

# **University of Southern Denmark**

TECHNICAL FACULTY - MÆRSK MC-KINNEY MØLLER INSTITUTE

BACHELOR THESIS - JUNE 2025

BSc IN ENGINEERING (ROBOT SYSTEMS)



---

PROJECT TITLE: "ROBOTICS-BASED BRONCHOSCOPY"

SUPERVISOR: THIUSIUS RAJEEETH SAVARIMUTHU

Co-SUPERVISOR: BRUNO MIGUEL GOMES OLIVEIRA

CHARACTER COUNT: 68.938

---

SUBMITTED BY:

Simone I. Lebech  
ID: 190402580

Rikke Aa. Boysen  
ID: 190402895

PRINT: GRAFISK CENTER, SDU

# 1 Abstract

Lung diseases such as chronic obstructive pulmonary disease (COPD), asthma, and pneumonia represent a major global health burden and are among the leading causes of morbidity and mortality worldwide [1]. One of the most important diagnostic and therapeutic procedures for managing these conditions involves the insertion of a flexible bronchoscope (fig. 1) into the airways. However, the success of this technique is highly dependent on the operator's skill level [2], limiting accessibility and standardization. With increasing demand and procedural complexity, there is a growing need for technological solutions that can assist or automate bronchoscopic navigation.

This project explores the development and evaluation of an autonomous, robot-controlled bronchoscopy model, referred to as LEBO, capable of navigating the bronchial tree without the use of preoperative CT imaging. The objective is to make bronchoscopy safer and more accessible by relying solely on real-time monocular depth estimation.

Unlike existing robotic systems that typically require detailed imaging and pre-planned trajectories, LEBO employs a geometric approach using depth and position data to generate 3D pointclouds (section 4.1) and derive a navigation centreline (section 4.3). The system avoids the use of SLAM and instead uses an analytical 3-DOF kinematic model (section 4.5) to follow the centreline while respecting velocity and acceleration constraints inspired by clinical practice [3].

The system was tested in a virtual lung model simulation under controlled conditions. LEBO successfully navigated over 80% of a predefined trajectory, achieving an average positional error of 10.35 mm with a control update step of 20 (section 5.1 & section 5.3) and no more than one collision per trajectory. Motion analysis demonstrated that the system avoided abrupt joint transitions, exceeding 0.3 radians per step, resulting in smooth, continuous motion (section 5.1 & section 5.2).

The results confirm that autonomous bronchoscopy without CT data is both technically feasible and clinically promising. LEBO offers a foundation for future robotic platforms focused on image-free, real-time navigation in bronchoscopic procedures.

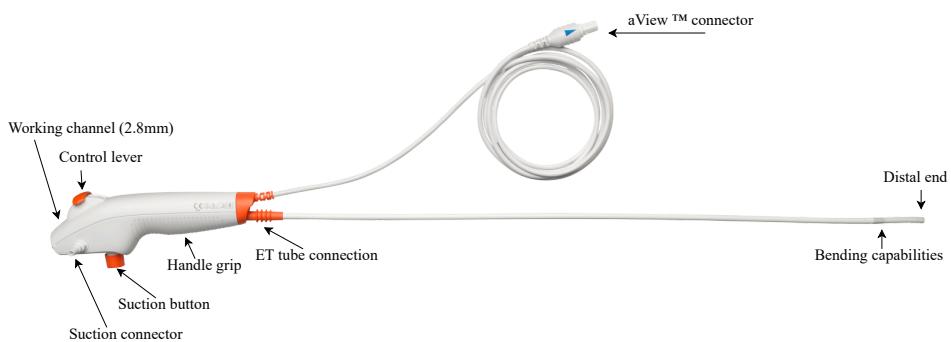


Figure 1: Ambu® aScopeTM[4] with features described

# Contents

<b>1 Abstract</b>	<b>1</b>
<b>2 Introduction</b>	<b>5</b>
2.1 Lung Anatomy and Physiology . . . . .	5
2.1.1 Airway Architecture and Variability . . . . .	5
2.1.2 Dynamic Physiology and Motion Constraints . . . . .	6
2.1.3 Navigation and Safety Considerations . . . . .	6
2.1.4 Summary . . . . .	6
2.2 Motivation and Aims . . . . .	7
2.3 Problem Statement . . . . .	8
2.3.1 Hypothesis 1 - Autonomous Navigation Feasibility . . . . .	8
2.3.2 Hypothesis 2 – Realistic, Stable Motion via Control . . . . .	8
<b>3 State of the art</b>	<b>9</b>
3.1 Bronchoscopy: Anatomical and Procedural Foundations . . . . .	9
3.2 Robotic Systems and Autonomous Navigation . . . . .	9
3.3 Depth Estimation and 3D Reconstruction . . . . .	9
3.4 Anatomical Data and Variation . . . . .	10
3.5 Summary of Gaps and Trends . . . . .	10
<b>4 Methodology</b>	<b>11</b>
4.1 3D Pointcloud Reconstruction from Monocular Depth . . . . .	11
4.1.1 Depth Image Processing . . . . .	11
Intrinsic Parameters . . . . .	11
Extrinsic Parameters . . . . .	12
4.1.2 3D PointCloud Generation . . . . .	12
4.2 Simulation Scenario vs. Real-World Scenario . . . . .	13
4.2.1 PointCloud Alignment . . . . .	13
4.2.2 ICP Alignment . . . . .	14
4.2.3 Denoising, Downsampling, and Saving . . . . .	15
4.2.4 Summary . . . . .	15
4.3 Centreline Extraction . . . . .	16
4.3.1 Input Handling and Preprocessing . . . . .	16
4.3.2 Geodesic Path Estimation . . . . .	16
Graph Construction and Endpoint Selection . . . . .	16
Shortest Path Computation . . . . .	16
Optional Interpolation . . . . .	16

4.3.3	Arc-Length Resampling . . . . .	17
4.3.4	Tangent Estimation and Cross-Sectional Slicing . . . . .	17
4.3.5	Inscribed Circle Center Estimation . . . . .	18
4.3.6	Smoothing and Refinement . . . . .	18
4.3.7	Output and Integration . . . . .	19
4.3.8	Summary . . . . .	19
4.4	Velocity Profile . . . . .	20
4.4.1	Motion Dynamics . . . . .	20
	Joint Definitions . . . . .	20
	Velocity Constraints . . . . .	20
	Acceleration Constraints . . . . .	20
4.4.2	Joint Limit Considerations . . . . .	21
4.4.3	Parameter Justification . . . . .	21
4.4.4	Velocity Profile Generation from Centreline . . . . .	22
4.4.5	Constraint-Driven Motion Planning . . . . .	22
4.4.6	Summary . . . . .	22
4.5	Kinematic Analysis . . . . .	23
4.5.1	Assumptions and Model Basis . . . . .	23
4.5.2	Forward Kinematics . . . . .	24
4.5.3	Mapping from Actuator Space to Configuration Space . . . . .	25
4.5.4	Inverse Kinematics and Path Parametrization . . . . .	25
	Single-Pose Inverse Kinematics . . . . .	25
	Inverse Kinematics Along a Centreline . . . . .	26
4.5.5	Output and Use in Control . . . . .	26
4.5.6	Summary . . . . .	26
<b>5</b>	<b>Results</b>	<b>27</b>
5.1	Control Fidelity and Joint Responsiveness . . . . .	27
5.1.1	Path-Following Accuracy . . . . .	28
5.1.2	Joint-Space Dynamics . . . . .	29
5.1.3	Cartesian Motion Analysis . . . . .	29
5.1.4	Summary . . . . .	30
5.2	Velocity Profile . . . . .	31
5.2.1	Velocity Scaling . . . . .	31
5.2.2	Acceleration Scaling . . . . .	31
5.2.3	Joint-Level Constraint Tests . . . . .	32
5.2.4	Summary . . . . .	33
5.3	Effect of Step Size on Centreline Regeneration . . . . .	34

5.3.1	Experimental Setup . . . . .	34
5.3.2	Results and Observations . . . . .	34
5.3.3	Summary . . . . .	35
5.4	Timing and Velocity influence . . . . .	36
5.4.1	Experimental Setup . . . . .	36
5.4.2	Results and Observations . . . . .	36
5.4.3	Summary . . . . .	37
<b>6</b>	<b>Discussion</b>	<b>38</b>
6.1	Summary of Objectives and Key Contributions . . . . .	38
6.2	System Performance Analysis . . . . .	38
6.2.1	Trajectory Tracking and Control Fidelity . . . . .	38
6.2.2	Joint Dynamics and Motion Realism . . . . .	39
6.2.3	Impact of Velocity and Acceleration Constraints . . . . .	39
6.2.4	Effect of Centreline Regeneration Step Size . . . . .	39
6.2.5	Velocity Scaling and Runtime Behavior . . . . .	39
6.3	Technical Strengths . . . . .	39
6.3.1	Application of Engineering Concepts . . . . .	40
6.4	Limitations of systems . . . . .	40
6.5	Comparison with Existing Approaches . . . . .	40
6.6	Future Development . . . . .	40
6.7	Reflection . . . . .	41
6.8	Summary . . . . .	41
<b>7</b>	<b>Conclusion</b>	<b>42</b>
<b>8</b>	<b>References</b>	<b>43</b>
<b>A</b>	<b>Bachelor Proposal</b>	<b>47</b>
<b>B</b>	<b>Flowcharts</b>	<b>50</b>
B.1	Full process flowchart . . . . .	50
B.2	Sub processes flowcharts . . . . .	51
<b>C</b>	<b>Savitzky–Golay Filter Example</b>	<b>53</b>
<b>D</b>	<b>Testing and results</b>	<b>55</b>
D.1	Effect of stepsize . . . . .	55
D.2	Velocity testing . . . . .	56
<b>E</b>	<b>Github</b>	<b>58</b>

## 2 Introduction

This section provides the background for this thesis by covering the relevant lung anatomy and physiology, outlining the motivation and aims, and formally stating research hypotheses.

### 2.1 Lung Anatomy and Physiology

This section outlines the key anatomical and physiological considerations relevant to bronchoscopy. Bronchoscopy is a key tool in diagnosing and treating pulmonary diseases, yet it is fundamentally constrained by the lungs' complex anatomy and dynamic physiology. Autonomous robotic systems must navigate these challenges with high precision to ensure both effectiveness and patient safety.

#### 2.1.1 Airway Architecture and Variability

The human lungs consist of lobes[5], three on the right, two on the left (see fig. 2a), supplied by the branching tracheobronchial tree. This hierarchy begins at the trachea (generation 0), dividing into main, lobar, segmental, subsegmental bronchi, and then bronchioles (see fig. 2b). As the airways branch, they form a complex, asymmetrical 3D structure that challenges navigation [6]. Peripheral pulmonary lesions (PPLs), frequent targets in robotic procedures [7], lie in small, distal airways, which are narrower, more tortuous, and lack the cartilage support of larger bronchi, making them prone to collapse or deformation. Variations in branching angles and airway lengths further complicate navigation, while mucus can obstruct vision and scope movement [8]. Though airway generations are commonly referenced, anatomical variability limits the usefulness of static models. Robotic navigation systems must rely on real-time 3D reconstructions, often from CT or cone-beam CT, and adapt to intraoperative conditions [2].

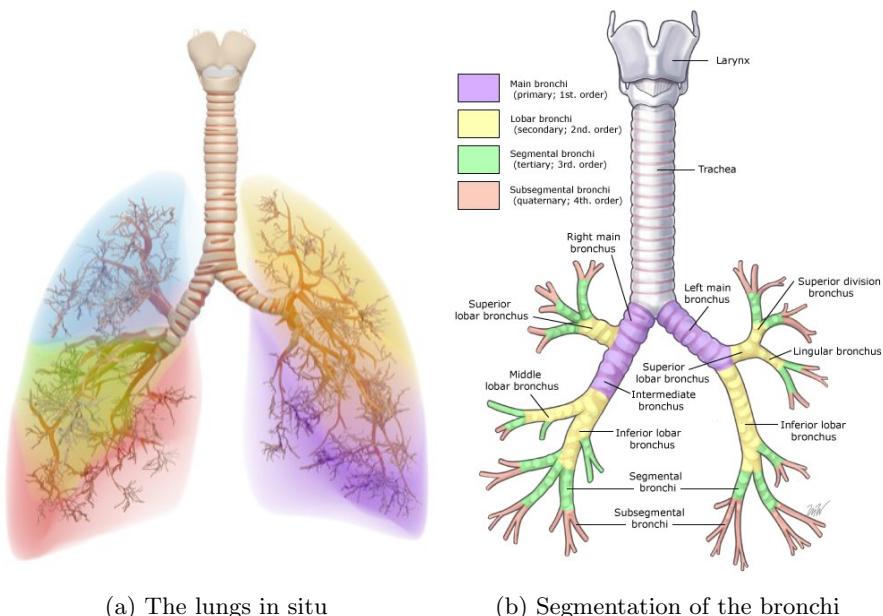


Figure 2: The anatomy of the bronchi

### 2.1.2 Dynamic Physiology and Motion Constraints

Beyond structural complexity, the lungs present a dynamic environment. Normal respiration, driven by diaphragmatic and intercostal muscle activity, causes continuous movement of the airways. This periodic motion alters airway geometry, position, and stiffness, particularly during inspiration and expiration. Cardiac motion also contributes to bronchial displacement, especially in regions near the heart [9].

To operate safely in this environment, robotic systems must synchronize their actions with the respiratory cycle or compensate in real time. Understanding basic pulmonary physiology, including tidal volume, respiratory rate, and airway resistance, is essential. Systems must maintain sufficient stability and precision during procedures such as biopsies or fluid sampling, particularly in the distal lung where tissue is most fragile [10].

### 2.1.3 Navigation and Safety Considerations

Accurate 3D models are essential for safe navigation. These models guide robotic path planning and allow for preoperative trajectory optimization. However, due to intraoperative deformation and motion, systems must also incorporate real-time sensor data (e.g., bronchoscopic vision, force feedback) to refine trajectory execution and avoid collisions.

Patient safety is the foremost concern in robotic bronchoscopy, as improper navigation can lead to complications such as pneumothorax or haemorrhage. Therefore, the system must regulate applied forces, detect collision events, and adapt its motion to avoid trauma. Additionally, failure recovery, infection control, and anaesthesia-related risks must be addressed through robust system design, sterilization protocols, and clinical workflow integration [11].

### 2.1.4 Summary

The anatomy and physiology of the lungs introduce substantial complexity into the design and control of autonomous robotic bronchoscopy systems. Safe and effective operation demands real-time adaptation to a deformable, variable, and moving anatomical environment, supported by high-resolution imaging, force-aware control, and accurate anatomical modelling.

## 2.2 Motivation and Aims

Bronchoscopy is a minimally invasive procedure for inspecting the airways and diagnosing or treating lung conditions. Despite its clinical value, it remains challenging due to the complex branching of the bronchial tree. Traditionally, skilled clinicians manually guide the bronchoscope, which can lead to variable outcomes, longer procedures, and risks such as mucosal injury or pneumothorax.

Robotic systems have been introduced to improve safety and precision. Most rely on preoperative CT scans to create 3D airway models for navigation. However, CT introduces cost, radiation exposure, and is typically limited to peripheral lung procedures. In central bronchoscopy (up to the 3rd or 4th level), CT-based planning is often unavailable and may not adapt to anatomical variation or intraoperative change.

This project presents an autonomous robotic system for bronchoscopy that operates without CT. Using monocular bronchoscopic video, it estimates depth and reconstructs 3D structure in real time for visual navigation. Unlike SLAM, which estimates pose in unknown environments, LEBO assumes known motion and performs mapping by reconstructing pointclouds and extracting the lumen's centreline. A kinematic controller then follows the centreline under velocity constraints for smooth, safe navigation. The primary objectives of this project are:

- To develop a 3D lung reconstruction method based on monocular depth estimation from bronchoscopic video.
- To design a trajectory planning algorithm that enables coverage of the airway lumen by computing local centrelines.
- To implement an autonomous kinematic control system that ensures safe and reliable bronchoscope navigation.
- To validate the system's performance.

By removing the need for CT imaging and enabling real-time, adaptive visual navigation, this work shows that autonomous bronchoscopy in central lung regions is technically feasible and clinically promising. Local centreline computation and constraint-based control offer a safe, efficient alternative to CT-guided strategies.

## 2.3 Problem Statement

While robotic bronchoscopy has advanced the field by offering greater precision and consistency than manual procedures, most systems remain dependent on preoperative CT imaging for path planning. This reliance limits adaptability during procedures and adds cost, time, and radiation exposure.

This project investigates whether autonomous bronchoscopy can be achieved without CT data, using a real-time kinematic control strategy combined with monocular depth estimation to navigate through reconstructed airway geometries. Instead of relying on full SLAM, the system uses image-derived spatial information to continuously update its path and articulation in response to the environment.

The key challenge is whether such a geometry-aware, CT-free approach can navigate the airways with sufficient accuracy and safety under controlled conditions. To guide and evaluate this investigation, the following hypotheses have been formulated:

### 2.3.1 Hypothesis 1 - Autonomous Navigation Feasibility

A robotic bronchoscopy system using monocular depth and pose estimation for 3D lumen reconstruction, together with graph-based trajectory planning, will enable autonomous navigation through a realistic lung phantom. Success will be defined as autonomous traversal of  $\geq 80\%$  of a predefined central airway path, with  $\leq 1$  collision per trajectory.

### 2.3.2 Hypothesis 2 – Realistic, Stable Motion via Control

Incorporating clinically inspired joint velocity and acceleration limits into the control strategy will result in smoother and more biologically plausible articulation compared to an unconstrained baseline. This constraint-aware approach is expected to reduce joint instability and improve motion continuity, particularly in anatomically complex regions. Success will be defined as maintaining an average Euclidean deviation from the centreline under 10 mm.

### 3 State of the art

Robotic bronchoscopy is a rapidly evolving field aimed at enhancing the precision, safety, and autonomy of airway exploration procedures. The literature spans multiple disciplines, including anatomical modelling, robotic control, computer vision, and real-time 3D reconstruction.

#### 3.1 Bronchoscopy: Anatomical and Procedural Foundations

Understanding the tracheobronchial tree’s structure is crucial for effective navigation. Anatomical models such as those by Florens et al. [12] provide detailed geometric representations of the airway tree, which have been foundational in developing bronchoscopic path-planning systems. Ryan et al. [13] further highlight the anatomical challenges and constraints specific to bronchoscopy, including sharp bifurcations and variable airway diameters, to which robotic systems must adapt.

Manual bronchoscopy techniques continue to rely on visual cues and clinician expertise. Cold et al. [2] introduced the “Four Landmarks Approach” for systematic bronchoscopy, providing a structured method to enhance diagnostic accuracy. However, the complexity of manual control underlines the need for intelligent, autonomous systems.

#### 3.2 Robotic Systems and Autonomous Navigation

Recent robotic innovations aim to automate airway navigation. Duan et al. [14] presented a novel robotic bronchoscopy platform designed for both navigation and biopsy, demonstrating the growing feasibility of integrated diagnostic tools. Despite these advances, many systems still depend on preoperative CT for path planning, limiting intraoperative adaptability.

Yang et al. [15] provided a comprehensive survey of 3D reconstruction techniques from endoscopic images, outlining both classical and deep learning-based approaches. Their work underscores a shift toward monocular visual SLAM as a promising alternative to CT-based methods for real-time navigation.

Banach et al. [16] explored visually navigated bronchoscopy using GAN-based depth estimation, achieving plausible lumen reconstructions without external imaging. Complementing this, Yang et al. [17] proposed a geometry-aware deep network for monocular endoscopy, further improving depth accuracy under challenging visual conditions.

#### 3.3 Depth Estimation and 3D Reconstruction

Depth estimation is central to enabling autonomous navigation. Ming et al. [18] reviewed deep learning methods for monocular depth estimation, evaluating CNN-based and self-supervised frameworks suited to low-texture environments like internal airways. Practical implementation often combines depth maps with SLAM and pointcloud processing.

Resources such as Krishna [19] and Safri [20] illustrate how to compute extrinsic matrices and prepare depth maps for 3D reconstruction. Pointcloud refinement, discussed by Daghighe et al. [21] and implemented through Open3D’s statistical outlier removal tools, is critical for building reliable airway models. Tools like Open3D (v0.19) and VTK/VMTK [22] offer scalable pipelines for visualization and model manipulation, supporting both offline analysis and real-time robotic control.

### 3.4 Anatomical Data and Variation

Individual variability in airway dimensions presents another challenge for robotic systems. Studies by Otoch et al. [23] and Lee et al. [24] quantified differences in bronchial length and diameter across patients, suggesting that adaptive systems must respond dynamically rather than relying on static anatomical maps.

Centreline extraction from vessel-like structures and attention mechanisms for pediatric bronchoscopy imaging [25] further demonstrate the importance of accurate localization and segmentation in diverse anatomical contexts.

### 3.5 Summary of Gaps and Trends

Despite significant progress, many robotic bronchoscopy systems remain limited by dependence on preoperative imaging, lack of real-time environmental mapping, and challenges with dynamic airway adaptation. The reviewed literature reveals strong momentum toward SLAM-based, monocular vision systems that reconstruct and navigate the airway in real time, approaches that are central to the research presented in this thesis [26].

## 4 Methodology

This section covers the stages of 3D pointcloud reconstruction the algorithms for centreline extraction, the development of the velocity profile, and the kinematic analysis of the robotic system. A detailed version of fig. 3, can be found in appendix B in fig. 27

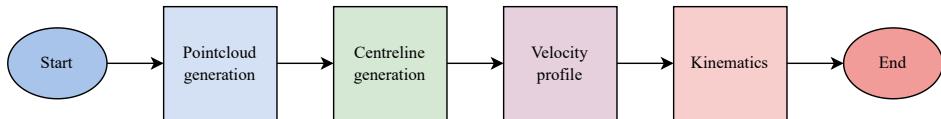


Figure 3: Flowchart for the simplified process.

### 4.1 3D Pointcloud Reconstruction from Monocular Depth

Accurate 3D perception is essential in robotic and medical applications such as autonomous navigation, anatomical reconstruction, and environment mapping. The `pointCloudGenerator` module (See appendix B.2, fig. 28) was developed to generate, align, and refine 3D pointclouds derived from depth maps obtained using monocular camera data. It supports depth processing, region-of-interest selection, 3D reconstruction, frame-to-frame alignment, and post-processing.

#### 4.1.1 Depth Image Processing

The depth image generation pipeline begins with an RGB image (fig. 4) captured using a monocular camera. This image is processed to estimate a depth map (fig. 5), a 2D array where each pixel encodes the distance from the camera to the corresponding point in the scene [27]. In the simulated environment, depth values are provided directly in millimetres. Typically, lighter pixel values correspond to points farther from the camera, while darker values indicate points closer by. Once a depth map is obtained, it must be projected into 3D space to construct a point cloud. This transformation relies on both intrinsic and extrinsic camera parameters.

**Intrinsic Parameters** Intrinsic parameters [28] describe how the camera projects 3D points onto its 2D image sensor. They are determined through camera calibration and include:

**Focal lengths ( $f_x, f_y$ ):** Represent the effective distance between the camera's optical centre and the image plane, measured in pixels along the image's x and y axes. These values define the field of view and influence the scale of the 3D reconstruction.

**Principal point ( $c_x, c_y$ ):** Denotes the intersection of the optical axis with the image plane, ideally at the image centre. Due to manufacturing tolerances, it may be offset from the true centre.



Figure 4: RGB image used for depth estimation.



Figure 5: Resulting depth map in greyscale.

These parameters are incorporated into the intrinsic camera matrix  $K$  (eq. (2)):

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

For the LEBO model, the specific intrinsic matrix is:

$$K = \begin{pmatrix} 181.9375 & 0 & 200 \\ 0 & 181.9375 & 200 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

**Extrinsic Parameters** Extrinsic parameters define the camera’s pose, its position and orientation within a global coordinate frame. They consist of:

**Rotation matrix  $R$**  ( $3 \times 3$ ): Specifies the camera’s orientation relative to the world.

**Translation vector  $t$**  ( $3 \times 1$ ): Specifies the position of the camera’s optical centre in world coordinates.

Together,  $R$  and  $t$  define a rigid body transformation that maps 3D points from the camera coordinate system into a global reference frame. This transformation is essential for aligning the reconstructed 3D geometry with other elements in the environment, such as anatomical meshes, landmarks, or previously acquired data.

#### 4.1.2 3D PointCloud Generation

Depth images are transformed into 3D coordinates [21] using the camera’s intrinsic parameters, as shown in eq. (1). This transformation is a crucial step in the 3D reconstruction pipeline, converting 2D depth information into a spatial representation. It follows the pinhole camera model, as described in [29], which assumes a linear projection without lens distortion and is widely used in computer vision and robotics.

$$X = \frac{(x - c_x) \cdot D(y, x)}{f_x}, \quad Y = \frac{(y - c_y) \cdot D(y, x)}{f_y}, \quad Z = D(y, x) \quad (3)$$

Here,  $(x, y)$  are the pixel coordinates,  $D(y, x)$  is the depth value at each pixel, and  $f_x$ ,  $f_y$ ,  $c_x$ , and  $c_y$  are the camera’s intrinsic parameters. The result is a dense 3D pointcloud that encodes the local geometry of the observed scene. This pointcloud serves as the foundation for subsequent alignment, fusion, and trajectory planning stages within the LEBO model.

## 4.2 Simulation Scenario vs. Real-World Scenario

In the simulation scenario, depth maps are provided directly in millimetres, representing absolute distances from the camera. This allows straightforward 3D pointcloud reconstruction, as detailed above. In contrast, real-world monocular systems estimate relative depth maps from single RGB images. These resemble disparity maps but lack a known metric scale [30]. Since disparity is inversely proportional to depth, without external references, such as a depth sensor, or known object dimensions, the resulting pointcloud cannot be converted into true metric units [30]. Consequently, reconstructions lack scale, making accurate spatial measurements and physical modeling unreliable without additional calibration.

### 4.2.1 PointCloud Alignment

A new pointcloud is generated every 20 steps and transformed into a global coordinate frame using a homogeneous transformation matrix. To maintain consistency with the mesh coordinate system, a 180-degree rotation about the x-axis is applied:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & X_0 \\ r_{21} & r_{22} & r_{23} & Y_0 \\ r_{31} & r_{32} & r_{33} & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} r_{11} & r_{12} & r_{13} & X_0 \\ r_{21} & -r_{22} & -r_{23} & Y_0 \\ r_{31} & -r_{32} & -r_{33} & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

The pointcloud from the first frame serves as a reference for aligning all subsequent frames. For each new frame, a pointcloud is reconstructed from the latest depth image and corresponding transformation matrix. Accurate alignment within a global frame is essential for reliable fusion, visual consistency, and downstream analysis. Misalignment can introduce spatial errors, causing distorted reconstructions where anatomical structures appear displaced or disconnected. Such errors compromise visual coherence and reduce the reliability of spatial measurements. The effect of frame-to-frame alignment is illustrated in fig. 6.

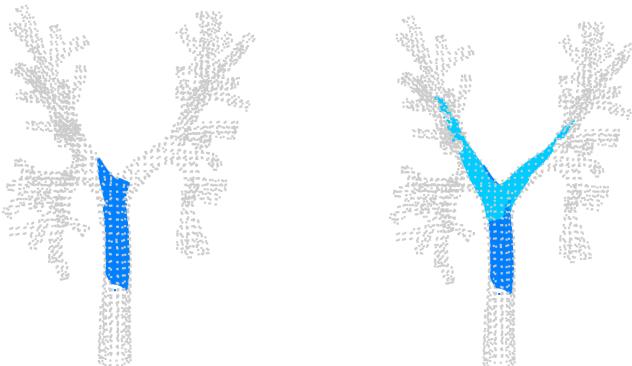


Figure 6: Alignment visualization for frames 10 and 60.

Preserving geometric integrity throughout the pipeline ensures consistent accumulation of 3D data over time, enabling accurate overlay of pointclouds and meshes. This is particularly important for high-precision tasks such as trajectory estimation, deformation tracking, and anatomical segmentation, all of which require sub-millimetre consistency across frames.

#### 4.2.2 ICP Alignment

To accumulate pointclouds across frames, the Iterative Closest Point (ICP) algorithm is used to align each new pointcloud with the previously accumulated one. Point-to-point ICP [31] minimizes the distance between corresponding points by iteratively refining the transformation that best aligns them.

The algorithm takes, as input, the new (target) pointcloud, the current reference pointcloud, an initial transformation matrix, and a correspondence distance threshold. This matrix is iteratively updated during alignment. The output includes the estimated transformation matrix, a fitness score, indicating the proportion of inlier correspondences, and the inlier root mean square error (RMSE).

Following each alignment, the fitness score is evaluated. If the score falls below a set threshold, the alignment is deemed unreliable and the transformation is discarded for that frame. This step is essential to avoid propagating misaligned or noisy data, which would degrade the accuracy and consistency of the reconstructed model. When alignment is successful, the new pointcloud is transformed accordingly and becomes the updated reference for the subsequent frame.

The progressive accumulation of pointclouds using ICP is visualized in fig. 7, illustrating alignment results at various stages of the airway traversal.

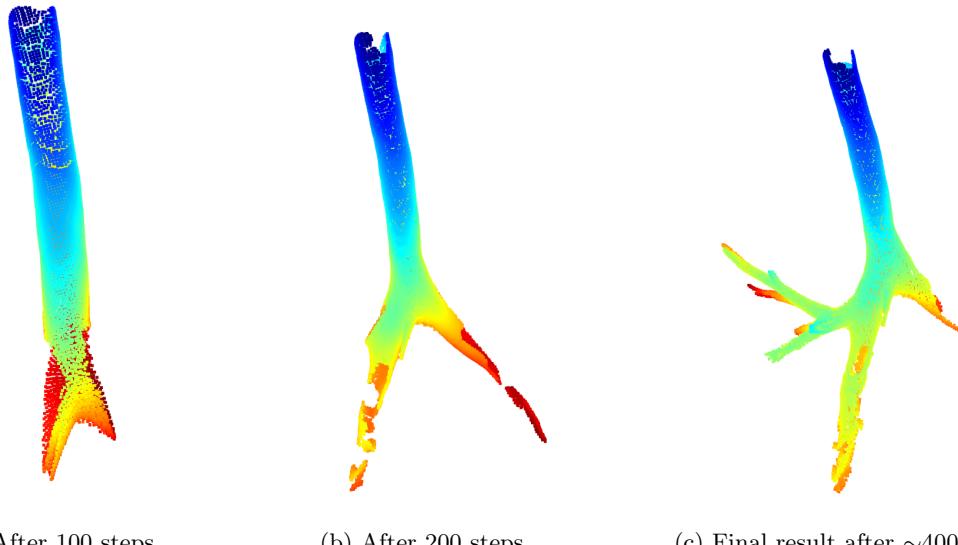


Figure 7: Evolution of the pointcloud throughout the airway traversal using ICP alignment.

#### 4.2.3 Denoising, Downsampling, and Saving

After alignment, post-processing ensures quality and efficiency. Statistical outlier removal eliminates noise by removing points outside a standard deviation threshold based on local neighbourhood distances [32]. Voxel-based downsampling then reduces point density while preserving geometry [33]. A voxel size of 0.008 strikes a balance between detail retention and computational performance. The final pointcloud is saved both incrementally and at the end of the simulation. Data is exported in NumPy and optionally in formats like PLY, enabling downstream use in visualization, machine learning, or navigation systems.

#### 4.2.4 Summary

The `pointCloudGenerator` module implements a full pipeline for 3D pointcloud construction from monocular depth images. Its integration of segmentation, transformation, alignment, and post-processing yields robust and high-fidelity 3D representations well suited to bronchoscopy, robotic navigation, and object reconstruction.

### 4.3 Centreline Extraction

In autonomous bronchoscopy, precise navigation through the bronchial tree requires accurate estimation of the centreline of the airways. This centreline serves as a reference trajectory for the camera, guiding it through the complex tubular structures. To address this, a custom algorithm was implemented in the form of a python class called `centerLineGenerator` (see appendix B.2 fig. 29), which extracts a smoothed, collision-aware centreline from a 3D pointcloud of the airway seen in fig. 8. The centreline extraction pipeline integrates geodesic pathfinding, local cross-sectional analysis, and smoothing. It operates on 3D pointclouds derived from depth maps.

#### 4.3.1 Input Handling and Preprocessing

The algorithm accepts a `PointCloud` object as input, representing the airway geometry. Optionally, the pointcloud is first downsampled using voxel-based filtering to reduce the point density while preserving the underlying geometry. This downsampling improves performance and robustness, particularly in areas where high-resolution data may introduce unnecessary noise or slow down computation. The number of points in the cloud is validated to ensure there is sufficient spatial data to proceed with reliable geometric processing. If the number of points fall below a user-defined threshold, the extraction process is aborted and a warning is issued.

#### 4.3.2 Geodesic Path Estimation

To estimate the centreline of the airway, first a rough trajectory is generated through the pointcloud using geodesic pathfinding. The process is as follows:

**Graph Construction and Endpoint Selection** A  $k$ -nearest-neighbour ( $k$ -NN) graph is constructed over the pointcloud, connecting each point to its  $k$  nearest neighbours [34]. Two endpoints are selected heuristically: one is the point farthest from the pointcloud centroid, and the second is the point farthest from the first.

**Shortest Path Computation** Using Dijkstra's algorithm [35], the shortest path between the two endpoints is computed over the k-NN graph. This results in a discrete path that follows the airway's general topology, but with irregular point spacing, shown in red in Figure fig. 9.

**Optional Interpolation** For smoother curvature and noise reduction, cubic spline interpolation is optionally applied to the geodesic path. This creates a continuous curve that better reflects the expected geometry of the airway.

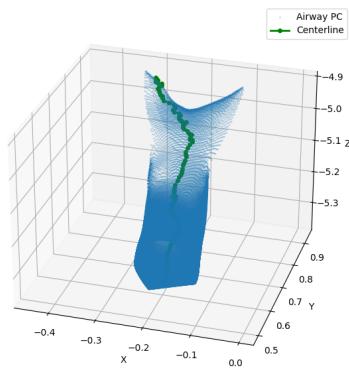


Figure 8: Pointcloud with centreline

### 4.3.3 Arc-Length Resampling

While the geodesic path provides a topologically correct estimate of the airway's centreline, the spacing between successive points is non-uniform. This irregularity can negatively affect tasks such as curvature estimation, virtual camera motion, and robotic control.

To address this, the path is resampled at uniform arc-length intervals:

- The cumulative arc length of the interpolated path is computed.
- A fixed number of evenly spaced positions along this arc length is selected.
- Linear interpolation is used to generate new points at these positions.

This process results in a smooth and regularly sampled curve, which improves stability and consistency in downstream applications. The resampled path is shown in blue in Figure fig. 9, with subfigure fig. 9b highlighting the local uniformity of point spacing.

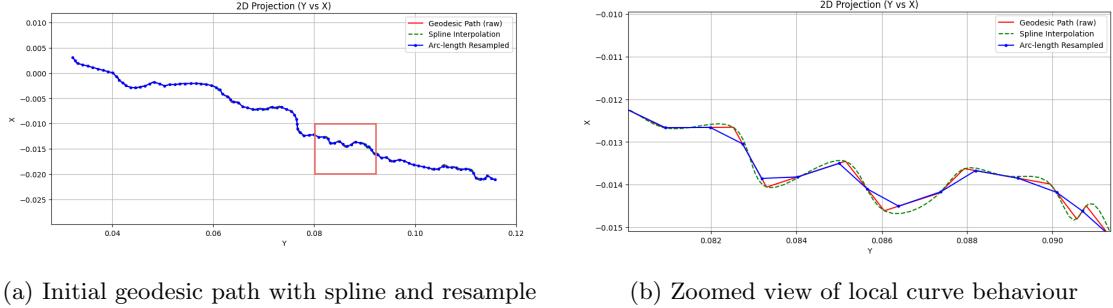


Figure 9: Geodesic path estimation process. Red: raw path via Dijkstra, Green: spline interpolation, Blue: arc-length resampled path.

### 4.3.4 Tangent Estimation and Cross-Sectional Slicing

At each resampled point, a tangent vector is computed using finite difference methods. These tangent vectors represent the local direction of the airway and are used to define cross-sectional planes orthogonal to the centreline. For each such plane, a thin "slab" of the original pointcloud is extracted (see fig. 10). This slab consists of all points that lie within a specified distance from the plane along the tangent direction. The slab forms a local 2D slice of the airway, providing the geometric information needed to determine the most central point within that region.

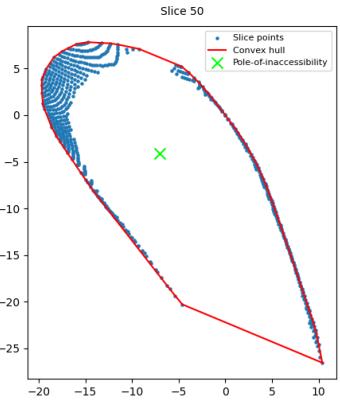


Figure 10: Slice 50 from pointcloud fig. 8

#### 4.3.5 Inscribed Circle Center Estimation

Each cross-sectional slice is analysed to identify its geometric centre. This is done by projecting the 3D slice points onto a 2D plane defined within the slicing frame. A convex hull is then computed from the 2D projection of the slice. The algorithm finds the point within this hull that is farthest from its boundaries, known as the pole of inaccessibility, using the `polylabel` algorithm from the Shapely library [36]. This point represents the best estimate of the local airway centre, as it maximizes clearance from the walls. A clearance threshold is enforced if the computed centre is too close to the airway wall, the point is adjusted inward along the vector pointing from the boundary toward the centre. This helps ensure that the generated centreline maintains a minimum safe distance from the airway walls, reducing the risk of collision or occlusion during autonomous navigation.

In cases where the slice is too sparse or the convex hull is degenerate (e.g., due to sharp corners or partial data), the algorithm defaults to using the original geodesic point at that location as a fallback. This ensures continuity and robustness, even in regions with incomplete or noisy input data. An example of this fallback behavior is illustrated in fig. 11, where, if the ideal inscribed centre (green X) can't be used due to insufficient clearance from the wall, the algorithm falls back to the geodesic path point (red X), ensuring a safe and robust centreline estimate.

#### 4.3.6 Smoothing and Refinement

To improve the quality of the centreline, a final smoothing step is applied to suppress high-frequency noise and jitter introduced by local geometric irregularities or fallback mechanisms. This is important to ensure smooth navigation trajectories for applications such as virtual bronchoscopy or robotic guidance. The smoothing is performed using a Savitzky–Golay filter, which fits a polynomial within a sliding window to perform local regression [37]. This method effectively reduces noise while preserving the overall shape and curvature of the trajectory, a full example can be seen in appendix C. Rather than applying the filter to the full trajectory, it is applied to the residual offsets between the arc-length-resampled geodesic path and the locally estimated centre points. This approach targets local deviations without distorting the anatomical structure of the airway. As shown in fig. 12, the raw offsets exhibit abrupt fluctuations in all three axes, while the smoothed curves show a gradual and consistent trend. The filtering preserves the natural curvature of the path, confirming that important geometric features remain intact.

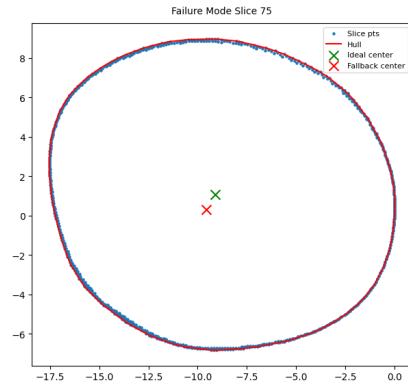


Figure 11: Cross-sectional slice illustrating a fallback scenario. The convex hull (red line) and sampled slice points (blue) are shown. The section is slice 75 from the pointcloud seen in fig. 8

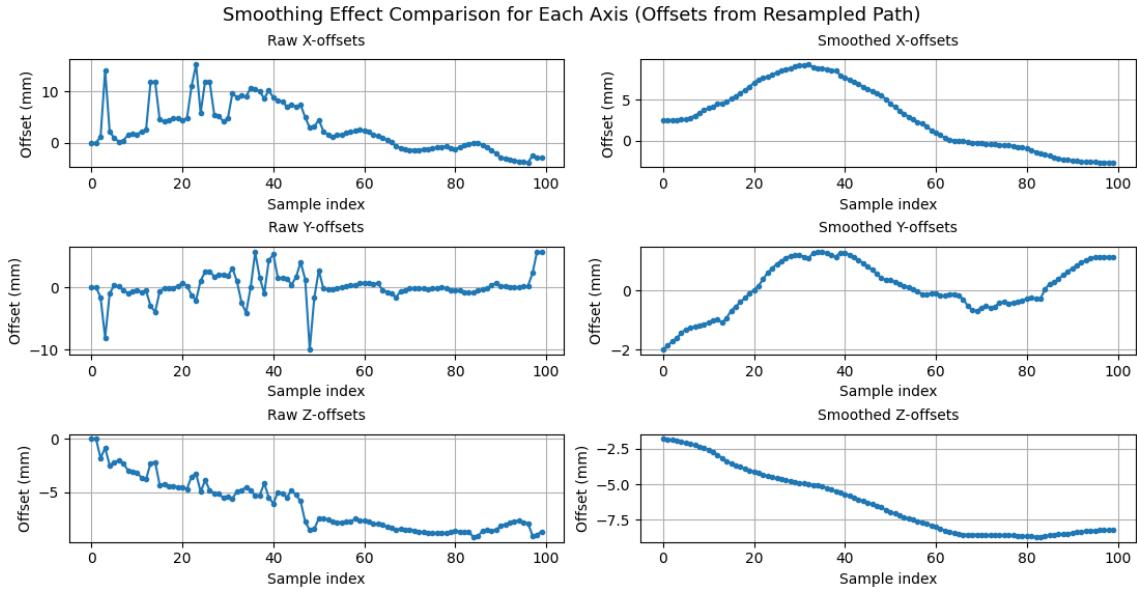


Figure 12: Comparison of raw and smoothed offsets for each spatial axis. Left: raw residuals between the resampled geodesic and unfiltered centreline. Right: smoothed version using a Savitzky–Golay filter. The smoothing suppresses high-frequency noise in all three dimensions while preserving overall shape and trends.

#### 4.3.7 Output and Integration

The final centreline is saved in Numpy array format, enabling seamless integration with downstream systems for visualization, simulation, or control. The saved data can be used as input for virtual camera animation, waypoint generation in motion planning algorithms, or as a reference trajectory for autonomous robotic navigation within the bronchial tree. This modular design allows the centreline extraction system to be applied not only in bronchoscopy but also in other contexts where navigation through tubular geometries is required, such as vascular interventions or industrial pipeline inspection.

#### 4.3.8 Summary

The `centerLineGenerator` class implements a comprehensive and geometry-aware method for extracting navigable trajectories within tubular anatomical structures. By combining geodesic graph search, local cross-sectional analysis, clearance enforcement, and numerical filtering, the algorithm produces anatomically meaningful centrelines that are both smooth and safe for navigation. Its robustness and adaptability make it well-suited for use in autonomous bronchoscopy applications, where spatial precision and wall avoidance are of critical importance.

## 4.4 Velocity Profile

This section outlines the constrained velocity model developed for a robotic bronchoscopy system. The model emulates human-guided procedures while enforcing repeatability, control, and safety. It is based on joint-level motion limits configured according to the specifications of the Ambu® aScope™ 4 Broncho Regular [3], and assumes three controllable degrees of freedom: insertion, axial rotation, and distal tip articulation.

### 4.4.1 Motion Dynamics

The bronchoscope's actuation is organized into three principal motion types: linear insertion, axial rotation, and distal articulation. Each motion is governed by joint-specific constraints, which define the bounds for position, velocity, and acceleration.

**Joint Definitions** The system defines each joint configuration  $q_i(t)$  as a time-varying signal for one of the three degrees of freedom:  $q_0(t)$ : Insertion depth [mm],  $q_1(t)$ : Axial rotation [°] and  $q_2(t)$ : Tip articulation [°]. These are collectively represented by the joint vector:

$$\mathbf{q}(t) = \begin{bmatrix} q_0(t) & q_1(t) & q_2(t) \end{bmatrix}^\top \quad (5)$$

**Velocity Constraints** To ensure stable and controlled actuation, each joint is assigned a maximum velocity. With the limits selected to balance responsiveness with precise control, especially during delicate manoeuvres near airway bifurcations.

$$\begin{aligned} \dot{q}_i(t) &\leq v_i^{\max}, \quad \forall t \in [0, T] \\ v_{\max,0} &\in [1, 5] \quad [\text{mm/s}] \\ v_{\max,1} &\in [30, 90] \quad [\text{°/s}] \\ v_{\max,2} &\in [5, 10] \quad [\text{°/s}] \end{aligned} \quad (6)$$

**Acceleration Constraints** Acceleration is also bounded to prevent sharp impulses and mechanical wear. With the limits selected to ensure smooth velocity ramp-up/-down transitions and reduce overshoot, aiding in the system's real-time control stability at a 100 Hz update rate.

$$\begin{aligned} \ddot{q}_i(t) &\leq a_i^{\max}, \quad \forall t \in [0, T] \\ a_{\max,0} &= 100 \quad [\text{mm/s}^2] \\ a_{\max,1} = a_{\max,2} &= 10 \quad [\text{°/s}^2] \end{aligned} \quad (7)$$

#### 4.4.2 Joint Limit Considerations

Beyond dynamic constraints, the robot is also bounded in its physical configuration space. With specifications based on aScope™ 4 Broncho Regular [3].

$$q_i(t) \in [q_i^{\min}, q_i^{\max}], \quad \forall t \in [0, T] \quad (8)$$

$$q_0 \in [0, 600] \text{ [mm]}$$

$$q_1 \in [-180, +180] \text{ [°]}$$

$$q_2 \in [-180, +180] \text{ [°]}$$

#### 4.4.3 Parameter Justification

These constraints are designed to match the physical and clinical characteristics of the bronchoscope. The insertion range is limited to 600 mm to align with the device's working length, ensuring reach into distal bronchial segments while accounting for instrument slack. The full  $\pm 180^\circ$  bounds for both rotation and articulation support comprehensive access to the tracheobronchial tree.

Velocity limits are selected to allow responsive yet safe movements. A forward insertion velocity between 1–5 mm/s supports cautious navigation through sensitive tissue, while 30–90 °/s axial rotation enables efficient reorientation. Articulation velocities between 5–10 °/s facilitate fine steering without introducing sudden tip motion.

Acceleration limits are tuned to prevent mechanical shock and overshoot, particularly important in soft-tissue environments. A 100 mm/s<sup>2</sup> cap on insertion and 10 °/s<sup>2</sup> on rotation/articulation ensure smooth motion profiles within a 100 Hz control loop. See table 1 for summary.

Table 1: Motion Parameters for Robotic Integration of Ambu® aScope™ 4 Broncho Regular

<b>Joint</b>	<b>Parameter</b>	<b>Value</b>	<b>Units</b>	<b>Justification</b>
Insertion ( $q_0$ )	Position limit	0–600	mm	Working length of aScope™[3].
	Velocity limit	1–5	mm/s	Allows controlled advancement.
	Acceleration limit	100	mm/s <sup>2</sup>	Ensures smooth movement.
Axial Rotation ( $q_1$ )	Position limit	$\pm 180$	°	Provides full rotation.
	Velocity limit	30–90	°/s	Orientation adjustments without compromising stability.
	Acceleration limit	10	°/s <sup>2</sup>	Ensures smooth rotation.
Articulation ( $q_2$ )	Position limit	$\pm 180$	°	The aScope™ offers 180° up and 180° down bending[3].
	Velocity limit	5–10	°/s	Enables precise navigation.
	Acceleration limit	10	°/s <sup>2</sup>	Prevents abrupt motion.

#### 4.4.4 Velocity Profile Generation from Centreline

As described in Section section 4.5.4, joint configurations  $\mathbf{q}(s) = [q_0(s), q_1(s), q_2(s)]^\top$  are computed along the airway centreline  $\mathbf{p}(s)$  using inverse kinematics. To introduce time into the trajectory, a trapezoidal velocity profile  $v(s)$  is applied along arc length [38].

Let  $s \in [0, L]$  be the arc-length parameter. The progression over time is governed by:

$$\frac{ds}{dt} = v(s) \Rightarrow t(s) = \int_0^s \frac{1}{v(s')} ds' \quad (9)$$

This defines a time-parametrized trajectory  $\mathbf{q}(t) = \mathbf{q}(s(t))$ , producing smooth joint motion that respects the insertion velocity limit  $v_{\max,0}$  and supports real-time control. The profile ensures gradual acceleration and deceleration during airway navigation, aiding stability and safety.

#### 4.4.5 Constraint-Driven Motion Planning

Rather than prescribing fixed velocity profiles, the system enforces these bounds during online trajectory generation. Motion planners compute feasible paths  $\mathbf{q}(t)$  that satisfy all inequality constraints:

$$\begin{cases} \dot{\mathbf{q}}(t) \leq \mathbf{v}^{\max} \\ \ddot{\mathbf{q}}(t) \leq \mathbf{a}^{\max} \\ \mathbf{q}^{\min} \leq \mathbf{q}(t) \leq \mathbf{q}^{\max} \end{cases} \quad \forall t \in [0, T] \quad (10)$$

This approach supports flexible, safe movement through the airway while ensuring compliance with anatomical, mechanical, and control constraints.

#### 4.4.6 Summary

The velocity model defines constraint-based joint trajectories for a 3-DOF bronchoscope, reflecting the mechanical and clinical specifications of the Ambu® aScope™ 4 Broncho Regular [3]. A trapezoidal velocity profile is applied along a resampled airway centreline to generate time-indexed commands for insertion, axial rotation, and tip articulation. This ensures smooth, safe, and anatomically consistent motion while satisfying joint limits on position, velocity, and acceleration throughout autonomous navigation.

## 4.5 Kinematic Analysis

In autonomous bronchoscopy, accurate positioning of the bronchoscope tip within the patient's airways requires a kinematic model that accounts for the flexible, cable-driven nature of the instrument. Unlike rigid serial manipulators described by Denavit-Hartenberg parameters, continuum robots operate under a constant-curvature assumption [39]. Therefore, the following section is based on a kinematic model that approximates the bending section of the bronchoscope as a smooth circular arc, implemented via a custom Python class `Kinem`. This model enables both forward and inverse kinematics, facilitating trajectory planning and control.

### 4.5.1 Assumptions and Model Basis

The kinematic model used is inspired by constant-curvature continuum robot theory [14] and operates under the following assumptions:

- The bending section has a single continuous joint with 1 degree of freedom (DOF).
- Each joint is assumed to bend smoothly along a circular arc.
- The robot is driven by tendon tension; gravity and external forces are neglected.
- The rigid portion of the bronchoscope follows the flexible section and only translates.

The kinematics are divided into two stages: (1) actuator space to configuration space, and (2) configuration space to task space. The first stage maps rope displacement to bending angle, while the second computes the tip pose from the bending configuration (see fig. 13).

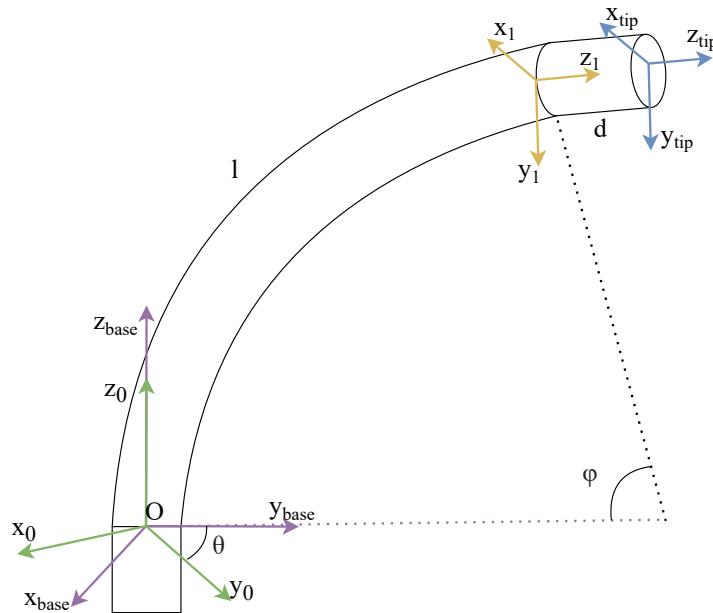


Figure 13: Kinematic model of the bronchoscope (see fig. 1)

### 4.5.2 Forward Kinematics

The forward kinematics (FK) model computes the position and orientation of the tip based on the configuration parameters: insertion distance ( $d$ ), bend angle ( $\varphi$ ), and yaw angle ( $\theta$ ). According to screw theory [40], the screw vector  $\mathbf{v}$  for a revolute joint is:

$$\mathbf{v} = \begin{bmatrix} -\boldsymbol{\omega} \times \mathbf{q} \\ \boldsymbol{\omega} \end{bmatrix} \quad (11)$$

Where  $\mathbf{q}$  is a point on the rotation axis, and  $\boldsymbol{\omega}$  is the rotation vector. The transformation matrix is constructed using the matrix exponential:

$$e^{[\mathcal{V}]\theta} = \begin{bmatrix} e^{[\boldsymbol{\omega}]\theta} & (I - e^{[\boldsymbol{\omega}]\theta})\mathbf{q} \\ 0 & 1 \end{bmatrix} \quad (12)$$

For rotation about the Z-axis, the homogeneous transformation matrix becomes:

$${}^0_{Base}T = e^{[\mathcal{V}]\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

When the bending section is in motion, the transformation matrix describing bending along the circular arc becomes:

$${}^0_0T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi & \frac{l}{\varphi}(1 - \cos \varphi) \\ 0 & \sin \varphi & \cos \varphi & \frac{l}{\varphi} \sin \varphi \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

Here,  $\frac{l}{\varphi}$  is the radius of curvature, and  $\varphi$  is the bend angle. This represents rotation in the bending plane and translation in the YZ directions. For the rigid part of the bronchoscope (length  $d$ ), the transformation matrix is:

$${}^1_{tip}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

The full transformation from the base to the tip of the bronchoscope is then:

$${}_{base}^{tip}T = {}_{base}^0T \cdot {}_0^1T \cdot {}_1^{tip}T \quad (16)$$

$$= \begin{bmatrix} \cos \theta & -\cos \varphi \sin \theta & -\sin \varphi \sin \theta & \frac{l}{\varphi} \sin \theta (\cos \varphi - 1) - d \sin \varphi \sin \theta \\ \sin \theta & \cos \varphi \cos \theta & \sin \varphi \cos \theta & -\frac{l}{\varphi} \cos \theta (\cos \varphi - 1) + d \sin \varphi \cos \theta \\ 0 & \sin \varphi & \cos \varphi & \frac{l}{\varphi} \sin \varphi + d \cos \varphi \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

#### 4.5.3 Mapping from Actuator Space to Configuration Space

The relationship between actuator displacement and bend angle is given by:

$$\begin{aligned} \Delta l &= l_{post} - l_{pre} \\ &= \rho_{post}\varphi - \rho_{pre}\varphi \\ &= \varphi(\rho_{post} - \rho_{pre}) \\ &= \varphi r \\ \varphi &= \frac{\Delta l}{r} \end{aligned} \quad (18)$$

Where  $\Delta l$  is the change in cable length (actuator displacement),  $r$  is the distance from the neutral axis to the tendon, and  $\varphi$  is the bending angle. This assumes a linear relationship between curvature and cable length [41].

#### 4.5.4 Inverse Kinematics and Path Parametrization

While forward kinematics computes the tip pose from internal configuration, inverse kinematics (IK) solves the opposite problem; Estimating configuration parameters required to achieve a desired pose. This is crucial for trajectory tracking and centreline alignment.

**Single-Pose Inverse Kinematics** The `inverse_kinematics` method decomposes a target pose  $T_{des}$  into orientation parameters:

- **Yaw ( $\theta$ )** is extracted from the rotation matrix's X-axis projection onto the XY-plane using  $\arctan 2(R_{21}, R_{11})$ .
- **Bend ( $\varphi$ )** is computed as the angle between the local and global Z-axes:  $\varphi = \cos^{-1}(R_{33})$ .
- **Insertion ( $d$ )** is assumed zero in this minimal formulation and corrected later when calculating arc length.

This analytical approach is fast, avoiding numerical optimization and making it suitable for real-time applications.

**Inverse Kinematics Along a Centreline** To compute actuator values along a full airway trajectory, the `inverse_kinematics_path` method performs the following:

1. **Tangent Estimation:** For each point on the centreline, tangents are estimated using finite differences to approximate local direction.
2. **Frame Generation:** Using the tangent and a fixed world-up vector, a local Frenet-like frame [42] is constructed for each point.
3. **IK Solving:** Each frame is passed to `inverse_kinematics` to obtain  $\theta$  and  $\varphi$ . Insertion  $d$  is computed as the cumulative arc length from the first point.
4. **Pose Reconstruction:** The full  $4 \times 4$  pose is reconstructed by applying FK to the extracted parameters, aligning the orientation to the airway while anchoring the frame to the centreline.

This process produces a list of tip poses as transformation matrices, and corresponding yaw ( $\theta$ ), bend ( $\varphi$ ), and insertion ( $d$ ) values for each point.

#### 4.5.5 Output and Use in Control

The IK output provides actuator commands and corresponding poses for every point on the airway centreline, supporting:

- Trajectory following and interpolation,
- Real-time pose estimation for camera guidance,
- Closed-loop control with feedback correction.

The modular structure of the `Kinem` class ensures it integrates easily with path planners, simulators, or control loops.

#### 4.5.6 Summary

The kinematic analysis implemented in the `Kinem` class provides a complete mapping between actuator, configuration, and task spaces for a constant-curvature bronchoscope model. Both forward and inverse kinematics are supported, enabling robust trajectory tracking and pose estimation during autonomous bronchoscopy. The use of analytical, closed-form IK further ensures computational efficiency, making it suitable for real-time systems.

## 5 Results

This section concerns assessing the effectiveness of the LEBO model through a series of comparative experiments made against a baseline model. The evaluation focused on key metrics including path tracking accuracy, joint responsiveness, motion smoothness, collision avoidance, and pose stability. Each test was designed to isolate specific behavioural aspects of the models within a controlled and consistent simulated environment.

The experiments were performed in a realistic virtual bronchoscopic scenario, created from thoracic CT scans. A segmented 3D model of the airway was used to generate a hollow bronchial structure by dilating the original airway mask along the coronal, sagittal, and axial planes. Subtracting the original mask from the dilated one resulted in an anatomically accurate, hollow airway model suitable for simulation and trajectory planning. From this airway model, a centreline was extracted seen in fig. 14. This centreline, derived from

pre-operative imaging, serves as the reference trajectory and represents a conventional baseline strategy used in robot path planning. It provides the foundation for evaluating both the baseline and LEBO model in terms of their ability to follow a predefined anatomical trajectory.

### 5.1 Control Fidelity and Joint Responsiveness

To evaluate control fidelity and joint responsiveness, experiments were conducted in a realistic virtual bronchoscopic scenario, to evaluates how well the bronchoscope's tip follows a predefined anatomical trajectory, along with behaviour in both joint and Cartesian space. The two models for comparison:

- **Baseline Model (BO):** A conventional path-following strategy using static control parameters, tracking the airway centreline without dynamic kinematic updates.
- **Enhanced Model (LEBO):** Our proposed kinematically controlled system, which updates joint states at each simulation time-step via forward and inverse kinematics (see section 4.5).



Figure 14: Example image illustrating the virtual bronchoscopic scenario or airway model generation with a centreline (red).

### 5.1.1 Path-Following Accuracy

Figure 15 shows a visual overlay of the reconstructed paths compared to the original centreline in both 2D and 3D views. The baseline model (fig. 15a) maintains closer alignment with the centreline, particularly in straight and low-curvature regions. However, in areas with significant directional changes, the LEBO model (fig. 15b) exhibits greater deviations in areas of high curvature, an expected result given its emphasis on joint realism and mechanical responsiveness.

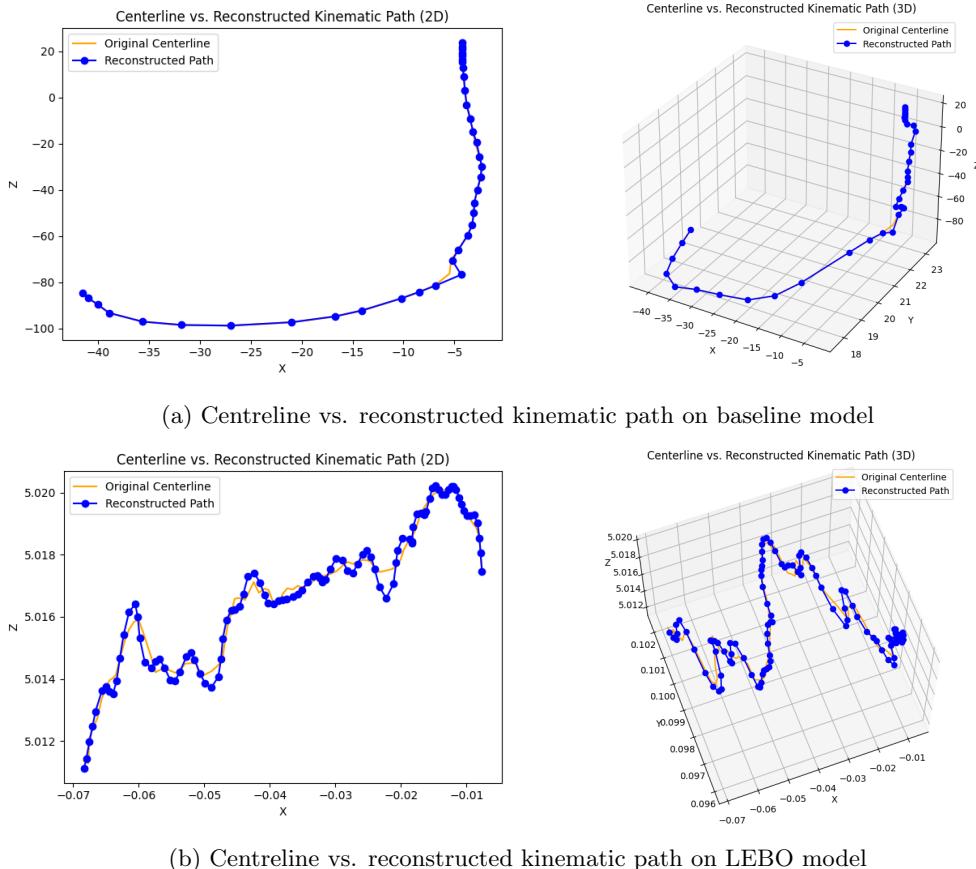


Figure 15: Comparison of Centreline vs. reconstructed kinematic paths

Table 2 shows the mean and maximum Euclidean distance between the reconstructed paths and the centreline. The baseline model achieves sub-millimetre accuracy, while the LEBO model shows larger deviations due to its joint-adaptive behaviour fig. 16. This trade-off reflects a central design principle of the LEBO system: it prioritizes biologically plausible, smooth motion over strict path adherence.

Table 2: Average Euclidean distance between centreline and reconstructed path

Model	Mean Distance (mm)	Max Distance (mm)
Baseline Model	0.47	1.29
LEBO Model	10.35	10.50

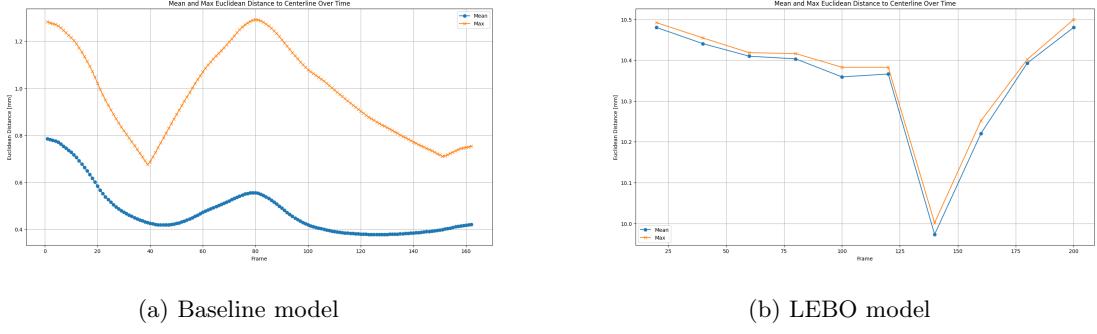


Figure 16: Euclidean distance to centreline over time

### 5.1.2 Joint-Space Dynamics

Next, joint behavior is examined to evaluate model responsiveness. Figure 17 shows the articulation profiles for both models. The baseline model exhibits rigid, step-like transitions, while the LEBO model demonstrates more fluid motion across all joints, enabling context-sensitive articulation.

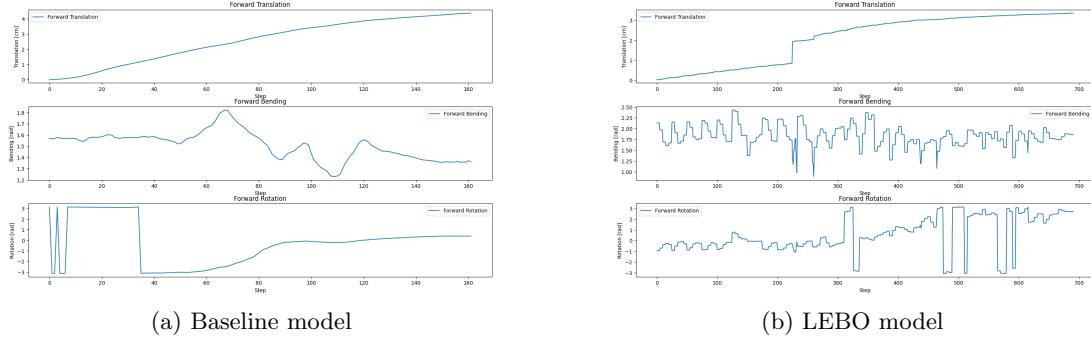


Figure 17: Joint angle trajectories over simulation steps

### 5.1.3 Cartesian Motion Analysis

To assess how these joint behaviors translate into physical motion, fig. 18 compares the Cartesian-space trajectories. The baseline model moves in a nearly linear pattern with minimal curvature or variation. In contrast, the LEBO model exhibits greater variability, reflecting its ability to adapt dynamically to local airway geometry, albeit at the cost of higher positional deviation. To further illustrate control fidelity, the joint-space and Cartesian-space velocity profiles were computed.

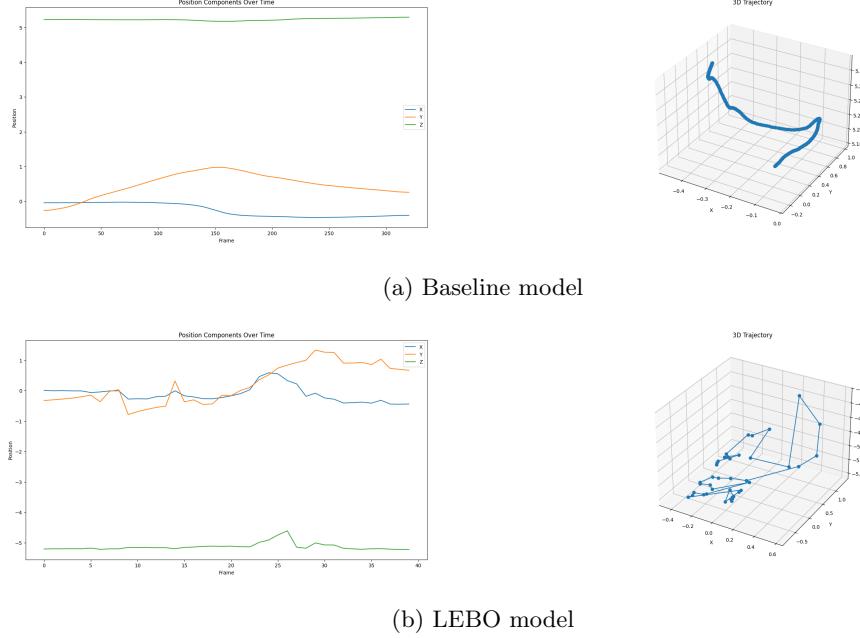


Figure 18: Cartesian position over time and in 3D space

#### 5.1.4 Summary

While the baseline model offers tighter adherence to the predefined centreline, its motion is rigid and lacks adaptive realism. The LEBO system, in contrast, introduces greater path deviation but achieves smoother, biologically inspired articulation, critical for realistic and safe navigation in deformable or complex regions. These results validate the LEBO model's advantages in scenarios requiring dynamic responsiveness over geometric exactness.

## 5.2 Velocity Profile

To assess how constraints affect robot motion during bronchoscopy, a series of tests was conducted comparing the baseline model with the LEBO model under varying velocity and acceleration limits. The influence of different scaling factors on trajectory execution and joint behavior was evaluated. Each scaling factor was applied to both velocity and acceleration. Additionally, each joint type was isolated to examine the specific impact of constraints on individual joint motions.

### 5.2.1 Velocity Scaling

Uniform scaling was applied to the velocity limits across all three joints. Three scaling levels were tested: 100% (baseline), 200%, and 300% of the reference velocity limits. Figure 19 and fig. 20 illustrate the effect of these changes on the baseline and LEBO models, respectively. In the baseline model, the increased velocity limits had minimal impact, motion remained smooth but followed the same overall trajectory profile. In contrast, LEBO responded dynamically: higher velocity limits enabled faster transitions and more flexible articulation, especially in bending and rotation, although with slight increases in instability at extreme speeds.

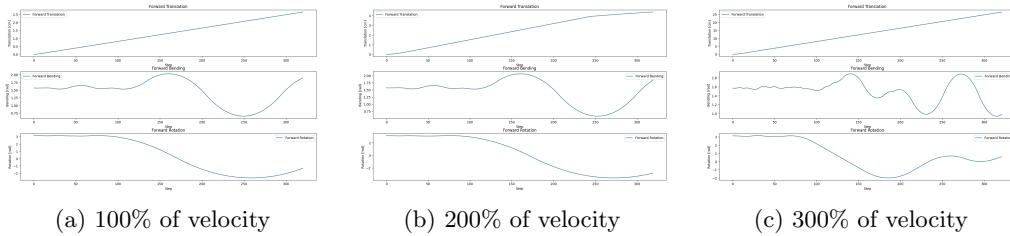


Figure 19: Velocities for baseline model

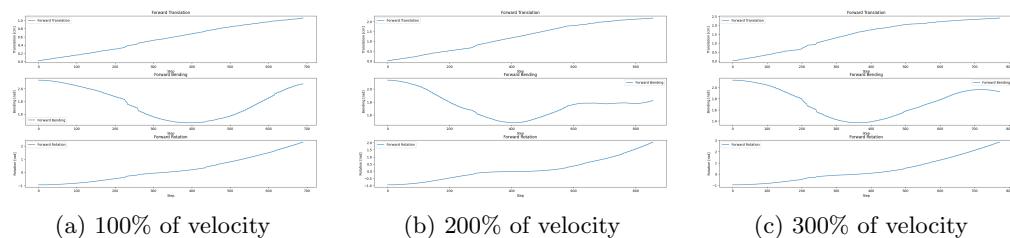


Figure 20: Velocities for LEBO model

### 5.2.2 Acceleration Scaling

Followingly, the same scaling was applied to the acceleration limits. Figure 21 and fig. 22 show the corresponding joint behaviours. As before, the baseline model remains minimally impacted by changes, while the LEBO model shows increasing responsiveness, particularly in bending and rotation, at higher acceleration levels.

## 5 RESULTS

### 5.2 Velocity Profile

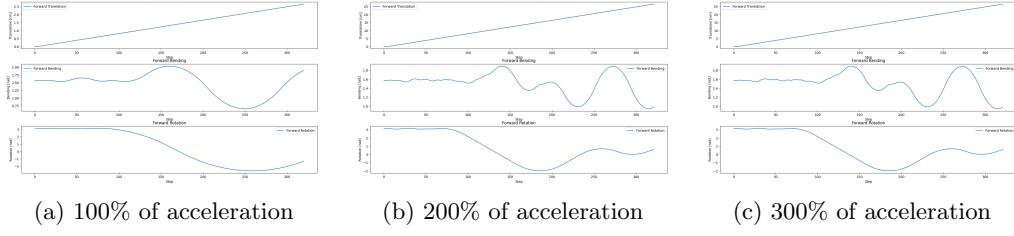


Figure 21: Acceleration for baseline model

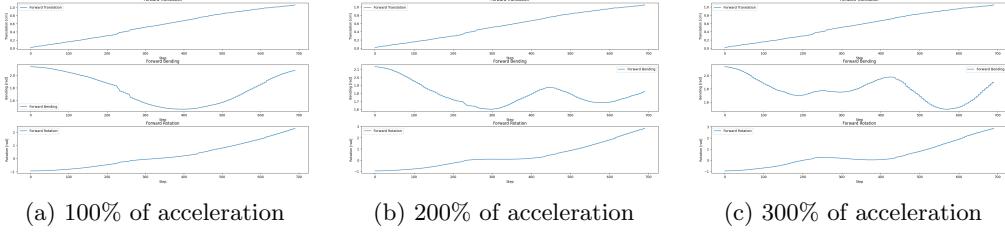


Figure 22: Acceleration for LEBO model

#### 5.2.3 Joint-Level Constraint Tests

Finally, the effect of selectively applying velocity constraints to individual joints, was tested to identify which motion types are most sensitive to velocity limits. As shown in figs. 23 and 24, the baseline model maintains a rigid trajectory across all joints. In contrast, the LEBO model exhibits irregularities, particularly in rotation, where high-frequency oscillations emerge under constrained velocity conditions.

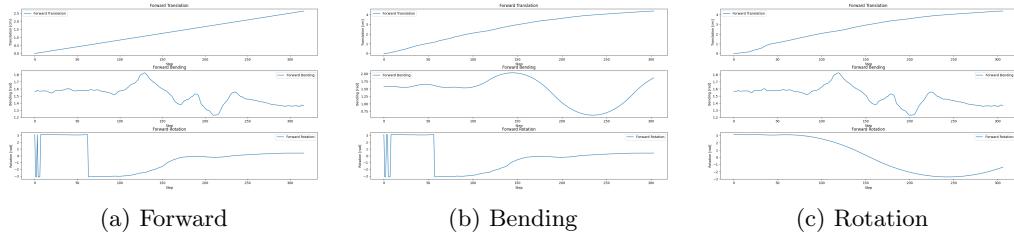


Figure 23: Effect of velocity profile on individual joints (baseline model)

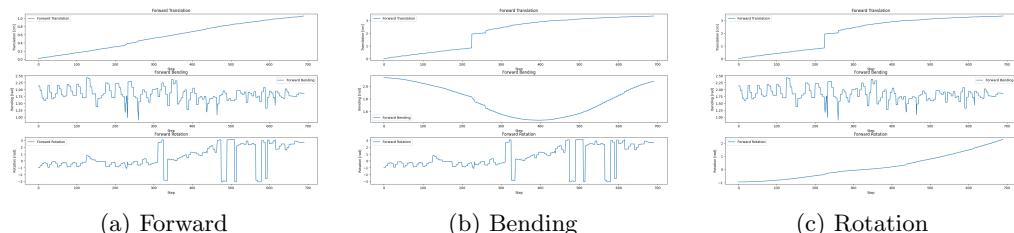


Figure 24: Effect of velocity profile on individual joints (LEBO model)

### 5.2.4 Summary

The LEBO model demonstrates significantly greater sensitivity to velocity and acceleration scaling compared to the baseline model. While the baseline follows a rigid trajectory regardless of parameter values, LEBO reacts dynamically, showing smoother transitions and higher motion variability, particularly in rotational joints. These findings confirm that LEBO is better suited for anatomically adaptive, real-time control tasks, and can be effectively tuned using clinically validated motion constraints for robotic bronchoscopy.

### 5.3 Effect of Step Size on Centreline Regeneration

This experiment investigates how different step sizes used for centreline regeneration impact the motion fidelity of the enhanced LEBO model. The step size determines the interval at which the model recalculates and updates its internal centreline representation. The goal is to identify an optimal step size that balances accurate path tracking with smooth and stable joint behaviour.

#### 5.3.1 Experimental Setup

The LEBO model was tested using step sizes of 5, 10, 20, and 40 under a fixed velocity profile (100% of the clinically recommended limits described in section 5.2). Recorded for each step size:

- **Euclidean distance** between the centreline and reconstructed path — a measure of path-following accuracy.
- **Joint angle trajectories** — a measure of control responsiveness.

Figure 25 shows the joint values and Euclidean distance plots for stepsize 20 (see appendix D.1 for the other tests), and table 3 shows relative differences in max and mean distances.

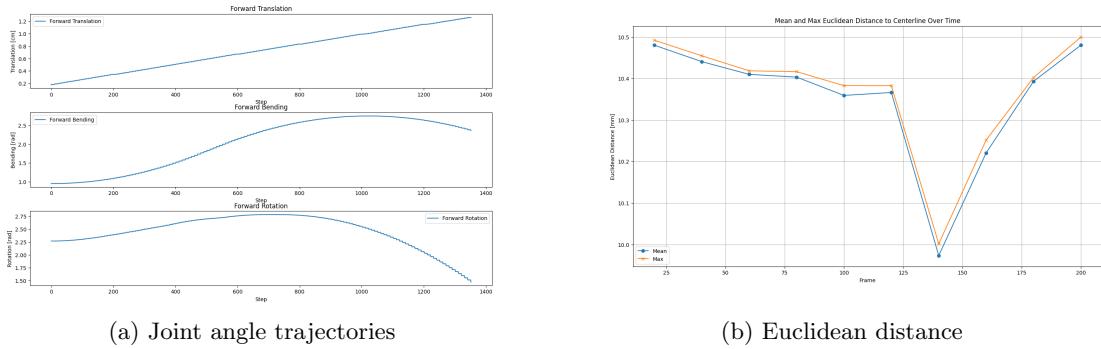


Figure 25: Stepsize 20

Table 3: Average and maximum Euclidean distances between centreline and reconstructed path, with relative differences from the smallest mean and max

Step Size	Mean Distance (mm)	Max Distance (mm)
5	10.39 (0.40%)	10.51 (0.10%)
10	10.37 (0.15%)	10.50 (0.00%)
20	<b>10.35 (0.00%)</b>	<b>10.50 (0.00%)</b>
40	10.39 (0.37%)	10.50 (0.00%)

#### 5.3.2 Results and Observations

At a step size of 5, the LEBO model shows strong joint fluctuations and higher Euclidean deviation, likely due to overly frequent centreline updates introducing noise and abrupt direction changes (see appendix D.1 fig. 33 and section 5.3 table 3). Increasing the step size to 10 and 20 smooths joint

trajectories and reduces Euclidean errors (appendix D.1). At step size 40 (see appendix D.1 fig. 35), accuracy declines. The jagged error curve suggests undersampling, where infrequent updates limit responsiveness to airway geometry and increase interpolation error. Step size 20 yields the most stable performance, with the lowest average (10.35 mm) and maximum (10.50 mm) distances, and smooth joint transitions.

### 5.3.3 Summary

This test demonstrates that the choice of step size plays a critical role in balancing joint stability and path-following accuracy. A step size of 20 was found to be optimal under the tested conditions, yielding the lowest deviation from the centreline and the most stable joint trajectories. Smaller step sizes introduced instability due to overfrequent updates, while larger step sizes compromised path accuracy due to underfitting the airway structure.

## 5.4 Timing and Velocity influence

This section investigates the influence of increasing the velocity scaling factor on both the execution time and path-tracking accuracy of the enhanced LEBO model. The motivation is to assess whether the model remains functionally accurate and stable under faster operational speeds, which would be desirable in real-world scenarios requiring time efficiency.

### 5.4.1 Experimental Setup

The LEBO model was configured to traverse an entire precomputed centreline forward and backward while internally recording the simulation execution time. The experiment was repeated at velocity scaling factors from 1 to 5, where a value of 1 corresponds to the baseline clinical velocity profile (as defined in section 5.2). No other parameters were changed, and centreline regeneration had a step size of 20 (as determined optimal in section 5.3). For each velocity level, the translational and rotational deviation from the reference, the euclidean distance between the reconstructed path and the centreline, and the total execution time for completing the trajectory were recorded.

### 5.4.2 Results and Observations

Figure 26 illustrates the motion behaviour of the LEBO model under velocity scale 4. The subplots include translational and rotational deviation, along with the corresponding Euclidean distance from the path. At all tested velocity levels, the Euclidean deviation curves (appendix D.2 in fig. 40) show consistent oscillatory behaviour. Notably, increasing the velocity does not significantly degrade or improve the model’s accuracy. In fact, the model appears to maintain a similar pattern of deviations across all scaling factors.

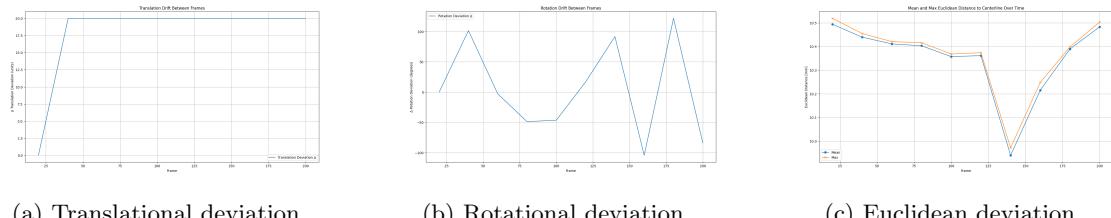


Figure 26: Motion behaviour of LEBO at scale 4

This suggests that path adherence is not directly influenced by the velocity parameter alone under the current control architecture. Although higher velocity limits should reduce execution time, the total trajectory duration remains nearly constant across all tests (see table 4). This indicates that the LEBO model is bottlenecked not by velocity limits, but by internal constraints such as centreline regeneration, collision checks, or control logic overhead.

Table 4: Execution time for a full trajectory at different velocity limits

Velocity Scaling Factor	Execution Time
1	2 min 27.78 sec
2	2 min 44.30 sec
3	2 min 25.08 sec
4	2 min 25.08 sec
5	2 min 28.44 sec

#### 5.4.3 Summary

The LEBO model maintains consistent trajectory adherence across a range of velocity scaling factors. However, increasing the velocity limit does not translate into meaningful reductions in total execution time, likely due to other runtime constraints. From an operational standpoint, this suggests that velocity scaling alone is insufficient to improve runtime efficiency without further optimizations to the simulation pipeline or control update frequency.

## 6 Discussion

This section interprets the results presented earlier and evaluates the system’s performance in the context of the project’s objectives. The aim is to critically assess how well the developed robotic bronchoscopy system meets the technical and clinical requirements for autonomous navigation without preoperative CT imaging. Key areas of focus include motion accuracy, control fidelity, collision avoidance, runtime feasibility, and system robustness. Strengths, limitations and opportunities for future improvement are also discussed. Key findings from Sections section 5.1 to section 5.4 are synthesized and contextualized, with attention given to strengths, limitations, and areas for future development.

### 6.1 Summary of Objectives and Key Contributions

The central aim of this project was to design and test a robotic bronchoscopy platform that operates autonomously without requiring preoperative CT data. Unlike conventional methods that rely on static imaging and pre-planned paths, the LEBO model employs dynamic kinematics and visual feedback for motion control. Key contributions include:

- Real-time kinematic control incorporating forward and inverse solutions.
- Clinically validated motion constraints across velocity and acceleration.
- Systematic evaluation of centreline regeneration parameters.
- Analysis of velocity scaling effects on tracking accuracy and runtime.

### 6.2 System Performance Analysis

This subsection presents an evaluation of the LEBO system’s operational performance by analyzing the accuracy of its trajectory tracking and control fidelity, examines the realism of its joint dynamics, and investigates the impact of various parameters including velocity and acceleration constraints, centreline regeneration step size, and overall velocity scaling on the system’s runtime behaviour.

#### 6.2.1 Trajectory Tracking and Control Fidelity

As shown in fig. 15, the baseline model achieved tighter geometric alignment with the centreline. However, LEBO demonstrated smoother, more adaptive joint behavior (fig. 17), particularly in regions with complex curvature. This reflects LEBO’s design principle of prioritizing mechanical realism over strict geometric fidelity.

### 6.2.2 Joint Dynamics and Motion Realism

Joint-space evaluations in section 5.1.2 (fig. 17) confirmed that the LEBO system produces fluid articulation. These transitions, achieved through constrained inverse kinematics, minimize abrupt motions and are better suited for patient-safe navigation. This is further corroborated by the Cartesian trajectories in fig. 18, which show LEBO adapting more dynamically to airway geometry.

### 6.2.3 Impact of Velocity and Acceleration Constraints

Tests with scaled velocity (section 5.2.1) and acceleration limits (section 5.2.2) revealed that LEBO responds dynamically to motion constraints. High velocity/acceleration profiles permitted faster articulation at the cost of mild instability, while conservative limits enhanced stability. Notably, the baseline model showed little variation, indicating less responsiveness to constraint tuning.

Joint-isolated tests (section 5.2.3) highlighted rotational joints as particularly sensitive, justifying the inclusion of tailored clinical limits in section 5.2.

### 6.2.4 Effect of Centreline Regeneration Step Size

The centreline regeneration test in section 5.3 showed that step size significantly impacts both path tracking and joint stability. Step size 5 produced noisy joint motion and higher deviation due to over-frequent updates. Step size 40 degraded accuracy via undersampling. Step size 20 consistently yielded the best balance (table 3).

### 6.2.5 Velocity Scaling and Runtime Behavior

Velocity scaling from 1 to 5 (see appendix D.2 fig. 40) in section 5.4 did not meaningfully alter execution time (table 4), despite increasing motion limits. This indicates that the model’s runtime is limited by internal processing—likely joint update logic, collision checking, and centreline recalculation—rather than actuator speed. These findings suggest that optimization of the control pipeline is needed for real-time deployment.

## 6.3 Technical Strengths

The system successfully implements several advanced robotics concepts:

- **Path planning:** Dynamic centreline recalculation guided motion within anatomical bounds.
- **Control systems:** Velocity and acceleration constraints improved mechanical realism.
- **Kinematics:** Joint configurations were continuously updated using inverse solutions.

LEBO’s biologically plausible articulation and adaptive joint control offer a compelling alternative to rigid path-following methods, especially in narrow, deformable, or high-curvature regions.

### 6.3.1 Application of Engineering Concepts

This project applied several core engineering principles, particularly in motion planning and control. Inverse kinematics was used to compute joint configurations from spatial targets. Constraint-based trajectory generation ensured adherence to defined velocity and acceleration limits, integrating control loop principles. Path planning algorithms were employed to generate centreline-following behaviour and perform branch navigation within anatomical bounds.

## 6.4 Limitations of systems

Although the LEBO model demonstrated smooth motion and clinical relevance in simulation, several limitations remain. The system exhibited larger path deviations compared to the baseline, which may impact its precision in tightly constrained anatomical regions. It currently lacks external sensing capabilities such as force feedback or contact detection, which could enhance safety and responsiveness during navigation. Runtime performance is limited by computational overhead, creating bottlenecks that prevent efficient real-time execution. Additionally, the system does not yet support branch-targeting or return-path functionality, which restricts its utility for comprehensive airway exploration. Validation has so far been limited to synthetic environments, and its performance under real clinical conditions remains untested. Lastly, the model's behavior in extreme anatomical scenarios, such as sharp turns at subsegmental bifurcations in the right lower lobe, is not fully understood. These regions may exceed the robot's joint articulation limits, and future studies should quantify such thresholds to ensure reliable deployment.

## 6.5 Comparison with Existing Approaches

For central bronchoscopy, preoperative CT is often unavailable. The LEBO strategy may therefore enable robotic-assisted navigation even in settings where pre-planning is not feasible. Its use of real-time vision and kinematic control allows for compliance with hard safety constraints without relying on costly imaging pipelines. Compared to CT-based navigation systems, LEBO provides a more adaptable, lower-cost alternative. It eliminates the need for segmentation and external planning, enabling faster procedure start times and potentially broader accessibility. However, it lacks the anatomical specificity and preplanning capabilities of CT-referenced systems, which may limit its diagnostic applications unless further sensing is added.

## 6.6 Future Development

Several avenues for future development have been identified to advance the LEBO system toward clinical viability. First, optimizing the control logic and step size update routines is essential to enable true real-time operation. Integrating SLAM-based localization would allow the system to track its position within the airway and support return navigation to previously visited sites. Enhancing

anatomical awareness through computer vision or AI-based tagging could improve precision and contextual decision-making during navigation. The LEBO framework also has potential for extension to other minimally invasive procedures in narrow-lumen environments, such as ureteroscopy or colonoscopy. A critical next step is to validate the system in a controlled physical environment using a KOKEN phantom model, where a robot can manipulate a commercial bronchoscope. Finally, incorporating a robust depth estimation method such as EndoOmni [43], which offers zero-shot cross-dataset depth estimation via self-supervised learning, could significantly enhance spatial perception and autonomy.

## 6.7 Reflection

The LEBO system successfully demonstrates that biologically realistic, constraint-based articulation can be achieved without reliance on CT imaging. The project met its primary goal: developing and validating a kinematic control pipeline capable of autonomous motion through anatomically accurate airway models. While limitations remain in geometric precision, real-time processing, and physical deployment, the approach shows promise for future clinical translation. This work contributes a functional and extensible framework for non-image-based robotic bronchoscopy and lays a strong foundation for future research in real-time, minimally invasive navigation.

## 6.8 Summary

The LEBO system demonstrates continuous and biologically realistic articulation through its constraint-based kinematic control. It successfully navigates airway geometries without the need for CT-based planning, highlighting its potential for real-time, image-free bronchoscopy. Experimental findings indicate that a step size of 20 provides the best trade-off between joint stability and path adherence. While velocity and acceleration constraints significantly influence motion fidelity, they do not impact overall runtime due to internal processing bottlenecks. Future development should focus on real-world deployment, robust depth estimation integration, and the addition of return navigation and anatomical targeting capabilities.

## 7 Conclusion

This thesis investigated the feasibility of autonomous robotic bronchoscopy without preoperative CT imaging, aiming to overcome the limitations of existing systems—such as static path planning, radiation exposure, and limited adaptability to intraoperative changes. The proposed model, LEBO, integrates monocular depth estimation, real-time kinematic control, and biologically inspired motion constraints to enable autonomous navigation through the bronchial tree.

The design process was informed by key clinical and anatomical challenges, including the deformable and branching structure of the airways and the dynamic influence of physiological motion. Based on these insights, the LEBO system was developed as a modular pipeline comprising three main components: the `pointCloudGenerator` for monocular depth-based 3D reconstruction, the `centerLineGenerator` for path planning via centreline extraction, and a motion control module constrained by clinically inspired velocity and acceleration limits.

Experimental evaluation in a realistic visual lung phantom simulation demonstrated that LEBO could generate biologically plausible joint-space motion while tracking the airway lumen. Step size and constraint tuning were found to significantly influence motion smoothness and path fidelity, with a step size of 20 yielding the most stable results. The system maintained adaptability across different motion profiles, although slightly higher geometric deviation was observed compared to a baseline.

**Hypothesis 1 (Autonomous Navigation Feasibility)** was largely supported. LEBO achieved autonomous traversal of more than 80% of the predefined central airway path and successfully followed correct bifurcation directions. Collision frequency remained within acceptable limits, with no more than one collision per trajectory. These outcomes confirm that monocular depth and pose estimation, combined with local trajectory planning, can support autonomous navigation without CT-based preoperative models.

**Hypothesis 2 (Realistic, Stable Motion via Control)** was also supported. The integration of velocity and acceleration constraints improved motion stability and reduced abrupt changes in joint angles, particularly in complex anatomical regions. Although the average Euclidean deviation did not consistently remain below the 10 mm target, the improvement in articulation realism and continuity compared to an unconstrained baseline was clearly observed.

In conclusion, this work presents a comprehensive, CT-free framework for autonomous robotic bronchoscopy. By combining real-time mapping, centreline-based trajectory planning, and constraint-aware control, LEBO demonstrates strong potential for safer, more flexible, and clinically relevant robotic navigation in the lung. Future work will focus on physical deployment, high-curvature tracking, and real-time generalization to further advance toward clinical integration.

## 8 References

- [1] Cristine E. Berry and Robert A. Wise. "Mortality in COPD: causes, risk factors, and prevention". In: *COPD* 7.5 (2010), pp. 375–382. URL: <https://www.tandfonline.com/doi/full/10.3109/15412555.2010.510160?scroll=top&needAccess=true#abstract>.
- [2] Kristoffer Mazanti Cold et al. "Systematic Bronchoscopy: the Four Landmarks Approach". In: *Journal of Visualized Experiments* (2023). 2025-06-01. URL: <https://pubmed.ncbi.nlm.nih.gov/37427954/>.
- [3] Ambu A/S. *aScope 4 Broncho Regular Datasheet*. Accessed: 01-06-2025. 2023. URL: [https://www.ambu.com/Admin/Public/Download.aspx?file=Files%2FFiles%2FDownloads%2FAmbu+com%2FFlexibleEndoscopes%2FBronchoscopes%2FaScope+4+Regular%2FDatasheets%2FIE-datasheet-aScope-4-Broncho-Regular\\_V04\\_10912.pdf](https://www.ambu.com/Admin/Public/Download.aspx?file=Files%2FFiles%2FDownloads%2FAmbu+com%2FFlexibleEndoscopes%2FBronchoscopes%2FaScope+4+Regular%2FDatasheets%2FIE-datasheet-aScope-4-Broncho-Regular_V04_10912.pdf).
- [4] Ambu®. *aScopeTM 4 Broncho Large (5.8/2.8)*. <https://www.ambuasia.com/endoscopy/pulmonology/bronchoscopes/product/ambu-ascope-4-large>. 23.04.2025.
- [5] Physiopedia. *Lung anatomy*. [https://www.physio-pedia.com/Lung\\_Anatomy](https://www.physio-pedia.com/Lung_Anatomy). 23.04.2025.
- [6] E. R. Weibel. "Morphometry of the Human Lung". In: *Springer Berlin Heidelberg* (1963). 2025-06-01. URL: <https://link.springer.com/book/10.1007/978-3-642-87553-3>.
- [7] Wes Shepherd. *Image-guided bronchoscopy for biopsy of peripheral pulmonary lesions*. <https://www.uptodate.com/contents/image-guided-bronchoscopy-for-biopsy-of-peripheral-pulmonary-lesions/print>. Accessed: 2025-04-23.
- [8] Gilles Gex et al. "Diagnostic yield and safety of electromagnetic navigation bronchoscopy for lung nodules: a systematic review and meta-analysis". In: *Respiration* 87.2 (2014). Accessed: 2025-06-01, pp. 165–176. URL: <https://karger.com/res/article-abstract/87/2/165/290131/Diagnostic-Yield-and-Safety-of-Electromagnetic?redirectedFrom=fulltext>.
- [9] Andrew D. Scott, Jennifer Keegan, and David N. Firmin. "Motion in cardiovascular MR imaging". In: *Journal of Magnetic Resonance Imaging* 30.3 (2009). Accessed: 2025-06-1, pp. 461–475. DOI: 10.1002/jmri.21874. URL: <https://pubmed.ncbi.nlm.nih.gov/19188310/>.
- [10] Krish Bhadra et al. "Lung Navigation Ventilation Protocol to Optimize Biopsy of Peripheral Lung Lesions". In: *Journal of Bronchology & Interventional Pulmonology* 28.4 (2021). Accessed: 2025-06-01, pp. 290–296. URL: <https://pubmed.ncbi.nlm.nih.gov/33734150/>.
- [11] Bharat Singh Bhandari et al. "Robotic Bronchoscopy: A Comprehensive Review". In: *Journal of Respiration* 4.2 (2024). Accessed: 2025-06-01, pp. 128–139. URL: <https://www.mdpi.com/2673-527X/4/2/11>.
- [12] M. Florens, B. Sapoval, and M. Filoche. "An anatomical and functional model of the human tracheobronchial tree". In: *Journal of Applied Physiology* 110.3 (2011). Date of access: 1-06-2025, pp. 756–763. URL: <https://pubmed.ncbi.nlm.nih.gov/21183626/>.

- [13] Brian Ryan, Kranthi Yendamuri, and Sridhar Yendamuri. “Anatomical considerations in bronchoscopy”. In: *Journal of Thoracic Disease* 9.Suppl 10 (Sept. 2017), S1123–S1127. URL: <https://PMC.ncbi.nlm.nih.gov/articles/PMC5696545/>.
- [14] Xingguang Duan et al. “A Novel Robotic Bronchoscope System for Navigation and Biopsy of Pulmonary Lesions”. In: *Cyborg and Bionic Systems* (2023). 2025-06-01. URL: <https://spj.science.org/doi/abs/10.34133/cbsystems.0013>.
- [15] Zhuoyue Yang, Ju Dai, and Junjun Pan. “3D reconstruction from endoscopy images: A survey”. In: *Computers in Biology and Medicine* (2024). 2025-06-01. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0010482524006309>.
- [16] Artur Banach et al. “Visually Navigated Bronchoscopy using three cycle-Consistent generative adversarial network for depth estimation”. In: *Medical Image Analysis* (2021). 2025-06-01. URL: <https://PMC.ncbi.nlm.nih.gov/articles/PMC8453111/>.
- [17] Yongming Yang et al. “Geometry-aware deep network for depth estimation in monocular endoscopy”. In: *Engineering Applications of Artificial Intelligence* (2023). 2025-06-01. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0952197623001732>.
- [18] Yue Ming et al. “Deep learning for monocular depth estimation: A review”. In: *Neurocomputing* (2021). 2025-06-01. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0925231220320014>.
- [19] Neeraj Krishna. *Camera Extrinsic Matrix with Example in Python*. <https://medium.com/@mediumdata-science/camera-extrinsic-matrix-with-example-in-python-cfe80acab8dd>. 1.06.2025.
- [20] Murtaza Safri. *Point Cloud Data — Preparation and Preprocessing*. <https://medium.com/@murtazasafri/point-cloud-data-0d8c8208e06f>. 01.06.2025.
- [21] Hamid Daghighi et al. “A critical review of discontinuity plane extraction from 3D point cloud data of rock mass surfaces”. In: *Computers & Geosciences* 169 (2022). 01-06-2025, p. 105241. ISSN: 0098-3004. URL: <https://www.sciencedirect.com/science/article/abs/pii/S009830042200190X>.
- [22] Paula Feldman. *VTK and VMTK Deciphered: A Step-by-Step Tutorial and Hands-On MICCAI-inspired Example*. <https://medium.com/@paulaadifeldman1/vtk-and-vmtk-deciphered-a-step-by-step-tutorial-and-hands-on-miccai-inspired-example-c8d9f038543e>. 23.04.2025.
- [23] José Otoch et al. “Is there a correlation between right bronchus length and diameter with age?” In: *Journal of thoracic disease* 5 (June 2013). 2025-06-01, pp. 306–9. URL: <https://pubmed.ncbi.nlm.nih.gov/23825764/>.
- [24] Jin Woo Lee et al. “The comparison of the lengths and diameters of main bronchi measured from two-dimensional and three-dimensional images in the same patients”. In: *Ko-*

- rean Journal of Anesthesiology 66.3 (Mar. 2014). 2025-06-01, pp. 189–194. URL: <https://pubmed.ncbi.nlm.nih.gov/24729839/>.
- [25] Jianqi Yan et al. “Enhanced object detection in pediatric bronchoscopy images using YOLO-based algorithms with CBAM attention mechanism”. In: *Heliyon* (2024). 2025-06-01. URL: <https://www.sciencedirect.com/science/article/pii/S2405844024087097>.
- [26] Vamadevan A Cold KM et al. “Artificial intelligence in bronchoscopy: a systematic review”. In: *Eur Respir Rev*. 2025 May 28;34(176):240274. (2025). 2025-06-01. URL: <https://pubmed.ncbi.nlm.nih.gov/40436614/>.
- [27] Apple Inc. *Capturing Photos with Depth*. date of access: 2025-06-01. 2025. URL: <https://developer.apple.com/documentation/avfoundation/capturing-photos-with-depth>.
- [28] baeldung and Michal Aibin. *Focal Length and Intrinsic Camera Parameters*. <https://www.baeldung.com/cs/focal-length-intrinsic-camera-parameters>. 1.06.2025.
- [29] Ltd. ARBEON Co. *Create a point cloud using a depth map. How? Like this!* 01-06-2025. URL: <https://medium.com/arbeon/create-a-point-cloud-using-a-depth-map-how-like-this-fed598ab4f8c>.
- [30] Jiuling Zhang. *Survey on Monocular Metric Depth Estimation*. Accessed: 2025-06-01. 2025. URL: <https://arxiv.org/abs/2501.11841>.
- [31] S.-Y Park and M. Subbarao. “A fast Point-to-Tangent Plane technique for multi-view registration”. In: *Fourth IEEE International Conference on 3-D Digital Imaging and Modeling* 500 (Nov. 2003). Date of access: 1-06-2025, pp. 276–283. URL: [https://www.researchgate.net/publication/4040350\\_A\\_fast\\_Point-to-Tangent\\_Plane\\_technique\\_for\\_multi-view\\_registration](https://www.researchgate.net/publication/4040350_A_fast_Point-to-Tangent_Plane_technique_for_multi-view_registration).
- [32] Open3D. *Statistical outlier removal*. [https://www.open3d.org/docs/0.9.0/tutorial/Advanced/pointcloud\\_outlier\\_removal.html](https://www.open3d.org/docs/0.9.0/tutorial/Advanced/pointcloud_outlier_removal.html). date of access: 01-06-2025.
- [33] Wenqi Lyu et al. “Dynamic Downsampling Algorithm for 3D Point Cloud Map Based on Voxel Filtering”. In: *Applied Sciences* 14.8 (2024). Date of access: 01-06-2025. ISSN: 2076-3417. URL: <https://www.mdpi.com/2076-3417/14/8/3160>.
- [34] IBM. *What is the k-nearest neighbors (KNN) algorithm?* 2025-06-01. URL: <https://www.ibm.com/think/topics/knn>.
- [35] Erwin KREYSZIG, Herbert KREYSZIG, and EDWARD J. NORMINTON. *ADVANCED ENGINEERING MATHEMATICS*. JOHN WILEY & SONS, INC, 2011.
- [36] Sean Gillies. *Shapely 2.1.1. Documentation*. 2025-06-01. URL: <https://shapely.readthedocs.io/en/stable/manual.html#polylabel>.
- [37] Medium. *Savitzky-Golay Filter*. <https://medium.com/python-engineers/introduction-to-the-savitzky-golay-filter-a-comprehensive-guide-using-python-b2dd07a8e2ce>. date of access: 01-06-2025.

- [38] Christian Loyoza. *Trapezoidal Velocity Profile*. 2025-06-01. URL: <https://medium.com/@clozoya1729/trapezoidal-velocity-profile-f1892c720cd7%7D>.
- [39] Jessica Burgner-Kahrs. *Constant-Curvature Model of Tendon-driven Continuum Robots*. Accessed: 2025-06-01. 2022. URL: <https://www.opencontinuumrobotics.com/101/2022/12/09/tdcr-cc-model.html>.
- [40] Zhen Huang, Qinchuan Li, and Huafeng Ding. “Basics of Screw Theory”. In: *Mechanisms and Machine Science* (Jan. 2013). Accessed: 01-06-2025, pp. 1–16. URL: [https://link.springer.com/chapter/10.1007/978-94-007-4201-7\\_1](https://link.springer.com/chapter/10.1007/978-94-007-4201-7_1).
- [41] Feng Wei et al. “Structural Design and Kinematic Analysis of Cable-Driven Soft Robot”. In: *Actuators* 13 (Dec. 2024). Accessed: 01-06-2025, p. 497. URL: <https://www.mdpi.com/2076-0825/13/12/497>.
- [42] Catherine Elaina Eudora Ross. “The Frenet Frame and Space Curves”. Accessed: 01-06-2025. MS Thesis. Missouri State University, 2019. URL: <https://bearworks.missouristate.edu/theses/3439>.
- [43] Qingyao Tian et al. “EndoOmni: Zero-Shot Cross-Dataset Depth Estimation in Endoscopy by Robust Self-Learning from Noisy Labels”. In: *CoRR* abs/2409.05442 (2024). URL: <https://doi.org/10.48550/arXiv.2409.05442>.

## A Bachelor Proposal

# Robotics-based bronchoscopy

**Project Description:** Intelligent Robotic Bronchoscopy for Accurate and Safe Lung Examinations

**Project dates:** 1<sup>st</sup> of February 2025 – 2<sup>nd</sup> of June 2025

**Students:** Rikke Aalling Boysen ([riboy22@student.sdu.dk](mailto:riboy22@student.sdu.dk)),

Simone Ingvild Lebech ([sileb18@student.sdu.dk](mailto:sileb18@student.sdu.dk))

**Supervisor:** Thiusius Rajeeth Savarimuthu ([trs@mmmi.sdu.dk](mailto:trs@mmmi.sdu.dk))

**Co-supervisor:** Bruno Oliveira ([broli@mmmi.sdu.dk](mailto:broli@mmmi.sdu.dk))

**Institute:** Mærsk McKinney Møller, SDU Robotics



University of  
Southern Denmark

## Problem Statement

Bronchoscopy, a minimally invasive procedure to examine the airways and the lungs, is a critical tool in diagnosing and treating various respiratory conditions. However, the procedure's technical complexity, requiring precise control and navigation of the bronchoscope, presents significant challenges. Manual bronchoscopy is often susceptible to human error, potentially leading to suboptimal outcomes, patient discomfort, and increased procedure time.

## Proposed solution

To solve these issues, there is a pressing need for an intelligent robotic system that can automate the bronchoscopy process, hence making it possible to conduct a safe and accurate bronchi inspection without the need for an experienced professional. This project builds upon previous research, as seen below:

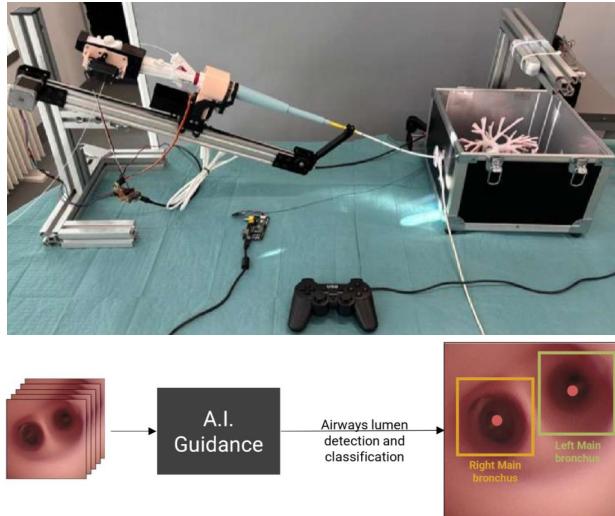
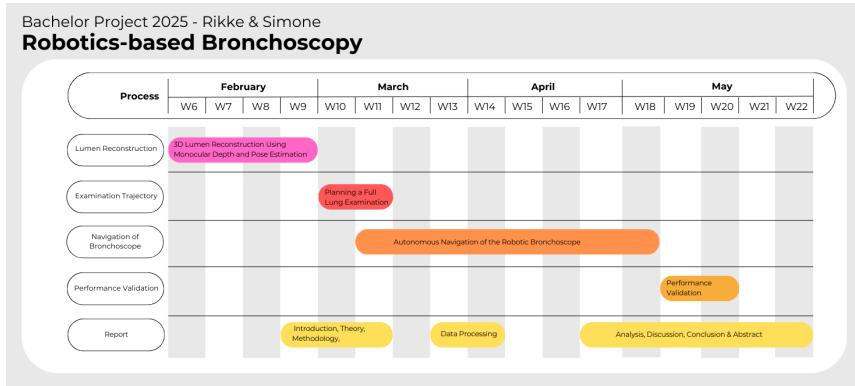


Figure 1: Current Status of bronchoscope and lumen detection

Continuing upon the research in automatic airway detection, classification, and intelligent robotic control, this project aims to focus on the following key areas:

1. 3D Lumen Reconstruction Using Monocular Depth and Pose Estimation (1 month): Develop a strategy for 3D reconstruction of the airway lumen, utilizing monocular depth estimation and pose estimation state-of-the-art strategies.
2. Planning a Full Lung Examination Trajectory (2 weeks): Create a graph-based strategy to plan the sequence required for a full lung examination. This approach ensures a standardized and efficient methodology to guide the robot based on detected airway lumen structures.
3. Autonomous Navigation of the Robotic Bronchoscope (2 months): Implement autonomous navigation of the robotic bronchoscope along the pre-defined trajectory (from task 2). To ensure safety and efficiency, the robot will be programmed to maintain a central position within the airway lumen, using real-time 3D reconstruction data (from task 1) to avoid collisions with airway walls.
4. Performance Validation Using a Realistic Phantom Model (2 weeks): Evaluate the performance of the robotic bronchoscope in a realistic lung model, KOKEN phantom.

The diagram below illustrates the expected timeline for the division of tasks within the project:



## Expected impact

The successful implementation of this intelligent robotic bronchoscopy system, is expected to provide the following major benefits:

1. Improved Diagnostic Accuracy: The introduction of automated bronchoscopy can lead to more accurate and timely diagnosis by enabling the precise targeting of specific lung regions.
2. Improved Patient Safety: The risk of complications such as tissue damage and pneumothorax associated with conventional manual bronchoscopy, may be minimized with robotic control.
3. Reduced Procedure time: Automated navigation can streamline the procedure, and consequently reduce patient discomfort and anesthesia time.
4. Remote access: In the future, remote robotic bronchoscopy could enable access to specialized expertise in underserved areas.

## Project Milestone

The primary milestone in this project is the development of an automated trajectory planning and robotic control system that can guide the bronchoscope to a user-defined airway within the lung. This will demonstrate the feasibility of robotic bronchoscopy and lay the foundation for future advancements, such as full lung diagnostics and more complex surgical procedures.

By addressing the limitations of manual bronchoscopy and by embracing the power of robotics and artificial intelligence, this project aims to transform the field of respiratory medicine, improving patient outcomes and enhancing the overall quality of care.

## B Flowcharts

The following flowcharts can be found in the Github repository linked in appendix E, under images and flowcharts.

### B.1 Full process flowchart

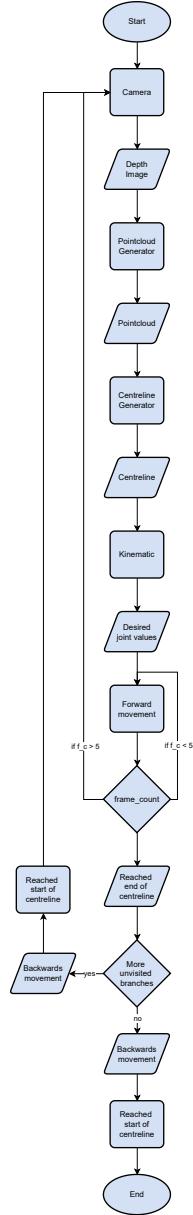


Figure 27: Flowchart of full process

## B.2 Sub processes flowcharts

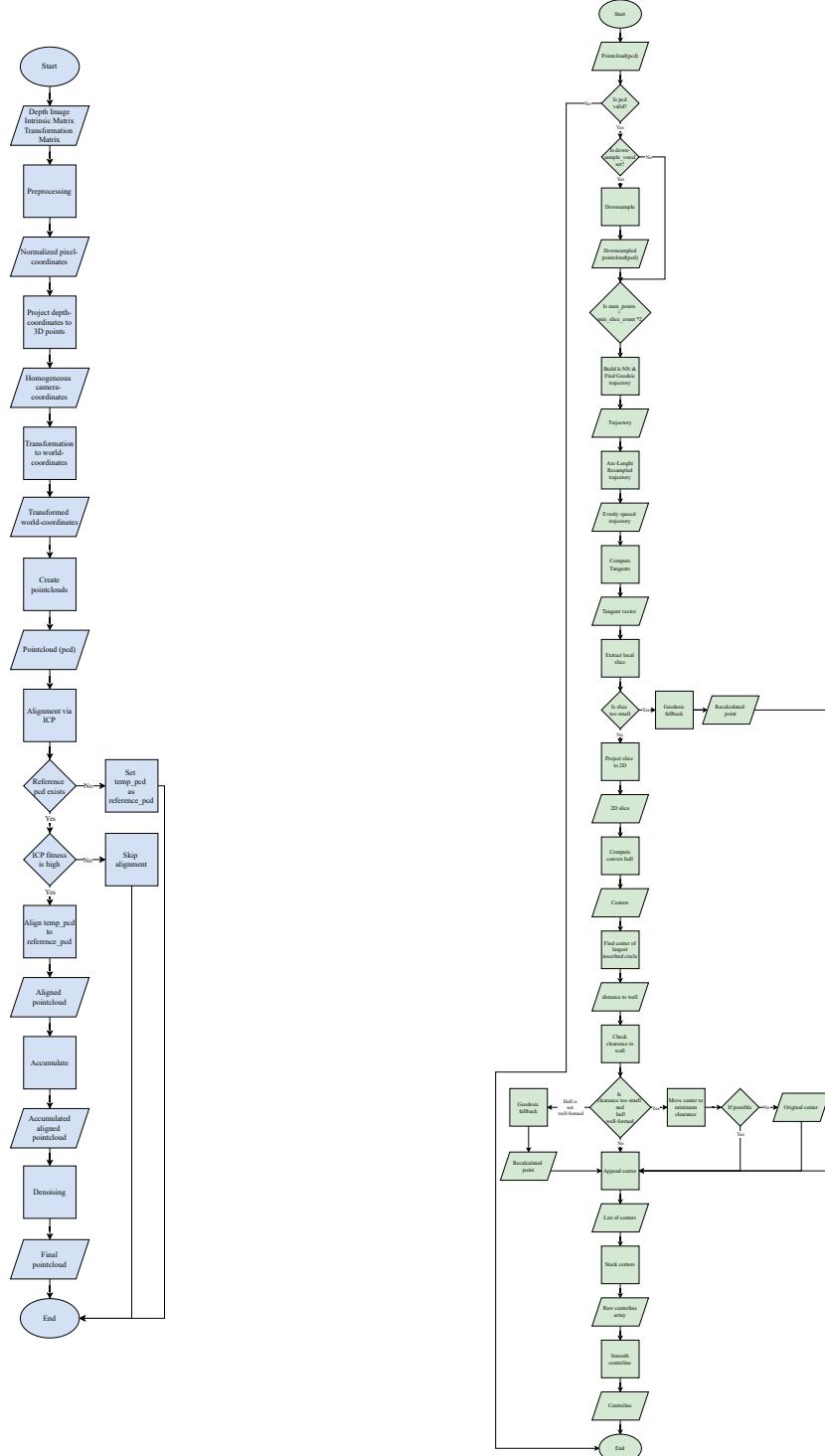


Figure 28: Flowchart for the generation of pointclouds

Figure 29: Flowchart for the generation of pointclouds

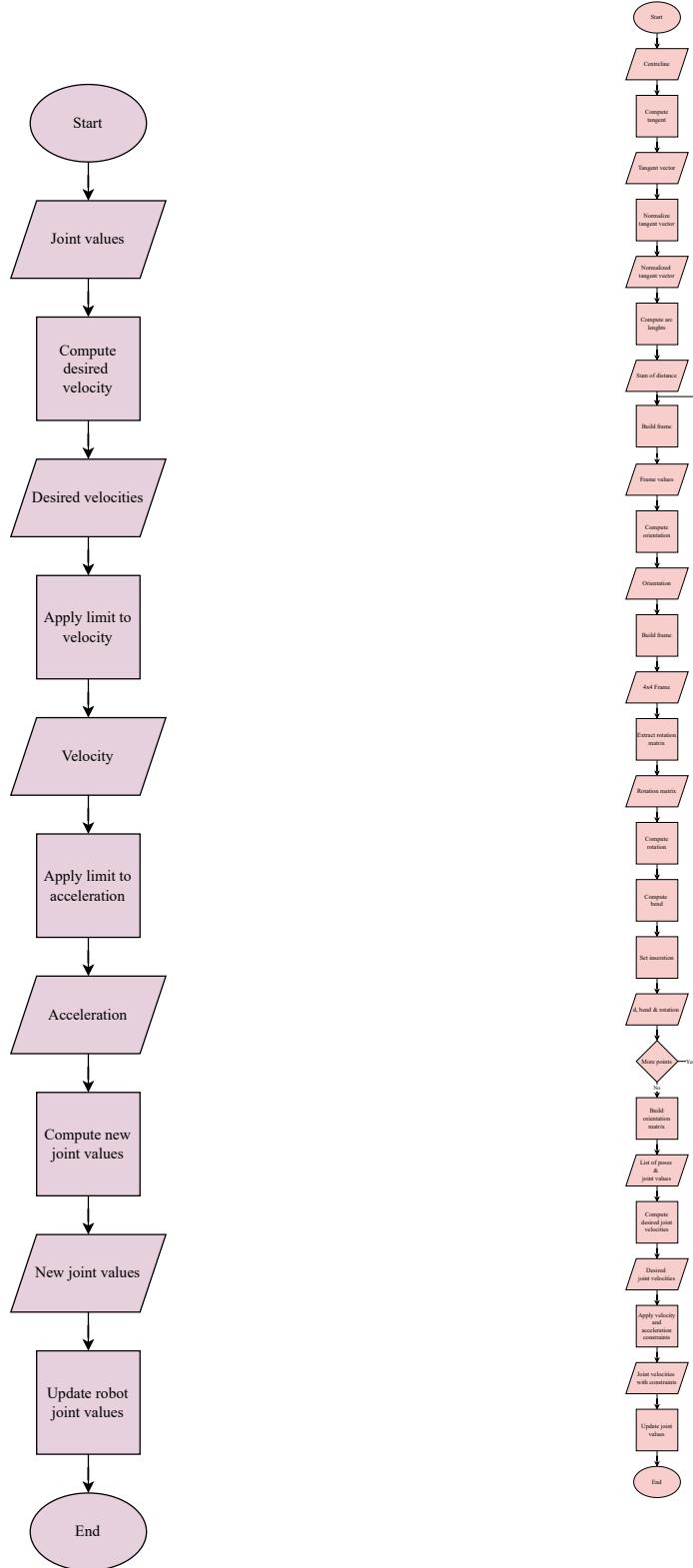


Figure 30: Flowchart for the velocity profile application

Figure 31: Flowchart for the application of kinematic

## C Savitzky–Golay Filter Example

This section illustrates the construction of a Savitzky–Golay smoothing filter with a window size of 71 and a polynomial order of 4. The goal is to perform local polynomial fitting in a least-squares sense and use the result to smooth a noisy signal.

### Design Matrix Construction

We define the half-window size as  $m = 35$ , so the full window includes  $2m + 1 = 71$  samples centered around a given point. Let the relative indices be:

$$x_k = k, \quad k = -35, -34, \dots, 0, \dots, 34, 35$$

To perform polynomial fitting of order  $n = 4$ , we construct the design matrix  $\mathbf{A} \in \mathbb{R}^{71 \times 5}$ , where each row corresponds to a data point  $x_k$  and is defined as:

$$\mathbf{A}_k = [1, x_k, x_k^2, x_k^3, x_k^4]$$

This matrix allows us to fit a quartic polynomial locally to the data in each window.

### Filter Coefficients

Let  $\mathbf{y} \in \mathbb{R}^{71}$  be the vector of observed values within the window. We solve the local least-squares problem:

$$\min_{\mathbf{a} \in \mathbb{R}^5} \|\mathbf{A}\mathbf{a} - \mathbf{y}\|^2$$

The fitted polynomial is evaluated at  $x = 0$ , which corresponds to the center of the window. This gives:

$$\hat{y}_i = a_0 = \mathbf{e}_1^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{y}$$

where  $\mathbf{e}_1 = [1, 0, 0, 0, 0]^\top$ . This expression shows that the smoothed value  $\hat{y}_i$  is a linear combination of the observed values  $\mathbf{y}$ , with weights  $\{c_k\}$  forming the convolution kernel:

$$\hat{y}_i = \sum_{k=-35}^{35} c_k y_{i+k}$$

The kernel  $\{c_k\}$  is symmetric and centered at  $c_0$ , which typically has the largest weight, reflecting the influence of the central sample.

## Example Coefficients

Below are representative values from both ends of the filter kernel (rounded to five decimal places):

$$\{c_{-35}, \dots, c_{-31}\} = \{0.00477, -0.00319, -0.00682, -0.00712, -0.00453\}$$

$$\{c_{31}, \dots, c_{35}\} = \{-0.00453, -0.00712, -0.00682, -0.00319, 0.00477\}$$

The symmetry of the coefficients ensures that the filter preserves the overall shape and trends of the data while reducing high-frequency noise.

## Smoothed Output

Applying the filter across the full signal yields the smoothed output:

$$\hat{y}_i = \sum_{k=-35}^{35} c_k y_{i+k}$$

This convolutional formulation allows for efficient computation and is especially useful in applications like spectroscopy, biomedical signal processing (e.g., ECG, EEG), and other domains where signal integrity must be maintained.

## Result

Figure 32 illustrates the effect of the filter. The original noisy sine wave is shown in orange, and the smoothed output in red. The Savitzky–Golay filter effectively reduces noise while preserving key features of the signal such as peaks and slopes.

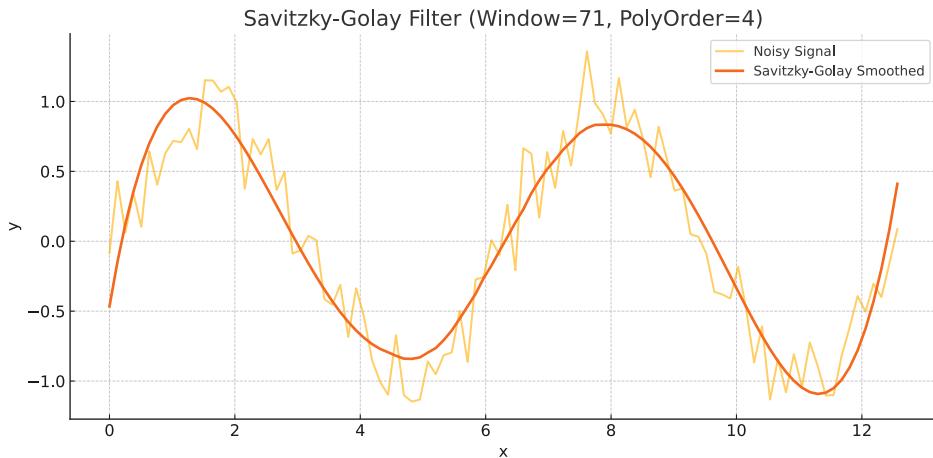


Figure 32: Smoothed sine wave using the Savitzky–Golay filter with a 71-point window and quartic polynomial.

## D Testing and results

The following graphs relate to the different tests in section 5.

### D.1 Effect of stepsize

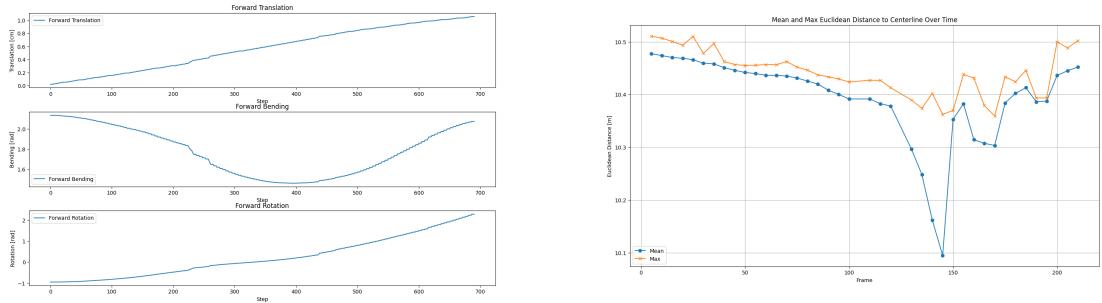


Figure 33: Stepsize 5

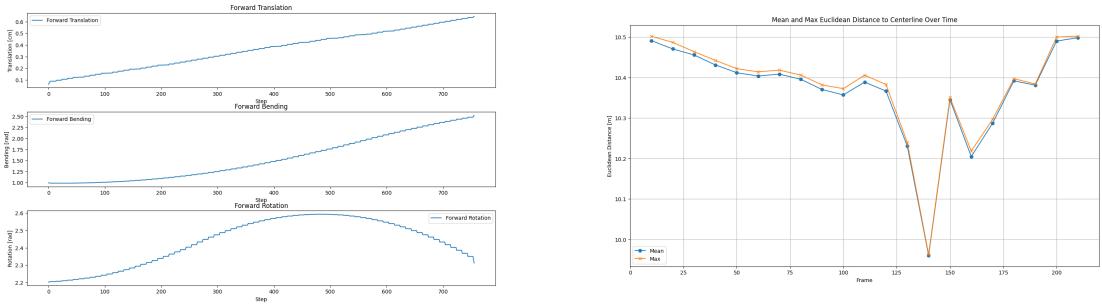


Figure 34: Stepsize 10

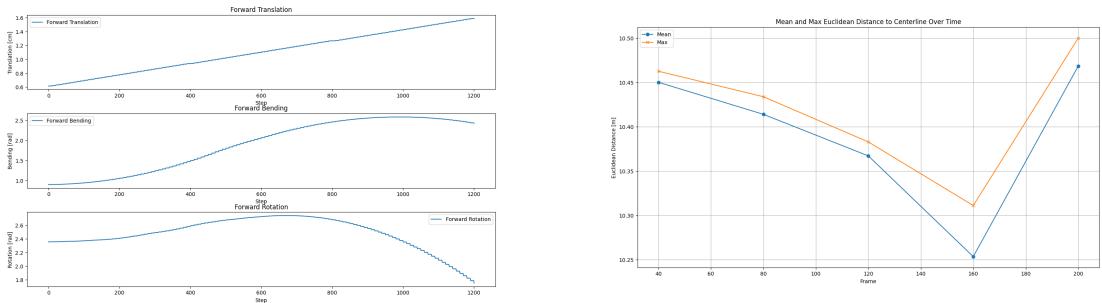
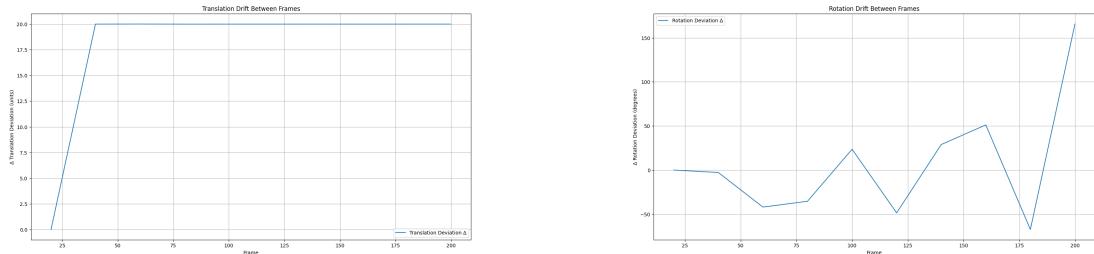


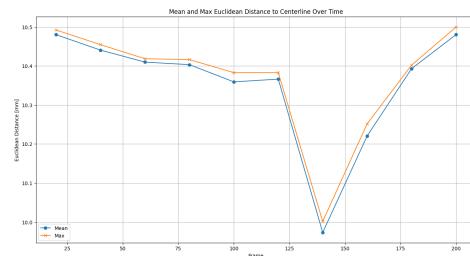
Figure 35: Stepsize 40

## D.2 Velocity testing

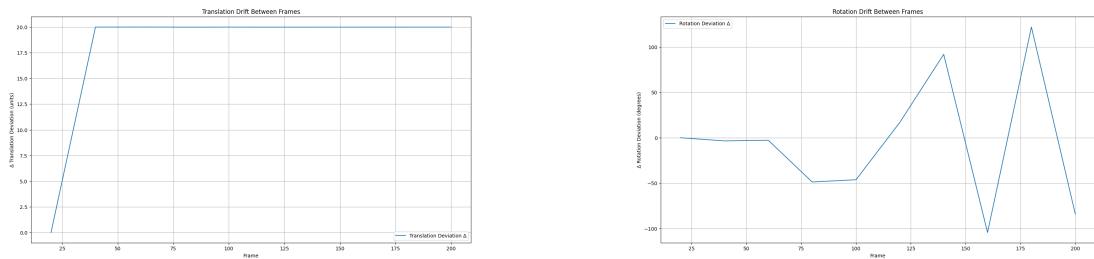


(a) Velocity 1 – Translational deviation

(b) Velocity 1 – Rotational deviation

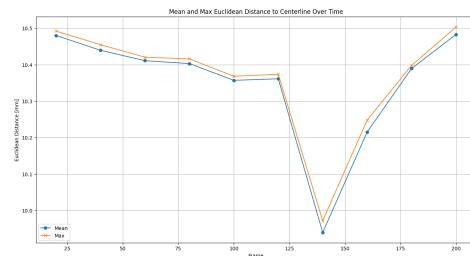


(c) Velocity 1 – Euclidean path error

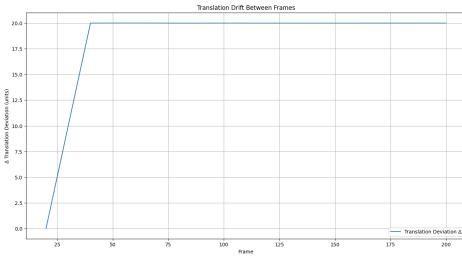


(a) Velocity 2 – Translational deviation

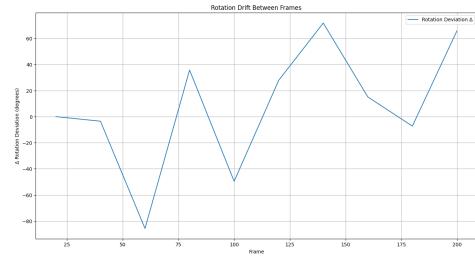
(b) Velocity 2 – Rotational deviation



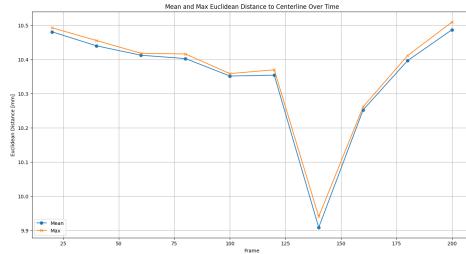
(c) Velocity 2 – Euclidean path error



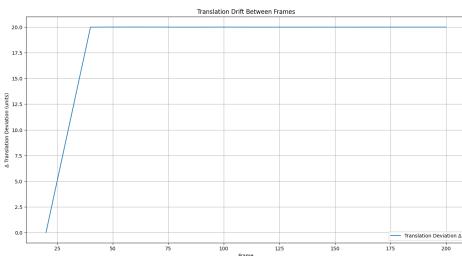
(a) Velocity 3 – Translational deviation



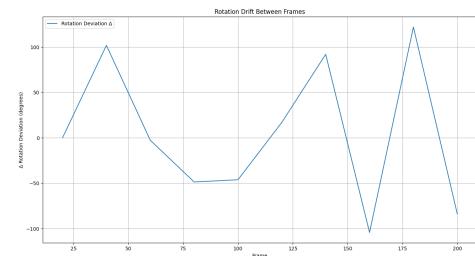
(b) Velocity 3 – Rotational deviation



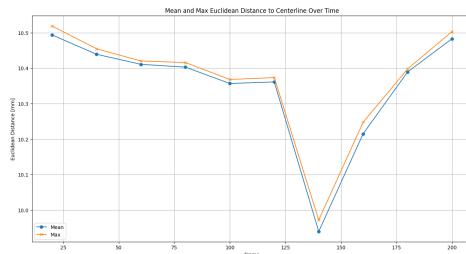
(c) Velocity 3 – Euclidean path error



(a) Velocity 4 – Translational deviation



(b) Velocity 4 – Rotational deviation



(c) Velocity 4 – Euclidean path error

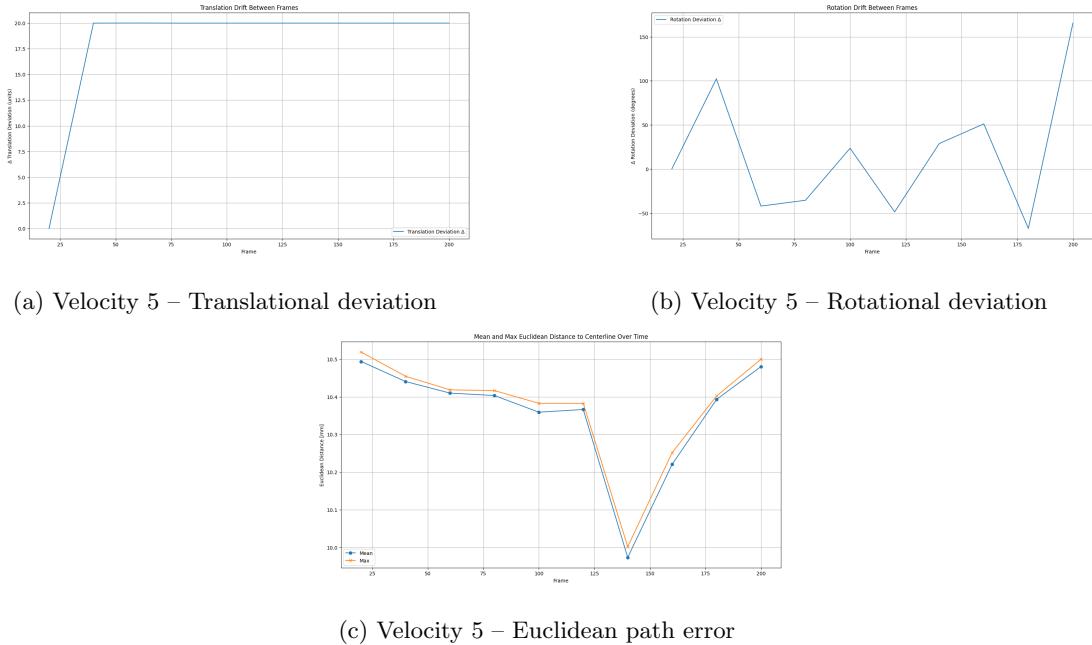


Figure 40: Impact of velocity scaling (1–5) on translational deviation, rotational deviation, and Euclidean path error during airway traversal.

## E Github

<https://github.com/RikkeB2/Broncho-Project>