SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Simulating Aeration at Birth: building an Open-Source Newborn Lung Model

TESI DI LAUREA MAGISTRALE IN
BIOMEDICAL ENGINEERING - INGEGNERIA BIOMEDICA

Luca Andriotto, 928454

Advisor:
Prof. Raffaele Dellacà

Co-advisors:
Dr. Chiara Veneroni

Academic year:
2023-2024

Abstract: During pregnancy, the fetal airways are filled with a fluid known as fetal lung fluid, which is essential for the development of airway width. Consequently, at birth, the respiratory system must expel this fluid to allow air to enter and exit, a process necessary for breathing (aeration). Fluid reabsorption begins a few days before birth through chemical processes involving sodium channels, and during natural childbirth, fluid is expelled from the mouth and nose due to the compression of the neonate's chest. In full-term infants, the likelihood of complications during aeration is very low. However, the scenario is vastly different for preterm infants, who are born before 37 weeks of gestation compared to the typical 40 weeks of a normal pregnancy. Although recruitment maneuvers have gained more interest in preterm ventilation, there is still no common medical strategy. Experimental procedures are tested on animals, presenting challenges in obtaining results due to the invasiveness of the procedures and associated ethical issues. In silico modeling of the adult lung has been useful for understanding pathophysiology and making diagnoses. Thus, the same approach could help analyze various recruitment strategies and their impact on the lung during initial aeration at birth. However, in silico models of neonatal lungs are limited to describing up to the first generation of the bronchial tree and are therefore inadequate for simulating the physiological changes that occur at birth.

Key-words: morphometric model, aeration process, lung, newborn, respiratory system

1. Introduction

During pregnancy, the fetal airways are filled with a fluid known as fetal lung fluid, which is essential for the development of airway width. Consequently, at birth, the respiratory system must expel this fluid to allow air to enter and exit, a process necessary for breathing (aeration). Fluid reabsorption begins a few days before birth through chemical processes involving sodium channels, and during natural childbirth, fluid is expelled from the mouth and nose due to the compression of the neonate's chest. In full-term infants, the likelihood of complications during aeration is very low. However, the scenario is vastly different for preterm infants, who are born before 37 weeks of gestation compared to the typical 40 weeks of a normal pregnancy.
Applying pressure at the entrance of the airways helps preterm infants remove the fluid and open the closed alveoli. Once the alveoli are recruited, less pressure is needed to keep them open and ventilate the lung. Thus, employing recruitment strategies at birth has the advantage of initiating ventilation in a more recruited lung,

using lower pressures, achieving more homogeneous lung aeration, and reducing the stress applied to the tissues. It is important to note, however, that at birth, the lung is more delicate and thus more susceptible to damage, even with routine ventilation procedures. Therefore, developing protective recruitment strategies at birth could lead to significant improvements in this area.

Although recruitment maneuvers have gained more interest in preterm ventilation, there is still no common medical strategy. Experimental procedures are tested on animals, presenting challenges in obtaining results due to the invasiveness of the procedures and associated ethical issues[7, 3]. In silico modeling of the adult lung has been useful for understanding pathophysiology and making diagnoses. Thus, the same approach could help analyze various recruitment strategies and their impact on the lung during initial aeration at birth. However, in silico models of neonatal lungs are limited to describing up to the first generation of the bronchial tree and are therefore inadequate for simulating the physiological changes that occur at birth.

These considerations are essential for developing accurate and useful models for studying lung function in neonates, especially in clinical contexts such as oscillatory ventilation. Consequently, both the anatomical and mechanical characteristics of the tissues needed to be appropriately modified to achieve a consistent model for the neonatal case in the time domain. Therefore, both the morphometric and electrical models were modified to implement the changes occurring at birth, including the adjustment of resistance and inductance values in the electrical models of the airways, as these depend on the viscosity and density of the medium passing through them (air or fetal lung fluid).

Initial attempts were made with "LTspice" and "CADENCE." Using open-source strategies for the model eliminates the need for proprietary software licenses, making the development process more accessible. The model was developed in line with data from the literature regarding both the geometry of the bronchial tree and the mechanical characteristics of its tissues, resulting in time-domain simulations of the aeration process, which allow for comparison of different ventilation strategies[11].

Aims of this projects are:

- Generate a model from neonatal CT scans to optimize the generation of airways, ensuring they adhere to the morphometric characteristics at various ages.
- Develop an open-source mechanical model that allows for the simulation of mechanical properties along with fluid dynamics (capillary pressure associated with the interface).

2. Anatomical models

Mathematical models developed for adult lungs cannot simply be scaled down to fit the lungs of newborns. In fact, newborn lungs are not simply one miniature version of adult lungs, but they present significant differences in terms of bronchial branch proportions, constituents of the airways[12], morphometric characteristics[5] and composition[4]. These differences must be taken into account when developing or adapting mathematical models to accurately represent the functioning of the lungs of the newborns. The structure of the lungs of newborns presents proportions different than that of adults. The branches of the airways they can have different sizes and arrangements. The components of lungs, like tissue and cells, can vary between newborns and adults. Morphometric studies, which analyze the shape and the structure of the lungs, show that there are differences between newborns and adults that need to be considered in the models. There are differences in the composition of lung tissue between newborns and adults who influence how the lungs function and respond to therapies.

Past work[11] considered an adult lung model linearly scaled to match newborn anatomical features. The advantage of this approach is that it respects the dimensions of trachea and bronchioles. It doesn't guarantee that the morphometric characteristics of the entire airway tree are respected. In this work, there are few airway generation parameters that can in fact be adapted, in order to better approximate the target morphometric characteristics.

2.1. Existing Infant Lungs Models

In literature, there exist models based on the ovine[7] and canine[3] anatomy.

3. Mechanical model of airways and acini

The electrical equivalent of tissues properties is modified to test the physiological change impact on aeration process.

Moreover, fetal fluid is incompressible, whereas air is not. This introduces an additional element in the electrical equivalent of air, as the compressibility of the gas is modeled through a capacitance. Additionally, it is necessary to consider the collapse of the airways. During the aeration process, an air-liquid interface is created, which in

turn generates surface tension that must be overcome to allow fluid movement within the bronchial tree. Once this surface tension is overcome, the diameters can expand, leading to an increase in lung volume.
Lutchen and Gillis [8] have developed a mechanical lung model in frequency-domain, describing airways and alveoli modules as displayed in fig. 1 and fig. 2, respectively.

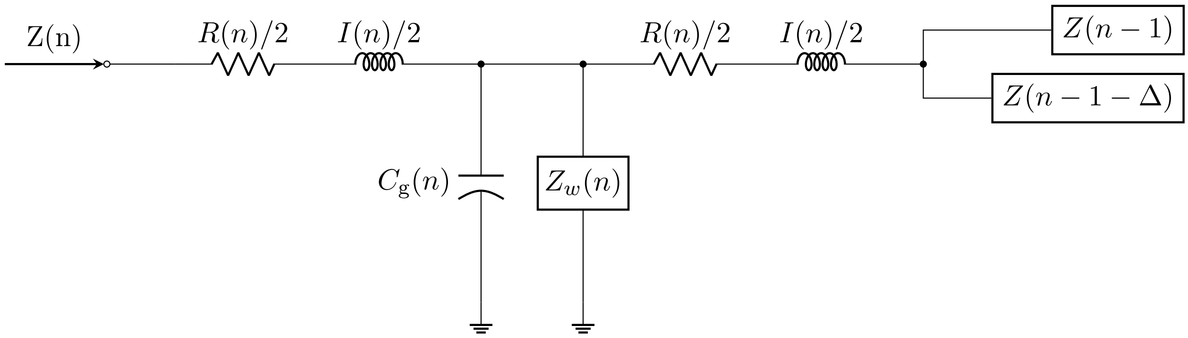


Figure 1: Impedance (Z) of a given order (n) of a single airway generation is calculated via an acoustic transmission line analysis, which accounts for shunting into gas compression in the tube ($C_{\mathrm{g}}(n)$) and into nonrigid airway walls (Z_{w})[8].

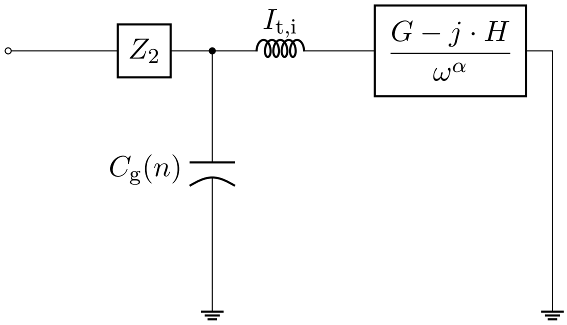


Figure 2: An alveolar-tissue element is attached to the terminal airways in the tree. There is gas compression corresponding to volume of the alveolus (C_{g}) and the tissue element is viscoelastic containing a tissue damping (G) coupled to elastance (H) to ensure a constant tissue hysteresis. R: resistance; Δ: recursion index, j: imaginary unit, $I_{\mathrm{t,i}}$: tissue inertance[8].

Mani [11] has defined a mechanical model in time-domain starting from the modules here described and properly changed.
A first implementation has been performed on «CADENCE» platform. This has the advantage of parallelism and speed. There are also some drawbacks to this approach: this framework is designed to simulate standard electrical components and it is not well-suited to develop time- and current integral-dependent components. Furthermore license is proprietary and machine specific. This limits the accessibility of of model design process.

4. Model development

Figure 3 provides a high-level overview of the main operations performed.

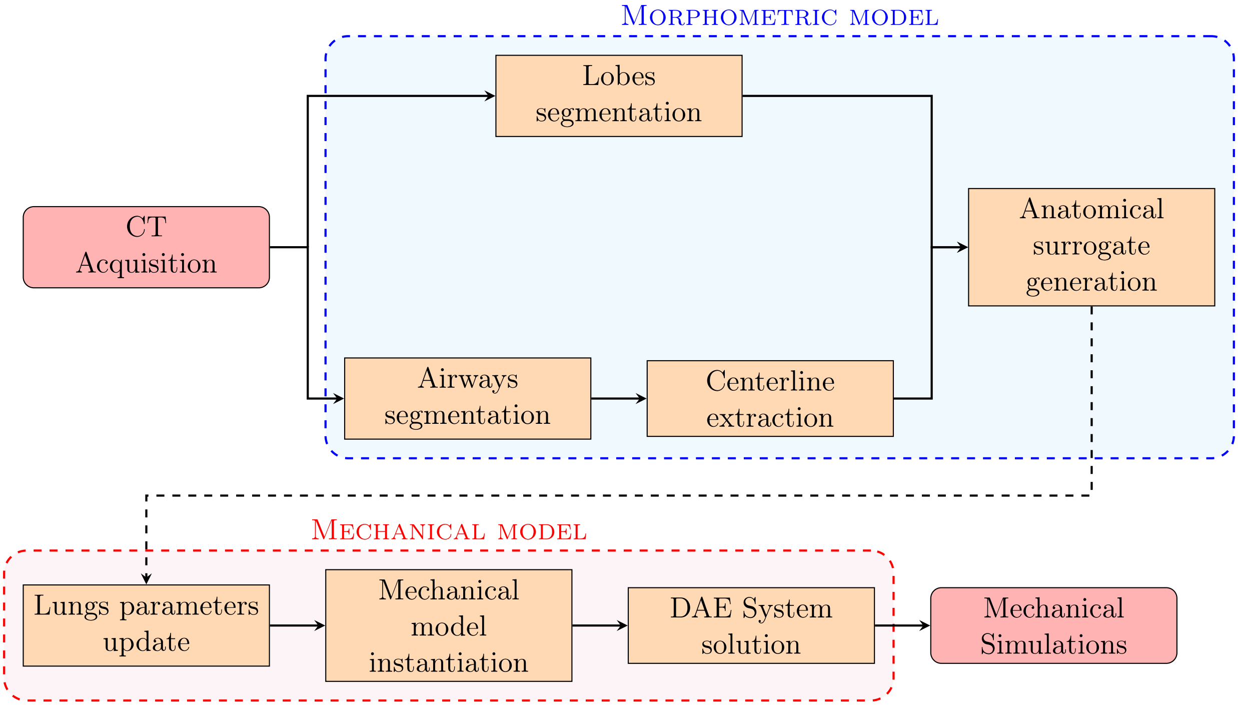


Figure 3: Data pipeline. The process begins with a *patient-specific image* (i.e. CT) of a premature newborn. The extracted data, comprising *two segmentations*, are then processed to obtain an anatomical surrogate of the airway tree. This is necessary due to scanner resolution not allowing for the discrimination and localization of small branches. From the resulting morphometric model, the *mechanical parameters* can be derived, which are essential for generating an accurate simulation model. Finally, a numerical solver for differential equations provides the final output.

4.1. Airway tree

There are different open-source platforms available for generating morphometric models. In particular:

1. AVATree (Windows-only)
2. Chaste (crossplatform) library

Due to problems related to «AVATree» source code compilation for Windows with «VisualStudio», Chaste library is selected for this project.
Chaste is a C++ open-source (BSD licensed) library developed by Oxford University. It has multiple use cases across various biomedical fields, with an emphasis on cardiac electrophysiology and cancer development[13]. It can be integrated into a C++ program or used via «User Project» (i.e. `ctest`). Specifically, the "AirwayGenerationTutorial" is considered as a first codebase and properly adapted to match newborn parameters[17].
The required input consists of two pieces of information:

- A *mesh of centerline points*. This mesh is provided in TetGen format, comprising "airways.node" and "airways.edge" files. The first file lists centerline points coordinates, respective sampled airways radius and a boolean value to indicate if the point is generative. The second one contains all the connections between pairs of points.
- Four (or five) *lobes segmentations* in STL format. These segmentations are necessary as they physically impose a limit on the growth algorithm.

4.1.1 CT Image Processing: Lung Segmentation, Centerline and Radii Extraction

«3D Slicer» is an open-source software used for CT image processing. Two extensions are installed:
«*Chest_imaging_platform*»: This extension enables semi-automatic segmentation of major airways from a single fiducial point. It can also extract adult lobes using three fiducial points per lung fissure. However, in our case, the fissures are not visible, necessitating manual intervention.
«*SlicerVMTK*»: This extension is used for extracting centerline points.

4.1.2 Generation of the Statistical Portion

The Chaste User Project reads the input files (see Section 4.1), and begins growing the anatomical surrogate from the points labeled as «generative». The algorithm operating under the hood is a modified version of the one described in [16, 1]. The generated output is available in various formats:

- vtu: Unstructured Grid (base64 encoded) format used by VTK library. It can be displayed by ParaView, an open-source viewer.
- node and edge: TetGen format. Such files are better suited for further processing.

This process is required as it is not possible to obtain high generations (aka small airways) by means of standard high-resolution CT[1].
The algorithm is based on a modified version of Tawhai, Pullan, and Hunter [16]. A uniform grid of seed points is created within each segmented lobar surface. Seed points approximately correspond to terminal bronchioles. Spacing of the seed point grid is set so that the mean volume around each of such points corresponded to the acinar volume (for adult being 187mm).
The starting points of the algorithm are the distal ends of the segmented airway centerlines. These points are referred to as growth apices.
An *adaptive threshold* on the distance between the seed points and growth apices is required to prevent spurious long airways being generated in the last few generations. Equation (1) describes such threshold:

$$T = \max(V_\mathrm{b} - n \cdot D_\mathrm{l}, 5\mathrm{mm}) \tag{1}$$

Where:
V_b is the diagonal size of the bounding box of the lobe being generated into
$D_\mathrm{l}(= V_\mathrm{b}/N)$ is the distance limit.
N is the maximum number of generations.
n is the current generation number.
With these definitions, the **growing algorithm** is described as such:

1. *Each seed point is associated to the closest growth apex within its lobe.* Seed point having a distance with respect to a growth apex greater than the aforementioned adaptive threshold is not associated to that distal end. If all distal ends are further than the threshold from the seed point, the seed point remains unlabeled.
2. *Calculation of the centroid of points assigned to each distal branch.*
3. *The plane defined by the centroid and the parent branch is used to split the points into two unequal sets.*
4. *Centroids of each of the new point sets are calculated.*

5. *For each set of points a new airway is generated* starting at the distal end and extending 40% of the distance towards the centroid of the point set.
6. *Generated branches are checked to determine whether it is terminal.* Branches whose length is less than 2mm are considered terminal. Also branches whose point set contains just a single point solely are considered terminal points. For all terminal branches their associated seed point is discarded from the global set.
7. *Iterate* until no seed point is available.

Diameters are computed by means of Equation (2).

$$\log D(x) = (x - N)\log(R_{\mathrm{d}}H) + \log(D_{\mathrm{N}}) \tag{2}$$

Where:
D is the aiway diameter.
x is the current Horsfield order.
N is the maximum Horsfield order.
D_{N} is the maximum diameter.
$R_{\mathrm{d}}H$ is the anti-log of the slope of airway diameter plotted against Horsfield order and is set to 1.15. This parameter in the code is named “DiameterRatio”.

4.2. Mechanical simulator

In order to perform simulation it is required to use an efficient differential equation solver. «DifferentialEquations.jl» wraps all available solvers (even C and Fortran ones) and it is very efficient [2, 15].
Julia is a free, open-source (MIT licensed), fast, scientific and numerical computing-oriented programming language. Its computational efficiency is comparable to that of statically-typed languages like C or Fortran. Moreover, its high-level code expressivity rivals that of languages like Python, R and MATLAB[6].
Two key features, inspired by the *Lisp Language*, are highlighted here.
Metaprogramming: Code is treated as any other Julia data structure, thus can be dynamically generated and manipulated at runtime.
Macros: They help instantiate the generated code in the body of a program.
Their importance is closely tied to the concept of Domain-Specific Languages (aka DSLs). These dialects are composed by abstractions that can be properly exploited to solve particular problems (e.g. modeling complex systems, solving differential equations).
Julia REPL has a built-in package manager (i.e. «`Pkg.jl`») used for managing project dependencies and ensuring the *repeatability* of computational setups. This is achieved by saving the required package names and commits into ‘Project.toml’ and ‘Manifest.toml’ files.

4.2.1 «`ModelingToolkit.jl`» handles Model Complexity

This Julia package encompasses all the tools necessary for model design. `ModelingToolkit.jl`» is equation-driven, requiring each system to be described by Differential-Algebraic Equations (i.e. DAEs) for subsequent solving[10]. Its built-in DSL optimizes every stage of modeling, from prototyping components to instantiating the complete system.
An acausal paradigm can be adopted, allowing users to reason in terms of *components*[14]. This modularity facilitates system extensibility compared to the causal approach, where the entire system of Differential-Algebraic Equations must be considered and manually simplified[9].
In particular, the usage of `@mtkmodel` macro enables hierarchical generation of building blocks recurring in the highest-order model (i.e. «Lungs»). Here is how information is structured within `@mtkmodel` macro.

Listing 1: `@mtkmodel`: a macro for systems prototyping.

```
@mtkmodel <name_of_model> begin
    @parameters begin
        # (Optional) Some constant (e.g. Resistance, Capacitance) ...
    end
    @components begin
        # (Optional) Some dependency system (e.g. Resistor, Capacitor) ...
    end
    @variables begin
        # (Optional) Internal variables ...
    end
```

```
        @equations begin
            # Differential Algebraic Equations describing the model's behavior.
        end
        @continuous_events begin
            # (Optional) Some callback function ...
        end
    end
```

Replicating the behavior of electrical components using this language is straightforward, once you are familiar with the syntax and understand the Differential-Algebraic Equations that represent their characteristics. Each generated system can then be composed into more complex ones, using the internal `@components` macro, thereby implementing the hierarchical structure mentioned earlier.
After describing the highest-order system, Julia compiler requires its instantiation before any simulation can be performed. This is accomplished using `@mtkbuild` macro, which minimizes the number of equations that need to be solved.

4.2.2 Blocks Description

Code modularity is directly reflected in the electrical equivalent circuit. Specifically, by encapsulating systems with the internal `@components` macro, it becomes possible to generate models of increasing complexity. This approach enables a clear separation between components belonging to different hierarchical levels and facilitate compartmentalization during the model design phase.
The following blocks are listed in a bottom-up order (from lowest to highest).

1. **Electrical components**. The simplest blocks are derived from «`ModelingToolkitStandardLibrary`», while integral-dependent ones rely on a modified mathematical block to manage both current integration and its timing correctly. Their behavior varies based on the neonatal pulmonary fluid interface.
   - *Current Integral-Dependent Inductor*: $L(t) = L_a + L_b \cdot \left(1 - \frac{\int idt}{V_{\text{FRC}}}\right)$
   - *Current Integral-Dependent Resistor*: $R(t) = R_a + R_b \cdot \left(1 - \frac{\int idt}{V_{\text{FRC}}}\right)$
   - *Diode*
   - *Inductor*
   - *Resistor*
2. **Modules**. Obtained by connecting the aforementioned components together into functional models representing a physiological structure.
   - *Alveolus*
   - *Airway*. It has a similar behavior with respect to a transmission line.
3. **Lungs**. Highest order model as it is a combination of alveoli and airways.

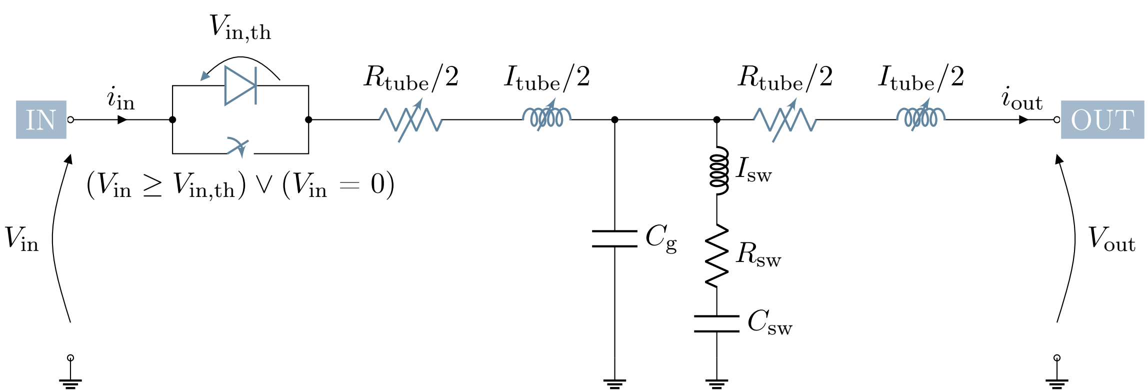


Figure 4: Airway equivalent circuit. In blue: all current integral-dependent components.

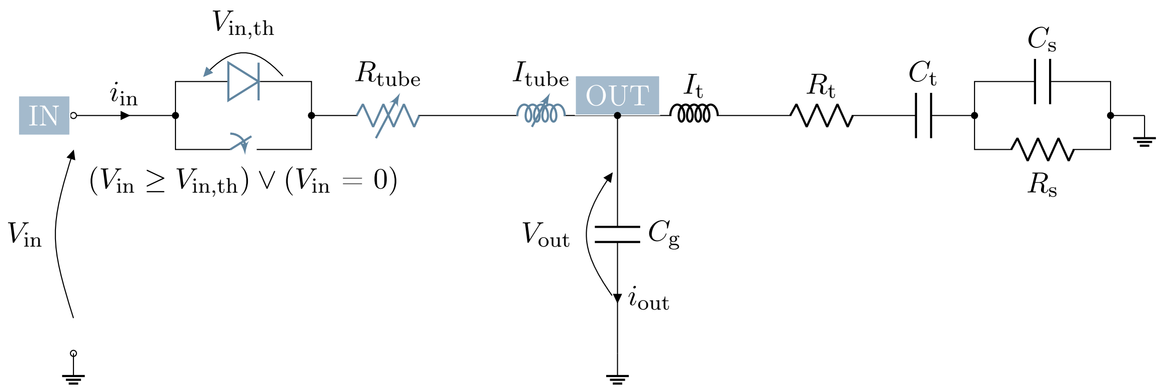


Figure 5: Alveolus equivalent circuit. In blue: all current integral-dependent components.

4.2.3 Callbacks' Role in State Variables Discontinuity Handling

Not all characteristics of electrical components can be defined solely by DAEs. Voltages or currents may suddenly change, triggered by a circuit event. In such cases, *continuous callback functions* can be employed to appropriately alter the value of state variables. These callbacks consist of two functions:

- `condition`: Specifies the event to be tested.
- `affect`: Defines how the state variable(s) should be changed.

The component-based approach allows for the definition of callbacks directly within the (sub)system being modeled.

4.2.4 Model Testing on A Subtree

Simulations are executed starting from a subnet, as the full circuit (comprising over 50k modules) requires more memory space than typically available on a common laptop.

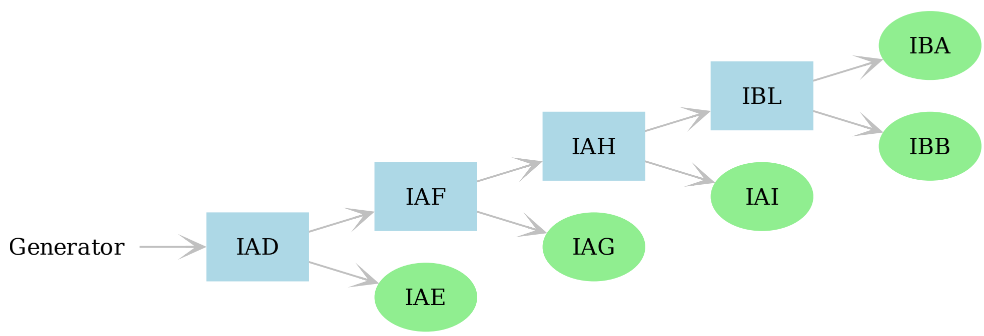


Figure 6: The simulated subtree. Airways are represented in light blue, alveoli in light green.

5. Results and Validations

5.1. Anatomical Surrogate

Anatomy of the airway tree can be displayed by using ParaView.

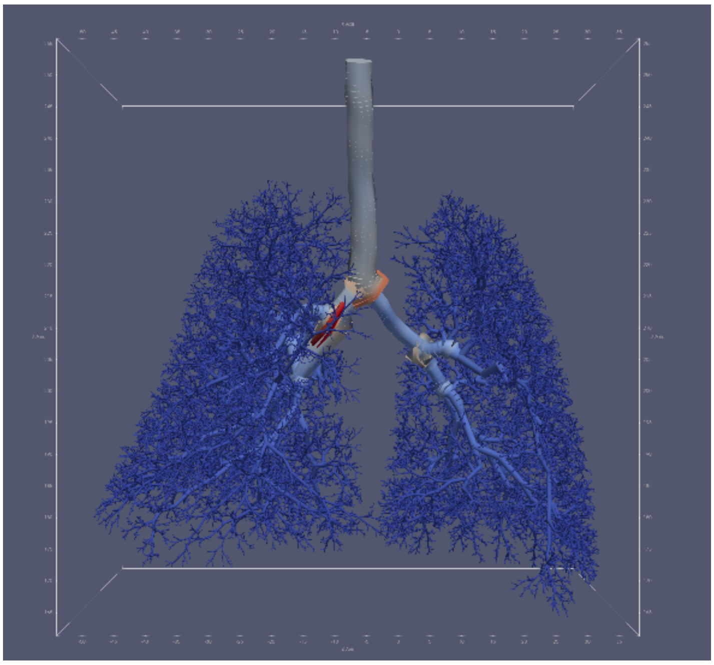
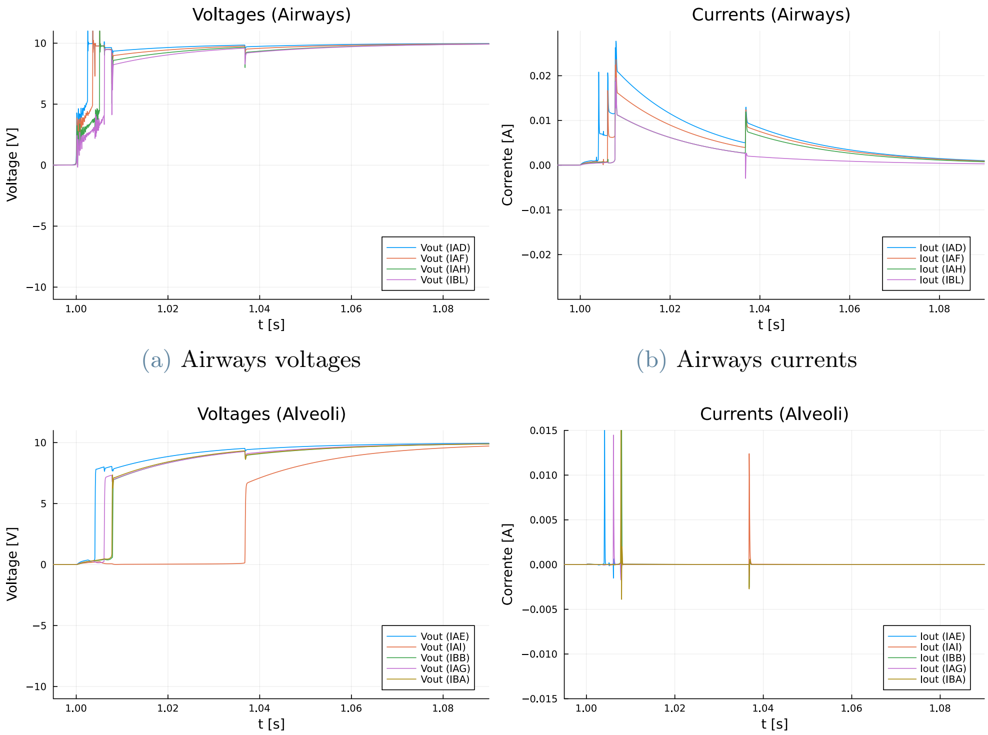


Figure 7: 3D rendered airway tree.

5.2. Mechanical Simulation

Different tests are displayed. They consider as thresholds value "vin_th" for each module diodes.
The first result describes modules voltages and currents, given a over-threshold stimulation for all of them.



(a) Airways voltages

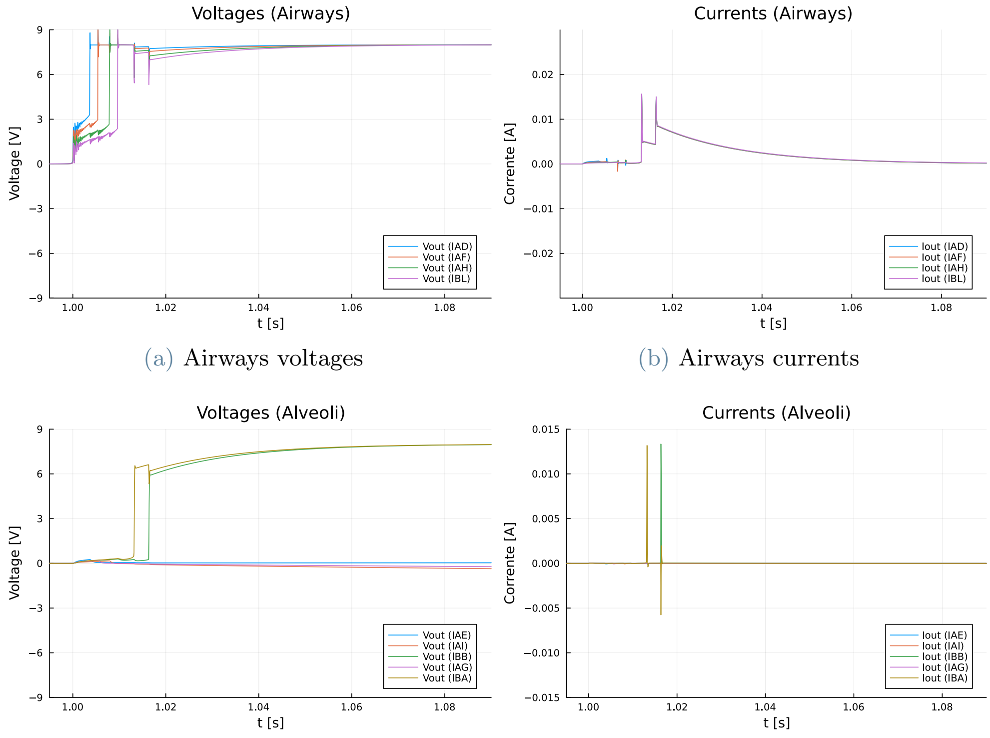
(b) Airways currents

(c) Alveoli voltages

(d) Alveoli currents

Figure 8: (Electrically equivalent) mechanical simulation for alveoli and aiways. Step amplitude is 10V.

The second one shows modules voltages and currents, given a under-threshold stimulation for some alveoli (IAE, IAI and IAG).



(a) Airways voltages

(b) Airways currents

(c) Alveoli voltages

(d) Alveoli currents

Figure 9: (Electrically equivalent) mechanical simulation for alveoli and aiways. Step amplitude is 8V.

6. Discussion and conclusion

As future development it is possible to edit properly airway generation parameters in order to better approximate the target morphometric characteristics, so to be able to produce a patient-specific anatomical model. This allows for gathering more accurate mechanical parameters for the simulation phase.

References

[1] Rafel Bordas et al. "Development and analysis of patient-based complete conducting airways models". en. In: *PLoS One* 10.12 (Dec. 2015), e0144105. DOI: 10.1371/journal.pone.0144105.

[2] *DifferentialEquations.jl Documentation*. URL: https://docs.sciml.ai/DiffEqDocs/stable/.

[3] Jacob Herrmann, Merryn H Tawhai, and David W Kaczka. "Regional gas transport in the heterogeneous lung during oscillatory ventilation". en. In: *J. Appl. Physiol.* 121.6 (Dec. 2016), pp. 1306–1318. DOI: 10.1152/japplphysiol.00097.2016.

[4] A A Hislop and S G Haworth. "Airway size and structure in the normal fetal and infant lung and the effect of premature delivery and artificial ventilation". en. In: *Am. Rev. Respir. Dis.* 140.6 (Dec. 1989), pp. 1717–1726. DOI: 10.1164/ajrccm/140.6.1717.

[5] K Horsfield et al. "Growth of the bronchial tree in man". en. In: *Thorax* 42.5 (May 1987), pp. 383–388. DOI: 10.1136/thx.42.5.383.

[6] *Julia Documentation*. URL: https://docs.julialang.org/.

[7] Ahmed M Al-Jumaily et al. "Pressure oscillation delivery to the lung: Computer simulation of neonatal breathing parameters". en. In: *J. Biomech.* 44.15 (Oct. 2011), pp. 2649–2658. DOI: 10.1016/j.jbiomech.2011.08.012.

[8] Kenneth R Lutchen and Heather Gillis. “Relationship between heterogeneous changes in airway morphometry and lung resistance and elastance”. en. In: *J. Appl. Physiol.* 83.4 (Oct. 1997), pp. 1192–1201. DOI: 10.1152/jappl.1997.83.4.1192.

[9] Yingbo Ma. *Scaling Equation-based Modeling to Large Systems.* 2024. URL: https://www.youtube.com/watch?v=c-bZ2v1uF14.

[10] Yingbo Ma et al. *ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling.* 2021. arXiv: 2103.05244 [cs.MS].

[11] Elisa Mani. *An in-silico morphometric model of the respiratory system for simulating the dynamics of lung aeration at birth during respiratory support.* Master Thesis. June 2020. URL: https://www.politesi.polimi.it/handle/10589/165063.

[12] P J Merkus, A A ten Have-Opbroek, and P H Quanjer. “Human lung growth: a review”. en. In: *Pediatr. Pulmonol.* 21.6 (June 1996), pp. 383–397. DOI: 10.1002/(sici)1099-0496(199606)21:6%3C383::aid-ppul6%3E3.0.co;2-m.

[13] Gary R. Mirams et al. “Chaste: An Open Source C++ Library for Computational Physiology and Biology”. In: *PLOS Computational Biology* 9.3 (Mar. 2013), pp. 1–8. DOI: 10.1371/journal.pcbi.1002970. URL: https://doi.org/10.1371/journal.pcbi.1002970.

[14] *ModelingToolkit.jl Documentation.* URL: https://docs.sciml.ai/ModelingToolkit/stable/.

[15] Chris Rackauckas, Christopher, and Qing Nie. “DifferentialEquations.jl–a performant and feature-rich ecosystem for solving differential equations in Julia”. In: *Journal of Open Research Software* 5.1 (2017).

[16] M. Howatson Tawhai, A. J. Pullan, and P. J. Hunter. “Generation of an Anatomically Based Three-Dimensional Model of the Conducting Airways”. In: *Annals of Biomedical Engineering* 28.7 (July 2000), pp. 793–802. ISSN: 1573-9686. DOI: 10.1114/1.1289457. URL: https://doi.org/10.1114/1.1289457.

[17] *TestAirwayGeneration (Chaste Tutorial).* URL: https://chaste.github.io/docs/user-tutorials/airwaygeneration/.

Abstract in lingua italiana

Durante la gravidanza, le vie aeree del feto sono riempite di un liquido noto come liquido polmonare fetale, essenziale per lo sviluppo della larghezza delle vie aeree. Di conseguenza, alla nascita, il sistema respiratorio deve espellere questo liquido per permettere l'entrata e l'uscita dell'aria, un processo necessario per la respirazione (aerazione). Il riassorbimento del liquido inizia qualche giorno prima del parto attraverso processi chimici che coinvolgono i canali del sodio e, durante il parto naturale, il liquido viene espulso dalla bocca e dal naso grazie alla compressione del torace del neonato. Nei neonati a termine, la probabilità di complicazioni durante l'aerazione è molto bassa. Tuttavia, lo scenario è molto diverso per i neonati pretermine, che nascono prima delle 37 settimane di gestazione rispetto alle tipiche 40 settimane di una gravidanza normale. Sebbene le manovre di reclutamento abbiano suscitato un maggiore interesse nella ventilazione dei pretermine, non esiste ancora una strategia medica comune. Le procedure sperimentali vengono testate sugli animali, presentando sfide nel ottenere risultati a causa dell'invasività delle procedure e dei problemi etici associati. La modellizzazione in silico del polmone adulto è stata utile per comprendere la patofisiologia e fare diagnosi. Pertanto, lo stesso approccio potrebbe aiutare ad analizzare le diverse strategie di reclutamento e il loro impatto sul polmone durante l'aerazione iniziale alla nascita. Tuttavia, i modelli in silico dei polmoni neonatali sono limitati alla descrizione fino alla prima generazione dell'albero bronchiale e, pertanto, non sono adeguati a simulare i cambiamenti fisiologici che avvengono alla nascita.

Parole chiave: modello morfometrico, processo di aerazione, polmone, neonato, sistema respiratorio