E}} = \{\tilde{x}_1, \ldots, \tilde{x}_N\}$, where $\tilde{x}_k(t) = x_k(t)/l_k$. Rescaling does change the geometry only through a scaling factor: $\tilde{X}_k = X_k/l_k$, and the effect of scaling on Frenet framework components is detailed in Section 3 of Chapter 2. From now on, all curves are supposed to be scaled to the unit length. Moreover, the *geometric shape* of the curves in $\mathcal{E}$ can be the same modulo rigid transformations such as translations or rotations. However, these rigid transformations are not of direct interest when we compare the inherent shape of the curves. Then, these additional variabilities are not included in what we call *shape* variations and must be modeled separately.

2.2 Karcher Mean Shape under Specific Riemannian Geometries

Having identified some important types of variation in the functional data set $\mathcal{E}$, we seek to define an appropriate mean shape. For all $x_k \in \mathcal{E}$, we have $x_k \in \mathbb{L}^2([0,1], \mathbb{R}^d)$ $(d \geq 2)$. The set of square-integrable functions $\mathbb{L}^2$ is a Hilbert space that is also a Riemannian manifold equipped with the $\mathbb{L}^2$ inner product: for $x_1, x_2 \in \mathbb{L}^2$,

$$\langle x_1, x_2 \rangle = \int_0^1 \langle x_1(t), x_2(t) \rangle_2 dt, \tag{6.1}$$

and whose tangent space is $T_{x_1}(\mathbb{L}^2) = \mathbb{L}^2$. Karcher (1977) introduced a widely used approach for establishing a mean on a manifold, utilizing the centroid of a density as its representative point.

Definition 6.1 (Karcher Mean, Karcher (1977))**.** *Let M be a Riemannian manifold. The Karcher mean μ of a probability density function f on M is defined as a local minimizer of the cost function:*

$$\rho(p) = \int_M d(p,q)^2 f(q) dq,$$

where dq denotes the reference measure used in defining the probability density f on M. The value of ρ at the Karcher mean is called the Karcher variance. Now, if we consider rather a finite set of independent

random samples from the density f: $p_1, \ldots, p_k \in M$, one can define the sample Karcher mean of these points $\hat{\mu}$ as the local minimizer of the function:

$$\rho_k(p) = \frac{1}{k} \sum_{i=1}^{k} d(p, p_i)^2,$$

where d is the geodesic distance on M.

The Karcher mean on a Riemannian manifold can be computed numerically with the following iterative algorithm.

Algorithm 3 Karcher mean on a Riemannian manifold M (Srivastava and Klassen, 2016)

Let $p_1, \ldots, p_k \in M$. Let μ_0 be an initial estimate of the Karcher mean. Set $j = 0$.

1: For each $k = 1, \ldots, N$, compute the tangent vector v_i such that $v_i = \exp_{\mu_j}^{-1}(p_i)$.
2: Compute the average direction $\bar{v} = \frac{1}{N} \sum_{k=1}^{N} v_i$.
3: If $\|\bar{v}\|$ is small, then stop. Else, update μ_j in the update direction using $\mu_{j+1} = \exp_{\mu_j}(\epsilon \bar{v})$, where $\epsilon > 0$ is small step size (typically 0.5).
4: Set $j = j + 1$ and return to 1.

Given this definition, we can then define the mean curve of $\mathcal{E}$ directly as the Karcher mean function in the Riemannian manifold $\mathbb{L}^2([0,1], \mathbb{R}^d)$, defined as the minimizer of

$$\arg\min_{x \in \mathbb{L}^2} \left(\sum_{k=1}^{N} \int_0^1 (x_i(t) - x(t))^2 dt \right). \tag{6.2}$$

This problem can be solved separately for each time point, given the *cross-sectional* or simply arithmetic sample mean of $\{x_1(t), \ldots, x_N(t)\}$ that is

$$\bar{x}(t) = \frac{1}{N} \sum_{k=1}^{N} x_k(t), \qquad \text{for all } k \in [0,1]. \tag{6.3}$$

The *arithmetic mean shape* can then be defined as the shape $\mu^{(\text{arithm})}$ of the mean: $\bar{x}(t) = \mu^{(\text{arithm})}(\bar{s}(t))$, where $t \mapsto \bar{s}(t)$ is the arc-length function of $\bar{x}$. However, this simple cross-sectional mean of $\mathcal{E}$ in $\mathbb{L}^2([0,1], \mathbb{R}^d)$ does not take into account potential variations in phase and shape among the curves. Ignoring these variations can lead to a distorted or misleading mean representation, as the averaging process does not appropriately account for these inherent differences and is dependent on rigid transformations. As outlined in Chapter 5, the focus of shape analysis is, therefore, to define the shape space as a Riemannian manifold from an appropriate mathematical representation of the curves that remains invariant under shape-preserving transformations (translation, rotation, scaling, and reparametrization). The Karcher mean is then derived from this Riemannian geometry. This principle is used by Srivastava et al. (2011) to define an alternative elastic mean shape of $\mathcal{E}$ based on the SRVF representations of curves.

Karcher mean in the SRVF shape space. Suppose that $x_k \in AC_0([0,1], \mathbb{R}^d)$ for all $k = 1, \ldots, N$. Using the geodesic distance $d_{\mathcal{S}}^{(\text{SRVF})}$ defined in equation (5.8), the SRVF mean $\mu^{(\text{SRVF})} \in AC_0([0,1], \mathbb{R}^d)$ is defined as

$$\mu^{(\text{SRVF})} = \arg\min_{\mu} \sum_{k=1}^{N} d_{\mathcal{S}}^{(\text{SRVF})}(x_k, \mu)^2. \tag{6.4}$$

In practice, this comes down to computing the Karcher mean in the space of square-root velocity functions. Let $q_k = \mathcal{R}_{\text{SRVF}}(x_k)$ be the SRVF of the curve $x_k \in \mathcal{E}$. The problem that is effectively solved is then

$$q_\mu^* = \arg\min_{q_\mu} \sum_{k=1}^{N} \inf_{\substack{O_k \in SO(d) \\ h_k \in \text{Diff}([0,1])}} \int_0^1 \left\| q_\mu(t) - O_k q_k(h_k(t)) \sqrt{\dot{h}_k(t)} \right\|_2^2 dt, \tag{6.5}$$

and $\mu^{(\text{SRVF})} = \mathcal{R}_{\text{SRVF}}^{-1}(q_\mu^*)$. It can be computed numerically using Algorithm 3. The tangent vectors in step 1 are computed as follows: find the optimal (O_k^*, h_k^*) that minimizes the distance between q_{μ_j} and q_k, set $q_k^* = O_k^* \sqrt{\dot{h}_k^*}(q_k \circ h_k^*)$, and compute:

$$v_k = \frac{\vartheta_k}{\sin(\vartheta_k)}(q_k^* - \cos(\vartheta_k) q_{\mu_j}), \quad \text{where} \quad \cos(\vartheta_k) = \langle q_{\mu_j}, q_k^* \rangle.$$

Moreover, the exponential map is given by $\exp_{q_\mu}(\epsilon \|\bar{v}\|) = \cos(\epsilon\|\bar{v}\|) q_\mu + \sin(\epsilon\|\bar{v}\|) \frac{\bar{v}}{\|\bar{v}\|}$.

The Karcher means defined in this section are formulated in terms of parameters or mathematical representations of individual curves, such as the SRVF. However, they presume precise knowledge of these parameters, which is not always the case in real-world scenarios. Especially when dealing with noise in discrete curve observations, the calculation of these means relies on statistical estimators for these representations, introducing potential non-negligible estimation errors that can significantly impact the quality of the means. This preprocessing step, necessary to obtain estimators of the individual curve representations under consideration, is typically not a primary focus in the literature on shape analysis, as the estimation of the curve's first derivative for SRVF computation is generally considered sufficiently robust, employing classical smoothing techniques. However, in our context, the objective is to define a mean shape based on the individual Frenet curvatures of a set of curves. As previously noted by Srivastava and Klassen (2016) and emphasized in Chapter 3 and Chapter 4, due to their dependence on higher-order derivatives, Frenet curvatures are particularly sensitive to observation noise. Therefore, deriving mean shapes from the Frenet curvatures of curves poses a statistical challenge that necessitates a cautious and robust approach within the framework of functional data modeling, given that Frenet curvatures are functions in an infinite-dimensional space.

3 Direct Estimation of Frenet-based Mean Shapes by Preprocessing

In Chapter 5, we introduce two Riemannian geometries under specific shape representations of a Euclidean curve $x \in \mathbb{R}^d$ based on its Frenet curvatures, each with distinct strengths and weaknesses. These mathematical representations are the unparametrized Frenet curvatures $\mathcal{R}_{\boldsymbol{\theta}}$ (5.13) and the square-root curvatures transform $\mathcal{R}_{\text{SRC}}$ (5.26). We demonstrate that each induces a specific distance on the shape space through the geodesic distance on the corresponding Riemannian manifold. Therefore, we can theoretically define the two corresponding Karcher mean shapes of the population of curves $\mathcal{E}$. In practice, a direct method for computing these Karcher means, given the statistical challenge of estimating Frenet curvatures, is to consider the statistical estimation of the individual Frenet curvatures of each curve in an efficient pre-processing step.

3.1 Karcher Mean in Frenet Curvatures Space

Suppose that for all $k = 1, \ldots, N$, $x_k \in F([0,1], \mathbb{R}^d)$ is a Frenet curve. Given the distance $d_{\mathcal{S}}^{(\boldsymbol{\theta})}$, defined in equation (5.19), the called Frenet curvatures mean shape $\mu^{(\boldsymbol{\theta})}$ is defined as

$$\mu^{(\boldsymbol{\theta})} = \arg\min_{\mu} \sum_{k=1}^{N} d_{\mathcal{S}}^{(\boldsymbol{\theta})}(x_k, \mu)^2. \tag{6.6}$$

Again, this comes down to computing the Karcher mean in the Riemannian manifold of Frenet curvatures $\mathcal{H}$. Let $\boldsymbol{\theta}_k \in \mathcal{H}$ be the Frenet curvatures of $x_k \in \mathcal{E}$, supposed here to be known. Then, we search for the optimal $\boldsymbol{\theta}_\mu^*$ that solve

$$\boldsymbol{\theta}_\mu^* = \arg\min_{\boldsymbol{\theta}_\mu} \sum_{k=1}^{N} \int_0^1 \|\boldsymbol{\theta}_\mu(s) - \boldsymbol{\theta}_k(s)\|_2^2 ds, \tag{6.7}$$

and set $\mu^{(\boldsymbol{\theta})} = \mathcal{R}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{\theta}_\mu^*)$, with any parameterization as we are only interested in defining the mean shape. The numerical computation of this mean does not require any alignment step and can be done directly with Algorithm 3 given the exponential map and its inverse: let $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{H}, v \in T_{\boldsymbol{\theta}_1}(\mathcal{H})$, we have $\exp_{\boldsymbol{\theta}_1}(v) = \boldsymbol{\theta}_1 + v$ and $\exp_{\boldsymbol{\theta}_1}^{-1}(\boldsymbol{\theta}_2) = \boldsymbol{\theta}_2 - \boldsymbol{\theta}_1$.

3.2 Karcher Mean in Square Root Curvatures Space

Suppose that for all $k = 1, \ldots, N$, $x_k \in F([0,1], \mathbb{R}^d)$ is a Frenet curve. Given the distance $d_{\mathcal{S}}^{(\mathrm{SRC})}$, defined in equation (5.28), the SRC mean shape $\mu^{(\mathrm{SRC})}$ is defined as

$$\mu^{(\mathrm{SRC})} = \arg\min_{\mu} \sum_{k=1}^{N} d_{\mathcal{S}}^{(\mathrm{SRC})}(x_k, \mu)^2. \tag{6.8}$$

In this case, as we are only interested in defining the mean shape function, we can use the reformulation principle of Brunel and Park (2019) (5.31) and directly solve the following problem

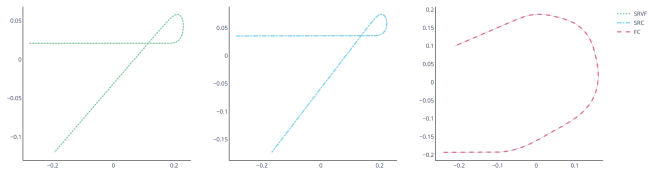
$$\boldsymbol{\theta}_\mu^* = \arg\min_{\boldsymbol{\theta}_\mu \in \mathcal{H}} \sum_{k=1}^{N} \inf_{\gamma_k \in \mathrm{Diff}_+([0,1])} \int_0^1 \left\| \frac{\boldsymbol{\theta}_\mu(s)}{\sqrt{\|\boldsymbol{\theta}_\mu(s)\|}} - \sqrt{\gamma_k'(s)} \frac{\boldsymbol{\theta}_k(\gamma_k(s))}{\sqrt{\|\boldsymbol{\theta}_k(\gamma_k(s))\|}} \right\|_2^2 + \|1 - \gamma_k'(s)\|_2^2 ds\,, \tag{6.9}$$

where $\boldsymbol{\theta}_k$ are the Frenet curvatures of $x_k \in \mathcal{E}$, supposed to be known. Then the mean $\mu^{(\mathrm{SRC})}$ is reconstruct by solving the Frenet-Serret ODE with the parameter $\boldsymbol{\theta}_\mu^*$. In practice, the numerical computation of $\boldsymbol{\theta}_\mu^*$ can also be done with Algorithm 3. An additional step for computing the optimal warping function $\gamma_k \in \mathrm{Diff}_+([0,1])$ solutions of the registration problems in (6.9) must be done before step 1 of the algorithm. Then the tangent vectors are computed as $v_k = \sqrt{\gamma_k'(s)} \frac{\boldsymbol{\theta}_k(\gamma_k(s))}{\sqrt{\|\boldsymbol{\theta}_k(\gamma_k(s))\|}} - \frac{\boldsymbol{\theta}_\mu(s)}{\sqrt{\|\boldsymbol{\theta}_\mu(s)\|}}$.

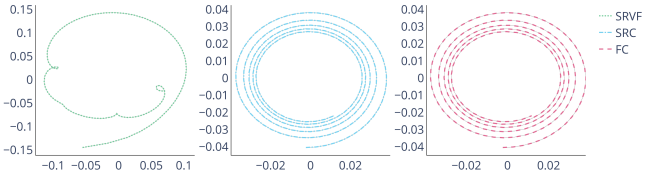
These two Karcher means (Frenet curvatures and SRC) and the SRVF one reflect the strengths and weaknesses of each of the corresponding Riemannian geometries. Indeed, computing these means on the synthetic data examples from Chapter 5 used to compute the pairwise distance matrices, where the exact Frenet curvatures functions of the curves are known, effectively summarizes the earlier analysis, as illustrated in Figure 6.1. Only the SRC mean shape is consistent, in all cases, with the shapes of the considered curves.

3.3 Preprocessing of Individual Frenet Curvatures

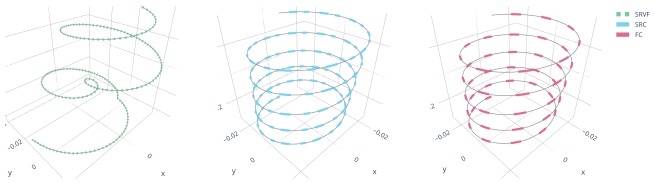
As mentioned earlier, in a typical application scenario, the Frenet curvatures $\boldsymbol{\theta}_k$ are not known, and the corresponding curves, $x_k \in \mathcal{E}$, are only observed discretely and with noise. Therefore, the previously defined Karcher means associated with the FC and SRC representations cannot be directly employed. In this section, the proposed method to approximate these Karcher means (FC and SRC) while accounting for observation noise involves first estimating each individual vector-valued function of Frenet curvatures in a preprocessing step. This preprocessing step must be particularly effective since the estimation of Frenet curvatures is highly sensitive to observation noise.



(a) Karcher means of loops considered in Figure 5.5



(b) Karcher means of spirals considered in Figure 5.8



(c) Karcher means of helices considered in Figure 5.11

Figure 6.1: Karcher means under SRVF representation (**green**), SRC representation (**blue**) and Frenet Curvatures representation (**pink**) on the three cases of synthetic data considered in Chapter 5.

Various methods are proposed in the literature to estimate Frenet curvatures for a single curve. The most common method relies on a smooth estimation of the first d derivatives of the Euclidean curve, followed by the application of the extrinsic formulas of Frenet curvatures from the derivatives (Kim et al., 2013; Sangalli et al., 2009; Lewiner et al., 2005). Park and Brunel (2019) propose more efficient and robust alternatives based on smooth estimation of the Frenet path, followed by minimizing a least squares criterion adapted to the geometry of the problem and the Lie group $SO(d)$ in which the variables lie. In this thesis, these methods are thoroughly detailed in Chapter 3. Additionally, we propose in Chapter 4 a promising alternative method to estimate latent variables in the Frenet framework directly from observations of the Euclidean curve using an Expectation-Maximization algorithm in the Lie group $SE(3)$. We do not delve into these various methods again in this chapter. Moreover, this preprocessing method involves estimating the Frenet curvatures of each curve individually and independently of each other. Consequently, their means will reflect the strengths and weaknesses of each of these estimation methods highlighted in the simulation studies of Part I.

The scope of this chapter is not to re-compare these methods through mean calculations but to introduce different global approaches to approximate Karcher means defined from Frenet curvatures. Thus, in this chapter, to compare this method based on the preprocessing of individual Frenet curvatures with the methods introduced in the following section, we consider the estimation method introduced by Park and Brunel (2019) (Algorithm 1 of Chapter 3) and summarized here. First, the d first derivatives of the curve $x_k \in \mathcal{E}$ are estimated using a local polynomial regression, and then an estimator of the Frenet path $\hat{Q}_k$ is computed with the Gram-Schmidt orthonormalization process. Given this estimator, the estimator of the Frenet curvatures $\hat{\boldsymbol{\theta}}_k$ is done by solving the following minimization problem,

$$\hat{\boldsymbol{\theta}}_{k;h,\lambda} = \arg\min_{\boldsymbol{\theta}} \mathcal{J}_{h,\lambda}(\boldsymbol{\theta}|\hat{Q}_k), \tag{6.10}$$

where

$$\mathcal{J}_{h,\lambda}(\boldsymbol{\theta}|\hat{Q}_k) = \int_0^1 \int_0^1 K_h(t-s) \left\| \log\left(\hat{Q}_k(t)^T \hat{Q}_k(s) \exp\left((t-s)A_{\boldsymbol{\theta}}\left(\frac{t+s}{2}\right)\right)\right)\right\|_F^2 dtds + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt\,, \tag{6.11}$$

$K(\cdot)$ is a kernel function with compact support, e.g. $K(u) = \frac{3}{4}(1-u)^2 \mathbb{1}_{[-1,1]}(u)$ and $K_h(u) = (1/h)K(u/h)$. The kernel $K(\cdot)$ and the bandwidth h define a prediction horizon for the flow. The regularization parameters λ is a hyperparameter that controls the amounts of smoothing of $\boldsymbol{\theta}$. This estimation method

is demonstrated in Part I, in comparison with others, to be sufficient for obtaining good estimators in cases where there is not too much noise or the geometry of the curve is not too complex. In other cases, it may tend to provide a somewhat overly smoothed estimator.

Finally, by applying this Frenet curvatures estimation method on each curve in $\mathcal{E}$ individually, the FC and SRC Karcher means can be approximated by using the formulas (6.6) and (6.8) given the estimators $\left\{\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_N\right\}$. In the following, we will denote these approximated mean shapes as $\hat{\mu}^{(\boldsymbol{\theta})}$ and $\hat{\mu}^{(\mathrm{SRC})}$.

4 Estimation of Frenet-based Mean Shapes in a Unified Functional Data Modelling Framework

Unlike the classical framework of shape analysis presented in Section 2, the initial objective of Park and Brunel (2019) was not on defining a proper metric on the shape space to define a Karcher mean. They were interested in developing a statistical characterization of a mean shape that enables the identification of the mean geometry and mean Frenet curvatures. Therefore, in their initial formulation, they do not assume the existence of a generative model for the mean shape or the mean parameter in relation to individual Frenet paths or curvatures but directly exploit the characteristic features of the Frenet paths as the solution of the ODEs, and consider the ODE as a model constraint.

However, we propose here to make the link between their definition and the shape analysis framework. By bridging this gap, we demonstrate that, through their formulation of the mean Frenet curvatures parameter optimization criterion, Park and Brunel (2019) attempt to approximate the Karcher mean in the shape space defined by Frenet curvatures representations (5.13), employing a regularization approach (Section 4.2). In fact, they provide an alternative approach to estimating Frenet-based Karcher means by directly modeling noise through the characterization of the mean parameter within a functional data modeling framework. Building upon these observations, we leverage their idea and formulation by proposing new methods to approximate Frenet-based mean shape definitions that exhibit more "elastic" properties (Section 4.3 and Section 4.4). In particular, we introduce a similar regularization approach to estimate the Karcher mean in the space of SRCs from noisy observations where curvatures parameters are not known (Section 4.4).

4.1 Statistical Characterization of Mean Frenet Curvatures via the Mean Frenet Vector Field

This section presents the statistical characterization and definition of the mean Frenet curvatures parameter proposed by Park and Brunel (2019).

4.1.1 Problem Formulation

Recall that any curve $x_k \in \mathcal{E}$ can be expressed as $x_k(t) = X_k(s_k(t))$, where $s_k(t) = \int_0^t \|\dot{x}_k(u)\|\, du$ is the corresponding arc-lenght function. The shape function X_k is associated with a Frenet-Serret ODE parametrized with respect to the functional parameter $\boldsymbol{\theta}_k$ corresponding to the Frenet curvatures:

$$X_k(s) = X_{k0} + Q_{k0} \int_0^s T_{\boldsymbol{\theta}_k}(u)\, du\,, \quad Q_k'(s) = Q_k(s) A_{\boldsymbol{\theta}_k}(s)\,.$$

Consequently, one can identify the shapes with the Frenet paths $\boldsymbol{Q} = \{Q_1, \ldots, Q_N\}$, or equivalently with the set of Frenet curvatures $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$. The aim is then to derive a mean parameter $\bar{\boldsymbol{\theta}}$ (and mean Frenet path $\bar{Q}$) for $\mathcal{E}$ as a measure of centrality that corresponds to the mean shape defined as

$$\bar{X}(s) = \bar{X}_0 + \bar{Q}_0 \int_0^s T_{\bar{\boldsymbol{\theta}}}(u)\, du\,, \quad \bar{Q}'(s) = \bar{Q}(s) A_{\bar{\boldsymbol{\theta}}}(s)\,,$$

which is independent of the variations in translations, rotations, and scalings. This parametrization of curves shows that the quotient space of arclength parametrized curves (under the group action of Euclidean motions) can be identified as the space of Frenet paths $\mathcal{F}_0$ (defined in (2.47)). Hence, working with a population of Frenet paths can be sufficient to define a mean shape.

A fundamental concept in solving ordinary differential equations lies in the concept of the flow over time t, which was introduced in Section 3.2 of Chapter 3. Here, we are dealing with a Frenet-Serret ODE that is parameterized by Frenet curvatures $\boldsymbol{\theta}_k$. The corresponding flow of this ODE, denoted as $\varphi_{\boldsymbol{\theta}_k}(t, \cdot)$, is defined as $\varphi_{\boldsymbol{\theta}_k}(t, Q_0) = Q_k(t)$, where $Q_k(0) = Q_0 \in SO(d)$. As the mapping $t \mapsto Q_k(t)$ represents the Frenet path solution of the Frenet-Serret ODE, its dependency on the parameter $\boldsymbol{\theta}_k$ inherently extends to the flow of the ODE as well. Therefore, one can see that the geometrical features $\boldsymbol{\theta}_k$ define the vector field $Q \mapsto QA_{\boldsymbol{\theta}_k}(s)$, and that the observable features such as Frenet paths are in fact the corresponding flows $\varphi_{\boldsymbol{\theta}_k}$.

4.1.2 Mean Frenet Curvatures Definition

Taking these observations into account, Park and Brunel (2019) propose a method for defining the mean shape through the mean Frenet curvatures parameter, which involves averaging the vector fields $Q \mapsto QA_{\boldsymbol{\theta}_k}(s)$. They introduce the notion of a mean vector field, defined on $SO(d)$, such that its solution paths are close to the individual Frenet paths Q_k, $k = 1, \ldots, N$. In other words, the mean vector field corresponds to the flow that provides *a best* approximation to all the individual flows.

As already introduced in Section 5 of Chapter 3, the group property of the flow, which can be interpreted as the *self-prediction property* (3.17), defines the following characterization between the solution of the ODE and the corresponding flow: the curve $s \mapsto Q(s)$ is a solution to $Q' = QA_{\boldsymbol{\theta}}$ if and only if

$$\int_0^1 \int_0^1 d\left(Q(t), \varphi_{\boldsymbol{\theta}}\left(t - s, s, Q(s)\right)\right)^2 dsdt = 0, \tag{6.12}$$

where $d(\cdot, \cdot)$ is a distance defined on $SO(d)$. Therefore, as we want to replace $\boldsymbol{\theta}_k$ by a common $\bar{\boldsymbol{\theta}}$, we require that the mean flow $\varphi_{\bar{\boldsymbol{\theta}}}$ should minimize the self-prediction errors for all the trajectories simultaneously. The individual error is measured by

$$\mathcal{V}(Q_k, \varphi_{\boldsymbol{\theta}}) = \int_0^1 \int_0^1 d\left(Q_k(t), \varphi_{\boldsymbol{\theta}}\left(t - s, s, Q_k(s)\right)\right)^2 dsdt\,. \tag{6.13}$$

Definition 6.2. *Let $Q_k \in SO(p), k = 1, \ldots, N$ be the independent and identically distributed random Frenet paths with the same distribution as Q, associated with parameters $\boldsymbol{\theta}_k \in \mathcal{H}$ satisfying $Q_k' = Q_k A_{\boldsymbol{\theta}_k}$. The mean parameter for the Frenet path Q is defined as*

$$\bar{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \mathcal{H}} \mathbb{E}\{\mathcal{V}(Q, \varphi_{\boldsymbol{\theta}})\}\,.$$

A noticeable feature of their formulation through the ODE flow is that they do not use the infinitesimal characterization of the differential equation based on the derivative.

The essential ingredients of this mean parameter definition based on the self-prediction criterion (6.13) are the distance function d and the representation of the flow $\varphi_{\boldsymbol{\theta}}$. The choice of these needs to be adapted to the underlying sample space. The Frenet differential equation is defined on the Lie group $SO(d)$. Therefore, ensuring the orthogonality constraint requires special treatment in developing a numerical algorithm to solve an ODE and also in tackling a parameter estimation problem in ODE, as numerical errors can accumulate and induce an uncontrolled bias. The extension of the theory of ODEs from Euclidean space to Lie groups or manifolds is well developed (Hairer et al., 2006). In particular,

the rotation group $SO(d)$ is a Lie Group that is also a differentiable manifold, with many remarkable properties that are essential in tackling the numerical problems (Absil et al., 2010). These backgrounds are introduced and detailed in Section 3 of Chapter 3.

4.2 Mean Parameter Estimation using the Geodesic Distance on $SO(d)$-valued Curves

Suppose that we have a sample of Frenet paths $\{Q_1, \ldots, Q_N\}$ with the corresponding parameters $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$. Park and Brunel (2019) develop an empirical criterion to construct an estimator of the mean parameter $\bar{\boldsymbol{\theta}}$ from the previous definition, in the spirit of a nonparametric function estimation problem. Their criterion depends on their specific choice for the distance on the space of $SO(d)$-valued curve in the self-prediction criterion (6.13), that is, the $SO(d)$ geodesic distance on the unparametrized Frenet paths. But also, it depends on the approximation of the flow they considered, which is based on the Magnus expansion. We detail the construction of their criterion in this section and demonstrate how it can be approximated as a weighted and regularized optimization criterion for the Karcher mean in the space of unparameterized Frenet curvatures $\mathcal{H}$ (6.6).

4.2.1 Estimation Criterion

From the Riemannian metric on $SO(d)$ (3.11), the geodesic distance in the space of measurable functions from $[0,1] \to SO(d)$ is defined as follows. Given $P_1, P_2 \in \mathbb{L}^2([0,1], SO(d))$,

$$d_1(P_1, P_2)^2 = \int_0^1 \| \log(P_1(t)^T P_2(t)) \|_F^2 dt. \tag{6.14}$$

In addition, as detailed in Section 3 of Chapter 3, the flow can be approximated from Theorem 3.1 as, for all $t, s \in [0,1]$ such that $|t-s| \leq 1$,

$$\varphi_{\boldsymbol{\theta}}(t-s, s, Q_k(s)) = Q_k(s) \exp(\Omega(t-s, s; \boldsymbol{\theta}))\,, \tag{6.15}$$

where Ω is the Magnus expansion (3.20). By combining this choice of distance with the flow approximation, the criterion (6.13) can be expressed as, for $k = 1, \ldots, N$

$$\mathcal{V}_1(Q_k, \varphi_{\boldsymbol{\theta}}) = \int_0^1 d_1(Q_k, \varphi_{\boldsymbol{\theta}}(\cdot, s, Q_k(s)))^2 ds \tag{6.16}$$

$$= \int_0^1 \int_0^1 \left\| \log\left(Q_k(t)^T Q_k(s) \exp\left(\Omega(t-s, s; \boldsymbol{\theta})\right)\right)\right\|_F^2 dt ds\,. \tag{6.17}$$

To allow for varying degrees of precision in the self-prediction, we incorporate weights according to the distance to initial values s in evaluating the solution at t and define a weighted criterion:

$$\breve{\ell}_{N,h}^{(1)}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=1}^{N} \int_0^1 \int_0^1 K_h(t-s) \| \log\left(Q_k(t)^T Q_k(s) \exp\left(\Omega(t-s, s; \boldsymbol{\theta})\right)\right) \|_F^2 \, ds dt\,, \tag{6.18}$$

where $K(\cdot)$ is a kernel function with compact support, e.g. $K(u) = \frac{3}{4}(1-u)^2 1_{[-1,1]}(u)$. The scaled kernel is denoted by $K_h(u) = (1/h)K(u/h)$. The kernel $K(\cdot)$ and the bandwidth h define a prediction horizon for the flow. In addition, we introduce a smooth regularization for functional parameter $\boldsymbol{\theta}$ with a penalty term

$$\mathcal{P}_\lambda(\boldsymbol{\theta}) = \lambda \|\boldsymbol{\theta}''\|_2^2 = \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt\,, \tag{6.19}$$

and define the empirical criterion as $\breve{\mathcal{I}}_{h,\lambda}^{(1)}(\boldsymbol{\theta}) = \breve{\ell}_h^{(1)}(\boldsymbol{\theta}) + \mathcal{P}_\lambda(\boldsymbol{\theta})$.

Definition 6.3. *Let $\{Q_1, \dots, Q_N\}$ be a sample of Frenet paths with parameters of Frenet curvatures $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N$. For a fixed h and λ, the sample mean vector field (or Frenet curvatures) is defined as the parameter $\breve{\boldsymbol{\theta}}^{(1)}_{h,\lambda}$ which minimizes the global prediction error $\breve{\mathcal{I}}^{(1)}_{h,\lambda}(\boldsymbol{\theta})$:*

$$\breve{\boldsymbol{\theta}}^{(1)}_{h,\lambda} \in \arg\min_{\boldsymbol{\theta}\in\mathcal{H}} \breve{\mathcal{I}}^{(1)}_{h,\lambda}(\boldsymbol{\theta})\,. \tag{6.20}$$

This definition can be viewed as a generalization of the mean in the scale-space view in nonparametric curve estimation (Chaudhuri and Marron, 2000; Wei and Panaretos, 2018).

Remark 6.1. *If the set of curves $\mathcal{E}$ is composed of a single curve x_0, the empirical criterion $\breve{\mathcal{I}}^{(1)}_{h,\lambda}(\boldsymbol{\theta})$ defined here to find the mean parameter corresponds to the one defined in Section 5 of Chapter 3 for the estimation of the Frenet curvatures $\boldsymbol{\theta}_0$ associated with the curve x_0. Therefore, in this case, the mean Frenet curvatures parameter is simply defined as the Frenet curvatures of the only curve composing $\mathcal{E}$. Note that it corresponds to the estimation criterion of the Frenet curvatures of a single curve used in Section 3 in the preprocessing step.*

From the definition of the mean Frenet curvatures $\breve{\boldsymbol{\theta}}^{(1)}_{h,\lambda}$ we can also define the mean Frenet path $\breve{Q}^{(1)}_{h,\lambda}(t) = \exp(\Omega(t, 0, \breve{\theta}^{(1)}_{h,\lambda}))$ and the corresponding mean shape $\breve{X}^{(1)}_{h,\lambda}$ obtained by integrating the gradient. However, it is rather difficult to compute this corresponding mean or to analyze it. Since the expression of Ω is generally intractable, we further derive a consistent approximation to the flow by truncating the Magnus expansion, see Hairer et al. (2006). In particular, we use the approximation of order 2 introduced in equation (3.21), which used a simple quadrature rule with the midpoint and a truncation after the first term of the Magnus expansion: $Q_{s+h} = Q_s \exp\left(hA_{\boldsymbol{\theta}}\left(s + \frac{h}{2}\right)\right)$, i.e. $\varphi_{\boldsymbol{\theta}}(h, s, Q_s) - Q_{s+h} = O(h^2)$. Using this approximation of the flow, we introduce an approximation, $\mathcal{I}^{(1)}_{h,\lambda}(\boldsymbol{\theta}) = \ell^{(1)}_{N,h}(\boldsymbol{\theta}) + \mathcal{P}_\lambda(\boldsymbol{\theta})$, to the criterion $\breve{\mathcal{I}}^{(1)}_{h,\lambda}$ and $\breve{\boldsymbol{\theta}}^{(1)}_{h,\lambda}$ valid for small h where

$$\ell^{(1)}_{N,h}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{k=1}^{N}\int_0^1\int_0^1 K_h(t-s)\left\|\log\left(Q_k(t)^T Q_k(s)\exp\left((t-s)A_{\boldsymbol{\theta}}\left(\frac{s+t}{2}\right)\right)\right)\right\|_F^2 dsdt\,.$$

The following proposition shows that, at first approximation, this approach is tractable and can be easily understood in terms of the geometry of the curves.

Proposition 6.1. *Let $Q_1, \dots, Q_N$ be Frenet paths with parameters $\boldsymbol{\theta}_k$, $k = 1, \dots, N$ in $\mathcal{H}$, satisfying $\|\boldsymbol{\theta}_k\|_2^2 \leq \frac{\pi}{2}$. Then, there exists $B > 0$, such that for all $\|\boldsymbol{\theta}\|_2 \leq B$,*

$$\breve{\mathcal{I}}^{(1)}_{h,\lambda}(\boldsymbol{\theta}) - \mathcal{I}^{(1)}_{h,\lambda}(\boldsymbol{\theta}) = O(h^3).$$

Proof. Because the Q_ks are Frenet paths, the main point is to provide a tractable approximation for the geodesic distance $\|\log\left(\exp\left(-\Omega(t-s, s; \boldsymbol{\theta}_k)\right)\exp\left(\Omega(t-s, s; \boldsymbol{\theta})\right)\right)\|_F^2$. If $\|\boldsymbol{\theta}_k\|_2^2 \leq \frac{\pi}{2}$ and $\|\boldsymbol{\theta}\|_2^2 \leq \frac{\pi}{2}$, we can use the Magnus expansion of Ω. The objective is to show that Ω can be replaced by the first terms of the Magnus expansion for small $t - s$. Indeed, for all s, we have $\Omega_{\boldsymbol{\theta}}(s+h, s) - \Omega^{[m]}_{\boldsymbol{\theta}}(s+h, s) = O(h^{2m})$ where $\Omega^{[m]}_{\boldsymbol{\theta}}$ is the truncation at level m:

$$\begin{aligned}\Delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k) &= \log\left(\exp\left(-\Omega(t-s, s; \boldsymbol{\theta}_k)\right)\exp\left(\Omega(t-s, s; \boldsymbol{\theta})\right)\right)\\ &= \log\left(\exp\left(-\Omega^{[m]}(t-s, s; \boldsymbol{\theta}_k) + R^{[m]}_k\right)\exp\left(\Omega^{[m]}(t-s, s; \boldsymbol{\theta}) + R^{[m]}\right)\right)\end{aligned}$$

For small $t - s$, the Baker-Campbell-Hausdorff, defined in Theorem 2.2, states that

$$\log\left(\exp\left((t-s)B(s)\right)\exp\left((t-s)C(s)\right)\right) = \sum_{n\geq 1}(t-s)^n z_n\left(B(s), C(s)\right)$$

with $z_n\left(B(s), C(s)\right)$ a homogeneous Lie polynomial of order m. The main term is $z_1 = B(s) + C(s)$ and $z_2 = \frac{1}{2}\left[B(s), C(s)\right]$. This means that

$$\begin{aligned}
\Delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k) =& \sum_{n\geq 1}(t-s)^n z_n\left(-\frac{1}{t-s}\Omega(t-s,s;\boldsymbol{\theta}_k), \frac{1}{t-s}\Omega(t-s,s;\boldsymbol{\theta})\right) \\
=&(t-s)\left(\frac{1}{t-s}\Omega(t-s,s;\boldsymbol{\theta}) - \frac{1}{t-s}\Omega(t-s,s;\boldsymbol{\theta}_k)\right) \\
&+ (t-s)^2\left[\frac{1}{t-s}\Omega(t-s,s;\boldsymbol{\theta}), \frac{1}{t-s}\Omega(t-s,s;\boldsymbol{\theta}_k)\right] \\
&+ (t-s)^3\sum_{n\geq 0}(t-s)^n z_{n+3}\left(\Omega(t-s,s;\boldsymbol{\theta}_k), \Omega(t-s,s;\boldsymbol{\theta})\right) \\
=&(t-s)\left(\frac{1}{t-s}\Omega(t-s,s;\boldsymbol{\theta}) - \frac{1}{t-s}\Omega(t-s,s;\boldsymbol{\theta}_k)\right) + (t-s)^2 G(\boldsymbol{\theta}, \boldsymbol{\theta}_k)
\end{aligned}$$

where G is a smooth bounded function (because $\sum_{n\geq 0}(t-s)^n z_{n+3}\left(\Omega(t-s,s;\boldsymbol{\theta}_k), \Omega(t-s,s;\boldsymbol{\theta})\right)$ is also a smooth and bounded function). For order $m = 1$, for any $\boldsymbol{\theta}$, we have $\frac{1}{t-s}\Omega^{[1]}(t-s,s;\boldsymbol{\theta}) = \frac{1}{t-s}\int_s^t A_{\boldsymbol{\theta}}(u)du$. From Theorem 4.2 in Iserles et al. (2000), truncating at order 1 provides an approximation of order 2, meaning that

$$\frac{1}{t-s}\left\{\Omega(t-s,s;\boldsymbol{\theta}) - \int_s^t A_{\boldsymbol{\theta}}(u)du\right\} = C(\boldsymbol{\theta})\,(t-s) \tag{6.21}$$

(with $C(\cdot)$ bounded function). By using the approximation in (6.21), we obtain a first-order approximation, i.e.

$$\begin{aligned}
\Delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k) &= (t-s)\left(\frac{1}{t-s}\int_s^t A_{\boldsymbol{\theta}}(u)du - \frac{1}{t-s}\int_s^t A_{\boldsymbol{\theta}_k}(u)du\right) + C(\boldsymbol{\theta})\,(t-s)^2 + (t-s)^2 G(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \\
&= (t-s)\left(\frac{1}{t-s}\int_s^t A_{\boldsymbol{\theta}}(u)du - \frac{1}{t-s}\int_s^t A_{\boldsymbol{\theta}_k}(u)du\right) + C_1(\boldsymbol{\theta}, \boldsymbol{\theta}_k)\,(t-s)^2
\end{aligned}$$

We propose to replace the integrand with a simple quadrature. This is the computational basis for the geometric integration of ODE in Lie groups (see Iserles et al. (2000)). This approximation is derived from the simple midpoint quadrature rule for integrating

$$\frac{1}{t-s}\int_s^t A_{\boldsymbol{\theta}}(u)du = A_{\boldsymbol{\theta}}\left(s + \frac{t-s}{2}\right) + c\|\boldsymbol{\theta}'\|_\infty \times (t-s)$$

Our final approximation is then

$$\Delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = (t-s)\left(A_{\boldsymbol{\theta}}\left(s + \frac{t-s}{2}\right) - A_{\boldsymbol{\theta}_k}\left(s + \frac{t-s}{2}\right)\right) + (t-s)^2\left\{C_1(\boldsymbol{\theta}, \boldsymbol{\theta}_k) + c\|\boldsymbol{\theta}'\|_\infty\right\}.$$

The same approximation shows that

$$\begin{aligned}
\left|\bar{\mathcal{I}}^{(1)}_{h,\lambda}(\boldsymbol{\theta}) - \mathcal{I}^{(1)}_{h,\lambda}(\boldsymbol{\theta})\right| &\leq \iint K_h(t-s)\|\Delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k)\|^2 dtds \\
&\leq \sum_{k=1}^N C_2(\boldsymbol{\theta}, \boldsymbol{\theta}_k)\int_0^1\int_0^1 K_h(t-s)|t-s|^3 dtds \\
&\leq \sigma_K^3 h^3 \sum_{k=1}^N C_2(\boldsymbol{\theta}, \boldsymbol{\theta}_k)
\end{aligned}$$

When $h \to 0$, the convergence of $\boldsymbol{\theta} \mapsto \breve{\mathcal{I}}^{(1)}_{h,\lambda}(\boldsymbol{\theta})$ to $\boldsymbol{\theta} \mapsto \mathcal{I}^{(1)}_{h,\lambda}(\boldsymbol{\theta})$ is then uniform on a bounded set in $\mathcal{H}$. ■

4.2.2 Limitations

The criterion considered by Park and Brunel (2019) for the estimation of the mean parameter $\bar{\boldsymbol{\theta}}$ is the approximated criterion $\mathcal{I}^{(1)}_{h,\lambda}(\boldsymbol{\theta}) = \ell^{(1)}_{N,h}(\boldsymbol{\theta}) + \mathcal{P}_\lambda(\boldsymbol{\theta})$. In the proof of the Proposition 6.1, we have obtained the simpler approximation of order 2 of the expression $\Delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \log\left(Q_k(t)^T Q_k(s) \exp\left((t-s) A_{\boldsymbol{\theta}}\left(\frac{s+t}{2}\right)\right)\right)$, that is,

$$\Delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = (t-s)\left(A_{\boldsymbol{\theta}}\left(\frac{t+s}{2}\right) - A_{\boldsymbol{\theta}_k}\left(\frac{t+s}{2}\right)\right) + (t-s)^2 C_2(\boldsymbol{\theta}, \boldsymbol{\theta}_k). \tag{6.22}$$

Therefore the criterion $\ell^{(1)}_{N,h}(\boldsymbol{\theta})$ can be approximated, using Proposition 5.3, as

$$\begin{aligned}
\ell^{(1)}_{N,h}(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{k=1}^{N} \int_0^1 \int_0^1 K_h(t-s) \left\| (t-s)\left(A_{\boldsymbol{\theta}}\left(\frac{t+s}{2}\right) - A_{\boldsymbol{\theta}_k}\left(\frac{t+s}{2}\right)\right) + (t-s)^2 C_2(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \right\|_F^2 dsdt \\
&= \frac{1}{N}\sum_{k=1}^{N} \int_0^1 \int_0^1 K_h(t-s)(t-s)^2 \left\| \left(A_{\boldsymbol{\theta}}\left(\frac{t+s}{2}\right) - A_{\boldsymbol{\theta}_k}\left(\frac{t+s}{2}\right)\right) \right\|_F^2 dsdt + O\left(h^3\right) \\
&= \frac{1}{N}\sum_{k=1}^{N} \int_0^1 \int_0^1 K_h(t-s)(t-s)^2 \left\| \boldsymbol{\theta}\left(\frac{t+s}{2}\right) - \boldsymbol{\theta}_k\left(\frac{t+s}{2}\right) \right\|_2^2 dsdt + O\left(h^3\right)
\end{aligned}$$

Then, the used criterion $\mathcal{I}^{(1)}_{h,\lambda}(\boldsymbol{\theta})$ is approximated for small h as

$$\mathcal{I}^{(1)}_{h,\lambda}(\boldsymbol{\theta}) = \frac{\sigma_K^2 h^2}{N}\sum_{k=1}^{N} \|\boldsymbol{\theta} - \boldsymbol{\theta}_k\|^2_{\mathbb{L}^2} + O(h^3) + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt. \tag{6.23}$$

Interestingly, this shows that the mean parameter defined through the mean of vector fields and obtained by minimization of $\mathcal{I}^{(1)}_{h,\lambda}(\boldsymbol{\theta})$, can be approximated for small h by the empirical mean of the individual Frenet curvatures. However, the criterion $\mathcal{I}^{(1)}_{h,\lambda}(\boldsymbol{\theta})$ is defined based on Frenet paths parameterized by arc-length. The approximation derived in equation (6.23) can then be interpreted as a weighted and penalized criterion for the computation of the Karcher mean in the space of unparameterized Frenet curvatures $\mathcal{H}$ (6.6). Consequently, with this approximation, it becomes evident that the estimated mean shape proposed in Park and Brunel (2019) shares the same drawbacks of lack of elasticity highlighted with the unparametrized Frenet curvatures Riemannian geometry (Figure 6.1), as it does not account for parametrization in its original formulation. Indeed, in their framework, Frenet paths are considered parametrized by arc length and, therefore, already independent of any time-reparametrization of the Euclidean curves. However, the modeling within a functional data framework they propose is particularly interesting from a statistical perspective because it introduces an alternative approach to approximating Karcher means by defining a unique criterion for estimating the mean parameter from Frenet paths observations, thus modeling the overall noise of the data in a unified manner.

4.3 Mean Parameter Estimation using an "Elastic" Geodesic Distance on $SO(d)$-valued Curves

To obtain an estimation criterion for the mean parameter with more “elastic” properties (more similar to frameworks defined by the SRVF or the SRC), we propose in this section a first extension of their approach by changing the distance on $SO(d)$-valued curves considered in the derivation of the self-prediction criterion (6.13).

4.3.1 Choice of the Distance

As highlighted in Chapter 5 and suggested by Srivastava and Klassen (2016), it is essential to adopt a curve representation that is dependent on the parameterization to establish a more *elastic* metric or criterion, facilitated by the registration problem that parameterization necessitates. For time-parametrized curves x_0 and $x_1 \in \mathbb{L}^2([0,1],\mathbb{R}^d)$, the registration problem is motivated by finding the most appropriate warping function $h \in \mathrm{Diff}_+([0,1])$ such that the two curves $x_1(h(t))$ and $x_0(t)$ look similar. Then, we have

$$x_0(t) = X_0(s_0(t))\,, \quad x_1(h(t)) = X_1(s_1(h(t)))\,,$$

and the corresponding time-parametrized Frenet paths are $Q_0(s_0(t))$ and $Q_1(s_1(h(t)))$. In order to use the idea and framework developed by Park and Brunel (2019) and detailed in the previous section, it is natural to express the registration problem between x_0 and x_1 in terms of the Frenet paths. As their framework is developed with arc-length parametrized Frenet paths, we consider the reformulation principle (Brunel and Park, 2019) $s_1 \circ h = \gamma \circ s_0$, where $\gamma \in \mathrm{Diff}_+([0,1])$. With this reformulation, the registration problem now consists of finding the optimal $\gamma \in \mathrm{Diff}_+([0,1])$ such that the two Frenet paths $Q_0(s_0(t))$ and $Q_1(\gamma(s_0(t)))$ look similar. Now, note that the Frenet path $s \mapsto \tilde{Q}_1(s) = Q_1(\gamma(s))$ is a solution of the following Frenet-Serret ODE (Brunel and Park, 2019):

$$\begin{aligned} \tilde{Q}_1'(s) &= Q_1'(\gamma(s))\gamma'(s) \\ &= Q_1\left(\gamma(s)\right) A_{\boldsymbol{\theta}_1}(\gamma(s))\gamma'(s) \\ &= \tilde{Q}_1(s) A_{\tilde{\boldsymbol{\theta}}_1}(s)\,, \end{aligned} \tag{6.24}$$

and so we have

$$\tilde{\boldsymbol{\theta}}_1(s) = \boldsymbol{\theta}_1(\gamma(s))\gamma'(s)\,. \tag{6.25}$$

Therefore, the flow of this ODE also satisfies the self-prediction property (3.17): for all $u, s \in [0,1]$ such that $|u-s| \leq 1$, we have

$$\tilde{Q}_1(u) = Q_1(\gamma(u)) = \varphi_{\tilde{\boldsymbol{\theta}}_1}(u-s, s, \tilde{Q}_1(s)) = \varphi_{\boldsymbol{\theta}_1(\gamma)\gamma'}(u-s, s, \tilde{Q}_1(s))\,. \tag{6.26}$$

This suggests to consider the elastic distance on the measurable curves from $[0,1] \to SO(d)$ defined, for $P_1, P_2 \in \mathbb{L}^2([0,1], SO(d))$, as

$$d_2(P_1,P_2)^2 = \inf_{\gamma \in \mathrm{Diff}_+([0,1])} \int_0^1 \| \log(P_1(u)^T P_2(\gamma(u)))\|_F^2 du\,. \tag{6.27}$$

4.3.2 Estimation Criterion

By using this distance d_2 and the flow approximation given by the Magnus expansion (Theorem 3.1), the criterion (6.13) can be expressed as, for $k = 1, \ldots, N$,

$$\mathcal{V}_2(Q_k, \varphi_{\boldsymbol{\theta}}) = \int_0^1 d_2(Q_k, \varphi_{\boldsymbol{\theta}}(\cdot, s, Q_k(s)))ds \tag{6.28}$$

$$= \inf_{\gamma \in \mathrm{Diff}_+([0,1])} \int_0^1 d_1(Q_k, \varphi_{\boldsymbol{\theta}(\gamma)\gamma'}(\cdot, s, Q_k(s)))ds \tag{6.29}$$

$$= \inf_{\gamma \in \mathrm{Diff}_+([0,1])} \int_0^1 \int_0^1 \left\| \log\left(Q_k(t)^T Q_k(s) \exp\left(\Omega(t-s, s; \boldsymbol{\theta}(\gamma)\gamma')\right)\right)\right\|_F^2 dtds\,. \tag{6.30}$$

Then the two estimation criteria $\mathcal{V}_1$ and $\mathcal{V}_2$ are related by

$$\mathcal{V}_2(Q_k, \varphi_{\boldsymbol{\theta}}) = \inf_{\gamma \in \Gamma_S} \mathcal{V}_1(Q_k, \varphi_{\boldsymbol{\theta}(\gamma)\gamma'})\,. \tag{6.31}$$

The relation (6.25) defines a "spatial" or geometric registration based on the family of deformations defined as $\boldsymbol{\theta} \mapsto \gamma \cdot \boldsymbol{\theta} = \gamma' \boldsymbol{\theta} \circ \gamma$, for any increasing diffeomorphism γ. This is a group action, i.e. for all γ_1, γ_2 diffeomorphisms, and any Frenet curvatures $\boldsymbol{\theta} \in \mathcal{H}$, we have

$$(\gamma_2 \circ \gamma_1) \cdot \boldsymbol{\theta} = \gamma_2 \cdot (\gamma_1 \cdot \boldsymbol{\theta}) \, . \tag{6.32}$$

Note that the stretching action by warping does not permit transforming any geometry into another. Indeed, if $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are two generalised curvatures in a 3-dimensional space, such that the torsion $\tau_0 > 0$ and $\tau_1 < 0$, then we cannot find γ such that $\gamma' \tau_1(\gamma) = \tau_0$ (Brunel and Park, 2019). Thus, our mean parameter is identified as the solution to a constrained minimization problem.

Following the derivation of the criterion $\breve{\mathcal{I}}^{(1)}_{h,\lambda}(\boldsymbol{\theta})$ (6.18) proposed by Park and Brunel (2019), we define the new criterion $\breve{\mathcal{I}}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta}) = \breve{\ell}^{(2)}_{h,\sigma}(\boldsymbol{\theta}) + \mathcal{P}_\lambda(\boldsymbol{\theta})$ for the estimation of the mean parameter where $\mathcal{P}_\lambda(\boldsymbol{\theta})$ is a penalty term on $\boldsymbol{\theta}$ defined as in (6.19),

$$\breve{\ell}^{(2)}_{h,\sigma}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=1}^{N} \inf_{\gamma_k} \left\{ \iint_{[0,1]} K_h(t-s) \| \log \left(Q_k(t)^T Q_k(s) \exp \left(\Omega(t-s, s; \boldsymbol{\theta} \cdot \gamma_k) \right) \right) \|_F^2 \, dsdt + \tilde{\mathcal{P}}_\sigma(\gamma_k) \right\}, \tag{6.33}$$

and $\tilde{\mathcal{P}}_\sigma(\gamma_k)$ is an additional penalty term to ensure the smoothness of the optimal warping functions $\gamma_k \in \mathrm{Diff}_+([0,1])$ and control the amount of warping with the coefficient $\sigma > 0$. This penalty term is defined as

$$\tilde{\mathcal{P}}_\sigma(\gamma) = \sigma \int_0^1 \|\gamma'(s) - 1\|^2 ds. \tag{6.34}$$

Definition 6.4. *Let $\{Q_1, \ldots, Q_N\}$ be a sample of Frenet paths with parameters of Frenet curvatures $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$. For a fixed h, λ and σ, the sample mean vector field (or Frenet curvatures) is defined as the parameter $\breve{\boldsymbol{\theta}}^{(2)}_{h,\lambda,\sigma}$ which minimizes the global prediction error $\breve{\mathcal{I}}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta})$:*

$$\breve{\boldsymbol{\theta}}^{(2)}_{h,\lambda,\sigma} \in \arg\min_{\boldsymbol{\theta} \in \mathcal{H}} \breve{\mathcal{I}}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta}) \, . \tag{6.35}$$

As previously mentioned, we can also define the mean Frenet path $\breve{Q}^{(2)}_{h,\lambda,\sigma}(t) = \exp\left(\Omega(t, 0, \breve{\theta}^{(2)}_{h,\lambda,\sigma})\right)$ and the corresponding mean shape $\breve{X}^{(2)}_{h,\lambda,\sigma}$ obtained by integrating the gradient. For their practical computation, we consider the same approximation of the flow as previously, using the Magnus expansion, leading to the approximated criterion $\mathcal{I}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta}) = \ell^{(2)}_{h,\sigma}(\boldsymbol{\theta}) + \mathcal{P}_\lambda(\boldsymbol{\theta})$, where the first term in $\ell^{(2)}_{h,\sigma}(\boldsymbol{\theta})$ is

$$\iint_{[0,1]} K_h(t-s) \left\| \log \left(Q_k(t)^T Q_k(s) \exp \left((t-s) A_{\boldsymbol{\theta}} \left(\gamma_k \left(\frac{s+t}{2} \right) \right) \gamma_k' \left(\frac{s+t}{2} \right) \right) \right) \right\|_F^2 dsdt \, . \tag{6.36}$$

Then, Proposition 6.1 is also true for the criterions $\mathcal{I}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta})$ and $\breve{\mathcal{I}}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta})$ and verified by the same demonstration. By denoting $\Delta^{(2)}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \log\left(Q_k(t)^T Q_k(s) \exp\left((t-s) A_{\boldsymbol{\theta}} \left(\gamma_k \left(\frac{s+t}{2} \right) \right) \gamma_k' \left(\frac{s+t}{2} \right) \right) \right)$ we can derive, similarly to the previous case, the approximation

$$\Delta^{(2)}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = (t-s) \left(\gamma_k' \left(\frac{s+t}{2} \right) A_{\boldsymbol{\theta}} \left(\gamma_k \left(\frac{s+t}{2} \right) \right) - A_{\boldsymbol{\theta}_k} \left(\frac{s+t}{2} \right) \right) + (t-s)^2 C_2(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \, . \tag{6.37}$$

Therefore, the criterion $\mathcal{I}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta})$ can be approximated as

$$\mathcal{I}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=1}^{N} \inf_{\gamma_k \in \mathrm{Diff}([0,1])} \left\{ 2\sigma_K^2 h^2 \|\boldsymbol{\theta}_k - \boldsymbol{\theta} \cdot \gamma_k\|_{\mathbb{L}^2}^2 + \sigma \int_0^1 \|\gamma_k'(t) - 1\|^2 dt \right\} + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt + O(h^3). \tag{6.38}$$

The approximation of the criterion $\mathcal{I}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta})$ obtained in (6.38), has the form of a weighted and regularized criterion for the optimization of a "elastic" Karcher mean in the space of Frenet curvatures $\mathcal{H}$. Therefore, given this new estimated mean parameter, we expect to obtain a mean shape closer to the Karcher mean in the SRC space. Indeed, as mentioned in Remark 5.1, the registration problem considered here differs from the SRC one (5.31) only in the template under consideration and deformed: here this is $\boldsymbol{\theta}$, and not $\frac{\boldsymbol{\theta}}{\|\sqrt{\boldsymbol{\theta}}\|}$ as in the case of the SRC transform. Furthermore, a significant difference arises in that the group action $(\boldsymbol{\theta}\cdot\gamma)(s) = \boldsymbol{\theta}(\gamma(s))\gamma'(s)$ is not symmetric with respect to the $\mathbb{L}^2$ norm, unlike the group action of $\mathrm{Diff}_+([0,1])$ on the SRC transform. However, one approach to address this issue would be to consider an $\mathbb{L}^1$ norm instead, under which the group action $\boldsymbol{\theta}\cdot\gamma$ becomes symmetric.

4.4 Mean Parameter Estimation using the "Elastic" SRV Distance on $SO(d)$-valued Curves

In this section, we aim to define a criterion for optimizing the mean parameter based on the "regularization-based" approach of Park and Brunel (2019), which approximates the parameter corresponding to the Karcher mean in the space of SRCs. For this purpose, we choose this time to use the SRV distance on $SO(d)$-valued curves to derive the explicit expression of the self-prediction criterion (6.13).

4.4.1 Choice of the Distance

We have introduced the SRV Transform for curves on $SO(d)$ in Definition 5.1 of Chapter 5, that is, for $P \in \mathbb{L}^2([0,1], SO(d))$, $q(P)(t) = P(t)^T\dot{P}(t)/\sqrt{\|\dot{P}(t)\|_F}$. This gives rise to the SRV distance between two parametrized curves on $SO(d)$ (Bauer et al., 2021):

$$d(P_1,P_2)^2 = \|\log(P_1(0)^TP_2(0))\|_F^2 + \int_0^1 \|q(P_1)(u) - q(P_2)(u)\|_F^2 du. \tag{6.39}$$

By approaching the problem from time-parametrized curves as before, we extend this distance on the quotient space $\mathbb{L}^2([0,1], SO(d))/\mathrm{Diff}([0,1])$, that is,

$$d_3(P_1,P_2)^2 = \|\log(P_1(0)^TP_2(0))\|_F^2 + \inf_{\gamma\in\mathrm{Diff}([0,1])}\int_0^1 \|q(P_1)(u) - q(P_2\circ\gamma)(u)\|_F^2 du. \tag{6.40}$$

4.4.2 Estimation Criterion

Again, this new distance can be used to derive an explicit expression of the self-prediction criterion (6.13), instead of the geodesic distance used to derive the estimation criteria $\breve{\mathcal{I}}^{(1)}_{h,\lambda}(\boldsymbol{\theta})$ and $\breve{\mathcal{I}}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta})$. In this case, the self-prediction error criterion $\mathcal{V}_3(Q_k,\varphi_{\boldsymbol{\theta}})$ is written as

$$\mathcal{V}_3(Q_k,\varphi_{\boldsymbol{\theta}}) = \int_0^1 d_3(Q_k, \varphi_{\boldsymbol{\theta}}(\cdot,s,Q_k(s)))ds \tag{6.41}$$

$$= \inf_{\gamma\in\mathrm{Diff}_+([0,1])}\int_0^1\int_0^1 \|q(Q_k)(t) - q(\varphi_{\boldsymbol{\theta}(\gamma)\gamma'}(\cdot,s,Q_k(s))(t)\|_F^2\, dtds. \tag{6.42}$$

Now, instead of considering the Magnus approximation of the flow, we can directly notice that from the definition of the flow we have $\varphi_{\boldsymbol{\theta}(\gamma)\gamma'}(t-s,s,Q_k(s)) = \tilde{Q}(t)$ such that $\tilde{Q}'(t) = \tilde{Q}(t)A_{\tilde{\boldsymbol{\theta}}}(t)$ with $\tilde{\boldsymbol{\theta}}(t) = \boldsymbol{\theta}(\gamma(t))\gamma'(t)$ and $\tilde{Q}(s) = Q_k(s)$. Thus, the SRV transform of the flow can be expressed as

$$q\left(\varphi_{\boldsymbol{\theta}(\gamma)\gamma'}(t-s,s,Q_k(s)\right) = \frac{A_{\tilde{\boldsymbol{\theta}}}(t)}{\sqrt{\|A_{\tilde{\boldsymbol{\theta}}}(t)\|_F}} = \sqrt{\gamma'(t)}\frac{A_{\boldsymbol{\theta}}(\gamma(t))}{\sqrt{\|A_{\boldsymbol{\theta}}(\gamma(t))\|_F}}, \tag{6.43}$$

which no longer depends on the variable s. Therefore, the self-prediction error criterion $\mathcal{V}_3(Q_k, \varphi_{\boldsymbol{\theta}})$ is simply

$$\mathcal{V}_3(Q_k, \varphi_{\boldsymbol{\theta}}) = \inf_{\gamma \in \mathrm{Diff}_+([0,1])} \int_0^1 \left\| q(Q_k)(t) - \sqrt{\gamma'(t)} \frac{A_{\boldsymbol{\theta}}(\gamma(t))}{\sqrt{\|A_{\boldsymbol{\theta}}(\gamma(t))\|_F}} \right\|_F^2 dt \,. \tag{6.44}$$

From this, we can define the estimation criterion $\breve{\mathcal{I}}^{(3)}_{h,\lambda,\sigma}(\boldsymbol{\theta}) = \breve{\ell}^{(3)}_{h,\sigma}(\boldsymbol{\theta}) + \mathcal{P}_\lambda(\boldsymbol{\theta})$ of the mean parameter with

$$\breve{\ell}^{(3)}_{h,\sigma}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=1}^{N} \inf_{\gamma_k} \left\{ \int_{[0,1]} \left\| q(Q_k)(t) - \sqrt{\gamma_k'(t)} \frac{A_{\boldsymbol{\theta}}(\gamma_k(t))}{\sqrt{\|A_{\boldsymbol{\theta}}(\gamma_k(t))\|_F}} \right\|_F^2 dt ds + \tilde{\mathcal{P}}_\sigma(\gamma_k) \right\} . \tag{6.45}$$

Definition 6.5. *Let $\{Q_1, \ldots, Q_N\}$ be a sample of Frenet paths with parameters of Frenet curvatures $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$. For a fixed h, λ and σ, the sample mean vector field (or Frenet curvatures) is defined as the parameter $\breve{\boldsymbol{\theta}}^{(3)}_{h,\lambda,\sigma}$ which minimizes the global prediction error $\breve{\mathcal{I}}^{(3)}_{h,\lambda,\sigma}(\boldsymbol{\theta})$:*

$$\breve{\boldsymbol{\theta}}^{(3)}_{h,\lambda,\sigma} \in \arg\min_{\boldsymbol{\theta} \in \mathcal{H}} \breve{\mathcal{I}}^{(3)}_{h,\lambda,\sigma}(\boldsymbol{\theta}) \,. \tag{6.46}$$

We define then the corresponding mean Frenet Path $\breve{Q}^{(3)}_{h,\lambda,\sigma}(t) = \exp\left(\Omega(t, 0, \breve{\theta}^{(3)}_{h,\lambda,\sigma})\right)$ and the mean shape $\breve{X}^{(3)}_{h,\lambda,\sigma}$. In this case, unlike the previous two ones, we do not need an additional approximation, as we already have an expression as a function of $\boldsymbol{\theta}$. Moreover, for $k \in [\![1, N]\!]$, Q_k is a solution of the Frenet-Serret ODE, $Q_k'(s) = Q_k(s) A_{\boldsymbol{\theta}_k}(s)$, and consequently, its SRV transform can be expressed as:

$$q(Q_k)(s) = \frac{Q_k(s)^T Q_k'(s)}{\sqrt{\|Q_k'(s)\|_F}} = \frac{A_{\boldsymbol{\theta}_k}(s)}{\sqrt{\|A_{\boldsymbol{\theta}_k}(s)\|_F}} \,. \tag{6.47}$$

Then, using the Proposition 5.3, the criterion $\breve{\mathcal{I}}^{(3)}_{h,\lambda,\sigma}(\boldsymbol{\theta})$ can simply be expressed as

$$\breve{\mathcal{I}}^{(3)}_{h,\lambda,\sigma}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=1}^{N} \inf_{\gamma_k} \left\{ \int_0^1 \left\| \frac{\boldsymbol{\theta}_k(t)}{\sqrt{\|\boldsymbol{\theta}_k(t)\|_2}} - \sqrt{\gamma_k'(t)} \frac{\boldsymbol{\theta}(\gamma_k(t))}{\sqrt{\|\boldsymbol{\theta}(\gamma_k(t))\|_2}} \right\|_2^2 dt \right.$$
$$\left. + \sigma \int_0^1 \|1 - \gamma_k'(t)\|^2 dt \right\} + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt \,. \tag{6.48}$$

The obtained equation (6.48) defines a regularized criterion for the optimization of the functional parameter $\boldsymbol{\theta}$ corresponding to an estimator of the Frenet curvatures of the Karcher mean shape in the shape space defined by the SRC transforms (6.8). In this case, the group action is symmetric under the $\mathbb{L}^2$ norm, which was not the case with the criterion $\breve{\mathcal{I}}^{(2)}_{h,\lambda,\sigma}$. Given raw and noisy observations of individual parameters $\boldsymbol{\theta}_k$, this "regularization-based" approach offers a unique estimation criterion that correctly expresses the necessary optimization constraints due to the infinite dimension of the mean parameter. Therefore, this constitutes an alternative approach to the "preprocessing-based" one introduced in Section 3, which requires N independent optimization problems.

4.5 Estimation Algorithms

Based on the statistical criteria developed in the previous sections, the estimation of the mean parameter $\boldsymbol{\theta}^{(p)}$ from given Frenet paths $\boldsymbol{Q} = \{Q_1, \ldots, Q_N\}$, in each case $p = \{1, 2, 3\}$, is done by solving,

$$\widehat{\boldsymbol{\theta}}^{(p)}_{h,\lambda,\sigma}(\cdot) = \arg\min_{\theta} \mathcal{I}^{(p)}_{h,\lambda,\sigma}(\boldsymbol{\theta}) \,. \tag{6.49}$$

4.5.1 Cases of $\mathcal{I}^{(1)}_{h,\lambda}$ and $\mathcal{I}^{(2)}_{h,\lambda,\sigma}$

We need to solve the nonparametric estimation problem (6.49), but in practice, we solve this by discretization. Suppose that $Q_k, k = 1, \dots, N$ is available at finite grid points $s_{ki}, i = 1, \dots, n_k$. Then we discretize the integral on a grid $0 = t_{k1} < t_{k2} < \cdots < t_{km_k} = 1$ and minimize for $\mathcal{I}^{(1)}_{h,\lambda}$:

$$\min_{\theta\in\mathcal{H}} \frac{1}{N}\sum_{k=1}^{N} \frac{1}{n_k m_k} \sum_{i,j=1}^{n_k,m_k} K_h(t_{kj}-s_{ki}) \left\| \log\left(Q_k(t_{kj})^T Q_k(s_{ki}) \exp\left((t_{kj}-s_{ki})\, A_{\boldsymbol{\theta}}\left(\frac{t_{kj}+s_{ki}}{2}\right)\right)\right)\right\|_F^2 + \lambda\|\boldsymbol{\theta}''\|_2^2 \tag{6.50}$$

and for $\mathcal{I}^{(2)}_{h,\lambda}$:

$$\min_{\theta\in\mathcal{H}} \frac{1}{N}\sum_{k=1}^{N} \inf_{\gamma_k} \frac{1}{n_k m_k} \sum_{i,j=1}^{n_k,m_k} K_h(t_{kj}-s_{ki}) \left\| \log\left(Q_k(t_{kj})^T Q_k(s_{ki}) \exp\left((t_{kj}-s_{ki})\, A_{\boldsymbol{\theta}\cdot\gamma_k}\left(\frac{t_{kj}+s_{ki}}{2}\right)\right)\right)\right\|_F^2 + \sigma\|1-\gamma_k'\|_2^2 + \lambda\|\boldsymbol{\theta}''\|_2^2. \tag{6.51}$$

As done in Chapter 3, because the presence of the exponential makes the optimization difficult, we use an additional approximation that provides a simple algorithm and simplifies the analysis of our estimator. We have shown in Section 5 of Chapter 3 that by denoting $u_{kij} = t_{kj} - s_{ki}$, $v_{kij} = \frac{t_{kj}+s_{ki}}{2}$,

$$L_{kij} = \log\left(Q_k(t_{kj})^T Q_k(s_{ki}) \exp\left(u_{kij} A_{\boldsymbol{\theta}}\left(v_{kij}\right)\right)\right) \quad \text{and} \quad R_{kij} = -\frac{1}{u_{kij}} \log\left(Q_i(t_{kj})^T Q_i(s_{ki})\right),$$

we can derive the following approximation:

$$\sum_{k,i,j=1}^{N,n_k,m_k} \frac{1}{N n_k m_k} K_h(u_{kij}) \|L_{kij}\|_F^2 = \sum_{k,i,j=1}^{N,n_k,m_k} \frac{1}{N n_k m_k} K_h(u_{kij}) u_{kij}^2 \|A_{\boldsymbol{\theta}}(v_{kij}) - R_{kij}\|_F^2 + O(h^3). \tag{6.52}$$

This motivates us to introduce the new approximated criteria,

$$\tilde{\mathcal{I}}^{(1)}_{h,\lambda}(\boldsymbol{\theta}; \boldsymbol{R}) = \sum_{k=1}^{N} \sum_{i,j=1}^{n_k,m_k} \frac{1}{N n_k m_k} K_h(u_{kij}) u_{kij}^2 \|A_{\boldsymbol{\theta}}(v_{kij}) - R_{kij}\|_F^2 + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt, \tag{6.53}$$

$$\tilde{\mathcal{I}}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta}; \boldsymbol{R}) = \sum_{k=1}^{N} \inf_{\gamma_k \in \mathrm{Diff}_+([0,1])} \sum_{i,j=1}^{n_k,m_k} \frac{1}{N n_k m_k} K_h(u_{kij}) u_{kij}^2 \|\gamma_k'(v_{kij}) A_{\boldsymbol{\theta}}(\gamma_k(v_{kij})) - R_{kij}\|_F^2 + \sigma \int_0^1 \|1-\gamma_k'(t)\|^2 dt + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt. \tag{6.54}$$

Using Proposition 5.3, we can replace in the previous criteria the Frobenious norm on $A_{\boldsymbol{\theta}}$ by the Euclidean norm on the relevant components $\boldsymbol{\theta}$. We define the weights $\omega_{kij} = (1/N n_k m_k) K_h(u_{kij}) u_{kij}^2$, and denote by $\boldsymbol{r}_{kij}$ the vector corresponding to the first sub-diagonal of the matrix R_{kij}, that is $\boldsymbol{r}_{kij} = L^T R_{kij}^{\vee}$ where L is defined as in (4.22) and $\vee$ is the operator introduced in (2.6). Then we obtain

$$\tilde{\mathcal{I}}^{(1)}_{h,\lambda,\sigma}(\boldsymbol{\theta}; \boldsymbol{R}) = \sum_{k=1}^{N} \sum_{i,j=1}^{n_k,m_k} \omega_{kij} \|\boldsymbol{\theta}(v_{kij}) - \boldsymbol{r}_{kij}\|_2^2 + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt. \tag{6.55}$$

$$\tilde{\mathcal{I}}^{(2)}_{h,\lambda,\sigma}(\boldsymbol{\theta}; \boldsymbol{R}) = \sum_{k=1}^{N} \inf_{\gamma_k} \sum_{i,j=1}^{n_k,m_k} \omega_{kij} \|\gamma_k'(v_{kij}) \boldsymbol{\theta}(\gamma_k(v_{kij})) - \boldsymbol{r}_{kij}\|_2^2 + \sigma \int_0^1 \|1-\gamma_k'(t)\|^2 dt + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt. \tag{6.56}$$

This gives rise to the computation of vector-valued smoothing splines (with splines of third order), defined at the knots v_{kij}, with the pseudo-observations $\boldsymbol{r}_{kij}$ (see Section 2.1.1 of Chapter 2). The only

difference with respect to the classical smoothing splines is the presence of the weights ω_{kij}. In the second case, before computing the mean function using a B-spline decomposition, we need to solve the N alignment problems:

$$\inf_{\gamma_k \in \mathrm{Diff}_+([0,1])} \sum_{i,j=1}^{n_k,m_k} \omega_{kij} \|\gamma_k'(v_{kij})\boldsymbol{\theta}(\gamma_k(v_{kij})) - \boldsymbol{r}_{kij}\|_2^2 + \sigma \int_0^1 \|1 - \gamma_k'(t)\|^2 dt. \tag{6.57}$$

Since $\boldsymbol{\theta}$ is also unknown, the alignment problem that is generally considered in this case is the "inverse" one. In other words, we solve instead the following problem:

$$\inf_{\gamma_k \in \mathrm{Diff}_+([0,1])} \sum_{i,j=1}^{n_k,m_k} \omega_{kij} \|\boldsymbol{\theta}(v_{kij}) - (\boldsymbol{r}_{kij} \cdot \gamma_k)(v_{kij})\|_2^2 + \sigma \int_0^1 \|1 - \gamma_k'(t)\|^2 dt. \tag{6.58}$$

However, the two alignment problems (6.57) and (6.58) are not equivalent here as the group action $(\boldsymbol{\theta} \cdot \gamma)(s) = \boldsymbol{\theta}(\gamma(s))\gamma'(s)$ is not symmetric with the $\mathbb{L}^2$ norm. Therefore, by considering (6.58) instead of (6.57) in the criterion (6.56), we make an additional approximation.

For the alignment of the raw estimates based on the problems (6.58), one can use various methods proposed in the literature (Kneip and Ramsay, 2008; Tucker et al., 2013; Tucker, 2018). The simplest one is to use a weighted Karcher mean algorithm, which we have implemented as detailed in Algorithm 4.

Algorithm 4 Alignment Algorithm - Weighted Karcher Mean

Given observations $(\boldsymbol{r}_k)_{k=1,\dots,N}$, set the initial values $y_k^{(0)} = \boldsymbol{r}_k$, $\nu^{(0)} = \boldsymbol{r}_i$, where $i = \arg\min_{1 \le i \le N} \|\boldsymbol{r}_i - \sum_k \omega_k y_k^{(0)}\|$. For $\ell \ge 1$, iterate the following steps until convergence $\|\sum_k \omega_k y_k^{(\ell)} - \nu^{(\ell-1)}\| < \epsilon$:

1: Update γ_k: $\gamma_k^{(\ell)} = \arg\min_\gamma \|\nu^{(\ell-1)} - (y_k^{(\ell-1)} \circ \gamma)\dot{\gamma}\|_2$ for $k = 1, \dots, N$ with a dynamic programming algorithm.
2: Update y_k: $y_k^{(\ell)} = (y_k^{(\ell-1)} \circ \gamma_k^{(\ell)})\dot{\gamma}_k^{(\ell)}$ for $k = 1, \dots, N$
3: Update ν: $\nu^{(\ell)} = \sum_k \omega_k y_k^{(\ell)}$

At step 1 of this algorithm, a standard Dynamic Programming algorithm (Bertsekas, 1995) is used to find the optimal warping functions.

4.5.2 Case of $\mathcal{I}^{(3)}_{h,\lambda,\sigma}$

Since the data assumed to be available are Frenet paths $\{Q_1, \dots, Q_N\}$, using the criterion $\mathcal{I}^{(3)}_{h,\lambda,\sigma}$ for estimating $\boldsymbol{\theta}$ requires an additional preprocessing step to compute the SRV transforms of the Frenet paths $\{q(Q_1), \dots, q(Q_N)\}$. For $k \in [\![1, N]\!]$, Q_k is a solution of the Frenet-Serret ODE, $Q_k'(s) = Q_k(s)A_{\boldsymbol{\theta}_k}(s)$, and consequently, as we have seen, its SRV transform can be expressed in two ways:

$$q(Q_k)(t) = \frac{Q_k(t)^T Q_k'(t)}{\sqrt{\|Q_k'(t)\|_F}} = \frac{A_{\boldsymbol{\theta}_k}(s)}{\sqrt{\|A_{\boldsymbol{\theta}_k}(s)\|_F}}. \tag{6.59}$$

Hence, we can think of two different methods for obtaining an estimate of $q(Q_k)$. The first approach involves estimating the first derivative of the $SO(d)$-valued function $s \mapsto Q_k(s)$ based on its potentially noisy observations. This is not straightforward and will require local polynomial regression methods adapted to curves on manifolds such as those proposed in Bickel and Li (2007); Hinkle et al. (2012, 2014). Alternatively, a second method for estimating $q(Q_k)$ is to employ its expression as a function of $\boldsymbol{\theta}_k$. The estimation of $\hat{\boldsymbol{\theta}}_k$ can be carried out using one of the methods proposed in Part I or by directly using the formula of the raw Frenet curvatures derived previously and also used in the EM

algorithm of Chapter 4, that is, for small $h > 0$, $\hat{\boldsymbol{\theta}}_k(s+h) = \frac{1}{h}\log\left(Q_k(s+h)^T Q_k(s)\right)$. We then obtain the approximated criterion

$$\tilde{\mathcal{I}}^{(3)}_{h,\lambda,\sigma}(\boldsymbol{\theta}) = \sum_{k=1}^{N} \inf_{\gamma_k} \sum_{j=1}^{m_k} \frac{1}{Nm_k} \left\| \gamma_k'(t_j) \frac{\boldsymbol{\theta}(\gamma_k(t_j))}{\sqrt{\|\boldsymbol{\theta}(\gamma_k(t_j))\|_2}} - \frac{\hat{\boldsymbol{\theta}}_k(t_j)}{\sqrt{\|\hat{\boldsymbol{\theta}}_k(t_j)\|_2}} \right\|_2^2 + \sigma \int_0^1 \|1 - \gamma_k'(t)\|^2 dt + \lambda \int_0^1 \|\boldsymbol{\theta}''(t)\|^2 dt\,. \tag{6.60}$$

Again, optimizing the parameter $\boldsymbol{\theta}$ from this latter criterion can be done with a B-splines smoothing algorithm and a similar alignment algorithm as Algorithm 4 adapted to this different group action.

4.5.3 Optimization of Hyperparameters

Our prediction error depends on h, λ, σ. We decide to fix $\sigma > 0$. The hyperparameter λ is chosen to be a vector in $\mathbb{R}^{d-1}$, that is, one regularization parameter for each Frenet curvatures. In particular, in three dimensions, we have $\lambda = (\lambda_1, \lambda_2)$. If h is too big, we integrate along the whole interval, and the errors accumulate, then it is better to restrict to a smaller interval. In our numerical studies, we select optimal parameters h, λ with a 10-fold cross-validation minimizing for each criterion the errors:

case $p = 1$:

$$\frac{1}{KN} \sum_{\ell=1}^{K} \sum_{k=1}^{N} \frac{1}{|T_\ell|} \sum_{i \in T_\ell} \left\| \log\left(Q_k(s_{ki})^T \,\mathrm{FSODEsol}\left(s_{ki}\,;\, \widehat{\boldsymbol{\theta}}^{-(\ell)}_{h,\lambda} \right) \right) \right\|_F^2 , \tag{6.61}$$

case $p = 2$:

$$\frac{1}{KN} \sum_{\ell=1}^{K} \sum_{k=1}^{N} \frac{1}{|T_\ell|} \sum_{i \in T_\ell} \left\| \log\left(Q_k(s_{ki})^T \,\mathrm{FSODEsol}\left(s_{ki}\,;\, \widehat{\boldsymbol{\theta}}^{-(\ell)}_{h,\lambda} \cdot \gamma_k^{-1} \right) \right) \right\|_F^2 , \tag{6.62}$$

case $p = 3$:

$$\frac{1}{KN} \sum_{\ell=1}^{K} \sum_{k=1}^{N} \frac{1}{|T_\ell|} \sum_{i \in T_\ell} \left\| q(Q_k)(s_{ki}) - q\left(\mathrm{FSODEsol}\left(\cdot\,;\, \widehat{\boldsymbol{\theta}}^{-(\ell)}_{h,\lambda} \cdot \gamma_k^{-1} \right) \right)(s_{ki}) \right\|_F^2 , \tag{6.63}$$

where T_ℓ is the ℓth index set based on $K = 10$ random partition of the observations $\{Q_k(s_{ki})\}$, $k = 1, \ldots, N, i = 1, \ldots, n_k$ and $\mathrm{FSODEsol}(s, \widehat{\boldsymbol{\theta}}^{-(\ell)}_{h,\lambda})$ is the solution, evaluated at point s, of the Frenet-Serret differential equation solved with parameters $\widehat{\boldsymbol{\theta}}^{-(\ell)}_{h,\lambda}$ which is estimated without the ℓth partition dataset and using hyperparameters h, λ. The differential equation is solved each time with the initial condition $\bar{Q}_0 = \arg\min_{Q \in SO(d)} \sum_{k=1}^{N} \|\log(Q^T Q_k(0))\|_F^2$.

4.6 Extension for Mean Shape of Spherical Curves in $\mathbb{S}^2$

Our formulation does not require a specific structure on the Euclidean curves. Nevertheless, it is of interest if our method is applicable to structured data such as curves on a manifold. Of course, it is possible to estimate the Frenet curvatures without additional knowledge on the manifold. However, since curvature and torsion for spherical curves in $\mathbb{S}^2$ are intrinsically related, direct estimation does not necessarily respect the constraints, but a constrained optimization is not obvious in this setting either. It turns out that, instead of modifying the algorithm, we can reformulate the problem under our Frenet framework for the spherical curves in three-dimensional space.

We consider a curve α on a sphere of radius R and center $(0,0,0)$. By definition we have $\|\alpha(t)\| = R$ for all $t \in [0,1]$. We now consider the curve parametrized by arc length. As for all $s \in [0, L]$, $\|\alpha(s)\| = R$, we have $\langle \alpha(s), \alpha(s) \rangle = R^2$ so $2\langle \alpha(s), \alpha'(s) \rangle = 0$, thus $\alpha(s)$ is orthogonal to $\alpha'(s)$ for all s. We denote

$\beta := \alpha'$. We define the spherical unit normal as $\gamma(s) = (1/R)\alpha(s) \wedge \beta(s)$. Since $\|\alpha(s)\|/R = 1 = \|\beta(s)\|$ for all s and the two are orthogonal, $\|\gamma(s)\| = 1$ too.

Definition 6.6. *Define the geodesic curvature of a spherical curve $\alpha : [0, L] \longrightarrow \mathbb{S}^2$ parametrised by arclength to be*

$$k_g(s) = \langle \alpha''(s), \gamma(s) \rangle .$$

The geodesic curvature measures the failure of a curve to be a geodesic.

Proposition 6.2. *(Frenet-Serret formula for spherical frames) Let $\alpha : [0, L] \longrightarrow \mathbb{S}^2$, unit sphere, be a spherical curve parametrised by arclength. Let $\beta(s) = \alpha'(s)$ and $\gamma(s) = \alpha(s) \wedge \beta(s)$. The vectors (α, β, γ) define the spherical frame and satisfy the following equation with $k_g(s) = \langle \alpha''(s), \gamma(s) \rangle$*

$$\begin{cases} \alpha'(s) = \beta(s) \\ \beta'(s) = -\alpha(s) + k_g(s)\gamma(s) \\ \gamma'(s) = -k_g(s)\beta(s) \end{cases}$$

Proof. The first equation is by definition. Since β has constant norm we know it is orthogonal to its derivative. We know that $\alpha''.\gamma = k_g$ by definition. Differentiating β we obtain $\beta' = \alpha''$, so $\beta'.\alpha = \alpha''.\alpha$. By differentiating the equality $\alpha.\alpha' = 0$ we obtain

$$0 = \alpha'.\alpha' + \alpha.\alpha'' = 1 + \alpha.\alpha''$$

so $\alpha.\alpha'' = -1$. Finally, we want to compute γ'. By differentiating $\gamma.\alpha = 0$ we have

$$0 = \gamma'.\alpha + \gamma.\alpha' = \gamma'.\alpha + \gamma.\beta = \gamma'.\alpha.$$

Moreover γ is orthogonal to γ'. Therefore we need to compute $\gamma'.\beta$. We differentiate $\gamma.\beta = 0$,

$$0 = \gamma'.\beta + \gamma.\beta' = \gamma'.\beta + \gamma.(-\alpha + k_g\gamma) = \gamma'.\beta - \gamma.\alpha + k_g = \gamma'.\beta + k_g.$$

Therefore $\gamma'.\beta = -k_g$. ■

This proposition implies that if one knows the initial position and direction, a given geodesic curvature function $k_g(s)$ determines a unique spherical curve parametrised by arclength. Therefore, we can directly apply our algorithm with the Frenet frame for spherical curves to obtain an estimate of the geodesic curvature $\hat{k}_g$. Then we reconstruct the curve by solving the spherical Frenet-Serret ODE above. This method ensures that the estimated mean is in $\mathbb{S}^2$.

5 Simulations Studies

We conduct simulation studies in $\mathbb{R}^3$ to assess the performance of the proposed methods in identifying mean geometry (curvature, torsion) and mean shape in finite samples. We compare the 7 means considered in the chapter. First, we have the arithmetic mean $\mu^{(arithm)}$ (6.3) (abbreviated "Arithm", plotted in **red**) and the SRVF Karcher mean $\mu^{(SRVF)}$ (6.4) (abbreviated "SRVF", plotted in **green**) defined in Section 2. Then, we have the two means computed with the "preprocessing-based" approach, defined in Section 3: the approximated Frenet curvatures Karcher mean $\hat{\mu}^{(\boldsymbol{\theta})}$ (abbreviated "FC", plotted in **pink**) and the approximated SRC Karcher mean $\hat{\mu}^{(SRC)}$ (abbreviated "SRC", plotted in **cyan**). Finally, we have the three means defined in Section 4 with the "regularization-based" approach, $X^{(1)}, X^{(2)}, X^{(3)}$, obtained from the three corresponding mean parameters defined in Definition 6.3, Definition 6.4 and Definition 6.5 (abbreviated "V1", "V2" and "V3", plotted in **blue**, **purple**, and **orange** respectively). We recall here that the means $\hat{\mu}^{(\boldsymbol{\theta})}$ and $V1$ both approximate the Frenet curvatures Karcher mean (6.6),

and the means $\hat{\mu}^{(\mathrm{SRC})}$ and $V3$ both approximate the SRC Karcher mean (6.8).

For computing the means V1, V2, and V3, the framework proposed by Park and Brunel (2019), along with the extension presented here, assumes the availability of Frenet paths observations. However, such observations are rarely readily accessible and typically derived from Euclidean curves through preprocessing steps. In Chapter 3, we introduced two methods for estimating Frenet paths from Euclidean curves as a preprocessing step: the Gram-Schmidt orthonormalization of curve derivatives estimated using local polynomial regression or the constrained local polynomial regression of Frenet vectors. Both methods are shown to be quite equivalent. In these simulations, we are using the Gram-Schmidt orthonormalization of curve derivatives estimated by local polynomial regression to estimate the Frenet paths used as "pseudo" observations for consistency with the Frenet paths estimation method used in the "preprocessing-based" approach. The selection of optimal hyperparameters for individual Frenet curvatures estimation in the "preprocessing-based" approach of Section 3, and mean parameters estimation in the "regularization-based" approach of Section 4 are done with a Bayesian optimization using the corresponding cross-validation criteria. The SRVF mean is computed using the python package `fdasrsf`.

5.1 Curves Simulated from a Generative Model on Frenet Curvatures

We define a generative model of curves in $\mathbb{R}^3$ from Frenet curvatures parameters. The population of Frenet curvatures is generated by adding amplitude and phase variability to reference parameters. That is, given the reference parameters $\kappa^{(ref)}(s) = \frac{1}{3}\exp(4\sin(5s - 0.75) + 0.25)$ and $\tau^{(ref)}(s) = 8s - 3$, for $s \in [0, 1]$, we generate each individual parameters $\boldsymbol{\theta}_k = (\kappa_k, \tau_k)^T$ with the model

$$\begin{cases} \kappa_k(s) & = a_k \kappa^{(ref)}(\gamma(s\,;\, b_k)) \\ \tau_k(s) & = c_k \tau^{(ref)}(\gamma(s\,;\, b_k)) \end{cases}$$

where $a_k \sim \mathcal{N}(1, \sigma_\kappa)$, $c_k \sim \mathcal{N}(0, \sigma_\tau)$, $b_k \sim \mathcal{N}(0, \sigma_\gamma)$, and $\gamma(s\,;\, b_k) = \frac{\exp(b_k s)-1}{\exp(b_k)-1}$ if $b_k \neq 0$, otherwise $\gamma(s\,;\, b_k) = s$. Then, given the parameters $\boldsymbol{\theta}_k$, the Frenet path Q_k and the curve X_k are constructed by solving the Frenet-Serret equations. First, we consider two populations of $N = 20$ curves. The first one has only amplitude variability in the population of parameters $\{\boldsymbol{\theta}_k\}_{k=1,\dots,N}$, that is $\sigma_\gamma = 0$. In the second population, we add phase variability with $\sigma_\gamma = 2$. In both cases we fix $\sigma_\kappa = 0.2$ and $\sigma_\tau = 1.5$. The two populations are plotted in grey in Figure 6.2 and Figure 6.3. In the spirit of the 2D loops example discussed in Chapter 5, the curves generated here take the form of 3D loops, where the loop's localization along the curve depends on the phase variability of the curvatures and the variability in torsions influences whether the loop is more or less planar. The 7 means of these populations are computed and plotted in colors on the same figures. The hyperparameters are optimized in the ranges $h \in [0.02, 0.1]$, $\lambda_1, \lambda_2 \in [10^{-15}, 10^{-8}]$.

5.1.1 Influence of Sample Size and Noise Level

In practice, curves may be available in the form of noisy observations. Thus, starting from the population of 20 curves generated with variability in both amplitude and phase of Frenet curvatures, we consider the following observation model:

$$\boldsymbol{y}_{ki} = X_k(s_i) + \boldsymbol{\epsilon}_{ki}, \qquad k = 1, \dots, N, \quad i = 1, \dots, n_k,$$

where $\boldsymbol{\epsilon}_{ki} \sim \mathcal{N}(0, \sigma_{\boldsymbol{y}}^2 \mathbf{I})$. In addition to testing the robustness of different means to noise and the number of discretization points, this simulation explores the significance of considering an explicit statistical model for the mean parameter estimation as proposed in the "regularization-based" approach of Section 4. This approach provides a unified framework for estimating both the mean and raw individual curvatures, whereas the "preprocessing-based" approach of Section 3 requires individual estimates of curvatures,

which can introduce bias. We consider the different sample sizes $n_k = 100, 200$ for all $k = 1, \dots, N$ and noise levels $\sigma_{\boldsymbol{y}} = 0.01, 0.005$. The selection of optimal hyperparameters is done in the ranges $h \in [0.02, 0.15]$, $\lambda_1, \lambda_2 \in [10^{-15}, 10^{-5}]$.

5.1.2 Influence of Random Shape-Preserving Transformations

In the case of real data, curves often have additional variations that do not change the intrinsic shape of the curve, known as shape-preserving transformations, such as translation, rotation, any parameterization, and variable length or scale. By definition, the average shape of a population of curves should not be affected by these variations. Therefore, to test the robustness of the various estimation methods of the mean shape to these variations, we propose to add variations of this type to the population of curves considered in Figure 6.3, having phase and amplitude variability in their Frenet curvatures. For each curve X_k, $k = 1, \dots, N$, we generate a random rotation $R_k \sim \mathcal{F}(I_3, \alpha)$ simulated from a Fisher-Langevin distribution (3.66) with mean identity and concentration $\alpha = 5$, a random length $l_k \sim \mathcal{U}(1, 10)$ and a random parameterization $h_k(t) = a_k \sin(2\pi t) + t$ is $a_k \neq 0$, otherwise $h_k(t) = t$, where $a_k \sim \mathcal{U}(-1.2, 1.2)$. Then, given the discrete regular time-grid $t_1 < \ldots < t_{n_k}$, with $n_k = 100$ for all k, the considered curves are

$$\boldsymbol{y}_{ki} = l_k R_k X_k(h_k(t_i)), \qquad k = 1, \dots, N,\ i = 1, \dots, n_k.$$

Without adding observation noise, we compute the 7 different means of this random population. The selection of optimal hyperparameters is done in the ranges $h \in [0.03, 0.12]$, $\lambda_1, \lambda_2 \in [10^{-15}, 10^{-8}]$.

5.1.3 Results

In the case of the population having only amplitude variability in Frenet curvatures, the various means displayed in Figure 6.2 demonstrate that FC, SRC, V1, V2, and V3 means are nearly identical. This is particularly apparent when looking at the shapes of the corresponding curvatures and torsions. In this scenario, alignment during mean curvature and torsion computation is typically unnecessary, resulting in V1 and V2 means having the same formulation and V3 being very similar. Additionally, in both cases, Figure 6.2 (amplitude variability only) and Figure 6.3 (amplitude and phase variability), the estimated means FC and SRC match the estimated V1 and V3 means respectively, as expected in these noise-free scenarios due to the approximations of the optimization criteria done and the powerful preprocessing method considered for computing the FC and SRC means. Thus, in noise-free scenarios, the difference between the two approaches is not significant. Furthermore, in both cases, we observe that arithmetic and SRVF mean shapes are inconsistent with the population's shapes. These two means exhibit two peaks of curvature, contrary to the single peak seen in all population curves, and, in the second case, inconsistent torsion shapes. This confirms the results and observations done in Chapter 5, showing that methods based on Frenet curvatures to define a mean shape are more consistent as they encode more information about the geometry.

Subsequently, observations with noisy data of the population having amplitude and phase variability in Frenet curvatures (Section 5.1.1) highlight the interest of the "regularization-based" approach proposed in Section 4 through the different criteria developed for mean shape estimation. By comparing the mean FC with the mean V1, and the mean SRC with the mean V3, the Table 6.1, Table 6.2, and Table 6.3 showing the distance between estimated parameters in the noise-free and noisy cases, demonstrate that means computed with a "regularization-based" approach are more robust in the presence of noise. This confirms the conjecture that individual estimation of Frenet curvatures as a preprocessing step introduces a bias in the computation of Karcher means, which is limited by the unified formulation proposed in the criteria of Section 4. These conclusions are also visible in Figure 6.4 and Figure 6.5. Curvatures and torsions in Figure 6.5 appear less noisy and disparate in the case of means V1, V2, and V3. Regarding the tables, they provide only an indication of the result since the $\mathbb{L}^2$ distance depends

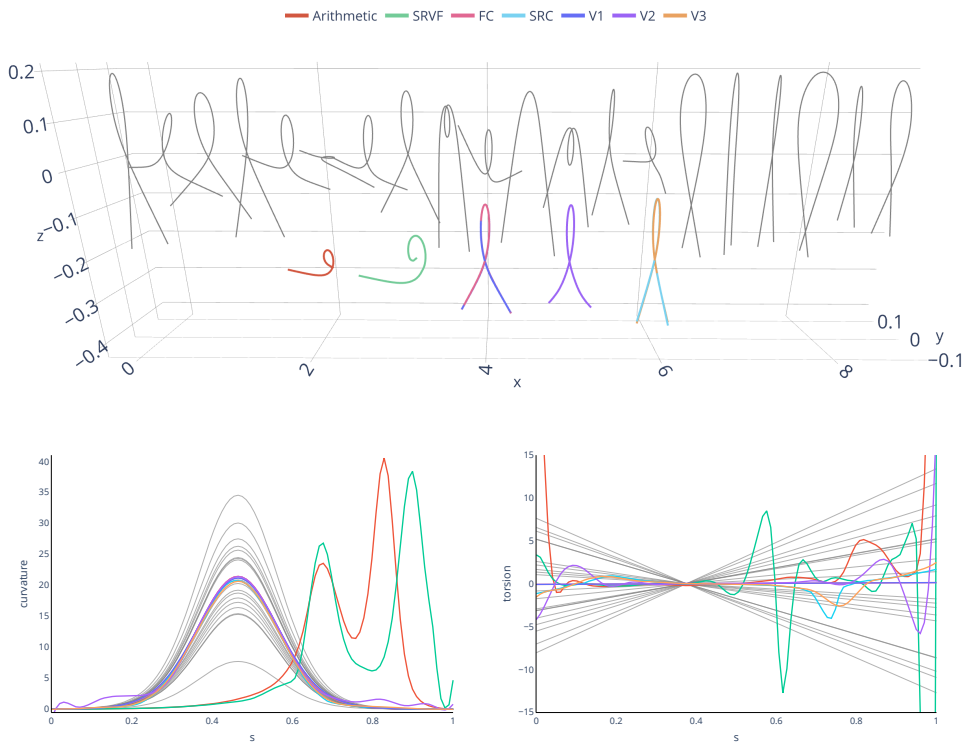


Figure 6.2: In grey: population of curves (top) and corresponding curvatures (bottom left) and torsions (bottom right) generated by adding **amplitude variability** to reference Frenet curvatures parameters. In color: 7 means and corresponding estimated curvature and torsion.

on the curve's parameterization, and for the arithmetic and SRVF means, curvatures and torsions correspond only to estimates. However, it appears that means based on Frenet curvatures are more stable concerning noise levels than the arithmetic and SRVF means. Moreover, as shown in Figure 6.4, the obtained means are smoother than the latter. This is mainly due to the implementation in which Frenet curvatures are estimated in a functional data framework, either smoothly with a B-spline decomposition.

Finally, by adding random shape-preserving transformations to the initial population visible in Figure 6.3, it appears from the results in Figure 6.6 that the SRVF and arithmetic means are not completely stable under these transformations. This instability is expected from the arithmetic mean but surprising from the SRVF one, which is theoretically well-defined concerning these transformations. It is likely due to the implementation. Specifically, in the SRVF case, the necessary reparameterization alignment is noise-sensitive, leading to different outcomes and significantly affecting the final mean. For means based on Frenet curvatures, there is no significant difference here between both estimation approaches. They are stable under these transformations. Additionally, comparing in Figure 6.7 the mean curvatures and torsions estimated by criteria V2 and V3, we observe that the estimates are less stable in the case of criterion V2. This is likely due to the fact that the reparameterization alignment problem in this case is not symmetrical, thus adding instability.

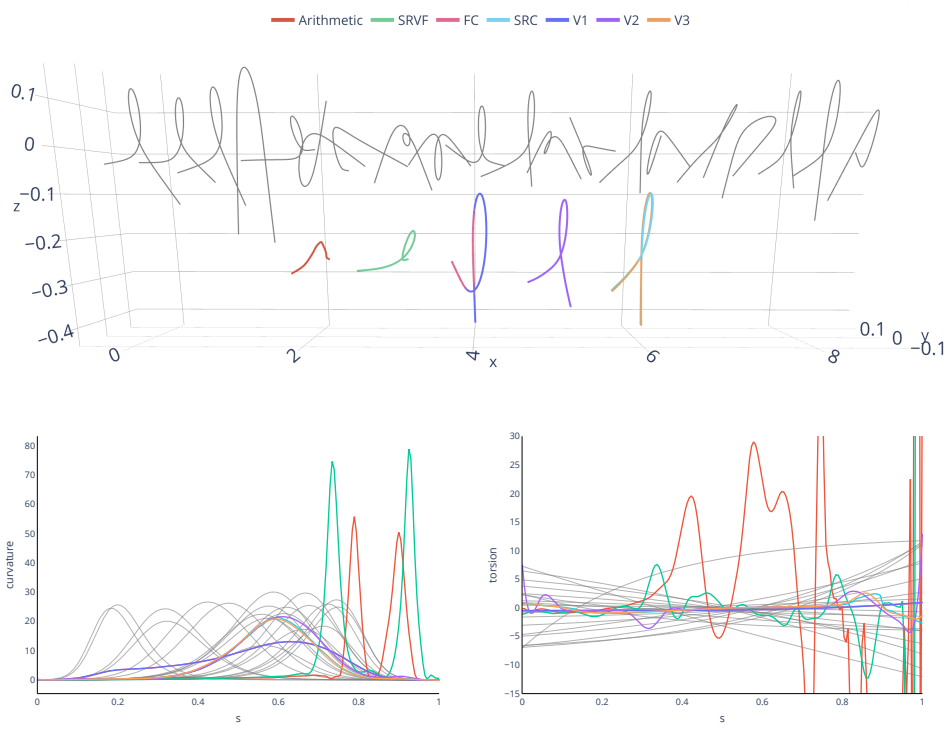


Figure 6.3: In grey: population of curves (top) and corresponding curvatures (bottom left) and torsions (bottom right) generated by adding **amplitude and phase variability** to reference Frenet curvatures parameters. In color: 7 means and corresponding estimated curvature and torsion.

n	$\sigma_{\boldsymbol{y}}$	Arithmetic	SRVF	FC	SRC	V1	V2	V3
100	0.005	0.024 (0.004)	0.320 (0.165)	0.365 (0.188)	0.526 (0.218)	0.271 (0.162)	0.492 (0.243)	0.439 (0.231)
	0.01	0.044 (0.006)	0.745 (0.266)	0.472 (0.141)	0.567 (0.251)	0.340 (0.130)	0.513 (0.253)	0.455 (0.229)
200	0.005	0.030 (0.003)	1.100 (0.365)	0.677 (0.282)	0.859 (0.485)	0.367 (0.199)	0.689 (0.423)	0.683 (0.400)
	0.01	0.058 (0.004)	1.787 (0.477)	0.870 (0.261)	0.914 (0.396)	0.510 (0.281)	0.792 (0.405)	0.717 (0.364)

Table 6.1: Estimation errors on **Euclidean curve**: $\mathbb{L}^2$ distance between estimated mean of noiseless population (Figure 6.3) and estimated mean of noisy population (Section 5.1.1). Means and standard deviations in parenthesis over the 100 repetitions of the simulation.

n	$\sigma_{\boldsymbol{y}}$	Arithmetic	SRVF	FC	SRC	V1	V2	V3
100	0.005	52.52 (9.50)	80.45 (20.78)	43.95 (4.48)	46.83 (16.57)	36.85 (5.21)	51.34 (12.95)	41.30 (13.23)
	0.01	66.96 (12.12)	101.01 (29.04)	72.66 (7.86)	61.40 (19.27)	62.65 (7.49)	77.14 (14.88)	56.78 (11.67)
200	0.005	99.65 (35.91)	136.78 (30.65)	86.34 (13.28)	89.42 (36.76)	62.85 (15.05)	95.80 (22.06)	71.51 (18.69)
	0.01	119.36 (44.29)	245.16 (37.10)	112.63 (16.55)	103.15 (31.49)	91.99 (15.54)	125.23 (25.29)	89.58 (19.68)

Table 6.2: Estimation errors on **curvature**: $\mathbb{L}^2$ distance between estimated mean of noiseless population (Figure 6.3) and estimated mean of noisy population (Section 5.1.1). Means and standard deviations in parenthesis over the 100 repetitions of the simulation.

n	σ_y	Arithmetic	SRVF	FC	SRC	V1	V2	V3
100	0.005	328.29 (60.64)	210.27 (23.31)	54.64 (10.09)	150.20 (15.54)	42.81 (15.75)	149.85 (40.53)	129.63 (40.33)
	0.01	332.56 (58.41)	194.92 (13.09)	57.16 (11.08)	149.74 (16.51)	43.16 (17.64)	152.01 (38.54)	121.53 (36.05)
200	0.005	694.17 (211.85)	573.37 (68.76)	102.51 (15.53)	263.54 (35.88)	70.55 (29.16)	255.95 (81.27)	234.71 (87.99)
	0.01	672.97 (139.42)	574.42 (15.07)	96.99 (17.65)	257.74 (34.02)	67.58 (29.07)	236.34 (73.39)	187.89 (68.96)

Table 6.3: Estimation errors on **torsion**: $\mathbb{L}^2$ distance between estimated mean of noiseless population (Figure 6.3) and estimated mean of noisy population (Section 5.1.1). Means and standard deviations in parenthesis over the 100 repetitions of the simulation.

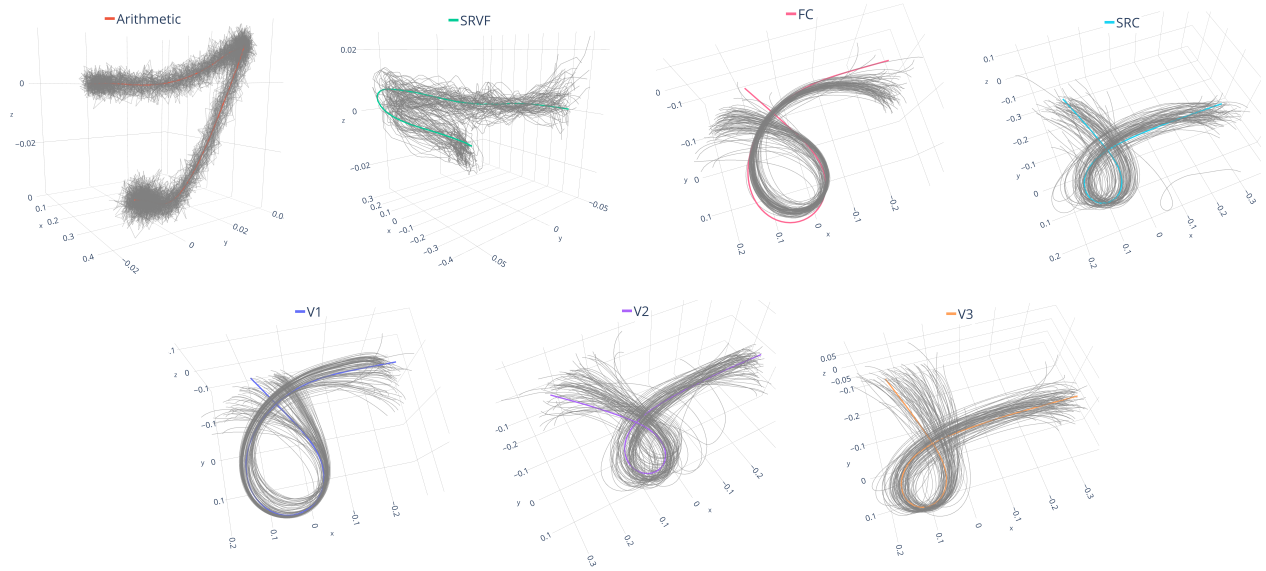


Figure 6.4: Case $n = 100$, $\sigma_y = 0.01$: 7 estimated means of the noiseless population in color and corresponding 100 estimated means of the noisy simulated populations (Section 5.1.1) in grey.

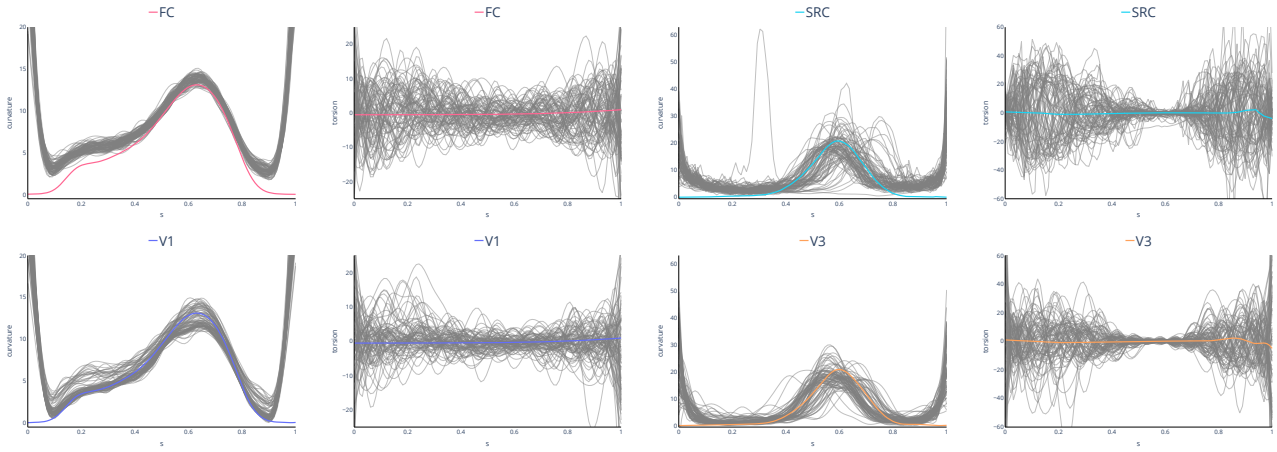


Figure 6.5: Case $n = 100$, $\sigma_y = 0.01$: curvature and torsion of estimated means of the noiseless population in color and corresponding 100 curvatures and torsions of estimated means of the noisy simulated populations (Section 5.1.1) in grey. Comparison between "preprocessing-based" and "regularization-based" approaches for approximation of Frenet curvatures Karcher mean (FC and V1), and SRC Karcher mean (SRC and V3).

	Arithmetic	SRVF	FC	SRC	V1	V2	V3
Error on X	0.497 (0.152)	0.503 (0.288)	0.100 (0.069)	0.212 (0.057)	0.009 (0.003)	0.104 (0.107)	0.069 (0.043)
Error on κ	102 (30)	108 (31)	0.254 (0.028)	3.075 (2.637)	0.308 (0.164)	6.688 (9.491)	4.791 (2.971)
Error on τ	377 (139)	199 (28)	12.78 (4.57)	14.69 (4.04)	12.85 (5.01)	32.22 (10.39)	18.82 (5.90)

Table 6.4: $\mathbb{L}^2$ distances between estimated means of the initial population (Figure 6.3) and estimated means of randomly transformed populations by shape-preserving transformations (Section 5.1.2). Means and standard deviations in parenthesis over the 100 repetitions of the simulation.

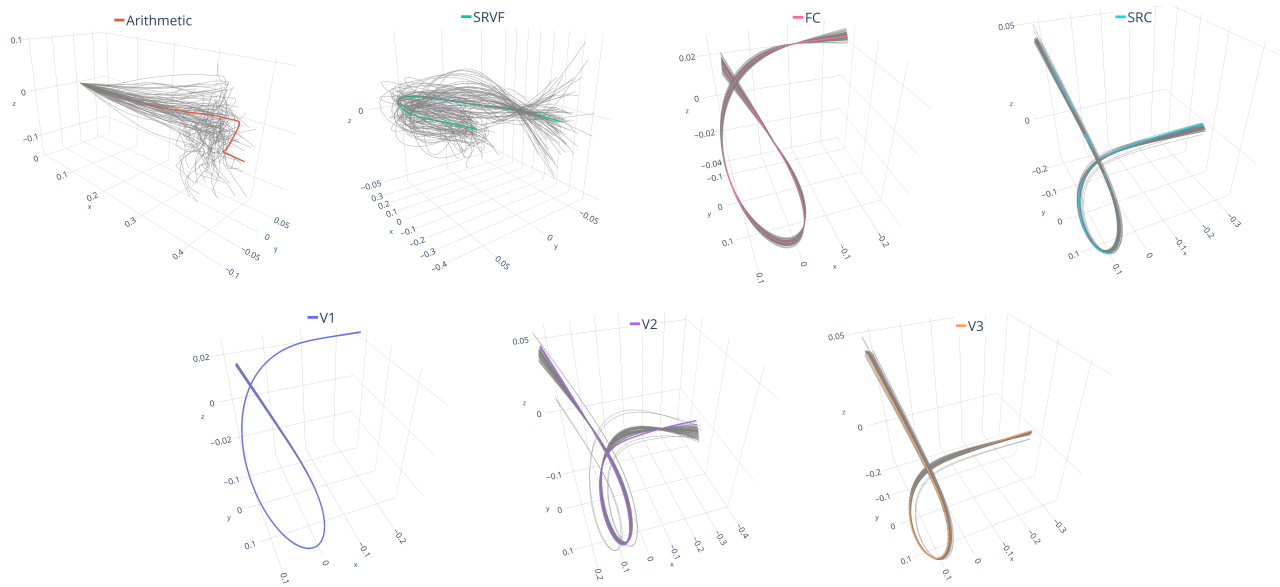


Figure 6.6: Means over randomly transformed population with shape-preserving transformations (Section 5.1.2): 7 estimated means of the initial population (Figure 6.3) in color and corresponding 100 estimated means of the noisy simulated populations in grey.

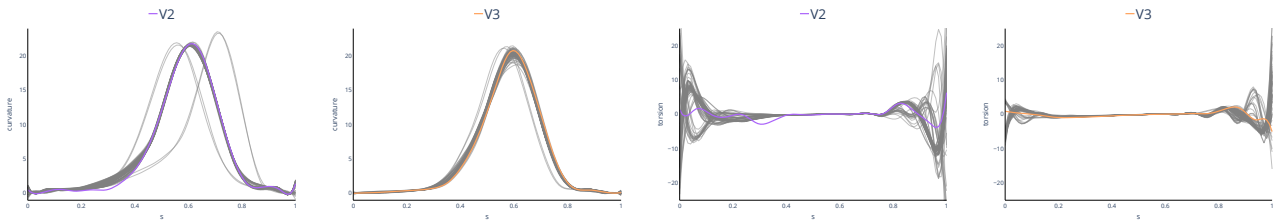


Figure 6.7: Case of randomly transformed population with shape-preserving transformations (Section 5.1.2): curvature and torsion of estimated means of the initial population in color and corresponding 100 curvatures and torsions of estimated means of the simulated populations in grey. Comparison between mean $X^{(2)}$ (V2) and mean $X^{(3)}$ (V3).

5.2 Curves on the Sphere

This scenario studies the special case of a population of curves lying on the manifold $\mathbb{S}^2$. We consider the generative model for curves on $\mathbb{S}^2$ described in Dai and Müller (2018). For $k = 1, ..., N$, the sample curves x_k are generated as $x_k : [0,1] \longrightarrow \mathbb{S}^2, x_k(t) = \exp_{\mu^{(ref)}(t)}\left(\sum_{\ell=1}^{20} \xi_\ell \phi_\ell(t)\right)$ with $\mu^{(ref)}(t) = \exp_{[0,0,1]}\left(\cos(\omega(t))\varphi(t), \sin(\omega(t))\varphi(t), 0\right)$ the reference function in $\mathbb{S}^2$, and the arbitrary chosen functions $\omega(t) = 10t + 0.5, \varphi(t) = 5(t+1)$. For $\ell = 1, ..., 20$, ξ_ℓ are generated by independent Gaussian distributions with mean zero and variance $0.07^{\ell/2}$. The functions $\phi_\ell(t)$ are defined on $[0,1]$ as $\phi_\ell(t) = 2^{-1/2} R_t[\Phi_\ell(t/2), \Phi_\ell((t+1)/2), 0]^T$, where R_t is the rotation matrix from $[0,0,1]$ to $\mu^{(ref)}(t)$, and $\{\Phi_\ell\}_{\ell=1}^{20}$ is the orthonormal Legendre polynomial basis on $[0,1]$. We finally add a random parametrization defined by $\gamma(s\,;\, b_k) = \frac{\exp(b_k s)-1}{\exp(b_k)-1}$ if $b_k \neq 0$, otherwise $\gamma(s\,;\, b_k) = s$, where $b_k \sim \mathcal{N}(0, 2.5)$. The measurements are then obtained, given the regular grid $t_1 < \cdots < t_{n_k}$ with $n_k = 100$ for all $k = 1, \ldots, N$, as

$$\boldsymbol{y}_{ki} = x_k(\gamma(t_i, b_k)), \qquad k = 1, \ldots, N,\ i = 1, \ldots, n_k.$$

In this case, we first compute the 7 different means without additional constraints for spherical curves. In addition, we compute, using the estimation criteria defined in Section 4, the 3 means based on the reformulation of the frame proposed in Section 4.6. Finally, for comparison, the arithmetic and SRVF mean shapes are also computed from the population of curves parameterized by arc length. The selection of optimal hyperparameters is done in the ranges $h \in [0.03, 0.1]$, $\lambda_1, \lambda_2 \in [10^{-15}, 10^{-8}]$.

Reference	Arithmetic	Arithmetic arc-length	SRVF	SRVF arc-length	FC	SRC
1.0 (0.0)	1.164 (0.287)	1.0 (0.045)	1.0 (0.069)	1.0 (0.035)	1.0 (0.018)	1.0 (0.02)

V1	V2	V3	V1 sphere	V2 sphere	V3 sphere
1.0 (0.02)	1.0 (0.053)	1.0 (0.047)	1.0 (0.001)	1.0 (0.001)	1.0 (0.001)

Table 6.5: Mean and standard deviation of the distances from each point along the curve to the center of the sphere.

Figure 6.8 and Table 6.5 demonstrate that considering the appropriate coordinate system and formulation for spherical curves, as proposed in Section 4.6, leads to mean shapes, using each of the three criteria, which lie on the sphere and are, therefore, more relevant. Indeed, means obtained without adding constraints related to spherical curves are not perfectly spherical. Furthermore, in the latter case, means based on Frenet curvatures are again more consistent with the geometry of the considered population of curves than the SRVF mean. Adapting the SRVF method to manifold data is also possible, and we refer to Su et al. (2014).

In terms of computational cost, in the setting of these simulations (20 curves, 100 sample points, 30 iterations of Bayesian optimization) and using a machine equipped with 96 CPUs and 376 GiB of RAM, the arithmetic mean is computed in $\approx 8e^{-5}$ seconds, the SRVF mean in ≈ 15 seconds. For the "preprocessing-based" approach, the FC and SRC means are computed in $\approx 1h15$, including the time of the preprocessing step. In the case of methods using a "regularization-based" approach, the mean V1 is computed in ≈ 50 minutes, the mean V2 in $\approx 1h30$, and finally, the mean V3 in ≈ 40 minutes. These times are given as an indication insofar as the computation times of these algorithms depend greatly on the parameters of the simulation. Furthermore, these calculation times could be accelerated by parallelizing the code whenever possible. This has not been done here because we are already parallelizing the 100 repetitions of each simulation.

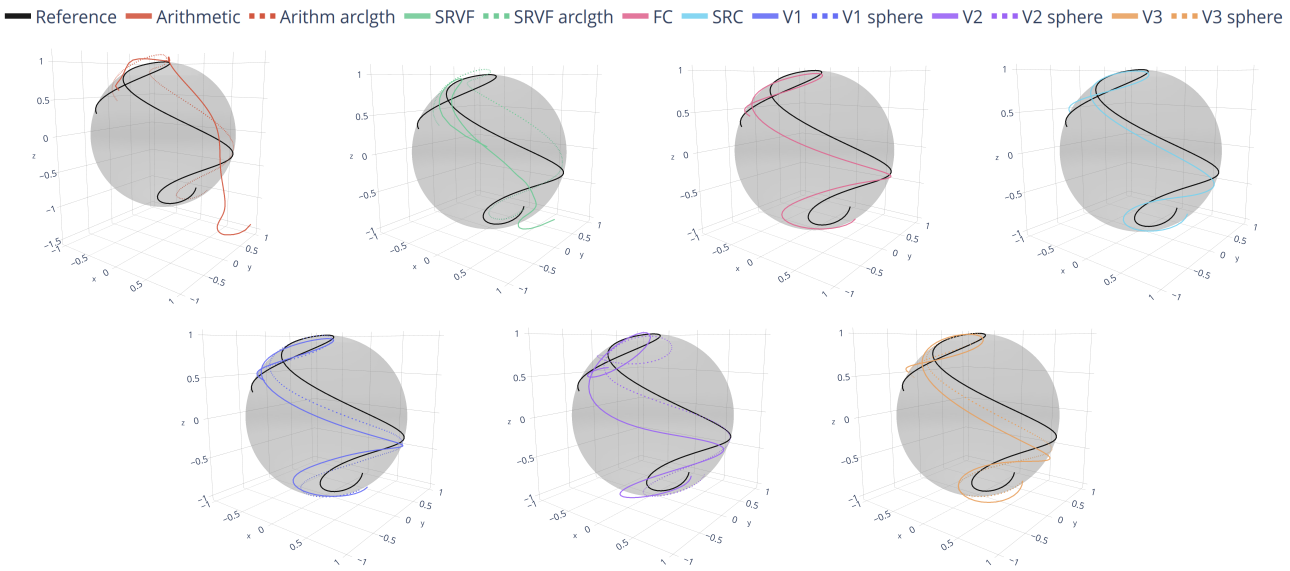
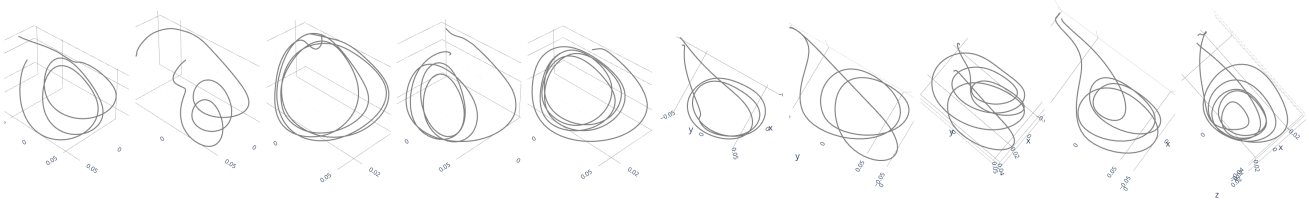


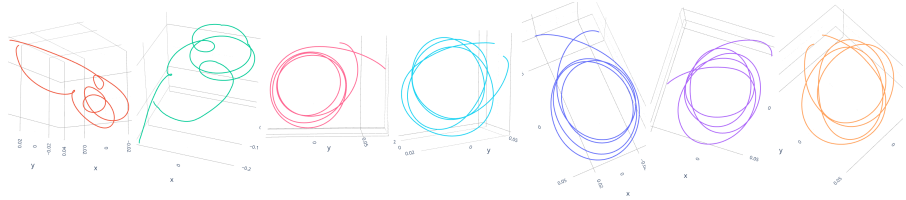
Figure 6.8: Spherical curves (Section 5.2): the 7 means without additional constraints are plotted in plain lines, and the 3 means using the "regularization-based" approach with adapted frame to spherical curves and the SRVF and arithmetic mean of arc-length parametrized curves are plotted in dotted lines.

6 Application to Sign Language Motion Trajectories

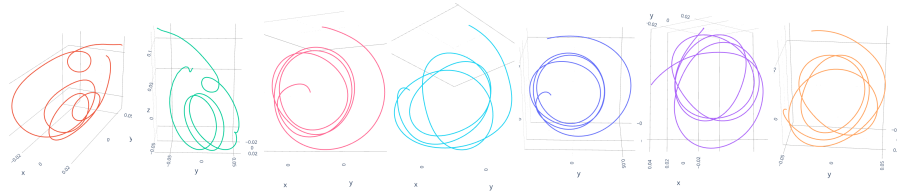
We are interested in analyzing a set of trajectories corresponding to repeated movements of the same sign in French Sign Language. As mentioned in the previous chapters, based on the literature on human motor control, Frenet curvatures appear to be particularly interesting parameters for studying and describing this type of movement. Therefore, we believe that using a statistical analysis method based on these parameters for such a curve population is more relevant. We propose to compare the 7 mean shapes defined in this chapter on this type of movement. That being said, although it forms the basis of statistical analysis of a population, it is not easy to quantify the quality of a mean and to compare the different possible definitions of the mean shape other than qualitatively. We consider the data described in Section 3.1. We propose to compute, for each sign, the 7 means of Aliza's movement trajectories, the 7 means of Thomas's movement trajectories, and finally, the 7 means of the union of these two sets. Compared to experiments with synthetic data, as mentioned in the application of Chapter 4, we have to deal with different irregular observation grids of trajectories in a population for estimating the mean Frenet curvatures within a functional data analysis framework. To address this, we consider, for each curve, a sub-grid corresponding to one out of every 4 points from the initial observation grid. Then, the knots grid used to define the B-spline decomposition of the mean parameters is defined by merging these different sub-grids. For all methods, the hyperparameters are selected with 30 iterations of a Bayesian optimization based on the corresponding cross-validation criteria in the ranges $h \in [0.05, 0.15]$ and $\lambda_1, \lambda_2 \in [10^{-30}, 10^{-5}]$. We show the obtained means for 5 signs. The sign "train" in Figure 6.9, the sign "avoir l'air" in Figure 6.10, the sign "essayer" in Figure 6.11, the sign "avril" in Figure 6.12 and the sign "autrefois" in Figure 6.13. We display the curves scaled to unit length. For better visualization and appreciation of the shapes, the grids are adjusted for each of the 3D curves and are not common to all. Indeed, without this, it is difficult to discern anything in still images like those presented here. We keep the same color code for the different means as the one used in the simulated data. Initial recorded trajectories and corresponding parameters are plotted in grey.



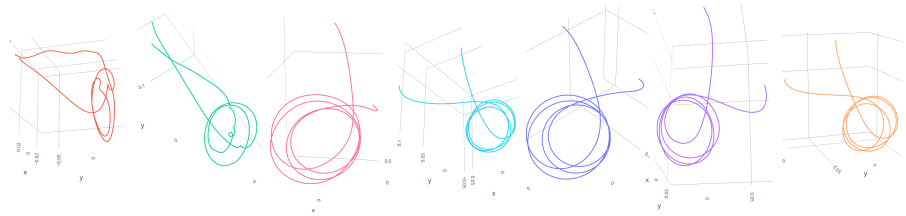
(a) Recorded trajectories of Thomas (first 5) and of Aliza (last 5).



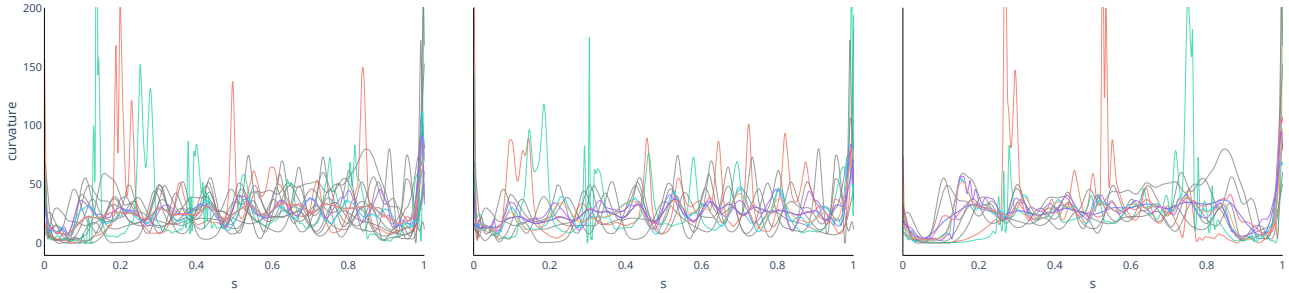
(b) Means on Thomas and Aliza trajectories.



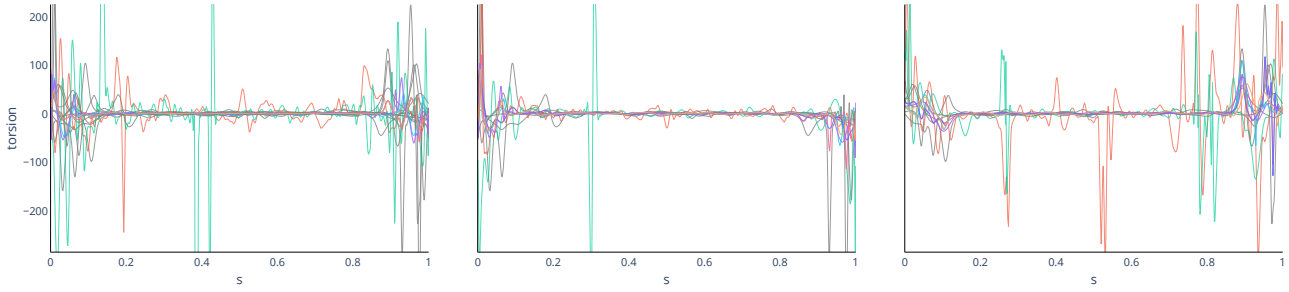
(c) Means on Thomas trajectories.



(d) Means on Aliza trajectories.

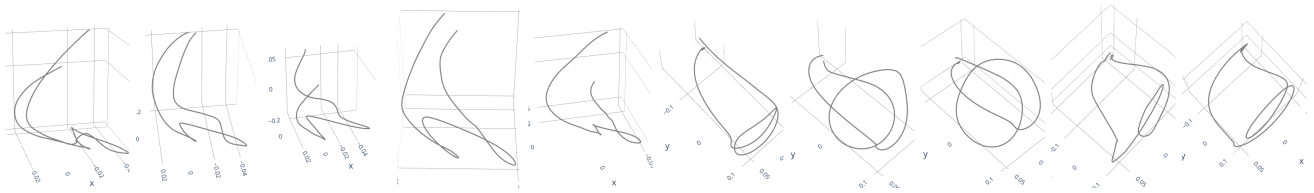


(e) Curvatures of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).

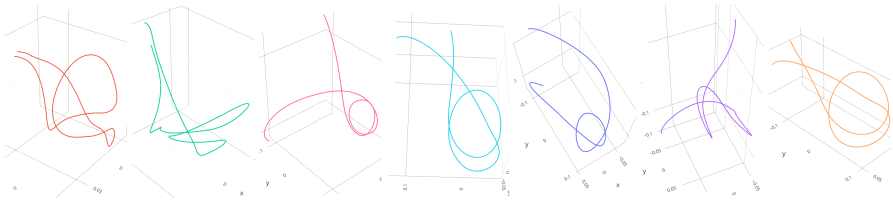


(f) Torsions of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).

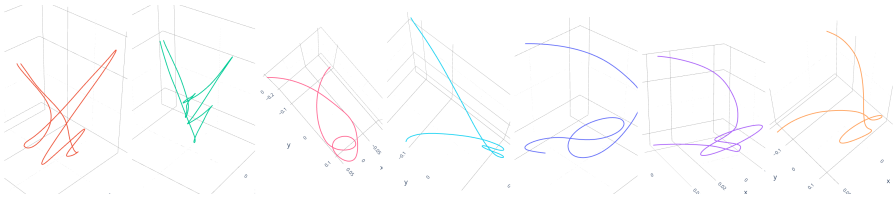
Figure 6.9: Sign "train".



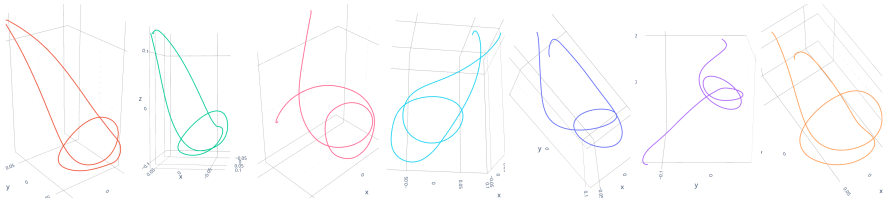
(a) Recorded trajectories of Thomas (first 5) and of Aliza (last 5).



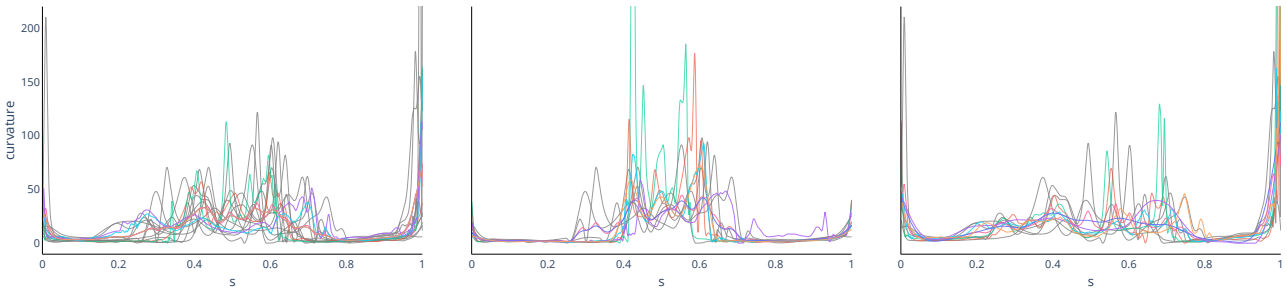
(b) Means on Thomas and Aliza trajectories.



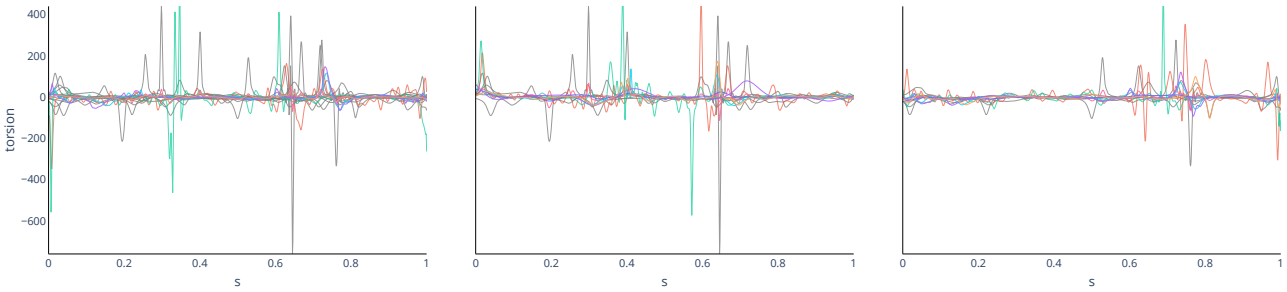
(c) Means on Thomas trajectories.



(d) Means on Aliza trajectories.

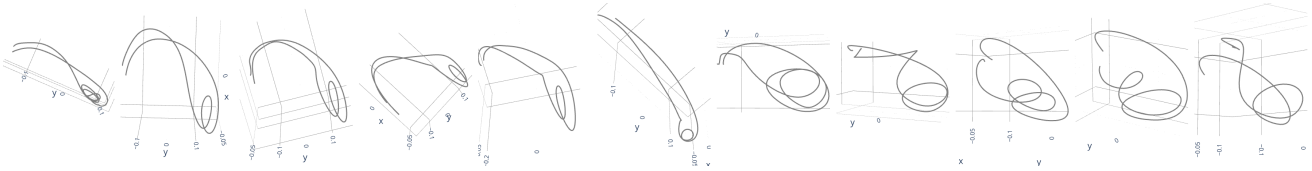


(e) Curvatures of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).

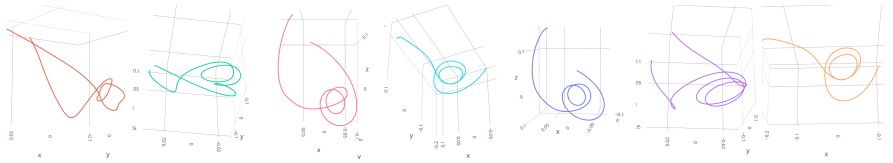


(f) Torsions of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).

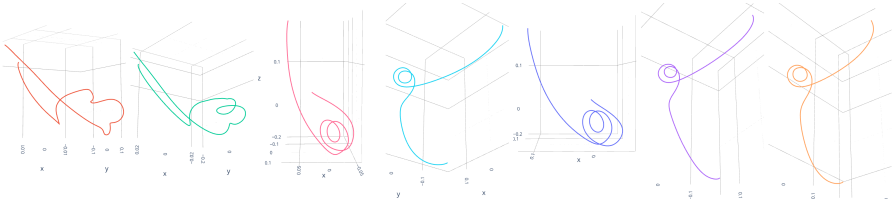
Figure 6.10: Sign "avoir l'air".



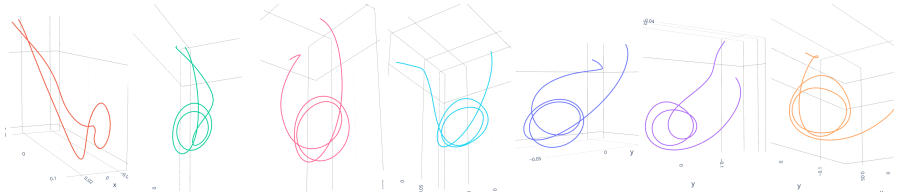
(a) Recorded trajectories of Thomas (first 6) and of Aliza (last 5).



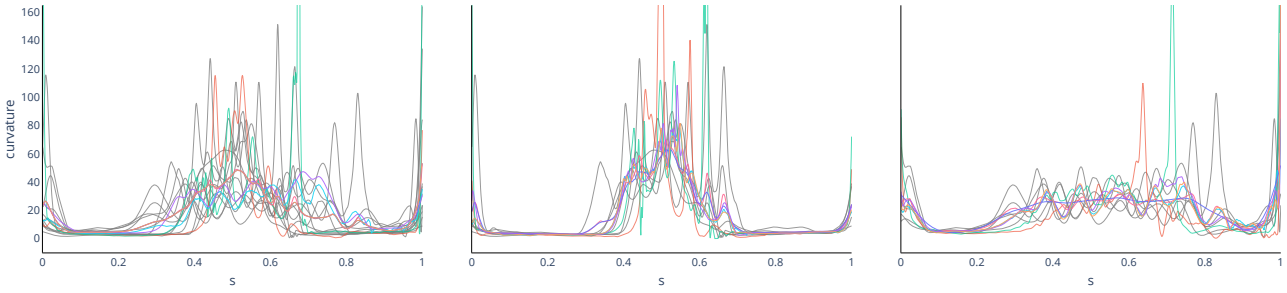
(b) Means on Thomas and Aliza trajectories.



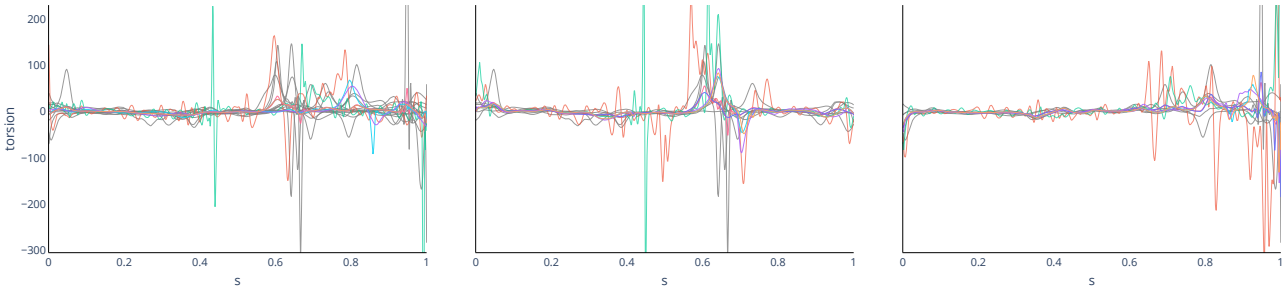
(c) Means on Thomas trajectories.



(d) Means on Aliza trajectories.

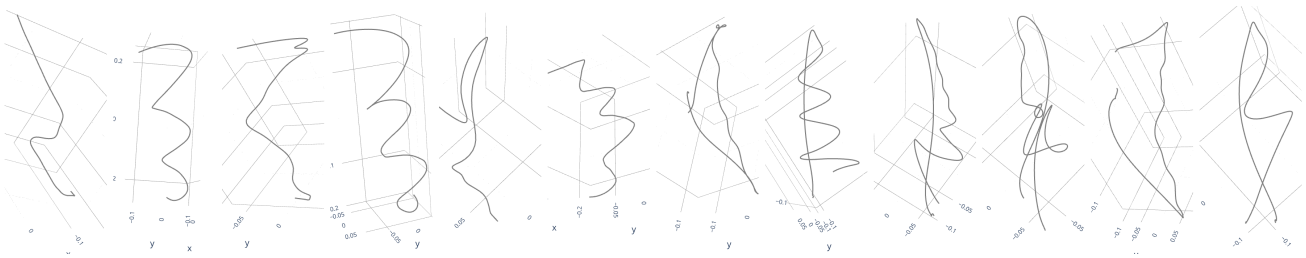


(e) Curvatures of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).

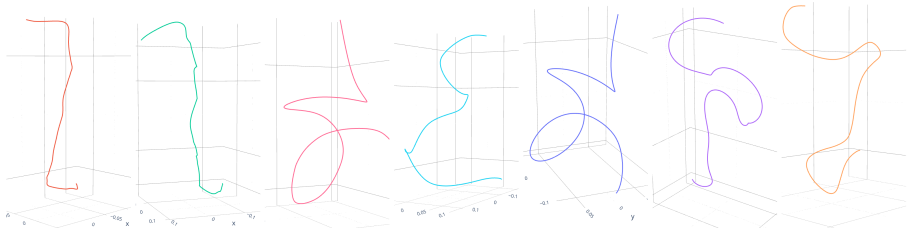


(f) Torsions of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).

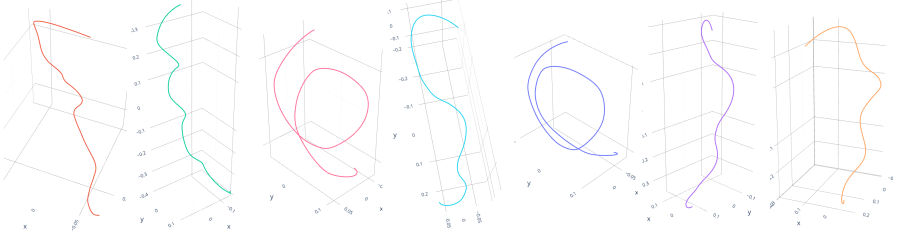
Figure 6.11: Sign "essayer".



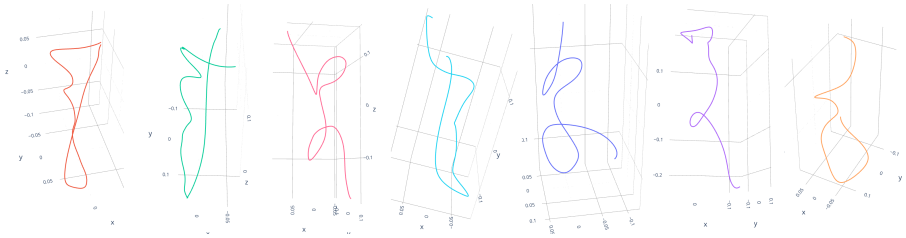
(a) Recorded trajectories of Thomas (first 5) and of Aliza (last 6).



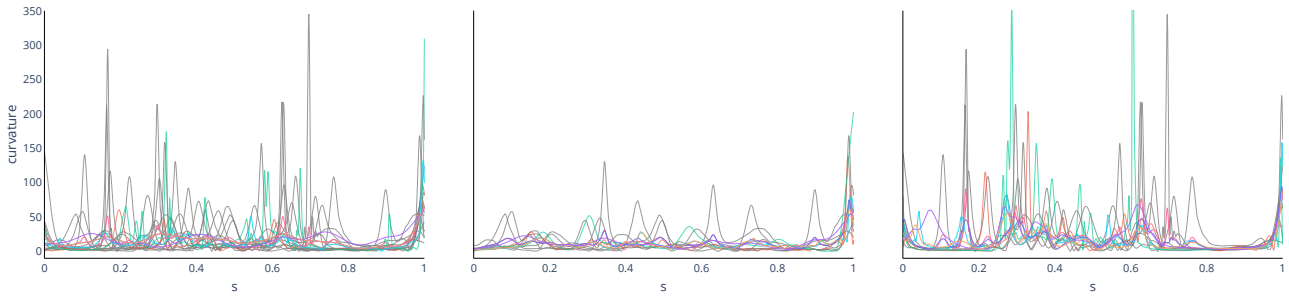
(b) Means on Thomas and Aliza trajectories.



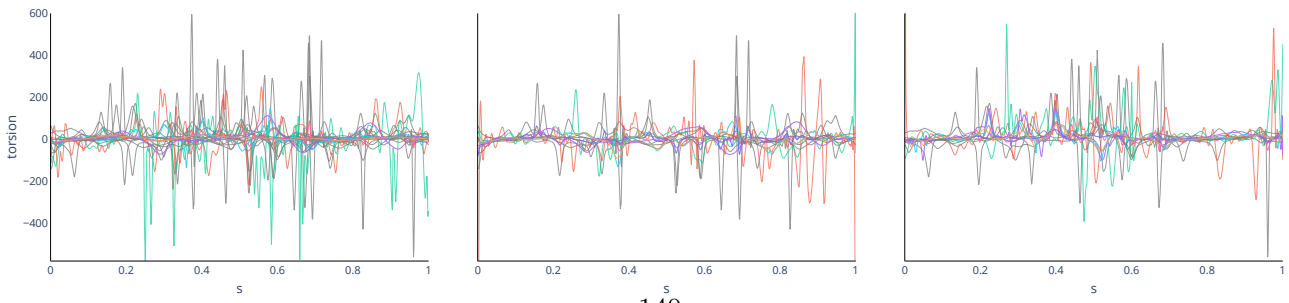
(c) Means on Thomas trajectories.



(d) Means on Aliza trajectories.

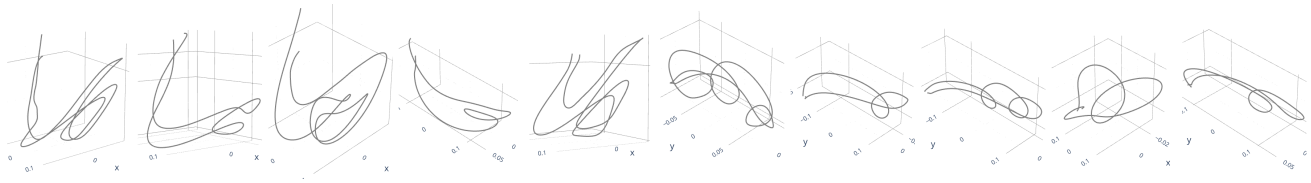


(e) Curvatures of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).

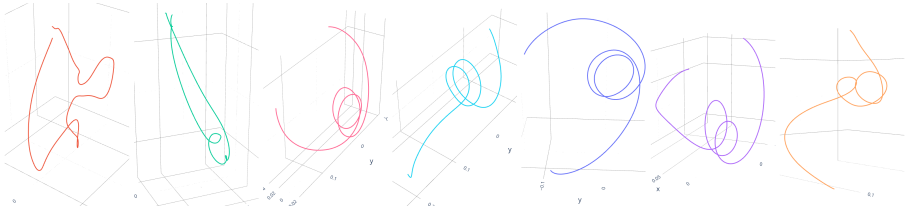


(f) Torsions of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).

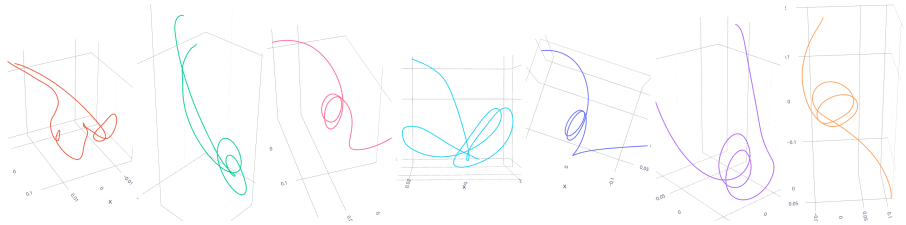
Figure 6.12: Sign "avril".



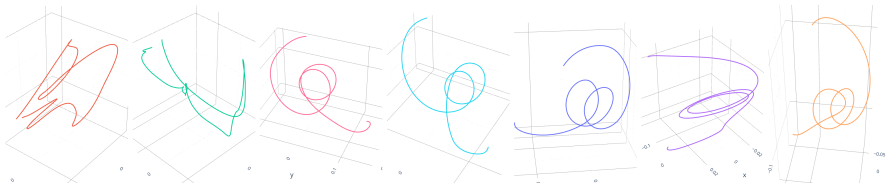
(a) Recorded trajectories of Thomas (first 5) and of Aliza (last 5).



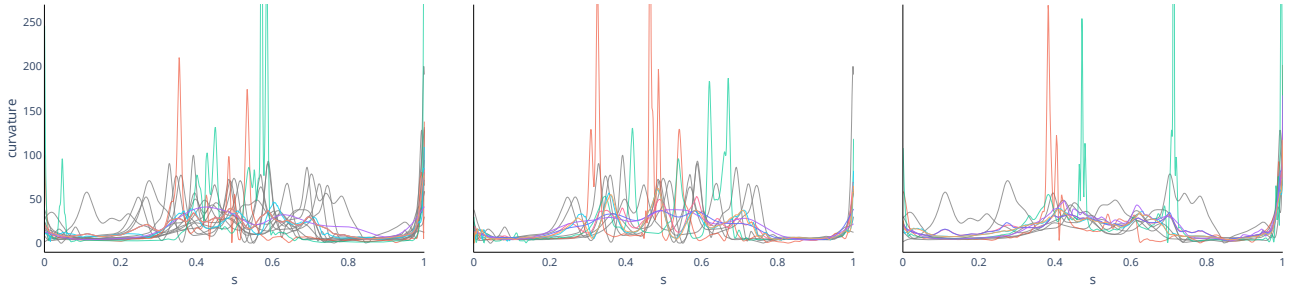
(b) Means on Thomas and Aliza trajectories.



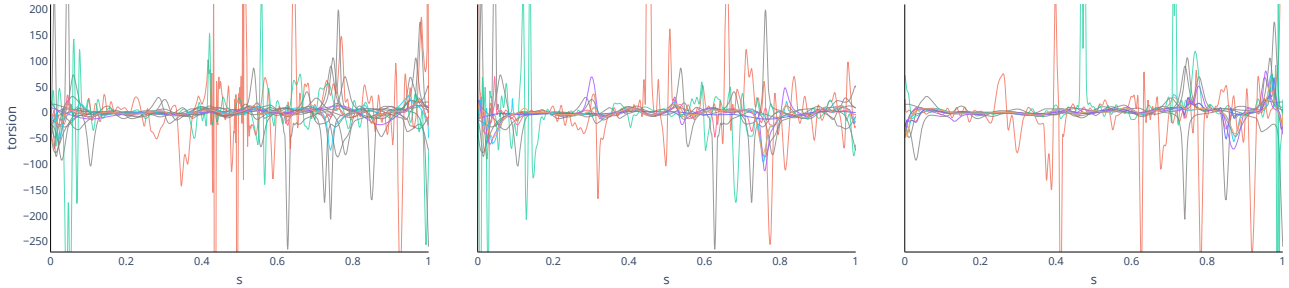
(c) Means on Thomas trajectories.



(d) Means on Aliza trajectories.



(e) Curvatures of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).



(f) Torsions of recorded and means trajectories: Thomas and Aliza (1st), Thomas only (2nd), Aliza only (3rd).

Figure 6.13: Sign "autrefois".

Applying the 7 different definitions and/or methods for computing means on sign language wrist motion trajectories confirms the results and observations obtained from the simulated data while highlighting new challenges or limitations. In most cases, very similar approximation of the FC and SRC Karcher means are obtained with the “preprocessing-based” and “regularization-based” approaches, as seen in Figure 6.9 on the sign "train". Indeed, as the data is almost entirely noise-free, this confirms the results observed with noise-free simulated data.

Furthermore, it is observed that several signs ("train" Figure 6.9, "essayer" Figure 6.11, "autrefois" Figure 6.13) exhibit circular motion trajectories with geometry not far from that of loops or helices used in Chapter 5. This type of geometry often appears in sign language, notably with a possible varying number of spins, which could correspond to nuances in the meaning of the word (such as "very" or "fast"). In these cases, as observed in Chapter 5, Frenet-based means are more coherent for this type of geometry, while Arithmetic and SRVF sometimes yield inconsistent results (see means on Aliza trajectories in Figure 6.13). Thus, this justifies developing these new Frenet-based means for sign language motion analysis. This geometry type also suits the unparametrized Frenet curvatures Karcher mean approximation methods (FC and V1) well due to the relatively flat curvature. In contrast, for less curved geometries such as the "avril" sign (Figure 6.12), the SRVF mean seems quite relevant.

The plots of Frenet curvatures offer another way to judge the quality of means. In some cases, clear patterns are found in the curvatures of captured trajectories that are absent in certain mean Frenet curvatures. Arithmetic and SRVF means, in several cases, exhibit excessively high and pointed curvature or torsion peaks (Figure 6.9, Figure 6.10, Figure 6.13, Figure 6.12). In the case of signs "avril" and "avoir l’air" these plots clearly highlight the limitation of the Frenet curvatures Karcher mean (FC and V1), which smooth curvature peaks due to the absence of alignment. Furthermore, for the "avoir l’air" sign (Figure 6.10), it is observed that the mean V2 is less effective than V3. This could result from less well-posed (non-symmetric) alignment in the V2 criterion. Here, a significant regularization is added for Frenet curvatures alignment in the V2 criterion; otherwise, significant pinching effects occur (see Section 2.2.1 in Chapter 2).

An important aspect to note is that there is sometimes a significant difference in the geometry of the shape between Thomas’s and Aliza’s movements, which is reflected in the mean shapes. There are fairly distinct means across the three sets, and sometimes mean shapes on the union of trajectories from Thomas and Aliza seem suboptimal (e.g., "avril" sign Figure 6.12). This underscores the impact of sign segmentation on mean computations. The issue here is that trajectories not only contain the sign movement but also the return-to-initial-position movement of the hands. Given that Thomas is taller than Aliza, curvature plots clearly show that these sections (at the edges) with nearly flat curvature are longer in Thomas. This justifies the importance of a method with alignment (SRVF, SRC, V2, V3). In several cases, SRC, V2, or V3 means exhibit consistent geometry in the middle but inconsistencies at the edges (curvature too flattened at the direction change to return to the initial position), and thus, the final mean shape seems unreliable. Furthermore, in these cases, SRVF and Arithmetic mean resemble Thomas’s movement more. It appears that they give more importance to the beginning and end of the movement, which spatially occupies more space but does not contain the sign information. In contrast, Frenet-based methods resemble Aliza’s movements (see the sign "avoir l’air" Figure 6.10).

7 Conclusion

In this chapter, we compare different definitions of the mean shape of a set of Euclidean trajectories and various methods for estimating these means. In particular, we highlight the difference between estimation methods inspired by the field of shape analysis and those defined from a more FDA perspective. The former involves two steps: an efficient preprocessing step to compute the considered representations and then the use of the Karcher mean in the corresponding shape space. The latter relies on defining a unique regularized statistical criterion to optimize the mean parameter within a functional data analysis

framework. We show that both approaches are not completely equivalent, especially in the case of noisy data, where the latter limits the bias that individual preprocessing adds in the other case. Regarding the definition of the mean shape itself, results on simulated data, but especially on real data, clearly demonstrate the need for methods including a registration step, such as the SRVF mean, the SRC mean, and V2 and V3 means. Moreover, in the sign language movements considered here, circular geometries are prevalent, for which methods based on Frenet curvatures are particularly relevant. Therefore, we believe that these methods, although more computationally expensive, are better suited to this problem than the SRVF-based method.

However, in some cases on sign language motion trajectories, SRC or V3 methods are not very satisfactory. It should be noted that they depend on the estimation method used for Frenet curvatures. Therefore, an improvement in results could be possible by estimating Frenet curvatures with the EM algorithm developed in Chapter 4, either in the "preprocessing-based" approach or in the "regularization-based" approach. In this latter case, it will require an adapted method involving iterations encompassing tracking and smoothing the Frenet paths with the EM algorithm and estimating the mean parameter with the criterion defined in this chapter. Furthermore, these limitations can also be due to sign segmentation. Therefore, one may question how to obtain segmentation only for the part of the sign that carries the meaning and does not include transitions from one position to another. Only deaf individuals would be capable of performing such segmentation, and even then, the continuous nature of the movement makes the boundary between transition and sign movement unclear. Therefore, an alternative will involve developing methods less sensitive to segmentation, meaning less dependent on precise knowledge of the start and end of each movement. For this purpose, we have thought of considering a partial alignment independent of the start and end during the Frenet curvatures alignment step, inspired by the work introduced in Bryner and Srivastava (2022). These points will be the subjects of future works.

7

Toward Sign Language Processing via Motion Trajectories Analysis

Abstract

In this chapter, we demonstrate applications of the tools and framework for motion trajectory analysis developed in this thesis, based on the Frenet framework, in the context of sign language processing. Firstly, using our robust estimators for curvature and torsion of motion trajectories, we assess the validity of power laws on wrist movement data in LSF. Subsequently, we introduce a generative model for wrist movement in sign language. This model relies on decomposing the variability of a set of trajectories through its mean geometry and non-linear warping functions proposed in Chapter 6. Lastly, we discuss our future research directions in sign language processing within the developed framework, particularly focusing on motion trajectory segmentation and clustering in sign language.

Contents

1 Kinematic Power Laws

Different types of human movement have been studied in neuroscience and motor control theory with the aim of developing a predictive model for voluntary human motion. It is widely accepted that the central nervous system selects specific movements to follow an optimal trajectory, minimizing certain costs. Among various cost functions or optimality laws studied in the literature, power laws relating

movement kinematics to the geometry of the trajectory have been explored (Flash and Berthoz, 2021). Specifically, these laws assert that the movement speed is a power function of the degree of curvature of the trajectory, described by Frenet curvatures. Notably, the 2/3 or 1/3 power laws relate the radius of curvature (inverse of curvature function) or, respectively, directly the curvature to the movement speed. The 2/3 power law, initially demonstrated for "constrained" arm movements (Lacquaniti et al., 1983), has been extensively studied and observed in a wide range of activities (Zago et al., 2018), such as elliptic trajectories in 2D (Viviani and Flash, 1995) and 3D space (Maoz et al., 2009), walking paths (Hicheur et al., 2005), force trajectories, eye movements, as well as in monkey movements (Abeles et al., 2013) and crawling motions of Drosophila larvae (Zago et al., 2016). Consequently, some researchers consider these laws as fundamental biological or kinematic constraints governing how organisms execute curved movements. Deviations from the 2/3 exponent have been observed at specific points of trajectories or for specific shapes. A generalization of this law, known as the 1/6 power law, incorporating the torsion variable, has been proposed by Pollick et al. (2009); Maoz et al. (2009) for three-dimensional movements. As we develop robust and accurate algorithms for curvature, torsion, and speed parameter estimations in this thesis, we explore whether our data from wrist movements in French sign language also follow these empirical laws. We consider here the 1/3 (equivalent to the 2/3 power law) and 1/6 power laws that are directly expressed as functions of the Frenet curvatures and the arc-length first derivative. We recall here their formulations:

$$\dot{s}(t) = C_1 \kappa(s(t))^{-1/3} \qquad \text{(1/3 power law)}, \tag{7.1}$$

$$\dot{s}(t) = C_2 \left(\kappa(s(t))^2 |\tau(s(t))|\right)^{-1/6} \qquad \text{(1/6 power law)}, \tag{7.2}$$

where C_1 and C_2 are constants, $\dot{s}$ is the curvilinear speed and κ and τ the curvature and torsion functions. Before looking at whether our data fit these laws, it is interesting to understand where they are derived from. Let's recall the extrinsic formulas of functional parameters curvature and torsion in a 3-dimensional space:

$$\kappa(s(t)) = \frac{\|\dot{x}(t) \times \ddot{x}(t)\|_2}{\|\dot{x}(t)\|_2^3}, \quad \tau(s(t)) = \frac{\det(\dot{x}(t), \ddot{x}(t), \dddot{x}(t))}{||\dot{x}(t) \times \ddot{x}(t)||_2^2}.$$

Knowing that $\dot{s}(t) = \|\dot{x}(t)\|_2$, we have

$$\dot{s}(t)^3 \kappa(s(t)) = \|\dot{x}(t) \times \ddot{x}(t)\|_2, \tag{7.3}$$

$$\dot{s}(t)^6 \kappa(s(t))^2 |\tau(s(t))| = |\det(\dot{x}(t), \ddot{x}(t), \dddot{x}(t))|. \tag{7.4}$$

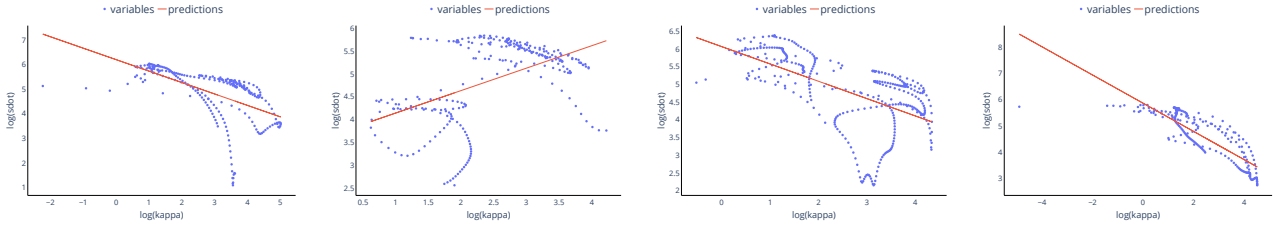
Therefore, we observe from equations (7.1) and (7.3) that the 1/3 power law models the function $t \mapsto \|\dot{x}(t) \times \ddot{x}(t)\|$, representing the area enclosed by the shape, by a constant. This corresponds to the movements often referred to as *constant equi-affine speed* movements. In the same way, we observe from equations (7.2) and (7.4) that the 1/6 power law models the function $t \mapsto |\det(\dot{x}(t), \ddot{x}(t), \dddot{x}(t))|$ by a constant as well. The determinant function models the volume enclosed by the shape, thus generalizing the previous type of movements as the *spatial equi-affine speed* movements in 3-dimensional space (Maoz et al., 2009). Note that this modeling induced by the 1/6 power law is only possible locally, given that the torsion function changes sign and vanishes locally. Basically, these power laws are tested using linear regression analysis to identify the exponent of the power functions that most accurately capture the relation between Frenet curvatures and curvilinear speed at each point along the trajectory. To transform these laws into a linear form, we simply use the logarithm function and then consider the regression equations:

$$\log(\dot{s}(t)) = \log(C_1) + \beta_1 \log(\kappa(s(t))), \tag{7.5}$$

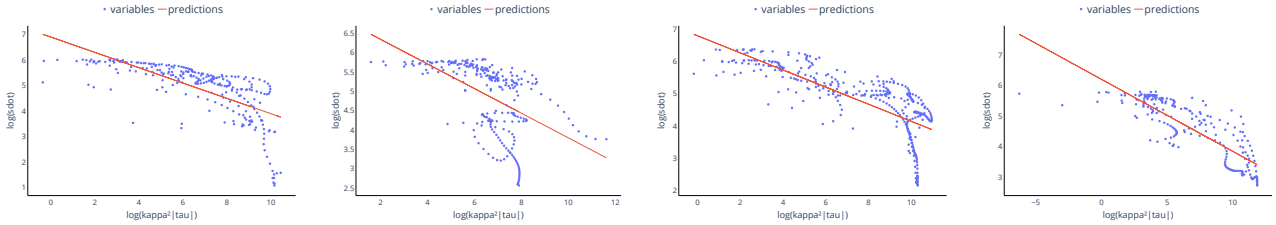
$$\log(\dot{s}(t)) = \log(C_2) + \beta_2 \log\left(\kappa(s(t))^2 |\tau(s(t))|\right). \tag{7.6}$$

	β	$\log(C)$	MSE	R^2
1/3 power law	-0.551 (0.204)	5.908 (0.646)	0.434 (0.250)	0.516 (0.193)
1/6 power law	-0.246 (0.068)	6.265 (0.603)	0.417 (0.231)	0.538 (0.164)

Table 7.1: Linear regressions (7.5) and (7.6) results. Mean and standard deviation in parenthesis of the estimated regression coefficients (β), intercepts ($\log(C)$), mean square errors (MSE), and R-squares (R^2) over all the wrist trajectories of French sign language movements in the data set described in Section 3.1

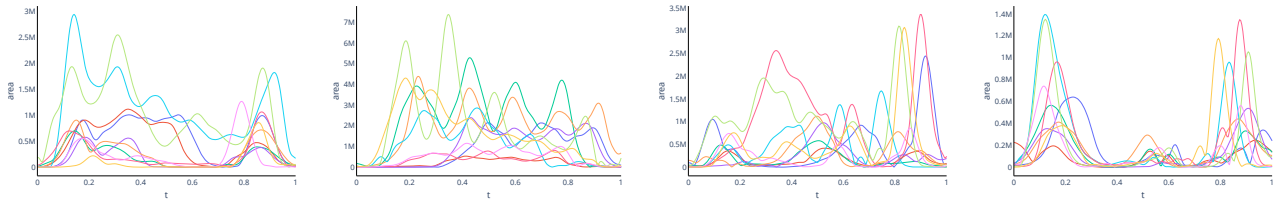


(a) 1/3 power law

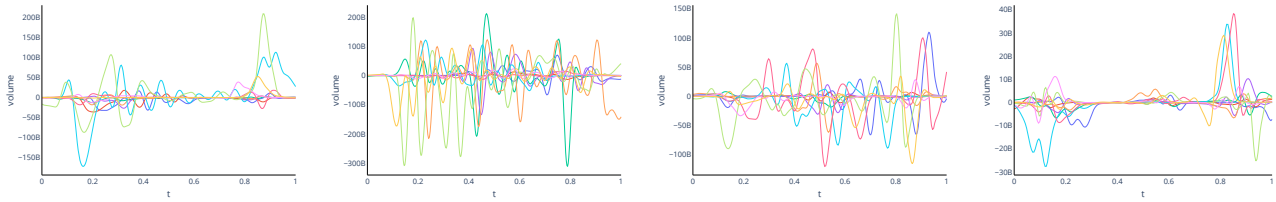


(b) 1/6 power law

Figure 7.1: Plot of the 1/3 power law (first row) and the 1/6 power law (second row) regression results for one trajectory of movement in LSF of signs "dimanche", "train", "avoir l'air", "bus" (left to right).



(a) $t \mapsto \|\dot{x}(t) \times \ddot{x}(t)\|$



(b) $t \mapsto \det(\dot{x}(t), \ddot{x}(t), \dddot{x}(t))$

Figure 7.2: Plot of the functions assumed to be constant, or of little influence, by the 1/3 (first row) and 1/6 (second row) power law models for the several trajectories of movement in LSF of signs "dimanche", "train", "avoir l'air", "bus", going from the left column to the right.

This analysis employs the logarithm of curvature or curvature and torsion values as the predictor variable and the logarithm of curvilinear speed values as the outcome variable.

Based on our estimators, we propose testing the two linear regression models (7.5) and (7.6), corresponding to the 1/3 and 1/6 power law, respectively, on the dataset described in Section 3.1. The results are summarized in Table 7.1. For 4 of these signs, we illustrate the linear regressions in Figure 7.1, and in Figure 7.2, we plot the two functions modeled by constants in each of these laws. The regression coefficients β in each case are far from the expected values according to the laws, with high standard deviations. In the case of the 1/3 power law, Figure 7.1 shows a case where β is positive. Although the 1/6 power law seems to describe the data slightly better than the 1/3 power law, these two linear regression models are clearly not complex enough to capture the relationship between these parameters in our data (results of mean square errors and R-squares in Table 7.1). Furthermore, Figure 7.2 shows that the functions modeled by constants in these laws exhibit significant variations in these data, which seem to depend notably on the sign considered. These observations mean that these variables contain important information about the motion trajectories and their variability and thus must be taken into account. Finally, it appears that these power laws, although observed in various human movements, do not correspond to sign language motions. It should be noted that in recent years, these laws have been controversial, especially by Marken and Shaffer (2017), who characterize them as a "behavioral illusion", resulting from observations forced by mathematical relationships among these parameters. They specifically discuss the impact of omitting descriptive variables corresponding to the functions $t \mapsto \|\dot{x}(t) \times \ddot{x}(t)\|$ and $t \mapsto \det(\dot{x}(t), \ddot{x}(t), \dddot{x}(t))$ can have in these cases. In fact, by fitting these laws with time-varying coefficient functions in place of constants $\log(C_1)$ or $\log(C_2)$, we find slopes closer to 1/3 and 1/6. That being said, Endres et al. (2013) use the 1/3 power law for unsupervised segmentation of wrist movements in Israeli sign language data into motion primitives. They segment a sign into "primitives" by seeking multiple fittings of the law with different coefficients and intercepts along the sign's trajectory. Indeed, this makes more sense looking at the plots of the 1/3 power law regressions in Figure 7.1. Thus, although not sufficient to fully describe these movements, this paper confirms the significance of parameters involved in this law (curvilinear speed and Frenet curvatures) for analyzing human motion. Finally, an interesting perspective would be to consider more complex functional regression models on a broad dataset, aiming to explore a nonlinear relationship between the parameters directly.

2 Generative Model of Wrist Movements

In this thesis, we have developed a framework based on the Frenet-Serret representation, allowing the decomposition of the geometry of a three-dimensional curve into its curvature and torsion functions. In Chapter 6, we present a model for the variability of a set of curves using nonlinear warping functions and average curvature, torsion, and arc length. Inspired by the work of Tucker et al. (2013), where they propose a generative model of scalar functions based on the SRVF representation, we have developed a generative model of sign language wrist movements using the Frenet-based analysis framework introduced in this thesis. For this, we apply principal component analyses to the identified descriptive components of the movement and impose a joint probability model on the principal coefficients of these components. Developing a generative model for sign language wrist trajectories can have several applications and benefits for the field. Firstly, it can augment data and enhance learning processes by generating additional realistic samples, addressing the limitations of sign language datasets. Furthermore, it can contribute to a deeper understanding of underlying models and structures of wrist trajectories in sign language, improving recognition systems and overall comprehension of sign language movements. Additionally, it can be used to synthesize new realistic sign language gestures for animated conversational avatars in entertainment, virtual reality, or education contexts. Finally, it can serve as a means to address the question of the anonymization of sign language movements. An initial version of this generative model was published in the conference article Chassat et al. (2022). Here, we present an enhanced version. In the original article, we used the V2 mean (Chapter 6) and associated warping functions. Since its publication, we have developed methods based on SRC (or V3 mean), which, as shown in the Chapter 6

results, are more robust and well-defined. Therefore, we use the SRC mean and associated alignment functions to develop our model. Additionally, the initial article only modeled the phase variability of Frenet curvatures, not their amplitude variations, as evident in the results. Hence, in this work, we propose a model that also considers this source of variation.

2.1 Variability Separation of Motion Trajectories

A motion trajectory $x : [0, t_f] \to \mathbb{R}^3$ can be broken down into its duration t_f, length l, position in space $Z^{(0)} \in SE(3)$ with a rotation component $Q^{(0)} \in SO(3)$ and a translation component $x^{(0)} \in \mathbb{R}^3$, and intrinsic shape. Describing the shape using its Frenet curvatures $\boldsymbol{\theta}(s) = (\kappa(s), \tau(s))^T$ and the arc-length function $s(t)$ scaled to unit length, we obtain the following curve reconstruction scheme from these components:

1. Solve the Frenet-Serret ODE $Q'(s) = Q(s)A_{\boldsymbol{\theta}}(s)$ with the initial condition $Q(0) = Q^{(0)}$.
2. Reconstruct the trajectory $t \mapsto x(t)$ as

$$x(t) = x^{(0)} + l \int_0^{s(t)} T(u)du \quad \text{for } t \in [0, t_f]. \tag{7.7}$$

Thus, given a set of N motion trajectories $\mathcal{E} = \{x_1, \ldots, x_N\}$, we can decompose the variability into a first "rigid transformation" part, consisting of lengths $\{l_k\}_{k=1,\ldots,N}$, durations $\{t_{f_k}\}_{k=1,\ldots,N}$, and positions in space $\{Z_k^{(0)}\}_{k=1,\ldots,N}$, and a second "intrinsic" part, comprises arc-length functions and Frenet curvatures. Moreover, as introduced in Chapter 5 and Chapter 6, variations in the intrinsic part can be subdivided into variations in parameterizations and variations in residual shapes. Specifically, these components can be described by arc length functions $\{s_k\}_{k=1,\ldots,N}$, aligned Frenet curvatures $\{\tilde{\boldsymbol{\theta}}_k\}_{k=1,\ldots,N}$ (describing "amplitude variability" of the shapes), and warping functions $\{\gamma_k\}_{k=1,\ldots,N}$ used for alignment (describing "phase variability" of the shapes). It's important to note that this decomposition depends inherently on the chosen definition of mean Frenet curvatures, defined in Chapter 6. Indeed, considering means $\mu^{(\boldsymbol{\theta})}$ or $X^{(1)}$ results in $\tilde{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_k$ and $\gamma_k = id$ for all k. However, considering means $\mu^{(SRC)}$, $X^{(3)}$, or $X^{(2)}$ leads to $\boldsymbol{\theta}_k(s) = \tilde{\boldsymbol{\theta}}_k(\gamma_k(s))\gamma_k'(s)$, where the warping function γ_k depends on the chosen mean's definition. Finally, given the mean Frenet curvatures computation algorithms of Chapter 6, the motion trajectories are separated into the following components:

1. a mean Frenet curvatures parameter $\mu_{\boldsymbol{\theta}}$,
2. $\left\{\tilde{\boldsymbol{\theta}}_k\right\}$ set of aligned Frenet curvatures $\subset \mathbb{L}^2([0,1], \mathbb{R}^2)$,
3. $\{\gamma_k\}$ set of optimal warping functions $\subset \text{Diff}_+([0,1])$,
4. $\{s_k\}$ set of scaled arc length functions $\subset \text{Diff}_+([0,1])$,
5. $\left\{Z_k^{(0)}\right\}$ set of positions in $SE(3)$,
6. $\{t_{f_k}\}$ set of durations in $\mathbb{R}$, and,
7. $\{l_k\}$ set of lengths in $\mathbb{R}$.

2.2 Analysis and Modeling of Components

One standard way to analyze variability components is through principal component analysis (PCA), which provides a parsimonious decomposition of the variability (Jolliffe, 2002). Basically, it consists of transforming variables that are correlated into new variables that are decorrelated from each other and

orthogonal, which we call *principal components*. However, the PCA is originally defined for Euclidean data. Among the components describing the variability in our trajectories that we have identified, most are elements of more complex spaces, functional space or Lie group, and thus require adapted versions of PCA for their analysis (see Section 2.1.2 of Chapter 2).

2.2.1 Analysis of Warping Functions

Two of the components, $\{s_k\}$ and $\{\gamma_k\}$, are warping functions in the space $\text{Diff}_+([0,1])$. However, as $\text{Diff}_+([0,1])$ is an infinite-dimensional and nonlinear manifold, we consider instead the space $\Psi([0,1])$ which represents the square root of the derivatives of elements in $\text{Diff}_+([0,1])$, i.e. we represent $\gamma \in \text{Diff}_+([0,1])$ by $\psi = \sqrt{\gamma'} \in \Psi([0,1])$. Any element of $\text{Diff}_+([0,1])$ is uniquely represented by an element of $\Psi([0,1])$: $\gamma(t) = \int_0^t \psi(s)^2 ds$. As already introduced in Chapter 5, $\Psi([0,1])$ is the unit sphere $\mathbb{S}_\infty$ of the Hilbert space $\mathbb{L}^2([0,1],\mathbb{R})$ and a Riemannian manifold equipped with the $\mathbb{L}^2$ metric (Marron et al., 2015; Tucker et al., 2013). The geodesic distance between two elements in $\Psi([0,1])$ is defined in equation (5.14). Therefore, the mean of a set of warping functions can be defined by the Karcher mean in the space $\Psi([0,1])$: given $\psi_1, \ldots, \psi_N \in \Psi([0,1])$,

$$\mu_\psi = \underset{\psi \in \Psi([0,1])}{\arg\min} \sum_{k=1}^{N} d_{\Psi([0,1])}(\psi, \psi_k), \quad \text{and} \quad \mu_\gamma = \int_0^t \mu_\psi(s)^2 ds \in \text{Diff}_+([0,1]). \tag{7.8}$$

This Karcher mean can be computed in practice using the Algorithm 3 define in Chapter 6, where the tangent vectors in step 1 are computed as

$$v_k = \frac{\vartheta_k}{\sin(\vartheta_k)}(\psi_k - \cos(\vartheta_k)\mu_\psi), \quad \text{with} \quad \cos(\vartheta_k) = \langle \mu_\psi, \psi_k \rangle,$$

and the exponential map is given as $\exp_{\mu_\psi}(\epsilon\|\bar{v}\|) = \cos(\epsilon\|\bar{v}\|)\mu_\psi + \sin(\epsilon\|\bar{v}\|)\frac{\bar{v}}{\|\bar{v}\|}$. The tangent vectors $\{v_k\}$ are elements of the tangent space $T_{\mu_\psi}(\Psi([0,1])) = \left\{v \in \mathbb{L}^2 \mid \int_0^1 v(t)\mu_\psi(t)dt = 0\right\}$. Given that $\Psi([0,1])$ is a nonlinear space, direct application of functional principal component analysis (fPCA) is not feasible. Therefore, Tucker et al. (2013) propose to make the analysis in the tangent space at the mean point $T_{\mu_\psi}(\Psi([0,1]))$. Within this tangent space, we define a sample covariance function $(t_1, t_2) \mapsto \frac{1}{N-1}\sum_{k=1}^N v_k(t_1)v_k(t_2)$, which is in practice computed using a finite number of points n, resulting in a $n \times n$ sample covariance matrix K_ψ. Performing singular value decomposition (SVD) on $\boldsymbol{K}_\psi$, i.e., $\boldsymbol{K}_\psi = \boldsymbol{U}_\psi \boldsymbol{\Sigma}_\psi \boldsymbol{V}_\psi^T$, yields the estimated principal components of $\{\psi_k\}$: the p principal directions $\boldsymbol{U}_{\psi,j}$ and the observed principal coefficients $c_{kj} = \langle v_k, U_{\psi,j} \rangle$ for $j = 1, \ldots, p$. The warping functions can be reconstructed from this decomposition as $\tilde{\gamma}_k(s) = \int_0^s \exp_{\mu_\psi}(\|\tilde{v}_k\|)(t)^2 dt$ where $\tilde{v}_k = \sum_{j=1}^p c_{kj}\boldsymbol{U}_{\psi,j}$.

Using this method, we compute the means and perform the functional principal components analysis of the two sets $\{s_k\}$ and $\{\gamma_k\}$, and we denote μ_s and μ_γ their respective means, and $\boldsymbol{c}^{(s)}, \boldsymbol{U}^{(s)}$ and $\boldsymbol{c}^{(\gamma)}, \boldsymbol{U}^{(\gamma)}$ their respective p_s and p_γ principal components.

2.2.2 Analysis of Aligned Functions

The aligned Frenet curvatures $\{\tilde{\boldsymbol{\theta}}_k\}$ are vector-valued functions in $\mathbb{L}^2([0,1],\mathbb{R}^2)$. So, in comparison with Tucker (2018)'s work, we need to analyze a set of bivariate functions observed in practice on a discrete grid of size n. However, as explained in Ramsay and Silverman (2005), computing the principal components analysis of bivariate functions results in computing standard PCA on the concatenated vectors with the two variables, curvature and torsion. Then, we denote the concatenated vectors $f_k = (\tilde{\kappa}_{k,1}, \ldots, \tilde{\kappa}_{k,N}, \tilde{\tau}_{k,1}, \ldots, \tilde{\tau}_{k,N})$ and define their sample covariance operator as

$$\boldsymbol{K}_{\boldsymbol{\theta}} = \frac{1}{N-1}\sum_{k=1}^{N}\left((f_k - f_{\mu_{\boldsymbol{\theta}}})(f_k - f_{\mu_{\boldsymbol{\theta}}})^T\right) \quad \in \mathbb{R}^{n\times n}.$$

Taking the SVD, $\boldsymbol{K}_{\boldsymbol{\theta}} = \boldsymbol{U}_{\boldsymbol{\theta}}\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\boldsymbol{V}_{\boldsymbol{\theta}}^T$, we obtain the estimated principal components of $\{\tilde{\boldsymbol{\theta}}_k\}$: the $p_{\boldsymbol{\theta}}$ principal directions $\boldsymbol{U}_{\boldsymbol{\theta},j}$ and the observed principal coefficients $c_{kj}^{(\boldsymbol{\theta})} = \langle v_k, U_{\boldsymbol{\theta},j} \rangle$ for $j = 1, \ldots, p_{\boldsymbol{\theta}}$.

2.2.3 Analysis of Rigid Transformations

To fully analyze the variability within a set of motion trajectories, we also need to consider variability in positions in space, lengths, and durations. The last two components are simply scalars so that we can perform a standard principal component analysis on vectors $(l_k, t_{f_k}) \in \mathbb{R}^2$. On the other hand, positions $\{Z_k\}$ lie in the non-Euclidean space $SE(3)$, but the standard principal component analysis is only applicable to Euclidean data. In such cases, a method called "tangent PCA" involves projecting points in $SE(3)$ onto the tangent space at the Frechet mean and subsequently conducting a standard PCA of these tangent vectors. Other generalizations of PCA to manifold-valued data are studied in Fletcher et al. (2004); Huckemann et al. (2010). Finally, we obtain two other sets of PCA coefficients: the p_{lt_f} principal coefficients $\boldsymbol{c}^{(lt_f)}$ and the p_Z principal coefficients $\boldsymbol{c}^{(Z)}$.

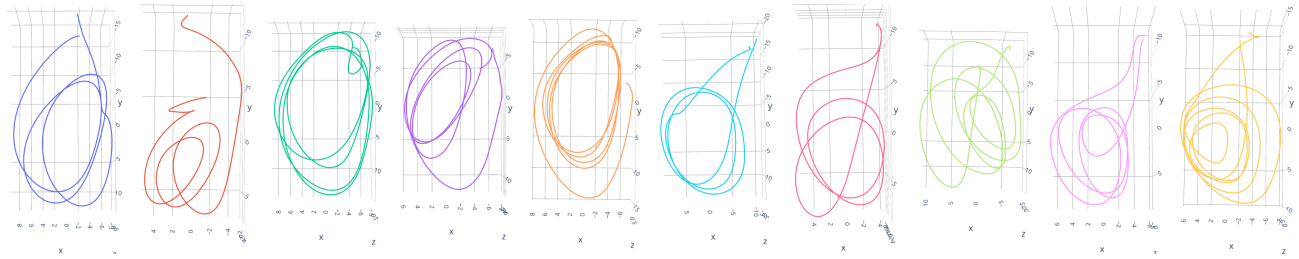
Note that, in practice, the positions $\{Z_k\}$ considered are not the initial positions $x_k(0)$ and initial Frenet frames $Q_k(s(0))$. Indeed, we observed that these quantities are extremely dependent on the estimation errors of the Frenet path and the Frenet curvatures, which are generally not well estimated at the boundaries. Therefore, we consider instead, for the translation part, the position in $\mathbb{R}^3$ of the centers of mass of each curve. For the rotation part, we consider the rotation matrices aligning each curve to the reconstructed mean trajectory, estimated by Procustes analysis.

2.2.4 Joint Gaussian Modeling of Components

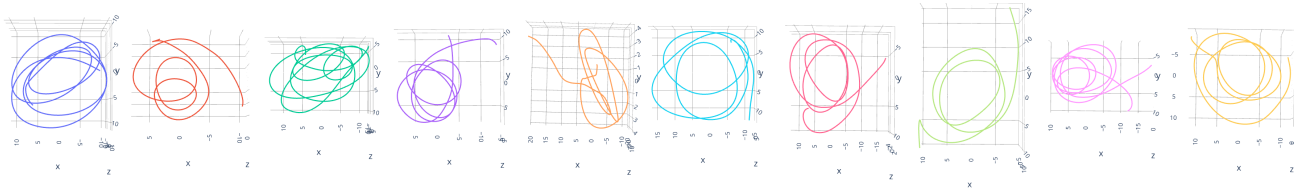
Inspired by the framework used by Tucker et al. (2013) for deriving their generative model, we induce a distribution on the set of motion trajectories by imposing a probability model on the PCA coefficients of the different variability components. Basically, we model the coefficients $\boldsymbol{c}^{(\boldsymbol{\theta})}, \boldsymbol{c}^{(\gamma)}, \boldsymbol{c}^{(s)}, \boldsymbol{c}^{(lt_f)}, \boldsymbol{c}^{(Z)}$ as multivariate normal random variables. Their joint covariance matrix is a square matrix of shape $p_{\boldsymbol{\theta}} + p_\gamma + p_s + p_{lt_f} + p_Z$ and models the correlation between the different variables.

2.3 Results

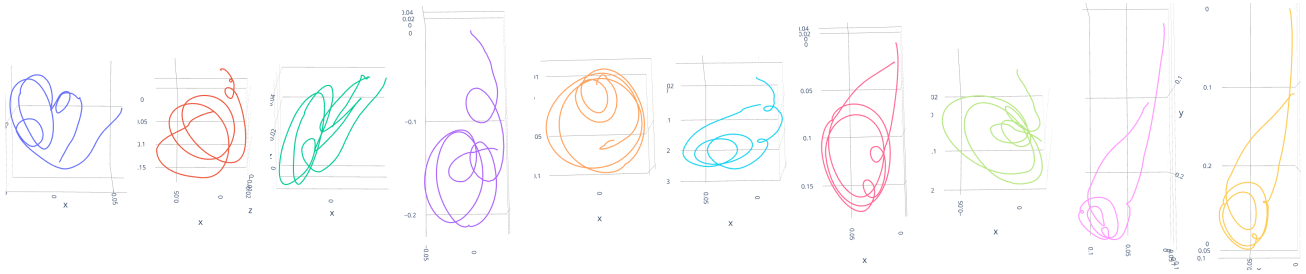
We apply our generative model on two sets of sign language wrist trajectories corresponding to the signs "train" and "essayer," for which the different means are displayed in Figure 6.9 and Figure 6.11, respectively. Finding suitable methods and metrics to assess the quality of the model is not straightforward as we do not have a law or feature to confirm that the movements generated correspond to human gestures. To have an element of comparison, we have also chosen to generate trajectories with a generative model based on the SRVF representation. This model, available in the `fdasrsf` Python library, operates as follows: the mean SRVF is computed from individual SRVFs, and then an fPCA is performed on the tangent space at this mean point. Subsequently, tangent random vectors are generated using a Gaussian model. In this case, only the shapes of the generated curves are compared, as the final length is not modeled in this generative model; therefore, the curves generated by this model are scaled to unit length. The results are visible in Figure 7.3 and Figure 7.4. For each sign, we generate 10 new trajectories with both models. The 3D grids are adapted to each curve and not common to all. Indeed, without this adaptation, it is sometimes challenging to appreciate the shape of certain trajectories.



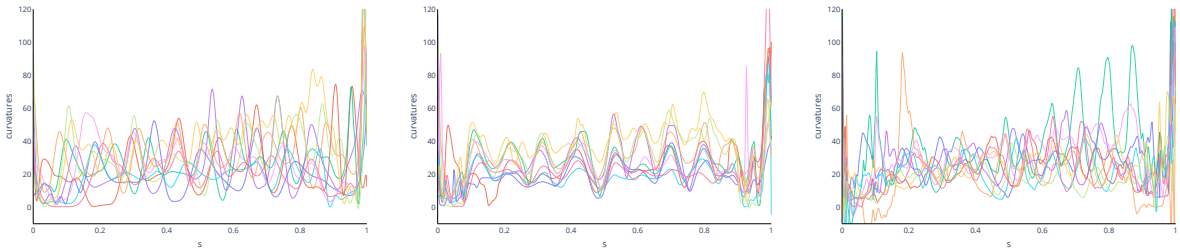
(a) Initial trajectories



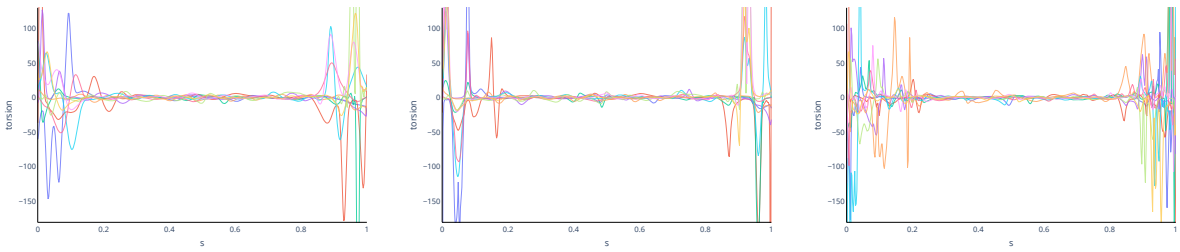
(b) Generated trajectories with our Frenet-based generative model.



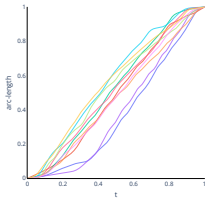
(c) Generated trajectories with the SRVF-based generative model.



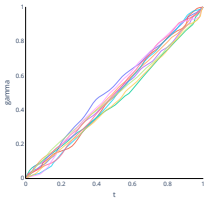
(d) Initial (left), aligned (middle) and generated (right) curvatures.



(e) Initial (left), aligned (middle) and generated (right) torsions.

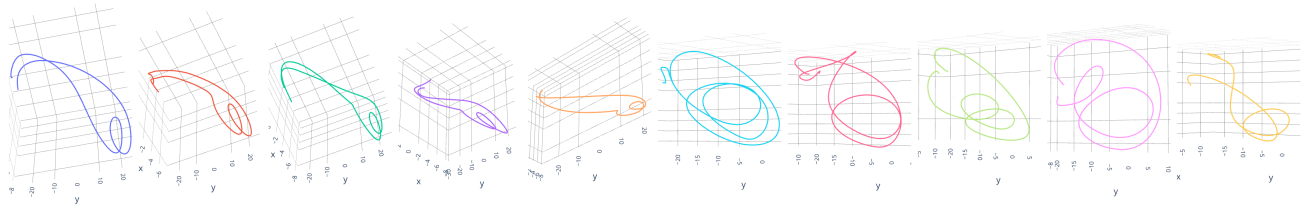


(f) Initial (left) and generated (right) arc-lengths.

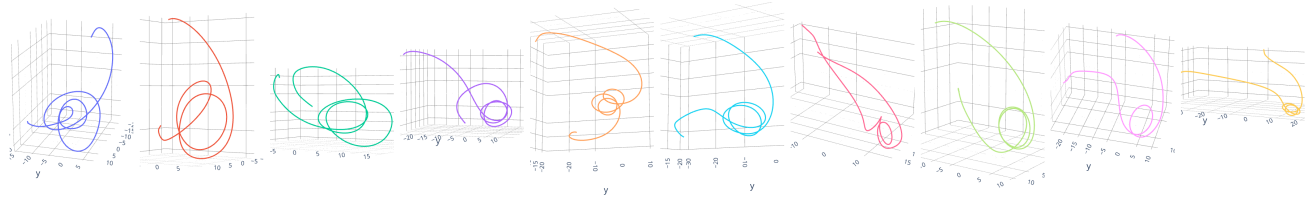


(g) Initial (left) and generated (right) warping functions.

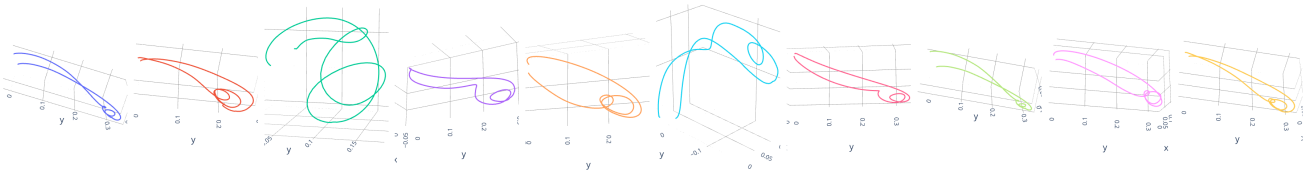
Figure 7.3: Generative model results of sign "train".



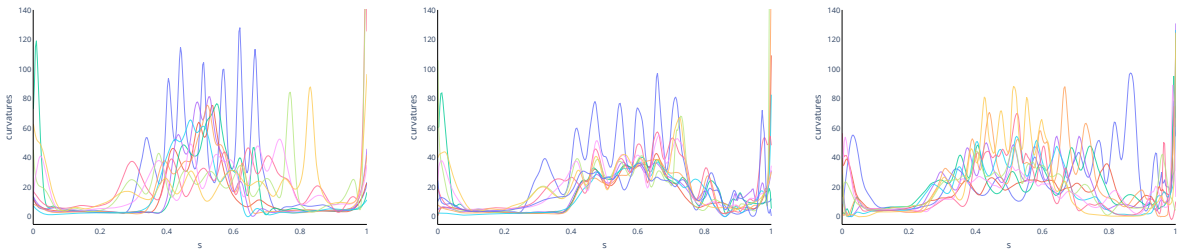
(a) Initial trajectories



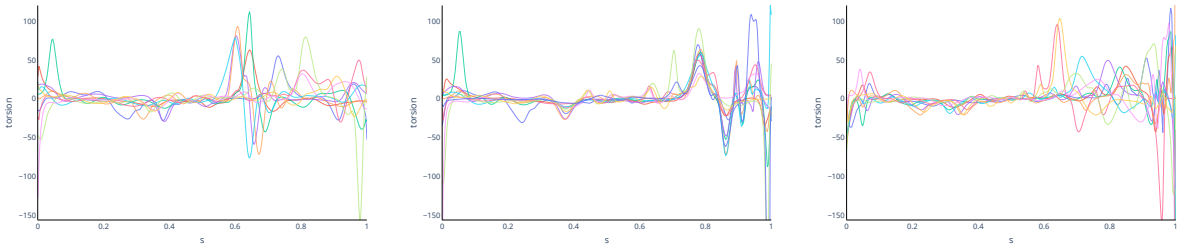
(b) Generated trajectories with our Frenet-based generative model.



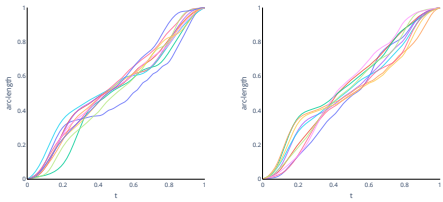
(c) Generated trajectories with the SRVF-based generative model.



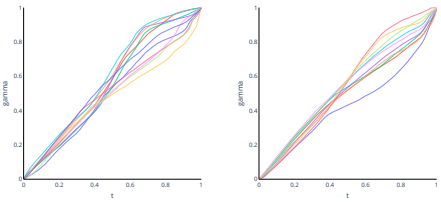
(d) Initial (left), aligned (middle) and generated (right) curvatures.



(e) Initial (left), aligned (middle) and generated (right) torsions.



(f) Initial (left) and generated (right) arc-lengths.



(g) Initial (left) and generated (right) warping functions.

Figure 7.4: Generative model results of sign "essayer".

The results from our model for the sign "train" reveal a faithful reproduction of the circular geometry in the initially captured trajectories, with multiple loops forming circles with radii relatively close to each other. In contrast, the curves generated by the SRVF-based model also exhibit circular loops but with more variable radii, encompassing both very small and large loops. This aligns with the previously observed trends in the mean shapes. The outcomes for curvatures and torsions are reasonably accurate, effectively capturing phase variability. However, some of the generated curvatures and torsions may appear less smooth than those in the initial movements, and the amplitude variability of the curvatures is somewhat attenuated.

For the sign "essayer", the SRVF-based generative model also demonstrates good performance. The simulated trajectories from our generative model better capture the variability of the initial trajectories compared to the SRVF-based model, which predominantly captures the shape of Thomas's movements. However, in the 10 signs generated by our model, the amplitude variability of the curvatures does not entirely match that of the initial signs. This discrepancy can be attributed to the smoother warping functions generated, given that the curvatures of the new trajectories are determined by the model $\tilde{\theta}_k(\gamma_k(s))\gamma_k'(s)$. The observed variations in the generated arc length and warping functions mirror those of the original trajectories. Regarding torsion, the results appear satisfactory in representing the underlying patterns.

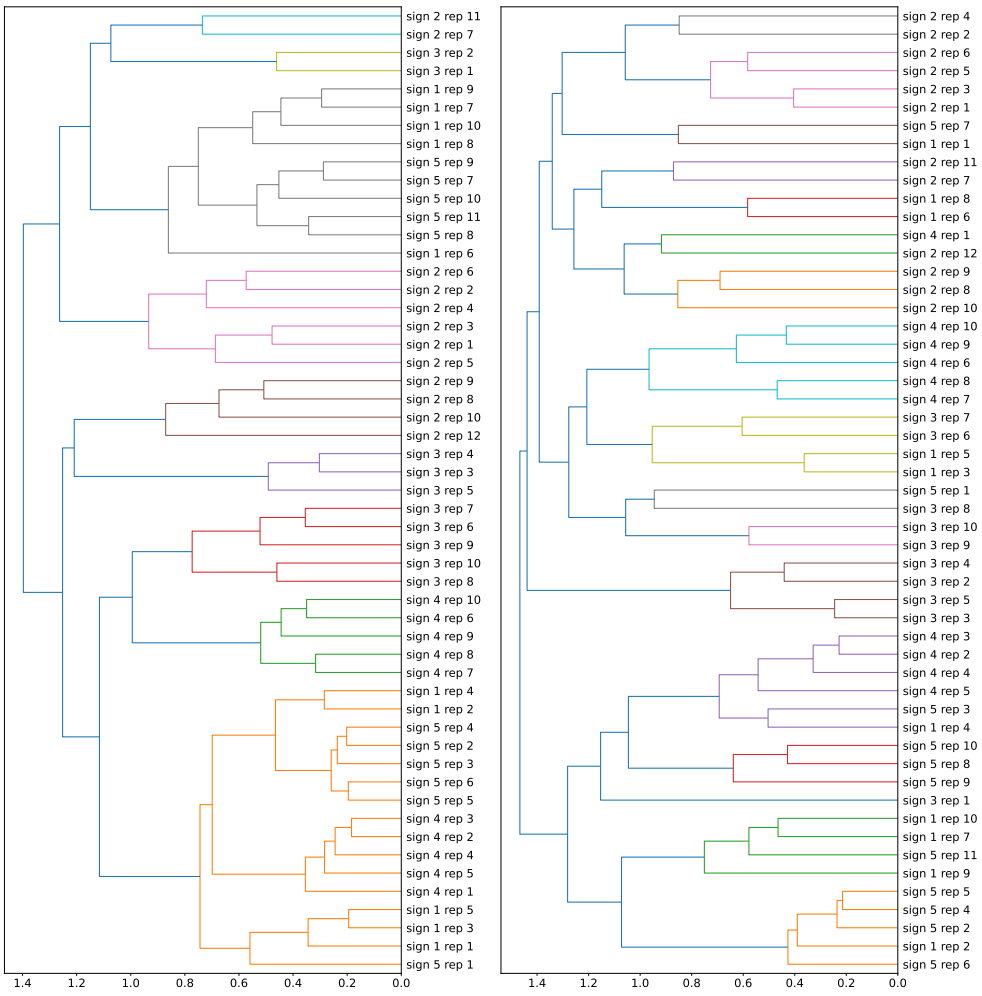
Our model demonstrates promising and encouraging results in the two presented examples. However, accurately evaluating the model's quality remains challenging, relying predominantly on qualitative assessments. To conduct a more comprehensive evaluation, a potential approach involves undertaking a study with a group of deaf individuals and an extensive sign corpus to determine if they can correctly identify signs from the generated movements. However, this study is currently limited as our current approach focuses only on the movements of a single point on the body (in this case, the wrist), whereas sign language involves movements of the entire body and facial expressions. Future work should, therefore, focus on developing a model capable of analyzing the simultaneous movement of multiple points on the body. For this, a simple approach could involve applying the current model independently to each point, but this would likely yield inconsistent results. An alternative strategy could be considering the simultaneous multivariate alignment of curvature and torsion of trajectories of different points on the body (defined during motion capture by the marker placement on the body). However, developing this approach requires additional time and could be the subject of future research work.

3 Exploring Future Works

We plan to explore methods for classifying and clustering signs using the descriptors and distances based on the Frenet framework introduced in this thesis. A subject for an internship was opened last year on this topic but remained unfilled due to a lack of applicants. Nevertheless, it is a subject we intend to explore in our future work. One of the issues we aim to address concerns determining an interesting segmentation for a sign language movement trajectory. Specifically, we seek to identify a limited number of trajectory segments, fewer than the number of signs, which can decompose any sign movement trajectory into a few of these segments. These segments can be regarded as letters in written languages or phonemes in spoken languages, forming building blocks to compose any sign. Therefore, we aim to develop a clustering method that maximizes the length of these segments while minimizing the total number of segments across all signs, i.e., the number of clusters. To achieve this, we could fully exploit both datasets of MocapLab (Section 3.1), which have been only partially used so far.

Regarding segmentation, one approach could be to use motor control principles to obtain potential segmentations. For example, by decomposing the velocity signal into a sum of log-normal profiles, or using multiple regressions of the two-thirds power law, as proposed by Endres et al. (2013), or by

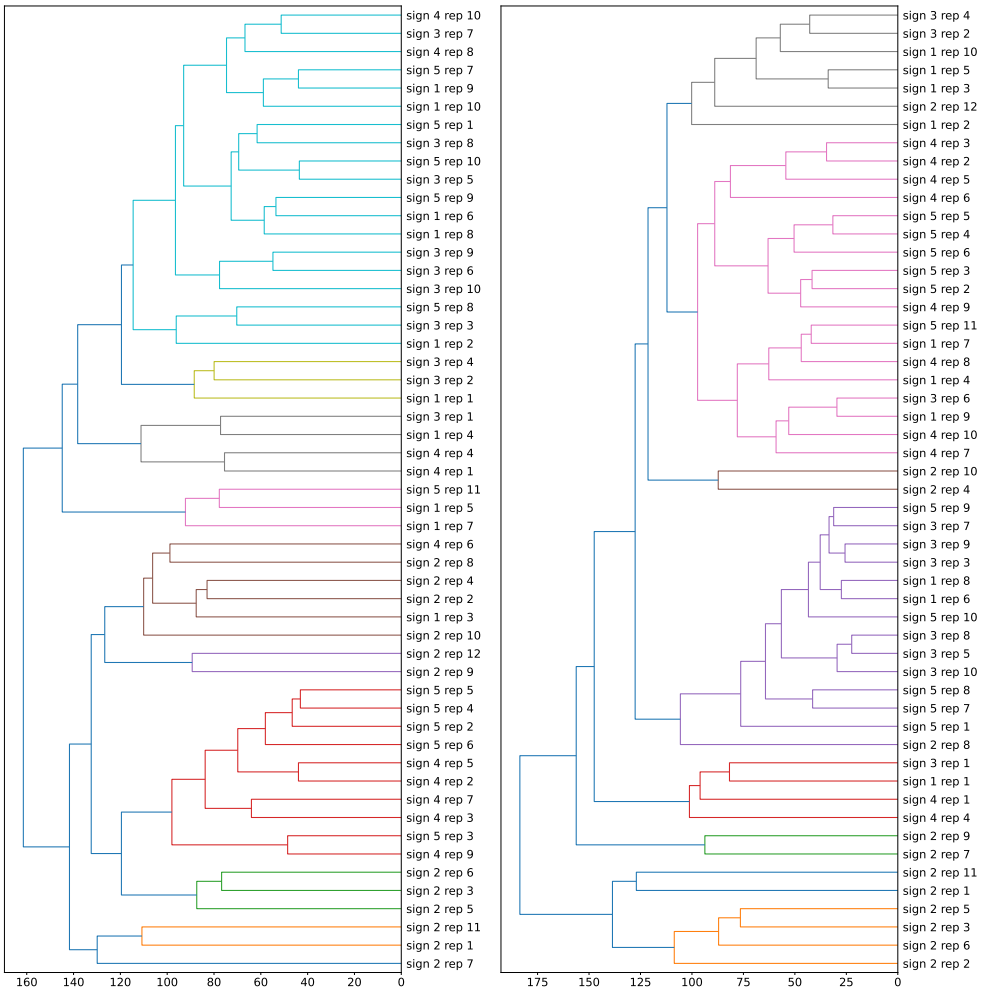
drawing inspiration from the works of Despinoy et al. (2016) who use the idea of persistence to segment the velocity profile. However, initial clustering tests based on our SRC distance suggest that it may not be directly suitable for this task. We applied a simple hierarchical clustering method to the approximately 10 repetitions of the 5 signs, for which we displayed the averages in Chapter 6. Figure 7.5 and Figure 7.6 show the dendrograms of these clusterings considering the entire trajectory and when removing 20% of the time at the beginning and end of the trajectories, using the SRVF and SRC distance respectively. It is observed that the SRVF distance seems more effective for this task, even if it is slightly less performant when removing transitional movements to the initial position at the beginning and end. It appears that the SRVF distance can even sometimes differentiate repetitions by Thomas (rep 1 to 5) from those by Aliza (rep 6 to 10), simply based on wrist movement. This interesting result reveals a limitation of our distance compared to the SRVF distance for this clustering task. Indeed, the SRVF distance has the advantage of always being between 0 and 1, which is not the case for the SRC distance. The SRVF representation of a curve always has a unit $\mathbb{L}^2$ norm, whereas the $\mathbb{L}^2$ norm of the SRC representation corresponds to $\int_0^1 \sqrt{\|\theta(t)\|}dt$, making SRC representations less suitable for direct comparisons.



(a) SRVF on 100%

(b) SRVF on 80%

Figure 7.5: Dendrograms generated through hierarchical clustering using the complete linkage criterion applied to the SRVF pairwise distance matrices on the set of repeated trajectories of signs "autrefois" (1), "avril" (2), "train" (3), "avoir l'air" (4), "essayer" (5): distances taken on 100% of the trajectories and on the 80% in the middle of the trajectories.



(a) SRC on 100%

(b) SRC on 80%

Figure 7.6: Dendrograms generated through hierarchical clustering using the complete linkage criterion applied to the SRC pairwise distance matrices on the set of repeated trajectories of signs "autrefois" (1), "avril" (2), "train" (3), "avoir l'air" (4), "essayer" (5): distances taken on 100% of the trajectories and on the 80% in the middle of the trajectories.

This suggests potential future theoretical work to improve our distance. On the other hand, it also suggests that we could potentially achieve good results with the SRVF distance for this problem.

Conclusion and Perspectives

The primary goal of this thesis, centered on developing tools to analyze sign language motion trajectories, has led us to formulating a new general mathematical framework for functional data shape analysis. Leveraging the power laws of motor control involving the trajectory's curvature, torsion, and velocity parameters, we identified the Frenet-Serret framework as potentially relevant to motion trajectory analysis. The development of a shape analysis framework of Euclidean curves from these parameters had been suggested by some, but it had never been fully explored or defined, primarily due to the challenge of estimating Frenet curvatures from discrete numerical observations of a curve.

In this context, within a functional data modeling framework, this thesis introduces a new EM algorithm for estimating these parameters. Compared to existing methods that inspired its development, our proposed algorithm provides more reliable estimators of Frenet curvatures and Frenet paths. These estimators are both smooth and capable of adapting to very fine geometric variations (e.g., small loops). Nevertheless, the two-step estimation methods, initially proposed by Park and Brunel (2019), for which we provide a comprehensive description and analysis in this thesis, along with an alternative method for smoothing the Frenet path, also prove relevant and robust in the case of low-noise data. These methods outperform the commonly used method based on the extrinsic formulas of curvatures. They estimate Frenet curvatures within a functional data framework with a B-spline decomposition that, however, involves many hyperparameters to optimize and is not effective at the boundaries. Although the EM algorithm is slightly more effective at the boundaries due to simultaneous optimization of the initial condition, this remains a significant problem for the subsequent use of these estimators. We believe that a potential improvement to the EM model would be formulating the problem as a latent force model, especially since it would better correspond to the physical model we consider and our assumption that Frenet curvatures are smooth parameters. However, this would significantly complicate the model formulation, which is already quite complex despite several simplifications, such as restricting it to a 3-dimensional space (and not arbitrary) and assuming a standard model for the error. Thus, extending this model to any dimension seems challenging and potentially not optimal, especially considering the high computational cost of the EM algorithm. Therefore, we believe that an interesting work for future research would be to explore the use of deep neural networks for this task, given that we can simulate a large amount of data (solution of the Frenet-Serret differential equation). This approach could potentially offer increased computational efficiency and even superior performance in terms of results. Nevertheless, the estimation algorithms proposed in this thesis have paved the way for the development of new shape analysis methods.

From the reliable and smooth estimators of the Frenet-Serret framework parameters, this thesis defines a new framework for the shape analysis of smooth curves in $\mathbb{R}^d$ ($d \geq 1$), fully exploiting the geometric information of the curves. Specifically, we introduce two new curve representations based on these parameters: the unparametrized Frenet Curvatures (FC) and the Square Root Curvature Transform (SRC), inspired by the SRV transform of Frenet paths into the group $SO(d)$. These representations establish a Riemannian geometry on the shape space, providing explicit formulas for distances and associated geodesics, invariant under the action of the reparametrization group. This perspective challenges

the field of functional data shape analysis, where the state-of-the-art SRVF representation is considered the most advanced and effective method. The results obtained in this thesis demonstrate that the SRC representation is particularly interesting, combining the strengths of both FC and SRVF. The advantages of using higher-order geometric information, beyond first-order information like SRVF, are evident in geodesics and Karcher means presented in this thesis. Unlike the FC representation but similar to the SRVF representation, the SRC depends on parametrization. Thus, we confirm the importance of considering a parametrization-dependent representation, allowing point alignment along the curve by quotienting the reparametrization group and defining an "elastic" shape analysis framework. However, these three representations are relevant, and the choice of the "best" representation may depend on the context and the type of data/curves under consideration. In particular, SRVF may be more suitable in certain applications due to its simplicity of estimation. From these representations, we can define the mean shape of a set of curves using Karcher means in the associated shape spaces they define. However, this thesis also suggests reconsidering how these mean shapes are calculated. In this regard, we propose to view the problem of estimating the mean shape from a statistical perspective based on the statistical characterization of mean Frenet curvatures proposed by Park and Brunel (2019) in a functional data modeling framework. Given the identified limitations of their statistical criterion for estimating these mean parameters, we propose two alternative criteria that, geometrically, have the same strengths as the SRC Karcher mean. Given the difficulty of Frenet curvatures estimation, this is particularly relevant in this context. These regularization-based criteria to approximate the Karcher means are shown in the thesis to be especially effective in the case of a set of noisy curves, as they reduce the bias and error due to the individual estimations of each Frenet curvatures required in a preprocessing-based approach. However, these criteria could be further improved using the EM algorithm proposed in the thesis to estimate Frenet curvatures and Frenet paths. Additionally, based on results from simulated data and sign language movements, it appears that the main limitation of the Frenet-based representations and means that we define still lies in the quality of Frenet curvatures estimators. This could explain the inconclusive first results obtained in clustering sign language movement trajectories using the SRC distance, where the SRVF distance performs quite well. Another potential advantage of the SRVF representation for this problem is that it always has a unit $\mathbb{L}^2$ norm, unlike the SRC, suggesting further research on our distance and representation. Furthermore, experiments on motion data have also highlighted the interest in future development of methods for estimating a partial mean or partial matching, not assuming that the beginning and end are the same for each curve in the considered set. We have started working on this topic and believe it can be useful in various applications. Recent papers have emerged in this research direction, including Antonsanti et al. (2022) in the context of LDDMM, to which I have contributed.

Finally, these tools offer a new framework for analyzing sign language motion trajectories. Although initially inspired by the power laws of motor control, we demonstrate that these are insufficient to capture the variations in curvature, torsion, and speed parameters in the case of right wrist movements in sign language. This thesis also introduces a generative model for wrist movement trajectories based on the three parameters of curvature, torsion, and speed, rather than the three Cartesian coordinates. This model shows promising results and is particularly interesting, given a potential way of addressing both the challenge of limited sign language databases and the anonymization issue of motion capture-acquired movements. While the proposed applications in this thesis for sign language processing are still limited, we discuss ideas for future research to exploit fully, for this purpose, the new mathematical framework proposed and the MocapLab data at our disposal. In particular, we are considering methods for segmenting and clustering motion trajectories based on these parameters, as well as applying these methods simultaneously to the movements of several points since sign language is not just a movement of the wrist. Finally, we believe that it is essential, for the future development of sign language processing methods, to collaborate closely with deaf people, who alone are aware of the real needs and the quality of the tools being proposed.

Finally, more generally, this type of wrist movement trajectory is becoming increasingly common nowadays, especially with the development of virtual reality (metaverse, virtual reality controllers, etc.). We believe the curve analysis framework developed in this thesis, based on the Frenet-Serret framework, could also be relevant in these contexts. Among other things, the new distances we have proposed, being differentiable, could be employed as loss functions in deep learning models using such data. This could offer advantages in terms of performance and learning efficiency compared to commonly used loss functions. Additionally, considering the use of the curvature, torsion, and speed descriptors we have proposed, rather than simple Cartesian coordinates, to train a neural network could be a relevant approach. This might be particularly advantageous when only the shape of the trajectory is of interest in the considered problem, as these descriptors provide a more meaningful representation of the geometric features of movements, potentially improving the network's ability to learn complex relationships. Finally, we sincerely hope that our open-source implementation of the methods developed in this thesis can contribute to such future research works.

A

Python Package FrenetFDA

Overview

All the methods and algorithms developed and detailed in this thesis have been implemented in the Python package **`FrenetFDA`**, available on GitHub (https://github.com/perrinechassat/FrenetFDA). This package combines various methods for estimating parameters within the Frenet framework for Euclidean curves and shape analysis methods based on these parameters. Some methods are specifically designed for 3-dimensional curves, which, nevertheless, cover the majority of real-world data cases where these methods can be useful. The implementation relies on several libraries. B-spline smoothing uses the `scikit-fda` library, and polynomial regression smoothing uses the `scikit-learn` library. Certain methods for Special Euclidean and Special Orthogonal Lie groups, as well as tangent PCA on these groups, are implemented with the `Geomstats` library. All methods related to the SRVF representation of a curve use the `fdasrsf` library. Additionally, curve alignment is implemented using dynamic programming (Bertsekas, 1995) in C++. To achieve this, we slightly modified the implementation found in the `fdasrsf` library to adapt it to the group action $(f \circ \gamma)\gamma'$. Bayesian optimizations are performed using the `scikit-optimize` library. For visualization tools, we use the `plotly` library. Finally, we frequently employ the `scipy` and `numpy` libraries. Since, in most of the experiments conducted, we parallelize various simulations, most functions do not contain directly parallelized sections. Therefore, there is room for code optimization in this regard.

Architecture

`FrenetFDA/`

- **`processing_Euclidean_curve/`**
  - **`unified_estimate_Frenet_state_space/`**
    - **`EM_Frenet_state_space.py`** ▷ Expectation-maximization algorithm for Frenet curvatures and Frenet path estimation (Chapter 4).
    - **`iek_filter_smoother_Frenet_state.py`** ▷ Invariant extended Kalman filter for tracking and smoothing of the Frenet path in $SE(3)$ (Chapter 4).
  - **`estimate_Frenet_curvatures.py`** ▷ Frenet curvatures extrinsic formulas (Chapter 3).

estimate_Frenet_path.py ▷ Gram-Scmidth orthogonalization and constrained local polynomial regression to estimate the Frenet path (Chapter 3).

preprocessing.py ▷ Local polynomial regression to estimate time derivatives and arc-length function (Chapter 3).

processing_Frenet_path/

estimate_Frenet_curvatures.py ▷ Penalized least square criterion to estimate Frenet curvatures given Frenet path observations (Chapter 3).

smoothing.py ▷ Karcher mean based smoother and tracking based smoother of a Frenet path (Chapter 3).

shape_analysis/

generative_model.py ▷ Generative model proposed for motion trajectories in Chapter 7.

riemannian_geometries.py ▷ Classes SRVF, SRC, and Frenet_curvatures including all shape analysis methods (distance, Karcher mean, geodesic path, etc.) (Chapter 5).

statistical_mean_shape.py ▷ Statistical criteria for mean shape estimation (Chapter 6).

utils/

Lie_group/

SE3_utils.py ▷ Methods related to the Special Euclidean group $SE(3)$ (Chapter 2).

SO3_utils.py ▷ Methods related to the Special Orthogonal group $SO(3)$ (Chapter 2).

Frenet_Serret_utils.py ▷ Functions to solve the Frenet-Serret differential equation.

alignement_utils.py ▷ Alignment algorithms based on group actions and parameters considered (Chapter 5 and Chapter 6).

curves_utils.py ▷ Tools for centering and rotating Euclidean curves.

smoothing_utils.py ▷ B-spline and local polynomial smoothing algorithms.

visualization.py ▷ Tools for plotting 2D and 3D curves.

B

Introduction en Français

Cette thèse se consacre à l'analyse mathématique du mouvement de la langue des signes à partir de données de capture de mouvement (motion capture). MocapLab (https://www.mocaplab.com/fr), une entreprise parisienne spécialisée dans la capture de mouvement, est l'une des principales entités européennes dans ce domaine. Leur portfolio diversifié regroupe plusieurs activités de recherche, se concentrant particulièrement sur la langue des signes explorée à travers plusieurs projets (SignEveil, StorySign, Rosetta LIMSI and LISN (2022)). Les données générées par MocapLab peuvent être considérées comme une collection de courbes euclidiennes multivariées. Ce type de données, correspondant à l'enregistrement de comportements ou de phénomènes dans le temps ou l'espace, peut être vu comme des observations d'une fonction. Il est devenu de plus en plus courant ces dernières années dans divers domaines (trajectoires d'avions, études de mouvement, biologie, enregistrement continue de la température, etc.), notamment en raison des avancées dans les technologies d'acquisition de données continues (capteurs, dispositifs embarqués, systèmes de capture de mouvement, etc.). L'analyse de ces données complexes nécessite des outils mathématiques sophistiqués et spécifiques, étant donné leur dimensionnalité élevée inhérente et leur nature continue. Ces défis ont motivé la recherche en *analyse de données fonctionnelles* (Ramsay and Silverman, 2005), visant à établir un cadre dédié pour les données modélisées sous forme de fonctions, et en *analyse de forme* (Srivastava and Klassen, 2016), qui cherche à développer des outils statistiques capables de capturer les formes complexes inhérentes à ce type de données, en utilisant des espaces mathématiques appropriés. L'engagement de MocapLab dans la recherche sur la langue des signes, associé à l'expertise du Laboratoire de Mathématiques et de Modélisation d'Évry (LaMME), a conduit à leur collaboration à travers cette thèse, généreusement soutenue par des subventions de la Région Île-de-France dans le cadre du projet "Paris Region PhD 2020". L'objectif de la thèse est de contribuer de manière significative à l'évolution des domaines de l'analyse de données fonctionnelles et de l'analyse de forme, dans un contexte motivé par l'analyse du mouvement de la langue des signes.

1 Contexte de la Langue des Signes

Les Langues des Signes (LS) constituent le principal moyen de communication pour environ 70 millions de personnes sourdes dans le monde, selon la Fédération Mondiale des Sourds (https://wfdeaf.org/our-work/). Ces langues comptent plus de 200 variations à travers le monde, en fonction des pays. En France, la Langue des Signes Française (LSF) est prédominante et considérée comme la langue principale pour de nombreuses personnes sourdes. Historiquement, les langues des signes ont eu du mal à être reconnues comme des langues indépendantes. Pendant l'Antiquité, la notion aristotélicienne selon laquelle ceux qui

ne peuvent pas parler ne peuvent pas penser entravait l'éducation des personnes sourdes. Ce n'est qu'au XVIe siècle que les éducateurs ont commencé à travailler avec les personnes sourdes, et en 1760, l'abbé de L'Épée a fondé la première école pour les sourds à Paris. Le Congrès International sur l'Éducation des Sourds de 1880 a marqué un tournant en favorisant les méthodes orales plutôt que la langue des signes dans l'éducation. Cette interdiction des langues des signes a persisté jusqu'à la fin du XXe siècle. Le travail novateur de Stokoe (1960) a entraîné un changement de perception et a stimulé des recherches approfondies sur les langues des signes. Cela a conduit la France à reconnaître la LSF comme une langue à part entière dans la loi française sur le handicap de 2005. Par la suite, la recherche linguistique, les programmes universitaires et la formation professionnelle pour les professions liées à la langue des signes se sont multipliés, mettant l'accent sur l'importance de l'accessibilité dans les espaces publics pour les personnes en situation de handicap. Aujourd'hui, malgré les progrès, des croyances obsolètes persistent, contribuant à la négligence linguistique et impactant le développement cognitif, socio-émotionnel et académique des personnes sourdes.

1.1 Linguistique de la Langue des Signes

Tout comme les langues parlées, les langues des signes sont régies par un système linguistique englobant des structures syntaxiques, morphologiques, sémantiques et phonologiques (Emmorey, 2001). Les LS sont des langues orales impliquant une communication en face-à-face : un signeur transmet un message en LS, et un observateur perçoit ce message (Figure B.1). Les LS utilisent la modalité visuo-gestuelle, impliquant des mouvements continus du visage, des mains, du corps du signeur et de l'espace environnant pour construire un discours. De plus, les signeurs incorporent l'iconicité, caractérisée par une forte ressemblance entre la forme des signes et leurs significations (Sallandre and Cuxac, 2002). Cela permet l'expression d'idées complexes sans recourir à un lexique standard.

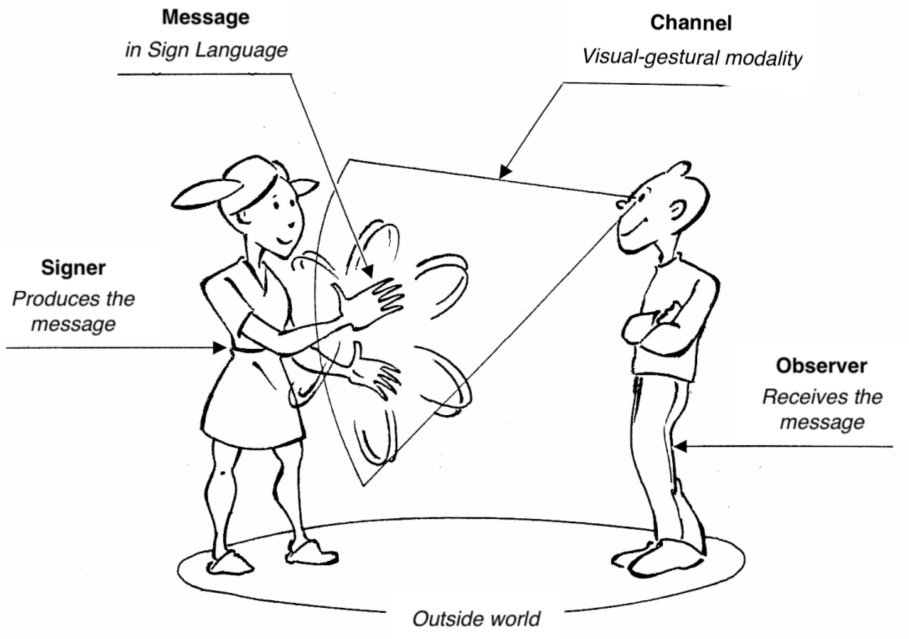


Figure B.1: Schéma de la Langue des Signes (illustration de Laurent Verlaine, Guitteny and Verlaine (2018)).

Récemment, Yin et al. (2021) a proposé une description détaillée des caractéristiques linguistiques de la langue des signes. Parmi ces caractéristiques, la *simultanéité* permet de transmettre autant d'informations qu'une langue parlée dans un laps de temps similaire, en utilisant plusieurs éléments

visuels pour transmettre différentes informations simultanément (Sandler, 2012). Alors que les composants manuels sont la principale source d'information émise, les expressions faciales sont cruciales pour dissiper certaines ambiguïtés de signes et/ou fournir des qualifications (par exemple, grand, petit). Un autre aspect distinctif de la langue des signes implique le placement d'éléments de discours à des emplacements spécifiques dans l'espace de signe. Pour rappeler ces éléments plus tard, les signeurs désignent leur position avec un regard ou un geste de pointage, un processus connu sous le nom de *référencement*, qui peut induire une modification dans le mouvement du signe en raison de changements de direction (par exemple, verbes directionnels) (Dudis, 2004). De plus, les signeurs utilisent l'*épellation par gestes manuels* pour indiquer des noms, des lieux ou des concepts nouveaux de la langue parlée qui n'ont pas de signe dédié (Wilcox, 1992). Cela implique un ensemble de gestes manuels correspondant à l'orthographe écrite. Enfin, le modèle proposé par Brentari (1998) décompose les propriétés articulatoires de la langue des signes en quatre classes phonémiques principales : la forme de la main, l'emplacement, le mouvement et l'orientation de la paume. Des caractéristiques non manuelles, telles que le regard, les mouvements de la tête et la position du torse, peuvent également être incluses (Johnson and Liddell, 2011). Cependant, il est essentiel de noter qu'il n'y a toujours pas de consensus parmi les linguistes concernant la description des langues des signes, et même le concept de signes peut varier en fonction des théories linguistiques.

1.2 Défis Actuels dans le Traitement de la Langue des Signes

La reconnaissance tardive des langues des signes en tant que langues officielles, associée à leur complexité linguistique et à leurs différences conceptuelles significatives par rapport aux langues parlées, a conduit à un retard considérable dans la compréhension des mécanismes sous-jacents et des méthodes de traitement en LS. Ce retard est évident par rapport aux connaissances et aux capacités de traitement des langues parlées et écrites. Aujourd'hui, la plupart des nouveaux outils de communication ou de traitement sont conçus pour les langues parlées. Des progrès significatifs, avec un impact considérable, ont été réalisés dans ces domaines, tels que la reconnaissance vocale et les assistants vocaux (*Siri*, *Google Assistant*), la traduction automatique entre langues (*DeepL*, *Google Translate*), et plus récemment, dans la génération automatique de texte (*ChatGPT*, *Bard*). Cependant, malgré le développement récent d'applications prometteuses pour les LS, comme *Signily*, *Spread Signs*, *SignEveil* (par MocapLab), et d'autres, des efforts supplémentaires substantiels sont nécessaires pour développer des outils pour les LS comparables à ceux disponibles aujourd'hui pour les langues parlées. Il y a un défi social réel et urgent dans le développement d'outils de traitement automatique pour les LS afin de réduire les barrières de communication auxquelles sont confrontés les signeurs et la discrimination causée par ce retard dans les avancées technologiques. La recherche sur les LS est relativement récente et limitée, se concentrant principalement sur la linguistique. Cependant, l'innovation dans les technologies des LS nécessite la participation d'autres domaines tels que l'informatique, les statistiques et les mathématiques.

Aujourd'hui, de nombreux défis entravent encore le progrès dans le traitement automatique des LS. Un défi majeur réside dans les méthodes de représentation. L'absence d'une forme écrite largement adoptée pour les LS conduit à l'utilisation de diverses modalités pour aborder le problème du traitement de la langue des signes. La plupart des travaux de recherche en LS, en particulier ceux basés sur des méthodes d'apprentissage profond, utilisent des données vidéo de signeurs (Rastgoo et al., 2021). Bien que ce soit la méthode de représentation la plus directe, sa grande dimensionalité rend le stockage, la transmission et l'encodage coûteux. De plus, l'anonymisation des vidéos (où le visage du signeur apparaît) reste un problème ouvert (Isard, 2020). D'autres études de recherche visent à développer des systèmes de notation écrite représentant les signes comme des caractéristiques visuelles discrètes de manière linéaire ou en deux dimensions. Des systèmes de notation universels tels que SignWriting (Sutton, 2014) et HamNoSys (Prillwitz and Zienert, 1990) existent, mais aucun n'a été largement adopté par la communauté des langues des signes. De plus, parmi les principales classes phonétiques de la langue des signes (forme de la main, emplacement, mouvement et orientation de la paume), représenter

le composant mouvement dans l'écriture reste un problème ouvert. Une autre représentation, souvent appelée *poses*, consiste en un squelette, une maille ou un ensemble de points dans l'espace représentant les emplacements d'articulation du signeur. Ces positions dans le temps peuvent être acquises avec un système de capture de mouvement, généralement coûteux, ou estimées directement à partir de vidéos (Cao et al., 2019; Wang et al., 2023) avec une qualité réduite et une perte potentielle d'informations. La représentation de la pose est généralement plus fiable et naturelle que les vidéos, permettant une analyse plus précise du mouvement. Cependant, elle implique un format continu et multidimensionnel qui n'est pas trivial à analyser et ne convient pas bien aux algorithmes actuels de traitement du langage naturel (NLP). De plus, même avec cette représentation, la question de l'anonymisation des données se pose. Tout comme pour les langues parlées, où les caractéristiques vocales peuvent révéler l'identité du locuteur, l'identité d'un signeur peut être discernable à travers ses mouvements, mais peu d'informations sont disponibles sur les caractéristiques du mouvement qui le caractérisent et sur la manière dont elles peuvent être manipulées (Bigand, 2022).

Le traitement automatique des LS implique le développement de méthodes pour des tâches typiques d'apprentissage automatique : segmentation, classification, reconnaissance, traduction et génération (Bragg et al., 2019; Yin et al., 2021). La segmentation en langue des signes consiste à détecter des unités significatives ou à identifier des sous-entités composant le signal (similaire aux phonèmes dans les langues parlées) (Santemiz et al., 2009; Farag and Brock, 2019; Bull et al., 2020). La reconnaissance consiste à associer une étiquette à un signe et diffère de la traduction, qui fait référence à la transition de la langue des signes à la langue parlée. Avec les progrès récents dans l'apprentissage profond, de nombreux efforts de recherche ont été menés pour ces tâches, utilisant principalement des données vidéo (Camgöz et al., 2020; Wadhawan and Kumar, 2020). Pour la génération de la langue des signes, la plupart des travaux utilisent des poses comme représentation. L'objectif ici est d'animer automatiquement des avatars, qui peuvent servir d'équivalents aux assistants vocaux pour les LS. Dans les méthodes existantes pour cette tâche, certaines génèrent et concatènent des signes isolés (Stoll et al., 2020) en utilisant la traduction automatique neuronale et des réseaux génératifs antagonistes, tandis que d'autres (Saunders et al., 2020b,a; Xiao et al., 2020) décodent de manière autonome une séquence à l'aide de transformateurs ou de réseaux neuronaux récurrents. Cependant, évaluer la qualité de ces modèles génératifs reste un défi. De plus, une limitation majeure pour le développement ultérieur de méthodes plus efficaces est le manque de données disponibles pour toutes les différentes représentations de la LS. Rastgoo et al. (2021) présentent une liste complète des ensembles de données existants pour la recherche sur la LS, la plupart étant conçus pour la classification des signes. À des fins de traduction, les corpus de signes continus disponibles contiennent beaucoup moins de paires de phrases que des ensembles de données similaires pour les langues parlées (Arivazhagan et al., 2019). De plus, de nombreux ensembles de données en LS discutés dans la littérature ne sont pas accessibles ou sont soumis à des restrictions strictes et des licences, principalement en raison du défi persistant d'anonymiser ce type de données. Par conséquent, développer des méthodes de traitement qui peuvent fonctionner efficacement avec des données limitées est particulièrement intéressant dans ce domaine. Enfin, une limitation commune dans le développement de modèles pour chacune de ces tâches, soulignée par Yin et al. (2021), est le manque d'une méthode de tokenisation efficace et normalisée pour la langue des signes. Il est donc crucial, avant tout, de mieux comprendre les propriétés et la structure complexe de la langue des signes.

2 Analyse des Trajectoires de Mouvement en Langue des Signes

Au cœur de cette thèse, notre exploration se concentre sur l'analyse du mouvement au sein des quatre classes phonémiques en LS (forme de la main, emplacement, mouvement et orientation de la paume). Notre objectif central est de permettre une analyse approfondie de ce qui pourrait être assimilé à un "signal de mouvement" en langue des signes, par analogie avec le signal auditif des langues parlées.

Nous cherchons à définir un cadre mathématique pertinent pour identifier les primitives, les descripteurs et autres "traits cinétiques" du mouvement en LS, en mettant particulièrement l'accent sur les mouvements des mains ou du poignet, qui constituent la principale source d'information en LS, bien que non exhaustive. L'objectif global de notre recherche est de révéler les mécanismes sous-jacents du mouvement humain en LS, améliorant ainsi notre compréhension actuelle de cette langue gestuelle. Cette approche est cruciale dans le contexte du traitement de la LS. Par exemple, dans la création d'animations d'avatars authentiques, étant donné que les êtres humains sont particulièrement sensibles au mouvement biologique et peuvent naturellement faire la distinction entre le mouvement humain et le mouvement synthétisé. De plus, dans le contexte de la transcription écrite de la LS, il est impératif de décomposer le composant mouvement en primitives ou éléments descriptifs. Ainsi, le rôle central de cette thèse est de développer des outils mathématiques et un cadre d'analyse spécifiquement adaptés à cette tâche, contribuant à combler les lacunes actuelles dans la compréhension du mouvement en langue des signes. À cet égard, nous avons la chance de travailler avec des données acquises par capture de mouvement, particulièrement bien adaptées à notre objectif parmi les différents types de représentation de la LS mentionnés précédemment.

2.1 Données de Capture de Mouvement

La capture de mouvement, ou mocap, est une technologie révolutionnaire permettant d'enregistrer et de reproduire les mouvements humains ou d'autres objets à l'aide de divers capteurs et/ou systèmes de caméras. Pour les mouvements humains, la mocap consiste à suivre les trajectoires de points clés sur le corps en mouvement au fil du temps, traduisant directement ces mouvements en une représentation numérique interprétative (vecteurs temporels en trois dimensions). Pendant plus d'un siècle, la mocap a évolué des techniques basées sur des images telles que la chronophotographie ou le rotoscopie pour se concentrer désormais sur des approches avancées telles que les systèmes optiques, mécaniques et inertiels.

Figure B.2: Système de capture de mouvement de MocapLab et espace d'acquisition des mouvements en langue des signes.

Le système de mocap utilisé par MocapLab est un système optique passif composé de 50 caméras Vicon dans un volume de 20 m^3 à 60 m^3 (Figure B.2). Les systèmes de capture de mouvement optiques basés sur des marqueurs (Vicon, Inc, Optitrack, Inc) sont considérés comme la référence pour l'analyse du mouvement humain. Ils utilisent des marqueurs rétro-réfléchissants placés sur des objets ou le corps en mouvement. Des caméras infrarouges enregistrent ensuite la position de ces marqueurs en réfléchissant la lumière infrarouge qu'ils émettent. La triangulation entre les caméras reconstruit

ensuite des trajectoires tridimensionnelles avec une précision extrêmement élevée - théoriquement avec une précision sub-millimétrique en utilisant le système de MocapLab. En raison de cette haute précision, ce type de système de mocap est un outil puissant pour créer du contenu virtuel réaliste et analyser le mouvement humain. En particulier, cela a permis à MocapLab de participer à des projets dans divers domaines, notamment l'animation d'avatars pour l'industrie cinématographique et les jeux vidéo (voir https://www.mocaplab.com/fr/gallerie), la recherche en biomécanique et en médecine, et la linguistique des langues des signes (LIMSI and LISN, 2022; Gibet et al., 2015). Ils sont notamment capables de mesurer simultanément le corps, le visage, la direction du regard et les doigts, tout en entravant au minimum les gestes du signeur.

Figure B.3: Le signeur en capture de mouvement et son avatar (trajectoire dessinée par le poignet mise en évidence en jaune).

L'objectif principal de la mocap est de capturer des mouvements de manière précise pour les reproduire de manière réaliste dans des contextes virtuels ou d'étude (Figure B.3). Cependant, bien que les systèmes optiques passifs offrent une précision spatiale exceptionnelle, ils posent également des défis et des limitations techniques. Des problèmes tels que les occultations, les interférences lumineuses et la confusion des marqueurs peuvent entraîner des données manquantes ou inexactes, nécessitant souvent un environnement contrôlé avec une configuration de caméra spécifique pour garantir des performances optimales. De plus, l'achat et l'utilisation de systèmes mocap sont coûteux, limitant la possibilité d'obtenir une grande quantité de données pour des applications telles que l'entraînement de modèles d'apprentissage profond. Bien qu'il existe des alternatives sans marqueurs moins coûteuses, comme Microsoft Kinect, qui utilise des capteurs de profondeur pour suivre les mouvements sans marqueurs physiques, leur précision n'est pas toujours suffisante pour de nombreuses applications. Ainsi, ce type de données est particulièrement adapté à l'étude détaillée du mouvement humain mais implique des méthodes d'analyse qui ne nécessitent pas une grande quantité de données.

2.2 Contexte sur l'Analyse des Trajectoires de Mouvement Humain

Les mouvements en langue des signes sont régis non seulement par des règles linguistiques, mais aussi par des principes inhérents au mouvement humain (Benchiheub, 2018). Il a été démontré que le mouvement humain est soumis à des contraintes biomécaniques et à des lois de contrôle moteur. L'étude du contrôle moteur est un problème complexe abordé par diverses disciplines telles que la psychologie, les sciences cognitives, la biomécanique et les neurosciences, et plus récemment étendu par l'application de techniques d'apprentissage automatique aux données de mouvement, fournissant des perspectives supplémentaires pour analyser les caractéristiques du mouvement humain. Une partie significative de la recherche en analyse du mouvement humain se concentre sur des mouvements spécifiques, où l'objectif est uniquement défini par le mouvement d'un point final bien identifié, appelé le "end-effector movement" (Polyakov et al., 2009; Carreno-Medrano et al., 2015). Ces études explorent les caractéristiques géométriques et cinématiques de ces mouvements et les contraintes qui les régissent. Dans le contexte

du mouvement en langue des signes, le mouvement du poignet ou de la main peut être considéré comme un “end-effector movement” (Endres et al., 2013).
En principe, le système musculosquelettique humain permet de nombreuses trajectoires potentielles pour un mouvement de la main vers un but. Les chercheurs en contrôle moteur émettent l'hypothèse que le système humain a développé des stratégies de contrôle optimal pour choisir entre ces possibilités par l'évolution, l'apprentissage et l'adaptation. Ainsi, on suppose que le mouvement humain est régi par des contraintes internes et externes assurant le succès du mouvement. Divers principes d'optimalité ont ensuite été proposés dans la littérature (Wochner et al. (2020), Oguz et al. (2018)). L'un des premiers principes mis en évidence par Flash and Hogan (1985) est le principe de minimisation des à-coups ou secousses. Les à-coups sont représentés par la troisième dérivée de la position par rapport au temps, c'est-à-dire la variation de l'accélération. Minimiser les à-coups vise à réduire les changements abrupts d'accélération, rendant le mouvement plus fluide et plus économe en énergie. Ce principe a conduit à d'autres principes de contrôle moteur, tels que le principe d'isochronie et la loi de la puissance des deux tiers (Viviani and Flash, 1995). Viviani and McCollum (1983) a démontré le principe d'isochronie, montrant que la vitesse des mouvements de dessin augmentait proportionnellement à l'amplitude de la trajectoire, maintenant le temps d'exécution de ces trajectoires complexes indépendant de la taille du mouvement. Cette relation linéaire entre la vitesse et l'étendue de la trajectoire a été démontrée pour diverses actions telles que l'écriture ou les mouvements de la main et du bras (Freund and Btidingen, 1978). L'analyse cinématique des mouvements circulaires a révélé une autre loi bien connue du mouvement, la loi de la puissance des deux tiers (également connue sous le nom de loi de la puissance d'un tiers), suggérant une corrélation non linéaire entre la vitesse du mouvement et la courbure. Initialement observée pour les mouvements planaires, une autre version de cette loi pour les mouvements spatiaux implique également le paramètre de torsion de la trajectoire de mouvement. La courbure et la torsion d'une courbe sont des paramètres qui décrivent entièrement sa géométrie. Ces lois de puissance sont formulées comme suit :

$$\dot{s}(t) = C_1 \kappa(s(t))^{-1/3} \qquad \text{(loi de puissance 1/3)}, \tag{B.1}$$

$$\dot{s}(t) = C_2 \left(\kappa(s(t))^2 |\tau(s(t))|\right)^{-1/6} \qquad \text{(loi de puissance 1/6)}, \tag{B.2}$$

où C_1 et C_2 sont des constantes, $\dot{s}$ est la vitesse curviligne et κ et τ sont la courbure et la torsion de la trajectoire de mouvement. Ils supposent que la vitesse augmente dans les portions moins courbées de la trajectoire et diminue de manière inverse dans les portions plus courbées. Endres et al. (2013) utilise la loi de la puissance des deux tiers pour segmenter les mouvements en langue des signes. Enfin, d'autres théories cinématiques des mouvements se trouvent dans la littérature, telles que le modèle Sigma-Lognormal, suggérant que les mouvements humains complexes consistent en des traits superposés, chacun ayant un profil de vitesse en forme de log-normale. Cette théorie a été particulièrement démontrée pour les mouvements de signature dans le plan (Ferrer et al., 2020) et étendue en trois dimensions par Schindler et al. (2018).

Ces lois et principes peuvent servir de points de départ pour l'analyse mathématique et statistique des trajectoires de mouvement. Ramsay et al. (1995) utilise le modèle Sigma-Lognormal pour proposer une analyse de données fonctionnelles de force de pincement du doigt humain. Dans un cadre similaire d'analyse de données fonctionnelles, Raket et al. (2016) a mené une expérience biomédicale sur les mouvements de la main. De nombreux autres travaux dans ce cadre n'étudient pas seulement le “end-effector movement”, mais l'ensemble du mouvement du squelette, utilisant principalement des données acquises par capture de mouvement. Parmi beaucoup d'autres, certains cherchent à aligner temporellement les mouvements (Olsen et al., 2018; Raket et al., 2016), d'autres à modéliser la variabilité spatiale et temporelle d'un ensemble de mouvements (Park et al., 2022), ou à analyser leurs formes (Celledoni et al., 2016). Dans la plupart de ces travaux, les trajectoires de mouvement sont mathématiquement représentées par leurs trois coordonnées cartésiennes. Cependant, cette représentation n'est pas adaptée

à l'analyse des formes de trajectoire, car elle dépend de toutes les transformations rigides (translations, rotations, mise à l'échelle, paramétrisation). Par conséquent, d'autres utilisent une représentation plus sophistiquée impliquant la dérivée de la trajectoire de mouvement, appelée fonction de vitesse à racine carrée (SRVF) (Srivastava et al., 2011), directement sur les courbes euclidiennes (Devanne et al., 2015; Park et al., 2022) ou sur des courbes avec des valeurs en rotations qui apparaissent dans l'animation de personnages (Celledoni et al., 2016). Enfin, plus récemment, Celledoni et al. (2019) et Yang et al. (2022) ont proposé d'utiliser des signatures (Fermanian, 2021) pour décrire les mouvements humains.

3 Approche et Contributions

Cette thèse vise à proposer un cadre mathématique adapté à l'analyse des trajectoires de mouvement de la langue des signes. Nous limitons l'analyse au déplacement d'un point correspondant au poignet droit. Ce choix est justifié par le fait qu'une partie significative de la communication en langue des signes repose sur les mouvements du poignet, ainsi que par la décomposition en quatre classes phonémiques proposée par Brentari (1998), qui isole la composante mouvement des autres. Par conséquent, notre analyse s'inscrit dans le cadre du mouvement de l'effecteur final. En nous appuyant sur les principes précédemment discutés du contrôle moteur, nous pouvons identifier des descripteurs qui semblent particulièrement pertinents pour l'analyse du mouvement humain. Parmi ceux-ci, nous nous intéressons à la vitesse, à la courbure et à la torsion de la trajectoire de mouvement. Les paramètres de courbure et de torsion dans un espace tridimensionnel sont les composantes des formules de Frenet-Serret, initialement définies par deux mathématiciens français, Jean Frédéric Frenet (1852) et Joseph Alfred Serret (1851). Le cadre défini par les formules de Frenet-Serret est particulièrement puissant pour décrire la géométrie d'une courbe euclidienne. Cela a inspiré notre approche du problème mais soulève deux principaux enjeux :

- Comment dériver des estimations lisses et fiables des paramètres de courbure et de torsion à partir de données numériques discrètes et potentiellement bruitées correspondant à la capture de mouvement du poignet ?
- Comment comprendre et analyser la variabilité inhérente à un ensemble de trajectoires de mouvement à partir de ces paramètres ?

Ces questions dépassent leur application au traitement de la langue des signes, trouvant une pertinence pour toute collection de courbes euclidiennes. Elles représentent des défis classiques en analyse de forme de courbes euclidiennes, où l'obstacle initial réside dans l'identification d'une représentation de courbe appropriée.

3.1 Jeux de données

Tout au long de la thèse, nous avons travaillé avec deux principaux jeux de données produits par MocapLab. Le premier provient du projet Rosetta (LIMSI and LISN, 2022), qui comprend la traduction en LSF de 194 gros titres de journaux, des descriptions de 28 photographies et 1200 signes isolés correspondant aux notions les plus courantes des gros titres de journaux. Tous ces mouvements ont été réalisés par le même signeur sourd. Dans ce jeu de données, nous avons isolé 143 signes bien segmentés répétés au moins 2 fois, mais seulement 14 sont répétés au moins 4 fois. Le deuxième a été capturé par MocapLab dans le cadre de cette thèse. En utilisant les dictionnaires LSF LSFPlus, Inc et LeDicoElix, Inc, nous avons sélectionné 60 signes monomanuels (c.-à-d. impliquant le mouvement d'une seule main) en LSF. Deux signeurs, une femme sourde et un homme entendant, ont effectué environ 5 répétitions de chaque signe, revenant à une position initiale entre chaque répétition, donnant un ensemble de 648 mouvements. Les deux jeux de données comprennent les mouvements de plusieurs points clés du corps du signeur, mais nous ne considérons que le mouvement du poignet droit dans cette thèse.

3.2 Contributions

Nous présentons maintenant les contributions de cette thèse aux problèmes mentionnés précédemment et l'organisation du manuscrit. En particulier, le Chapitre 2 de la thèse n'est pas inclus dans ces contributions car il s'agit d'un chapitre de référence sur les concepts mathématiques utilisés tout au long du manuscrit qui comprend des introductions à la géométrie différentielle et riemannienne, aux groupes de Lie matriciels, à l'analyse des données fonctionnelles et de forme, et au cadre de Frenet-Serret.

La Partie I de la thèse se concentre sur l'estimation des courbures de Frenet.

- Le Chapitre 3 rassemble les méthodes existantes pour estimer les courbures de Frenet d'une courbe euclidienne à partir de ses observations numériques discrètes et bruitées. Ces méthodes impliquent l'estimation de variables latentes correspondant soit aux dérivées de la courbe, soit au chemin de Frenet dans une étape de prétraitement. Elles incluent l'utilisation des formules extrinsèques des courbures de Frenet et les méthodes proposées par Park and Brunel (2019), traitant l'estimation des courbures de Frenet comme un problème d'estimation de paramètres d'une équation différentielle. Notre contribution distinctive réside dans l'introduction d'une approche alternative pour l'estimation lisse du chemin de Frenet basée sur un algorithme de tracking. Cette méthode se révèle plus efficace du point de vue computationnel et produit des résultats comparables à ceux des méthodes existantes. En contribution, nous fournissons également des descriptions théoriques complètes de ces différentes méthodes et une comparaison approfondie à travers des études sur des données simulées.

- Le Chapitre 4 aborde le défi posé par les méthodes existantes d'estimation des courbures de Frenet, toutes dépendantes de l'estimation de variables latentes dans une étape distincte de prétraitement. Nous introduisons une nouvelle méthode qui traite l'estimation des variables latentes et des paramètres de manière unifiée en formulant le problème dans le groupe de Lie $SE(3)$. Cette méthode implique le développement d'un algorithme d'espérance-maximisation fonctionnel, où l'étape d'espérance utilise un filtre de Kalman étendu invariant sur le groupe de Lie $SE(3)$, et l'estimation de paramètres est formulée dans le cadre de l'analyse de données fonctionnelles en utilisant la maximisation pénalisée de la vraisemblance attendue. L'amélioration de la précision des estimateurs obtenus avec cette approche est démontrée par des expériences sur des données synthétiques et en appliquant les différentes méthodes aux données de mouvement du poignet en langue des signes.

La Partie II de la thèse est consacrée au développement d'un cadre d'analyse de forme basée sur le cadre de Frenet-Serret.

- Le Chapitre 5 établit un cadre de géométrie riemannienne pour les courbes lisses dans $\mathbb{R}^d, d > 1$, basé sur les courbures de Frenet. Nous introduisons deux nouvelles représentations d'une courbe euclidienne lisse : les Courbures de Frenet non paramétrées (FC) et la Transformée de Courbures à Racine Carrée (SRC, Square Root Curvatures transform), inspirée par la transformée de vitesse à racine carrée du chemin de Frenet à valeurs dans $SO(d)$. Les deux représentations prennent en compte toutes les caractéristiques géométriques de la forme, fournissant des géodésiques probables qui évitent les artefacts rencontrés par les représentations utilisant uniquement des informations géométriques du premier ordre, comme la fonction de vitesse à racine carrée (SRVF). La Transformée de Courbures à Racine Carrée combine les points forts des deux autres, dépendant de la paramétrisation comme le SRVF elle implique une étape d'étirement et recalage nécessaire pour obtenir des géodésiques cohérentes dans certains cas. Notre méthode, soutenue par des données simulées, démontre sa pertinence pour l'analyse des mouvements humains. Ce chapitre est publié sous forme d'article Chassat et al. (2023) pour la conférence ICCV 2023.

- Le Chapitre 6 aborde le défi de définir une forme moyenne d'un ensemble de courbes euclidiennes dans $\mathbb{R}^d, d > 1$. L'approche classique en analyse de forme considère la moyenne de Karcher dans

les espaces de formes définis par les représentations SRVF, SRC ou FC. Nous comparons deux approches distinctes pour l'estimation pratique des moyennes de Karcher basées sur les courbures de Frenet, en tenant compte de leur sensibilité au bruit des observations. La première repose sur une étape de prétraitement robuste impliquant une estimation efficace de chaque courbure de Frenet individuellement dans un cadre d'analyse de données fonctionnelles. En alternative, nous introduisons une approche basée sur la régularisation inspirée du travail de Park and Brunel (2019). Dans un cadre de modélisation de données fonctionnelles, ils formulent explicitement le problème en fournissant une caractérisation statistique du paramètre moyen des courbures de Frenet par le biais du concept de champ vectoriel moyen. En reliant leur travail à l'analyse de forme, nous démontrons que leur critère de régularisation pour l'estimation du paramètre moyen approche la moyenne de Karcher des FC. En exploitant leur idée, nous définissons deux critères alternatifs impliquant une étape d'alignement des courbures individuelles. Les points forts des moyennes estimées obtenues par une approche basée sur la régularisation sont démontrés sur des données simulées bruitées. Dans le cas de données moins bruitées, telles que nos trajectoires de mouvements en langue des signes, les deux approches montrent des différences marginales, mais les méthodes basées sur les courbures de Frenet et tenant compte de la paramétrisation semblent plus efficaces. Une partie de ce travail a été rédigée dans l'article Park et al. (2022) en tant que première révision de l'article Park and Brunel (2019), mais nécessite encore un examen approfondi.

- Le Chapitre 7 se présente comme un chapitre indépendant démontrant des applications plus pratiques des outils et du cadre développés pour l'analyse des trajectoires de mouvement dans le traitement des langues des signes basé sur le cadre de Frenet. En utilisant nos estimateurs robustes pour la courbure et la torsion des trajectoires de mouvement, nous évaluons la validité des lois de puissance sur les données de mouvements du poignet en LSF. De plus, nous introduisons un modèle génératif pour le mouvement du poignet en langue des signes, basé sur la décomposition de la variabilité d'un ensemble de trajectoires à travers sa géométrie moyenne et des fonctions de déformation non linéaires dérivées du paramètre de courbures de Frenet. Notre discussion s'étend aux futures directions de recherche dans le traitement des langues des signes dans le cadre établi, se concentrant particulièrement sur la segmentation et le clustering des trajectoires de mouvement en langage des signes.

Par ailleurs, l'implémentation de toutes les méthodes définies dans cette thèse a contribué au développement du package Python `FrenetFDA`, disponible sur GitHub (https://github.com/perrinechassat/FrenetFDA) et détaillé en annexe de la thèse, regroupant tous les outils nécessaires pour effectuer une analyse de courbe en dimension inférieure ou égale à 3 basée sur le cadre de Frenet.

3.3 Publications

Ces contributions ont donné lieu aux publications et pre-prints suivants. D'autres articles sont également en cours de rédaction et seront soumis dans les prochains mois.

Articles de Conférences

- **Chassat Perrine**, Juhyun Park, Nicolas Brunel. "*Shape Analysis of Euclidean Curves under Frenet-Serret Framework*". International Conference on Computer Vision (ICCV), Paris, 2023 (accepted for oral presentation), https://openaccess.thecvf.com/content/ICCV2023/papers/.

- **Chassat Perrine**, Juhyun Park, Nicolas Brunel. "*Analysis of variability in sign language hand trajectories: development of generative model*". Proceedings of the 8th International Conference on Movement and Computing (MOCO '22), Chicago, 2022.
DOI: https://doi.org/10.1145/3537972.3537999

Article en Cours de Révision

- Park Juhyun, Nicolas Brunel, **Perrine Chassat**. "*Curvature and Torsion estimation of 3D functional data: A geometric approach to build the mean shape under the Frenet Serret framework*". Preprint arXiv:2203.02398, 2021.

Articles en Préparation

- **Chassat Perrine**, Juhyun Park, Nicolas Brunel. "*Functional Expectation-Maximization Algorithm on SE(3) for Frenet Curvatures Estimation*". (à venir).

Bibliography

Abeles, M., M. Diesmann, T. Flash, T. Geisel, M. Hermann, and M. Teicher (2013). Compositionality in neural control: an interdisciplinary study of scribbling movements in primates. *Frontiers in Computational Neuroscience 7*.

Absil, P.-A., R. Mahony, and R. Sepulchre (2010). *Optimization On Manifolds: Methods And Applications*. Princeton University Press.

Ansley, C. F. and R. Kohn (1985). Estimation, Filtering, and Smoothing in State Space Models with Incompletely Specified Initial Conditions. *The Annals of Statistics 13*(4), 1286 – 1316.

Antonsanti, P.-L., T. Benseghir, V. Jugnon, M. Ghosn, P. Chassat, J. Glaunès, and I. Kaltenmark (2022). How to register a live onto a liver ? partial matching in the space of varifolds. *Machine Learning for Biomedical Imaging 1*, 1–30.

Arivazhagan, N., A. Bapna, O. Firat, D. Lepikhin, M. Johnson, M. Krikun, M. X. Chen, Y. Cao, G. F. Foster, C. Cherry, W. Macherey, Z. Chen, and Y. Wu (2019). Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR abs/1907.05019*.

Backenroth, D., J. Goldsmith, M. D. Harran, J. C. Cortes, J. W. Krakauer, and T. Kitago (2018, jul). Modeling Motor Learning Using Heteroscedastic Functional Principal Components Analysis. *Journal of the American Statistical Association 113*(523), 1003–1015.

Baras, J. S., A. Bensoussan, and M. R. James (1988). Dynamic observers as asymptotic limits of recursive filters: Special cases. *SIAM Journal on Applied Mathematics 48*(5), 1147–1158.

Barfoot, T. D. (2017). State estimation for robotics. *State Estimation for Robotics*, 1–368.

Barfoot, T. D. and P. T. Furgale (2014). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics 30*, 679–693.

Barrau, A. and S. Bonnabel (2015). Intrinsic filtering on lie groups with applications to attitude estimation. *IEEE Transactions on Automatic Control 60*.

Barrau, A. and S. Bonnabel (2017). The invariant extended kalman filter as a stable observer. *IEEE Transactions on Automatic Control 62*, 1797–1812.

Barrau, A. and S. Bonnabel (2018). Stochastic observers on lie groups: A tutorial. *Proceedings of the IEEE Conference on Decision and Control 2018-Decem*, 1264–1269.

Bauer, M., N. Charon, E. Klassen, and A. L. Brigant (2021). Intrinsic riemannian metrics on spaces of curves: Theory and computation. *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, 1–35.

Bauer, M., N. Charon, E. Klassen, S. Kurtek, T. Needham, and T. Pierron (2022, September). Elastic Metrics on Spaces of Euclidean Curves: Theory and Algorithms. arXiv:2209.09862 [math].

Bauer, M., C. Maor, and P. Michor (2022, June). Sobolev metrics on spaces of manifold valued curves. *ANNALI SCUOLA NORMALE SUPERIORE - CLASSE DI SCIENZE*, 47.

Benchiheub, M. (2018). *Contribution à l'analyse des mouvements 3D de la Langue des Signes Française (LSF) en Action et en Perception*. Ph. D. thesis.

Bertsekas, D. (1995). *Dynamic Programming and Optimal Control*, Volume 1. Athena Scientific.

Bickel, P. J. and B. Li (2007). Local polynomial regression on unknown manifolds. In *Complex Datasets and Inverse Problems*, pp. 177–186. Institute of Mathematical Statistics.

Biegler, L. T., J. J. Damiano, and G. E. Blau (1986). Nonlinear parameter estimation: A case study comparison. *AIChE Journal 32*(1), 29–45.

Bigand, F. (2022). *Extracting human characteristics from motion using machine learning : the case of identity in Sign Language To cite this version : Extracting human characteristics from motion using machine*. Ph. D. thesis.

Blanes, S., F. Casas, J. A. Oteo, and J. Ros (2009). The magnus expansion and some of its applications. *Physics Reports 470*, 151–238.

Bonnabel, S. (2007). Left-invariant extended kalman filter and attitude estimation. In *46th IEEE Conference on Decision and Control*, pp. 1027–1032.

Boothby, W. M. (2003). *An Introduction to Differentiable Manifolds and Riemannian Geometry*.

Bourmaud, G., R. Megret, A. Giremus, and Y. Berthoumieu (2013). Discrete extended kalman filter on lie groups. *European Signal Processing Conference*.

Bourmaud, G., R. Mégret, M. Arnaudon, and A. Giremus (2015). Continuous-discrete extended kalman filter on matrix lie groups using concentrated gaussian distributions. *Journal of Mathematical Imaging and Vision 51*, 209–228.

Bragg, D., O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, C. Vogler, and M. Ringel Morris (2019). Sign language recognition, generation, and translation: An interdisciplinary perspective. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19, New York, NY, USA, pp. 16–31. Association for Computing Machinery.

Brentari, D. (1998). *A Prosodic Model of Sign Language Phonology*.

Brunel, N. and Q. Clairon (2015, 01). A tracking approach to parameter estimation in linear ordinary differential equations. *Electronic Journal of Statistics 9*, 2903–2949.

Brunel, N. J.-B. and J. Park (2019). The frenet-serret framework for aligning geometric curves. In F. Nielsen and F. Barbaresco (Eds.), *Geometric Science of Information*, Cham, pp. 608–617. Springer International Publishing.

Bryner, D. and A. Srivastava (2022). Shape analysis of functional data with elastic partial matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*(12), 9589–9602.

Brynjarsdóttir, J. and A. O'Hagan (2014). Learning about physical parameters: the importance of model discrepancy. *Inverse problems 30*, 114007.

Bull, H., M. Gouiffès, and A. Braffort (2020, 01). *Automatic Segmentation of Sign Language into Subtitle-Units*, pp. 186–198.

Camgöz, N. C., O. Koller, S. Hadfield, and R. Bowden (2020). Sign language transformers: Joint end-to-end sign language recognition and translation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10020–10030.

Camille Jordan (1874). Sur la théorie des courbes dans l'espace à n dimensions. *C. R. Acad. Sci. Paris 79*, 795–797.

Campbell, D. and O. Chkrebtii (2013). Maximum profile likelihood estimation of differential equation parameters through model based smoothing state estimates. *Mathematical Biosciences 246*(2), 283–292.

Cao, J., J. Z. Huang, and H. Wu (2012). Penalized nonlinear least squares estimation of time-varying parameters in ordinary differential equations. *Journal of Computational and Graphical Statistics 21*, 42–56.

Cao, Z., G. Martinez, T. Simon, S.-E. Wei, and Y. Sheikh (2019, 07). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence PP*, 1–1.

Cardot, H. and P. Sarda (2005). Estimation in generalized linear models for functional data via penalized likelihood. *Journal of Multivariate Analysis 92*(1), 24–41.

Carmo, M. P. d. (1976). *Differential geometry of curves and surfaces*. Prentice-Hall.

Carreno-Medrano, P., S. Gibet, and P.-F. Marteau (2015). End-effectors trajectories: An efficient low-dimensional characterization of affective-expressive body motions. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 435–441.

Carroll, C., H. G. Müller, and A. Kneip (2020). Cross-component registration for multivariate functional data, with application to growth curves. *Biometrics*.

Celledoni, E., M. Eslitzbichler, and A. Schmeding (2016). Shape analysis on lie groups with applications in computer animation. *Journal of Geometric Mechanics 8*(3), 273–304.

Celledoni, E., P. E. Lystad, and N. Tapia (2019). Signatures in shape analysis: An efficient approach to motion identification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11712 LNCS*, 21–30.

Chassat, P., J. Park, and N. Brunel (2022). Analysis of variability in sign language hand trajectories: Development of generative model. In *Proceedings of the 8th International Conference on Movement and Computing*, MOCO '22, New York, NY, USA. Association for Computing Machinery.

Chassat, P., J. Park, and N. Brunel (2023, October). Shape analysis of euclidean curves under frenet-serret framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4027–4036.

Chaudhuri, P. and J. S. Marron (2000). Scale space view of curve estimation. *The Annals of Statistics 28*, 408–428.

Chen, J. and H. Wu (2008). Efficient local estimation for time-varying coefficients in deterministic dynamic models with applications to hiv-1 dynamics. *Journal of the American Statistical Association 103*, 369–384.

Chiou, J.-M., Y.-T. Chen, and Y.-F. Yang (2014). Mutivariate functional principal component analysis: A normalization approach. *Statistica Sinica 24*, 1571–1596.

Chirikjian, G. S. (2009). *Stochastic Models, Information Theory and Lie Groups Volume 2, Analytic Mehods and Modern Applications*, Volume 53.

Chong, G. (2002). *Smoothing Spline ANOVA Models*. Springer New York, NY.

Chow, G. (1986). *Analysis and Control of Dynamic Economic Systems*. R.E. Krieger.

Clairon, Q. and N. J. Brunel (2019). Tracking for parameter and state estimation in possibly misspecified partially observed linear ordinary differential equations. *Journal of Statistical Planning and Inference 199*, 188–206.

Cole, S., H. Chu, and S. Greenland (2013, 10). Maximum likelihood, profile likelihood, and penalized likelihood: A primer. *American journal of epidemiology 179*.

Dai, W. and M. G. Genton (2018). Multivariate functional data visualization and outlier detection. *Journal of Computational and Graphical Statistics 27(4)*, 923–934.

Dai, X. and H.-G. Müller (2018). Principal Component Analysis for Functional Data on Riemannian Manifolds and Spheres. *The Annals of Statistics 46*, 3334–3361.

de Boor, C. (1978, 01). *A Practical Guide to Spline*, Volume Volume 27.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological) 39*(1), 1–38.

Despinoy, F., D. Bouget, G. Forestier, C. Penet, N. Zemiti, P. Poignet, and P. Jannin (2016). Unsupervised trajectory segmentation for surgical gesture recognition in robotic training. *IEEE Transactions on Biomedical Engineering 63*, 1280–1291.

Devanne, M., H. Wannous, P. Pala, S. Berretti, M. Daoudi, and A. Del Bimbo (2015). Combined shape analysis of human poses and motion units for action segmentation and recognition. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG) 07*, 1–6.

do Carmo, M. (1992). *Riemannian Geometry*. Mathematics (Boston, Mass.). Birkhäuser.

Dryden, I. and K. Mardia (1998). *Statistical shape analysis*. Wiley Series in Probability and Statistics. Wiley.

Du, P. and X. Wang (2014). Penalized likelihood functional regression. *Statistica Sinica 24*(2), 1017–1041.

Duan, X., H. Sun, and L. Peng (2013). Riemannian means on special euclidean group and unipotent matrices group. *The Scientific World Journal 2013*.

Dubey, P. and H. G. Müller (2019). Fréchet analysis of variance for random objects. *Biometrika 106*(4), 803–821.

Dudis, P. G. (2004). Body partitioning and real-space blends. *Cognitive Linguistics 15*(2), 223–238.

Dwivedi, R., N. Ho, K. Khamaru, M. I. Jordan, M. J. Wainwright, and B. Yu (2018). Singularity, misspecification and the convergence rate of em. *The Annals of Statistics*.

Eisenberger, M. and D. Cremers (2020). Hamiltonian dynamics for real-world shape interpolation. In *European Conference on Computer Vision (ECCV)*.

Emmorey, K. (2001, 11). *Language, Cognition, and the Brain: Insights From Sign Language Research*.

Endres, D., Y. Meirovitch, T. Flash, and M. A. Giese (2013). Segmenting sign language into motor primitives with bayesian binning. *Frontiers in Computational Neuroscience 7*, 1–13.

Fan, J. and I. Gijbels (1996). *Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Farag, I. and H. Brock (2019, 05). Learning motion disfluencies for automatic sign language segmentation. *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Fermanian, A. (2021, 05). Embedding and learning with signatures. *Computational Statistics & Data Analysis 157*, 107148.

Ferraty, F. and P. Vieu (2006). *Nonparametric functional data analysis*. Springer Series in Statistics. New York: Springer. Theory and practice.

Ferrer, M. A., M. Diaz, C. Carmona-Duarte, and R. Plamondon (2020). idelog: Iterative dual spatial and kinematic extraction of sigma-lognormal parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence 42*(1), 114–125.

Flash, T. and A. Berthoz (2021). *Space-Time Geometries for Motion and Perception in the Brain and the Arts*. Lecture Notes in Morphogenesis. Springer.

Flash, T. and N. Hogan (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience 5*(7), 1688–1703.

Fletcher, P. T., C. Lu, S. M. Pizer, and S. Joshi (2004). Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging 23*, 995–1005.

Freund, H. J. and H. J. Btidingen (1978). The relationship between speed and amplitude of the fastest voluntary contractions of human arm muscles. *Experimental Brain Research 31*, 1–12.

Gibet, S., F. Lefebvre-Albaret, L. Hamon, R. Brun, and A. Turki (2015, 09). Interactive editing in french sign language dedicated to virtual signers: Requirements and challenges. *Universal Access in the Information Society 15*.

Glaunès, J., A. Qiu, M. Miller, and L. Younes (2008, 12). Large deformation diffeomorphic metric curve mapping. *International journal of computer vision 80*, 317–336.

Goldsmith, J. and T. Kitago (2016). Assessing systematic effects of stroke on motor control by using hierarchical function-on-scalar regression. *Journal of the Royal Statistical Society. Series C: Applied Statistics 65*(2), 215–236.

Green, P. and B. Silverman (1993). *Nonparametric Regression and Generalized Linear Models: A roughness penalty approach*. Chapman and Hall/CRC (1st Edition).

Grenander, U. (1993). *General Pattern Theory: A Mathematical Study of Regular Structures*.

Guitteny, P. and L. Verlaine (2018). *Comprendre la langue des signes*. Dictionnaires bilingues. International Visual Theatre.

Hairer, E., C. Lubich, and G. Wanner (2006). *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer Series in Computational Mathematics. Springer.

Hall, B. (2003). *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Graduate Texts in Mathematics. Springer.

Happ, C. and S. Greven (2018). Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains. *Journal of the American Statistical Association 113*(522), 649–659.

Happ, C., F. Scheipl, A.-A. Gabriel, and S. Greven (2019). A general framework for multivariate functional principal component analysis of amplitude and phase variation. *Stat 8*, e220.

Hartikainen, J., M. Seppänen, and S. Särkkä (2012). State-space inference for non-linear latent force models with application to satellite orbit prediction. *Proceedings of the 29th International Conference on Machine Learning, ICML 2012 1*, 903–910.

Hartikainen, J. and S. Särkkä (2010). Kalman filtering and smoothing solutions to temporal gaussian process regression models. *Proceedings of the 2010 IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2010*, 379–384.

Hartikainen, J. and S. Särkkä (2011). Sequential inference for latent force models. *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*, 311–318.

Hicheur, H., S. Vieilledent, M. J. E. Richardson, T. Flash, and A. Berthoz (2005). Velocity and curvature in human locomotion along complex curved paths: a comparison with hand movements. *Experimental Brain Research 162*, 145–154.

Higham, N. J. (2008). *Functions of Matrices*. Society for Industrial and Applied Mathematics.

Hinkle, J., P. T. Fletcher, and S. Joshi (2014). Intrinsic polynomials for regression on riemannian manifolds. *J. Math. Imaging Vis. 50*(1–2), 32–52.

Hinkle, J., P. Muralidharan, P. T. Fletcher, and S. Joshi (2012). Polynomial regression on riemannian manifolds. In *Computer Vision – ECCV 2012*, pp. 1–14. Springer Berlin Heidelberg.

Hirsh, S. M., S. M. Ichinaga, S. L. Brunton, J. N. Kutz, and B. W. Brunton (2021, oct). Structured time-delay models for dynamical systems with connections to frenet–serret frame. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 477*(2254).

Huckemann, S., T. Hotz, and A. Munk (2010). Intrinsic shape analysis: Geodesic pca for riemannian manifolds modulo isometric lie group actions. *Statistica Sinica 20*(1), 1–58.

Isard, A. (2020, May). Approaches to the anonymisation of sign language corpora. In *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, Marseille, France, pp. 95–100. European Language Resources Association (ELRA).

Iserles, A., H. Munthe-Kaas, S. Norsett, and A. Zanna (2000). Lie-group methods. *Acta Numerica 9*, 215–365.

Jacod, J. and A. Shiryaev (2013). *Limit theorems for stochastic processes* (2nd ed. ed.), Volume 288 of *Grundlehren der mathematischen Wissenschaften*. Springer.

Jakubiak, J., F. Silva Leite, and R. C. Rodrigues (2006). A two-step algorithm of smooth spline generation on riemannian manifolds. *Journal of Computational and Applied Mathematics 194*(2), 177–191.

James, G. M. (2002). Generalized linear models with functional predictors. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 64*(3), 411–432.

Jean Frédéric Frenet (1852). Sur les courbes à double courbure. *Journal de Mathématiques Pures et Appliquées 1e série, 17*.

Johnson, R. E. and S. K. Liddell (2011). Toward a phonetic representation of signs: Sequentiality and contrast. *Sign Language Studies 11*(2), 241–274.

Jolliffe, I. (2002). *Principal component analysis*. 2nd edn. New York, NY: Springer-Verlag.

Joseph Alfred Serret (1851). Sur quelques formules relatives à la théorie des courbes à double courbure. *Journal de Mathématiques Pures et Appliquées 1e série, 16*.

Kac, V. G. (1990). *Infinite-Dimensional Lie Algebras* (3 ed.). Cambridge University Press.

Kalman, R. E. (1960, 03). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering 82*(1), 35–45.

Karcher, H. (1977). Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics 30*(5), 509–541.

Kendall, D. G., D. Barden, T. K. Carne, and H. Le (1999). *Shape & Shape Theory*. Wiley Series in Probability and Statistics. John Wiley & Sons Ltd.

Kendall, W. and H. Le (2010, 01). *Statistical Shape Theory*, Volume 10, pp. 348–373.

Kendrick, D. (1981). *Stochastic Control for Economic Models*. Economics handbook series. McGraw-Hill.

Kim, K. R., I. L. Dryden, H. Le, and K. E. Severn (2021). Smoothing splines on riemannian manifolds, with applications to 3d shape space. *Journal of the Royal Statistical Society. Series B: Statistical Methodology 83*, 108–132.

Kim, K.-R., P. T. Kim, J.-Y. Koo, and M. R. Pierrynowski (2013). Frenet-serret and the estimation of curvature and torsion. *IEEE Journal of Selected Topics in Signal Processing 7*(4), 646–654.

Kneip, A. and T. Gasser (1992). Statistical tools to analyze data representing a sample of curves. *The Annals of Statistics 20*(3), 1266–1305.

Kneip, A. and J. O. Ramsay (2008). Combining registration and fitting for functional models. *Journal of the American Statistical Association 103*(483), 1155–1165.

Koestler, L., D. Grittner, M. Moeller, D. Cremers, and Z. Lähner (2022). Intrinsic neural fields: Learning functions on manifolds. In *European Conference on Computer Vision (ECCV)*.

Kühnel, W. (2002). *Differential Geometry: Curves - Surfaces - Manifolds*.

Kurtek, S., A. Srivastava, E. Klassen, and Z. Ding (2012). Statistical modeling of curves using shapes and related features. *Journal of the American Statistical Association 107*(499), 1152–1165.

Lacquaniti, F., C. Terzuolo, and P. Viviai (1983). The law relating the kinematic and figural aspects of drawing movements. *Acta Psychologica (Amst.) 54*, 115–130.

Lang, S. (2006). Differential and riemannian manifolds. *Springer 1999*, 1–6.

LaPlace, P. S. (1820). *Théorie analytique des probabilités*. Courcier.

Le, H. (2004). Estimation of riemannian barycentres. *LMS Journal of Computation and Mathematics 7*, 193–200.

LeDicoElix, Inc. https://dico.elix-lsf.fr/. Accessed: (30/11/2023).

Lewiner, T., J. D. Gomes, H. Lopes, and M. Craizer (2005). Curvature and torsion estimators based on parametric curve fitting. *Computers & Graphics 29*(5), 641–655.

LIMSI and LISN (2022). rosetta-lsf. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

Lin, Z. and F. Yao (2019). Intrinsic Riemannian functional data analysis. *The Annals of Statistics 47*(6), 3533–3577.

Liu, L., M. Levine, and Y. Zhu (2009). A functional em algorithm for mixing density estimation via nonparametric penalized likelihood maximization. *Journal of Computational and Graphical Statistics 18*, 481–504.

Liu, X. and H.-G. Müller (2004). Functional convex averaging and synchronization for time-warped random curves. *Journal of the American Statistical Association 99*(467), 687–699.

LSFPlus, Inc. https://lsfplus.fr/. Accessed: (30/11/2023).

Lyons, R. (2016). Frobenius theorem two ways. *Lecture note*.

Lähner, Z., E. Rodola, F. R. Schmidt, M. M. Bronstein, and D. Cremers (2016). Efficient globally optimal 2d-to-3d deformable shape matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Magnus, W. (1954). On the exponential solution of differential equations for a linear operator. *Communications on Pure and Applied Mathematics 7*(4), 649–673.

Maoz, U., A. Berthoz, and T. Flash (2009). Complex unconstrained three-dimensional hand movement and constant equi-affine speed. *Journal of Neurophysiology 101*, 1002–1015.

Mardia, K. and P. Jupp (1999). *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley.

Marken, R. S. and D. M. Shaffer (2017). The power law of movement: an example of a behavioral illusion. *Experimental Brain Research 235*, 1835–1842.

Marron, J. S., J. O. Ramsay, L. M. Sangalli, and A. Srivastava (2015). Functional data analysis of amplitude and phase variation. *Statistical Science 30*, 468–484.

McCullagh, P. and J. Nelder (1960). *Generalized Linear Models*, Volume 4.

Mehmet, A. (2005). Submanifolds of Riemannian Product Manifolds. *Turkish Journal of Mathematics 29*.

Meng, X.-L. and D. van Dyk (1997). The em algorithm–an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society. Series B (Methodological) 59*(3), 511–567.

Miao, H., X. Xia, A. S. Perelson, and H. Wu (2011). On identifiability of nonlinear ode models and applications in viral dynamics. *SIAM Review 53*(1), 3–39.

Moskowitz, M. and H. Abbaspour (2007). *Basic Lie Theory*. World Scientific Publishing Company.

Müller, H. G. and U. Stadtmüller (2005). Generalized functional linear models. *Annals of Statistics 33*, 774–805.

Needham, T. (2019). Shape analysis of framed space curves. *Journal of Mathematical Imaging and Vision 61*(8), 1154–1172.

Oguz, O. S., Z. Zhou, and D. Wollherr (2018). A hybrid framework for understanding and predicting human reaching motions. *Frontiers in Robotics and AI 5*.

Olsen, N. L., B. Markussen, and L. L. Raket (2018). Simultaneous inference for misaligned multivariate functional data. *Journal of the Royal Statistical Society. Series C: Applied Statistics 67*, 1147–1176.

Optitrack, Inc. Optitrack NaturalPoint. https://optitrack.com/. Accessed: (30/11/2023).

Park, C., S. D. Noh, and A. Srivastava (2022). Data science for motion and time analysis with modern motion sensor data. *Operations Research*.

Park, J., N. Brunel, and P. Chassat (2022). Curvature and torsion estimation of 3d functional data: A geometric approach to build the mean shape under the frenet serret framework. arXiv:2203.02398 [stat].

Park, J. and N. J.-B. Brunel (2019). Mean curvature and mean shape for multivariate functional data under frenet-serret framework. arXiv:1910.12049 [stat].

Patrangenaru, V. and L. Ellingson (2015). *Nonparametric Statistics on Manifolds and Their Applications to Object Data Analysis*. CRC Press.

Pennec, X., S. Sommer, and T. Fletcher (Eds.) (2020). *Riemannian Geometric Statistics in Medical Image Analysis*. Academic Press.

Petersen, A. and H. G. Müller (2019). Fréchet regression for random objects with Euclidean predictors. *The Annals of Statistics 47*(2), 691–719.

Pilté, M., S. Bonnabel, and F. Barbaresco (2017). Tracking the frenet-serret frame associated to a highly maneuvering target in 3d. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 1969–1974.

Pollick, F., U. Maoz, A. Handzel, P. Giblin, G. Sapiro, and T. Flash (2009). Three-dimensional arm movements at constant equi-affine speed. *Cortex 45*, 325–339.

Polyakov, F., E. Stark, R. Drori, M. Abeles, and T. Flash (2009). Parabolic movement primitives and cortical states: Merging optimality with geometric invariance. *Biological Cybernetics 100*(2), 159–184.

Prillwitz, S. and H. Zienert (1990). Hamburg notation system for sign language: Development of a sign writing with computer application. *In Current Trends in European Sign Language Research. Proceedings of the 3rd European Congress on Sign Language Research*.

Qi, X. and H. Zhao (2010). Asymptotic efficiency and finite-sample properties of the generalized profiling estimation of parameters in ordinary differential equations. *The Annals of Statistics 38*(1), 435 – 481.

Raket, L. L., B. Grimme, G. Schöner, C. Igel, and B. Markussen (2016). Separating timing, movement conditions and individual differences in the analysis of human movement. *POLS Computational Biology 12(9)*, 1–27.

Ramsay, J. and G. Hooker (2017). *Dynamic Data Analysis Modeling Data with Differential Equations*. Springer New York, NY.

Ramsay, J. and B. Silverman (2005). *Functional data analysis* (Second ed.). Springer Series in Statistics. Springer.

Ramsay, J. O. and C. J. Dalzell (1991). Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological) 53*, 539–572.

Ramsay, J. O., G. Hooker, D. Campbell, and J. Cao (2007). Parameter estimation for differential equations: A generalized smoothing approach. *Journal of the Royal Statistical Society. Series B: Statistical Methodology 69*, 741–796.

Ramsay, J. O. and X. Li (1998). Curve registration. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 60*(2), 351–363.

Ramsay, J. O., X. Wang, and R. Flanagan (1995). A functional data analysis of the pinch force of human fingers. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 44*(1), 17–30.

Rastgoo, R., K. Kiani, and S. Escalera (2021). Sign language recognition: A deep survey. *Expert Systems with Applications 164*, 113794.

Reiss, P. T. and R. T. Ogden (2007). Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association 102*, 984–996.

Rentmeesters, Q. and P.-A. Absil (2011, 01). Algorithm comparison for karcher mean computation of rotation matrices and diffusion tensors. *European Signal Processing Conference*.

Saba, M. (2012). On the usage of the curvature for the comparison of planar curves.

Sallandre, M.-A. and C. Cuxac (2002). Iconicity in sign language: A theoretical and methodological point of view. In I. Wachsmuth and T. Sowa (Eds.), *Gesture and Sign Language in Human-Computer Interaction*, pp. 173–180. Springer Berlin Heidelberg.

Samir, C., P. A. Absil, A. Srivastava, and E. Klassen (2012, feb). A gradient-descent method for curve fitting on riemannian manifolds. *Found. Comput. Math. 12*(1), 49–73.

Sandler, W. (2012, 03). The phonological organization of sign languages. *Language and linguistics compass 6*, 162–182.

Sangalli, L. M., P. Secchi, S. Vantini, and A. Veneziani (2009). Efficient estimation of three-dimensional curves and their derivatives by free-knot regression splines, applied to the analysis of inner carotid artery centrelines. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 58*(3), 285–306.

Santemiz, P., O. Aran, M. Saraclar, and L. Akarun (2009, 11). Automatic sign segmentation from continuous signing via multiple sequence alignment. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*, 2001 – 2008.

Sarkka, S. T. (2006). *Recursive bayesian inference on stochastic differential equations*, Volume 68.

Saunders, B., N. Camgoz, and R. Bowden (2020a). *Progressive Transformers for End-to-End Sign Language Production*, pp. 687–705.

Saunders, B., N. C. Camgoz, and R. Bowden (2020b). Everybody sign now: Translating spoken language to photo realistic sign language video. *ArXiv abs/2011.09846*.

Schindler, R., M. Bouillon, R. Plamondon, and A. Fischer (2018). Extending the sigma-lognormal model of the kinematic theory to three dimensions. Zenodo.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological) 47*(1), 1–52.

Small, C. G. (1996). *The Statistical Theory of Shape*. Springer-Verlag.

Sommer, S., T. Fletcher, and X. Pennec (2020). *Introduction to differential and Riemannian geometry*.

Srivastava, A. and E. Klassen (2016). *Functional and Shape Data Analysis*. Springer Series in Statistics. Springer New York.

Srivastava, A., E. Klassen, S. H. Joshi, and I. H. Jermyn (2011). Shape analysis of elastic curves in euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*(7), 1415–1428.

Stillwell, J. (2008). *Naive Lie Theory*. Springer New York, NY.

Stokoe, William C., J. (1960, 01). Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf. *The Journal of Deaf Studies and Deaf Education 10*(1), 3–37.

Stoll, S., N. Camgoz, S. Hadfield, and R. Bowden (2020, 04). Text2sign: Towards sign language production using neural machine translation and generative adversarial networks. *International Journal of Computer Vision*.

Su, J., I. Dryden, E. Klassen, H. Le, and A. Srivastava (2012). Fitting smoothing splines to time-indexed, noisy points on nonlinear manifolds. *Image and Vision Computing 30*(6), 428–442.

Su, J., S. Kurtek, E. Klassen, and A. Srivastava (2014). Statistical analysis of trajectories on riemannian manifolds: Bird migration, hurricane tracking and video surveillance. *Annals of Applied Statistics 8(1)*, 530–552.

Su, Z., E. Klassen, and M. Bauer (2017). The square root velocity framework for curves in a homogeneous space. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 680–689.

Surazhsky, T. and G. Elber (2002). Metamorphosis of planar parametric curves via curvature interpolation. *International Journal of Shape Modeling 8*, 201–216.

Sutton, V. (2014). *Lessons in Sign Writing: Textbook*. Deaf Action Committee for Sign Writing.

Särkkä, S. (2010). *Bayesian Filtering and Smoothing*. Cambridge University Press.

Särkkä, S. and J. Sarmavuori (2013). Gaussian filtering and smoothing for continuous-discrete dynamic systems. *Signal Processing 93*, 500–510.

Tang, R. and H.-G. Müller (2008). Pairwise curve synchronization for functional data. *Biometrika 95*(4), 875–889.

Thomas Fletcher, P. (2013, nov). Geodesic regression and the theory of least squares on riemannian manifolds. *Int. J. Comput. Vision 105*(2), 171–185.

Tixiang, W. (1989). Morse theory on banach manifolds. *Acta Mathematica Sinica 5*, 250–262.

Tu, L. (2008). *An Introduction to Manifolds*. Springer New York, NY.

Tucker, J. D. (2018). Functional component analysis and regression using elastic methods. *Angewandte Chemie International Edition, 6(11), 951–952.*, 10–27.

Tucker, J. D., J. R. Lewis, and A. Srivastava (2019). Elastic functional principal component regression. *Statistical Analysis and Data Mining: The ASA Data Science Journal 12*(2), 101–115.

Tucker, J. D., W. Wu, and A. Srivastava (2013). Generative models for functional data using phase and amplitude separation. *Computational Statistics and Data Analysis 61*, 50–66.

Varah, J. M. (1982). A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing 3*(1), 28–46.

Vicon, Inc. Vicon Motion System. https://www.vicon.com/. Accessed: (30/11/2023).

Viviani, P. and T. Flash (1995). Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning. *Journal of experimental psychology. Human perception and performance 21 1*, 32–53.

Viviani, P. and G. McCollum (1983). The relation between linear extent and velocity in drawing movements. *Neuroscience 10*(1), 211–218.

Wadhawan, A. and P. Kumar (2020, jun). Deep learning-based sign language recognition system for static signs. *Neural Comput. Appl. 32*(12), 7957–7968.

Wahba, B. G. (1978). Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 364–372.

Wang, J., J. Chiou, and H. Müller (2016). Functional data analysis. *Annual Review of Statistics and Its Application 3*, 257–295.

Wang, Q., Y.-Y. Chang, R. Cai, Z. Li, B. Hariharan, A. Holynski, and N. Snavely (2023). Tracking everything everywhere all at once. *IEEE International Conference on Computer Vision, ICCV 2023 abs/2306.05422*.

Wang, Q. and E. Weinan (2021). An expectation-maximization algorithm for continuous-time hidden markov models. pp. 1–34. arXiv:2103.16810.

Wang, Y. and G. S. Chirikjian (2006). Error propagation on the euclidean group with applications to manipulator kinematics. *IEEE Transactions on Robotics 22*, 591.

Wang, Y. and G. S. Chirikjian (2008). Nonparametric second-order theory of error propagation on motion groups. *International Journal of Robotics Research 27*, 1258–1273.

Warner, F. W. (1983). *Foundations of Differentiable Manifolds and Lie Groups*. 0072-5285. Springer New York, NY.

Wecker, W. E. and C. F. Ansley (1983). The signal extraction approach to nonlinear regression and spline smoothing. *Journal of the American Statistical Association 78*, 81–89.

Wei, S. and V. M. Panaretos (2018). Empirical evolution equations. *Electronic Journal of Statistics 12*(1), 249–276.

Whittle, P. (1990). *Risk-sensitive optimal control*. Wiley-Interscience series in systems and optimization.

Wilcox, S. (1992). *The Phonetics of Fingerspelling*. Studies in Speech Pathology and Clinical Linguistics. John Benjamins Publishing Company.

Wochner, I., D. Driess, H. Zimmermann, D. F. B. Haeufle, M. Toussaint, and S. Schmitt (2020). Optimality principles in human point-to-manifold reaching accounting for muscle dynamics. *Frontiers in Computational Neuroscience 14*.

Xiao, Q., M. Qin, and Y. Yin (2020, 02). Skeleton-based chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. *Neural Networks 125*.

Yang, W., T. Lyons, H. Ni, C. Schmid, and L. Jin (2022). *Developing the Path Signature Methodology and Its Application to Landmark-Based Human Action Recognition*, pp. 431–464. Cham: Springer International Publishing.

Yao, F., H.-G. Müller, and J.-L. Wang (2005, December). Functional linear regression analysis for longitudinal data. *The Annals of Statistics 33*(6).

Yin, K., A. Moryossef, J. Hochgesang, Y. Goldberg, and M. Alikhani (2021). Including signed languages in natural language processing. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 7347–7360.

Younes, L. (1998). Computable Elastic Distances between Shapes. *SIAM Journal on Applied Mathematics 58*(2), 565–586. Publisher: Society for Industrial and Applied Mathematics.

Younes, L. (2010). *Shapes and Diffeomorphisms*. Applied Mathematical Sciences. Springer.

Younes, L. (2018). Elastic distance between curves under the metamorphosis viewpoint. arXiv:1804.10155.

Zago, M., F. Lacquaniti, and A. Gomez-Marin (2016). The speed-curvature power law in drosophila larval locomotion. *Biology Letters 12*(10), 20160597.

Zago, M., A. Matic, T. Flash, A. Gomez-Marin, and F. Lacquaniti (2018). The speed-curvature power law of movements: a reappraisal. *Experimental Brain Research 236*, 69–82.

Zhang, C., H. Yan, S. Lee, and J. Shi (2020). Dynamic Multivariate Functional Data Modeling via Sparse Subspace Learning. *Technometrics*.

Zhu, H., J. S. Morris, F. Wei, and D. D. Cox (2017). Multivariate functional response regression, with application to fluorescence spectroscopy in a cervical pre-cancer study. *Computational Statistics and Data Analysis 111*, 88–101.

Zhu, Z., S. M. Rezayat Sorkhabadi, Y. Gu, and W. Zhang (2022). Invariant extended kalman filtering for human motion estimation with imperfect sensor placement. In *2022 American Control Conference (ACC)*, pp. 3012–3018.

Álvarez, M., D. Luengo, and N. D. Lawrence (2009, 16–18 Apr). Latent force models. In D. van Dyk and M. Welling (Eds.), *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, Volume 5 of *Proceedings of Machine Learning Research*, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, pp. 9–16. PMLR.

Álvarez, M. A., L. Rosasco, and N. D. Lawrence (2011). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning 4*, 195–266.

université PARIS-SACLAY | ÉCOLE DOCTORALE de mathématiques Hadamard (EDMH)

Titre: Analyse de Données Fonctionnelles et Analyse de Formes dans le cadre de Frenet-Serret : Application à l'Analyse de Trajectoires de Mouvement de Langue des Signes
Mots clés: Analyse de Données Fonctionnelles, Analyse de Forme, Cadre de Frenet-Serret, Analyse de Trajectoires de Mouvement, Langues des Signes, Groupes de Lie

Résumé: Cette thèse vise à déterminer le cadre mathématique le plus adapté et des descripteurs pertinents pour l'analyse des trajectoires de mouvement en langue des signes. En nous appuyant sur les principes du contrôle moteur, nous avons identifié le cadre défini par les formules de Frenet-Serret comme particulièrement pertinent pour cette tâche. Ainsi, en introduisant de nouvelles approches d'analyse de courbes basées sur le cadre de Frenet, cette thèse contribue au développement de nouvelles méthodes dans les domaines de l'analyse de données fonctionnelles et de l'analyse de forme. La première partie de ce travail aborde le défi de l'estimation lisse des paramètres de courbures de Frenet, en traitant le problème comme une estimation de paramètres d'une équation différentielle dans $SO(d), (d \geq 1)$. Nous introduisons un algorithme EM fonctionnel qui définit une méthode d'estimation unifiée des variables dans le groupe $SE(3)$, fournissant des estimateurs lisses, plus fiables et robustes que les méthodes existantes.
Dans la deuxième partie, deux nouvelles représentations des courbes lisses dans $\mathbb{R}^d$ sont introduites, dont la Square Root Curvatures (SRC) transform, établissant un nouveau cadre géométrique riemannien qui utilise les informations géométriques d'ordre supérieur et dépend de la paramétrisation, surpassant alors la représentation state-of-the-art Square-Root Velocity Function (SRVF) sur des résultats synthétiques. Étant donné une collection de courbes, ce type de géométrie nous permet de définir des critères statistiques efficaces pour estimer les formes moyennes de Karcher sur les espaces de formes riemanniens associés, qui se révèlent particulièrement performants sur des données bruitées. Enfin, ce cadre développé ouvre la voie à des applications plus pratiques dans le traitement de la langue des signes, comprenant l'étude des lois puissances sur nos données et le développement d'un modèle génératif pour le mouvement d'un point en langue des signes.

Title: Functional and Shape Data Analysis under the Frenet-Serret Framework: Application to Sign Language Motion Trajectories Analysis
Keywords: Functional Data Analysis, Shape Analysis, Frenet-Serret Framework, Analysis of Motion Trajectories, Sign Language, Lie groups

Abstract: This thesis aims to determine the optimal mathematical framework and relevant descriptors for analyzing sign language motion trajectories. Drawing on principles of motor control, we identified the framework defined by the Frenet-Serret formulas as particularly suitable for this task. By introducing new curve analysis approaches based on the Frenet framework, this thesis contributes to developing novel methods in functional data analysis and shape analysis. The first part of this thesis addresses the challenge of smoothly estimating Frenet curvature parameters, treating the problem as parameter estimation of differential equation in $SO(d), (d \geq 1)$. We introduce a functional EM algorithm that defines a unified variable estimation method in the $SE(3)$ group, providing smoother estimators that are more reliable and robust than existing methods.
In the second part, two new representations of smooth curves in $\mathbb{R}^d, (d \geq 1)$ are introduced, including the Square Root Curvatures (SRC) transform, establishing a new Riemannian geometric framework. It outperforms the state-of-the-art Square-Root Velocity Function (SRVF) representation on synthetic results using higher-order geometric information and parametrization dependence. Given a collection of curves, this type of geometry allows us to define efficient statistical criteria for estimating Karcher mean shapes on the associated Riemannian shape spaces, proving particularly effective on noisy data. Finally, this developed framework opens the door to more practical applications in sign language processing, including the study of power laws on our data and the development of a generative model for a point motion in sign language.