



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

ANALYSIS OF DIFFERENT AVERAGING
TECHNIQUES TO DESCRIBE THE KINEMATICS OF
HUMAN OCCUPANTS IN FRONTAL
DECELERATIONS

Autor: Alicia Seguí Corbatón

Director: Francisco J. López Valdés

Madrid

Julio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
“ANALYSIS OF DIFFERENT AVERAGING TECHNIQUES TO DESCRIBE THE
KINEMATICS OF HUMAN OCCUPANTS IN FRONTAL DECELERATIONS”
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2024-2025 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio
de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.



Fdo.: Alicia Seguí Corbatón Fecha: 20/7/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Francisco J. López Valdés Fecha: 20/7/2025



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

ANALYSIS OF DIFFERENT AVERAGING
TECHNIQUES TO DESCRIBE THE KINEMATICS OF
HUMAN OCCUPANTS IN FRONTAL
DECELERATIONS

Autor: Alicia Seguí Corbatón

Director: Francisco J. López Valdés

Madrid

Julio de 2025

ANÁLISIS DE DIFERENTES TÉCNICAS DE PROMEDIO PARA DESCRIBIR LA CINEMÁTICA DE OCUPANTES EN DECELERACIONES FRONTALES

Autora: Seguí Corbatón, Alicia.

Director: López-Valdés, Francisco J.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

El objetivo de este proyecto es analizar y comparar tres métodos de estandarización de trayectorias (B-splines con Análisis de Componentes Principales (ACP), análisis postural para el desplazamiento máximo y reparametrización por longitud de arco con registro de señales) para caracterizar la variabilidad del movimiento de los ocupantes durante deceleraciones frontales a baja velocidad. Los resultados demuestran que estas técnicas pueden identificar eficazmente algunas diferencias cinemáticas entre voluntarios jóvenes y mayores, aunque el tamaño limitado de la muestra restringe la robustez estadística. La aplicación combinada de estos enfoques ofrece un marco de desarrollo para futuros estudios biomecánicos.

Palabras clave: biomecánica, ensayos con voluntarios, deceleración frontal, análisis de trayectorias, modelado estadístico, variabilidad relacionada con la edad.

1. Introducción

Cada día, millones de personas viajan en automóvil, confiando en que los cinturones de seguridad y los airbags los protegerán en caso de accidente. Detrás de estos sistemas de seguridad hay décadas de investigación sobre cómo se mueve el cuerpo humano durante los accidentes. Tradicionalmente, los maniquíes de pruebas de choque y los modelos computacionales han ayudado a los ingenieros a diseñar vehículos más seguros. Sin embargo, estos no son capaces de capturar completamente cómo se mueven las personas reales, con diferentes edades y formas corporales. Esto es especialmente relevante para los ancianos, que son más vulnerables a las lesiones y cuyos cuerpos responden de manera diferente en los accidentes.

Para mejorar la seguridad en vehículos, especialmente con una población cada vez más envejecida, se necesitan mejores formas de analizar y comprender cómo se mueve el cuerpo humano en situaciones que se acerquen a accidentes reales. Este trabajo aborda dicho desafío aplicando y comparando tres métodos analíticos que estudian datos de movimiento de voluntarios sometidos a simulaciones suaves de choques a baja velocidad.

La tesis se centra en desarrollar y probar un conjunto de métodos de análisis para comprender mejor cómo se mueven las personas durante estos ensayos. Se basa en datos de movimiento recogidos de trece voluntarios varones (nueve adultos jóvenes y cuatro adultos mayores). Durante cada prueba, marcadores reflectantes rastrearon el movimiento de puntos clave en la mitad superior del cuerpo, capturando información detallada sobre cómo respondían la cabeza, el cuello y el torso. El objetivo fue aplicar tres enfoques analíticos diferentes a estos datos, cada uno diseñado para simplificar el movimiento complejo identificando los patrones más significativos. Estos métodos se utilizaron para explorar cómo varía el movimiento entre individuos y para evaluar si características como la edad, la altura o la masa corporal influyen en la forma en que alguien se mueve durante un evento de choque. Comparando las fortalezas

y limitaciones de cada método, el proyecto espera contribuir al desarrollo de sistemas de retención vehicular más seguros e inclusivos.

2. Metodología

El análisis se centró en tres métodos complementarios diseñados para interpretar el movimiento registrado de los voluntarios durante la deceleración.

- **Método 1: Aproximación con B-Splines y Análisis de Componentes Principales de trayectorias individuales**

El primer método consistió en aproximar cada trayectoria de movimiento utilizando curvas denominadas B-splines, un enfoque similar al presentado por Samuels et al. en 2015. Esto ayudó a reducir el ruido y simplificar los datos. Posteriormente, se utilizó el ACP para identificar las principales formas en que estas trayectorias variaban entre los sujetos. Finalmente, se aplicaron modelos de regresión para analizar qué cantidad de esta variación podía asociarse a factores como la edad, la altura o la masa corporal.

- **Método 2: Análisis de la posición relativa de los marcadores en el instante de desplazamiento máximo**

El segundo método se centró en un único momento de la deceleración: el fotograma en que la cabeza alcanzó su posición más adelantada. En ese instante clave, se analizaron las posiciones relativas de los puntos anatómicos mediante ACP, junto con cálculos de ángulos articulares y distancias entre puntos corporales. Esto permitió identificar cómo diferían la postura y la alineación corporal entre individuos y grupos de edad.

- **Método 3: Reparametrización por Longitud de Arco y Registro de Señales para generación de trayectorias promedio y corredores de confianza**

El último método analizó la trayectoria completa de movimiento de cada marcador, pero en lugar de hacerlo fotograma a fotograma, alineó las trayectorias según la longitud de arco, una forma de describir la forma del recorrido independientemente del tiempo. Utilizando esta técnica, se generaron trayectorias promedio y corredores de variabilidad, proporcionando una visión más clara de las tendencias generales y diferencias entre grupos de edad. Esto se implementó usando la herramienta de código ARCGen desarrollada por Hartlen y Cronin en 2022.

En conjunto, los tres métodos ofrecen perspectivas complementarias sobre el movimiento de los ocupantes, alcanzando una comprensión más completa de cómo se desplazan las personas durante impactos frontales a baja velocidad.

La Figura 1.1-1. muestra un diagrama esquemático que resume el flujo de trabajo seguido.

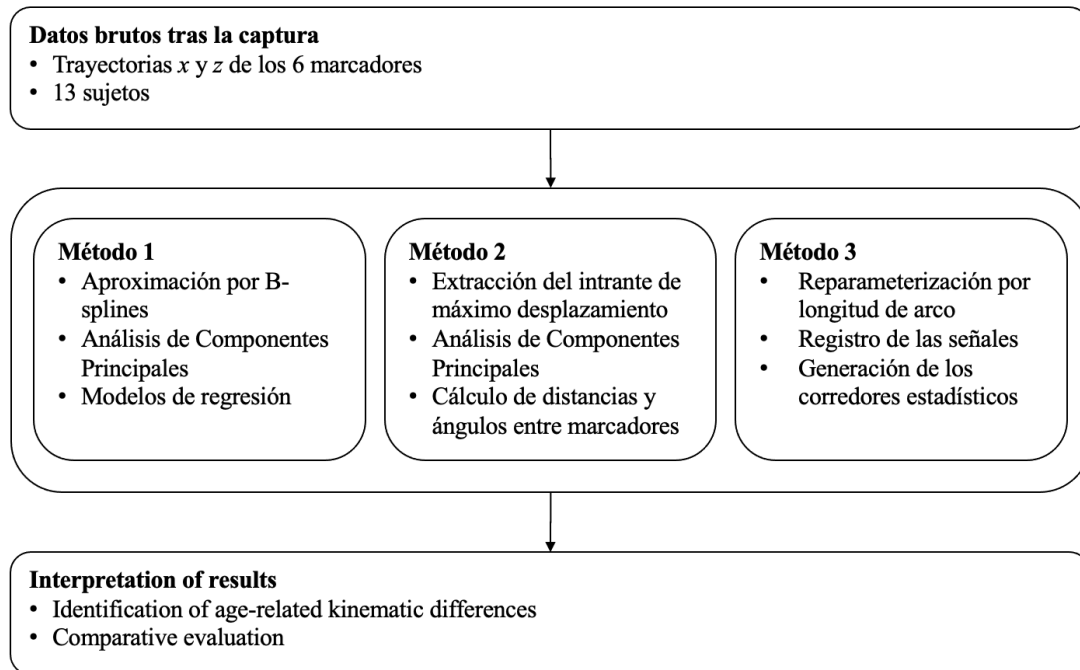


Figura 1.1-1. Esquema del flujo metodológico

3. Resultados

Los resultados obtenidos con los tres métodos analíticos revelan perspectivas complementarias sobre cómo se mueven los individuos durante deceleraciones frontales a baja velocidad, con énfasis en las diferencias entre voluntarios jóvenes y mayores.

• Método 1

Este método capturó cómo se movían los marcadores anatómicos individualmente durante la deceleración, comprimiendo sus trayectorias mediante B-splines y analizándolas con ACP. Mientras que los principales patrones de movimiento (PC1) no reflejaron fuerte vinculación a las características de los sujetos, el segundo componente del ACP (PC2) reveló diferencias sutiles pero consistentes entre participantes jóvenes y mayores. En particular, los marcadores torácicos como T1 y T4 mostraron valores de PC2 significativamente mayores en el grupo de mayores, lo que sugiere respuestas posturales distintas. El tamaño limitado de la muestra y el modelado univariante limitaron en cierta medida la solidez estadística de los resultados.

• Método 2

Analizando la configuración corporal en el momento de máximo desplazamiento de la cabeza, este método reveló diferencias posturales relacionadas con la edad. Mientras que los componentes principales capturaron la variación general de la forma, pero sin vínculos claros con la edad o la antropometría, el tercer componente (PC3) se asoció significativamente al grupo de edad. Los participantes mayores mostraron diferencias sistemáticas en la alineación espinal, evidenciadas también por un ángulo T4 reducido (indicativo de mayor flexión torácica) y una tendencia hacia la extensión cervical.

- **Método 3**

Este método alineó las trayectorias completas de cada marcador según su forma y generó curvas promedio de desplazamiento con corredores de variabilidad para cada grupo de edad. Reveló que los sujetos mayores tendían a presentar un desplazamiento hacia adelante consistentemente menor, especialmente en los marcadores torácicos como T1, T4 y T8, y seguían trayectorias más ascendentes en comparación con los jóvenes. Además, la variabilidad en el movimiento de los mayores fue mayor, especialmente en las etapas finales del movimiento, lo que sugiere menor consistencia en la resolución de la deceleración. Un ejemplo de esta transformación se muestra en la Figura 1.1-2, que presenta las trayectorias brutas de todos los voluntarios para un marcador de ejemplo (T4) y la trayectoria media resultante con el corredor de variabilidad tras el procesamiento basado en longitud de arco.

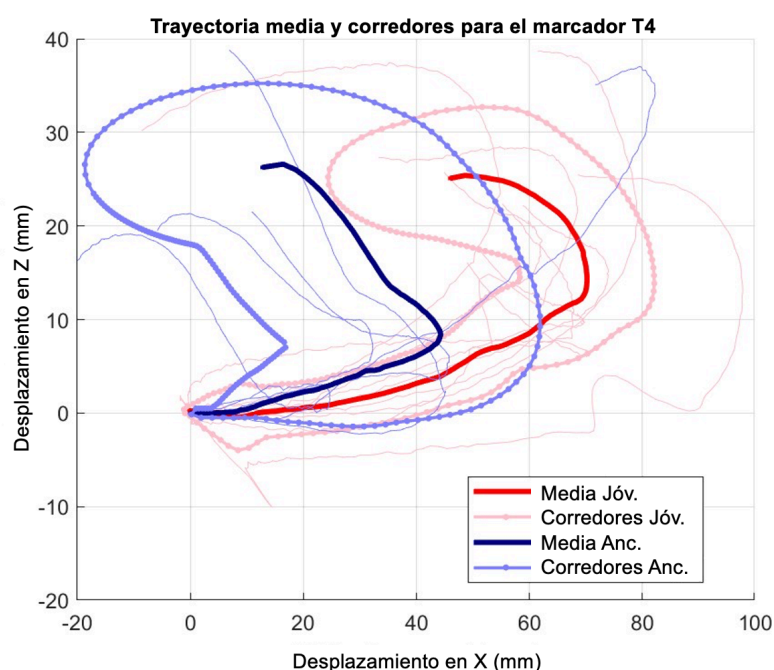


Figura 1.1-2. Ejemplo de procesamiento de trayectorias para el marcador T4. Trayectorias brutas de todos los sujetos y trayectoria media y corredores de variabilidad obtenidos tras la reparametrización por longitud de arco y registro de señales para ambos grupos de voluntarios.

En conjunto, cada método aportó una perspectiva diferente para analizar los datos: el primero destacó tendencias continuas de las trayectorias y sus predictores físicos, el segundo capturó características posturales estáticas en un instante crítico, y el tercero resaltó la variabilidad y los patrones a nivel de grupo a lo largo de toda la longitud de movimiento.

4. Conclusiones

Los tres métodos analíticos desarrollados y aplicados en esta tesis (compresión B-spline con ACP, análisis postural basado en el tiempo y registro de trayectorias por longitud de arco) demostraron ser eficaces para simplificar e interpretar la compleja cinemática observada durante deceleraciones frontales a baja velocidad. En conjunto, proporcionaron una visión multifacética de cómo se mueven los individuos en tales escenarios, revelando tanto tendencias generales como diferencias relacionadas con la edad en la postura y el movimiento.

Más importante aún, este trabajo ha demostrado el potencial de estas técnicas como herramientas para el análisis biomecánico. A pesar de las limitaciones inherentes a un tamaño de muestra pequeño, que restringieron la solidez estadística y la aplicabilidad general de algunos hallazgos, los métodos produjeron resultados interpretables y anatómicamente fundamentados de manera consistente. Con acceso a conjuntos de datos más grandes y diversos, estos mismos enfoques podrían ofrecer conocimientos más definitivos sobre el comportamiento de los ocupantes, respaldar mejoras en el diseño de sistemas de seguridad y aumentar la fidelidad de sustitutos humanos como los maniquíes de pruebas y los modelos virtuales.

Así, esta tesis sienta las bases para el desarrollo de herramientas analíticas capaces de capturar todo el espectro de variabilidad humana en la respuesta al movimiento. A medida que evoluciona el diseño de la seguridad vehicular, tener en cuenta las diferencias de edad, tamaño corporal y postura es imprescindible. Los tres métodos exploran caminos prometedores, ofreciendo enfoques estructurados y escalables para traducir datos complejos de movimiento en conclusiones significativas que puedan respaldar la próxima generación de modelos biomecánicos y sistemas de seguridad.

5. Referencias

- [SAMU15] Samuels, M.A., Seacrist, T., Huang, S., Balasubramanian, S., Lopez-Valdes, F.J., Kent, R.W., Arbogast, K.B., "Modeling spatial trajectories in dynamics testing using basis splines: application to tracking human volunteers in low-speed frontal impacts", *Computer Methods in Biomechanics and Biomedical Engineering*, 18:12, pp. 1323–1332, 2015.
- [HART22] Hartlen D.C., Cronin D.S., "Arc-Length Re-Parametrization and Signal Registration to Determine a Characteristic Average and Statistical Response Corridors of Biomechanical Data", *Frontiers in Bioengineering and Biotechnology*, vol. 10, 2022.

ANALYSIS OF DIFFERENT AVERAGING TECHNIQUES TO DESCRIBE THE KINEMATICS OF HUMAN OCCUPANTS IN FRONTAL DECELERATIONS

Author: Seguí Corbatón, Alicia.

Director: López-Valdés, Francisco J.

Collaborating Entity: ICAI – Universidad Pontificia Comillas.

ABSTRACT

The aim of this project is to analyze and compare three trajectory standardization methods (B-splines with PCA, peak displacement posture analysis, and arc-length reparameterization and signal registration) to characterize occupant motion variability during low-speed frontal decelerations. The results demonstrate that these techniques can effectively identify some kinematic differences between young and elderly volunteers, although the limited sample size constrains statistical robustness. The combined application of these approaches offers a comprehensive framework for future biomechanical studies.

Keywords: biomechanics, volunteer testing, frontal deceleration, trajectory analysis, statistical modeling, age-related variability.

1. Introduction

Every day, millions of people travel in cars, trusting that seatbelts and airbags will protect them in the event of a crash. Behind these safety systems lies decades of research into how the human body moves during collisions. Traditionally, crash test dummies and computer models have helped engineers design safer vehicles, but they can't fully capture how real people, with different ages and body types, actually move. This is especially important for older adults, who are more vulnerable to injury and whose bodies respond differently in crashes.

To improve vehicle safety for everyone, especially an aging population, we need better ways to analyze and understand how human bodies move in realistic crash-like situations. This thesis explores that challenge by applying and comparing three advanced analytical methods to study motion data from volunteers subjected to gentle, low-speed crash simulations.

The thesis focuses on developing and testing a set of analysis methods to better understand how people move during said low-speed frontal car crashes. The study is based on motion data collected from thirteen male volunteers (nine young adults and four older adults). During each test, reflective markers tracked the movement of key points on the upper body, capturing detailed information about how the head, neck, and torso responded. The aim was to apply three different analytical approaches to this data, each designed to simplify the complex motion while preserving meaningful patterns. These methods were then used to explore how movement varies across individuals and to assess whether characteristics like age, height, or body mass influence how someone moves during a crash-like event. By comparing the strengths and limitations of each method, the project hopes to support the development of safer, more inclusive vehicle restraint systems.

2. Methodology

The analysis focused on three complementary methods designed to interpret the recorded motion of the volunteers during deceleration.

- **Method 1: B-Spline approximation and Principal Component Analysis of individual landmark trajectories**

The first method involved approximating each motion trajectory using smooth curves called B-splines, an approach similar to the one presented in Samuels et al. in 2015. This helped reduce noise and simplify the data. PCA was then used to identify the main ways in which these trajectories varied between subjects. Finally, regression models were applied to see how much of this variation could be linked to factors like age, height, or body mass.

- **Method 2: Landmark relative positioning analysis at peak displacement**

The second method focused on a single moment during the deceleration: the frame in which the head reached its furthest forward position. At this key instant, the relative positions of anatomical landmarks were analyzed using PCA, along with calculations of joint angles and distances between body points. This helped identify how posture and body alignment differed across individuals and age groups.

- **Method 3: Arc-Length Re-Parameterization and Signal Registration for generation of average trajectories and confidence corridors**

The last method looked at the full trajectory of motion for each landmark, but instead of analyzing it frame by frame, it aligned the trajectories based on arc length, a way of describing the shape of the path regardless of timing. Using this technique, average trajectories and variability corridors were generated, offering a clearer view of general trends and differences between age groups. This was implemented using the open-source ARCGen toolbox developed by Hartlen and Cronin in 2022.

Together, these three methods provide different but complementary perspectives on occupant motion, allowing for a more complete understanding of how people move during low-speed frontal impacts.

Figure 1.1-3 displays a schematic summary diagram of the workflow followed.

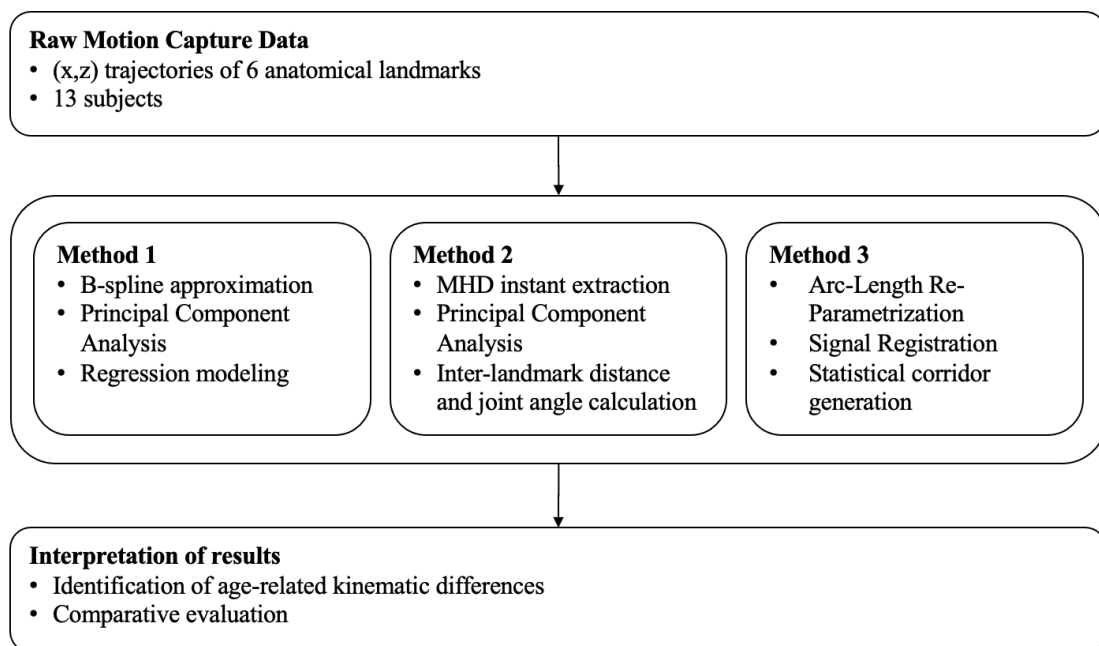


Figure 1.1-3. Schematic of the methodological framework

3. Results

The results obtained from the three analytical methods reveal complementary insights about how individuals move during low-speed frontal decelerations, with particular emphasis on differences between younger and older volunteers.

- **Method 1**

This method captured how individual body landmarks moved during deceleration by compressing their trajectories using B-splines and analyzing them with PCA. While the main motion patterns (PC1) were not strongly linked to subject characteristics, the second PCA component (PC2) revealed subtle but consistent differences between young and elderly participants. Notably, thoracic landmarks like T1 and T4 showed significantly higher PC2 values in the elderly group, suggesting distinct postural responses. The limited sample size and univariate modeling somewhat constrained the statistical strength of the results.

- **Method 2**

By analyzing the body configuration at the moment of maximum head displacement, this method revealed age-related differences in static posture. While the main PCA components captured general shape variation without clear links to age or anthropometry, the third component (PC3) was significantly associated with age group. Elderly participants showed systematic differences in spinal alignment, supported by a reduced T4 angle (indicative of greater thoracic flexion) and a trend toward cervical extension.

- **Method 3**

This method aligned the full motion trajectories of each landmark based on their shape and generated average displacement curves with variability corridors for each age group. The results showed that elderly subjects had consistently reduced forward motion, particularly at thoracic landmarks like T1, T4, and T8, and followed more upward-curving paths compared to younger individuals. Additionally, the variability in elderly motion was greater, especially during the later stages of the movement, suggesting less consistency in how deceleration was resolved. An example of this transformation is presented in Figure 1.1-4, which displays the raw motion paths of all volunteers for an example landmark (T4) and the resulting mean trajectory with variability envelope after arc-length-based processing.

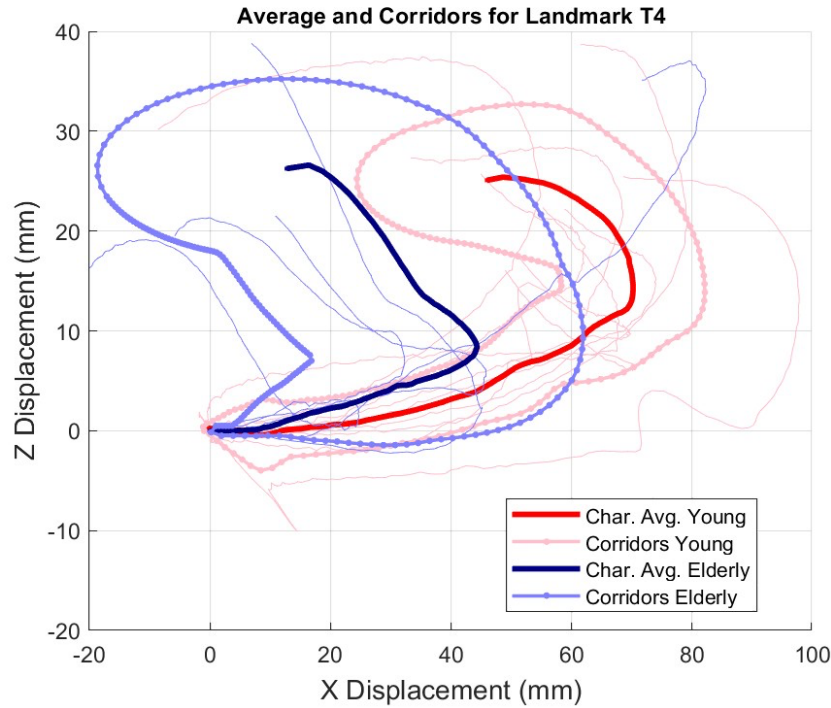


Figure 1.1-4 Example of trajectory processing for the T4 landmark. Raw motion paths from all subjects and mean trajectory and variability corridors obtained after arc-length re-parameterization and signal registration for both groups of volunteers.

Overall, each method contributed a different lens through which to view the data: the first emphasized continuous trajectory trends and physical predictors, the second captured static postural features at a critical instant, and the third highlighted variability and group-level patterns across the full motion length.

4. Conclusions

The three analytical methods developed and applied in this thesis (B-spline compression with PCA, time-based posture analysis, and arc-length trajectory registration) each proved effective in simplifying and interpreting the complex kinematics observed during low-speed frontal decelerations. Together, they provided a multifaceted view of how individuals move in such scenarios, revealing both general trends and age-related distinctions in posture and motion.

More importantly, this work demonstrated the potential of these techniques as tools for biomechanical analysis. Despite the inherent limitations of a small sample size, which constrained the statistical strength and broader applicability of some findings, the methods consistently produced interpretable and anatomically grounded outputs. With access to larger and more diverse datasets, these same approaches could offer more definitive insights into occupant behavior, support improvements in safety system design, and enhance the fidelity of human surrogates such as crash test dummies and virtual models.

Thus this thesis lays the groundwork to developing analytical tools capable of capturing the full spectrum of human variability in motion response. As vehicle safety design evolves, accounting for differences in age, body size, and posture is necessary. The methods explored here demonstrate a promising path forward, offering structured, scalable approaches for

translating complex motion data into meaningful insights that can support the next generation of biomechanical models and safety systems.

5. References

- [SAMU15] Samuels, M.A., Seacrist, T., Huang, S., Balasubramanian, S., Lopez-Valdes, F.J., Kent, R.W., Arbogast, K.B., "Modeling spatial trajectories in dynamics testing using basis splines: application to tracking human volunteers in low-speed frontal impacts", *Computer Methods in Biomechanics and Biomedical Engineering*, 18:12, pp. 1323–1332, 2015.
- [HART22] Hartlen D.C., Cronin D.S., "Arc-Length Re-Parametrization and Signal Registration to Determine a Characteristic Average and Statistical Response Corridors of Biomechanical Data", *Frontiers in Bioengineering and Biotechnology*, vol. 10, 2022.

CONTENTS

1.	<i>Introduction</i>	27
1.1.	Context & Problem Statement	27
1.2.	Motivation	28
1.3.	Objectives	29
1.3.1.	Objective 1	29
1.3.2.	Objective 2	29
1.3.3.	Objective 3	29
1.3.4.	Objective 4	30
2.	<i>Experimental data</i>	31
2.1.	Background on Volunteer Testing	31
2.1.1.	Rationale for Volunteer Testing	31
2.1.2.	Anthropomorphic Test Devices (ATDs): Utility and Limitations	31
2.1.3.	Comparison with Other Test Subjects (PMHS, children, etc.)	32
2.1.4.	Ethical Framework	32
2.1.5.	Evolution of Volunteer Test Platforms	33
2.2.	Volunteer Testing Protocol	34
2.2.1.	Test Setup	34
2.2.2.	Safety and Ethics	36
2.2.3.	Instrumentation and Data Acquisition	36
2.2.4.	Volunteer Sample and Preparation	39
3.	<i>Methodology</i>	43
3.1.	Software Environment	43
3.2.	Method 1: B-Spline approximation and Principal Component Analysis of individual landmark trajectories	44
3.2.1.	Method Overview	44
3.2.2.	Literature background	45
3.2.3.	Purpose and justification	47
3.2.4.	Mathematical background	47
3.2.5.	Implementation	52
3.3.	Method 2: Landmark relative positioning analysis at peak displacement	60
3.3.1.	Method overview	60
3.3.2.	Literature background	61
3.3.3.	Purpose and justification	62

3.3.4.	Mathematical background.....	63
3.3.5.	Implementation	65
3.4.	Method 3: Arc-Length Re-Parameterization and Signal Registration for generation of average trajectories and deviation corridors	68
3.4.1.	Method Overview	68
3.4.2.	Literature background.....	69
3.4.3.	Purpose and justification.....	70
3.4.4.	Mathematical background.....	70
3.4.5.	Implementation	73
4.	Results	77
4.1.	Method 1: B-Spline approximation and Principal Component Analysis of individual landmark trajectories	77
4.1.1.	Spline fitting and dimensionality reduction.....	77
4.1.2.	Principal Component Analysis outcomes	80
4.1.3.	Regression modeling results	85
4.2.	Method 2: Landmark relative positioning analysis at peak displacement.....	87
4.2.1.	Frame selection and posture vector extraction	87
4.2.2.	PCA Outcomes	89
4.2.3.	Regression Modeling Results	90
4.2.4.	Posture Metrics: Inter-Landmark Distances and Joint Angles	91
4.3.	Method 3: Arc-Length Re-Parameterization and Signal Registration for generation of average trajectories and deviation corridors	95
4.3.1.	Group-Specific ARCGen Outputs	95
4.3.2.	Landmark 1: Head	96
4.3.3.	Landmark 2: C4	97
4.3.4.	Landmark 3: T1	98
4.3.5.	Landmark 4: T4	99
4.3.6.	Landmark 5: T8	100
4.3.7.	Landmark 6: H Point.....	101
4.3.8.	Summary of observed patterns	102
5.	Discussion	103
5.1.	Method 1: B-Spline approximation and Principal Component Analysis of individual landmark trajectories	103
5.1.1.	Key Findings and interpretation	103
5.1.2.	Limitations and Potential Improvements.....	103
5.2.	Method 2: Landmark relative positioning analysis at peak displacement.....	105

5.2.1.	Key Findings and Interpretation	105
5.2.2.	Limitations and Potential Improvements	106
5.3.	Method 3: Arc-Length Re-Parameterization and Signal Registration for generation of average trajectories and deviation corridors	108
5.3.1.	Key Findings and Interpretation	108
5.3.2.	Limitations and Potential Improvements	108
5.4.	Method comparison	109
5.4.1.	Dimensional focus and granularity	109
5.4.2.	Sensitivity to inter-subject variation	110
5.4.3.	Interpretability and Biomechanical Meaning	110
5.4.4.	Robustness and Limitations	110
5.4.5.	Summary of Strengths and Limitations	111
5.5.	Real-World Applications	113
6.	Conclusion	115
6.1.	Degree of Objective Fulfillment	116
6.2.	Future Work	117
7.	References	119

LIST OF FIGURES

Figura 1.1-1. Esquema del flujo metodológico.....	7
Figura 1.1-2. Ejemplo de procesamiento de trayectorias para el marcador T4. Trayectorias brutas de todos los sujetos y trayectoria media y corredores de variabilidad obtenidos tras la reparametrización por longitud de arco y registro de señales para ambos grupos de voluntarios.....	8
Figure 1.1-3. Schematic of the methodological framework.....	12
Figure 1.1-4 Example of trajectory processing for the T4 landmark. Raw motion paths from all subjects and mean trajectory and variability corridors obtained after arc-length re-parameterization and signal registration for both groups of volunteers.	14
Figure 2.2-1, Sled Test Configuration, showing the rigid seat, flexible backrest, footrest, and three-point seatbelt system. (LOPE17)	35
Figure 2.2-2. Sled Deceleration Pulse – Sled deceleration pulse corridors in the young (blue) and elderly (green) groups. Solid lines are the average deceleration within the group. Shaded area corresponds to the one standard deviation corridor. (VIVE21)	35
Figure 2.2-3. Head Mount - Detail of the 6 degree-of-freedom head cube and of the position of the sensors on the head of one of the volunteers. (VIVE21)	36
Figure 2.2-4. Instrumentation Layout - Example of the experimental setup showing the placement of load cells, accelerometers, and reflective markers on the sled platform and volunteer. (LOPE17).....	38
Figure 2.2-5. Lateral Video Comparison - Representative still frames comparing the motion of a young and an elderly volunteer during frontal deceleration. (LOPE17)	41
Figure 3.2-1. Raw x–z trajectories of a selected anatomical landmark (T1) across all test trials. The trajectories are high-resolution and exhibit significant inter-subject variability, motivating the need for a compact, standardized representation such as B-spline approximation.	53
Figure 3.2-2. Example of original trajectory (red line) and fitted B-spline curve (blue line) for the T1 landmark during a representative trial. The fitted curve preserves the essential shape of the motion with reduced data complexity.....	55
Figure 3.2-3. Projection of T1 landmark results onto the first two principal components. Each point represents one test trial, color-coded by subject group (young vs. elderly). While group tendencies begin to emerge, overlap remains high and interpretation is non-quantitative.....	57

Figure 3.2-4. 3D scatter plot of PCA scores on components 1–3 for the T1 landmark. The projection provides additional insight into the distribution of trajectory variation but does not offer conclusive group separability.....	58
Figure 3.3-1. Example of landmark trajectories during a test, with the frame of maximum head displacement (MHD) highlighted in black. Dark gray and light gray indicate frames before and after MHD, respectively.....	66
Figure 3.4-1. Three biomechanical response signals before arc-length re-parameterization and registration. The signals display misaligned peaks and valleys, resulting in a less representative average response.....	72
Figure 3.4-2. The same signals after applying arc-length re-parameterization and registration. Temporal alignment of key features is improved while preserving individual shape characteristics.....	73
Figure 3.4-3. Example ARCGen outputs for the Head landmark. Average trajectories and 95% statistical corridors are shown for young (red) and elderly (blue) groups.....	75
Figure 4.1-1. Raw X–Z trajectories of the five anatomical landmarks (Head, C4, T1, T4, and T8) for each trial. Displacements have been zeroed relative to the initial Head position.....	77
Figure 4.1-2. Original (x, z) motion trajectories of all six anatomical landmarks for a representative trial. Each plot shows the full deceleration interval for a single landmark, as recorded at 1,000 Hz.....	78
Figure 4.1-3. Same trajectories, now overlaid with cubic B-spline fits. The splines replicate the overall shape of each signal while filtering noise and enabling dimensionality reduction.....	79
Figures 4.1-4. Projections of subject trajectories onto the first two principal components for all landmarks, individually.....	84
Figure 4.2-1. Landmark trajectories for Subject 1694 with the frame of maximum head displacement (MHD) highlighted in black. Dark gray and light gray lines indicate frames before and after MHD, respectively. Each frame connects anatomical landmarks in the sagittal plane.....	88
Figure 4.2-2 Posture configuration of Subject 1694 at the frame of maximum head displacement, expressed relative to the T8 landmark. Coordinates represent sagittal-plane displacement, with landmark labels indicating anatomical identity.	89
Figure 4.2-3. Posture of all subjects at maximum head displacement, re-referenced to T8, shown as a sequence of line segments connecting Head to T8. Red: young group; blue: elderly group.	92

Figure 4.3-1. ARCGen outputs for the Head landmark. Average trajectories and 95% statistical corridors are shown for young (red) and elderly (blue) groups. Greater variability is observed in the elderly group during the central portion of the arc-length path.	96
Figure 4.3-2. ARCGen outputs for the C4 landmark. Average trajectories and 95% statistical corridors are shown for young (red) and elderly (blue) groups. The elderly group displays increased dispersion during the central arc-length segment.	97
Figure 4.3-3. ARCGen outputs for the T1 landmark. The elderly group (blue) shows increased dispersion across the entire arc-length range compared to the young group (red), whose trajectories remain more tightly clustered.....	98
Figure 4.3-4. ARCGen outputs for the T4 landmark. The elderly group (blue) exhibits lower forward and vertical excursion but a broader statistical corridor compared to the young group (red).....	99
Figure 4.3-5. ARCGen outputs for the T8 landmark. The elderly group (blue) shows a shorter forward excursion, accompanied by increased variability compared to the young group (red).....	100
Figure 4.3-6. ARCGen outputs for the H-Point landmark. Individual trajectories are highly variable, especially in the young group. The elderly group shows a more vertically oriented average path and broader dispersion overall.	101
Figure 5.5-1. Hypothetical seating configurations for fully automated vehicles (FAVs) used in the survey by Koppel et al. to explore occupant preferences under various travel scenarios (KOPP19).	113

LIST OF TABLES

Table 2.2-1. Instrumentation Summary - Overview of the sensors used in the study, including their type, mounting location, and measurement recorded.	37
Table 2.2-2. Volunteer Anthropometric Data - Summary of the anthropometric characteristics of all participating volunteers.....	39
Table 3.2-1. Example of individual and cumulative explained variance for the T1 landmark. The first two or three components account for the majority of the total variance, making the rest largely redundant for modeling purposes.	56
Table 4.1-1. RMSE Summary Table - Mean RMSE between original trajectories and B-spline approximations, averaged over all trials. Evaluation performed on 10-point uniform resampling of each trajectory.	80
Table 4.1-2. Cumulative variance explained (%) by the first three principal components for each anatomical landmark.	81
Table 4.1-3. Best one-variable regression models for each landmark and principal component. Grey shading indicates $p < 0.05$	85
Table 4.2-1. Variance explained by the first 3 principal components extracted from posture vectors at MHD.	90
Table 4.2-2. Summary of best single-variable regression models for PC1, PC2, and PC3 derived from posture vectors at MHD. Grey shading indicates $p < 0.05$	91
Table 4.2-3. Inter-landmark distances (mm) at MHD. Values represent mean $\pm$ standard deviation per group. P-values from two-sample t-tests included to assess group differences. Grey shading indicates $p < 0.05$	93
Table 4.2-4. Internal joint angles ($^{\circ}$) at MHD. Values represent mean $\pm$ standard deviation. P-values are based on Welch's t-test (no assumption of equal variance). Grey shading indicates $p < 0.05$	93
Table 5.4-1. Comparative summary table of the three analytical methods.....	112

1. INTRODUCTION

1.1. Context & Problem Statement

In the field of impact biomechanics, understanding the mechanical response of the human body during vehicle collisions is a fundamental concern. Experimental volunteer tests, post-mortem human surrogate (PMHS) studies, and computational models have all contributed to major advancements in restraint systems and vehicle design. Yet, accurately capturing and analyzing human motion during impact events remains a challenge. Unlike anthropomorphic test devices (ATDs), human occupants exhibit substantial inter-subject variability: differences in age, morphology, flexibility, and neuromuscular response all influence how a person moves during an accident. This variability complicates both the development of generalizable restraint systems and the validation of human body models intended to simulate real-world occupants.

One of the main difficulties lies in how to meaningfully represent and compare the kinematic responses of different individuals subjected to the same loading conditions. Raw motion trajectories are often complex and noisy, and patterns of variation may be obscured by differences in timing, magnitude, or posture. As a result, there is a critical need for data processing techniques that can reduce variability while preserving biomechanically relevant information. Such techniques can allow researchers to extract consistent trends, highlight inter-individual differences, and support the formulation of generalized insights from inherently variable experimental data.

This need for robust methods for standardizing data from human responses is becoming increasingly relevant as populations age. Worldwide, life expectancy has risen due to advancements in medical technology, nutrition, and living conditions, leading to an increasing proportion of older individuals within the population. By 2050, the world's population of individuals aged 60 years and older will double, reaching 2.1 billion (WHO25). This demographic shift, combined with the fact that people are driving later into life (BABU19), highlights the importance of protecting older occupants in automotive crashes. Age-related changes such as decreased spinal flexibility, altered muscle tone, and delayed reaction times can affect occupant motion during frontal impacts. Moreover, injury tolerance tends to decline with age, making older adults more vulnerable to the same crash conditions that younger individuals might withstand. These factors underscore the importance of developing analysis methods capable of characterizing motion variability in a way that is sensitive to age-related differences.

To address this problem, this thesis explores and compares multiple methodological approaches for analyzing human kinematic data from low-speed frontal deceleration tests. Using a dataset involving both young and elderly volunteers subjected to controlled loading conditions, the aim is not only to manage variability but to understand it, identifying how different techniques capture inter-subject differences, and assessing their suitability for extracting meaningful biomechanical insights from heterogeneous data.

1.2. Motivation

Improving the safety and effectiveness of automotive restraint systems begins with addressing the significant variability in biomechanical responses among individuals. This variability, as previously outlined, poses a considerable challenge in the field of impact biomechanics. Traditional methods for standardizing responses fall short of accurately capturing the differences between volunteers. Ensuring that safety systems account for a wide range of individual responses is both a scientific and ethical imperative.

This research gains particular relevance in the context of the aging global population. The demographic trends highlighted earlier indicate a growing number of older individuals who are not only more vulnerable to injuries during crashes but also present unique biomechanical characteristics. Existing safety measures have been predominantly developed based on data from crash test dummies that represented younger or mid-aged adults, leaving a critical gap in understanding and accommodating the specific needs of elderly occupants. Addressing this gap is crucial for ensuring equitable safety standards and aligns with broader societal goals of inclusivity and fairness.

Beyond the demographic perspective, the project is primarily motivated by the inherent challenges of combining data from diverse human subjects into coherent and meaningful insights. Biomechanical experiments, especially those involving human volunteers, are often limited by ethical, practical, and technical constraints. The variability in response trajectories, whether due to physical differences, experimental noise, or both, can obscure meaningful patterns and hinder the development of predictive models. This makes it essential to explore and refine advanced statistical methods capable of synthesizing this variability into reliable insights.

Additionally, techniques like Principal Component Analysis (PCA), B-splines, and arc-length re-parameterization represent powerful tools for handling large datasets with inherent variability. However, each method has its limitations, and the ability to assess their relative strengths and weaknesses in a practical context is important for advancing the field. By evaluating and comparing these techniques, this thesis aims not only to contribute to the development of better biomechanical models but also to provide a framework for future research in areas where variability and individual characteristics play a central role.

In summary, this work is motivated by the dual need to enhance vehicle safety for all occupants, especially older individuals, and to address the methodological challenges of creating accurate and representative trajectory standardization methods. By exploring innovative approaches, this thesis aims to contribute to both the understanding of impact biomechanics and the practical application of this knowledge. The outcomes of this research may have the potential to guide the design of future restraint systems as well as broader efforts to bridge the gap between experimental data and real-world applications.

1.3. Objectives

The primary goal is to explore and compare three advanced methods for data standardization, aiming to find reliable insights regarding the forward displacement of selected anatomical landmarks of car occupants during a frontal impact. The methods under analysis are Principal Component Analysis (PCA) combined with B-splines for each anatomical landmark, Time Analysis of Landmark Relative Positions with PCA, and Arc-Length Re-Parameterization with Signal Registration. These techniques will be applied to a dataset that includes both young (mean age: 23 years old) and elderly (mean age: 72 years old) volunteers in low-speed frontal deceleration tests. The specific objectives of the thesis are as follows:

1.3.1. Objective 1.

B-Spline approximation and Principal Component Analysis of individual landmark trajectories

The first objective is to implement a method that combines PCA with cubic B-splines to analyze the spatial trajectories of anatomical landmarks during frontal impacts. By applying PCA to the B-splines of key landmarks, this method aims to identify the principal directions of variance across subjects with varying body sizes and ages. Additionally, regression modeling using subject characteristics as predictors will be employed to understand how these characteristics influence the kinematic trajectories.

1.3.2. Objective 2.

Landmark relative positioning analysis at peak displacement

The second objective involves implementing a method based on time analysis of the relative positions of anatomical landmarks, followed by PCA to identify the principal components of motion. Unlike the previous method, which treated each landmark independently over time, this approach captures the overall shape formed by the alignment of the head and spinal vertebrae at a specific instant. By doing so, it reflects the deformed posture of the upper body at the peak of motion. By applying PCA, major directions of variance will try to be identified in the movement of different anatomical regions. Regression models will later be developed using subject characteristics to try and find relationships with specific trajectories. Finally, this method is also expected to offer a more subtle understanding of how the body segments interact during a crash and how these interactions contribute to overall occupant displacement. For that, inter-landmark distances and joint angles will be studied. This will provide an alternative approach to standardization by focusing on the relative placement of different landmarks rather than the absolute positions.

1.3.3. Objective 3.

Arc-Length Re-Parameterization and Signal Registration for generation of average trajectories and confidence corridors

The third objective is to apply arc-length re-parameterization and signal registration method to the biomechanical data. This technique aims to improve the alignment of signals by adjusting for differences in signal shapes, particularly in complex oscillatory data. By re-parameterizing the signals based on arc-length and using signal registration to align key features such as peaks and valleys, this method seeks to reduce distortion and improve the accuracy of characteristic averages and response corridors. The goal is to assess how well this approach can handle the complex trajectories that are present in the dataset.

1.3.4. Objective 4.

Comparison and evaluation of each method

The fourth objective is to compare the three implemented methods in terms of their ability to standardize and analyze biomechanical data. This comparison will assess the effectiveness, strengths, and weaknesses of each method in identifying key motion patterns. The evaluation will focus on how well each approach handles variability across subjects with differing characteristics and how they contribute to an improved understanding of occupant displacement during a frontal impact.

2. EXPERIMENTAL DATA

2.1. Background on Volunteer Testing

2.1.1. Rationale for Volunteer Testing

Research into the biomechanics of human occupants during automotive impacts has traditionally relied on anthropomorphic test devices (ATDs) and post-mortem human subjects (PMHS). While these tools have provided invaluable data for injury threshold development and restraint system evaluation, they remain limited in their ability to capture the full spectrum of human kinematic responses. In particular, PMHS testing is inherently restricted to deceased individuals and does not allow for the study of active muscle responses, variability across repeated trials, or non-injurious motion patterns. ATDs, while repeatable and scalable, are simplified mechanical surrogates that fail to capture important individual-specific anatomical and physiological differences, especially those related to age, morphology, and tissue properties (SUN16).

In contrast, human volunteer testing offers a unique opportunity to study the dynamic behavior of live subjects in safe, controlled low-speed impacts. These tests allow for direct observation and measurement of head, neck, and torso motion, as well as real-time interaction with restraint systems. Volunteer studies are especially valuable in contexts where live human variability, such as that introduced by aging, body posture, or neuromuscular activation, is expected to play a significant role in occupant response. By focusing on low-severity deceleration pulses that pose no risk of injury, these studies can provide ethically sound and scientifically robust data for validating human body models and improving safety systems (VIVE21).

In addition to their ability to capture active responses, volunteer studies offer the advantage of tightly controlled experimental conditions while still reflecting real-world occupant behavior). Laboratory-based volunteer testing allows for precise manipulation of variables such as posture, awareness, and muscle activation, factors that are difficult to isolate in cadaveric or retrospective crash analyses. Although volunteer tests must remain below injury thresholds, they have proven essential for evaluating dynamic kinematic responses and for validating computational human models that simulate physiological responses under controlled loads (CRAN10).

2.1.2. Anthropomorphic Test Devices (ATDs): Utility and Limitations

Despite their widespread use in occupant safety testing, ATDs are limited in their ability to replicate the complex and age-dependent biomechanical behavior of real humans. These devices, typically based on the 50th percentile adult male, are constructed with rigid components and simplified anatomical structures that do not reflect the variability seen in human anatomy, particularly in elderly individuals. One of the main criticisms of current ATDs is their inability to simulate physiological changes such as thoracic stiffening, reduced spinal flexibility, and variations in soft tissue response that occur with aging (VIVE21).

Moreover, many biofidelity standards used in ATD development originate from a narrow set of PMHS experiments. For instance, the force–displacement corridors that shaped the thoracic behavior of the adult male ATD were derived from a limited group of older PMHS subjects, with insufficient representation of younger adults (BIOM24). As highlighted in recent literature, a single ATD cannot realistically replicate both young and elderly adults due to fundamental differences in their biomechanical properties. This gap has prompted calls for the development of new surrogates, such as multiple ATDs for different age groups or designs with adjustable thoracic stiffness to accommodate age-related variability (SUN16).

2.1.3. Comparison with Other Test Subjects (PMHS, children, etc.)

While ATDs remain the dominant tools for repeatable crash testing, and PMHS data serve as a basis for injury threshold development, neither approach alone is sufficient for understanding human kinematics during low-severity impacts. Ethical and legal constraints prevent the use of living subjects in high-speed scenarios, and PMHS studies cannot capture real-time human behavior such as muscle bracing or postural variability (SUN16). These limitations are particularly problematic when studying underrepresented populations such as children or elderly adults, where anatomical and physiological variability plays a central role in impact response. Pediatric ATDs, for example, are often based on geometrically scaled-down adult models rather than child-specific biomechanics, leading to questioned biofidelity (BIOM24). Similarly, adult ATDs do not adequately reflect the age-related changes in spine mobility and thoracic compliance seen in elderly individuals. Volunteer testing, especially at low speeds, has therefore emerged as the most viable method for ethically capturing real-world kinematics in these vulnerable populations (ARBO09).

The real-world applicability of volunteer studies has also been supported by recent evidence comparing volunteer-based rear-impact data with national crash databases. In 2018, over 1,200 volunteer exposures from 51 studies were compiled. They found symptom reporting rates, including neck pain, nearly identical to those observed in real-world rear-end collisions of comparable severity. Statistical analysis using operating characteristic curves demonstrated that the likelihood of injury beyond transient symptoms is extremely low in controlled volunteer exposures, even under impact severities similar to those encountered in everyday crashes (CORM18). This alignment between laboratory and real-world outcomes reinforces the external validity of volunteer testing as a method for studying occupant kinematics and low-level trauma without inducing harm.

2.1.4. Ethical Framework

The use of human volunteers in biomechanical testing introduces specific ethical requirements that go beyond those encountered in most clinical research. Because the participants are typically healthy individuals with no direct medical benefit from the study, risk mitigation, informed consent, and regulatory compliance become central to the study design. To ensure participant safety, low-speed deceleration pulses are selected based on prior studies known to be safe and non-injurious. For example, test pulses are

often derived from scaled-down versions of amusement park ride dynamics (ARBO09), with added precautions such as acoustic warnings and post-test monitoring. Additionally, compensation for participation is carefully calibrated to avoid coercion, typically covering time and discomfort but not offering undue inducement. These measures reflect current standards for ethical volunteer testing and are essential for maintaining public trust and participant autonomy in biomechanical research.

2.1.5. Evolution of Volunteer Test Platforms

Volunteer testing platforms have also evolved significantly to balance scientific rigor with participant safety and comfort. Early systems often used rigid sled setups in laboratory environments with limited adjustment to individual anthropometry. In contrast, more recent platforms incorporate adaptable seat configurations, precise restraint positioning, and adjustable pulse characteristics to replicate real-world vehicle postures more accurately (VIVE21). These improvements have enabled a more reliable and reproducible capture of kinematic data across different subject groups.

These advancements have been paralleled by creative approaches to platform design in both pediatric and adult testing contexts. Notably, the Children's Hospital of Philadelphia (CHOP) developed a pioneering low-speed sled system inspired by amusement park bumper cars. This setup enabled testing of pediatric volunteers in frontal, near-side, and rear impact conditions while maintaining accelerations within safe tolerances. The system has since informed adult volunteer studies as well, demonstrating the value of cross-population testing frameworks (BIOM24). More recently, researchers have begun to explore real-world platforms, such as autonomous public transport vehicles, to assess occupant kinematics during unexpected braking scenarios under naturalistic conditions. These studies, while still limited in number, illustrate a growing trend toward more realistic testing environments in biomechanics, aiming to complement laboratory-based methods without compromising safety or ethical standards.

2.2. Volunteer Testing Protocol

The experimental data used in this study were obtained from a series of low-speed frontal deceleration tests performed on human volunteers as part of the SENIORS project, funded by the European Commission. The tests were conducted at the Impact Laboratory (I3A) of the University of Zaragoza and adhered to strict ethical standards approved by the Clinical Research Ethics Committee of Aragón (CEICA). All participants provided written informed consent prior to testing. The following information regarding the data obtention method was retrieved from Lopez-Valdes et al., 2017 (LOPE17) and Vives-Torres et al., 2021 (VIVE21).

2.2.1. Test Setup

The test configuration consisted of a rigid seat, a rigid footrest, and a backrest constructed using flexible metallic wire segments. This assembly was designed to replicate the seating posture and pelvic displacement typically experienced in a passenger vehicle during a frontal crash. Both the footrest and seatbelt anchorage points were adjustable to match the anthropometry of each volunteer, ensuring realistic and repeatable posture.

The restraint system employed a non-retractor, non-force-limited three-point seatbelt. Initial pretensioning was manually applied using a spring scale, targeting approximately 50 N of force. This value was reduced if the volunteer reported discomfort during preparation.

The sled was programmed to deliver a triangular deceleration pulse with a peak of approximately 3.5 g and a duration of around 100 milliseconds. The target velocity at impact was 9 km/h. These parameters were chosen to ensure a safe and non-injurious response, in line with previous studies using similar pulse characteristics derived from scaled-down amusement park rides (ARBO09). The sled configuration, including the seating and restraint system, is shown in Figure 2.2-1 (LOPE17), and the deceleration pulse shape recorded during testing is displayed in Figure 2.2-2 (VIVE21).



Figure 2.2-1, Sled Test Configuration, showing the rigid seat, flexible backrest, footrest, and three-point seatbelt system. (LOPE17)

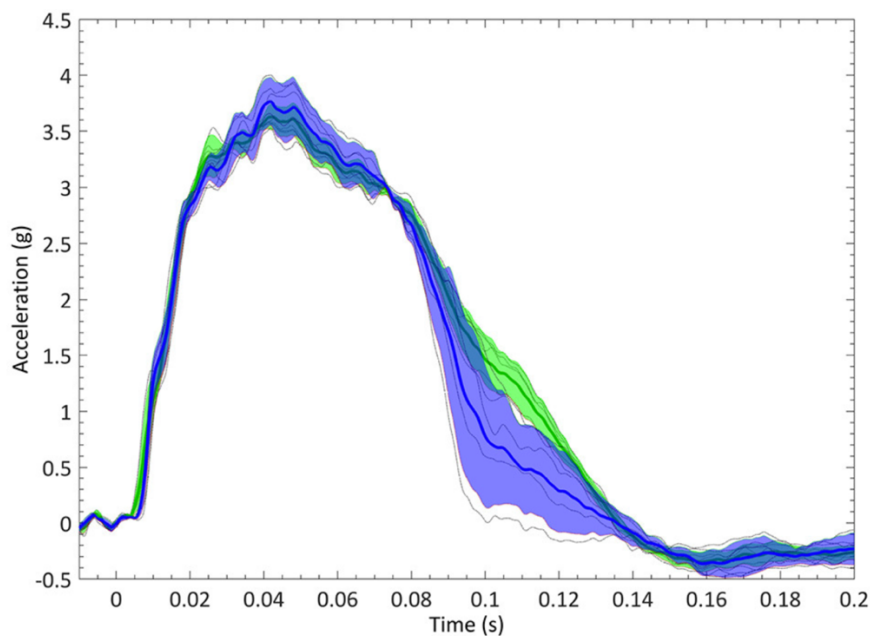


Figure 2.2-2, Sled Deceleration Pulse – Sled deceleration pulse corridors in the young (blue) and elderly (green) groups. Solid lines are the average deceleration within the group. Shaded area corresponds to the one standard deviation corridor. (VIVE21)

2.2.2. Safety and Ethics

All procedures were designed to prioritize volunteer safety. Participants were prescreened to confirm the absence of any medical conditions that could increase their risk of injury. Prior to testing, volunteers were briefed on the experimental procedure, instructed to remain relaxed throughout the event, and alerted with an acoustic signal shortly before the onset of deceleration. Each individual underwent multiple test trials, but only one (typically the third) was selected for analysis, unless sensor or marker issues necessitated using another.

Post-test monitoring ensured that no injuries or discomfort occurred, and no adverse events were reported. The study complied with applicable ethical guidelines, and data collection was structured to ensure anonymization and secure handling.

2.2.3. Instrumentation and Data Acquisition

A comprehensive instrumentation system was implemented to capture both kinematic and kinetic responses. The sled platform included:

- Two accelerometers for recording sled deceleration
- A six-degree-of-freedom (6-DOF) load cell under the seat
- Two 6-DOF load cells under the footrest
- Four load cells integrated into the shoulder and lap portions of the seatbelt

The configuration of the load cells can be seen in Figure 2.2-4.

To capture head kinematics, a triaxial accelerometer and angular rate sensor were mounted on a lightweight elastic headband worn by the volunteer, as can be seen in Figure 2.2-3. All sensor signals were acquired at a sampling frequency of 10,000 Hz using an external data acquisition system (PCI-6254, National Instruments, Austin, TX). Filtering was performed using low-pass filters, with cutoff frequencies individually selected according to the characteristics of each signal, ensuring that relevant signal content was preserved.

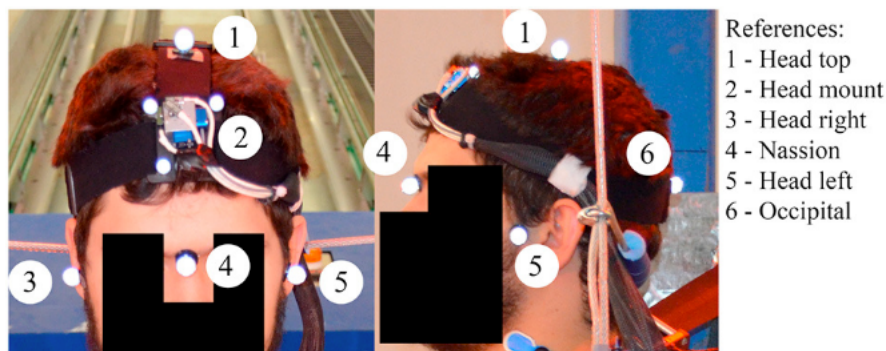


Figure 2.2-3. Head Mount - Detail of the 6 degree-of-freedom head cube and of the position of the sensors on the head of one of the volunteers. (VIVE21)

An overview of all sensor types, locations, and measurements is provided in

Table 2.2-1. Some sensor malfunctions occurred during select trials (e.g., tests 1680, 1691, and 1780), which were excluded from the analysis.

<i>Sensor Type</i>	<i>Location</i>	<i>Measurement</i>
Accelerometers (2 units)	Base of sled platform	Used to record sled deceleration
Load cell (6-DOF)	Under seat base	Measures vertical and shear forces at seat base
Load cells (6-DOF, 2 units)	Under footrest (left and right)	Measures foot–rest interaction forces
Seatbelt load cell (shoulder)	Between shoulder and D-ring	Measures belt tension at upper torso
Seatbelt load cell (lower shoulder)	Arbitrary position on shoulder belt	Complementary shoulder load measurement
Seatbelt load cells (lap, 2 units)	Left and right lap belt segments	Measures lap belt forces bilaterally
Triaxial accelerometer	Headband mount near CG (Endevco 7264C)	Measures linear head acceleration
Angular rate sensor	Same headband mount (DTS ARS-PRO 18K)	Measures rotational head motion

Table 2.2-1. Instrumentation Summary - Overview of the sensors used in the study, including their type, mounting location, and measurement recorded.

Kinematic trajectories were obtained using a 10-camera optoelectronic motion capture system (Vicon TS series) operating at 1,000 Hz. Reflective markers were placed on specific anatomical landmarks of each volunteer to enable accurate tracking of body motion. These landmarks included the external auditory meatus (bilateral), nasion, opisthocranium, cervical spine (C4), thoracic spine (T1, T4, T8), acromion (bilateral), greater trochanter, and anterior superior iliac spine (ASIS). Additionally, a triad of markers was mounted on the head sensor plate to allow transformation of the sensor data from its physical location on the headband to the estimated head center of gravity, defined as the midpoint between the bilateral external auditory meatus markers.

Based on these markers, a local head coordinate system was defined using the Frankfurt anatomical plane, with its origin at the estimated head center of gravity. All kinematic data were then transformed into the global laboratory coordinate system, in which the x -

axis pointed forward and the z-axis upward. For compatibility with force measurements, head displacements and trajectories were expressed relative to the occipital condyle joint.

The motion capture system operated by detecting retro-reflective spherical markers placed on the volunteers within a calibrated three-dimensional volume. The configuration of said markers is illustrated also in Figure 2.2-4. Prior to testing, a calibration procedure was performed to determine the optical characteristics, position, and orientation of each of the ten cameras relative to a fixed global coordinate system (GCS) anchored to the laboratory. In addition, a local coordinate system (LCS) was defined relative to the test buck, with its origin located at the front-right corner of the seat. The local x-axis pointed forward at a 30-degree angle clockwise from the subject's frontal anatomical axis, the z-axis pointed vertically upward, and the y-axis was defined to form a right-handed coordinate system. Unless otherwise noted, all displacement data are expressed relative to this LCS. Marker positions were reconstructed in three dimensions using a photogrammetric algorithm implemented in the Vicon Nexus software package (Nexus 1.8.5, Vicon, Oxford, UK), which triangulated each target's position at every time step based on synchronized images from the multiple camera views.

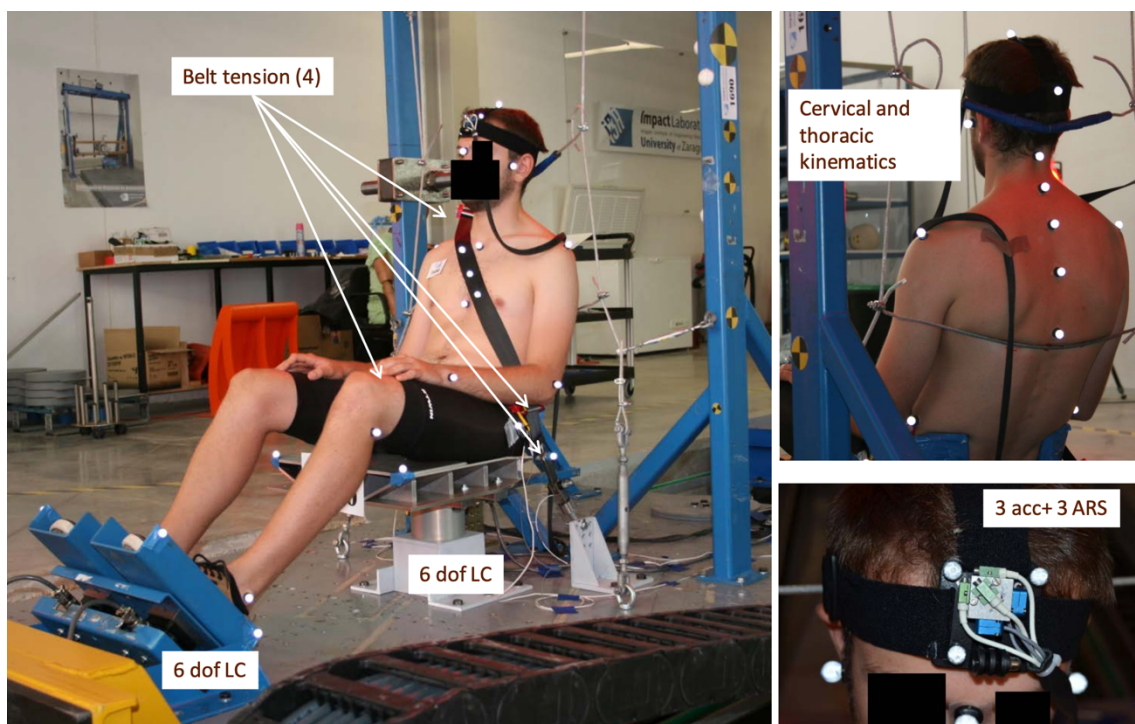


Figure 2.2-4. Instrumentation Layout - Example of the experimental setup showing the placement of load cells, accelerometers, and reflective markers on the sled platform and volunteer. (LOPE17)

2.2.4. Volunteer Sample and Preparation

A total of 13 male volunteers participated in the tests: 9 younger adults aged 18–30 and 4 elderly adults over 65. The cohort was selected to approximate the anthropometric dimensions of a 50th percentile male (target height: 175 cm, target mass: 78 kg). Anthropometric measurements such as height, weight, head girth, and neck circumference were recorded before testing. These measurements are summarized in Table 2.2-2.

Test ID	Group	Subject ID	Age (years)	Stature (cm)	Weight (kg)	Head girth (cm)	Neck girth (cm)	Seated height (cm)
1679	Young	Vol 01	18	171.0	75.5	59.5	38.0	62.4
1684	Young	Vol 02	18	176.5	77.7	57.0	36.5	66.4
1689	Young	Vol 03	21	179.5	73.0	59.0	37.0	64.1
1694	Young	Vol 04	21	179.0	79.4	58.0	37.0	65.5
1767	Young	Vol 05	22	167.0	75.3	55.0	38.5	63.3
1858	Young	Vol 06	28	172.0	68.4	56.0	37.5	57.1
1865	Young	Vol 07	36	174.0	73.0	59.5	38.0	62.5
1869	Young	Vol 08	26	174.0	64.6	57.0	37.0	60.7
1873	Young	Vol 09	18	173.0	86.7	61.0	43.0	63.1
1771	Elderly	Vol 10	71	164.0	81.0	59.0	46.5	60.6
1774	Elderly	Vol 11	71	176.5	99.1	60.0	46.0	64.6
1778	Elderly	Vol 12	85	165.3	78.2	57.0	41.5	63.2
1780	Elderly	Vol 13	67	169.0	88.2	59.5	44.5	65.7
1862	Elderly	Vol 14	66	172.5	89.6	58.5	41.0	62.3

Table 2.2-2. Volunteer Anthropometric Data - Summary of the anthropometric characteristics of all participating volunteers.

Volunteers were seated with their pelvis aligned to the seat centerline, and adjustments were made to achieve a torso inclination of approximately 60° , a thigh angle near 12° , and a tibial inclination of about 45° . The seatbelt's D-ring was positioned in vertical alignment with the external auditory meatus and laterally offset by roughly 100 mm from the acromion. This setup was repeated across all trials to maintain consistency in initial posture and restraint configuration.

Figure 2.2-5 presents representative frames from lateral video footage, illustrating the motion comparison between a young and an elderly volunteer at matched time points during deceleration (LOPE17).

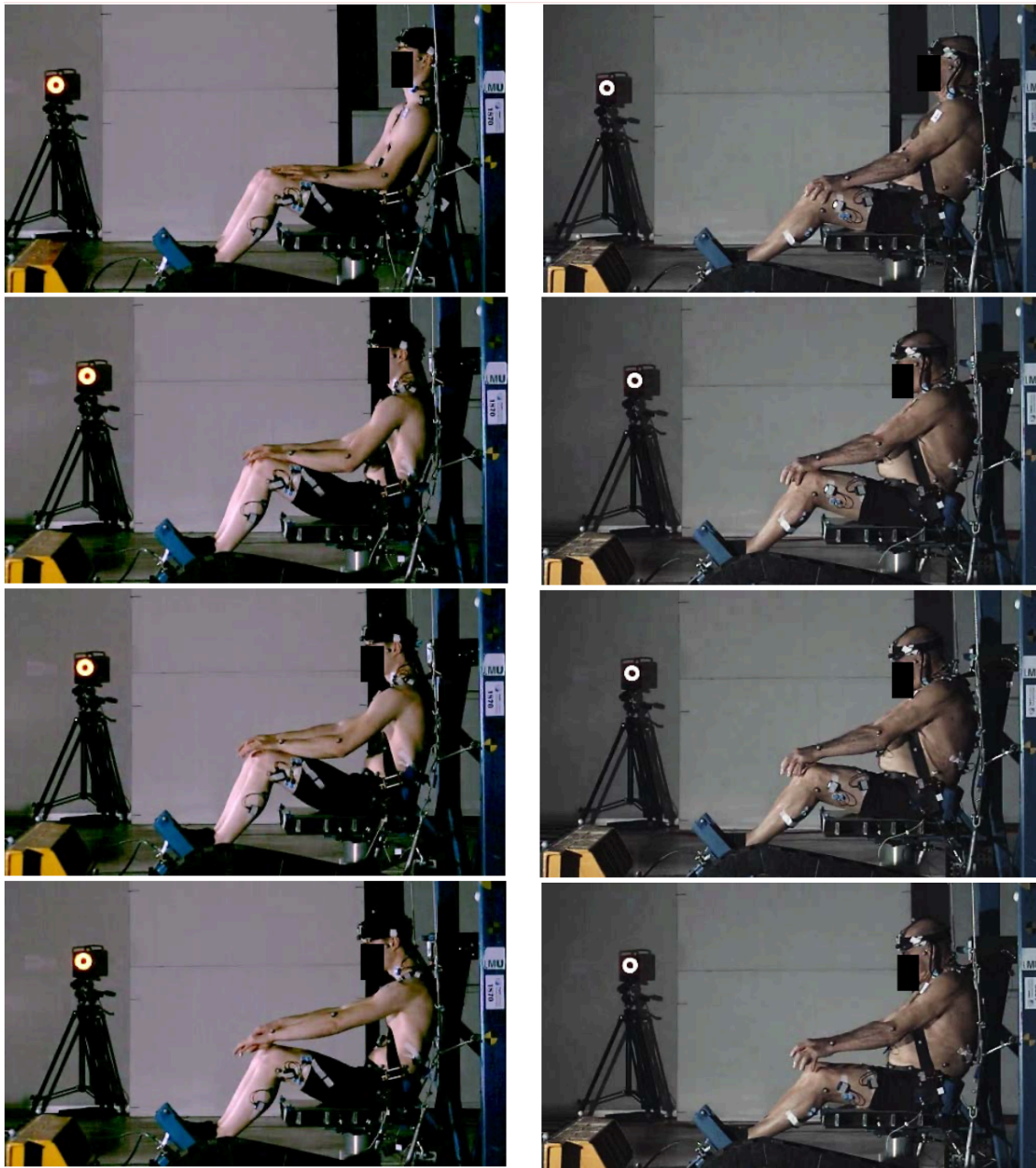


Figure 2.2-5. Lateral Video Comparison - Representative still frames comparing the motion of a young and an elderly volunteer during frontal deceleration. (LOPE17)

3. METHODOLOGY

3.1. Software Environment

All analyses and data processing were performed in MATLAB R2023a, which provided the main environment for numerical computation and visualization. Raw motion capture data were stored in Microsoft Excel files, each containing the full set of landmark trajectories per test trial. MATLAB was used to import, structure, and process these datasets across all methods. Graphs and other figures were also generated in MATLAB. Statistical modeling of principal component scores against anthropometric variables was conducted in GRETL.

Methods 1 and 2 (B-spline and PCA of trajectories, and the analysis of posture at maximum head displacement) leveraged MATLAB's built-in functions for curve approximation of landmark motion, construction of fixed-length feature vectors, computation of principal components, re-referring of landmarks and any other needed operations. All regression models were subsequently estimated in GRETL.

Method 3 (Arc-length re-parameterization, signal registration, and statistical corridor generation) was implemented using the open-source ARCGen Toolbox developed by Hartlen and Cronin (HART22), which can be found in the Annex section, ARCGen. ARCGen was integrated into MATLAB and applied to perform arc-length normalization, signal registration, and computation of characteristic average trajectories and variability envelopes.

This integrated software environment allowed consistent application of all analytical steps, while accommodating the methodological differences between approaches.

3.2. Method 1: B-Spline approximation and Principal Component Analysis of individual landmark trajectories

3.2.1. Method Overview

The analysis presented in this section is designed to analyze differences in upper-body motion between test subjects during low-speed frontal decelerations. It follows a structured pipeline that transforms raw positional data into a form suitable for statistical regression. The process integrates three main stages: curve fitting using B-splines, dimensionality reduction via Principal Component Analysis, and regression modeling to explore the relationship between motion characteristics and subject-specific anthropometric features.

The starting point of the analysis consists of high-frequency optoelectronic motion capture data recorded for 6 different anatomical landmarks (head, C4, T1, T4, T8, and H-point) per each of the 14 volunteer in the study. Each of these trajectories consists of hundreds of time-sampled (x, z) coordinates. While this level of temporal resolution is valuable for capturing dynamic motion, it also presents a challenge: it is high-dimensional, noisy, and contains more information than is meaningful to analyze directly for statistical comparisons between subjects. The high number of time points per trial is also impractical to incorporate into a regression model due to overfitting and multicollinearity.

To address this and reduce the dimensionality while preserving the essential structure of the trajectories, the first step is to fit each landmark's motion with a B-spline curve. A B-spline is a parametric curve defined by a small number of control points, which approximate the shape of the original signal without forcing the curve to interpolate each individual time point. In this thesis, spline order and point spacing are selected such that the fitted curve retains key movement features while discarding high-frequency noise and redundant data.

After the B-spline fitting process, each trajectory has been converted into a fixed number of control points. These control points are then arranged into a feature vector for each test, capturing the position of each landmark along its path in a compressed and standardized form. Since the trajectories are defined in 2D space, each control point contributes two values (x and z), and the final vector contains $2n$ values per trajectory, where n is the number of control points used. However, even after this compression, the number of variables remains large (16 values per landmark), and these coordinates are often highly correlated across subjects.

To further reduce complexity and facilitate statistical modeling, the dataset of control point vectors is processed with Principal Component Analysis (PCA). PCA is a linear dimensionality reduction technique that identifies the directions in which the data varies most. It transforms the original coordinates into a new set of orthogonal variables, known as principal components (PCs), ranked by the amount of variance they explain. In this context, each PC represents a characteristic mode of variation in the trajectory shape (e.g., forward translation, vertical deflection, curvature) and each subject receives a score for each component, indicating how their motion differs along that mode from the average.

Importantly, the components are uncorrelated by construction, which is crucial for the next step.

The final stage is regression modeling. The principal component scores obtained for each trajectory are used as dependent variables in linear regression models. The predictors are the anthropometric and demographic characteristics of each subject (e.g., seated height, body mass, age group). The purpose of these models is to identify statistically significant relationships between the shape of a subject's trajectory and their physical attributes.

In summary, the method is designed to transform complex trajectory data into a standardized, low-dimensional form suitable for regression analysis. The ultimate goal is to quantify how much of the variation in landmark motion can be explained by subject characteristics, enabling biomechanical interpretation of the observed differences between young and elderly volunteers.

3.2.2. Literature background

The methodological framework combining B-spline fitting, PCA, and regression modeling draws on a well-established foundation in the biomechanics literature. Each component of this pipeline has been applied in related contexts to reduce data dimensionality, extract interpretable features from motion trajectories, and investigate relationships between movement patterns and subject-specific parameters. This section reviews key sources that support the use of each step, with a focus on studies involving human kinematic analysis in crash testing, gait, and injury prediction contexts.

B-spline modeling of human motion

As mentioned, B-spline curves are a type of polynomial functions that offer local control, smoothness, and computational efficiency, making them ideal for approximating complex and noisy biomechanical trajectories. Their flexibility in shaping curves without overfitting to individual time points is particularly valuable in crash testing contexts, where raw motion capture data are often dense and noisy.

A particularly relevant reference for the present methodology is the study by Samuels et al. (2015), which closely mirrors the structure and intent of the approach used here. In their work, the authors modeled the sagittal-plane trajectories of multiple anatomical landmarks (head top, nasion, EAM, C4, T1, and pelvis) during low-speed frontal impact tests involving pediatric and adult volunteers. To compress and regularize the data, they fitted cubic B-splines to the recorded landmark trajectories, using eight control points per curve. This configuration was chosen to ensure a root mean square error (RMSE) below 1% of the mean head excursion, balancing fidelity and dimensionality (SAMU15).

Similar uses of B-spline-based trajectory compression have been reported in rehabilitation studies using inertial sensors (e.g., stroke recovery monitoring), and in robotic control systems for trajectory planning. In these applications, B-splines are valued for their ability to efficiently represent smooth motions with relatively few parameters while respecting biomechanical constraints (HWAN23; WANG88).

Principal Component analysis in biomechanics

Principal Component Analysis (PCA) is a common tool in biomechanics for reducing the dimensionality of multivariate datasets while retaining dominant patterns of variation.

In the previously mentioned study by Samuels et al. (2015), after B-spline fitting, they constructed a matrix of control point coordinates and performed PCA to identify dominant modes of variation. The first two principal components captured the majority of variability across subjects and were found to correlate strongly with erect seated height, especially in the head and spinal landmarks (SAMU15).

PCA has also been used to decompose full-body motion data in gait and posture studies. For instance, Federolf et al. (2013) applied PCA to 117-dimensional marker trajectories and interpreted the resulting components as "principal movements," such as forward sway or arm-leg coordination (FEDE13). These studies illustrate PCA's strength as a non-reductionist method: it does not presuppose which features are important but instead allows dominant biomechanical trends to emerge from the data.

In crash biomechanics, PCA has also been applied in injury risk modeling. Brumbelow, M.L. (2023), for example, used PCA to reduce multiple lower-extremity injury metrics into orthogonal components that could then be analyzed using logistic regression to identify key predictors of injury (BRUM23).

Regression of PCA-derived features to anthropometric variables

The final step in the pipeline involves regressing PCA-derived features (i.e., scores) against anthropometric variables to quantify how motion patterns vary across the population. This type of analysis has proven effective in identifying biomechanically meaningful relationships, such as the influence of age or seated height on specific principal movement components.

Continuing with the study by Samuels et al. (2015), the obtained principal component scores were regressed against anthropometric variables to create a statistical model capable of predicting trajectory shapes for anthropomorphic test devices (ATDs) of varying size. The model was validated by reconstructing trajectories for known ATD dimensions (e.g., 6-year-old and 10-year-old dummies), showing that body size influences not just the extent but also the shape and curvature of movement paths (SAMU15).

In addition, Sun et al. (2016) developed a regression-based method to generate biomechanical response corridors from pelvis impact data. They applied PCA to aligned force signals, regressed the resulting PC scores to anthropometric variables such as waist breadth and vertex-to-symphysis distance, and used Monte Carlo simulation to generate individualized corridor bounds. Their method outperformed traditional normalization approaches by preserving anthropometric influences rather than scaling them out (SUN16).

Similarly, Donnelly and Moorhouse (2012) advocated for regression modeling of aligned response curves to preserve timing and magnitude features specific to individual subjects, rather than relying on averaged or normalized signals (DONN12). Their later work on

deformation energy approaches (DONN14) further reinforced the need to model inter-subject variability directly instead of treating it as noise.

3.2.3. Purpose and justification

This method was selected to address a specific analytical challenge: how to extract and compare the underlying structure of individual motion trajectories without oversimplifying them. The goal is not to summarize raw movement but to isolate which aspects of trajectory shape vary meaningfully across subjects and whether those variations relate to physical attributes.

B-splines were chosen over direct time-series analysis because they offer a compact yet geometrically meaningful representation. Unlike simple downsampling or filtering, B-splines retain localized curvature and motion tendencies while reducing noise and dimensionality. This is particularly useful when comparing subjects with different morphologies, where minor fluctuations can obscure broader motion patterns.

PCA was then applied to the B-spline control points not only for compression, but to expose dominant modes of variation that are uncorrelated and interpretable. These components allow comparisons across individuals in terms of specific, decoupled movement patterns, something that would be difficult to achieve using raw coordinates.

The final regression step is justified by the need to move from description to explanation. Rather than simply observing differences in motion, the method aims to quantify how much of those differences are associated with subject-level variables. This enables statistically grounded inferences about how body size or age affect trajectory shape, and offers a framework for predicting motion trends beyond the observed sample.

Each element in this sequence was selected to maximize interpretability and comparability, while keeping the analysis as robust as possible against small sample size and high-dimensional input. The full process, spline fitting, PCA, and regression, aims to preserve key structure, filter irrelevant noise, and isolate variation that may be biomechanically meaningful.

3.2.4. Mathematical background

B-Spline curve representation

A B-spline curve is a piecewise-defined parametric function commonly used to approximate smooth trajectories with compact representation and local control. In this work, B-splines are used to model the two-dimensional motion trajectories of anatomical landmarks recorded over time during frontal deceleration tests. The goal is to replace high-dimensional raw time-series data with a fixed-length, smooth representation that preserves biomechanically relevant features and enables further statistical analysis.

General formulation

Let $d \in \mathbb{N}$ be the degree of the spline (i.e., the polynomial degree of each segment), and let $k = d + 1$ be the corresponding spline order. A B-spline curve of order k defined over a domain $t \in [a, b] \subset \mathbb{R}$ is expressed as:

$$S(t) = \sum_{i=0}^n P_i N_{i,k}(t)$$

where:

- $S(t) \in \mathbb{R}^2$ is the point on the curve at parameter t ,
- $P_i \in \mathbb{R}^2$ is the i -th control point (to be determined),
- $N_{i,k}(t)$ is the i -th B-spline basis function of order k ,
- $n + 1$ is the number of control points.

The basis functions $N_{i,k}(t)$ are piecewise polynomials constructed recursively and determine the weight of each control point at a given parameter value t . Their construction depends on a knot vector $T = \{t^0, t^1, \dots, t_m\}$, which partitions the domain and defines where each basis function becomes active.

Recursive definition of Basis functions

The B-spline basis functions are defined using the Cox–de Boor recursion:

For $k = 1$:

$$N_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

For $k > 1$:

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

Any term with a denominator equal to zero is conventionally taken as zero. The result is a set of $m - k$ overlapping functions, each basis function $N_{i,k}(t)$ being nonzero only over a sequence of at most k consecutive knot spans. This local support is what grants B-splines their flexibility and computational efficiency.

Knot Vector and its role

The knot vector $T = \{t_0, t_1, \dots, t_m\}$ is a non-decreasing sequence of real values that determines how the domain is partitioned into segments, and how the basis functions blend across these segments.

Given:

- Spline degree d ,
- Number of control points $n + 1$,
- Order $k = d + 1$,

the knot vector must have $m + 1 = n + k + 1$ entries.

In this work, a uniform clamped knot vector is used. This means:

- The internal knots are uniformly spaced over $[0,1]$,
- The first and last knots are each repeated k times (i.e., $t_0 = t_1 = \dots = t_{k-1} = 0$, and $t_{m-k+1} = \dots = t_m = 1$),

which ensures that the curve interpolates the first and last control points:

$$S(0) = P_0, \quad S(1) = P_n$$

This is desirable when the start and end of the trajectory correspond to physical time endpoints that must be preserved (e.g., onset and end of deceleration).

Curve Evaluation and Dimensionality

To evaluate a B-spline curve numerically, the basis functions $N_{i,k}(t)$ are computed at a discrete set of parameter values $t \in [0,1]$, and the curve values are obtained as:

$$S(t_j) = \sum_{i=0}^n P_i N_{i,k}(t), \quad \text{for } j = 1, \dots, T$$

This can be written in matrix form as:

$$\mathbf{S} = \mathbf{B} \cdot \mathbf{P}$$

where:

- $\mathbf{S} \in \mathbb{R}^{T \times 2}$ contains the curve points at T parameter values,
- $\mathbf{B} \in \mathbb{R}^{T \times (n+1)}$ is the basis matrix with entries $B_{ji} = N_{i,k}(t_j)$,
- $\mathbf{P} \in \mathbb{R}^{(n+1) \times 2}$ is the matrix of control point coordinates.

In other words, for each evaluation point t_j the vector S_j is computed, such that each row of matrix $\mathbf{B}$ is the set of basis function values at t_j and multiplying yields all spline points.

This way, the control points P_i become the parameters to estimate in the curve fitting process.

Use in trajectory compression

Each motion trajectory is defined by a sequence of measured positions $\{s_1, \dots, s_T\} \subset \mathbb{R}^2$. Instead of storing all T points (typically on the order of hundreds), we represent the trajectory using a small, fixed number of control points P_i . These points fully define the fitted curve and are sufficient to reconstruct an approximation of the original trajectory.

For use in later stages of analysis (Principal Component Analysis), the control point coordinates are flattened into a row vector of length $2(n+1)$. This fixed-length vector serves as a compressed, standardized representation of the trajectory.

So, if each control point P_i has two coordinates (x_i, z_i) , then for each trajectory, all control points are collected into a 1D vector such as

$$[x_0, z_0, x_1, z_1, \dots, x_n, z_n]$$

with $2(n+1)$ features.

Principal Component Analysis

PCA is a linear transformation technique used to extract the dominant modes of variation from multivariate data. In the present study, PCA is applied to the control point vectors

obtained from B-spline trajectory approximations. The objective is to represent each trajectory using a set of uncorrelated variables (principal components) that summarize the major patterns of shape variability across subjects.

Data representation and centering

Let $\mathbf{X} \in \mathbb{R}^{N \times D}$ denote the data matrix, where:

- N is the number of trajectories (i.e., trials/subjects for a given landmark),
- $D = 2(n + 1)$ is the number of features per trajectory, corresponding to the flattened x and z coordinates of $n + 1$ B-spline control points.

Each row $\mathbf{x}_i \in \mathbb{R}^D$ of $\mathbf{X}$ represents one trajectory in feature space.

The first step in PCA is to center the data by subtracting the empirical mean vector $\bar{\mathbf{x}} \in \mathbb{R}^D$:

$$\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{1} \bar{\mathbf{x}}^T$$

where $\mathbf{1} \in \mathbb{R}^{N \times 1}$ is a column vector of ones, and $\tilde{\mathbf{X}}$ is the mean-centered data matrix.

Covariance matrix and eigenvalue decomposition

The next step is to compute the sample covariance matrix:

$$\mathbf{C} = \frac{1}{N-1} \tilde{\mathbf{X}}^T \tilde{\mathbf{X}} \in \mathbb{R}^{D \times D}$$

The principal components are obtained as the eigenvectors of $\mathbf{C}$. That is, we solve:

$$\mathbf{C}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}$$

where:

- $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_D] \in \mathbb{R}^{D \times D}$ contains the eigenvectors (principal axes),
- $\mathbf{\Lambda} = \text{diag}(\lambda^1, \dots, \lambda_D)$ is a diagonal matrix of eigenvalues ordered such that $\lambda^1 \geq \lambda^2 \geq \dots \geq \lambda_D \geq 0$.

Each eigenvalue λ_j quantifies the variance explained by the corresponding principal component $\mathbf{v}_j$.

Alternatively, PCA can be computed via Singular Value Decomposition (SVD) of $\tilde{\mathbf{X}}$:

$$\tilde{\mathbf{X}} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$$

where:

- $\mathbf{U} \in \mathbb{R}^{N \times D}$ contains the left singular vectors (subject-wise scores),
- $\mathbf{V} \in \mathbb{R}^{D \times D}$ contains the principal directions,
- $\mathbf{\Sigma} \in \mathbb{R}^{D \times D}$ is diagonal with singular values $\sigma_j = \sqrt{\lambda_j (N-1)}$.

Principal Component scores and projection

The principal component scores are obtained by projecting the centered data onto the eigenvectors:

$$\mathbf{Z} = \tilde{\mathbf{X}}\mathbf{V}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times D}$ contains the scores for each subject along each principal component.

In practice, only the first $K \ll D$ components are retained. These scores z_{ij} then represent each trajectory in a low-dimensional, orthogonal space:

$$z_{ij} = \tilde{\mathbf{x}}_i^T \mathbf{v}_j, \quad \text{for } j = 1, \dots, K$$

Reconstruction and interpretation

Each input trajectory $\mathbf{x}_i$ can be approximately reconstructed using the mean vector and the first K principal components:

$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}} + \sum_{j=1}^K z_{ij} \mathbf{v}_j$$

Each component $\mathbf{v}_j$ represents a mode of trajectory variation, and the score z_{ij} measures how strongly that mode is expressed in subject i . Because the components are orthogonal, they can be used independently in downstream statistical modeling.

Linear regression of Principal Component scores

Following the application of PCA, each trajectory is represented as a low-dimensional vector of principal component scores. The goal of this final analytical step is to determine whether these scores, capturing shape variability in landmark trajectories, can be explained by subject-specific parameters. This is accomplished using linear regression, a classical statistical method for modeling relationships between a scalar response and multiple explanatory variables.

Model structure

Let $z_i \in \mathbb{R}$ denote the score of the i -th subject along a particular principal component (e.g., PC1 or PC2). Let $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T \in \mathbb{R}^p$ be the vector of predictor variables (e.g., age, seated height, mass) for that subject.

The regression model is defined as:

$$z_i = \beta_0 + \sum_{j=1}^p \beta_j x_{ij} + \varepsilon_i$$

or in matrix-vector notation for all N observations:

$$\mathbf{z} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where:

- $\mathbf{z} \in \mathbb{R}^N$ is the response vector of PC scores,
- $\mathbf{X} \in \mathbb{R}^{N \times (p+1)}$ is the design matrix, with a column of ones for the intercept and remaining columns for the predictors,
- $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$ is the vector of regression coefficients,
- $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ is the residual vector, assumed to follow $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, independent and identically distributed.

Each regression is performed separately for each retained principal component. This is valid due to the orthogonality of the components; the scores across different PCs are uncorrelated by construction, allowing independent modeling.

Parameter estimation

The vector of coefficients β is estimated using ordinary least squares (OLS), which minimizes the residual sum of squares:

$$\hat{\beta} = \arg \min_{\beta} ||\mathbf{z} - \mathbf{X}\beta||_2^2$$

The closed-form solution is:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{z}$$

provided that $\mathbf{X}^T \mathbf{X}$ is invertible, which holds if the predictor variables are linearly independent.

Statistical interpretation

Each coefficient $\hat{\beta}_j$ represents the expected change in the PC score associated with a one-unit increase in predictor x_j , holding all other predictors constant. The statistical significance of each predictor can be evaluated using t -tests, the overall model fit via *adjusted* R^2 , and residual analysis to verify assumptions such as homoscedasticity and normality of errors.

Because the regressions are applied to orthogonal PCs, the influence of subject-specific variables can be analyzed component by component, each reflecting a different mode of shape variation in the trajectory.

In this study, the set of predictor variables used in each regression model was kept small and interpretable, consisting of anthropometric and demographic features such as age group, seated height, and body mass. As each regression was performed separately per principal component and per landmark, the approach ensured clarity in interpreting how specific physical characteristics relate to distinct modes of trajectory variation.

3.2.5. Implementation

Overview of workflow

The implementation of this method followed a structured, modular workflow executed independently for each of the six anatomical landmarks (head, C4, T1, T4, T8, and H-point). The pipeline comprised four main stages:

1. Approximation of each landmark trajectory using B-spline curves,
2. Vectorization of the resulting control points into standardized feature vectors,
3. Dimensionality reduction via Principal Component Analysis, and
4. Statistical modeling of selected principal component scores through linear regression.

At each stage, decisions regarding parameters, data formatting, and processing logic were made with a focus on reproducibility, consistency across landmarks, and biomechanical interpretability. No cross-linking between landmarks was introduced at any point; all data transformations and analyses were performed separately to preserve anatomical independence in the resulting components.

The methodology was implemented using custom MATLAB routines, structured around a sequence of well-defined data transformations and standardized inputs and outputs. The overall workflow ensured traceability from raw motion trajectories to regression-ready features, with intermediate results archived for each processing step.

B-Spline fitting

The initial stage of the processing pipeline consisted of fitting B-spline curves to raw motion trajectories recorded for each anatomical landmark. Before this approximation was applied, the raw trajectories exhibited high temporal resolution and substantial variability across subjects and repetitions. While this level of detail is essential for capturing motion precisely, it also results in high-dimensional signals that are difficult to compare directly and impractical to model statistically in their raw form.

An example of these raw trajectories are shown in Figure 3.2-1. It displays the x - z motion of a single anatomical landmark (T1) across all 14 volunteers. The variability and density of the curves underscore the necessity of introducing a compact, standardized representation capable of preserving biomechanical meaning while reducing dimensionality.

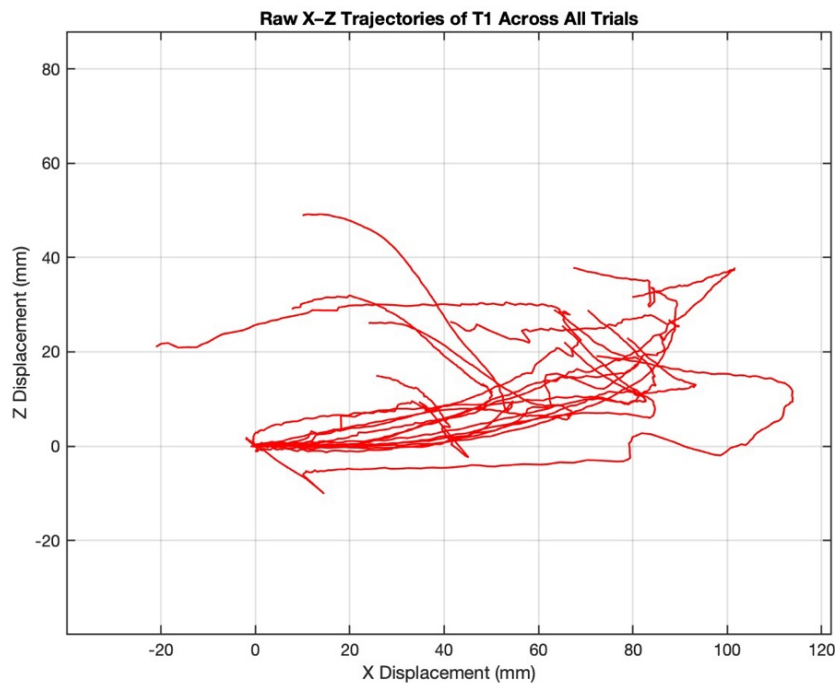


Figure 3.2-1. Raw x - z trajectories of a selected anatomical landmark (T1) across all test trials. The trajectories are high-resolution and exhibit significant inter-subject variability, motivating the need for a compact, standardized representation such as B-spline approximation.

The raw trajectory input data were stored in a collection of Excel (.xlsx) files, with one file per test trial and volunteer. Each file contained the full set of landmark trajectories recorded during that trial, structured as time-series of two-dimensional (x, z) coordinates sampled at uniform intervals. This structure allowed synchronized access to all landmarks for a given test but required that each landmark be processed individually across files.

The spline fitting process was implemented in MATLAB using the custom routine *Spline_analysis_all_tests_together.m* which can be found in section 3.1.1 of the Annex. Upon execution, the script iterated through the directory of Excel files, extracted the trajectory data for each landmark of interest, and normalized the time domain to $t \in [0,1]$. No interpolation or filtering was required at this stage, as all recordings were acquired at a consistent sampling frequency and time span.

Cubic B-splines (order $k = 4$, degree $d = 3$) were used for all fits, with a uniform clamped knot vector, ensuring that each spline curve interpolated the first and last trajectory points while maintaining C^2 continuity throughout. The control points $\{P_0, P_1, \dots, P_n\} \subset \mathbb{R}^2$ were estimated by solving a least-squares problem over the full sequence of recorded positions for each trajectory. Based on methodological precedent from Samuels et al. (2015) and to ensure consistency across trials, the number of control points was fixed to 8 for all landmarks (SAMU15).

To validate the adequacy of the spline approximation, the root mean square error (RMSE) was computed between the fitted curve and the original trajectory, using a uniform resampling of 10 points along the time domain. This allowed a standardized assessment of approximation quality across trials. The resulting RMSE values were used to confirm that the selected number of control points preserved key trajectory features while achieving substantial data compression.

An example of spline fitting performance is shown in Figure 3.2-2., where the fitted trajectory is overlaid with the original data for a representative trial and landmark T1. The fitted curve preserves the general shape and curvature of the motion while substantially reducing the dimensionality of the signal. The fitted trajectories for all landmarks can be found in the Spline fitting and dimensionality reduction Results section.

The final output of this step consisted of the estimated control point coordinates $\{P_0, \dots, P_n\}$, flattened into a $1 \times 2(n+1)$ row vector ordered as $(x_0, z_0, x_1, z_1, \dots, x_n, z_n)$. These vectors served as standardized inputs to the next stage of the pipeline.

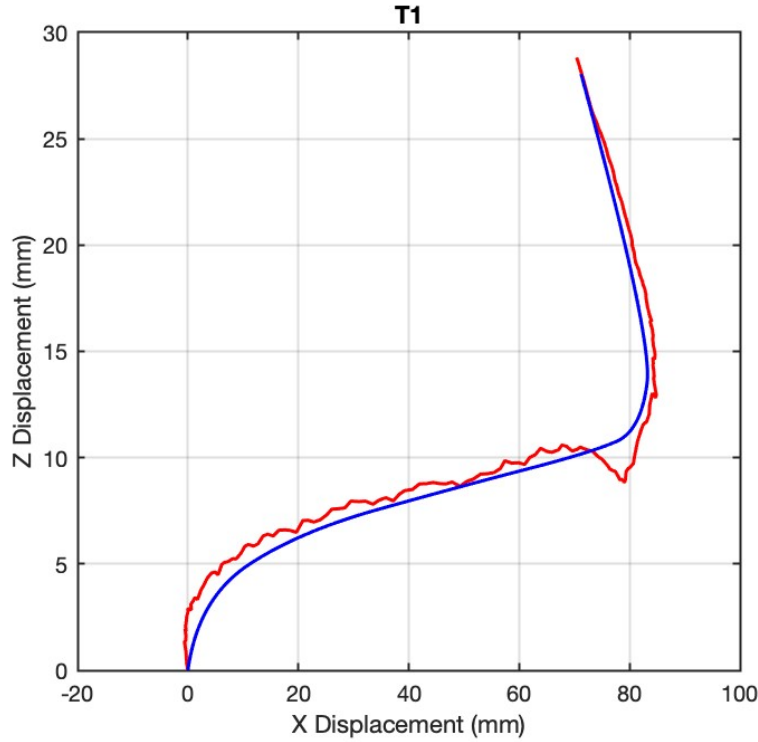


Figure 3.2-2. Example of original trajectory (red line) and fitted B-spline curve (blue line) for the T1 landmark during a representative trial. The fitted curve preserves the essential shape of the motion with reduced data complexity.

Construction of PCA input matrix

Once each trajectory had been converted into a fixed-length vector of B-spline control point coordinates, these vectors were assembled into input matrices for PCA. One matrix was created for each anatomical landmark to ensure that the modes of variation extracted in subsequent steps would reflect landmark-specific motion patterns without cross-contamination between body regions.

This step was executed using the MATLAB script *All_coordinates_any_n.m*, found in section 3.1.2. of the Annex. Upon execution, the user is prompted to specify the landmark of interest. The script then iterates through all processed test trials, retrieves the corresponding $1 \times 2(n + 1)$ control point vectors for that landmark, and assembles them into a matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, where:

- N is the number of trials for the selected landmark (1 per volunteer),
- $D = 2(n + 1)$ is the feature dimension, representing the x and z coordinates of each control point, interleaved in consistent order.

The control point vectors are stacked row-wise to form the full PCA input matrix. Because the number and ordering of control points were identical across all trajectories, the resulting matrix was structurally homogeneous and ready for direct analysis.

No additional transformations were applied at this stage; the values in $\mathbf{X}$ remained in physical units (millimeters). Mean-centering was performed internally during the PCA step. The completed input matrices were saved in MATLAB's native *.mat* format and passed to the next stage for dimensionality reduction.

Principal Component Analysis

PCA was applied separately to each landmark's input matrix to reduce dimensionality and isolate dominant patterns of trajectory variation. All PCA computations were carried out using the MATLAB script *PCA_Analysis* (found in section 3.1.3 of the Annex), which took as input the control point matrix produced in the previous stage.

For each landmark, the script returned a full set of principal component scores and the corresponding percentage of variance explained by each component. The output was analyzed to determine how many components should be retained for further modeling. Rather than using a fixed cutoff, this decision was guided by inspecting the distribution of variance across components and identifying the point at which adding further components yielded marginal returns.

Table 3.2-1. provides an example of this evaluation for the T1 landmark. In this case, the first three components together captured over 90% of the total variance, suggesting that a small number of components was sufficient to represent most of the variability in the trajectories. The results for all landmarks can be found in the Spline fitting and dimensionality reduction Results section.

PC Index	Variance explained (%)	Cumulative variance explained (%)
1	57.2	57.2
2	28.9	86.1
3	6.1	92.2
4	2.5	94.7
5	1.6	96.3

Table 3.2-1. Example of individual and cumulative explained variance for the T1 landmark. The first two or three components account for the majority of the total variance, making the rest largely redundant for modeling purposes.

To qualitatively examine the structure of the reduced space, the script also generates visual projections of the data in the PCA coordinate system. Figure 3.2-3. shows an example scatter plot of results for Landmark T1 projected onto the first two principal components. While some group-wise trends (e.g., young vs. elderly) begin to emerge, the

overlap between classes is still substantial. Similarly, Figure 3.2-4. illustrates the first three components in a 3D projection, offering a fuller view of variation but still lacking clear separation. The 2D graphs for all landmarks can be found in the Principal Component Analysis outcomes section. The analysis of 3D graphs was discontinued due to the complex analysis required and the lack of potential for valuable insights compared to the 2D graphs.

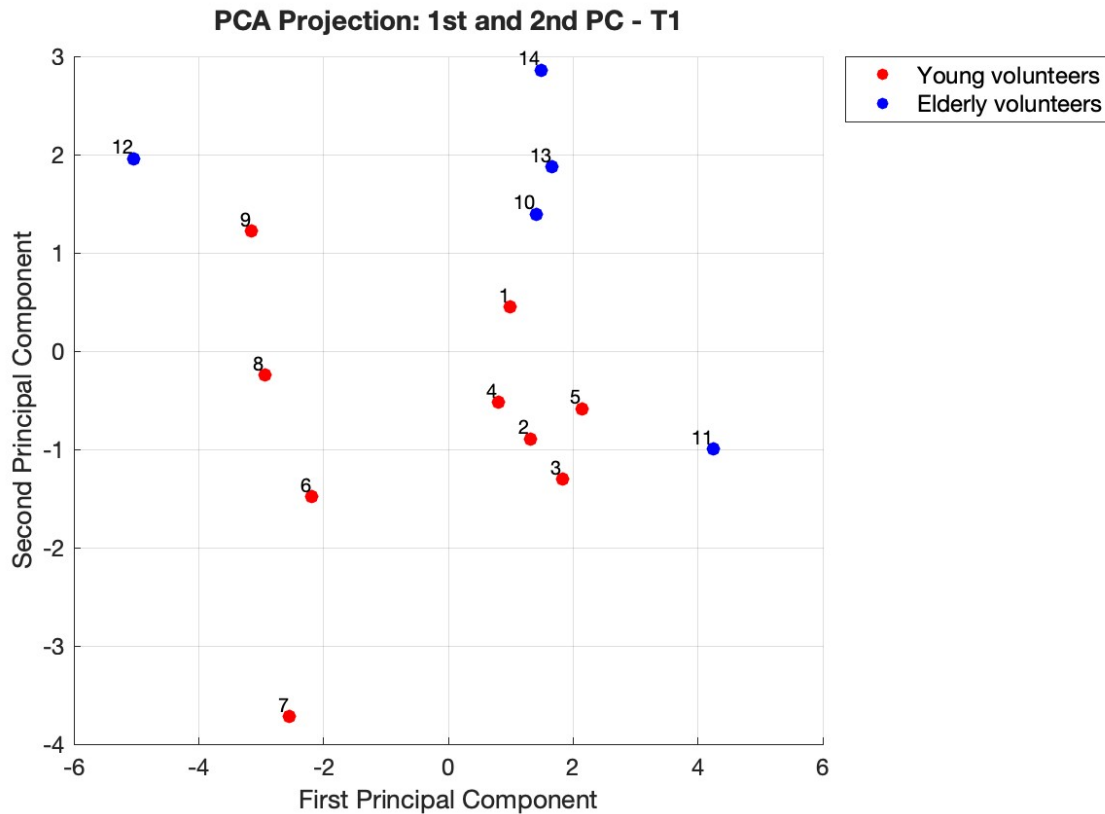


Figure 3.2-3. Projection of T1 landmark results onto the first two principal components. Each point represents one test trial, color-coded by subject group (young vs. elderly). While group tendencies begin to emerge, overlap remains high and interpretation is non-quantitative.

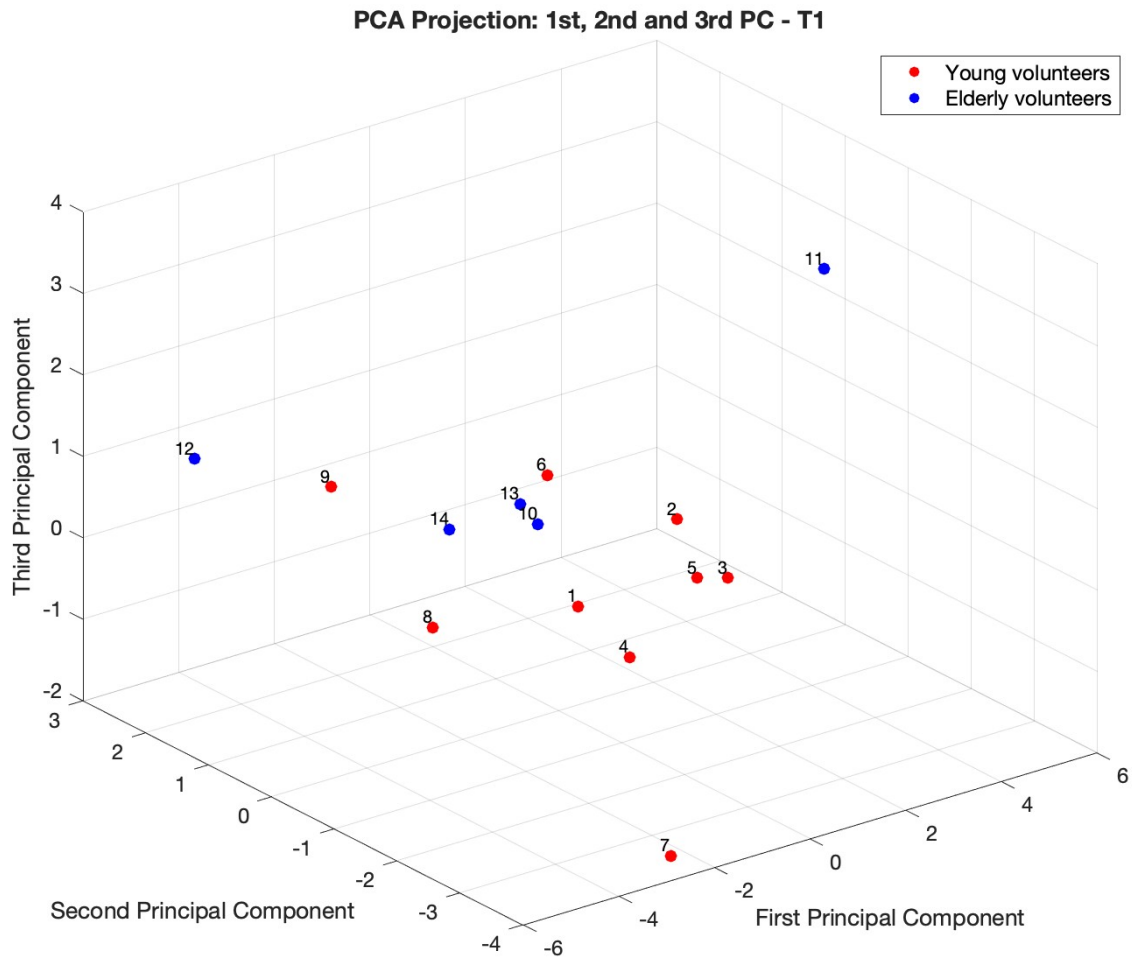


Figure 3.2-4. 3D scatter plot of PCA scores on components 1–3 for the T1 landmark. The projection provides additional insight into the distribution of trajectory variation but does not offer conclusive group separability.

These plots provide an initial indication that age-related differences may be reflected in the trajectory structure, but they do not establish statistical significance or quantify the strength of the relationship. For that reason, further modeling via regression is required to assess whether specific principal components can be explained by anthropometric or demographic variables.

Regression modeling

The final stage of the pipeline involved constructing linear regression models to assess whether variations in trajectory shape, as captured by principal component scores, could be explained by subject-specific characteristics. This was done independently for each landmark and for each of the retained components (typically PC1 and PC2).

Data preparation

Principal component scores were exported from MATLAB into a structured Excel file, where each row corresponded to a single test trial. Alongside the scores, the following subject characteristics were included:

- Kind (categorical: young or elderly)
- Age (years)
- Stature (cm)
- Weight (kg)
- Seated height (cm)
- Head girth (cm)
- Neck girth (cm)

The structure of the spreadsheet grouped data by landmark and aligned all variables in a format suitable for direct import into GRETL.

Modeling procedure

Linear regression models were constructed using GRETL. For each landmark and each principal component, the PC score was treated as the dependent variable, and the set of subject characteristics was tested as potential predictors.

Initially, multiple-variable models were tested using combinations of the predictors listed above. However, these models consistently demonstrated poor fit, unstable coefficients, or strong multicollinearity. As a result, the final approach focused on single-variable models, which consistently outperformed the multivariate alternatives in terms of parsimony, statistical significance, and interpretability.

Each model then followed the form:

$$z_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

where z_i is the score for a given principal component and trial i , and x_i is a single subject characteristic. Separate models were estimated for each landmark and each retained PC.

Output handling and model selection

The regression outputs included coefficient estimates, p-values, residual diagnostics, and goodness-of-fit metrics. From all models evaluated, those with statistically significant predictors and stable fit were flagged for reporting in the Regression modeling results section.

Attempts were also made to reconstruct trajectories by projecting predicted PC scores back into the spline control point space, but the prediction errors were found to be prohibitively high. As such, these reconstruction experiments were not retained for final analysis.

3.3. Method 2: Landmark relative positioning analysis at peak displacement

3.3.1. Method overview

The methodology presented in this section was developed to complement the trajectory-based B-spline analysis by offering a different perspective on postural variation during frontal deceleration. Instead of analyzing landmark motion across entire time series, this approach focuses on spatial configurations at selected time frames. Its goal is to characterize the relative positions of torso landmarks during a key instant in the deceleration event, enabling both visualization and statistical exploration of group-level morphological differences.

The analysis begins by examining the planar (x, z) trajectories of five anatomical landmarks (Head, C4, T1, T4, T8), recorded during 14 test trials involving young and elderly volunteers. For each trial, these trajectories are expressed relative to the initial position of the head to emphasize relative rather than absolute motion. This normalization step is intended to isolate torso configuration independently of initial positioning within the global reference frame. The H-point landmark was excluded from this analysis due to its inconsistent trajectories and excessive spatial separation from the other landmarks, which could distort subsequent comparisons.

To identify suitable instants for posture analysis, each trajectory set is sampled at a fixed number of uniformly spaced time frames. These snapshots represent the evolution of the landmark configuration from the onset of motion to the point of maximum forward displacement. Among these, a single frame corresponding to the maximum anterior displacement of the head is selected for further analysis. This time point is treated as a proxy for the moment of peak torso excursion, where individual morphological and postural differences are expected to be most pronounced. Although this frame selection was guided empirically by observing head displacement patterns, it can be biomechanically justified on the basis that the head tends to lead upper-body motion in frontal impacts, and its maximum excursion often coincides with the culmination of torso flexion. This choice ensures that the posture being analyzed is representative of the peak response phase.

At this selected time frame, the (x, z) coordinates of the five landmarks are extracted and compiled into a feature vector representing the instantaneous torso configuration. To eliminate differences caused by overall body displacement, all coordinates are shifted so that the T8 landmark is at the origin. This allows the analysis to focus on the relative positions of the other landmarks.

Principal Component Analysis (PCA) is then applied to the set of re-referenced configuration vectors. This transformation identifies dominant modes of shape variation across subjects, enabling a compact representation of postural differences. The first three principal components are retained for further analysis, as they account for the majority of inter-subject variability. Each trial is thus characterized by a set of three uncorrelated scores, which quantify how its torso posture differs along key geometric axes.

To explore potential associations between posture and subject characteristics, linear regression models are constructed using each principal component score as a response

variable. Independent variables include age, anthropometric dimensions, and categorical group membership. These models are used to test whether the observed modes of variation in posture are systematically related to individual features. As in previous analyses, multivariable models were avoided due to the limited sample size and risk of overfitting. Instead, single-variable regressions were prioritized to enhance statistical robustness.

Finally, a complementary analysis was performed to examine spatial configurations of the torso at the moment of maximum head displacement (MHD). After having translated each trial's data so that the T8 landmark coincided across subjects, two types of metrics were extracted. First, Euclidean distances between adjacent landmarks were computed to evaluate segment-wise spacing across age groups. Second, internal joint angles were calculated at key spinal junctions to capture localized curvature and postural differences. Statistical comparisons were carried out using two-sample t-tests. This dual-metric procedure was motivated by the hypothesis that aging may influence not only global motion trajectories, but also the geometric organization and flexion patterns of the upper torso at peak loading.

Overall, this method offers a complementary lens for analyzing subject-specific differences in seated posture during dynamic loading. By isolating and analyzing the torso configuration at maximum displacement, it enables biomechanical interpretation of posture patterns and supports statistical modeling of group differences in geometric configuration.

3.3.2. Literature background

Several strands of existing research support the methodological choices made in this analysis, including the use of peak displacement timing for frame selection, the application of PCA to posture data, the incorporation of anthropometric variables for group comparisons, and the practice of re-referencing landmark coordinates to isolate shape differences. This section reviews the main contributions from the literature that justify and contextualize each of these components.

Use of maximum head displacement for time selection

The selection of a single time frame based on the instant of maximum head displacement (MHD) draws from established practices in impact biomechanics. Studies examining cervical spine motion during frontal deceleration have consistently shown that maximum head excursion occurs at a predictable interval after impact and corresponds closely to peak torso flexion, providing a reliable temporal marker for postural comparison (FREJ23).

This approach is particularly valuable in volunteer testing, where the severity of impacts is constrained by ethical considerations. In such contexts, MHD has been validated as a meaningful reference point for assessing postural deviation without requiring full trajectory analysis (HIGU19). These findings support the use of head displacement as a biomechanical proxy for maximum whole-body excursion during frontal loading.

PCA applied to spatial configurations of landmarks

PCA has been widely adopted to analyze human posture, particularly in reducing the complexity of spatial marker data into interpretable modes of variation. Federolf et al. (2016) introduced the concept of “principal movements” by applying PCA to kinematic data from balance tasks, showing that a small number of components could capture nearly all postural variance (FEDE16).

Similar applications have been used in impact biomechanics, where PCA has identified dominant posture configurations, referred to as “principal positions”, that effectively discriminate between subject groups and loading conditions (SUN16).

These studies demonstrate that PCA can be successfully applied not only to trajectories but also to static spatial configurations extracted at key time points, enabling dimensionality reduction without loss of biomechanical relevance.

3.3.3. Purpose and justification

The motivation behind this method is to identify systematic differences in torso posture between young and elderly subjects during low-speed frontal decelerations. Rather than analyzing full motion trajectories, this approach focuses on a single, carefully selected time frame that captures the peak of the event, making it particularly suited for summarizing posture under dynamic conditions without requiring time-normalized data. By isolating and analyzing the configuration of the upper torso at the moment of MHD, the method aims to uncover variations in spinal alignment or segment relationships that may reflect age-related differences in posture control or mechanical response.

The decision to rely on spatial landmark data at a specific time point (rather than using time series or curve-based representations) was guided by both practical and analytical considerations. In practical terms, working with single-frame data reduces the dimensionality of the dataset, avoids the need for time alignment or warping, and provides direct interpretability in terms of posture. Analytically, it enables the use of PCA to identify compact, orthogonal modes of variation that summarize the configuration of the torso in a data-driven way. This supports the development of regression models that can assess whether observed variations correlate with age or anthropometric features.

Although the use of the MHD frame was not determined through biomechanical modeling, it is justified based on empirical and physiological reasoning. The head is typically the most mobile and forward-reaching segment in frontal impacts, and its peak displacement tends to coincide with maximum torso flexion. Selecting this instant as a common reference frame therefore captures the posture at or near the mechanical limit of excursion, where differences in control, flexibility, or morphology are most likely to appear.

Re-referencing to a stable landmark such as T8 is justified as a way to remove global translation effects and allow for comparison of shape and configuration independently of whole-body motion. This is particularly relevant in seated frontal deceleration tests, where volunteers may exhibit variable forward excursions due to differences in muscle

activation or restraint interaction, but the relative shape of the spine remains a more stable indicator of biomechanical behavior.

Additionally, the analysis includes a secondary examination of inter-landmark distances and angles, re-referenced to a common anatomical point. This was introduced to evaluate whether specific spinal segments exhibit consistent spatial/angular divergence between groups.

Overall, the method was designed to be complementary to the B-spline-based trajectory analysis presented earlier, offering a posture-centric view of group differences. It supports a more intuitive understanding of how spinal alignment varies at peak excursion and enables interpretable statistical modeling of the effects of aging and morphology on torso configuration.

3.3.4. Mathematical background

This analysis operates on posture vectors constructed from anatomical landmark positions at a single time frame per subject, the moment of MHD. Each vector captures the torso's spatial configuration in the sagittal plane and serves as input for dimensionality reduction via PCA. Unlike the earlier spline-based trajectory analysis, this method is temporally discrete and emphasizes shape rather than motion.

Construction of posture vectors

Let $N = 14$ denote the number of test trials and $L = 5$ the number of landmarks: Head, C4, T1, T4, and T8. For each trial i , the two-dimensional coordinates of each landmark at MHD are denoted:

$$(x_{i,l}, z_{i,l}) \quad \text{for } l = 1, \dots, L \text{ and for } i = 1, \dots, N$$

To eliminate global displacement effects, all positions are re-referenced relative to the T8 landmark:

$$\tilde{x}_{i,l} = x_{i,l} - x_{i,T8}, \quad \tilde{z}_{i,l} = z_{i,l} - z_{i,T8}$$

The resulting relative coordinates are flattened into a row vector:

$$\mathbf{p}_i = [\tilde{x}_{i,1}, \tilde{z}_{i,1}, \dots, \tilde{x}_{i,L}, \tilde{z}_{i,L}] \in \mathbb{R}^{2L}$$

These posture vectors form a data matrix:

$$\mathbf{P} \in \mathbb{R}^{N \times 2L}, \quad \text{with rows } \mathbf{p}_i$$

This representation captures the torso's posture at MHD, relative to a common reference point, and standardized across subjects.

Principal Component Analysis

The matrix $\mathbf{P}$ is mean-centered, and PCA is applied as described in Method 1's Principal Component Analysis section. Briefly, this involves computing the sample covariance matrix of $\mathbf{P}$, performing eigenvalue decomposition, and projecting each subject's posture vector onto the resulting eigenvectors. Let:

$$\mathbf{Z} = \mathbf{P}_c \mathbf{V}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times K}$ is the matrix of PCA scores, and $K = 3$ the principal components that are kept for further analysis.

Each score $z_{i,j}$ quantifies how much subject i 's posture deviates along the j -th principal axis of shape variation. Because the PCA is performed on re-referenced data, these axes reflect geometric configurations independent of absolute position.

Regression of Principal Component scores

To explore whether variation in postural configuration is related to subject characteristics, the scores of the first three principal components are used as dependent variables in simple linear regression models. The structure and estimation of these models follow the same formulation described in Method 1's Regression of PCA-derived features to anthropometr section, and are not repeated here.

Each regression model includes a single predictor variable, such as age, head girth, or categorical group (young vs. elderly), selected based on interpretability and fit quality. Multivariable models were tested but consistently showed poor performance, and thus only single-variable regressions are retained.

Pairwise landmark distance analysis

In a secondary analysis, both linear distances and joint angles between consecutive landmarks at MHD are computed for each subject.

Inter-Landmark distances

Given re-referenced coordinates $(\tilde{x}_{i,l}, \tilde{z}_{i,l})$, the Euclidean distance between landmarks l and $l + 1$ is:

$$d_{i,l} = \sqrt{(\tilde{x}_{i,l+1} - \tilde{x}_{i,l})^2 + (\tilde{z}_{i,l+1} - \tilde{z}_{i,l})^2}$$

This yields four distance values (between 5 landmarks) per subject:

$$\{d_{i,1}, d_{i,2}, d_{i,3}, d_{i,4}\}$$

Distances are grouped by age category (young vs. elderly) and compared using a two-sample t-test. Let D_y and D_e be the subsets of distances for young and elderly subjects, respectively. For each pair of landmarks, the test evaluates:

$$H_0: \mu_y = \mu_e \quad vs. \quad H_1: \mu_y \neq \mu_e$$

The corresponding p-values quantify whether observed differences in mean spacing are statistically significant.

Joint angles

To assess curvature at key spinal junctions, angles were computed at three internal landmarks using triplets of adjacent points. For three points $A = (x_a, z_a)$, $B = (x_b, z_b)$ and $C = (x_c, z_c)$, the internal angle at B is given by:

$$\theta = \cos^{-1} \left(\frac{(\vec{A} - \vec{B}) \cdot (\vec{C} - \vec{B})}{\|\vec{A} - \vec{B}\| \cdot \|\vec{C} - \vec{B}\|} \right)$$

where $\vec{A}, \vec{B}, \vec{C} \in \mathbb{R}^2$ denote 2D coordinate vectors in the sagittal plane. This yields three angles per subject:

$$\{\theta_{C4}, \theta_{T1}, \theta_{T4}\}$$

Statistical comparisons between age groups were again performed using two-sample t-tests for each angle.

3.3.5. Implementation

Overview of workflow

This method was implemented through a four-stage process designed to extract torso posture at the moment of maximum head displacement (MHD) and analyze inter-subject variation using Principal Component Analysis and group-level statistical modeling. The complete pipeline included the following steps:

1. Selection of the MHD frame and extraction of landmark positions,
2. Dimensionality reduction via PCA,
3. Regression modeling of the resulting scores, and
4. Group comparison of inter-landmark distances and joint angles.

Each step was performed using custom MATLAB routines and supporting software, and applied individually across the 14 available test trials.

Posture vector construction

The MATLAB script `Time_based_noHPoint.m` (which can be found in section 3.2.1 of the Annex) was used to process the raw trajectory Excel files for each subject. The script loaded the raw (x, z) trajectories of five landmarks (Head, C4, T1, T4, T8) and allowed the user to specify the number of time frames to extract per trial. A value of 10 evenly spaced frames was selected to ensure adequate temporal resolution while minimizing data redundancy.

For each test, the frame corresponding to maximum anterior head displacement was identified automatically. An example of landmark trajectories and the identified MHD frame is shown in Figure 3.3-1.

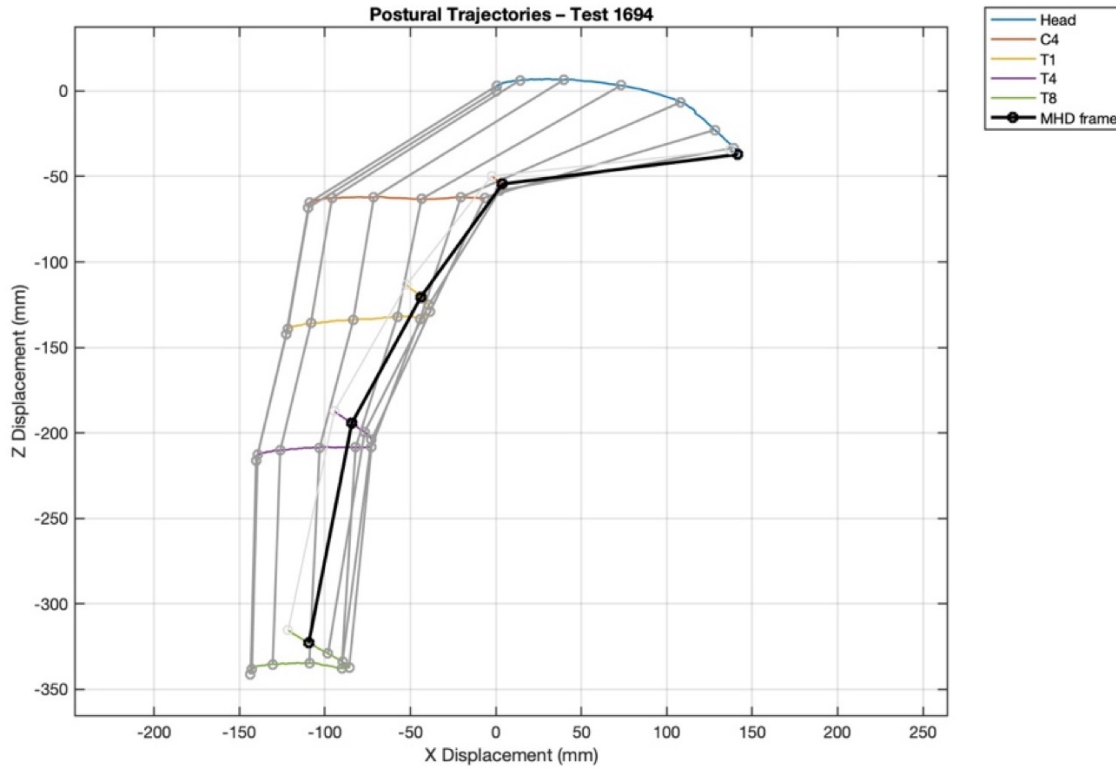


Figure 3.3-1. Example of landmark trajectories during a test, with the frame of maximum head displacement (MHD) highlighted in black. Dark gray and light gray indicate frames before and after MHD, respectively.

The (x, z) coordinates of the five landmarks at this frame were extracted and re-referenced relative to the T8 landmark to remove global translation. These ten values (5 landmarks $\times$ 2 coordinates) were flattened into a row vector and stored for each trial.

This process yielded a posture matrix $\mathbf{P} \in \mathbb{R}^{14 \times 10}$, with each row representing the configuration of the upper torso at MHD in a standardized reference frame.

Principal Component Analysis

The PCA was carried out using the MATLAB script

Pca_analysis_p2.m (found in section 3.2.2. of the Annex), which takes in the matrix $\mathbf{P}$ as input. The data were mean-centered prior to decomposition, and the analysis was restricted to the first three principal components, which collectively explained the majority of variance in the sample.

The resulting score matrix $\mathbf{Z} \in \mathbb{R}^{14 \times 3}$ was saved for use in the regression stage. These components were interpreted as specific modes of shape variation reflecting differences in curvature, alignment, or elevation between landmarks, at MHD.

Regression modeling

The PCA scores were exported to an Excel file where each row corresponded to a single test trial and each column represented one of the three retained principal components. The file also included subject-specific variables such as age, kind (young vs. elderly), head girth, and other anthropometric measures.

As in Method 1's Regression modeling implementation section, regression models were implemented in GRETL. Each principal component was modeled separately using one independent variable, selected based on empirical performance. Multivariable models were tested but consistently produced unstable or uninterpretable results, and were therefore discarded.

Inter-landmark distances and angles analysis

The final section of *Time_based_noHPoint.m* prompts the user to select a reference landmark for alignment. This feature was used to visualize the torso configurations of all subjects at MHD, re-centered around a common anatomical point (typically C4 or T1). The goal was to enable qualitative inspection of group-level differences in posture.

The quantitative analysis was carried out using the MATLAB script *Relative_positions_analysis.m* (available in section 3.2.3 of the Annex), which computed the Euclidean distances between each pair of adjacent landmarks for every subject, as well as joint angles for all triplets of consecutive landmarks. These distances and angles were grouped by age category (young vs. elderly), and two-sample t-tests were performed to assess statistical significance.

3.4. Method 3: Arc-Length Re-Parameterization and Signal Registration for generation of average trajectories and deviation corridors

3.4.1. Method Overview

This final method aims to characterize the central tendency and variability of landmark trajectories by constructing a representative average path and corresponding confidence corridors. Unlike the approaches presented in previous sections, the present technique maintains the full curve geometry while providing statistical descriptors of its dispersion across subjects. It does so by leveraging arc-length as a geometric parameter, rather than time, to re-parameterize the trajectories. This allows for meaningful comparison and averaging even when signal durations vary, or when timing is inconsistent across trials due to inter-subject variability.

The technique follows the methodology described by Hartlen and Cronin (2022), and is implemented via the ARCGen toolbox (HART22), available in section ARCGen of the Annex. ARCGen is specifically designed to compute characteristic averages and statistical corridors for sets of biomechanical curves that may be non-monotonic, multi-dimensional, or irregularly sampled. Its three-stage process (arc-length re-parameterization, signal registration, and statistical envelope construction) is well-suited to the kinematic data in this study.

The first stage, arc-length re-parameterization, transforms each input trajectory into a representation where position is a function of normalized arc length, rather than time. This is particularly valuable for signals that vary in speed or duration across subjects, as it ensures that key geometric features (e.g., peaks or inflection points) are aligned according to their pathwise progression rather than absolute timing. Normalization of arc length to the unit interval $[0, 1]$ enables uniform sampling across signals with different total lengths, facilitating subsequent averaging.

In the second stage, optional signal registration is performed. While arc-length normalization already provides a degree of alignment, it does not guarantee that corresponding features (such as the peak forward excursion of the head) occur at exactly the same normalized arc-length across all signals. To address this, ARCGen applies a set of signal-specific warping functions that subtly shift the resampling positions to improve feature alignment. These warping functions are constrained to be strictly monotonic, preserving the sequence of points along the trajectory. Their flexibility is controlled by two key parameters: the number of interior control points and a penalty factor that limits excessive distortion. Both parameters were manually set based on the shape complexity of the trajectories and the expected number of salient inflection points.

Finally, once all signals are represented in a common arc-length domain and optionally registered, ARCGen performs statistical analysis at each resampled arc-length location. At every normalized arc-length value, it computes the mean and standard deviation of the x and z coordinates across subjects, assuming a local bivariate normal distribution. These statistics define an elliptical confidence region at each point along the mean trajectory. The union of these ellipses forms an envelope, or “corridor,” that represents the inter-subject variability of the signal set. Because there is no analytical expression for the

envelope of such overlapping ellipses, a marching-squares algorithm is used to numerically extract the outer boundary. The final output consists of the characteristic average trajectory, and the inner and outer bounds of the statistical corridor.

In summary, the arc-length-based method introduced in this section offers a complementary perspective to the previous analyses by operating directly on the full geometry of each trajectory. It enables robust averaging of motion paths across subjects by using shape-based alignment rather than time-based synchronization, and produces both a representative curve and statistical corridors that describe inter-subject variability. This approach is particularly suitable for analyzing signals with heterogeneous durations and complex features, and is applied here to evaluate group-level trends in upper-body motion during frontal deceleration.

3.4.2. Literature background

The arc-length-based approach implemented in this section builds upon a growing body of work that addresses the limitations of traditional time-domain averaging techniques in biomechanics. When analyzing human motion, signals recorded across subjects often exhibit temporal misalignments, unequal durations, and variations in key feature locations. These discrepancies can distort point-wise averages and undermine the validity of statistical summaries. As a result, alternative frameworks have been developed to enable consistent alignment and representation of such signals, with arc-length reparameterization and signal registration emerging as robust solutions.

A foundational contribution in this area is the previously mentioned study by Hartlen and Cronin (2022), which introduced a unified methodology for computing characteristic averages and statistical response corridors in biomechanical data. Their method combines arc-length normalization with constrained signal registration to produce smooth, feature-preserving mean curves and confidence regions that remain valid even for signals that are non-monotonic or terminate at different points. The approach is demonstrated across a range of applications, including oscillatory head kinematics and hysteretic thoracic force–displacement responses, and is made accessible through the open-source ARCGen software package, available in the Annex section ARCGen. Their work addresses both geometric consistency and statistical rigor, offering an alternative to ad hoc signal averaging or manual segmentation (HART22).

Several related studies have explored similar challenges in other domains. White et al. (2023) applied arc-length methods to tissue compression and impact responses, highlighting their utility in cases where signals do not share a common endpoint or exhibit substantial shape variability (WHIT23).

Meanwhile, Chau et al. (2005) proposed a global registration criterion for biomechanical curves that optimizes time-warping functions to align signal features while minimizing cross-correlation errors (CHAU05).

Dynamic time warping (DTW), as discussed by Lee (2019), represents a related but distinct approach to curve alignment. Although DTW has proven effective in synchronizing features in gait data, it does not offer a probabilistic framework for representing variability, nor does it operate naturally in the geometric domain of signal

shape (LEE19). In contrast, methods based on arc-length re-parameterization provide a direct means of aligning curves along their physical trajectories, making them better suited for biomechanical applications where spatial accuracy is critical.

Together, these studies support the adoption of arc-length and registration-based approaches for analyzing inter-subject motion data. They provide a methodological foundation for the technique applied in this thesis and reinforce its applicability to the study of kinematic variability during low-speed frontal deceleration events.

3.4.3. Purpose and justification

The inclusion of an arc-length-based method in this thesis responds to the need for a shape-focused approach capable of summarizing biomechanical trajectories with minimal distortion, especially when signals vary in duration and structure. In the context of frontal deceleration tests, volunteer trajectories often differ not only in timing but also in the spatial evolution of motion. These differences challenge traditional time-based averaging, which assumes consistent temporal alignment and can obscure key motion features when this assumption fails.

Rather than compressing trajectories into low-dimensional control point sets or isolating a representative frame, this method retains the full geometric path of each signal and compares trajectories on the basis of normalized arc-length. This shift from time to shape as the organizing principle enables meaningful alignment of features such as turning points, inflection changes, and reversals, even when they occur at different times across trials. As a result, the average trajectory produced better reflects the underlying motion pattern and avoids the artificial smoothing that can result from time-domain averaging.

This method is particularly valuable in the present context because it allows for direct comparison of complex landmark trajectories without requiring artificial signal trimming or manual adjustment. It also accommodates non-monotonic motion and inter-subject variation without enforcing uniform test durations or synchronized peak locations. In doing so, it generates interpretable outputs that preserve biomechanical meaning and allow group-level variability to be visualized and compared under consistent spatial conditions.

Ultimately, this approach adds a complementary layer of analysis to those already presented in Methods 1 and 2. It provides a statistical framework that reflects the full curve geometry of upper-body motion during deceleration and supports the investigation of inter-group differences with minimal preprocessing or abstraction.

3.4.4. Mathematical background

The method applied in this section is based on arc-length re-parameterization and signal registration to compute a characteristic average and statistical response corridor from a set of trajectories. Let $\gamma_i(t) = (x_i(t), z_i(t))$ represent the planar trajectory of a given landmark during test i , with $t \in [0, T_i]$ denoting time and T_i the duration of the signal.

Arc-length reparameterization

Since the recorded trajectories may differ in length and progression due to inter-subject variability, to better produce meaningful comparisons, each trajectory is reparametrized with respect to its arc length.

The arc-length function $s_i(t)$ of signal $\gamma_i(t)$ is computed as:

$$s_i(t) = \int_0^t \|\dot{\gamma}_i(\tau)\| d\tau$$

This function measures the cumulative distance traveled along the trajectory up to time t . To standardize across signals, the arc length is normalized by the total length $L_i = s_i(T_i)$, yielding the normalized arc-length parameter:

$$\tilde{s}_i(t) = \frac{s_i(t)}{L_i}$$

This maps each trajectory to a common domain $[0,1]$, where $\tilde{s} = 0$ corresponds to the start of the motion and $\tilde{s} = 1$ to the endpoint. The reparametrized trajectory is then expressed as:

$$\tilde{\gamma}_i(s) = \gamma_i(\tilde{\xi}_i(s))$$

where $\tilde{\xi}_i(s)$ is the inverse of the normalized arc-length function. In practice, all signals are resampled at M equally spaced arc-length values $s_j = \frac{j}{M-1}$, $j = 0, 1, \dots, M-1$. This ensures that each trajectory is represented by the same number of points, aligned by geometric progression rather than time.

Signal Registration

To further improve feature alignment across signals, a non-linear registration step is applied via warping functions. Each reparametrized signal $\tilde{\gamma}_i(s)$ is transformed into a registered version $\hat{\gamma}_i(s)$ using a strictly monotonic warping function $\varphi_i(s)$:

$$\hat{\gamma}_i(s) = \tilde{\gamma}_i(\varphi_i(s))$$

These warping functions are optimized numerically to improve alignment of local features across subjects while penalizing excessive deformation. They are defined by a small number of control points and a penalty parameter, which are manually specified in the ARCGen execution script.

Once all signals are expressed in the same arc-length domain and optionally registered, the average trajectory is computed pointwise:

$$\bar{\gamma}(s_j) = \frac{1}{N} \sum_{i=1}^N \hat{\gamma}_i(s_j)$$

Statistical Corridor Development

To quantify variability around the average, a bivariate normal distribution is fitted at each arc-length location s_j , using the sample covariance matrix Σ_j of the set $\{\hat{\gamma}_i(s_j)\}$ for $i = 1$ to N . The corresponding 95% confidence region is described by the ellipse:

$$\mathcal{E}_j = \{x \in \mathbb{R}^2 \mid (x - \bar{\gamma}(s_j))^T \Sigma_j^{-1} (x - \bar{\gamma}(s_j)) \leq k^2\}$$

with $k^2 = 5.991$, the 95% confidence threshold for the chi-squared distribution with two degrees of freedom.

The full corridor is formed by the union of these ellipses across all arc-length points. Since the envelope cannot be computed analytically, a marching-squares algorithm is used to numerically extract its outer boundary. ARCGen outputs three primary elements: the characteristic average trajectory, and the inner and outer boundaries of the statistical corridor, all of which are used to visualize shape consistency and variation across trials. For a more illustrative explanation of this method's output, see Figure 3.4-1. and Figure 3.4-2.

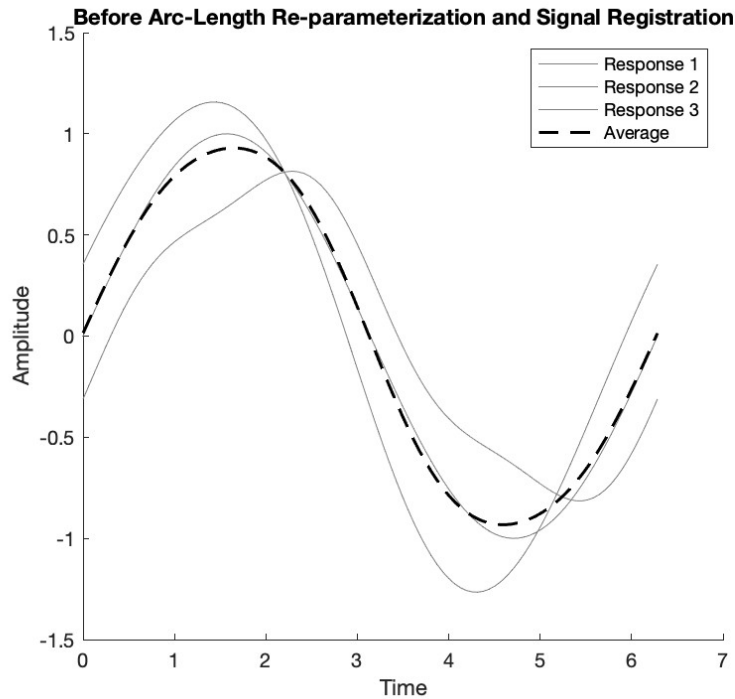


Figure 3.4-1. Three biomechanical response signals before arc-length re-parameterization and registration. The signals display misaligned peaks and valleys, resulting in a less representative average response.

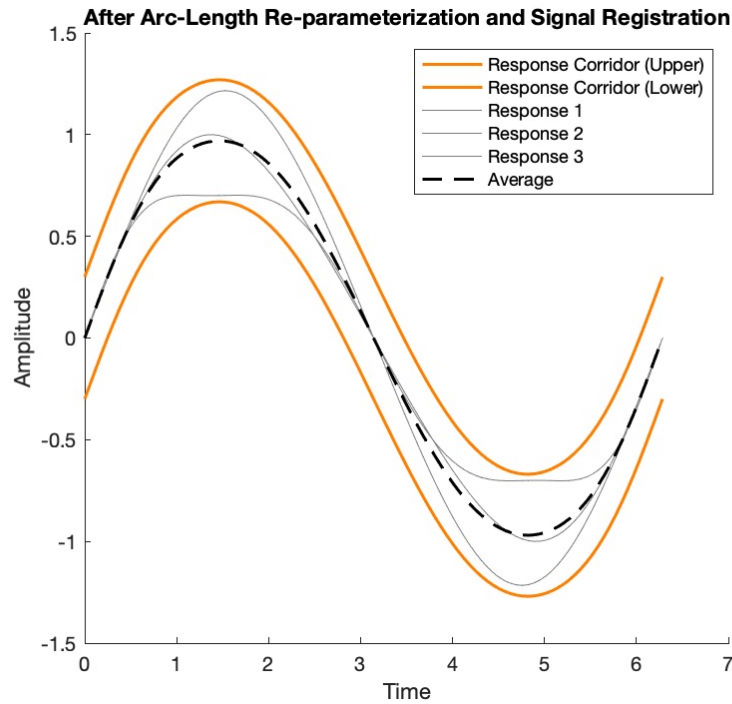


Figure 3.4-2. The same signals after applying arc-length re-parameterization and registration. Temporal alignment of key features is improved while preserving individual shape characteristics.

3.4.5. Implementation

The implementation of the arc-length re-parameterization and corridor analysis was carried out using the ARCGen toolbox provided by Hartlen and Cronin (2022), along with a set of support scripts developed for preprocessing and structuring the raw experimental data (HART22). The full procedure was executed in MATLAB and follows two main stages:

1. Data formatting and signal preparation
2. Batch execution of ARCGen.

Data formatting for ARCGen input and signal preprocessing

The ARCGen toolbox requires that each input trajectory be provided as a separate .csv file, structured as a column vector of 2D coordinates (x, z). Since the original dataset consisted of multi-landmark Excel files (one per test), a preparatory MATLAB script was developed to extract and reshape the data. The script *generate_csvs_from_excels.m* (available in section 3.3.1 of the Annex) reads the raw Excel files containing all landmark trajectories for each test and automatically generates individual .csv files, one per landmark per test.

Once all .csv files were generated, the next step was to compile them into a single MATLAB structure compatible with ARCGen. This was performed using the *PreProcessInputSignals.m* script, which is included with the original ARCGen

distribution, available in section 2.1.2. of the Annex. When executed, the script concatenates all relevant .csv files, and stores them in a .mat file. This file serves as the main input for the following ARCGen execution step.

If the same set of input signals is to be reused across multiple runs, the preprocessing step can be skipped by directly loading the .mat file into the MATLAB workspace. This avoids redundant processing and preserves naming consistency across runs.

Execution and output generation

The computation of the average trajectories and corresponding corridors was carried out using ARCGen's routine, *Arcgen_executer_comparison.m* (section 3.3.2 of the Annex) which was adapted from the open-source file *Arcgen_executer.m*, (section 2.1.3 of the Annex), to be able to process the tests separately by age group (young vs elderly).

During execution, the default ARCGen parameters used in Hartlen and Connin's (2022) study were retained, with the exception of the number of control points used in the warping function, which was set to two. This value allowed for moderate feature alignment without overfitting the warping curves. The algorithm automatically registered the input signals, computed the mean curve across samples, and generated the associated confidence corridors.

Outputs from ARCGen consisted of a characteristic average trajectory and two statistical corridors representing the distribution of observed responses. These curves allowed the comparison of motion consistency across groups and provided an interpretable geometric summary of landmark behavior under frontal deceleration. Figure 3.4-3. displays an example output graph for landmark Head. The complete set of graph is contained in the Group-Specific ARCGen Outputs section.

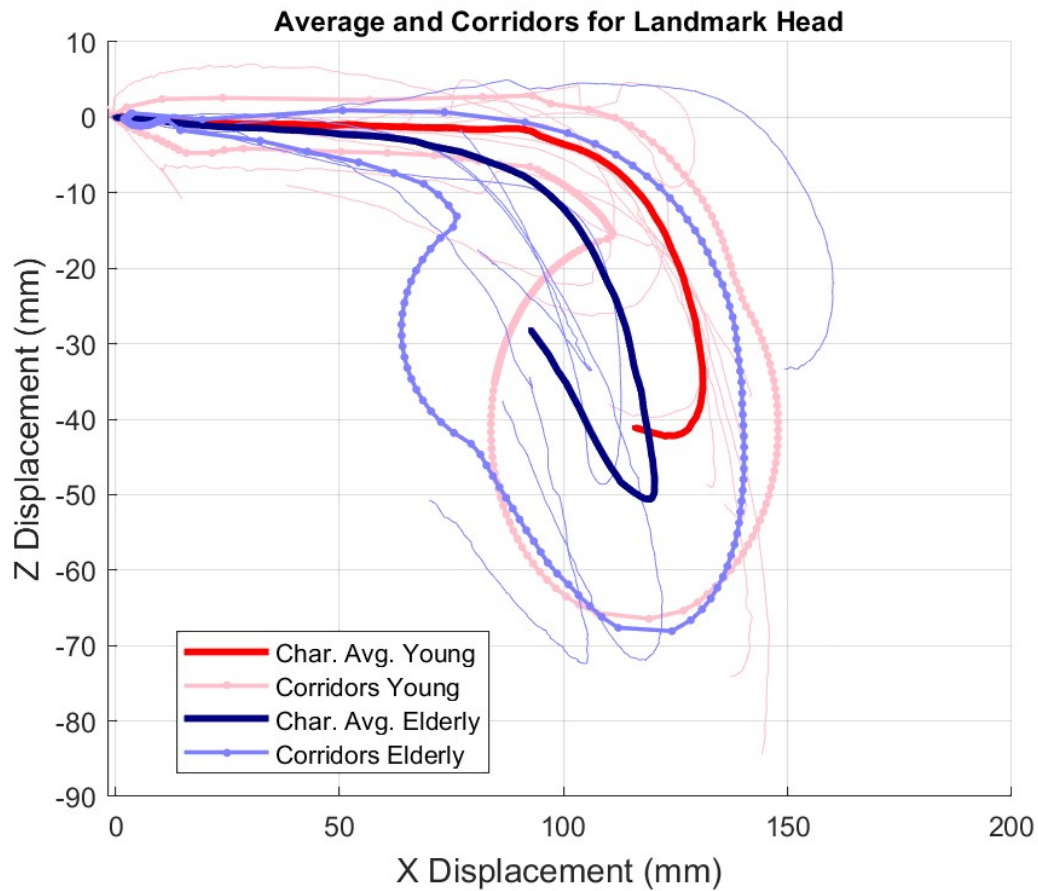


Figure 3.4-3. Example ARCGen outputs for the Head landmark. Average trajectories and 95% statistical corridors are shown for young (red) and elderly (blue) groups.

4. RESULTS

4.1. Method 1: B-Spline approximation and Principal Component Analysis of individual landmark trajectories

4.1.1. Spline fitting and dimensionality reduction

This section presents the outputs of the B-spline fitting process applied to the anatomical landmark trajectories. The goal was to reduce signal dimensionality while preserving key movement patterns relevant for inter-subject comparison and statistical modeling.

For initial reference, Figure 4.1-1. displays the initial trajectories, for landmarks Head through T8, for an example test. This is provided as context of the relative positioning of landmarks against each other, and of real geometric shape of the trajectories. For the shake of mathematical analysis and visualization, the proportions of axes in following graphs may be distorted, which is why this reference graph is provided.

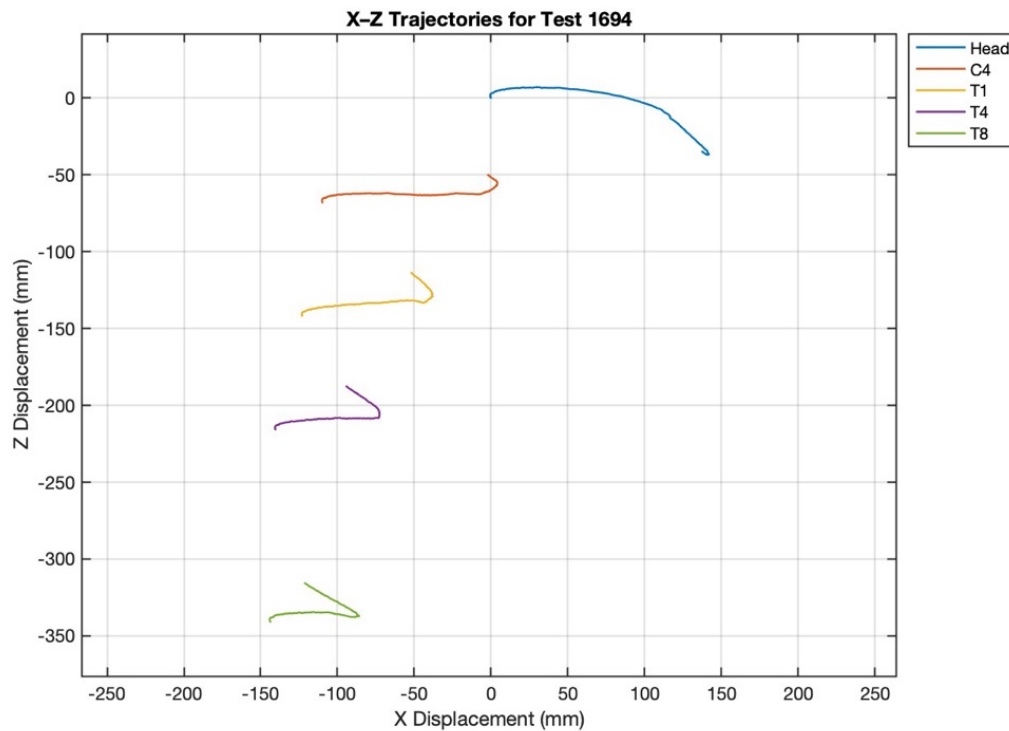


Figure 4.1-1. Raw X–Z trajectories of the five anatomical landmarks (Head, C4, T1, T4, and T8) for each trial. Displacements have been zeroed relative to the initial Head position.

For the coming analysis, individual landmark trajectories are studied separately. Figure 4.1-2. shows the (x, z) trajectories of all six landmarks (Head, C4, T1, T4, T8, H-point) individually for a single test trial. The signals vary substantially in curvature and extent, with some landmarks (e.g., Head) showing larger excursions than others.

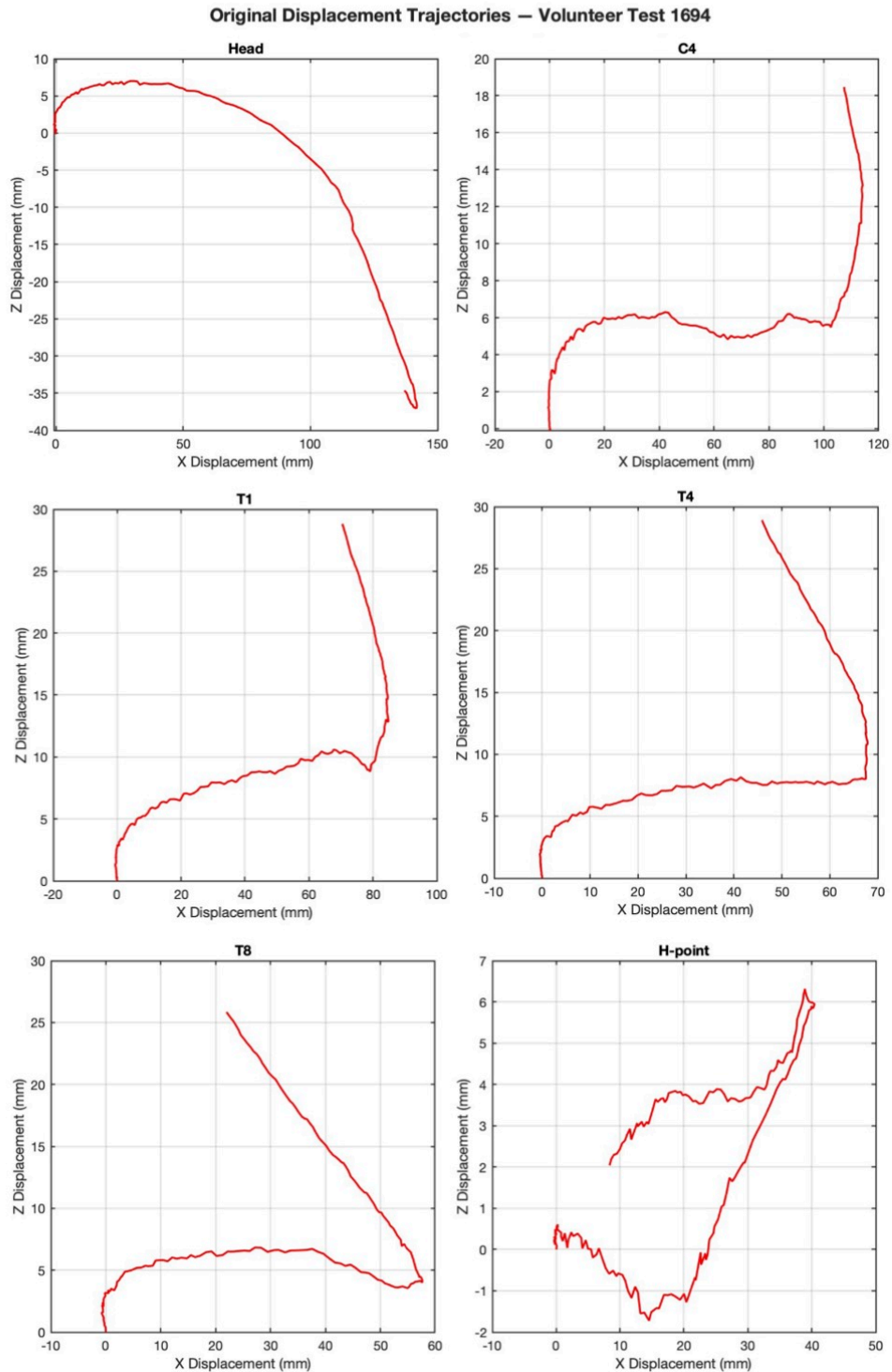


Figure 4.1-2. Original (x, z) motion trajectories of all six anatomical landmarks for a representative trial. Each plot shows the full deceleration interval for a single landmark, as recorded at 1,000 Hz.

Following B-spline fitting, each trajectory was approximated using 8 control points per landmark. Figure 4.1-3. displays the same six trajectories overlaid with their respective

spline approximations. The fitted curves track the main movement paths while smoothing out high-frequency variability.

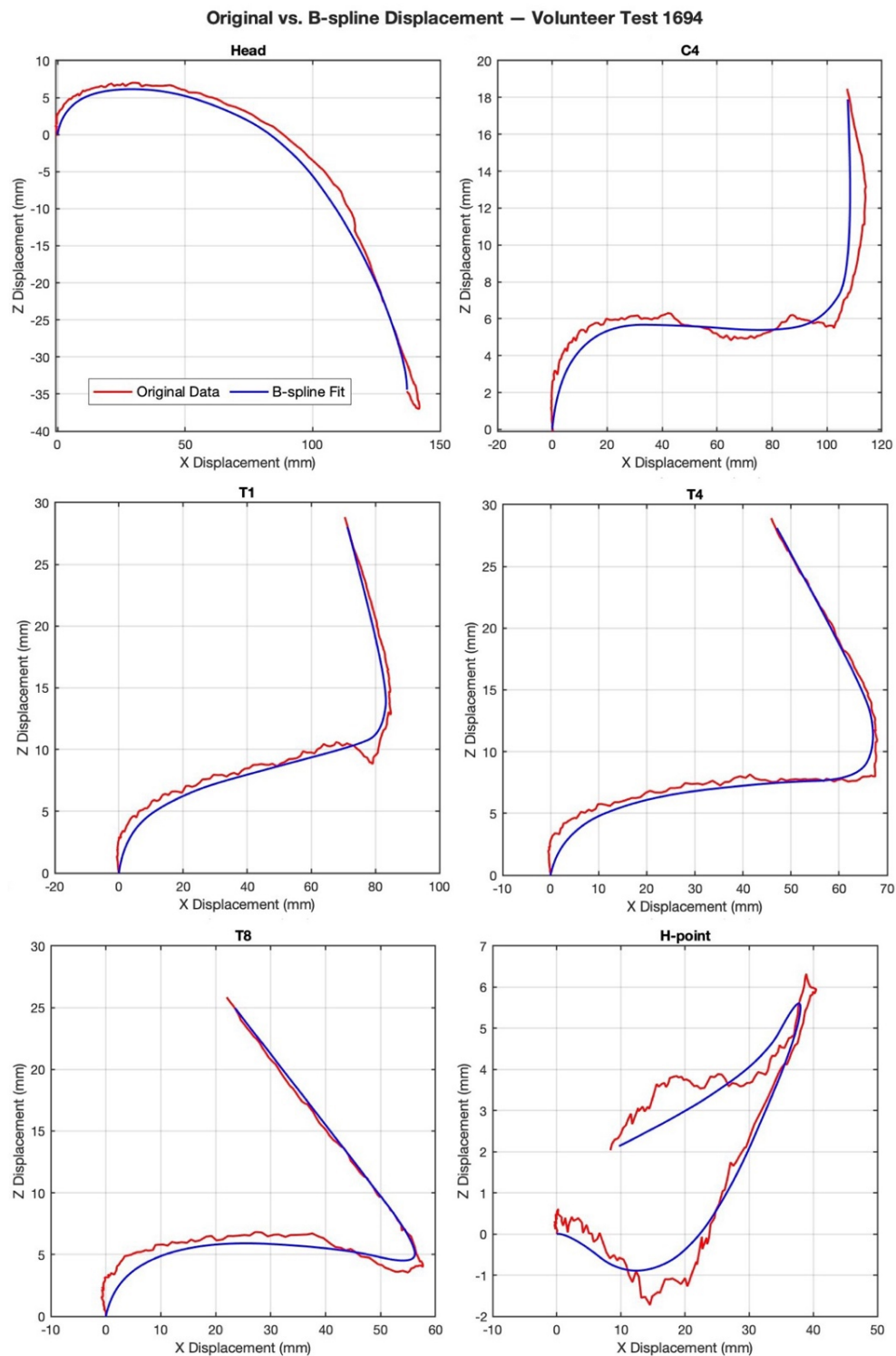


Figure 4.1-3. Same trajectories, now overlaid with cubic B-spline fits. The splines replicate the overall shape of each signal while filtering noise and enabling dimensionality reduction.

To evaluate fit quality, the root mean square error (RMSE) was computed between each original trajectory and its spline approximation, after resampling both signals at 10 uniformly spaced points along the time axis. Table 4.1-1. reports the mean RMSE values across all subjects for each landmark. Error values remained low across the board, confirming that the approximation was consistent and reliable.

Landmark	Mean RMSE (mm)	Standard Deviation (mm)
Head	7.953	3.3235
C4	3.2135	1.7682
T1	3.4282	1.5185
T4	3.4649	1.3725
T8	3.3414	1.276
H-point	1.8192	0.69731

Table 4.1-1. RMSE Summary Table - Mean RMSE between original trajectories and B-spline approximations, averaged over all trials. Evaluation performed on 10-point uniform resampling of each trajectory.

Each fitted trajectory was encoded as a 16-dimensional feature vector, formed by flattening the eight control points into $(x_0, z_0, x_1, z_1, \dots, x_7, z_7)$. These vectors served as standardized inputs to the next analytical stage: Principal Component Analysis.

4.1.2. Principal Component Analysis outcomes

Principal Component Analysis (PCA) was performed independently for each anatomical landmark using the 16-dimensional vectors obtained from the spline control points. The goal was to identify the dominant modes of inter-subject variation and assess the effectiveness of dimensionality reduction.

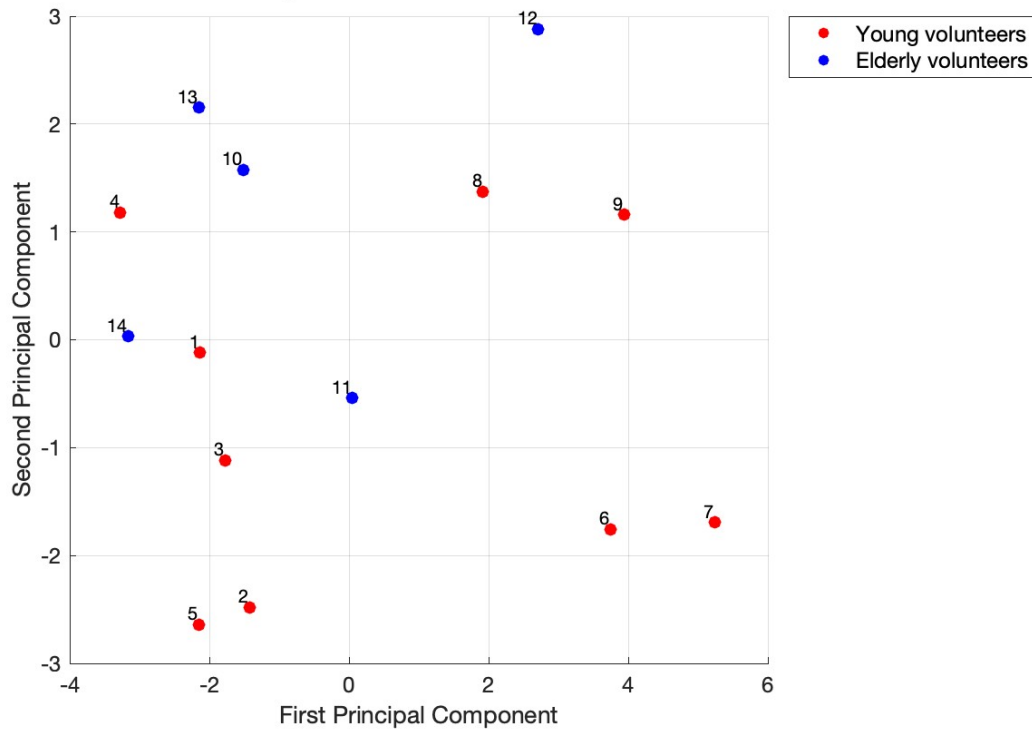
Table 4.1-2. reports the cumulative variance explained by the first three principal components for each landmark. In all cases, PC1 alone captured over 40% of the variance, and the combination of the first three components consistently accounted for more than 80%, confirming that the spline-transformed data contained compact and structured variability.

Landmark	PC1 (%)	PC1 + PC2 (%)	PC1 + PC2 + PC3 (%)
Head	60.45	82.80	89.74
C4	56.15	77.06	86.28
T1	50.27	71.87	84.91
T4	47.55	73.95	84.84
T8	49.13	79.05	89.60
H-point	42.51	66.07	84.27

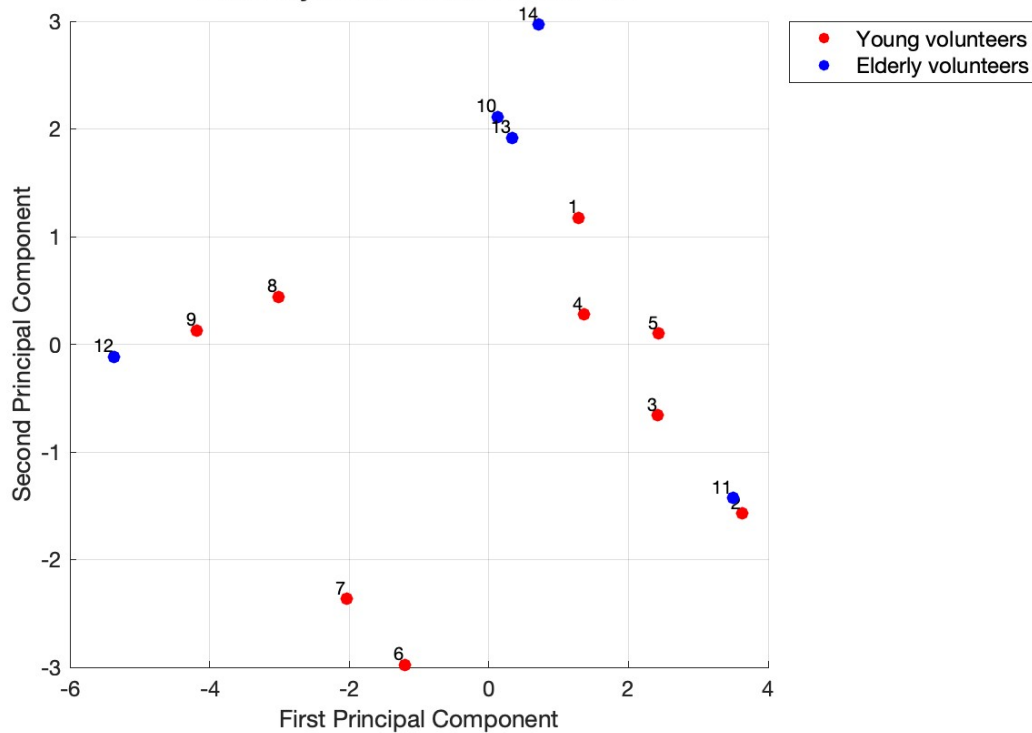
Table 4.1-2. Cumulative variance explained (%) by the first three principal components for each anatomical landmark.

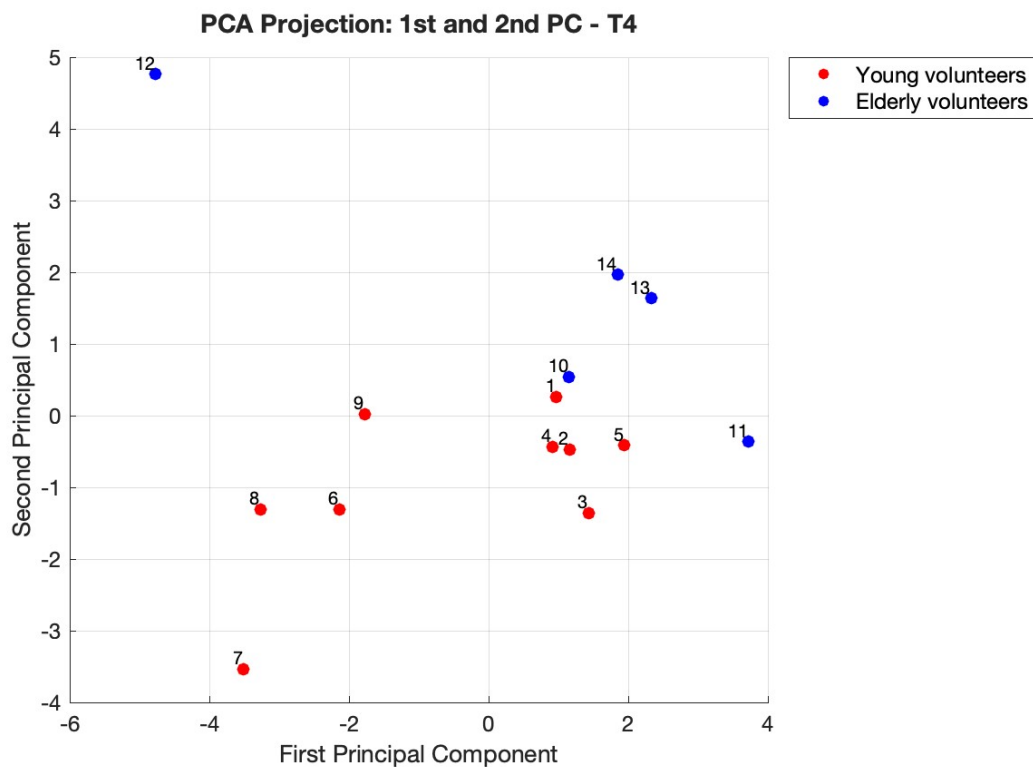
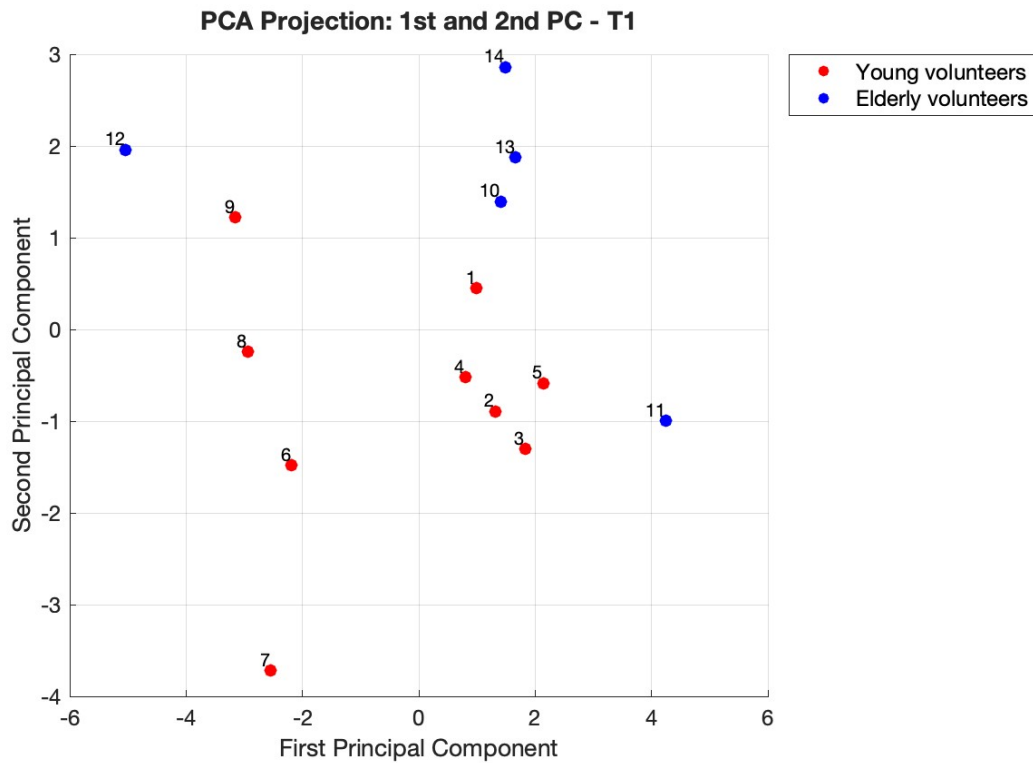
To explore the distribution of subject trajectories in the reduced space, 2D scatter plots were generated for all six landmarks, projecting each trial onto the first two principal components. These plots are shown in Figures 4.1-4, with points color-coded by group (young vs. elderly). In general, the distributions exhibit substantial overlap between groups. Some elderly subjects appear as outliers in certain landmarks (e.g., Subject 14 in T4 and T8), but no consistent visual pattern emerges across landmarks. This suggests that while PCA captured meaningful modes of variation, group-level differentiation based on visual inspection alone is not conclusive.

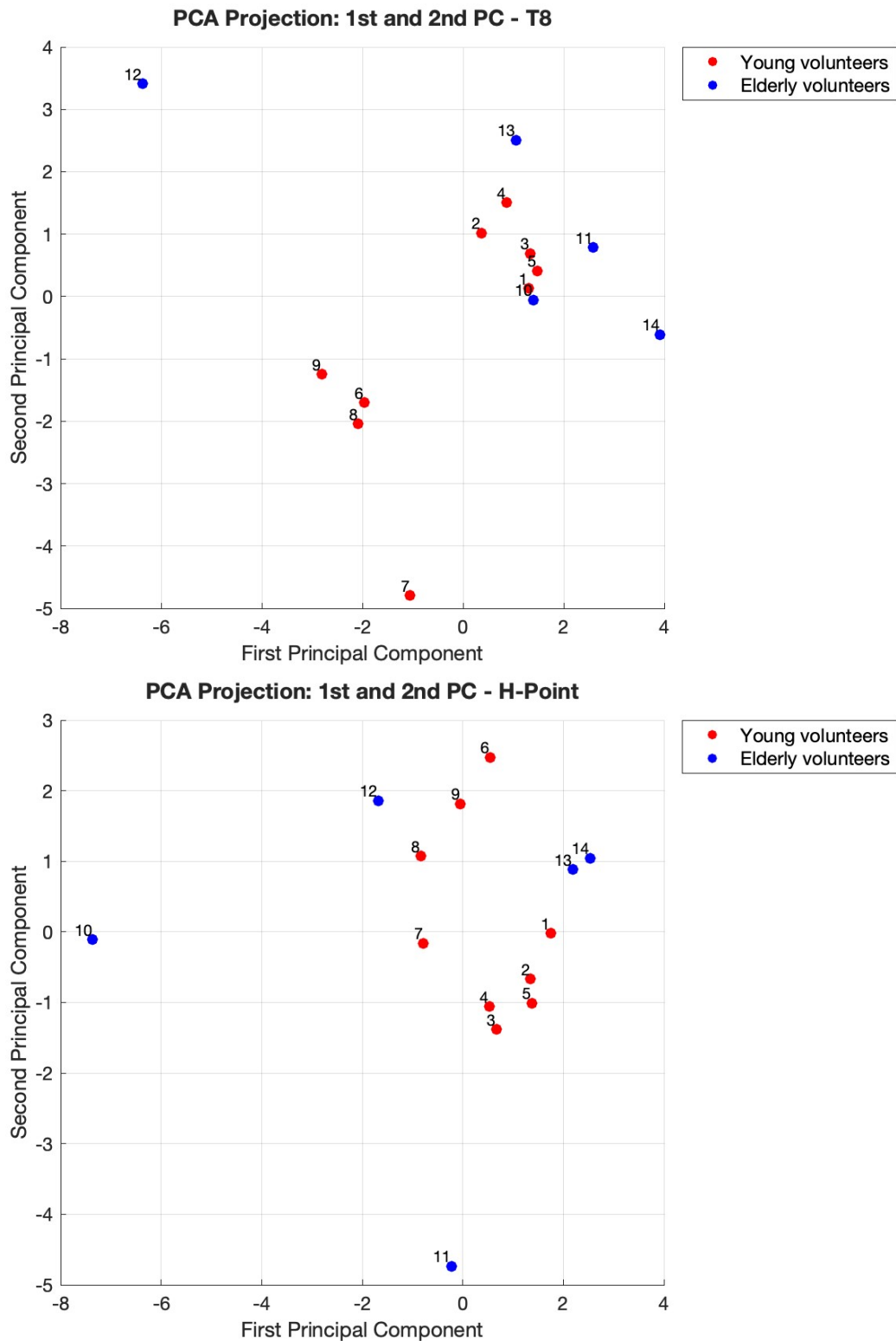
PCA Projection: 1st and 2nd PC - Head



PCA Projection: 1st and 2nd PC - C4







Figures 4.1-4. Projections of subject trajectories onto the first two principal components for all landmarks, individually.

These projections provided a geometric summary of inter-subject differences in motion shape, but they did not support direct interpretation regarding demographic effects. The relationship between principal component scores and subject characteristics was therefore examined through regression analysis in the following section.

4.1.3. Regression modeling results

Following PCA, regression modeling was performed to investigate whether the principal component scores, representing dominant trajectory shape modes, could be explained by subject characteristics. Each retained component (PC1 and PC2) for each landmark was modeled independently using one predictor at a time, as outlined in the Implementation section of Regression modeling. The candidate predictors included age group (“kind”), seated height, and other anthropometric measures. This section presents the models that yielded statistically meaningful associations ($p < 0.05$), as well as critical commentary on models with borderline or inconclusive results. Table 4.1-3. reports the best-fitting model for each landmark and principal component.

Landmark	PC Index	Predictor	Coefficient	p -value	Adjusted R^2
Head	1	Seated height	-0.558	0.0948	0.15
C4	1	Seated height	0.495	0.128	0.114
T1	1	Seated height	0.461	0.1351	0.108
T4	1	Seated height	0.484	0.1037	0.139
T8	1	Seated height	0.274	0.3845	-0.014
H-point	1	Seated height	0.346	0.2305	0.044
Head	2	Kind	1.897	0.0496	0.225
C4	2	Kind	1.696	0.0733	0.18
T1	2	Kind	2.203	0.0158	0.346
T4	2	Kind	2.658	0.0066	0.429
T8	2	Seated height	0.452	0.0494	0.225
H-point	2	Seated height	-0.379	0.0654	0.193

Table 4.1-3. Best one-variable regression models for each landmark and principal component. Grey shading indicates $p < 0.05$.

These results indicate that PC2, not PC1, was more responsive to subject characteristics. Specifically, PC2 of the T1 and T4 landmarks showed strong associations with group category (young vs. elderly), both in terms of effect size and explained variance (adj. $R^2 > 34\%$). The positive coefficients imply that elderly subjects consistently scored higher on these components, suggesting systematic shape differences in mid- and lower-spine trajectories. For T8, seated height was significantly correlated with PC2 scores, though the explained variance remained modest (adj. $R^2 = 22.5\%$).

The regression model for PC2 of the Head also reached statistical significance ($p = 0.0496$), but its adjusted R^2 was also low (22.5%), suggesting that group differences exist but account for only a more modest share of the variability.

In contrast, none of the PC1 regressions achieved significance at $p < 0.05$, despite visible trends in scatter plots. Several models, such as PC1 of the Head ($p = 0.0948$), C4 ($p = 0.128$), T1 ($p = 0.1351$), and T4 ($p = 0.1037$), showed weak-to-moderate associations with seated height, but their low explanatory power (adjusted $R^2 < 15\%$) and borderline significance do not warrant inclusion. These may suggest biomechanical trends worth exploring with larger samples, but cannot support strong claims.

No models involving other predictors (e.g., age in years, neck girth, body mass) yielded statistically significant results, and no multivariable models were retained due to multicollinearity or unstable coefficients. Attempts to predict full trajectories by reconstructing spline control points from PCA regression outputs were also explored, but the resulting errors were prohibitively large and have not been included in the final analysis.

4.2. Method 2: Landmark relative positioning analysis at peak displacement

4.2.1. Frame selection and posture vector extraction

This subsection reports the initial outputs of Method 2, focused on isolating subject posture at the frame of maximum forward excursion. For each trial, a single time frame was selected corresponding to the maximum anterior displacement of the Head landmark in the x -direction. The goal was to extract a posture snapshot that best captured inter-subject variability in spinal alignment during peak deceleration.

The analysis excluded the H-point landmark due to its irregular trajectory and spatial distance from the rest of the torso markers, which risked distorting posture comparisons. This decision was based on visual inspection of trajectory data and confirmed during preprocessing, where H-point motion appeared inconsistent across trials and decoupled from upper-body dynamics.

The frame selection process started by loading the (x, z) trajectories for the five remaining landmarks (Head, C4, T1, T4, and T8) from each test. For each trial, ten evenly spaced time frames were identified along the deceleration event. The instant at which the head reached its furthest forward displacement was also detected and used to extract posture.

Figure 4.2-1. shows the output for a representative trial (Subject 1694). The five landmark trajectories are visualized as sagittal segments across the deceleration event, with the MHD frame shown in black, and preceding/following frames in dark and light gray, respectively. This visualization provides insight into the progression of torso motion and allows verification that the selected frame indeed captures the moment of peak head displacement.

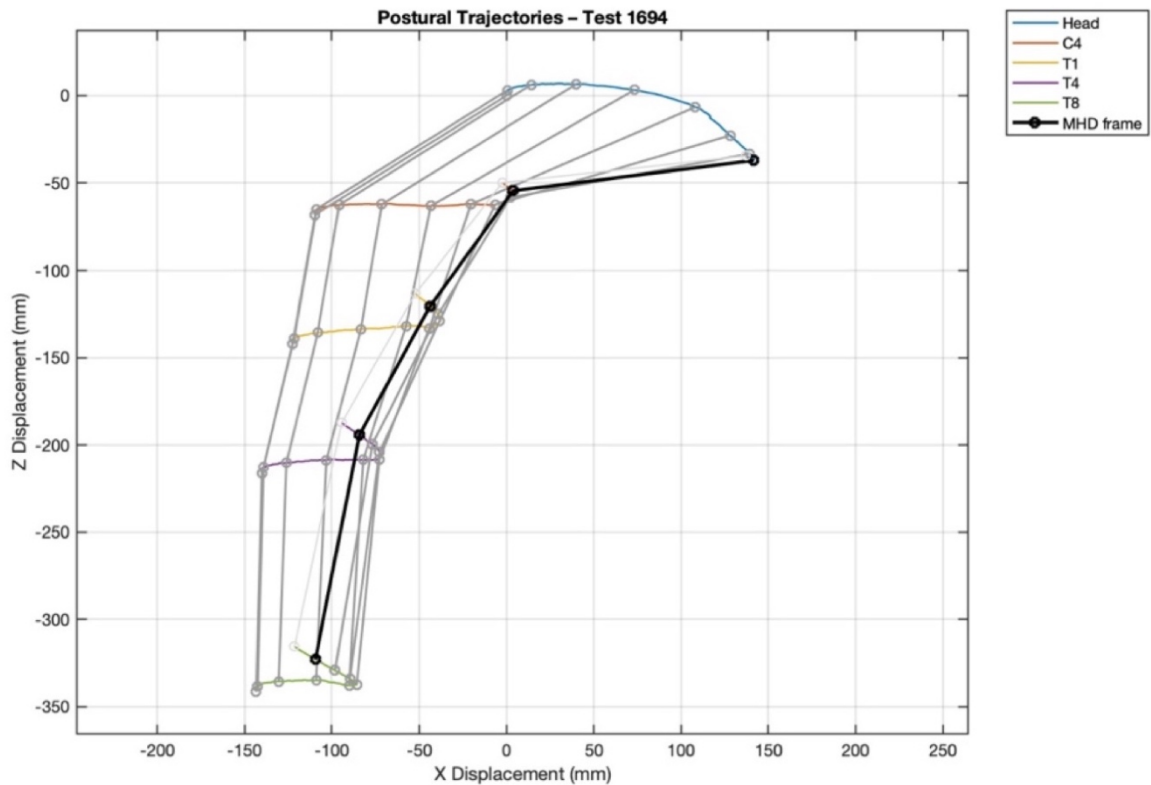


Figure 4.2-1. Landmark trajectories for Subject 1694 with the frame of maximum head displacement (MHD) highlighted in black. Dark gray and light gray lines indicate frames before and after MHD, respectively. Each frame connects anatomical landmarks in the sagittal plane.

At the selected frame, the (x, z) coordinates of the five landmarks were extracted and then re-referenced by subtracting the position of the T8 landmark. This translation removed global body displacement, allowing comparisons of postural configuration across subjects on a common frame of reference. The resulting posture vector comprised 10 values (five landmarks $\times$ two coordinates) and was flattened into a row vector.

Figure 4.2-2. displays the posture configuration of Subject 1694 at MHD after re-referencing. All landmarks are expressed relative to T8, now located at the origin. This spatial arrangement reveals the internal posture of the upper torso at peak excursion, offering a geometric description of spinal alignment independent of the subject's global movement.

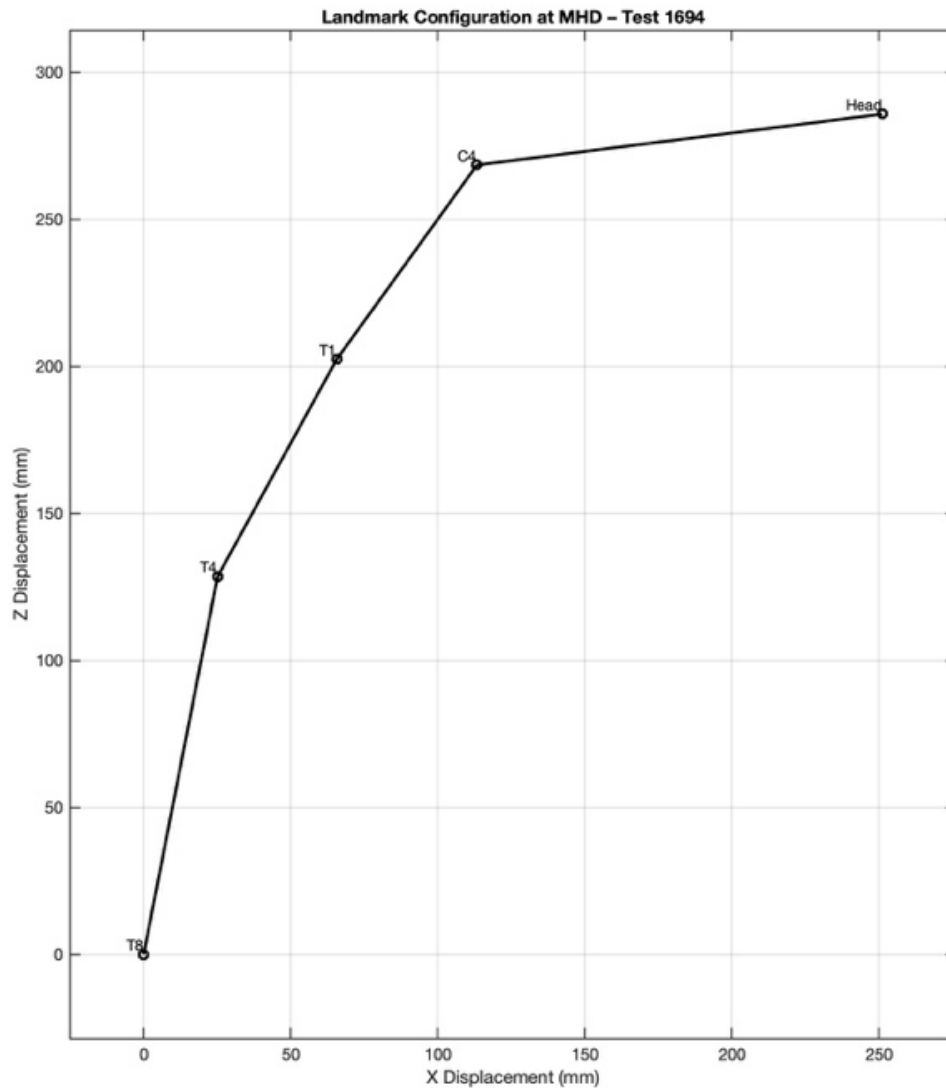


Figure 4.2-2 Posture configuration of Subject 1694 at the frame of maximum head displacement, expressed relative to the T8 landmark. Coordinates represent sagittal-plane displacement, with landmark labels indicating anatomical identity.

This process was repeated for all 14 valid test trials, yielding a 14×10 matrix of standardized posture vectors. These vectors formed the input to the dimensionality subsequent reduction stage.

4.2.2. PCA Outcomes

After posture vector extraction, PCA was applied to reduce dimensionality and identify dominant modes of variation in upper torso configuration at the moment of MHD. Each trial was represented by a 10-dimensional vector composed of (x, z) coordinates for five anatomical landmarks re-referenced to T8. PCA was performed on the resulting 14×10 matrix.

Table 4.2-1. reports the variance explained by the first three principal components, which were retained for subsequent modeling. PC1 accounted for nearly half of the total variance, while the cumulative variance explained by the first three components exceeded 85%, indicating that a compact representation of inter-subject posture differences was achieved. The retained PCA scores for each subject served as dependent variables in the regression analysis that follows.

PC Index	Variance explained (%)	Cumulative variance explained (%)
1	46.5	46.5
2	28.4	74.9
3	13.7	88.6

Table 4.2-1. Variance explained by the first 3 principal components extracted from posture vectors at MHD.

4.2.3. Regression Modeling Results

To evaluate whether the principal components derived from posture vectors at MHD could be systematically explained by subject characteristics, simple linear regressions were performed for each retained component. For each of the three retained principal components (PC1, PC2, and PC3), a separate linear regression was constructed using a single predictor variable. The candidate predictors included subject characteristics such as group (young vs. elderly), head girth, and age. Models were evaluated individually, and the one with the lowest p-value and clearest biomechanical interpretability was retained for each component. This one-variable strategy was adopted because multivariable models showed poor performance in preliminary testing. All regressions were performed using ordinary least squares in GRETL, and residual diagnostics showed no major violations of linear model assumptions.

Table 3.5 summarizes the best-fitting model for each component. The table reports the predictor used, estimated regression coefficient, p-value, and adjusted R^2 .

PC Index	Predictor	Coefficient	p-value	Adjusted R^2
1	Kind	-1.498	0.193	0.065
2	Head girth	-0.313	0.296	0.015
3	Age (years)	0.0146	0.037	0.259

Table 4.2-2. Summary of best single-variable regression models for PC1, PC2, and PC3 derived from posture vectors at MHD. Grey shading indicates $p < 0.05$.

Of the three principal components, only the regression model for PC3 reached statistical significance at the 5% level. In this case, *age* emerged as a significant predictor, with a p -value of 0.037 and an adjusted R^2 of 0.259. The positive coefficient (0.0146) indicates that PC3 scores increase slightly with age. While this model does not explain the majority of the variance, it accounts for a non-trivial portion and suggests that PC3 captures a postural variation that correlates meaningfully with aging. The direction and significance of the effect indicates that PC3 captures a difference in torso posture that increases with age.

In contrast, the models for PC1 and PC2 show weak and statistically inconclusive results. The best regression of PC1, on group membership (young vs. elderly), yielded a non-significant p -value (0.193) and an adjusted R^2 of just 6.5%. Although the negative coefficient (−1.498) suggests that elderly subjects tend to score lower on PC1, the lack of significance combined with the small effect size does not support this as a reliable relationship. Similarly, PC2 was regressed against head girth, producing an even less promising model: the p -value was 0.296, and the adjusted R^2 dropped to just 1.5%. These figures indicate no meaningful explanatory power, and the small sample size further limits the ability to detect subtle effects.

Taken together, these results imply that only one of the three dominant shape modes (PC3) shows a statistically credible association with subject characteristics, and even that model explains only about 26% of the observed variability. The other components appear to reflect postural variations that are either not well captured by the recorded anthropometric variables or are driven by more complex, multivariate interactions not modeled here.

4.2.4. Posture Metrics: Inter-Landmark Distances and Joint Angles

To further assess group-level differences in torso configuration at the moment of maximum head displacement (MHD), two complementary analyses were conducted: (1) an evaluation of distances between consecutive anatomical landmarks, and (2) an analysis of internal joint angles formed by triplets of adjacent landmarks. Both approaches aimed to capture biomechanically meaningful distinctions between young and elderly postures under frontal loading.

Before conducting these analyses, Figure 4.2-3. provides a qualitative overview of subject postures at the MHD frame. Each configuration has been re-referenced to the T8 landmark, removing translational effects and enabling direct comparison of internal alignment. The plotted segments connect the Head, C4, T1, T4, and T8 landmarks for each subject, with red lines representing young volunteers and blue lines, elderly. The graph seems to reveal that elderly subjects tend to show more upright, extended upper torso postures, while younger subjects display a broader range of configurations, including more forward-flexed profiles. This pattern is especially noticeable in the Head and C4 segments, suggesting age-related variation in cervical posture under frontal loading. However, no quantitative results can be obtained from just observation here.

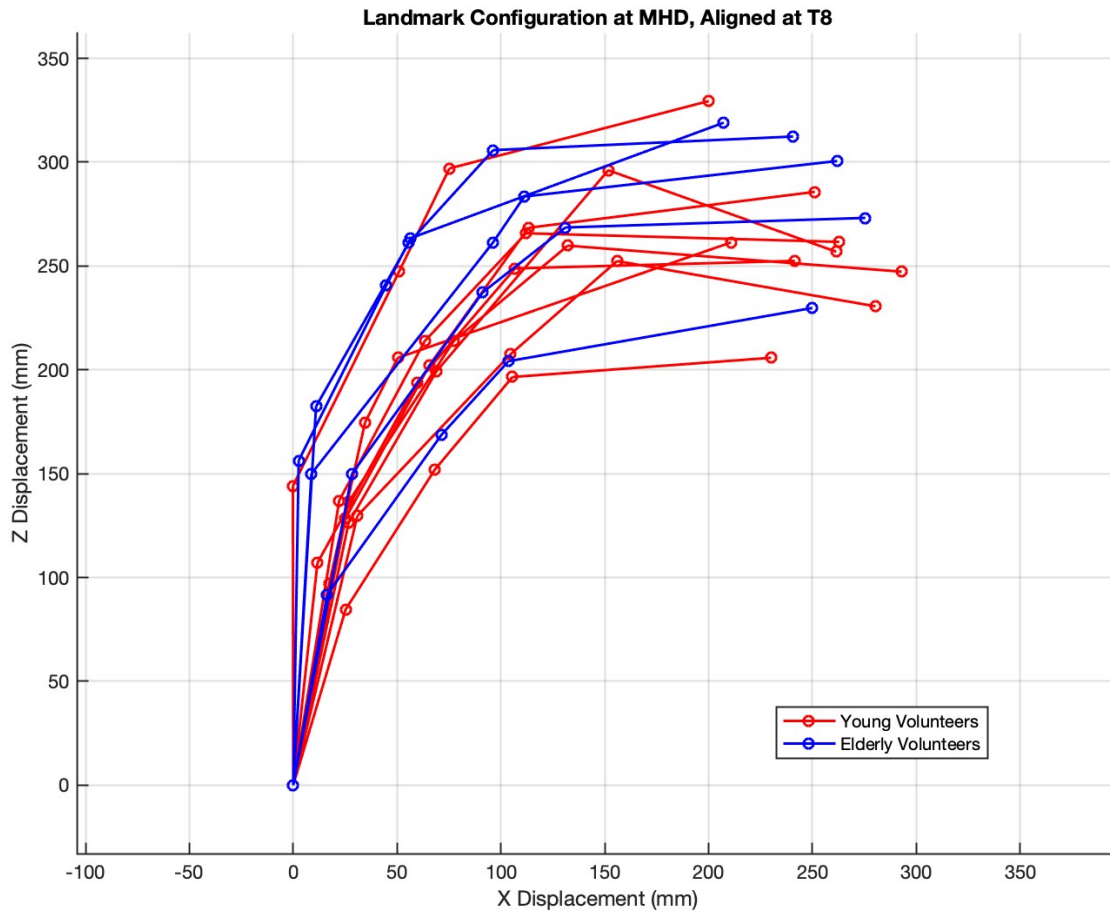


Figure 4.2-3. Posture of all subjects at maximum head displacement, re-referenced to T8, shown as a sequence of line segments connecting Head to T8. Red: young group; blue: elderly group.

Inter-Landmark Distance Analysis

The first analysis calculates the Euclidean distances between adjacent landmarks: Head–C4, C4–T1, T1–T4, and T4–T8. The group means and statistical comparisons are shown in Table 4.2-3.

Segment	Young ($n = 9$)	Elderly ($n = 5$)	p -value
Head–C4	139.17 ± 17.90 mm	149.98 ± 6.72 mm	0.224
C4–T1	71.16 ± 25.10 mm	42.22 ± 15.42 mm	0.039
T1–T4	92.17 ± 12.63 mm	105.81 ± 21.07 mm	0.151
T4–T8	123.31 ± 19.97 mm	147.00 ± 32.81 mm	0.115

Table 4.2-3. Inter-landmark distances (mm) at MHD. Values represent mean $\pm$ standard deviation per group. P-values from two-sample t-tests included to assess group differences. Grey shading indicates $p < 0.05$.

The only segment showing a statistically significant group difference was C4–T1, which was markedly shorter in elderly subjects ($p = 0.039$). However, it is important to note that this analysis involved four independent comparisons, each using the same dataset. To control for the increased risk of Type I error (false positive) in this context, a Bonferroni correction would adjust the significance threshold from $\alpha = 0.05$ to $\alpha = 0.0125$. Under this stricter criterion, none of the inter-landmark distance differences would reach statistical significance, including C4–T1. Given the small sample size and correlated nature of these measurements, such corrections may be overly conservative, but they underscore the need to interpret the distance-based findings with caution. The high variance observed across most segments further complicates interpretation, limiting the strength of conclusions drawn from mean comparisons alone.

Joint Angle Analysis

To obtain a more anatomically grounded view of postural variation, joint angles were calculated at three internal nodes of the torso using the Head–C4–T1, C4–T1–T4, and T1–T4–T8 triplets. These angles represent flexion or extension at key cervical and thoracic junctions and are invariant to global translation or segment length. Group averages and statistical results are summarized in Table 2.2-1.

Joint	Angle Definition	Young (n=9)	Elderly (n=5)	p-value
C4	Head–C4–T1	132.81 $\pm$ 2.36°	137.61 $\pm$ 4.94°	0.055
T1	C4–T1–T4	169.81 $\pm$ 5.59°	173.18 $\pm$ 6.56°	0.599
T4	T1–T4–T8	160.15 $\pm$ 8.33°	152.99 $\pm$ 4.11°	0.027

Table 4.2-4. Internal joint angles (°) at MHD. Values represent mean $\pm$ standard deviation. P-values are based on Welch's t-test (no assumption of equal variance). Grey shading indicates $p < 0.05$.

Of the three angles analyzed, the T4 angle (T1–T4–T8) exhibited a statistically significant difference ($p = 0.027$), with elderly subjects showing a more flexed posture (smaller angle) in the lower thoracic region. The C4 angle also showed a trend toward significance ($p = 0.055$), suggesting a more extended cervical posture in the elderly group, though this result falls just above the conventional $\alpha = 0.05$ threshold. The T1 angle showed no group difference.

Compared to the distance-based metrics, these angle results provide clearer and more interpretable biomechanical insight, reflecting postural curvature rather than absolute

displacement. However, statistical power remains limited by the small sample size ($n = 14$) and unbalanced group sizes.

4.3. Method 3: Arc-Length Re-Parameterization and Signal Registration for generation of average trajectories and deviation corridors

4.3.1. Group-Specific ARCGen Outputs

This subsection presents the outputs of Method 3, which characterizes landmark trajectories using arc-length-based averaging and corridor generation. The analysis focuses on the geometric dispersion of torso movement patterns and how these vary by age group.

The ARCGen software, as released, was originally configured to generate a single characteristic average trajectory and associated statistical corridor from a pooled dataset. For the purposes of this study, the ARCGen execution scripts were adapted to operate separately on the young and elderly subgroups, allowing for age-specific analysis of postural variation. This adjustment preserved the method's core features (arc-length re-parameterization, signal registration, and two-dimensional statistical corridor construction) while enabling groupwise geometric comparisons.

Only the results from the age-separated runs are presented below, as they provide clearer insights into group-level differences. For each landmark, the figures display the average trajectory and 95% confidence corridor for both age groups, plotted along normalized arc-length. In the background, in lighter hues, the individual trajectories can also be observed.

4.3.2. Landmark 1: Head

Figure 4.3-1. shows the ARCGen outputs for the Head landmark, comparing the characteristic average trajectories and statistical corridors of the young and elderly groups. Both curves follow a similar overall shape: an initial upright position followed by forward displacement and a gradual return.

The average trajectory of the elderly group shows a smaller horizontal excursion than that of the young group, particularly near the midpoint of the arc-length path. In contrast, the young group exhibits a more pronounced forward curvature, indicating greater forward displacement during peak motion.

Corridor width and average position differs most noticeably around the region of maximum excursion. The elderly group displays a visibly broader corridor in this mid-trajectory segment, indicating higher inter-subject variability in forward displacement.

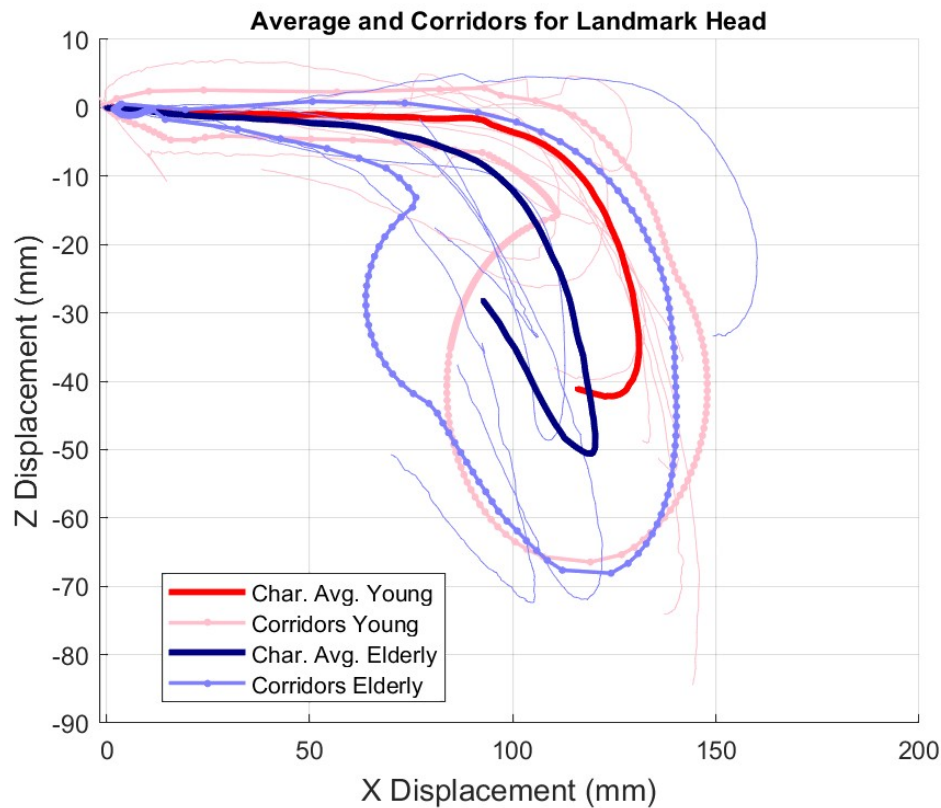


Figure 4.3-1. ARCGen outputs for the Head landmark. Average trajectories and 95% statistical corridors are shown for young (red) and elderly (blue) groups. Greater variability is observed in the elderly group during the central portion of the arc-length path.

4.3.3. Landmark 2: C4

Figure 4.3-2 presents the ARCGen outputs for the C4 landmark. Both groups follow a trajectory consistent with anterior flexion and subsequent partial recovery. The young group's average curve exhibits greater forward progression, while the elderly group's path turns upright after maximum displacement.

Corridor width remains moderate for both groups at the initial and final segments but diverges in the central region. As observed with the Head, the elderly group displays a broader confidence corridor around the midpoint of the trajectory. This indicates higher variability in the magnitude and path of forward displacement during that segment. The young group shows a more compact envelope, suggesting greater consistency in motion patterns.

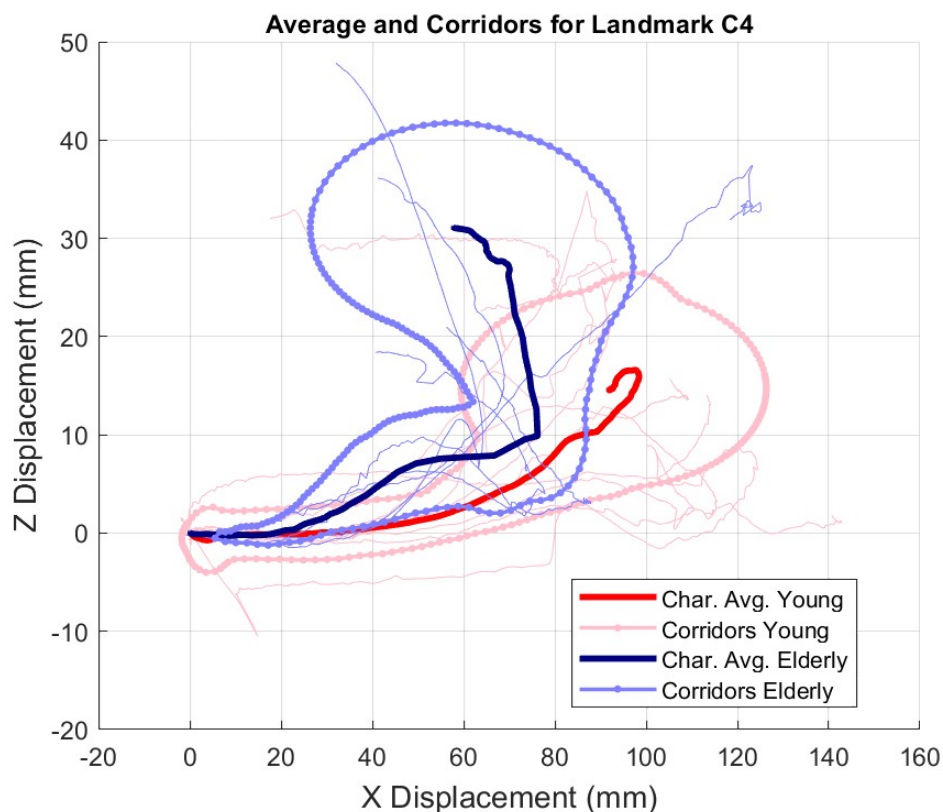


Figure 4.3-2. ARCGen outputs for the C4 landmark. Average trajectories and 95% statistical corridors are shown for young (red) and elderly (blue) groups. The elderly group displays increased dispersion during the central arc-length segment.

4.3.4. Landmark 3: T1

Figure 4.3-3. shows the ARCGen-derived average trajectories and confidence corridors for the T1 landmark. It is noticeable that the young group reaches farther in the anterior direction, with the average trajectory extending, well beyond that of the elderly group.

This difference in forward excursion is most visible around the midpoint of the arc-length path, where the young trajectory continues progressing along the x -axis, while the elderly trajectory begins to decelerate and curve upward. The average trajectory of the young group extends farther in the anterior direction, indicating a longer forward excursion along the x -axis. The elderly group follows a more compact trajectory, with reduced horizontal reach.

In addition, variability in the elderly group is more pronounced throughout the entire trajectory. The corridor remains visibly wider across nearly all arc-length segments, not just near peak displacement. In contrast, the young group maintains a more compact envelope, especially at the start of the path.

This consistent widening of the corridor for the elderly group suggests increased dispersion in both the magnitude and timing of forward torso movement at the T1 level.

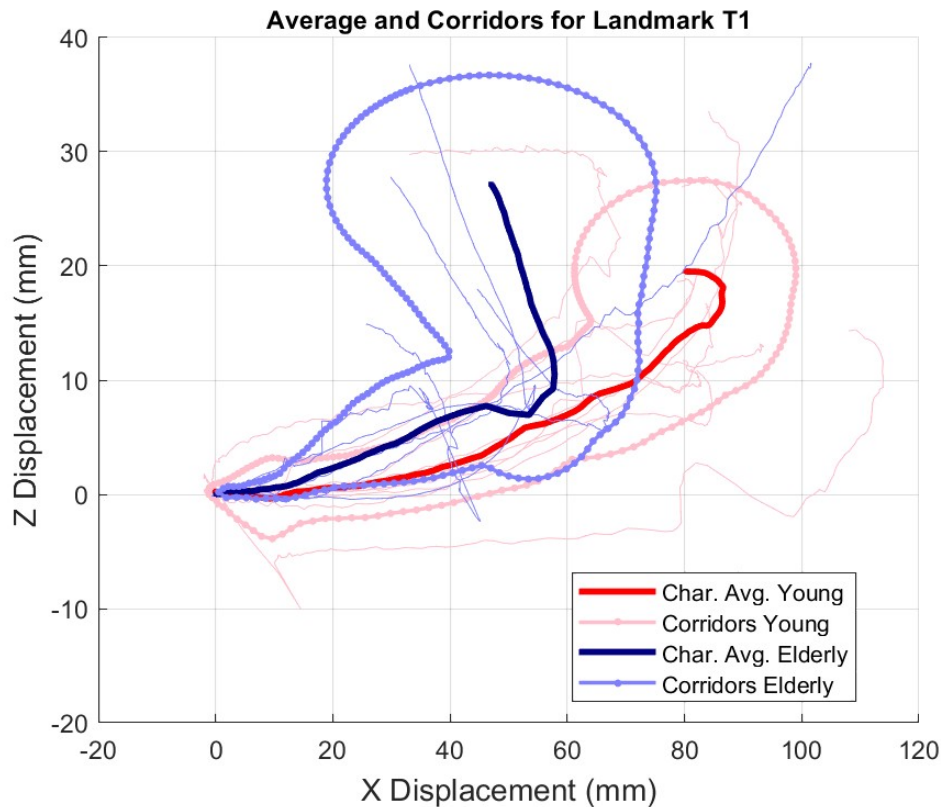


Figure 4.3-3. ARCGen outputs for the T1 landmark. The elderly group (blue) shows increased dispersion across the entire arc-length range compared to the young group (red), whose trajectories remain more tightly clustered.

4.3.5. Landmark 4: T4

Figure 4.3-4. displays the ARCGen outputs for the T4 landmark. The average trajectory of the young group extends farther forward compared to the elderly group, resulting in a visibly longer and more curved displacement path in the xz -plane. The elderly group follows a more compact trajectory, with reduced forward reach but a still pronounced vertical component.

Corridor width is markedly greater for the elderly group throughout most of the arc-length path. This indicates increased inter-subject variability in the elderly group, both in the extent and direction of motion. The young group maintains a narrower envelope, with dispersion concentrated primarily around the middle arc-length segment.

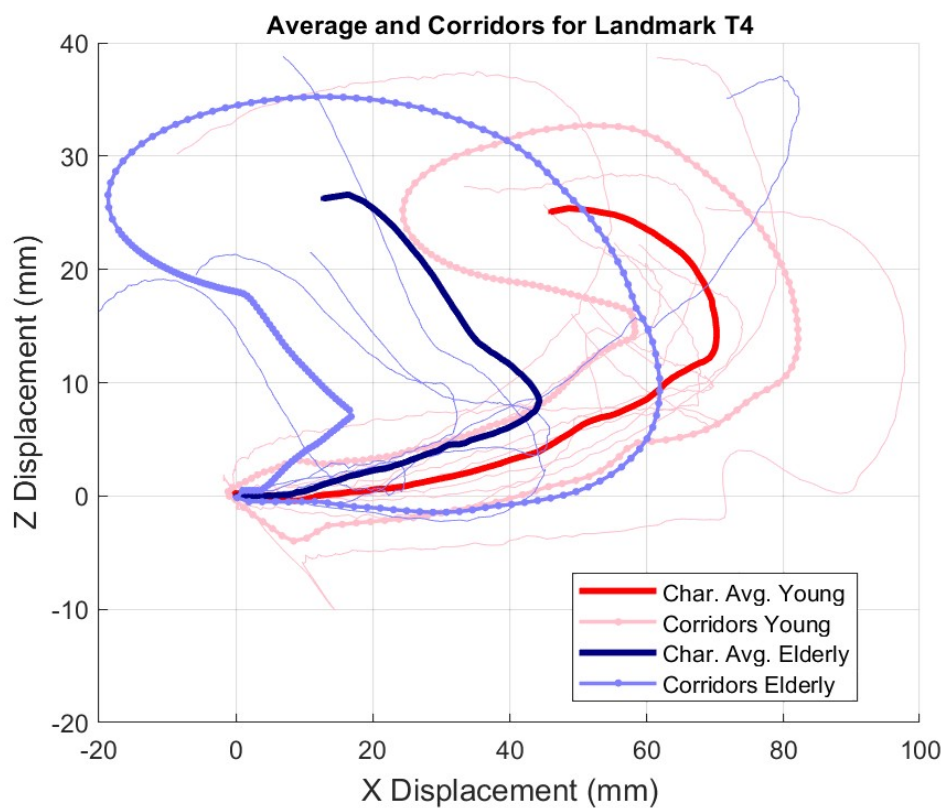


Figure 4.3-4. ARCGen outputs for the T4 landmark. The elderly group (blue) exhibits lower forward and vertical excursion but a broader statistical corridor compared to the young group (red).

4.3.6. Landmark 5: T8

Figure 4.3-5. displays the ARCGen outputs for the T8 landmark. The average trajectories for young and elderly subjects follow distinct paths. The young group shows a forward-arching trajectory, extending prominently along the x-axis. In contrast, the elderly group displays a more posteriorly displaced curve.

The difference in forward excursion is substantial: the young group's trajectory extends nearly to $x \approx 80$ mm, while the elderly group's curve reaches only around $x \approx 40$ mm, then loops upward and backward. This reflects a shorter anterior displacement and a steeper vertical component in the elderly group's motion pattern.

Corridor width also differs between groups. The elderly group shows greater dispersion throughout the trajectory, with a visibly broader confidence envelope, especially in the upper and final arc-length segments. The young group maintains a more compact envelope overall, with moderate widening near peak forward displacement.

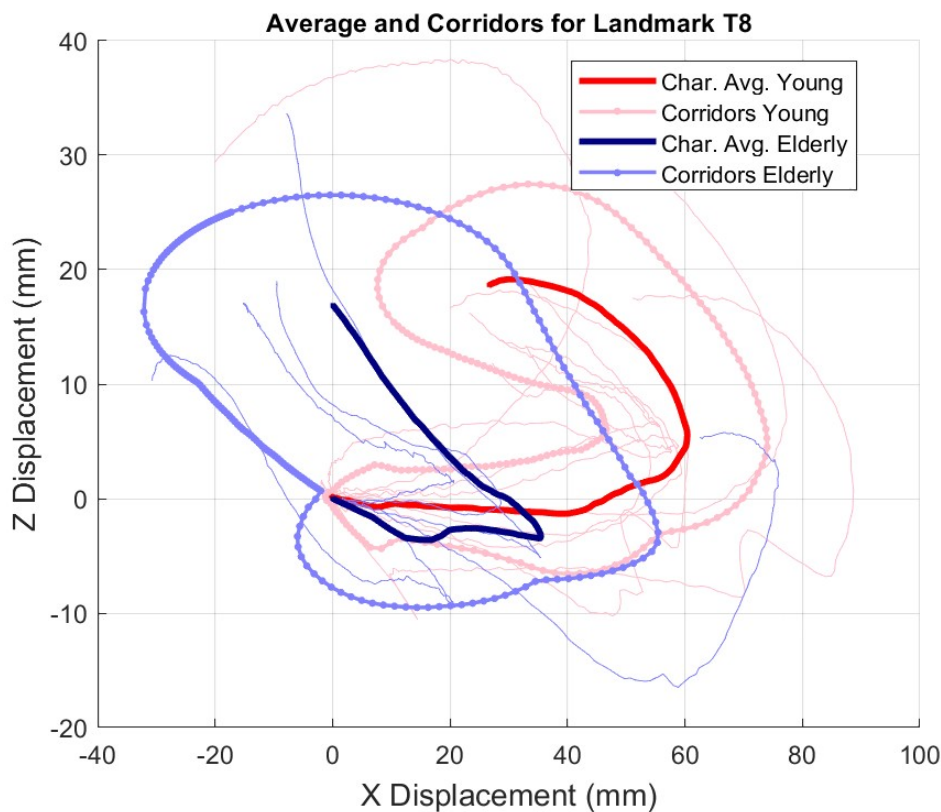


Figure 4.3-5. ARCGen outputs for the T8 landmark. The elderly group (blue) shows a shorter forward excursion, accompanied by increased variability compared to the young group (red).

4.3.7. Landmark 6: H Point

Figure 4.3-6. displays the ARCGen outputs for the H-Point landmark. Compared to previous landmarks, the individual trajectories show substantially more scatter, especially within the young group. Displacement patterns vary widely in direction and amplitude, with several trajectories deviating markedly from the group's average.

Despite this variability, the average trajectories remain reasonably well-formed for both age groups. The young group's mean path extends farther forward, while the elderly group shows a more upward-oriented displacement. However, this distinction is significantly less consistent than at all other landmarks.

The statistical corridors are broad and irregular. The envelopes span a wide region in both horizontal and vertical directions, reflecting high inter-subject variability. This pattern suggests that motion at the H-Point is less constrained or less consistently expressed across individuals.

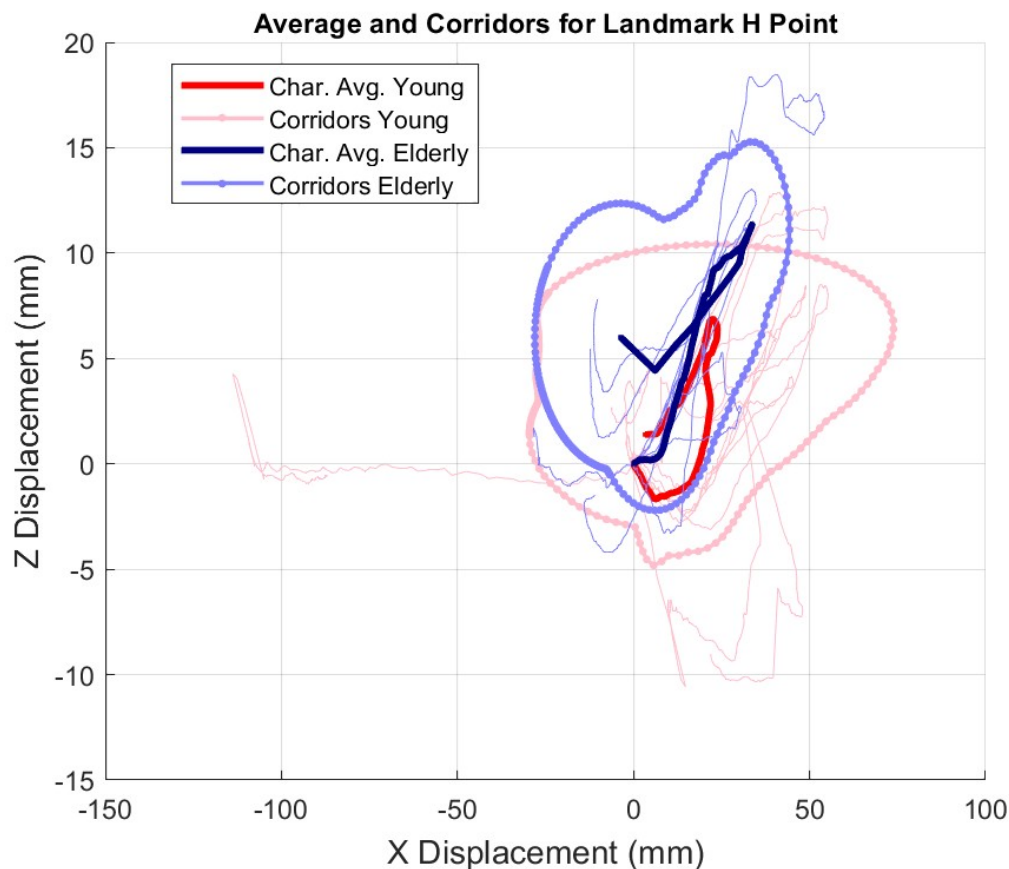


Figure 4.3-6. ARCGen outputs for the H-Point landmark. Individual trajectories are highly variable, especially in the young group. The elderly group shows a more vertically oriented average path and broader dispersion overall.

4.3.8. Summary of observed patterns

Across all landmarks, the ARCGen outputs reveal consistent geometric differences between age groups, particularly in the extent of forward excursion and the magnitude of inter-subject variability.

The young group systematically displays longer anterior displacement across all landmarks, as evidenced by average trajectories extending farther along the x -axis. In contrast, elderly trajectories are typically more compact, with reduced forward reach and a tendency toward greater upward (downward for the Head) curvature, especially at C4 and T8.

Statistical corridors are generally broader in the elderly group, indicating higher dispersion in posture, specially throughout the latter half of the trajectory. The H-Point stands out as qualitatively different: both groups exhibit irregular and noisy trajectories, with less coherent patterns and broader, more amorphous confidence envelopes. This suggests lower consistency of motion at this anatomical reference point compared to the others.

Overall, the method successfully captures some age-related differences in both displacement magnitude and inter-individual variability along the arc-length-normalized movement paths. However, the relatively small sample size, particularly after subgroup separation, limits the generalizability of these patterns and may amplify the influence of outliers on both average trajectories and corridor width.

5. DISCUSSION

5.1. Method 1: B-Spline approximation and Principal Component Analysis of individual landmark trajectories

5.1.1. Key Findings and interpretation

The results obtained through spline-based dimensionality reduction and landmark-specific PCA revealed several interpretable trends in posture variation during low-speed frontal decelerations. Although primary shape modes (PC1) were not significantly associated with subject characteristics, secondary modes (PC2) consistently captured inter-group differences, particularly between young and elderly volunteers.

For thoracic landmarks (notably T1 and T4), PC2 scores were significantly higher in elderly subjects. This component typically reflected asymmetry or trajectory curvature beyond the primary forward excursion encoded by PC1. The consistent positive coefficients associated with the elderly group suggest that these volunteers exhibited distinct postural behaviors, possibly involving increased forward curvature or altered torso control patterns in response to deceleration. These differences may reflect age-related changes in spinal flexibility, muscle tone, or neuromotor control.

The Head and T8 landmarks also showed significant or borderline associations, though with lower explained variance. In the case of the Head, PC2 scores correlated with group membership ($p = 0.0496$), potentially capturing variations in upper spinal compensation or head control strategies. At the H-point and C4, no strong statistical effects were detected, suggesting that pelvic anchoring and lower cervical posture may be less influenced by aging under these loading conditions.

Overall, Method 1 demonstrated that even low-dimensional summaries of movement patterns, derived from compressed control-point representations, can reveal biomechanically relevant trends. While PC1 captured the majority of motion amplitude, PC2 emerged as a more sensitive discriminator of posture shape differences tied to subject characteristics, especially age group.

5.1.2. Limitations and Potential Improvements

While Method 1 successfully captured some dominant shape patterns and revealed age-related differences in postural behavior, several limitations constrain the generalizability and interpretive depth of the findings.

A primary limitation concerns sample size. With only fourteen trials available, and an uneven distribution between young and elderly subjects, statistical power was inherently limited. This is especially relevant for regression modeling, where the low number of degrees of freedom limited the robustness of model fits and discarded multivariate analysis. Several models involving PC1 scores (e.g., with seated height as predictor) showed modest trends that did not reach statistical significance. Given their consistent direction and borderline p-values, it is plausible that these effects reflect genuine biomechanical variation that the current sample was underpowered to detect.

Another limitation relates to the dimensionality reduction process. The use of B-splines with eight control points per trajectory provided a compact and interpretable representation of movement, and the RMSE analysis confirmed that this configuration preserved most key features. However, the fixed number of control points, while methodologically consistent and supported by literature, may have introduced constraints in capturing localized features in highly dynamic trajectories. Although the approximation proved robust across landmarks, future studies could explore adaptive schemes that tailor the number of control points to the complexity of each trajectory, potentially enhancing the resolution of subtle shape differences.

The use of landmark-specific PCA introduced interpretability benefits but also methodological fragmentation: since each PCA was performed independently, cross-landmark correlations in motion (e.g., coordinated movement of T1 and T4) were not captured. A joint PCA or multivariate approach could potentially reveal coupled behaviors that single-landmark decompositions may have overlooked.

In the regression stage, only univariable linear models were considered, due to the small sample size. While this choice was methodologically necessary, it limited the capacity to explore interaction effects or combined predictors (e.g., age group $\times$ seated height). Additionally, some regressions (e.g., T8 PC1) returned negative adjusted R^2 values, highlighting both poor predictive capacity and the sensitivity of PCA scores to small numerical variation.

Future improvements could involve increasing the dataset size, applying adaptive or hierarchical spline fitting, using joint-PCA frameworks to capture inter-landmark coordination, and exploring non-linear or regularized regression models to enhance prediction stability. These steps would strengthen both the statistical validity and biomechanical interpretability of the analysis.

5.2. Method 2: Landmark relative positioning analysis at peak displacement

5.2.1. Key Findings and Interpretation

The second method focused on static posture configuration at the moment of maximum head displacement (MHD), using a fixed-frame analysis of relative landmark positions. This approach provided a complementary perspective to the trajectory-based spline method by isolating geometric differences in body posture independent of temporal progression or amplitude.

The principal component analysis performed on re-referenced landmark coordinates revealed that the dominant modes of variation (PC1 and PC2) captured global geometric changes in torso configuration, but did not significantly correlate with subject characteristics. This suggests that while these components effectively describe posture diversity within the sample, their variability does not align strongly with age or anthropometric factors.

The third principal component (PC3), by contrast, showed a statistically significant association with age group, emerging as the sole interpretable link between posture at MHD and subject-level variation. Although PC3 accounted for a smaller proportion of total variance, its effect was consistent and meaningful: elderly subjects displayed systematically different scores, suggesting the presence of age-related postural adaptations that are not captured by the primary shape axes. Given the standardized PCA framework and the direction of the effect, it is reasonable to interpret PC3 as encoding a localized morphological shift, possibly linked to shifts in spinal alignment between cervical and thoracic regions.

These observations were complemented by a distance-based analysis of adjacent landmarks. While most segment-wise comparisons did not reach statistical significance, a notable exception was the C4–T1 segment, which appeared shorter in elderly subjects. However, this difference did not survive correction for multiple comparisons and was accompanied by high variability, particularly within the younger group. The absence of consistent trends across other segments, and the limited statistical robustness of the findings, suggest that inter-landmark distances may be less reliable indicators of age-related postural differences in this context. Their sensitivity to initial alignment, anthropometric variability, and small-sample noise likely contributed to the weak and inconsistent group effects observed.

The joint angle analysis was performed right after, and quantified internal angles formed by spinal landmark triplets. Elderly subjects exhibited a statistically significant reduction in the T4 angle (T1–T4–T8), indicative of increased flexion in the lower thoracic spine. A borderline significant difference at the C4 angle (Head–C4–T1) also suggested a trend toward more extended cervical posture in elderly individuals.

Altogether, Method 2 demonstrated that postural variation at a single key time frame can reflect meaningful biomechanical differences between age groups. The combination of PCA-derived shape descriptors and joint angle measurements identified age-related differences in both thoracic curvature and cervical alignment. While the primary PCA components were not strongly associated with subject variables, the consistent group

effect observed in PC3 and the significant reduction in the T4 angle among elderly subjects support the interpretation that aging influences torso configuration under deceleration. These findings complement the dynamic trajectory analysis presented in Method 1 by highlighting static postural signatures that emerge at the moment of maximum excursion.

5.2.2. Limitations and Potential Improvements

Although Method 2 provided a useful alternative to trajectory-based analyses by focusing on posture at a single critical time frame, several limitations constrain the interpretive strength and generalizability of its findings.

A core limitation lies in the sample size and statistical power, already noted in the Discussion of Method 1, Limitations and Potential Improvements. With only 14 trials and an uneven distribution between young and elderly groups, the detection of subtle posture differences is inherently limited. However, this issue is arguably more pronounced in Method 2, where posture is assessed using a single frame per subject. This static snapshot increases susceptibility to trial-specific noise, measurement artifacts, or individual variability that might otherwise be averaged out across time. Furthermore, the exclusion of the H-point due to poor signal quality reduced anatomical coverage and may have masked relevant differences in pelvic or lumbar posture.

In terms of data reduction, the principal component analysis applied to posture vectors condensed inter-subject variation into a low-dimensional form. However, the lack of strong associations between PC1 or PC2 and subject characteristics raises concerns about their interpretive value. These components captured global shape variation but did not align meaningfully with age or anthropometry.

This highlights a limitation in the explanatory strength of single-variable models in this context. While the PCA efficiently condensed posture variation into a compact set of scores, the available regressors (age, subject group, and anthropometrics) were not capable of fully explaining that variation. The significant result for PC3 stands as the only interpretable link between posture at MHD and subject-level differences, specifically aging.

The inter-landmark distance analysis was similarly limited. Only one segment (C4–T1) showed a statistically significant difference between groups, and that result could not survive correction for multiple comparisons. In contrast, the joint angle analysis produced more anatomically grounded results, identifying a significant group effect at T4 and a trend at C4. However, the angles used were still coarse descriptors of complex spinal geometry, and their interpretation is constrained by the small sample and lack of multivariable modeling.

Overall, Method 2 offered useful insights into static posture differences between groups, but its explanatory power was limited by both the structure of the dataset and the nature of the variables considered. Future improvements could involve expanding the sample, incorporating richer geometric descriptors (e.g., curvature or 3D features), and applying multivariate or regularized modeling approaches to better characterize the interplay between posture and individual factors. Combining static and dynamic analyses may also

help to distinguish whether observed differences at MHD reflect meaningful biomechanical adaptations or transient positions shaped by external constraints.

5.3. Method 3: Arc-Length Re-Parameterization and Signal Registration for generation of average trajectories and deviation corridors

5.3.1. Key Findings and Interpretation

The outputs obtained through arc-length re-parameterization and statistical corridor generation revealed several age-related trends in postural behavior during frontal deceleration. Across nearly all landmarks, the average trajectories of the elderly group exhibited reduced forward excursion compared to the young group. This was most pronounced at T1, T4, and T8, where the young subjects showed a clear anterior displacement, whereas elderly trajectories curved upward sooner and remained more compact.

In addition to trajectory shape differences, corridor width analysis consistently indicated greater variability in the elderly group. This increased dispersion was most apparent during the mid-to-late segments of the arc-length path, coinciding with peak excursion and recovery phases. At T4 and T8, the elderly group's variability extended throughout the latter half of the movement, suggesting greater inconsistency in how forward motion was resolved.

C4 and T1 also revealed differences in vertical displacement. The elderly group often followed a more upward-oriented path, possibly reflecting different spine or head positioning under deceleration. In contrast, the young group maintained flatter or more forward-directed trajectories.

The H-Point landmark presented a notable exception. Individual trajectories were highly scattered, especially among young subjects, and the resulting confidence corridors were broad and irregular. While average trajectories could still be computed, the high variability and lack of directional consistency suggest that motion at this anatomical reference point may be less biomechanically constrained or more affected by external factors, such as seat interaction or measurement noise.

Overall, the method effectively exposed geometric differences between age groups, capturing both the average displacement patterns and the range of variation in a manner that complements the findings of prior methods.

5.3.2. Limitations and Potential Improvements

While Method 3 provides a valuable geometric characterization of postural behavior, several limitations restrict the depth and generalizability of its findings.

Foremost among these is, again, the limited sample size. Once split by age group, the dataset contained only a small number of trials per condition, reducing statistical robustness. This constraint may have inflated the influence of outliers on both average trajectory shape and corridor width. The pronounced variability observed at certain landmarks (particularly the H-Point) may partially reflect this limited sampling, and broader conclusions should be drawn with caution.

Another limitation concerns the assumption of geometric comparability inherent to arc-length-based methods. ARCGen assumes that the underlying signal shapes are

sufficiently similar to justify normalization and registration in a common space. When this assumption is violated, as at the H-Point, the method may produce misleadingly broad or amorphous corridors that obscure true biomechanical structure. Further filtering or trajectory quality checks may be necessary before applying the method at loosely constrained anatomical regions.

The method also discards temporal dynamics in favor of shape-based alignment. While this is appropriate for the current study's emphasis on geometry, it prevents analysis of timing-related effects such as response delays or speed of motion, factors that may also vary with age.

Finally, all ARCGen parameters, including the number of warping control points and the warping penalty, were held constant across landmarks. While this maintained consistency, it may not represent the optimal configuration for each anatomical region. Future studies could explore adaptive tuning strategies to balance flexibility and stability in the warping process.

Increasing the dataset size and incorporating anthropometric normalization could help improve result reliability. Additionally, pairing ARCGen outputs with timing- or force-based data may offer a more comprehensive view of age-related postural control strategies during low-speed impacts.

5.4. Method comparison

The three analytical methods developed and applied in this thesis

1. B-spline-based trajectory compression with PCA,
2. Time-based posture extraction with PCA and metric analysis, and
3. Arc-length-based trajectory averaging with statistical corridor generation,

offer complementary perspectives on kinematic variability in frontal deceleration tests. Each method emphasizes different aspects of the motion and encodes the data using distinct mathematical frameworks, which results in varying sensitivity to inter-subject differences and age-related effects.

5.4.1. Dimensional focus and granularity

Method 1 analyzes entire motion trajectories by encoding their temporal evolution through B-spline control points. This approach preserves curve shape while reducing dimensionality, enabling fine-grained analysis of trajectory shape through PCA. In contrast, Method 2 collapses the motion to a single instant, the frame of maximum head displacement (MHD), and focuses on spatial configuration rather than temporal progression. Method 3 retains full trajectory geometry but discards temporal alignment, instead using arc-length as the basis for comparison. Consequently, each method operates at a different resolution of temporal information: Method 1 emphasizes compressed time-series trends, Method 2 captures instantaneous posture, and Method 3 abstracts time altogether in favor of geometric similarity.

5.4.2. Sensitivity to inter-subject variation

In terms of capturing inter-subject differences, Method 1 demonstrated that secondary principal components (especially PC2) were more responsive to age-related variation than primary motion trends (PC1), which tended to reflect global amplitude. T1 and T4 landmarks in particular showed significant group effects, suggesting that mid-thoracic posture during deceleration varies systematically with age.

Method 2, despite its lower-dimensional nature, identified a statistically significant relationship between PC3 and age, and additionally detected age-related differences in spinal flexion via inter-landmark distance and joint angle analyses. These findings suggest that while posture at MHD may not strongly reflect gross anthropometric variation, it still captures specific localized differences in spinal alignment that correlate with aging.

Method 3 revealed the most consistent group-level geometric differences, particularly in the extent of forward displacement and the breadth of inter-subject variability. Elderly participants exhibited shorter anterior excursions and broader confidence corridors across nearly all landmarks. These effects were spatially consistent and visually interpretable, confirming that Method 3 is especially effective for identifying population-level trends and variability in motion geometry.

5.4.3. Interpretability and Biomechanical Meaning

Method 1 excels in identifying low-dimensional modes of variation and associating them with anatomical motion trends, but its outputs (PCA scores) require interpretation through statistical modeling. Method 2 offers direct anatomical interpretability through posture vectors and angles but is limited by its focus on a single time frame and low explanatory power of primary components. Method 3 provides intuitive, spatially explicit visualizations of average motion and inter-subject variability, making it particularly suitable for communicating its findings.

5.4.4. Robustness and Limitations

All three methods are constrained by the small sample size, limiting their power to detect subtle or multivariate effects.

Method 1, in particular, depends on the fidelity of spline fitting and consistency in control point configuration; while cubic B-splines with fixed parameters ensure comparability, they may limit sensitivity to localized shape differences or abrupt curvature changes. Additionally, conducting PCA independently per landmark prevents the detection of inter-regional coordination, constraining biomechanical interpretation to isolated regions.

Method 2 is inherently limited by its reliance on a single frame, which captures spatial configuration but discards temporal information and dynamic features such as velocity or timing of inflection. Its explanatory power is further reduced by the dimensional simplicity of the posture vectors and the exclusion of the H-point, which limits anatomical coverage.

Method 3 enhances robustness by eliminating temporal alignment requirements through arc-length normalization, but this approach assumes that trajectory shapes are sufficiently consistent for registration to be meaningful, an assumption that fails in regions with irregular or poorly constrained motion, such as the H-point. Moreover, the method does not quantify relationships between trajectory shape and subject characteristics, restricting its analytical scope to descriptive comparisons.

5.4.5. Summary of Strengths and Limitations

Table 5.4 provides a structured comparison of the three analytical methods in terms of their temporal scope, usability, and methodological strengths and limitations.

Criterion	Method 1: B-Spline PCA	Method 2: MHD Posture	Method 3: Arc-Length Averaging
Temporal Coverage	Full landmark trajectories encoded via B-spline control points; captures the entire motion profile over time but in compressed form	Single time frame selected at maximum head displacement; all dynamic information outside this frame is discarded	Full trajectory preserved in spatial terms; time axis removed and replaced with normalized arc length to enable shape-based alignment
Key Strength	Enables landmark-specific statistical modelling of trajectory shape using PCA and regression; identifies distinct modes of motion variability tied to anthropometric variables	Provides direct anatomical interpretation of posture at a critical moment; posture vectors, joint angles, and inter-landmark distances offer biomechanically meaningful comparisons	Generates average trajectories and dispersion corridors that capture both central tendency and variability of motion patterns; highly visual and robust to inter-subject timing differences
Best For	Detecting motion trends that correlate with subject characteristics such as seated height or group; useful for identifying subtle shape differences across the entire trajectory	Characterizing postural alignment differences between young and elderly subjects at the moment of peak forward excursion; complements trajectory-based analyses with spatial metrics	Comparing global motion tendencies and variability across groups without assumptions about time synchronization; ideal for illustrating age-related dispersion or consistent geometric trends
Main Limitation	Regression models have limited explanatory power due to small sample size; interpretation of	Single-frame focus misses dynamic changes and relies on accurate frame selection; PCA results showed weak	Group comparisons are qualitative; results can be distorted in landmarks with high noise or

	abstract PCs may be ambiguous	association with subject traits beyond PC3; constrained statistical power	inconsistent motion patterns (e.g., H-point)
--	-------------------------------	---	--

Table 5.4-1. Comparative summary table of the three analytical methods

Together, these methods provide a multi-angle analysis of volunteer motion under frontal deceleration. Their combination enables robust cross-validation of findings and supports a more detailed understanding of age-related variation in occupant kinematics than any single method could achieve on its own.

It is worth noting that a related analysis of this same experimental dataset was presented by López-Valdés et al. at an international conference of the Association for the Advancement of Automotive Medicine. Their work focused on describing the overall kinematic and dynamic responses of young and elderly volunteers, including comparisons of head and thoracic displacements, belt forces, and angular velocities. While the present thesis adopts a different perspective centered on dimensionality reduction and shape-based standardization techniques, several of the observed trends, such as reduced forward excursion and increased variability in the elderly group, are consistent with the findings reported in their presentation.

5.5. Real-World Applications

Although the research conducted in this thesis is fundamentally exploratory and situated at a low Technology Readiness Level (TRL 1–2), it contributes methods and analytical frameworks that may become increasingly relevant in the evolving landscape of vehicle safety.

Over the past few decades, occupant protection systems and vehicle interior layouts have converged toward a high degree of standardization. Safety technologies such as frontal airbags, pre-tensioners, and load limiters are now widely implemented, and most developments have centered on fine-tuning these existing systems rather than introducing disruptive changes. Although the emergence of electric vehicles has required adaptations to account for different mass distributions and structural layouts, such as underfloor battery packs that lower the center of gravity, these changes have largely preserved the traditional occupant seating orientation and restraint strategies (HILS21).

However, the rise of fully automated vehicles (FAVs) is expected to introduce a significant shift in occupant safety paradigms. As Koppel et al. (2019) demonstrated in an international survey, many users anticipate adopting non-traditional seating positions (including reclined, rotated, or rearward-facing configurations) during travel in autonomous vehicles. Such configurations challenge the assumptions underpinning conventional restraint design, which has historically relied on occupants being forward-facing and seated upright. To illustrate said variety of possible interior layouts considered, Figure 5.5-1. presents the five hypothetical seating configurations evaluated in the study (KOPP19).

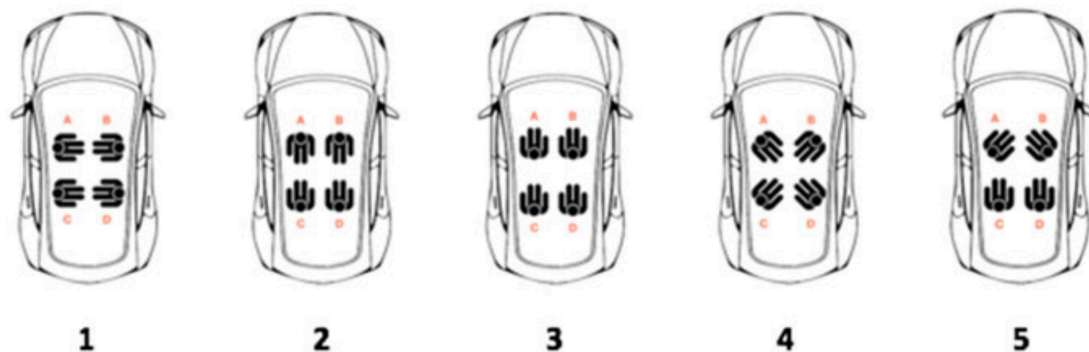


Figure 5.5-1. Hypothetical seating configurations for fully automated vehicles (FAVs) used in the survey by Koppel et al. to explore occupant preferences under various travel scenarios (KOPP19).

This context underscores a critical need for new biomechanical analysis methods capable of characterizing kinematic variability in more diverse and less constrained occupant postures. The methodologies developed in this thesis could provide foundational tools to support future research and validation in this area. For example, the capacity to statistically describe deviations from nominal trajectories and postures may be applied to the development of next-generation anthropomorphic test devices or computational human models adapted to the interior layouts of FAVs.

In this sense, while the present work is not directly translatable to applied safety system design, it occupies an essential place early in the innovation chain. Advancing from TRL 1–2 toward higher readiness levels will require further experimental validation, integration with advanced simulation environments, and collaboration with industrial partners focused on vehicle interior and restraint system design. Nonetheless, as the automotive industry accelerates the deployment of autonomous driving technologies, the analytical approaches proposed here may contribute to improving occupant protection in novel seating configurations.

6. CONCLUSION

The analyses presented in this thesis collectively demonstrate that different dimensionality reduction and averaging techniques can effectively capture the kinematic variability of seated human volunteers subjected to low-speed frontal decelerations.

The B-spline and PCA method successfully compressed trajectory data into principal components, showing that seated height and body mass were significant predictors of forward displacement patterns during deceleration.

The time-based analysis of landmark configurations at maximum head displacement offered a complementary perspective, identifying posture-related differences between young and elderly subjects, particularly in thoracic curvature and inter-landmark distances.

Finally, the arc-length re-parameterization and signal registration approach generated representative mean trajectories and variability corridors, illustrating consistent group-level trends in motion paths while accommodating temporal and geometric heterogeneity across trials.

Together, these results confirm that combining trajectory-based and posture-centric methods yields a more complete understanding of occupant response variability and underscores the influence of age-related factors on kinematic behavior during controlled decelerations.

6.1. Degree of Objective Fulfillment

The objectives defined at the start of this work have been followed and completed in a consistent manner.

The first objective was to implement Principal Component Analysis on B-spline representations of landmark trajectories. When applied, the method allowed the complex motion of the head and torso to be reduced to a set of principal components that captured the main patterns of movement. Regression models showed that these components were linked to anthropometric factors like seated height and body mass, demonstrating that the approach could detect some associations between trajectory shapes and individual characteristics.

The second objective focused on analyzing the relative positions of landmarks at the moment of maximum head displacement. This method was also fully developed and applied. By isolating posture at a single key instant, it became possible to highlight geometric differences between age groups that might not be visible in time-based data. In particular, the analysis showed that elderly subjects had distinct patterns in thoracic curvature and cervical alignment.

The third objective was to use arc-length re-parameterization and signal registration to generate average motion trajectories and variability corridors. This technique worked as intended. It provided a reliable way to align and summarize trajectories without relying on time normalization, and it produced representative mean paths that illustrated the consistency of group-level trends despite variability in timing and amplitude.

Finally, the comparative evaluation of all three methods confirmed their complementarity. The B-spline approach was most effective for connecting motion patterns to anthropometry, the time-based posture analysis was better at detecting static postural differences, and the arc-length method offered a clear visualization of mean motion and dispersion. Together, they formed a coherent set of tools that addressed the complexity of occupant kinematics more completely than any single technique could do on its own.

6.2. Future Work

Several directions could be explored to build on the findings of this thesis and address its limitations.

First, expanding the sample size would improve the statistical power of the analyses. A larger and more balanced cohort, especially with an even number of volunteers in both groups, could help confirm the observed trends in postural variation and strengthen the associations between principal components and anthropometric factors.

Second, further development of the arc-length re-parameterization method could include testing different registration parameters or exploring alternative warping constraints to improve alignment of key motion features. This may lead to even more precise average trajectories and variability corridors.

Third, future studies could incorporate additional variables beyond planar (x, z) trajectories, such as three-dimensional motion data, rotational displacements, dynamic load measurements, or muscle activation signals. Including these dimensions would provide a more comprehensive view of occupant response, allow finer characterization of motion patterns, and help link kinematic variability to mechanical loads and neuromuscular control strategies, thereby supporting deeper biomechanical interpretation and improving the relevance of the findings for safety applications.

Finally and most importantly, while the methods developed and applied in this thesis proved effective in extracting, summarizing, and interpreting kinematic variability, their outcomes were inevitably limited by the small data sample. This constrained the statistical strength and generalizability of some findings. Nevertheless, the analytical approaches, B-spline compression, posture-focused PCA, and arc-length signal registration, demonstrated strong potential as versatile tools for biomechanical analysis. Applied to larger and more consistent datasets, these methods are well positioned to deliver robust insights into occupant motion and to support the development of improved safety models and human ATDs.

As a final remark, it is planned to prepare a scientific article derived from this work. The purpose of this publication will be to present the main findings and methodological approach in a concise format appropriate for a peer-reviewed journal. This article is expected to serve as a way to share the results with the wider biomechanics research community. The aim is to have the manuscript completed and submitted before the end of 2025.

7. REFERENCES

- [SAMU15] Samuels, M.A., Seacrist, T., Huang, S., Balasubramanian, S., Lopez-Valdes, F.J., Kent, R.W., Arbogast, K.B., "Modeling spatial trajectories in dynamics testing using basis splines: application to tracking human volunteers in low-speed frontal impacts", *Computer Methods in Biomechanics and Biomedical Engineering*, 18:12, pp. 1323–1332, 2015.
- [HART22] Hartlen D.C., Cronin D.S., "Arc-Length Re-Parametrization and Signal Registration to Determine a Characteristic Average and Statistical Response Corridors of Biomechanical Data", *Frontiers in Bioengineering and Biotechnology*, 10:843148. doi: 10.3389/fbioe.2022.843148, 2022.
- [WHO25] World Health Organization. "Ageing and health", 2025
- [BABU19] Babulal, G. M., Vivoda, J., Harmon, A., Carr, D. B., Roe, C. M., & Zikmund-Fisher, B. "Older Adults' Expectations about Mortality, Driving Life and Years Left without Driving". *Journal of Gerontological Social Work*, 62(8), 912–929, 2019
- [UNIT15] United Nations, "Transforming our world: the 2030 Agenda for Sustainable Development", *Department of Economic and Social Affairs*, 2015
- [SUN16] Sun, W., Kent, R., Brolin, K., "A method for developing biomechanical response corridors based on principal component analysis", *Journal of Biomechanics*, 49(3), pp. 410–417, 2016.
- [VIVE21] Vives-Torres, C.M., Galgano, A., Forman, J.L., López-Valdés, F.J., "Comparison of Upper Neck Loading in Young Adult and Elderly Volunteers During Low Speed Frontal Impacts", *Frontiers in Bioengineering and Biotechnology*, 9:684003, 2021.
- [CRAN10] Crandall, J.R., Bose, D., Forman, J., Untaroiu, C.D., Arregui-Dalmases, C., Shaw, C.G., Kerrigan, J.R., "Human Surrogates for Injury Biomechanics Research", *Clinical Anatomy*, Vol. 24, pp. 362–371, 2010.
- [LOPE17] Lopez-Valdes F.J., Juste-Lorente O., Lorente A., Piqueras-Lorente A., Danauskienė A., Muehlbauer J., Shick S., Symeonidis, Maza-Frechin M., Peldschus S., "Kinematics and dynamic responses of young and elderly occupants in low-speed frontal tests", *Traffic Injury Prevention*, 2017.
- [BIOM24] Frontiers Research Team, "Biomechanical Analysis of Low-Speed Frontal Deceleration Tests: Methods, Ethics, and Data Processing", *Frontiers in Bioengineering and Biotechnology*, 2024.
- [ARBO09] Arbogast, K.B., Balasubramanian, S., Seacrist, T., et al., "Comparison of kinematic responses of the head and spine for children and adults in low-speed frontal sled tests", *Stapp Car Crash Journal*, Vol. 53, pp. 329–372, 2009.
- [CORM18] Cormier, J., Gwin, L., Reinhart, L., Wood, R., Bain, C., "A Comprehensive Review of Low-Speed Rear Impact Volunteer Studies and a Comparison to Real-World Outcomes", *Spine*, Vol. 43, No. 18, pp. 1250–1258, 2018.
- [HWAN23] Hwang Y.T., Y.Q. Tung, C.S. Chen, B.S. Lin, "B-Spline Modeling of Inertial Measurements for Evaluating Stroke Rehabilitation Effectiveness", *IEEE*

Transactions on Neural Systems and Rehabilitation Engineering, vol. 31, pp. 4008–4016, 2023.

- [WANG88] Wang K., "B-splines joint trajectory planning", *Computers in Industry*, vol. 10, no. 2, pp. 113–122, 1988.
- [FEDE13] Federolf, P., Rohmeyer, L., Zander, V., Rein, R., Baumann, C., Peham, C., "Principal component analysis of biomechanical movement data: a method for analyzing variability in gait and posture", *Journal of Biomechanics*, 46(14):2542–2547, 2013.
- [BRUM23] Brumbelow, M.L., "Female Driver Lower Extremity Injury: Contributing Factors and Crash Test Relevance", *Proceedings of the International Research Council on the Biomechanics of Injury (IRCOBI)*, IRC-23-32, 2023.
- [DONN12] Donnelly, B.R., Moorhouse, K., "Optimized phasing of PMHS response curves for biofidelity targets", *IRCOBI Conference Proceedings*, 2012.
- [DONN14] Donnelly, B.R., Moorhouse, K.M., Rhule, H.H., Stammen, J.A., "A deformation energy approach to normalizing PMHS response data and developing biofidelity targets for dummy design", *Proceedings of the 2014 IRCOBI Conference*, 2014.
- [FREJ23] Frej, D. "Analysis of Head Displacement during a Frontal Collision at a Speed of 20 km/h—Experimental Studies." *Sustainability*, 15, 16015, 2023.
- [HIGU19] Higuchi, K., Yoshida, S., Ono, K., Yamazaki, K., Nakahara, H., "Behaviour of ATD, PMHS and Human Volunteer in Frontal Crash Test", *International Journal of Automotive Engineering*, Vol. 10, No. 4, 2019.
- [FEDE16] Federolf, P.A., "A novel approach to study human posture control: 'Principal movements' obtained from a principal component analysis of kinematic marker data", *Journal of Biomechanics*, Vol. 49, No. 3, pp. 364–370, 2016.
- [WHIT23] White J., Cheng J., Mahmood A., "A General Method for Computing an Average Curve and Statistical Corridors Using Arc-Length Re-Parameterization", *Proceedings of the IRCOBI Conference 2023*, pp. 215–226.
- [CHAU05] Chau T., Young S., Redekop S., "Managing Variability in the Summary and Comparison of Gait Data", *Journal of NeuroEngineering and Rehabilitation*, vol. 2, article 22, 2005.
- [LEE19] Lee M., "Application of Dynamic Time Warping Algorithm for Pattern Similarity of Gait", *Journal of Biomedical Engineering Research*, vol. 40, no. 3, pp. 181–188, 2019.
- [HILS21] Hilster D., L. Leestemaker, A. Hoen, "Safety and Electric Passenger Cars", *CE Delft – Committed to the Environment*, Informe 21.200180.047, 2021.
- [KOPP19] Koppel S., J. Jiménez Octavio, K. Bohman, D. Logan, W. Raphael, L. Quintana Jimenez, F. Lopez-Valdes, "Seating configuration and position preferences in fully automated vehicles", *Traffic Injury Prevention*, DOI: 10.1080/15389588.2019.1625336, june 27th, 2019

ANNEX

1. ALIGNMENT WITH SUSTAINABLE DEVELOPMENT GOALS

This thesis aligns with several Sustainable Development Goals (SDGs) (UNIT15) by advancing knowledge and practices that enhance human health, safety, and innovation in sustainable technologies. Specifically, it relates to the following SDGs:

1.1. Goal 3: Good Health and Well-being

Given that road traffic accidents are a leading cause of death worldwide, the focus of this thesis on improving occupant safety during automotive crashes directly contributes to better health and well-being. By developing more accurate methods for analyzing occupant biomechanics during impacts, this research supports efforts to reduce injury risks in automotive accidents, which is specifically remarked in Goal 3.6.

1.2. Goal 9: Industry, Innovation, and Infrastructure

The analysis of different averaging techniques for describing occupant kinematics also aligns closely with Sustainable Development Goal 9, particularly in the areas of innovation and sustainable industrialization. By developing and testing more accurate methods to analyze and standardize biomechanical data, it aims to contribute to the advancement of automotive safety technology, which is a crucial aspect of sustainable transportation. These methods may allow for better-informed decisions in the design of safety features in vehicles, making them more effective for a diverse population, including older adults. This aligns with the goal's emphasis on building resilient infrastructure and promoting inclusive industrialization.

1.3. Goal 10: Reduced Inequality

By focusing on creating more accurate methods that account for variability across different age groups, particularly comparing young and elderly volunteers, this thesis aims to contribute to the reduction of inequalities in vehicle safety. Older people are often at higher risk for injury in automotive accidents due to changes in their physical condition, including reduced flexibility and slower reaction times. By addressing these disparities, this research may help design more inclusive safety systems, providing effective protection for all occupants and reducing disparities among vulnerable populations.

1.4. Goal 12: Responsible Consumption and Production

By improving the accuracy and reliability of biomechanical data analysis methods, this thesis can contribute to the more efficient use of resources in vehicle safety testing and development. More precise analysis methods can lead to better-targeted improvements in safety features, reducing the need for extensive, repeated crash tests that are resource-intensive. This supports the goal of promoting sustainable consumption and production patterns by enhancing the effectiveness of existing resources, which could reduce both financial and material waste in the automotive industry.

2. ARCGEN

In this section, the main ARCGen original code files are provided.

Adjustments were made to some of them to be able to generate differentiated graphs for the young and elderly groups. These modified code files can be found in section 3.3 of the Annex.

Further ARCGen resources including the original article as well as supporting files can be found in the following site:

<https://es.mathworks.com/matlabcentral/fileexchange/116975-arcgen-arc-length-based-averaging-and-statistics?status=SUCCESS>

2.1. ARCGen Code Library

2.1.1. Arcgen.m

```
%% ARCGen - Arc-length Response Corridor Generator
%
% ARCGen, short for Arc-length Response Corridor Generation, provides
% automated calculation of a characteristic average and response corridors
% on input signals regardless of if said signals are non-monotonic or
% hystertic. This is accomplished by re-parameterizing input signals based
% on arc-length. Corridors are extracted using a marching squares
% algorithm.
%
% If you use ARCGen in your research, please use the following citation:
%   Hartlen D.C. and Cronin D.S. (2022), "Arc-Length Re-Parametrization
%   and Signal Registration to Determine a Characteristic Average and
%   Statistical Response Corridors of Biomechanical Data." Frontiers
%   in Bioengineering and Biotechnology* 10:843148.
%   doi: 10.3389/fbioe.2022.843148
%
% ARCGen is released under a GNU GPL v3 license. No warranty or support is
% provided. The authors hold no responsibility for the validity, accuracy,
% or applicability of any results obtained from this code.
%
% This function has one mandatory input, four outputs, and many optional
% inputs. Optional inputs are defined using name-value pair arguments.
%
% Usage notes:
% It is common to see errors when running this function if the number of
% resampling points or corridor resolution is too sparse or signals
% exhibit significant variability not accounted for through signal
% registration. This tends to manifest in either truncated corridors or the
% code termininating in an error. Often increasing resampling points or
% corridor resolution. Turning 'Diagnostics' to 'detailed' can help
% identify these issues.
%
% Computed corridors will often not extend all the way to the shared origin
% of input signals. This is because small low st. dev. at this shared point
% is too low to be captured during corridor extraction with the marching
% squares algorithm. There are two solutions to this problem. First, one
% could force a minimum corridors size using the 'MinCorridorWidth' option.
% Second, one could manually extend corridors in post-processing.
%
% MANDATORY INPUTS:
% -----
% inputSignals: ARCGen can accomadate three types of input format
% 1) A [nSignal,2] structured array consisting of the following
%    entries. Entries are case-sensitve
%    + data: an [m,2] array containing ordered x-y data
%    + specId: character array containing an identifier for each signal
% 2) A [nSignal,1] structured array consisting of only signal data, no
%    signal IDs. Entries are case-sensitve
%    + data: an [m,2] array containing ordered x-y data
% 3) A cell array of length nSignal containing [m,2] arrays of each input
%    signal.
%
% OPTIONAL INPUTS:
% -----
% nResamplePoints: integer defining the number of points used to
% re-parameterize input signals. Default: 100.
% CorridorRes: integeer defining the number of grid points used for the
% marching squares algorithm. The sampling grid for the marching
% squares algorithm extends 120% of extreme corridors. This parameter
% defines the number of points along each side of the grid.
% Default: 100. It is common to increase this significantly.
% NormalizeSignals: character array used to turn on signal normalization.
% Options: 'on' (default), 'off'
% EllipseKFact: float used to scale the major and minor axis of the
```



```
% ellipses used for corridor generation. This value corresponds to
% the square root of the chi-squared CDF. Default: 1.0 (creates
% corridors one standard deviation along the x and y axes)
% Diagnostics: character array used to activate diagnostic plots. Useful
% for debugging errors. Options: 'off' (default), 'on', 'detailed'.
% MinCorridorWidth: Factor used to enforce a minimum corridor width. Any
% st.dev. less than 'MinCorridorFactor'*max(st.dev.) is replaced with
% 'MinCorridorFactor'*max(st.dev.). x & y axes are handled
% separately. A value of 0 (default) disables forcing minimum width.
% nWarpCtrlPts: integer that sets the number of interior control points
% used for signal registration. A value of 0 (default) disables
% signal registration
% WarpingPenalty: float specifying the penalty factor used during the
% signal registration process. A value of 10^-2 (default) to 10^3 is
% recommended, but the exact value will need to be tuned to a
% specific problem.
% UseParallel: Character array used to enable parallel thread calculations
% for signal registration and envelope extraction. Significantly
% reduces runtime when signals have 100k+ points or 500+ resampling
% points and corridor resolution. Requires the Parallel Computing
% Toolbox Options: 'on', 'off' (default).
%
% MANDATORY OUTPUTS:
% -----
% charAvg: an [nResamplePoints,2] array containing the computed
% characteristic average.
% innerCorr: an [nResamplePoints,2] array containing points defining the
% inner corridor
% outerCorr: an [nResamplePoints,2] array containing points defining the
% outer corridor
%
% OPTIONAL OUTPUTS:
% -----
% processedSignalData: a structure array that outputs processed signals,
% basic statistics, and warping control points
% debugData: a structure that provides a wealth of debugging information,
% including raw average and st. dev. data, correlation scores before
% and after registration, and other.
%
% Copyright (c) 2022 Devon C. Hartlen

function [charAvg, innerCorr, outerCorr, varargout] = ...
    arcgen(inputSignals,varargin)

%% Setup Name-Value Argument parser
nvArgObj = inputParser;
addParameter(nvArgObj, 'nResamplePoints', 100);
addParameter(nvArgObj, 'Diagnostics', 'off');
addParameter(nvArgObj, 'NormalizeSignals', 'on');
addParameter(nvArgObj, 'EllipseKFact', 1);
addParameter(nvArgObj, 'CorridorRes', 100);
addParameter(nvArgObj, 'MinCorridorWidth', 0);
addParameter(nvArgObj, 'nWarpCtrlPts', 0);
addParameter(nvArgObj, 'WarpingPenalty', 1e-2);
addParameter(nvArgObj, 'UseParallel', 'off');
nvArgObj.KeepUnmatched = true;
nvArgObj.CaseSensitive = false;

parse(nvArgObj,varargin{:});
nvArg = nvArgObj.Results; % Structure created for convenience

% check if parallel toolbox is installed, then if parpool is running. Error
% out if not installed, start pool if not already started.
v = ver;
hasParallel = any(strcmp(cellstr(char(v.Name)), 'Parallel Computing Toolbox'));
if strcmp(nvArg.UseParallel,'on')
    if ~hasParallel
        error('Parallel Computing Toolbox is not installed. Set option UseParallel to
off')
```

```

end
p = gcp('nocreate');
if isempty(p)
    parpool();
end
end

%% Add third party functions to path
funcPath = mfilename('fullpath');
funcPath = fileparts(funcPath);
addpath(fullfile(funcPath, 'ThirdPartyFunctions'));

%% Process input options
% Check if structure with specID, struct w/o specID, cell array. Error out
% otherwise. Places inputs into structure format.
if isstruct(inputSignals)
    if ~isfield(inputSignals,'specId')
        for iSignal = 1:length(inputSignals)
            inputSignals(iSignal).specId = ...
                ['Signal ' num2str(iSignal,'%3d')];
        end
    end
elseif iscell(inputSignals)
    inputSignals = cell2struct(inputSignals,'data');
    for iSignal = 1:length(inputSignals)
        inputSignals(iSignal).specId = ['Signal ' num2str(iSignal,'%3d')];
    end
end

%% Compute arclength based on input signal datapoints
% Do not perform normalization
if strcmp(nvArg.NormalizeSignals,'off')
    for iSignal = 1:length(inputSignals)
        temp = inputSignals(iSignal).data; % Temporary for convenience
        % Compute arc-length between each data point
        segments = sqrt( (temp(1:end-1,1)-temp(2:end,1)).^2 ...
            + (temp(1:end-1,2)-temp(2:end,2)).^2);
        alen = cumsum([0;segments]);
        % Append cumulative arc length to data array
        inputSignals(iSignal).data = [inputSignals(iSignal).data,alen];
        % Compute normalized arc-length
        inputSignals(iSignal).maxAlen = max(alen);
        inputSignals(iSignal).data = [inputSignals(iSignal).data,...
            alen./inputSignals(iSignal).maxAlen];
        % Determine max [x,y] data
        tempMax = max(temp,[],1);
        inputSignals(iSignal).xMax = tempMax(1);
        inputSignals(iSignal).yMax = tempMax(2);
        % Remove spurious duplicates
        [~,index,~] = unique(inputSignals(iSignal).data(:,4));
        inputSignals(iSignal).data = inputSignals(iSignal).data(index,:);
    end

% Perform magnitude normalization based on bounding box
elseif strcmp(nvArg.NormalizeSignals,'on')
    % Determine bounding box of individual signals
    for iSignal = 1:length(inputSignals)
        tempMin = min(inputSignals(iSignal).data,[],1);
        inputSignals(iSignal).xMin = tempMin(1);
        inputSignals(iSignal).yMin = tempMin(2);
        tempMax = max(inputSignals(iSignal).data,[],1);
        inputSignals(iSignal).xMax = tempMax(1);
        inputSignals(iSignal).yMax = tempMax(2);
    end
    xBound = [mean([inputSignals.xMin]), mean([inputSignals.xMax])];
    yBound = [mean([inputSignals.yMin]), mean([inputSignals.yMax])];
    % Normalize the axis of each signal, then do arc-length calcs
    for iSignal = 1:length(inputSignals)
        temp = inputSignals(iSignal).data; % Temporary for convenience

```

```
% Normalize from bounding box to [-1,1]
temp = [temp(:,1)./(xBound(2)-xBound(1)),...
        temp(:,2)./(yBound(2)-yBound(1))];
% Compute arc-length between each data point
segments = sqrt( (temp(1:end-1,1)-temp(2:end,1)).^2 ...
                + (temp(1:end-1,2)-temp(2:end,2)).^2);
alen = cumsum([0;segments]);
% Append cumulative arc length to data array
inputSignals(iSignal).data = [inputSignals(iSignal).data,alen];
% Compute normalized arc-length
inputSignals(iSignal).maxAlen = max(alen);
inputSignals(iSignal).data = [inputSignals(iSignal).data,...
                              alen./inputSignals(iSignal).maxAlen];
% Determine max [x,y] data
tempMax = max(abs(temp),[],1);
inputSignals(iSignal).xNormMax = tempMax(1);
inputSignals(iSignal).yNormMax = tempMax(2);
% Remove spurious duplicates
[~,index,~] = unique(inputSignals(iSignal).data(:,4));
inputSignals(iSignal).data = inputSignals(iSignal).data(index,:);
end

% Error handling if NormalizeSignals argument is not defined correctly
else
    error('Normalization method not recognized')
end

% Compute mean and median arc-length deviation
meanAlen = mean([inputSignals.maxAlen]);
for iSignal=1:length(inputSignals)
    inputSignals(iSignal).meanDevs = ...
        inputSignals(iSignal).maxAlen-meanAlen;
end

medianAlen = median([inputSignals.maxAlen]);
for iSignal=1:length(inputSignals)
    inputSignals(iSignal).medianDev = ...
        inputSignals(iSignal).maxAlen-medianAlen;
end

%% Resample response signal based on normalized arc-length
for iSignal=1:length(inputSignals)
    % Linear-interpolation for x,y data against arc-length
    normAlen = linspace(0,inputSignals(iSignal).data(end,4),...
        nvArg.nResamplePoints)';
    resampX = interp1(inputSignals(iSignal).data(:,4),...
        inputSignals(iSignal).data(:,1), normAlen);
    resampY = interp1(inputSignals(iSignal).data(:,4),...
        inputSignals(iSignal).data(:,2), normAlen);
    % Resulting array is normalized arc-length, resampled x, resam. y
    inputSignals(iSignal).normalizedSignal = [normAlen, resampX, resampY];
end

%% For each resampled point, determine average and standard deviation across signals
% Initialize arrays
charAvg = zeros(nvArg.nResamplePoints,2);
stdevData = zeros(nvArg.nResamplePoints,2);

for iPoints=1:nvArg.nResamplePoints
    clear temp; % probably cleaner way to do this.
    % collect specific point from each signal
    for iSignal=1:length(inputSignals)
        temp(iSignal,:) = inputSignals(iSignal).normalizedSignal(iPoints,2:3);
    end
    charAvg(iPoints,:) = mean(temp,1);
    stdevData(iPoints,:) = std(temp,1);
end
% Assign characteristic average and st. dev. data to a debug structure
debugOutput.charAvg = charAvg;
```

```

debugOutput.stdevData = stdevData;

%% Align normalized arc-length signals based on minimized correlation.
% Enabled by option 'nWarpCtrlPts'. If 0, skip alignment.
if nvArg.nWarpCtrlPts > 0
    % Assemble signal matrices prior to correlation
    signalX = zeros(nvArg.nResamplePoints, length(inputSignals));
    signalY = zeros(nvArg.nResamplePoints, length(inputSignals));
    for i=1:length(inputSignals)
        signalX(:,i) = inputSignals(i).normalizedSignal(:,2);
        signalY(:,i) = inputSignals(i).normalizedSignal(:,3);
    end
    [meanCorrScore, corrArray] = evalCorrScore(signalX,signalY);
    % Assign pre-optimized correlation scores to debug structure
    debugOutput.preWarpCorrArray = corrArray;
    debugOutput.preWarpMeanCorrScore = meanCorrScore;

    % Optimize warp points for arbitrary n warping points. Build bounds,
    % constraints, and x0s
    nWarp = nvArg.nWarpCtrlPts;
    nSignal = length(inputSignals);

    if nWarp == 1 % nWarp == 1 is a special case as inequalities aren't needed
        x0 = 0.50.*ones(nSignal*2,1);
        lb = 0.15.*ones(nSignal*2,1);
        ub = 0.85.*ones(nSignal*2,1);
        A = [];
        b = [];
    elseif nWarp >= 15
        error('Specifying more than 10 interior warping points is not supported')
    else
        x0 = zeros(nWarp*(nSignal*2),1);
        for i = 1:nWarp
            x0(((i-1)*nSignal)+(1:nSignal) + (i-1)*nSignal) =
i/(nWarp+1).*ones(nSignal,1);
            x0(((i-1)*nSignal)+(1:nSignal)+i*(nSignal)) = i/(nWarp+1).*ones(nSignal,1);
        end
        lb = 0.05.*ones(nWarp*(nSignal*2),1);
        ub = 0.95.*ones(nWarp*(nSignal*2),1);
        A = zeros((nWarp-1)*(nSignal*2), nWarp*(nSignal*2));
        b = -0.05.*ones((nWarp-1)*(nSignal*2), 1); % Force some separation between
warped points
        for iSignal = 1:(nSignal*2)
            for iWarp = 1:(nWarp-1)
                A(iSignal+(iWarp-1)*(nSignal*2), iSignal+(iWarp-1)*(nSignal*2)) = 1;
                A(iSignal+(iWarp-1)*(nSignal*2), iSignal+iWarp*(nSignal*2)) = -1;
            end
        end
    end
end

% Setup optimization options ('UseParallel' option active here)
if strcmp(nvArg.UseParallel,'on')
    optOptions = optimoptions('fmincon',...
        'MaxFunctionEvaluations',max(3000, (nWarp+1).*1000),...
        'Display','off',...
        'UseParallel',true);
else
    optOptions = optimoptions('fmincon',...
        'MaxFunctionEvaluations',max(3000, (nWarp+1).*1000),...
        'Display','off',...
        'UseParallel',false);
end

% Execute optimization and compute warped signals
optWarpArray = fmincon(@(x)warpingObjective(x,nWarp,...
    inputSignals,nvArg),...
    x0, A, b, [], [], lb, ub, [], optOptions);
optWarpArray = reshape(optWarpArray,[],nWarp);
[warpedSignals, signalX, signalY] = ...

```

```
warpArcLength(optWarpArray,inputSignals,nvArg.nResamplePoints);

% Compute correlation score
[meanCorrScore, corrArray] = evalCorrScore(signalX,signalY);
% Assign warped correlation scores to debug structure
debugOutput.warpedCorrArray = corrArray;
debugOutput.warpedMeanCorrScore = meanCorrScore;

% Replace 'normalizedSignal' in 'responseSignal' and compute average and
% standard deviation.
for iSignal = 1:length(inputSignals)
    inputSignals(iSignal).normalizedSignal = warpedSignals{iSignal};
    inputSignals(iSignal).warpControlPoints = ...
        [[0,optWarpArray(iSignal+nSignal,:),1];...
        [0,optWarpArray(iSignal,:),1]];
end
for iPoints=1:nvArg.nResamplePoints
    clear temp; % probably cleaner way to do this.
    % collect specific point from each signal
    for iSignal=1:length(inputSignals)
        temp(iSignal,:) = ...
            inputSignals(iSignal).normalizedSignal(iPoints,2:3);
    end
    charAvg(iPoints,:) = mean(temp,1);
    stdevData(iPoints,:) = std(temp,1);
end
end

%% Clamp minimum corridor width. Disabled if 'MinCorridorWidth' == 0
% Include influence of corridor scaling factor 'EllipseKFact'
if nvArg.MinCorridorWidth > 0
    % Replace any stDevData below maximum st.dev. * 'MinCorridorWidth'
    index = stdevData <...
        (nvArg.MinCorridorWidth .* max(stdevData) .* nvArg.EllipseKFact);
    stdevData(index(:,1),1) = (nvArg.MinCorridorWidth .* nvArg.EllipseKFact...
        .* max(stdevData(:,1)));
    stdevData(index(:,2),2) = (nvArg.MinCorridorWidth .* nvArg.EllipseKFact...
        .* max(stdevData(:,2)));
end

%% Diagnostic: Plot normalized signals and St. Devs.
if strcmp(nvArg.Diagnostics,'on') || strcmpi(nvArg.Diagnostics,'detailed')
    figure('Name','Diagnostic Signals');
    cmap = lines(length(inputSignals));

    % Plot normalized x,y data
    subplot(2,2,1); hold on;
    for iSignal=1:length(inputSignals)
        pSignal(iSignal) = plot(inputSignals(iSignal).normalizedSignal(:,2),...
            inputSignals(iSignal).normalizedSignal(:,3),'-','...',...
            'color',cmap(iSignal,:),...
            'DisplayName',inputSignals(iSignal).specId);
        if (strcmp(nvArg.NormalizeSignals,'off') || ...
            strcmp(nvArg.NormalizeSignals,'on'))
            continue
        else
            plot(inputSignals(iSignal).data(inputSignals(iSignal).alignInd,1),...
                inputSignals(iSignal).data(inputSignals(iSignal).alignInd,2),...
                'kx','LineWidth',2.0)
        end
    end
    xlabel('x-data')
    ylabel('y-data')
    legend(pSignal, 'location', 'Best')
    title('Arc-length Discretized Normalized Signals')

    % Plot warppping functions
    subplot(2,2,2); hold on
```

```

clear pSignal
if nvArg.nWarpCtrlPts > 0
    colours = lines(nSignal);
    for iSignal = 1:nSignal
        pSignal(iSignal) = plot(inputSignals(iSignal).data(:,4),...

pchip([0,optWarpArray(iSignal+nSignal,:),1],[0,optWarpArray(iSignal,:),1],...
        inputSignals(iSignal).data(:,4)),...
        '.-','DisplayName',inputSignals(iSignal).specId,...
        'color',colours(iSignal,:),...
        'DisplayName',inputSignals(iSignal).specId);

plot([0,optWarpArray(iSignal+nSignal,:),1],[0,optWarpArray(iSignal,:),1],'x',...
        'color',colours(iSignal,:), 'MarkerSize',12,'LineWidth',2.0)
        title('Warping functions');
        legend(pSignal, 'location', 'Best')
    end
else
    title('No Warping Performed');
end
plot([0,1],[0,1],'--','color',0.3.*[1,1,1])
xlabel('Unwarped Normalized Arc-length')
ylabel('Warped Normalized Arc-length')

% Plot normalized x data against arc-length with st. dev.
subplot(2,2,3); hold on;
errorbar(inputSignals(1).normalizedSignal(:,1),charAvg(:,1),...
        stdevData(:,1),'color',0.5.*[1,1,1])
cmap = lines;
for iSignal=1:length(inputSignals)
    plot(inputSignals(iSignal).normalizedSignal(:,1),...
        inputSignals(iSignal).normalizedSignal(:,2),'.-',...
        'color',cmap(iSignal,:))
end
xlabel('Normalized Arc-length')
ylabel('x-data')
title('Average and St.Dev. of X-Data')

% Plot normalized y data against arc-length with st. dev.
subplot(2,2,4); hold on;
errorbar(inputSignals(1).normalizedSignal(:,1),charAvg(:,2),...
        stdevData(:,2),'color',0.5.*[1,1,1])
cmap = lines;
for iSignal=1:length(inputSignals)
    plot(inputSignals(iSignal).normalizedSignal(:,1),...
        inputSignals(iSignal).normalizedSignal(:,3),'.-',...
        'color',cmap(iSignal,:))
end
xlabel('Normalized Arc-length')
ylabel('y-data')
title('Average and St.Dev. of Y-Data')
end

if strcmpi(nvArg.Diagnostics,'detailed')
    % Plot ellipses
    figure('Name','Ellipses and Corridor Extraction Debug'); hold on;
    cmap = cbrewer2('set2',2);
    colormap(cmap);
    % plot ellipses based on standard deviation
    for iPoint=1:nvArg.nResamplePoints
        ellipse(stdevData(iPoint,1).*nvArg.EllipseKFact,...
            stdevData(iPoint,2).*nvArg.EllipseKFact,0,...
            charAvg(iPoint,1), charAvg(iPoint,2),...
            0.8.*[1,1,1]);
    end
    cmap = lines(length(inputSignals));
    for iSignal=1:length(inputSignals)
        plot(inputSignals(iSignal).data(:,1),...
            inputSignals(iSignal).data(:,2),'-',...

```

```

        'DisplayName',inputSignals(iSignal).specId,...
        'Color', cmap(iSignal,:))
    end
    plot(charAvg(:,1),charAvg(:,2),'.-k','DisplayName','Char Avg',...
        'LineWidth',2.0,'MarkerSize',16)
end

%% Begin marching squares algorithm
% Create grids based on upper and lower of characteristic average plus 120%
% of maximum standard deviation
scaleFact = 1.2*nvArg.EllipseKFact;
[xx,yy] = meshgrid(...
    linspace(min(charAvg(:,1)) - scaleFact*max(stdevData(:,1)), ...
        max(charAvg(:,1)) + scaleFact*max(stdevData(:,1)),...
        nvArg.CorridorRes),...
    linspace(min(charAvg(:,2)) - scaleFact*max(stdevData(:,2)), ...
        max(charAvg(:,2)) + scaleFact*max(stdevData(:,2)),...
        nvArg.CorridorRes));
zz = zeros(size(xx)); % initialize grid of ellipse values

% For each grid point, find the max of each standard deviation ellipse
kFact = nvArg.EllipseKFact; % faster if no struct call in inner loop.
nRes = nvArg.CorridorRes; % again, for speed
% If 'UseParallel' is 'on', grid evaluation is performed using a parallel
% for loop.
if strcmp(nvArg.UseParallel,'on')
    parfor iPt = 1:nRes
        for jPt = 1:nRes
            zz(iPt,jPt) = max(...
                ((xx(iPt,jPt) - charAvg(:,1)).^2 ./ ...
                    (stdevData(:,1).*kFact).^2 ...
                    + (yy(iPt,jPt) - charAvg(:,2)).^2 ./ ...
                    (stdevData(:,2).*kFact).^2).^(-1));
        end
    end
% otherwise, use a standard forloop
else
    for iPt = 1:nRes
        for jPt = 1:nRes
            zz(iPt,jPt) = max(...
                ((xx(iPt,jPt) - charAvg(:,1)).^2 ./ ...
                    (stdevData(:,1).*kFact).^2 ...
                    + (yy(iPt,jPt) - charAvg(:,2)).^2 ./ ...
                    (stdevData(:,2).*kFact).^2).^(-1));
        end
    end
end

% The following segments is the marching squares algorithm. The goal of this
% algorithm is to find the zz=1 isoline, as this represents the outer
% boundary of all ellipses.
%
% Described in brief, this algorithm goes through each point, looking at
% its and its neighbours values. There are only 16 configurations of these
% squares or cells. Based on the configuration, add the appropriate line
% segments. This method uses linear interpolation to increase accuracy.
% Initialize line segments for speed. This line may cause issues, as it
% assumes maximum size. Bump up 10 if it does.
lineSegments = zeros(10*max(nvArg.nResamplePoints,nvArg.CorridorRes),4);
iSeg = 0;
for iPt = 1:(nvArg.CorridorRes-1) % Rows (y-axis)
    for jPt = 1:(nvArg.CorridorRes-1) % Columns (x-axis)
        % Cell value definition
        % 1 -- 2
        % |   |
        % |   |
        % 8 -- 4
        %
        % REMEMBER!!!!

```

```
% array(i,j) = array(rows, columns,) = array(y,x)

% By carefully defining cell values and definitions, we can use
% binary to simplify logic though a integer based switch case %
cellValue = ...
    1*(zz(iPt,jPt)>1) + ...
    2*(zz(iPt+1,jPt)>1) + ...
    4*(zz(iPt+1,jPt+1)>1) + ...
    8*(zz(iPt,jPt+1)>1) + 1;

switch cellValue
    case 1
        % No Vertices
    case 2
        % South-West
        iSeg = iSeg+1;
        lineSegments(iSeg,:) = ...

[interpVal(xx(iPt,jPt),zz(iPt,jPt),xx(iPt,jPt+1),zz(iPt,jPt+1)),yy(iPt,jPt),...
    xx(iPt,jPt),
interpVal(yy(iPt,jPt),zz(iPt,jPt),yy(iPt+1,jPt),zz(iPt+1,jPt))];
    case 3
        % West-North
        iSeg = iSeg+1;
        lineSegments(iSeg,:) = ...
            [xx(iPt+1,jPt),interpVal(yy(iPt,jPt),zz(iPt,jPt),
yy(iPt+1,jPt),zz(iPt+1,jPt))),...
            interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),
xx(iPt+1,jPt+1),zz(iPt+1,jPt+1)),yy(iPt+1,jPt)];
    case 4
        % North-South
        iSeg = iSeg+1;
        lineSegments(iSeg,:) = ...

[interpVal(xx(iPt,jPt),zz(iPt,jPt),xx(iPt,jPt+1),zz(iPt,jPt+1)),yy(iPt,jPt) ...
    interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),xx(iPt+1,jPt+1),
zz(iPt+1,jPt+1)),yy(iPt+1,jPt)];
    case 5
        % North-East
        iSeg = iSeg+1;
        lineSegments(iSeg,:) = ...

[interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),xx(iPt+1,jPt+1),zz(iPt+1,jPt+1)),yy(iPt+1,jPt+1),
...
    xx(iPt+1,jPt+1),interpVal(yy(iPt+1,jPt+1),zz(iPt+1,jPt+1),
yy(iPt,jPt+1),zz(iPt,jPt+1))];
    case 6 % Ambiguous
        centerVal = mean([zz(iPt,jPt), zz(iPt+1,jPt), zz(iPt+1,jPt+1), zz(iPt,
jPt+1))]);
        if centerVal >= 1
            % West-North
            iSeg = iSeg+1;
            lineSegments(iSeg,:) = ...
                [xx(iPt+1,jPt),interpVal(yy(iPt,jPt),zz(iPt,jPt),
yy(iPt+1,jPt),zz(iPt+1,jPt))),...
                interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),
xx(iPt+1,jPt+1),zz(iPt+1,jPt+1)),yy(iPt+1,jPt)];
            % South - East
            iSeg = iSeg+1;
            lineSegments(iSeg,:) = ...

[interpVal(xx(iPt,jPt+1),zz(iPt,jPt+1),xx(iPt,jPt),zz(iPt,jPt)),yy(iPt,jPt+1),...
xx(iPt,jPt+1),interpVal(yy(iPt,jPt+1),zz(iPt,jPt+1),yy(iPt+1,jPt+1),zz(iPt+1,jPt+1))];
        else
            % South-West
            iSeg = iSeg+1;
            lineSegments(iSeg,:) = ...
```



```
[interpVal(xx(iPt,jPt),zz(iPt,jPt),xx(iPt,jPt+1),zz(iPt,jPt+1)),yy(iPt,jPt),...
    xx(iPt,jPt),
interpVal(yy(iPt,jPt),zz(iPt,jPt),yy(iPt+1,jPt),zz(iPt+1,jPt))]);
    % North-East
    iSeg = iSeg+1;
    lineSegments(iSeg,:) = ...

[interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),xx(iPt+1,jPt+1),zz(iPt+1,jPt+1)),yy(iPt+1,jPt+1),
...
    xx(iPt+1,jPt+1),interpVal(yy(iPt+1,jPt+1),zz(iPt+1,jPt+1),
yy(iPt,jPt+1), zz(iPt,jPt+1))]);
    end
    case 7
        % West-East
        iSeg = iSeg+1;
        lineSegments(iSeg,:) = ...

[xx(iPt,jPt),interpVal(yy(iPt,jPt),zz(iPt,jPt),yy(iPt+1,jPt),zz(iPt+1,jPt)),...
xx(iPt,jPt+1),interpVal(yy(iPt,jPt+1),zz(iPt,jPt+1),yy(iPt+1,jPt+1),zz(iPt+1,jPt+1))]);
    case 8
        % South - East
        iSeg = iSeg+1;
        lineSegments(iSeg,:) = ...

[interpVal(xx(iPt,jPt+1),zz(iPt,jPt+1),xx(iPt,jPt),zz(iPt,jPt)),yy(iPt,jPt+1),...
xx(iPt,jPt+1),interpVal(yy(iPt,jPt+1),zz(iPt,jPt+1),yy(iPt+1,jPt+1),zz(iPt+1,jPt+1))]);
    case 9
        % South - East
        iSeg = iSeg+1;
        lineSegments(iSeg,:) = ...

[interpVal(xx(iPt,jPt+1),zz(iPt,jPt+1),xx(iPt,jPt),zz(iPt,jPt)),yy(iPt,jPt+1),...
xx(iPt,jPt+1),interpVal(yy(iPt,jPt+1),zz(iPt,jPt+1),yy(iPt+1,jPt+1),zz(iPt+1,jPt+1))]);
    case 10
        % West-East
        iSeg = iSeg+1;
        lineSegments(iSeg,:) = ...

[xx(iPt,jPt),interpVal(yy(iPt,jPt),zz(iPt,jPt),yy(iPt+1,jPt),zz(iPt+1,jPt)),...
xx(iPt,jPt+1),interpVal(yy(iPt,jPt+1),zz(iPt,jPt+1),yy(iPt+1,jPt+1),zz(iPt+1,jPt+1))]);
    case 11 % Ambiguous
        centerVal = mean([zz(iPt,jPt), zz(iPt+1,jPt), zz(iPt+1,jPt+1), zz(iPt,
jPt+1)]);
        if centerVal >= 1
            % South-West
            iSeg = iSeg+1;
            lineSegments(iSeg,:) = ...

[interpVal(xx(iPt,jPt),zz(iPt,jPt),xx(iPt,jPt+1),zz(iPt,jPt+1)),yy(iPt,jPt),...
    xx(iPt,jPt),
interpVal(yy(iPt,jPt),zz(iPt,jPt),yy(iPt+1,jPt),zz(iPt+1,jPt))]);
    % North-East
    iSeg = iSeg+1;
    lineSegments(iSeg,:) = ...

[interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),xx(iPt+1,jPt+1),zz(iPt+1,jPt+1)),yy(iPt+1,jPt+1),
...
    xx(iPt+1,jPt+1),interpVal(yy(iPt+1,jPt+1),zz(iPt+1,jPt+1),
yy(iPt,jPt+1), zz(iPt,jPt+1))]);
    else
        % West-North
        iSeg = iSeg+1;
        lineSegments(iSeg,:) = ...
```

```

[xx(iPt+1,jPt),interpVal(yy(iPt,jPt),zz(iPt,jPt),
yy(iPt+1,jPt),zz(iPt+1,jPt)),...
interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),
xx(iPt+1,jPt+1),zz(iPt+1,jPt+1)),yy(iPt+1,jPt))];
% South-East
iSeg = iSeg+1;
lineSegments(iSeg,:) = ...

[interpVal(xx(iPt,jPt+1),zz(iPt,jPt+1),xx(iPt,jPt),zz(iPt,jPt)),yy(iPt,jPt+1),...
xx(iPt,jPt+1),interpVal(yy(iPt,jPt+1),zz(iPt,jPt+1),yy(iPt+1,jPt+1),zz(iPt+1,jPt+1)))];
end
case 12
% North-East
iSeg = iSeg+1;
lineSegments(iSeg,:) = ...

[interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),xx(iPt+1,jPt+1),zz(iPt+1,jPt+1)),yy(iPt+1,jPt+1),
...
xx(iPt+1,jPt+1),interpVal(yy(iPt+1,jPt+1),zz(iPt+1,jPt+1),
yy(iPt,jPt+1),zz(iPt,jPt+1)))];
case 13
% North-South
iSeg = iSeg+1;
lineSegments(iSeg,:) = ...

[interpVal(xx(iPt,jPt),zz(iPt,jPt),xx(iPt,jPt+1),zz(iPt,jPt+1)),yy(iPt,jPt) ...
interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),xx(iPt+1,jPt+1),
zz(iPt+1,jPt+1)),yy(iPt+1,jPt))];
case 14
% West-North
iSeg = iSeg+1;
lineSegments(iSeg,:) = ...
[xx(iPt+1,jPt),interpVal(yy(iPt,jPt),zz(iPt,jPt),
yy(iPt+1,jPt),zz(iPt+1,jPt)),...
interpVal(xx(iPt+1,jPt),zz(iPt+1,jPt),
xx(iPt+1,jPt+1),zz(iPt+1,jPt+1)),yy(iPt+1,jPt))];
case 15
% South-West
iSeg = iSeg+1;
lineSegments(iSeg,:) = ...

[interpVal(xx(iPt,jPt),zz(iPt,jPt),xx(iPt,jPt+1),zz(iPt,jPt+1)),yy(iPt,jPt),...
xx(iPt,jPt),
interpVal(yy(iPt,jPt),zz(iPt,jPt),yy(iPt+1,jPt),zz(iPt+1,jPt)))];
case 16
% No vertices
end
end
end
lineSegments = lineSegments(1:iSeg,:);

% Extract list of unique vertices from line segmens
vertices = [lineSegments(:,1:2);lineSegments(:,3:4)];
vertices = uniquetol(vertices,eps,'ByRows', true);

% Create a vertex connectivity table. The 1e-12 value is here because
% floats don't round well and == will not work.
vertConn = zeros(size(lineSegments,1),2);
for i = 1:length(vertConn)
index = all(abs(lineSegments(:,1:2) - vertices(i,:)) < 1e-12,2);
vertConn(index,1) = i;
index = all(abs(lineSegments(:,3:4) - vertices(i,:)) < 1e-12,2);
vertConn(index,2) = i;
end

%% Start line segments sorting and envelope extraction
nEnvelopes = 1;
allEnvelopes(1,1) = 1; % First entry is always vertex 1

```

```

for i = 1:size(vertConn,1)-1
    % save vertex to find
    vertToFind = vertConn(i,2);
    j = i+1; % helper index
    % Find connecting node
    foundShiftedInd = ...
        find(any(vertConn(j:end,:) == vertToFind,2), 1, 'first');
    % If we have found an index
    if ~isempty(foundShiftedInd)
        foundInd = foundShiftedInd + i;
        % swap found vert conn row with j row
        temp = vertConn(j,:);
        % Now, decide whether to flip found row. We want vertex 2 of
        % previous line to be node 1 of the new line.
        if (vertConn(foundInd,1) == vertToFind)
            vertConn(j,:) = vertConn(foundInd, [1,2]);
        else
            vertConn(j,:) = vertConn(foundInd, [2,1]);
        end
        % Logic to prevent overwriting, if found row is next row.
        if (foundInd ~= j)
            vertConn(foundInd,:) = temp;
        end
    % If we did not find an index, we either may have an open envelope or
    % envelope may be convex and loops back on itself.
    else
        % Check to see if we can find the first vertex in envelope
        % appearing again (check for closure)
        vertToFind = vertConn(allEnvelopes(nEnvelopes,1));
        foundShiftedInd = ...
            find(any(vertConn(j:end,:) == vertToFind,2), 1, 'first');
        % If we do not find an index, it means this envelope is complete
        % and manifold
        if isempty(foundShiftedInd)
            % Assign indices to finish current envelope, initialize next
            allEnvelopes(nEnvelopes,2) = i;
            nEnvelopes = nEnvelopes + 1;
            allEnvelopes(nEnvelopes, 1) = j;
        else
            % This error should only occur if envelopes extend beyond
            % sampling grid, which they should not.
            error('Literal Edge Case')
        end
    end
end
allEnvelopes(nEnvelopes,2) = j;

% Find largest envelope
[~,envInds] = max(allEnvelopes(:,2)-allEnvelopes(:,1));

% Convert indices in envelopes to array of (x,y)
envInds = allEnvelopes(envInds, :);
envelope = vertices(vertConn(envInds(1):envInds(2),1,:),:);

% For debugging, plot all envelopes
if strcmpi(nvArg.Diagnostics,'detailed')
    for iEnv = 1:nEnvelopes
        envInds = allEnvelopes(iEnv, :);
        plot(vertices(vertConn(envInds(1):envInds(2),1),1),...
            vertices(vertConn(envInds(1):envInds(2),1),2),...
            '-b','LineWidth',1.0)
    end
end

%% Divide the envelope into corridors.
% To break the largest envelop into inner and outer corridors, we need to
% account for several edge cases. First, we test to see if there are any
% intercepts of the characteristic average and the largest envelope.

```

```

closedEnvelope = [envelope; envelope(1,:)];
[~,~,indexIntercept] = polyxpoly(closedEnvelope(:,1),closedEnvelope(:,2),...
    charAvg(:,1),charAvg(:,2));

% If we find two intercepts, then we have no problem
if size(indexIntercept,1) >=2
    iIntStart = indexIntercept(1,1);
    iIntEnd = indexIntercept(end,1);

% If we find only one intercept, we need to determine if the intercept is a
% the start or end of the envelope. Then we need to extend the opposite
% side of the characteristic average to intercept the envelope.
elseif size(indexIntercept,1) == 1
    % If the single found point is inside the envelope, the found intercept
    % is at the end. Therefore extend the start
    if inpolygon(charAvg(indexIntercept(2),1),...
        charAvg(indexIntercept(2),2), envelope(:,1),envelope(:,2))
        iIntEnd = indexIntercept(1);
        [iIntStart,~] = rayxpoly(charAvg(2,:)',...
            (charAvg(1,:)-charAvg(2,:))', closedEnvelope);
        iIntStart = iIntStart(1);

    % If the single found point is outside the envelope, the found
    % intercept is the start
    else
        iIntStart = indexIntercept(1);
        [iIntEnd,~] = rayxpoly(charAvg(end-1,:)',...
            (charAvg(end,:)-charAvg(end-1,:))', closedEnvelope);
        iIntEnd = iIntEnd(1);
    end

% If we find no intercepts, we need to extend both sides of characteristic
% average to intercept the envelop.
else
    [iIntStart,~] = rayxpoly(charAvg(2,:)',...
        (charAvg(1,:)-charAvg(2,:))', closedEnvelope);
    iIntStart = iIntStart(1);

    [iIntEnd,~] = rayxpoly(charAvg(end-1,:)',...
        (charAvg(end,:)-charAvg(end-1,:))', closedEnvelope);
    iIntEnd = iIntEnd(1);
end

% To divide inner or outer corridors, first determine if polygon is clockwise
% or counter-clockwise. Then, based on which index is large, separate out
% inner and outer corridor based on which intercept index is larger.
if ispolycw(envelope(:,1),envelope(:,2))
    if iIntStart > iIntEnd
        outerCorr = [envelope(iIntStart:end,:);envelope(1:iIntEnd,:)];
        innerCorr = envelope(iIntEnd:iIntStart,:);
    else
        outerCorr = envelope(iIntStart:iIntEnd,:);
        innerCorr = [envelope(iIntEnd:end,:);envelope(1:iIntStart,:)];
    end
else
    if iIntStart > iIntEnd
        innerCorr = [envelope(iIntStart:end,:);envelope(1:iIntEnd,:)];
        outerCorr = envelope(iIntEnd:iIntStart,:);
    else
        innerCorr = envelope(iIntStart:iIntEnd,:);
        outerCorr = [envelope(iIntEnd:end,:);envelope(1:iIntStart,:)];
    end
end

% Resample corridors. Use nResamplePoints. Because corridors are
% non-monotonic, arc-length method discussed above is used.
% Start with inner corridor. Magnitudes are being normalized.
segments = sqrt(((innerCorr(1:end-1,1)-innerCorr(2:end,1))./max(innerCorr(:,1))).^2 ...
    + ((innerCorr(1:end-1,2)-innerCorr(2:end,2))./max(innerCorr(:,2))).^2);

```

```

alen = cumsum([0;segments]);
alenResamp = linspace(0,max(alen),nvArg.nResamplePoints)';
innerCorr = [interp1(alen,innerCorr(:,1),alenResamp),...
    interp1(alen,innerCorr(:,2),alenResamp)];
% Outer Corridor
segments = sqrt(((outerCorr(1:end-1,1)-outerCorr(2:end,1))./max(outerCorr(:,1))).^2 ...
    + ((outerCorr(1:end-1,2)-outerCorr(2:end,2))./max(outerCorr(:,2))).^2);
alen = cumsum([0;segments]);
alenResamp = linspace(0,max(alen),nvArg.nResamplePoints)';
outerCorr = [interp1(alen,outerCorr(:,1),alenResamp),...
    interp1(alen,outerCorr(:,2),alenResamp)];

%% Add limits to detailed debug plot
if strcmpi(nvArg.Diagnostics,'detailed')
    xlim([min(xx(:)),max(xx(:))])
    ylim([min(yy(:)),max(yy(:))])
end

varargout{1} = inputSignals;
varargout{2} = debugOutput;
end % End main function

%% helper function to perform linear interpolation to an isovalue of 1 only
function val = interpVal(x1, y1, x2, y2)
    val = x1+(x2-x1)*(1-y1)/(y2-y1);
end

%% Function used to evaluate correlation score between signals
function [meanCorrScore, corrScoreArray] = evalCorrScore(signalsX,signalsY)
% Correlation score taken from the work of Nusholtz et al. (2009)
% Compute cross-correlation matrix of all signals to each other
corrMatX = corrcoef(signalsX);
corrMatY = corrcoef(signalsY);
% Convert matrices to a single score
nSignal = size(corrMatX,2);
corrScoreX = (1/(nSignal*(nSignal-1)))*(sum(sum(corrMatX))-nSignal);
corrScoreY = (1/(nSignal*(nSignal-1)))*(sum(sum(corrMatY))-nSignal);
% Compute a single metric for optimization purposes. Using simple mean
meanCorrScore = 0.5*(corrScoreX+corrScoreY);
corrScoreArray = [corrScoreX, corrScoreY];
end

%% Function used to compute objective for optimization
function [optScore, penaltyScore] =
warpingObjective(optimWarp,nCtrlPts,inputSignals,nvArg)
% Control points are equally spaced in arc-length.
% optimwarp is a column vector with first warped control point in the
% first nSignal indices, then 2nd control point in the next nSignal indices

% warpArray = reshape(optimWarp,length(inputSignals),nCtrlPts);
nSignal = length(inputSignals);
warpArray = reshape(optimWarp,[],nCtrlPts);
% Compute a warping penalty
penaltyScore = warpingPenalty(warpArray,nvArg.WarpingPenalty,nvArg);
penaltyScore = mean(penaltyScore);

% Perform warping - non-mex version
% [~, signalsX, signalsY] = warpArcLength(warpArray,inputSignals,...
%     nvArg.nResamplePoints);

% IMPORTANT: This is a compiled mex version of warpArcLength. The mex
% function cannot be modified. If warpArcLength is updated later, you will
% also need to recompile the mex function
signalCellArray = cell(nSignal,1);
for i=1:nSignal
    signalCellArray{i} = inputSignals(i).data;
end
[~, signalsX, signalsY] =
warpArcLength_mex(warpArray,signalCellArray,nvArg.nResamplePoints);

```

```
% Compute correlation score
[corrScore, ~] = evalCorrScore(signalsX,signalsY);
% corrScore is a maximization goal. Turn into a minimization goal
optScore = 1-corrScore+penaltyScore;

end

%% Function used to warp arc-length
function [warpedSignals, signalsX, signalsY]...
    = warpArcLength(warpArray, inputSignals, nResamplePoints)
% Warp array: each row is warping points for an input signal, each column
% is warped point. Control points are interpolated on [0,1] assuming
% equal spacing.
nSignals = length(inputSignals);

% lmCtrlPts = linspace(0,1,2+nCtrlPts);
% lmCtrlPts = [0,warpArray(end,:),1];

% Initialize matrices
signalsX = zeros(nResamplePoints, nSignals);
signalsY = zeros(nResamplePoints, nSignals);
warpedSignals = cell(nSignals,1);

for iSignal = 1:nSignals
    % Assign responseSignal data array to matrix for brevity
    signal = inputSignals(iSignal).data;

    lmCtrlPts = [0,warpArray(iSignal+nSignals,:),1];

    % prepend 0 and append 1 to warp points for this signal to create valid
    % control points.
    warpedCtrlPts = [0,warpArray(iSignal,:),1];

    % Construct warping function using SLM. This warps lmAlen to shiftAlen.
    % Use warping function to map computed arc-lengths onto the shifted
    % system. use built-in pchip function. This is a peicewise monotonic
    % cubic spline. Signifincantly faster than SLM.
    warpedNormAlen = pchip(lmCtrlPts,warpedCtrlPts,signal(:,4));

    % Now uniformly resample normalized arc-length
    resamNormwarpedAlen = linspace(0,1, nResamplePoints)';
    resampX = interp1(warpedNormAlen, signal(:,1),
resamNormwarpedAlen,'linear','extrap');
    resampY = interp1(warpedNormAlen, signal(:,2),
resamNormwarpedAlen,'linear','extrap');
    % Assign to array for correlation calc
    signalsX(:,iSignal) = resampX;
    signalsY(:,iSignal) = resampY;

    % Assemble a cell array containing arrays of resampled signals. Similar
    % to 'normalizedSignal' in 'inputSignals' structure
    warpedSignals{iSignal} = [resamNormwarpedAlen,resampX,resampY];
end

end

%% Penalty function to prevent plateaus and extreme divergence in warping functions
function [penaltyScores] = warpingPenalty(warpArray,penaltyFactor,nvArg)
% Compute an array of penalty scores based on MSE between linear, unwarped
% arc-length and warped arc-length. Aim is to help prevent plateauing.
[nSignals, nCtrlPts] = size(warpArray);
nSignals = nSignals/2;
% lmCtrlPts = [0, warpArray(end,:), 1];
penaltyScores = zeros(nSignals,1);
unwarpedAlen = linspace(0,1,nvArg.nResamplePoints);

for iSignal=1:nSignals
    penaltyScores(iSignal) = sum((unwarpedAlen - ...
```

```

        pchip([0,warpArray(iSignal+nSignals,:),1],...
        [0,warpArray(iSignal,:),1],unwarpedAlen).^2);
end

penaltyScores = penaltyScores.*penaltyFactor;
end

%% Function to find intercept of Ray and Polygon
function [indices, intercepts] = rayxpoly(basePt, dirVec, poly)
% Finds the intersections of a ray and polygon by incrementally solving the
% ray-line segment problem.
%
% Algorithm: rootllama.wordpress.com/2014/06/20/ray-line-segment-intersection-test-in-
2d/
%
% basePt and dirVec are [2,1] vectors. Poly is a list of vertices in a
% closed polygon
%
% If multiple intercepts are found, they are sorted from closest to base
% point to furthest.

nVerts= size(poly,1)-1; % Closed polygon so legnth+1

indices = [];
intercepts = [];

% Cycle through line segments, and check if ray intercepts line segments
for iVert = 1:nVerts
    % a is first point, b is second point
    a = poly(iVert,:);
    b = poly(iVert+1, :);

    % Define three helper vectors
    v1 = basePt-a;
    v2 = b-a;
    v3 = [-dirVec(2), dirVec(1)];

    % t1 is parameter for ray
    t1 = (v2(1)*v1(2) - v1(1)*v2(2))/dot(v2, v3);
    % t2 is parameter for line segment
    t2 = dot(v1, v3)/dot(v2, v3);

    % Ray intercepts segment iff t1 is positive (forward ray projection)
    % and 0<t2<=1
    if ( (t1>0) && (t2>0) && (t2<=1) )
        % record first index of line segment and coordinates of intercept
        indices = [indices; iVert];
        intercepts = [intercepts; (a+(b-a)*t2)'];
    end
end

% If more than one intercept, sort them from closest to furthest
if size(indices,1) > 1
    [~, sortInd] = sort(vecnorm(intercepts-basePt',2,2));
    indices = indices(sortInd);
    intercepts = intercepts(sortInd,:);
end

end

```

2.1.2. PreProcessInputSignals.m

```
%% Pre-process Input Signals
%
% This script pre-processes input signals which will be subsequently used
% for to generate a characteristic average and response corridors.
% Preprocessing amalgamates several curves, ensures the validity of said
% curves, and saves the data into a single MATLAB data file for later use.
%
% This script is provided as part of ARCGen, which is released under a GNU
% GPL v3 license. No warranty or support is provided. The authors any
% responsibility for the validity, accuracy, or applicability of any
% results obtained from this code.
%
% Input signals must be saved in individual CSV files, with data saved in
% columns. The user can set which two column indices are used as the input
% x and y data.
%
% Corridor generation is performed in a separate script
%
% Corridor generation scripts requires that input data be organized using a
% structure array. The structure array must have two entries per response
% curve
%   + data: [n,2] array of x-y data
%   + specId: A character string used as a specimen identifier.
%
% This script has four options to specify "specId" in a programatic
% fashion. This is defined using "flagAlterSpecID".
%   + "No": "specId" is taken directly from the file name of the .csv
%   + "RemoveUnderscore": "specId" is the file name of the .csv with
%       underscores replaced with spaces
%   + "Sequential": "specId" is defined sequential with "sequentialBase"
%       used as a prefix
%   + "Manual": "specId" is defined using the cell array "manualSpecIds".
%       "manualSpecId" must be the same length as .csv file being
%       processed.
%
% Copyright (c) 2022 Devon C. Hartlen

%% Initialization
fclose all;
close all;
clear;
clc;

addpath('ThirdPartyFunctions') % Path to 3rd party functions

%% Select desired data files to be processed
% This is accomplished with a UI
inputFileNames = uipickfiles('Output','struct');
% correct file names
for iFile = 1:length(inputFileNames)
    [~,name,ext] = fileparts(inputFileNames(iFile).name);
    inputFileNames(iFile).specId = name;
end

%% Alter specimen ID from file name (if desired)
% Cases: 'No','RemoveUnderscore','Sequential','Manual'
flagAlterSpecID = 'No';
sequentialBase = 'ID';
manualSpecIds = {...
    'ID1';...
    'ID2';...
};
switch flagAlterSpecID
    case 'No'
        disp('Skipping rename')
    case 'RemoveUnderscore'
        for iFile = 1:length(inputFileNames)
```



```

        inputFileNames(iFile).specId = ...
            replace(inputFileNames(iFile).specId, '_(x)', '');
        inputFileNames(iFile).specId = ...
            replace(inputFileNames(iFile).specId, '_', ' ');
    end
    case 'Sequential'
        for iFile = 1:length(inputFileNames)
            inputFileNames(iFile).specId = ...
                [sequentialBase ' ' num2str(iFile,'%3d')];
        end
    case 'Manual'
        if length(manualSpecIds) ~= length(inputFileNames)
            error('Not enough manual IDs specified')
        else
            for iFile = 1:length(inputFileNames)
                inputFileNames(iFile).specId = manualSpecIds{iFile};
            end
        end
    end
end

%% Load response curves
% Specify x,y columns of datafile to be loaded
indicesCurves = [1,2];
inputSignals = struct([]); % initialization
for iFile = 1:length(inputFileNames)
    curveData = readmatrix(inputFileNames(iFile).name); % R2020a required
    inputSignals(iFile).specId = inputFileNames(iFile).specId;
    inputSignals(iFile).data = curveData(:,indicesCurves);
end

%% Save response curves to file
uisave({'inputSignals'})

```

2.1.3. Arcgen_executer.m

```
[charAvg, innerCorr, outerCorr, processedSignalData, debugData] = arcgen(inputSignals,  
'nWarpCtrlPts', 2);
```

3. MATLAB CODE LIBRARY

3.1. Method 1: B-Spline approximation and Principal Component Analysis of individual landmark trajectories

3.1.1. Spline_analysis_all_tests_together.m

```
% First index: subject
% Second index: trial
% Third index: landmark (1- head, 2-C4, 3-T1, 4-T4, 5-T8, 6-Hpoint)
clear all;

% List of test names
test_names = {'1679', '1684', '1689', '1694', '1767', '1858', '1865', '1869', '1873',
'1771', '1774', '1778', '1780', '1862'};
%test_names = {'1679', '1684', '1689', '1694', '1767', '1771', '1774', '1778', '1780',
'1858', '1862', '1865', '1869', '1873'};
num_test = length(test_names);

n = input('How many control points? ', 's');
n = str2num(n);
order = 3;

if (n < order)
    disp([' !!! Error: Choose n >= order=', num2str(order), ' !!!']);
    return;
end

for ctr_test = 1:num_test
    test_name = test_names{ctr_test};

    file_name = strcat('/Users/alicia/Desktop/mobios/Lopez-Valdes 2024- Comparison of
volunteers and PMHS/Data analysis w splines/data/', test_name, '.xlsx');

    % Head
    if strcmp(test_name, '1684')
        data{ctr_test,1}(:,1) = xlsread(file_name, 3, 'B106:B311'); %only 300 ms:
approximate time when the reflex response may begin to influence
        data{ctr_test,1}(:,2) = xlsread(file_name, 3, 'D106:D311'); %same
    else
        data{ctr_test,1}(:,1) = xlsread(file_name, 3, 'B106:B405'); %only 300 ms:
approximate time when the reflex response may begin to influence
        data{ctr_test,1}(:,2) = xlsread(file_name, 3, 'D106:D405'); %same
    end
    %data{ctr_test,1}(:,1) = xlsread(file_name, 3, 'B106:B405'); %only 300 ms:
approximate time when the reflex response may begin to influence
    %data{ctr_test,1}(:,2) = xlsread(file_name, 3, 'D106:D405'); %same

    % C4
    data{ctr_test,2}(:,1) = xlsread(file_name, 3, 'H106:H405'); %only 300 ms:
approximate time when the reflex response may begin to influence
    data{ctr_test,2}(:,2) = xlsread(file_name, 3, 'J106:J405'); %same

    % T1
    if strcmp(test_name, '1865')
        data{ctr_test,3}(:,1) = xlsread(file_name, 3, 'K106:K260');
        data{ctr_test,3}(:,2) = xlsread(file_name, 3, 'M106:M260');
    else
        data{ctr_test,3}(:,1) = xlsread(file_name, 3, 'K106:K405');
        data{ctr_test,3}(:,2) = xlsread(file_name, 3, 'M106:M405');
    end

    % T1
    %data{ctr_test,3}(:,1) = xlsread(file_name, 3, 'K106:K405'); %only 300 ms:
approximate time when the reflex response may begin to influence
    %data{ctr_test,3}(:,2) = xlsread(file_name, 3, 'M106:M405'); %same
```

```
% T4
data{ctr_test,4}(:,1) = xlsread(file_name, 3, 'N106:N405'); %only 300 ms:
approximate time when the reflex response may begin to influence
data{ctr_test,4}(:,2) = xlsread(file_name, 3, 'P106:P405'); %same

% T8
data{ctr_test,5}(:,1) = xlsread(file_name, 3, 'Q106:Q405'); %only 300 ms:
approximate time when the reflex response may begin to influence
data{ctr_test,5}(:,2) = xlsread(file_name, 3, 'S106:S405'); %same

% H-point
data{ctr_test,6}(:,1) = xlsread(file_name, 3, 'E106:E405'); %only 300 ms:
approximate time when the reflex response may begin to influence
data{ctr_test,6}(:,2) = xlsread(file_name, 3, 'G106:G405'); %same

% Zeroing displacements
for ctr_var = 1:6
    data{ctr_test,ctr_var}(:,1) = data{ctr_test,ctr_var}(:,1) -
data{ctr_test,ctr_var}(1,1);
    data{ctr_test,ctr_var}(:,2) = data{ctr_test,ctr_var}(:,2) -
data{ctr_test,ctr_var}(1,2);
end

for ctr_var = 1:6
    if strcmp(test_name, '1865') && ctr_var == 3
        cut = 0;
    elseif strcmp(test_name, '1684') && ctr_var == 1
        cut = 6;
    else
        cut = 100;
    end

    interval = floor((length(data{ctr_test,ctr_var}) - cut) / (n));

    % 100 is to stop at t=200 ms

    fprintf('El valor de interval es: %d\n', interval);

    %fig{ctr_var} = figure;
    x(1) = data{ctr_test,ctr_var}(1,1);
    x(n) = data{ctr_test,ctr_var}(end-cut,1);
    z(1) = data{ctr_test,ctr_var}(1,2);
    z(n) = data{ctr_test,ctr_var}(end-cut,2);

    for i = 2:n-1
        x(i) = data{ctr_test,ctr_var}((i-1) * interval, 1);
        z(i) = data{ctr_test,ctr_var}((i-1) * interval, 2);
    end

    for i = 1:n
        p{ctr_test,ctr_var}(i,:) = [x(i); z(i)];
    end

    %{
    plot(p{ctr_test,ctr_var}(:,1), p{ctr_test,ctr_var}(:,2), 'k-', 'LineWidth', 2);
    hold on; box on;
    plot(p{ctr_test,ctr_var}(:,1), p{ctr_test,ctr_var}(:,2), 'ro', 'MarkerSize', 5,
'MarkerFaceColor', 'r');
    %}

    % Method 1
    T = linspace(0, 1, i - order + 2);

    if strcmp(test_name, '1865') && ctr_var==3
        y = linspace(0, 1, 156);
```

```

else
    y = linspace(0, 1, 201);
end

p_spl{ctr_test,ctr_var} = DEBOOR(T, p{ctr_test,ctr_var}, y, order);
% Method 2
% y = linspace(0, 201,

%{
    plot(p_spl{ctr_test,ctr_var}(:,1), p_spl{ctr_test,ctr_var}(:,2), 'b-',
'LineWidth', 2);
    hold off;
%}

% Assess of RMS error
% Evaluation of spline at all X positions
RMSE{ctr_test,ctr_var} = 0;
% Ensure lengths match for RMSE calculation
min_length = min(length(p_spl{ctr_test,ctr_var}(:,1)),
length(data{ctr_test,ctr_var}(:,1)));
aux = zeros(1, min_length);
% k{ctr_test,ctr_var} = 0;
%for ctr = 1:length(p_spl{ctr_test,ctr_var}(:,1))
for ctr = 1:min_length
    % if floor(p_spl(ctr,1) - data{ctr_test,ctr_var}(ctr+99,1)) < 1
    aux(ctr) = (p_spl{ctr_test,ctr_var}(ctr,2) -
data{ctr_test,ctr_var}(ctr,2))^2;
    RMSE{ctr_test,ctr_var} = RMSE{ctr_test,ctr_var} + aux(ctr);
    % k{ctr_test,ctr_var} = k{ctr_test,ctr_var} + 1;
    % else
    % aux = 0;
end
RMSE{ctr_test,ctr_var} = ((1 / length(aux(ctr))) * RMSE{ctr_test,ctr_var})^0.5;

%% RMSE calculation with evenly spaced points
RMSE_spaced{ctr_test,ctr_var} = 0;
% Number of points including first and last
n_points = 10;

% Ensure sample_indices are within bounds
max_idx = length(p_spl{ctr_test,ctr_var});
sample_indices = round(linspace(1, max_idx, n_points));

aux_spaced = zeros(1, length(sample_indices));
valid_count = 0; % Counter to track valid indices

for ctr = 1:length(sample_indices)
    idx = sample_indices(ctr);
    if idx > max_idx
        idx = max_idx;
    end
    aux_spaced(ctr) = (p_spl{ctr_test,ctr_var}(idx,2) -
data{ctr_test,ctr_var}(idx,2))^2;
    RMSE_spaced{ctr_test,ctr_var} = RMSE_spaced{ctr_test,ctr_var} +
aux_spaced(ctr);
end
RMSE_spaced{ctr_test,ctr_var} = sqrt(mean(aux_spaced));
end

%% Create a figure to display original data and spline curves

landmark_names = {'head', 'C4', 'T1', 'T4', 'T8', 'Hpoint'};
figure;

% Loop over each variable (landmark)
for ctr_var = 1:6
    % Plot original data points
    subplot(2, 3, ctr_var);

```

```

        plot(data{ctr_test, ctr_var}(:, 1), data{ctr_test, ctr_var}(:, 2), 'ro',
'MarkerSize', 5, 'MarkerFaceColor', 'r');
        hold on;

        % Plot spline curve
        plot(p_spl{ctr_test, ctr_var}(:, 1), p_spl{ctr_test, ctr_var}(:, 2), 'b-',
'LineWidth', 2);

        % Plot chosen RMSE points
        sample_indices = round(linspace(1, length(p_spl{ctr_test, ctr_var}), n_points));
        scatter(p_spl{ctr_test, ctr_var}(sample_indices, 1), p_spl{ctr_test,
ctr_var}(sample_indices, 2), ...
        'c', 'filled');

        scatter(data{ctr_test, ctr_var}(sample_indices, 1), data{ctr_test,
ctr_var}(sample_indices, 2), ...
        'g', 'filled');

        % Set plot title
        title(sprintf('Variable: %s', landmark_names{ctr_var}));
        xlabel('X-coordinate');
        ylabel('Y-coordinate');
        legend('Original Data', 'Spline Curve');
        grid on;
    end

    % Adjust figure layout
    figure_title = sprintf('Test: %s. Comparison of Original Data and Spline Curves for
Each Landmark', test_names{ctr_test});
    sgtitle(figure_title);

end

%% Create p_eld

% Initialize p_eld as an empty cell array
p_eld = cell(5, 6); % Assuming 5 tests for older people

% Move data from rows 10 to 14 of p to p_eld
p_eld = p(10:14, :);

% Clear rows 10 to 14 of p
p(10:14, :) = [];

%% Analyse RMSE

max_RMSE = max(cellfun(@max, RMSE_spaced));
fprintf('Maximum RMSE Head for %d RMSE points: %.4f\n', n_points, max_RMSE(1));
average_RMSE = mean(cellfun(@mean, RMSE_spaced));
fprintf('Average RMSE Head for %d RMSE points: %.4f\n', n_points, average_RMSE(1));

%% Calculate average and standard deviation of RMSE_spaced for each landmark

average_RMSE_spaced = zeros(1, 6);
stdev_RMSE_spaced = zeros(1, 6);

for ctr_var = 1:6
    % Extract RMSE_spaced values for the current landmark
    rmse_values = [];
    for ctr_test = 1:num_test
        rmse_values = [rmse_values, RMSE_spaced{ctr_test, ctr_var}];
    end
    average_RMSE_spaced(ctr_var) = mean(rmse_values);
    stdev_RMSE_spaced(ctr_var) = std(rmse_values);
end

T = table(landmark_names', average_RMSE_spaced', stdev_RMSE_spaced', ...

```

```
'VariableNames', {'Landmark', 'Average_RMSE_spaced', 'Stdev_RMSE_spaced'});  
disp(T);
```

3.1.2. All_coordinates_any_n.m

```
%% Chose landmark

disp('Landmark IDs are as follows:');

% Landmarks array
landmarks = {'1- head', '2- C4', '3- T1', '4- T4', '5- T8', '6- Hpoint'};

% Display each landmark on a new line
for i = 1:length(landmarks)
    fprintf('%s\n', landmarks{i});
end

l = input('Which landmark? ', 's');
l = str2num(l);

%% Young
for i = 1:9
    for j = 1:n % Loop through the number of control points
        X(i, (2*j)-1) = p{i, l}(j, 1); % X-coordinate
        X(i, 2*j) = p{i, l}(j, 2); % Y-coordinate
    end
end

%% Elderly
for i = 10:14
    for j = 1:n % Loop through the number of control points
        X(i, (2*j)-1) = p_eld{i-9, l}(j, 1); % X-coordinate
        X(i, 2*j) = p_eld{i-9, l}(j, 2); % Y-coordinate
    end
end
```


3.1.3. PCA_Analysis

```
% Standardize the Data
X_standardized = zscore(X);

% Perform PCA
[coeff, score, latent, tsquared, explained] = pca(X_standardized);

% Define Group Labels
groups = [ones(9, 1); 2*ones(5, 1)]; % 1 = Young, 2 = Elderly
groupNames = {'Young volunteers', 'Elderly volunteers'};
colors = ['r', 'b']; % red = young, blue = elderly

% Fonts and sizes
fontName = 'Helvetica';
fontSizeAxis = 12;
fontSizeLabel = 10;
markerSize = 50;

%% 2D PCA Plot: PC1 vs PC2
figure('Color', 'w', 'Units', 'centimeters', 'Position', [5, 5, 16, 12]);
t = tiledlayout(1,1, 'Padding', 'compact', 'TileSpacing', 'compact');
sgtitle('PCA Projection: 1st and 2nd PC - H-Point', ...
    'FontSize', fontSizeAxis + 2, 'FontWeight', 'bold');
ax = nexttile;

hold on;
for i = 1:max(groups)
    idx = groups == i;
    scatter(score(idx,1), score(idx,2), markerSize, colors(i), 'filled', 'DisplayName',
        groupNames{i});
    for j = find(idx)
        text(score(j,1), score(j,2), sprintf('%d', j), ...
            'VerticalAlignment','bottom', 'HorizontalAlignment','right', ...
            'FontName', fontName, 'FontSize', fontSizeLabel);
    end
end
xlabel('First Principal Component', 'FontName', fontName, 'FontSize', fontSizeAxis);
ylabel('Second Principal Component', 'FontName', fontName, 'FontSize', fontSizeAxis);
set(gca, 'FontName', fontName, 'FontSize', fontSizeAxis);
legend('Location', 'bestoutside', 'FontName', fontName, 'FontSize', fontSizeAxis);
grid on;
hold off;

%% 3D PCA Plot: PC1 vs PC2 vs PC3
figure('Color', 'w', 'Units', 'centimeters', 'Position', [5, 5, 16, 12]);
t = tiledlayout(1,1, 'Padding', 'compact', 'TileSpacing', 'compact');
sgtitle('PCA Projection: 1st, 2nd and 3rd PC - T1', ...
    'FontName', 'Helvetica', 'FontSize', fontSizeAxis + 2, 'FontWeight', 'bold');

ax3 = nexttile;
hold on;

for i = 1:max(groups)
    idx = groups == i;
    scatter3(score(idx,1), score(idx,2), score(idx,3), markerSize, colors(i), 'filled',
        'DisplayName', groupNames{i});
    for j = find(idx)
        text(score(j,1), score(j,2), score(j,3), sprintf('%d', j), ...
            'VerticalAlignment','bottom', 'HorizontalAlignment','right', ...
            'FontName', 'Helvetica', 'FontSize', fontSizeLabel);
    end
end
xlabel('First Principal Component', 'FontName', 'Helvetica', 'FontSize', fontSizeAxis);
ylabel('Second Principal Component', 'FontName', 'Helvetica', 'FontSize', fontSizeAxis);
zlabel('Third Principal Component', 'FontName', 'Helvetica', 'FontSize', fontSizeAxis);
set(gca, 'FontName', 'Helvetica', 'FontSize', fontSizeAxis);
legend('Location', 'bestoutside', 'FontName', 'Helvetica', 'FontSize', fontSizeAxis);
grid on;
```

```
view(3);  
hold off;  
  
%% Display explained variance  
ExplainedVariance = table( ...  
    (1:3)', ...  
    explained(1:3), ...  
    'VariableNames', {'PrincipalComponent', 'VarianceExplained_Percent'});  
disp(ExplainedVariance);
```

3.2. Method 2: Landmark relative positioning analysis at peak displacement

3.2.1. Time_based_noHPoint.m

```
clear all;

%% UPLOAD DATA
% List of test names
test_names = {'1679', '1684', '1689', '1694', '1767', '1858', '1865', '1869', '1873',
'1771', '1774', '1778', '1780', '1862'};
num_test = length(test_names);
variable_names = {'Head', 'C4', 'T1', 'T4', 'T8'}; % Removed H-point

% Prompt the user to enter the number of evenly spaced points
num_points = input('Enter the number of evenly spaced points: ');

% Calculate indices for evenly spaced points
indices_longest = round(linspace(1, 200, num_points));

for ctr_test = 1:num_test
    test_name = test_names{ctr_test};

    file_name = strcat('/Users/alicia/Desktop/mobios/Lopez-Valdes 2024- Comparison of
volunteers and PMHS/Data analysis w splines/data/', test_name, '.xlsx');

    % Head
    data{ctr_test,1}(:,1) = xlsread(file_name, 3, 'B106:B305');
    data{ctr_test,1}(:,2) = xlsread(file_name, 3, 'D106:D305');

    % C4
    data{ctr_test,2}(:,1) = xlsread(file_name, 3, 'H106:H305');
    data{ctr_test,2}(:,2) = xlsread(file_name, 3, 'J106:J305');

    % T1
    if strcmp(test_name, '1865')
        data{ctr_test,3}(:,1) = xlsread(file_name, 3, 'K106:K260');
        data{ctr_test,3}(:,2) = xlsread(file_name, 3, 'M106:M260');
    else
        data{ctr_test,3}(:,1) = xlsread(file_name, 3, 'K106:K305');
        data{ctr_test,3}(:,2) = xlsread(file_name, 3, 'M106:M305');
    end

    % T4
    data{ctr_test,4}(:,1) = xlsread(file_name, 3, 'N106:N305');
    data{ctr_test,4}(:,2) = xlsread(file_name, 3, 'P106:P305');

    % T8
    data{ctr_test,5}(:,1) = xlsread(file_name, 3, 'Q106:Q305');
    data{ctr_test,5}(:,2) = xlsread(file_name, 3, 'S106:S305');

    % Zeroing displacements and select evenly spaced data points
    head_initial_coords = data{ctr_test,1}(1, :); % Get initial coordinates of the head
    for ctr_var = 1:5 % Adjust loop to exclude H-point
        data{ctr_test, ctr_var} = data{ctr_test, ctr_var} - head_initial_coords; %
Zeroing relative to the head's initial position

        num_data_points = size(data{ctr_test,ctr_var}, 1);
        valid_indices = indices_longest(indices_longest <= num_data_points);
        selected_data{ctr_test, ctr_var} = data{ctr_test, ctr_var}(valid_indices, :);
    end
end

%% STORE SELECTED VALUES IN selected_values_all_tests MATRIX

% Number of variables (body parts)
num_vars = 5; % Adjusted to exclude H-point
```

```
% Initialize cell array to store selected values for each test
selected_values_all_tests = cell(num_test, 1);

for ctr_test = 1:num_test
    % Initialize a matrix to store the selected values for the current test
    num_selected_points = size(selected_data{ctr_test, 1}, 1);
    selected_values = NaN(num_selected_points, num_vars * 2); % 2 columns for each
    variable (x and y)

    % Iterate over each variable
    for ctr_var = 1:num_vars
        % Extract the selected points for the current variable
        selected_points = selected_data{ctr_test, ctr_var};

        % Store the selected points in the matrix
        if ~isempty(selected_points)
            num_points = size(selected_points, 1);
            selected_values(1:num_points, (ctr_var - 1) * 2 + 1:ctr_var * 2) =
selected_points;
        end
    end

    % Store the selected values for the current test
    selected_values_all_tests{ctr_test} = selected_values;
end

%% PLOT ALL TRAJECTORIES FOR ALL TESTS

% Prompt the user to enter the test number for plotting
test_selection = input('Enter the test number to plot (1 to 14) or "all": ', 's');

if strcmpi(test_selection, 'all')
    num_plots = num_test;
else
    num_plots = 1;
    ctr_test = str2double(test_selection);
end

% Setup figure for multiple subplots if needed
if num_plots > 1
    figure;
    hold on;
end

for i = 1:num_plots
    if num_plots > 1
        ctr_test = i;
        subplot(ceil(sqrt(num_plots)), ceil(sqrt(num_plots)), i);
    end

    test_data = selected_values_all_tests{ctr_test};

    % Find the index of the point with the highest x-coordinate for the head (variable
1)
    [~, max_idx] = max(test_data(:, 1));

    % Plotting
    if num_plots == 1
        figure;
    end
    hold on;

    colors = lines(5); % Generate distinct colors for each variable

    % Initialize plot handles
    plot_handles = gobjects(1, 5);

    for ctr_var = 1:5 % Adjust loop to exclude H-point
```

```
% Plot the entire dataset
plot_handles(ctr_var) = plot(data{ctr_test, ctr_var}(:, 1), data{ctr_test,
ctr_var}(:, 2), 'Color', colors(ctr_var, :));

% Highlight the selected points
plot(selected_data{ctr_test, ctr_var}(:, 1), selected_data{ctr_test, ctr_var}(:,
2), 'o', 'Color', colors(ctr_var, :), 'MarkerFaceColor', colors(ctr_var, :));
end

for frame_idx = 1:num_points
% Plot segments for current time frame
% Initialize arrays to store x and z coordinates
x_coords = zeros(1, num_vars);
z_coords = zeros(1, num_vars);

% Extract x and z coordinates for current time frame
for ctr_var = 1:num_vars
x_coords(ctr_var) = test_data(frame_idx, 2*ctr_var - 1); % x coordinate
z_coords(ctr_var) = test_data(frame_idx, 2*ctr_var); % z coordinate
end

% Plot segments connecting consecutive variable points
for ctr_var = 1:num_vars - 1
x_segment = [x_coords(ctr_var), x_coords(ctr_var + 1)];
z_segment = [z_coords(ctr_var), z_coords(ctr_var + 1)];

% Determine the color based on the segment position
if frame_idx < max_idx
color = [0.5, 0.5, 0.5]; % Grey
linewidth = 1.5;
elseif frame_idx == max_idx
color = [0, 0, 0]; % Black
linewidth = 2;
else
color = [0.8, 0.8, 0.8]; % Light grey
linewidth = 1;
end

plot(x_segment, z_segment, '-o', 'Color', color, 'LineWidth', linewidth);
end
end

% Customize the plot
title(['Data and Selected Points for Test ' test_names{ctr_test}]);
xlabel('X Displacement');
ylabel('Z Displacement');
grid on;

% Only display legend for the last plot
if num_plots == 1 || i == num_plots
legend(plot_handles, variable_names); % Corrected legend
end

if num_plots == 1
hold off;
end
end

if num_plots > 1
hold off;
end

%% PLOT LONGEST TRAJECTORIES CHOSEN ALIGNMENT

% Prompt user to choose variable to align
disp('Landmark IDs:');
landmarks = {'1- head', '2- C4', '3- T1', '4- T4', '5- T8'};
for i = 1:length(landmarks)
fprintf('%s\n', landmarks{i});
end
```

```

end
align_var_index = input('Enter the index of the variable to align: ');

% Create a figure for plotting all trajectories
figure;
hold on;

plot_handles = gobjects(1, 14);

% Loop over each test
for ctr_test = 1:14
    test_data = selected_values_all_tests{ctr_test};

    % Find the index of the point with the highest x-coordinate
    [~, max_idx] = max(test_data(:, 1));

    % Initialize arrays to store x and z coordinates
    x_coords = zeros(1, num_vars);
    z_coords = zeros(1, num_vars);

    % Extract x and z coordinates for the point with the highest x-coordinate
    for ctr_var = 1:num_vars
        x_coords(ctr_var) = test_data(max_idx, 2 * ctr_var - 1); % x coordinate
        z_coords(ctr_var) = test_data(max_idx, 2 * ctr_var);      % z coordinate
    end

    % Calculate initial coordinates based on user input
    align_initial_x = x_coords(align_var_index);
    align_initial_z = z_coords(align_var_index);

    % Adjust all variables relative to the chosen variable
    for ctr_var = 1:num_vars
        x_coords(ctr_var) = x_coords(ctr_var) - align_initial_x;
        z_coords(ctr_var) = z_coords(ctr_var) - align_initial_z;
    end

    % Plot segments connecting consecutive variable points
    for ctr_var = 1:num_vars - 1
        x_segment = [x_coords(ctr_var), x_coords(ctr_var + 1)];
        z_segment = [z_coords(ctr_var), z_coords(ctr_var + 1)];

        % Determine the color based on the test number
        if ctr_test < 10
            color = [1, 0, 0]; % red for test 1 to 9
        else
            color = [0, 0, 1]; % blue for test 10 to 14
        end

        % Plot the segment and store the plot handle
        plot_handles(ctr_test) = plot(x_segment, z_segment, '-o', 'Color', color,
'LineWidth', 1.5);
    end
end

% Customize the plot
landmark_name = variable_names{align_var_index};
title(sprintf('Largest Head Displacement Trajectories, aligned at %s', landmark_name));
xlabel('X Axis');
ylabel('Z Axis');
grid on;

% Create legend for plot handles
legend_entries = {'Young Volunteers', 'Elderly Volunteers'};
legend_colors = [plot(nan, nan, '-o', 'Color', [1, 0, 0], 'LineWidth', 2), ...
    plot(nan, nan, '-o', 'Color', [0, 0, 1], 'LineWidth', 2)];
legend(legend_colors, legend_entries);

% Release the hold on the current figure
hold off;

```

3.2.2. Pca_analysis_p2.m

```
%% STORE AND ZERO VALUES OF COORDINATES AT MAXIMUM HEAD DISPLACEMENT

% Initialize cell array to store max displacement values for each test
max_displacement_values_all_tests = cell(num_test, 1);

for ctr_test = 1:num_test
    test_data = selected_values_all_tests{ctr_test};

    % Find the index of the point with the highest x-coordinate for the head (variable
1)
    [~, max_idx] = max(test_data(:, 1));

    % Initialize matrix to store the coordinates at maximum head displacement
    max_displacement_values = NaN(1, num_vars * 2); % 2 columns for each variable (x and
y)

    % Extract x and z coordinates for the point with the highest x-coordinate
    for ctr_var = 1:num_vars
        max_displacement_values(1, (ctr_var - 1) * 2 + 1) = test_data(max_idx, (ctr_var
- 1) * 2 + 1); % x coordinate
        max_displacement_values(1, (ctr_var - 1) * 2 + 2) = test_data(max_idx, (ctr_var
- 1) * 2 + 2); % z coordinate
    end

    % Get the coordinates of the last variable (T8) at maximum displacement
    t8_x = max_displacement_values(1, (num_vars - 1) * 2 + 1);
    t8_z = max_displacement_values(1, (num_vars - 1) * 2 + 2);

    % Zero the coordinates relative to T8
    for ctr_var = 1:num_vars
        max_displacement_values(1, (ctr_var - 1) * 2 + 1) = max_displacement_values(1,
(ctr_var - 1) * 2 + 1) - t8_x; % zeroed x coordinate
        max_displacement_values(1, (ctr_var - 1) * 2 + 2) = max_displacement_values(1,
(ctr_var - 1) * 2 + 2) - t8_z; % zeroed z coordinate
    end

    % Store the zeroed max displacement values for the current test
    max_displacement_values_all_tests{ctr_test} = max_displacement_values;
end

%% Convert cell array to matrix
num_tests = length(max_displacement_values_all_tests);
num_features = num_vars * 2; % Each variable has x and z coordinates

% Initialize matrix
X = NaN(num_tests, num_features);

for ctr_test = 1:num_tests
    X(ctr_test, :) = max_displacement_values_all_tests{ctr_test};
end

% Standardize the Data
X_standardized = zscore(X);

% Perform PCA
[coeff, score, latent, tsquared, explained] = pca(X_standardized);

% Define Group Labels
groups = [ones(9, 1); 2*ones(5, 1)]; % 1 for Young volunteers, 2 for Elderly volunteers

% Set font and size
fontName = 'Times New Roman';
fontSizeAxis = 16;
fontSizeLabel = 14;

% 2D Plot for the first two principal components
```

```
figure;
hold on;
colors = ['r', 'b']; % Red for Young volunteers, Blue for Elderly volunteers
groupNames = {'Young volunteers', 'Elderly volunteers'};

% Plot each group with a loop
for i = 1:max(groups)
    idx = groups == i;
    scatter(score(idx,1), score(idx,2), colors(i), 'filled');

    % Label each point
    individuals = find(idx);
    for j = individuals
        text(score(j,1), score(j,2), sprintf('%d', j), 'VerticalAlignment','bottom',
            'HorizontalAlignment','right', 'FontName', fontName, 'FontSize', fontSizeLabel);
    end
end

% Customizing the 2D plot
set(gca, 'FontName', fontName, 'FontSize', fontSizeAxis);
xlabel('First Principal Component', 'FontName', fontName, 'FontSize', fontSizeAxis);
ylabel('Second Principal Component', 'FontName', fontName, 'FontSize', fontSizeAxis);
legend(groupNames, 'FontName', fontName, 'FontSize', fontSizeAxis);
hold off;

% 3D Plot for the first three principal components
figure;
hold on;
% Plot each group in 3D
for i = 1:max(groups)
    idx = groups == i;
    scatter3(score(idx,1), score(idx,2), score(idx,3), colors(i), 'filled');

    % Label each point
    individuals = find(idx);
    for j = individuals
        text(score(j,1), score(j,2), score(j,3), sprintf('%d', j),
            'VerticalAlignment','bottom', 'HorizontalAlignment','right', 'FontName', fontName,
            'FontSize', fontSizeLabel);
    end
end

% Customizing the 3D plot
set(gca, 'FontName', fontName, 'FontSize', fontSizeAxis);
xlabel('First Principal Component', 'FontName', fontName, 'FontSize', fontSizeAxis);
ylabel('Second Principal Component', 'FontName', fontName, 'FontSize', fontSizeAxis);
zlabel('Third Principal Component', 'FontName', fontName, 'FontSize', fontSizeAxis);
legend(groupNames, 'FontName', fontName, 'FontSize', fontSizeAxis);
grid on; % Ensure the grid is on for the 3D plot
view(3); % Ensure the view is in 3D
hold off;

% Displaying the variance explained by the first three components
fprintf('Variance explained by the first principal component: %.2f%%\n', explained(1));
fprintf('Variance explained by the second principal component: %.2f%%\n', explained(2));
fprintf('Variance explained by the third principal component: %.2f%%\n', explained(3));
```


3.2.3. Relative_positions_analysis.m

```
% Initialize Variables for Storing Data
variable_distances = cell(1, num_vars - 1); % Initialize cell array to store distances
p_values = NaN(1, num_vars - 1); % Initialize array to store p-values

% Calculate Distances for each pair of variables
for pair_idx = 1:num_vars - 1
    distances = NaN(num_test, 1); % Initialize array to store distances for current pair

    for ctr_test = 1:num_test
        test_data = selected_values_all_tests(ctr_test);

        % Find the index of the point with the highest x-coordinate for the head
        (variable 1)
        [~, max_idx] = max(test_data(:, 1));

        % Extract x and z coordinates for the current pair of variables at the frame of
        maximum head x-coordinate
        var1_x = test_data(max_idx, 2 * pair_idx - 1); % x-coordinate of variable 1
        var1_z = test_data(max_idx, 2 * pair_idx); % z-coordinate of variable 1

        var2_x = test_data(max_idx, 2 * pair_idx + 1); % x-coordinate of variable 2
        var2_z = test_data(max_idx, 2 * pair_idx + 2); % z-coordinate of variable 2

        % Calculate Euclidean distance between variable 1 and variable 2 at the
        identified frame
        distance_var1_var2 = sqrt((var1_x - var2_x)^2 + (var1_z - var2_z)^2);

        % Store the distance
        distances(ctr_test) = distance_var1_var2;
    end

    % Store distances for the current pair
    variable_distances{pair_idx} = distances;

    % Perform t-test to compare means for the current pair
    distances_young = distances(1:9);
    distances_elderly = distances(10:end);
    [~, p, ~, ~] = ttest2(distances_young, distances_elderly);

    % Store p-value for the current pair
    p_values(pair_idx) = p;
end

%% Compare Distances

% Display distances for each pair of variables
fprintf('Distances between consecutive variables at frame of maximum head
displacement:\n');
fprintf('%-20s %-30s %-30s %-20s\n', 'Variable Pair', 'Mean ± Std Dev (Young)', 'Mean ±
Std Dev (Elderly)', 'p-value');

for pair_idx = 1:num_vars - 1
    % Separate distances for young and elderly volunteers for the current pair
    distances_young = variable_distances{pair_idx}(1:9);
    distances_elderly = variable_distances{pair_idx}(10:end);

    % Calculate statistics for the current pair
    mean_distance_y = mean(distances_young);
    std_dev_distance_y = std(distances_young);
    mean_distance_e = mean(distances_elderly);
    std_dev_distance_e = std(distances_elderly);

    % Display results in table format
    fprintf('%-20s %-30s %-30s %.4f\n', ...
        sprintf('%s & %s', variable_names{pair_idx}, variable_names{pair_idx + 1}),
        ...
```

```

        sprintf('%.2f ± %.2f', mean_distance_y, std_dev_distance_y), ...
        sprintf('%.2f ± %.2f', mean_distance_e, std_dev_distance_e), ...
        p_values(pair_idx));
end

%% Initialize storage for angles at C4, T1, and T4
angles_all_tests = NaN(num_test, 3); % Columns: [C4, T1, T4]

for ctr_test = 1:num_test
    test_data = selected_values_all_tests{ctr_test};

    % Find MHD frame (max head x-displacement)
    [~, max_idx] = max(test_data(:, 1));

    % Extract (x, z) coordinates of landmarks at MHD
    coords = zeros(num_vars, 2); % Rows = landmarks, Cols = [x z]
    for i = 1:num_vars
        coords(i, :) = test_data(max_idx, (2*i-1):(2*i));
    end

    % Compute internal angles
    angle_C4 = compute_angle(coords(1,:), coords(2,:), coords(3,:)); % Head-C4-T1
    angle_T1 = compute_angle(coords(2,:), coords(3,:), coords(4,:)); % C4-T1-T4
    angle_T4 = compute_angle(coords(3,:), coords(4,:), coords(5,:)); % T1-T4-T8

    % Store results
    angles_all_tests(ctr_test, :) = [angle_C4, angle_T1, angle_T4];
end

function ang = compute_angle(A, B, C)
    % Returns angle in degrees at point B
    BA = A - B;
    BC = C - B;
    ang = acosd(dot(BA, BC) / (norm(BA) * norm(BC)));
end

```

3.3. Method 3: Arc-Length Re-Parameterization and Signal Registration for generation of average trajectories and deviation corridors

3.3.1. generate_csvs_from_excels.m

```
% Clear all variables
clear all;

% List of test names
test_names = {'1679', '1684', '1689', '1694', '1767', '1858', '1865', '1869', '1873',
'1771', '1774', '1778', '1780', '1862'};
num_test = length(test_names);

% Initialize data cell array
data = cell(num_test, 6);

% Populate data cell array from Excel files
for ctr_test = 1:num_test
    test_name = test_names{ctr_test};

    file_name = strcat('/Users/alicia/Desktop/mobios/Lopez-Valdes 2024- Comparison of
volunteers and PMHS/Data analysis w splines/data/', test_name, '.xlsx');

    % Head
    if strcmp(test_name, '1684')
        data{ctr_test,1}(:,1) = xlsread(file_name, 3, 'B106:B311'); %only 300 ms:
approximate time when the reflex response may begin to influence
        data{ctr_test,1}(:,2) = xlsread(file_name, 3, 'D106:D311'); %same
    else
        data{ctr_test,1}(:,1) = xlsread(file_name, 3, 'B106:B405'); %only 300 ms:
approximate time when the reflex response may begin to influence
        data{ctr_test,1}(:,2) = xlsread(file_name, 3, 'D106:D405'); %same
    end

    % C4
    data{ctr_test,2}(:,1) = xlsread(file_name, 3, 'H106:H405'); %only 300 ms:
approximate time when the reflex response may begin to influence
    data{ctr_test,2}(:,2) = xlsread(file_name, 3, 'J106:J405'); %same

    % T1
    if strcmp(test_name, '1865')
        data{ctr_test,3}(:,1) = xlsread(file_name, 3, 'K106:K260');
        data{ctr_test,3}(:,2) = xlsread(file_name, 3, 'M106:M260');
    else
        data{ctr_test,3}(:,1) = xlsread(file_name, 3, 'K106:K405');
        data{ctr_test,3}(:,2) = xlsread(file_name, 3, 'M106:M405');
    end

    % T4
    data{ctr_test,4}(:,1) = xlsread(file_name, 3, 'N106:N405'); %only 300 ms:
approximate time when the reflex response may begin to influence
    data{ctr_test,4}(:,2) = xlsread(file_name, 3, 'P106:P405'); %same

    % T8
    data{ctr_test,5}(:,1) = xlsread(file_name, 3, 'Q106:Q405'); %only 300 ms:
approximate time when the reflex response may begin to influence
    data{ctr_test,5}(:,2) = xlsread(file_name, 3, 'S106:S405'); %same

    % H-point
    data{ctr_test,6}(:,1) = xlsread(file_name, 3, 'E106:E405'); %only 300 ms:
approximate time when the reflex response may begin to influence
    data{ctr_test,6}(:,2) = xlsread(file_name, 3, 'G106:G405'); %same
end

% Directory to save the CSV files
```

```
outputDir = '/Users/alicia/Desktop/mobios/Lopez-Valdes 2024- Comparison of volunteers  
and PMHS/ARCGen/csv_data';  
if ~exist(outputDir, 'dir')  
    mkdir(outputDir);  
end  
  
% Variable names corresponding to each column  
variableNames = {'Head', 'C4', 'T1', 'T4', 'T8', 'H_point'};  
  
% Loop through each test and variable to save data as CSV  
for ctr_test = 1:size(data, 1)  
    for variable = 1:size(data, 2)  
        % Extract data for current test and variable  
        currentData = data{ctr_test, variable};  
  
        % Create a file name  
        fileName = fullfile(outputDir, sprintf('test%s_%s.csv', test_names{ctr_test},  
variableNames{variable}));  
  
        % Write data to CSV file  
        writematrix(currentData, fileName);  
  
        % Display a message  
        fprintf('Data from test %s (%s) written to %s\n', test_names{ctr_test},  
variableNames{variable}, fileName);  
    end  
end
```

3.3.2. Arcgen_executer_comparison.m

```
% Request user input for the landmark
landmarkNames = {'Head', 'C4', 'T1', 'T4', 'T8', 'H Point'};
prompt = 'Enter the landmark number (1 - Head, 2 - C4, 3 - T1, 4 - T4, 5 - T8, 6 - H
Point): ';
landmarkIndex = input(prompt);
% Validate the input
if landmarkIndex < 1 || landmarkIndex > 6
    error('Invalid input. Please enter a number between 1 and 6.');
```

```
end
landmarkName = landmarkNames{landmarkIndex};
%% Zero displacements
% Define the number of rows to keep (200 ms)
numRowsToKeep = 180;
% Split the inputSignals into young and elderly groups
inputSignalsYoung = inputSignals(1:9);
inputSignalsElderly = inputSignals(10:14);
% Create temporary variables to store modified signals for both groups
tempSignalsYoung = inputSignalsYoung;
tempSignalsElderly = inputSignalsElderly;
% Process the "young" group of input signals
for i = 1:length(tempSignalsYoung)
    % Get the current signal data
    currentData = tempSignalsYoung(i).data;

    % Zero the displacements to the first value
    zeroedData = currentData - currentData(1, :);

    % Keep only the first 200 rows
    choppedData = zeroedData(1:numRowsToKeep, :);

    % Update the data field in tempSignalsYoung with the zeroed and chopped data
    tempSignalsYoung(i).data = choppedData;
end
% Process the "elderly" group of input signals
for i = 1:length(tempSignalsElderly)
    % Get the current signal data
    currentData = tempSignalsElderly(i).data;

    % Zero the displacements to the first value
    zeroedData = currentData - currentData(1, :);

    % Keep only the first 200 rows
    choppedData = zeroedData(1:numRowsToKeep, :);

    % Update the data field in tempSignalsElderly with the zeroed and chopped data
    tempSignalsElderly(i).data = choppedData;
end
%% Execute arcgen for both groups
[charAvgNormYoung, innCorrNormYoung, outCorrNormYoung, proCurveDataNormYoung] = ...
    arcgen(tempSignalsYoung,...
        'Diagnostics', 'off', ...
        'nWarpCtrlPts', 2,...
        'warpingPenalty', 1e-2);
[charAvgNormElderly, innCorrNormElderly, outCorrNormElderly, proCurveDataNormElderly] =
    ...
    arcgen(tempSignalsElderly,...
        'Diagnostics', 'off', ...
        'nWarpCtrlPts', 2,...
        'warpingPenalty', 1e-2);
%% Plot input signals with ARCGen corridors for both groups
figure('Name','Normalization', 'Color','w', 'Units','centimeters',
    'Position',[5,5,16,12]);hold on;
title(['Average and Corridors for Landmark ', landmarkName], ...
    'FontName','Helvetica', 'FontSize',10, 'FontWeight','bold');
xlabel('X Displacement (mm)', 'FontName','Helvetica', 'FontSize',12);
ylabel('Z Displacement (mm)', 'FontName','Helvetica', 'FontSize',12);
% Colors for different groups
```

```

colorYoungSignal = [1, 0.75, 0.8]; % Light pink
colorYoungAvg = [1, 0, 0]; % Red
colorYoungCorr = [1, 0.75, 0.8]; % Light pink
colorElderlySignal = [0.5, 0.5, 1]; % Light blue
colorElderlyAvg = [0, 0, 0.5]; % Dark blue
colorElderlyCorr = [0.5, 0.5, 1]; % Light blue
% Plot signals for the "young" group
for iPlot = 1:length(tempSignalsYoung)
    plot(tempSignalsYoung(iPlot).data(:,1),...
        tempSignalsYoung(iPlot).data(:,2),...
        'Color',colorYoungSignal,...
        'LineWidth',0.5);
end
% Plot ARCGen average and corridors for the "young" group
% Plot ARCGen average and corridors for the "young" group
hAvgYoung = plot(charAvgNormYoung(:,1), charAvgNormYoung(:,2), '-.',...
    'DisplayName','Char. Avg. Young','MarkerSize',5,...
    'LineWidth',2.5,'Color',colorYoungAvg);
hCorrInnerYoung = plot(innCorrNormYoung(:,1), innCorrNormYoung(:,2), '-.'
    ', 'MarkerSize',10,...
    'DisplayName','Corridors Young',...
    'LineWidth',1.5,'Color',colorYoungCorr);
hCorrOuterYoung = plot(outCorrNormYoung(:,1), outCorrNormYoung(:,2), '-.'
    ', 'MarkerSize',10,...
    'DisplayName','Outer Young',...
    'LineWidth',1.5,'Color',colorYoungCorr);
% Plot signals for the "elderly" group
for iPlot = 1:length(tempSignalsElderly)
    plot(tempSignalsElderly(iPlot).data(:,1),...
        tempSignalsElderly(iPlot).data(:,2),...
        'Color',colorElderlySignal,...
        'LineWidth',0.5);
end
% Plot ARCGen average and corridors for the "elderly" group
% Plot ARCGen average and corridors for the "elderly" group
hAvgElderly = plot(charAvgNormElderly(:,1), charAvgNormElderly(:,2), '-.',...
    'DisplayName','Char. Avg. Elderly','MarkerSize',5,...
    'LineWidth',2.5,'Color',colorElderlyAvg);
hCorrInnerElderly = plot(innCorrNormElderly(:,1), innCorrNormElderly(:,2), '-.'
    ', 'MarkerSize',10,...
    'DisplayName','Corridors Elderly',...
    'LineWidth',1.5,'Color',colorElderlyCorr);
hCorrOuterElderly = plot(outCorrNormElderly(:,1), outCorrNormElderly(:,2), '-.'
    ', 'MarkerSize',10,...
    'DisplayName','Outer Elderly',...
    'LineWidth',1.5,'Color',colorElderlyCorr);
% Display the legend with only the averages and corridors
legend([hAvgYoung, hCorrInnerYoung, hAvgElderly, hCorrInnerElderly], ...
    {'Char. Avg. Young', 'Corridors Young', 'Char. Avg. Elderly', 'Corridors Elderly'},
    ...
    'Location', 'best');
grid on;
hold off;

```