

Respirascan: Using Breathing Patterns As a Biomarker to Diagnose Restrictive Lung Diseases

Jayveer Kochhar

Received February 19, 2025

Accepted August 23, 2025

Electronic access October 31, 2025

Respiratory issues like Interstitial Lung Diseases (ILDs), Occupational and Environmental Lung Diseases, and Seronegative Spondyloarthropathies (SSA) are difficult to identify because their symptoms closely resemble other musculoskeletal and pulmonary disorders. Traditional diagnostic methods—spirometry, X-rays, and CT scans—are costly, expose patients to radiation, and are not widely available in rural areas. This study presents RespiraScan, a portable, non-invasive diagnostic device that analyzes chest movement using Inertial Measurement Unit (IMU) sensors and applies machine learning for classification. Chest expansion data is captured in three dimensions, Kalman filtered to reduce noise, and analyzed using Long Short-Term Memory (LSTM) neural networks. The system classifies three categories of restrictive lung diseases: Ankylosing Spondylitis, ILDs, and Occupational Lung Diseases, achieving a 90% accuracy post-processing. The results support the potential of wearable, AI-driven devices for real-time respiratory evaluation and early disease detection.

Keywords: LSTM Model, S3-Microcontroller, Lung Disease Prediction Key for ILD diagnosis. Differentiate patterns like UIP, NSIP, HP. AI tools enhance interpretation. Body Plethysmography: Measures TLC, confirming restriction severity. Essential in ILD management.

Introduction

Millions of individuals suffer from respiratory diseases such as Interstitial Lung Diseases (ILDs), Occupational and Environmental Lung Diseases, and Seronegative Spondyloarthropathies (SSA), including Ankylosing Spondylitis (AS).¹ These conditions lead to restrictive lung disease, characterized by limited chest wall expansion and reduced lung compliance. Early diagnosis remains difficult due to symptom overlap with other respiratory and musculoskeletal disorders. Standard diagnostic tools—spirometry, imaging (X-ray, CT), and blood tests like HLA-B27—face significant limitations: low sensitivity, high costs, radiation risks, and dependence on specialized facilities. Manual methods like tape-measured chest expansion are error-prone, non-standardized, and insensitive to subtle respiratory changes. Consequently, clinicians often struggle to distinguish between benign back pain and serious respiratory pathology. Spirometry is the most common diagnostic test but is unsuitable for children or patients with neuromuscular limitations. Imaging provides structural details but involves radiation and high cost. Manual measurements are widely used yet inaccurate. Prototype wearable sensors have emerged but face challenges such as size, lack of breath sound integration, and calibration needs.² This study proposes a solution via a portable IMU-based device that measures multidirectional chest displacement. Signal accuracy is enhanced through Kalman filtering, quaternions, and trapezoidal

integration. A secondary module performs lung sound analysis using Mel Frequency Cepstral Coefficients (MFCCs) to detect wheezing, crackles, or diminished sounds. A Long Short-Term Memory (LSTM) model processes this data for automatic disease classification.³ The goal is to build a remote, non-invasive device that combines mechanical chest motion and respiratory sound analysis. By integrating these dimensions, the system enables comprehensive detection of restrictive lung diseases.

The device benefits both healthcare providers and patients. Clinicians gain a quantifiable, repeatable diagnostic tool, improving diagnostic precision. Patients—especially in rural settings—gain affordable radiation-free monitoring. The system also supports telemedicine through remote respiratory pattern tracking. Its machine learning foundation ensures continuous improvement, making it suitable for use in clinical practice, home care, and digital health ecosystems.

Below are the disease that can be diagnosed using the breathing pattern

Idiopathic Pulmonary Fibrosis (IPF)

IPF presents in older adults with progressive dyspnea, dry cough, and “Velcro” crackles. HRCT shows a UIP pattern with subpleural honeycombing and basal reticulation. Pathology reveals patchy fibrosis and fibroblast foci. Prognosis is poor with a median survival of 35 years.

Non-Specific Interstitial Pneumonia (NSIP)

NSIP occurs in younger patients and may be linked to autoimmune disease. HRCT shows ground-glass opacities with basal predominance and subpleural sparing. Histology shows uniform interstitial inflammation/fibrosis. Prognosis is better than IPF; many respond to steroids.

Hypersensitivity Pneumonitis (HP)

HP is caused by immune response to inhaled antigens. Chronic HP mimics IPF but often affects upper lobes with mosaic attenuation. Histology shows bronchiolocentric inflammation and granulomas. Avoiding exposure can halt or reverse disease progression.

Desquamative Interstitial Pneumonia (DIP)

DIP affects smokers, presenting with ground-glass opacities on HRCT. Alveoli are filled with pigmented macrophages. Fibrosis is mild. With smoking cessation and steroids, prognosis is good.

Acute Interstitial Pneumonia (AIP)

AIP leads to rapid respiratory failure. CT shows bilateral ground-glass opacities and consolidation. Pathology reveals diffuse alveolar damage (DAD). Mortality is high, though survivors may recover partially or develop fibrosis.

Cryptogenic Organizing Pneumonia (COP)

COP mimics pneumonia but does not improve with antibiotics. HRCT shows patchy peripheral consolidations. Pathology shows Masson bodies in alveoli. Responds well to steroids, though relapses are common.

Sarcoidosis

Sarcoidosis affects multiple organs; lungs are most common. Non-caseating granulomas and bilateral hilar lymphadenopathy are typical. HRCT shows upper lobe nodules. Prognosis is variable; many cases resolve spontaneously.

Lymphoid Interstitial Pneumonia (LIP)

LIP presents gradually and is often associated with autoimmune diseases. HRCT shows ground-glass opacities with cysts. Histology reveals diffuse lymphoid infiltration. Risk of lymphoma transformation exists but is low.

Respiratory Bronchiolitis-Associated ILD (RB-ILD)

RB-ILD occurs in smokers with mild symptoms. HRCT shows upper lobe ground-glass opacities and centrilobular nodules. Pigmented macrophages are seen histologically. Improves with smoking cessation.

Asbestosis

Results from long-term asbestos exposure. Presents decades later with dyspnea and lower lobe fibrosis. HRCT shows subpleural fibrosis and pleural plaques. Risk of lung cancer and mesothelioma is increased.

Diagnostic Technologies

HRCT: Key for ILD diagnosis. Differentiate patterns like UIP, NSIP, HP. AI tools enhance interpretation.

Body Plethysmography: Measures TLC, confirming restriction severity. Essential in ILD management.

Spirometry & DLCO: FVC reduced, FEV₁/FVC normal/high. DLCO reflects gas exchange efficiency.

Decline in FVC or DLCO indicates progression.⁴

Pulse Oximetry & ABG: Oximetry detects exertional desaturation. ABG confirms hypoxemia or hypercapnia in advanced disease.

Wearables: Chest straps, oximeters, and home spirometry aid remote monitoring. Promising for early exacerbation detection and patient engagement.⁵

Methodology

Why I Chose to Use an IMU

In designing RespiraScan, the fundamental requirement was to measure chest wall movement accurately and externally in a non-invasive, real-time manner. Restrictive lung diseases such as Idiopathic Pulmonary Fibrosis (IPF), Asbestosis, and Hypersensitivity Pneumonitis cause progressive stiffening of the lungs and chest wall, leading to reduced chest expansion. Capturing these subtle biomechanical changes without exposing the patient to radiation, discomfort, or expensive clinical machinery was critical to making RespiraScan practical for early diagnosis and point-of-care screening.

Traditional diagnostic tools like CT scans, MRI, and even spirometry either require bulky hospital equipment, expose patients to radiation, or do not provide direct insights into mechanical chest wall dynamics. Hence, a method that could externally track displacement and orientation of the chest during the breathing cycle, accurately and wirelessly, was essential.

An Inertial Measurement Unit (IMU) emerged as the ideal solution because it provides real-time data on linear acceleration, angular velocity, and 3D orientation. IMUs are capable of

continuous dynamic tracking of physical movement in all three spatial dimensions (x , y , z) and are compact, lightweight, and low-power perfect for wearable and patient-friendly designs.

Secondly, IMUs are lightweight, compact, low-power, and affordable, enabling the development of a wearable, wireless system. Unlike imaging modalities (CT, MRI) or spirometry, which require bulky equipment and clinical settings, IMUs allow bedside, ambulatory, or home-based monitoring, aligning perfectly with the point-of-care goals of RespiraScan.

Additionally, IMUs allowed RespiraScan to strategically position multiple sensors over anatomical landmarks (e.g., sternum, lateral chest wall, scapula) to capture regional variations in chest expansion, which is critical for distinguishing different types of ILDs and chest wall abnormalities.

Thus, choosing IMUs addressed the twin needs of the project: accurate external chest wall movement measurement and portable, cost-effective respiratory monitoring, aligning perfectly with the vision of RespiraScan as a transformative point-of-care diagnostic tool.

The Problem of Gimbal Lock and the Need for Quaternions

When measuring 3D chest movement, it is not enough to track just linear displacement; rotational orientation must also be accurately captured. However, traditional methods that use Euler angles (pitch, roll, yaw)

for representing rotations are fundamentally limited due to a phenomenon called gimbal lock.

Gimbal lock occurs when two of the three rotation axes align, causing a loss of one degree of freedom. In simpler terms, when certain rotations happen (especially large-angle motions), the system becomes unable to distinguish between two rotational directions. This is catastrophic for high-precision medical measurements because it leads to ambiguities and errors in orientation tracking, exactly where accurate breathing motion detection is critical.

To solve this, RespiraScan adopts quaternion mathematics for representing orientation. Quaternions are four-dimensional hypercomplex numbers that encode 3D rotations without suffering from gimbal lock. They allow smooth, continuous, and robust tracking of chest wall rotations during breathing, regardless of how the body moves.

Moreover, quaternions:

- Avoid singularities (no sudden jumps in orientation data).
- Enable efficient sensor fusion algorithms (combining accelerometer, gyroscope, and magnetometer data).
- Simplify interpolation and integration of rotations, which is crucial for deriving displacement from IMU data

Sensor Bias, Sensor Bias Drift, and Gravity Estimation

In any IMU-based system like **RespiraScan**, ensuring accurate measurement of orientation and displacement is critically dependent on the quality of raw sensor data. However, several fundamental challenges arise in practice, namely **sensor bias**, **sensor bias drift**, and the correct **estimation of gravity**, which must be carefully addressed before higher-level algorithms like the Mahony filter can function effectively.

Sensor Bias

Sensor bias refers to a systematic error where a sensor's output consistently deviates from the true value by a constant offset, even when no motion is present. In an IMU:

- The **accelerometer bias** causes incorrect readings of acceleration, even at rest.
- The **gyroscope bias** results in nonzero angular velocity outputs when the sensor is stationary. Mathematically, if the true measurement is x_{true} and the bias is b , the sensor reports:

$$x_{\text{measured}} = x_{\text{true}} + b$$

This constant offset leads to errors that accumulate over time when integrating measurements to estimate velocity, displacement, or orientation.

Sensor Bias Drift

Bias drift refers to the time-varying change in the sensor bias. While bias is initially constant, real-world sensors experience slow fluctuations due to:

- Temperature changes
- Aging of sensor components
- Mechanical stress
- Environmental factors

Mathematically, the bias becomes a function of time:

$$b = b(t)$$

leading to:

$$x_{\text{measured}}(t) = x_{\text{true}}(t) + b(t)$$

Bias drift causes cumulative orientation errors and affects the long-term stability of IMU measurements, making its correction critical for accurate motion tracking.

Gravity Estimation

A fundamental task in IMU-based orientation tracking is the estimation of the direction of gravity from accelerometer measurements.

Ideally, when stationary, the accelerometer should measure only gravity:

$$\mathbf{a}_{measured} = \mathbf{g}$$

where \mathbf{g} is the gravitational acceleration vector. In the global (world) frame, this is typically:

$$\mathbf{g}_{ref} = (0, 0, 1)$$

However, due to sensor bias, noise, and dynamic motion, the measured acceleration is distorted:

$$\mathbf{a}_{measured} = \mathbf{g} + \mathbf{a}_{motion} + \mathbf{b}_{accel} + \mathbf{n}$$

where:

- \mathbf{a}_{motion} is the actual motion-induced acceleration,
- \mathbf{b}_{accel} is the accelerometer bias,
- \mathbf{n} is random noise.

In RespiraScan, because thoracic movement is minimal, even small dynamic accelerations must be distinguished from gravitational components, necessitating accurate orientation-based gravity estimation.

Expected Gravity Vector in the Sensor Frame

To correct the gyroscope's drift and apply feedback control in the Mahony filter, we require the predicted direction of gravity *in the sensor (body) frame*.

This is achieved by rotating the known global gravity vector \mathbf{g}_{ref} into the sensor frame using the current orientation quaternion \mathbf{q} :

$$\hat{\mathbf{g}} = \mathbf{q} \otimes (0, \mathbf{g}_{ref}) \otimes \mathbf{q}^{-1}$$

- $\hat{\mathbf{g}}$ is the expected gravity direction in sensor coordinates,
- \mathbf{q} is the estimated orientation quaternion of the sensor with respect to the global frame,
- \mathbf{q}^{-1} is its inverse (i.e., the quaternion conjugate),
- \otimes denotes quaternion multiplication,
- $(0, \mathbf{g}_{ref})$ is a pure quaternion (zero scalar part) representing the global gravity vector.

This rotated vector $\hat{\mathbf{g}}$ is compared with the normalized accelerometer reading $a_{measured}$ to compute a corrective error:

$$e = a_{measured} \times \hat{\mathbf{g}}$$

This error drives the PI controller in the Mahony filter, enabling accurate drift correction and alignment of the estimated orientation with gravity essential for precise displacement estimation in patients with restricted chest motion.

Quaternion Operations Explained

A quaternion q can be represented as:

$$q = (q_w, q_x, q_y, q_z)$$

where q_w is the scalar (real) part and (q_x, q_y, q_z) is the vector (imaginary) part. The inverse (or conjugate) of a unit quaternion is:

$$q^{-1} = (q_w, -q_x, -q_y, -q_z)$$

Quaternion multiplication between $p = (p_w, \mathbf{p}_v)$ and $q = (q_w, \mathbf{q}_v)$ is defined as:

$$p \otimes q = (p_w q_w - \mathbf{p}_v \cdot \mathbf{q}_v, p_w \mathbf{q}_v + q_w \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v)$$

where:

- \cdot denotes the dot product,
- \times denotes the cross product.

Thus, to compute $\hat{\mathbf{g}}$:

1. Express the global gravity $(0, 0, 1)$ as a pure quaternion $(0, 0, 0, 1)$,
2. Rotate it using the quaternion inverse and the estimated orientation quaternion.

Physical Meaning

If the IMU is perfectly aligned with the global frame, it will measure gravity as $(0, 0, 1)$. When the IMU tilts, $\hat{\mathbf{g}}$ indicates how gravity is seen from the IMU's perspective.

Comparing $\hat{\mathbf{g}}$ with the actual measured acceleration enables computation of the orientation error for feedback correction in the Mahony filter.

Mahony Filter: Overview, Mathematical Derivation, and Implementation

The Mahony filter is a sensor fusion algorithm used to estimate orientation using data from an Inertial Measurement Unit (IMU), which typically contains a gyroscope and an accelerometer. It is well-suited for environments where magnetic disturbances are minimal, and hence operates efficiently without a magnetome-

ter. Compared to the Madgwick filter, Mahony performs more stably in low-noise settings and incorporates a Proportional-Integral (PI) control loop that corrects gyroscopic drift using accelerometer-based reference gravity.

Accurate orientation estimation is essential when transforming accelerometer data into meaningful positional or biomechanical metrics. Two widely used filters for this purpose are the Mahony and Madgwick filters, both of which fuse gyroscope and accelerometer data (and optionally magnetometer data) to estimate the orientation quaternion. In this study, we selected the Mahony filter based on several empirical and theoretical considerations.

Madgwick Filter Overview The Madgwick filter is an orientation filter that uses a gradient descent algorithm to minimize the error between the measured direction of gravity (or the magnetic field) and the direction predicted by the current orientation quaternion. Its main advantages include:

- Rapid convergence even in high-dynamic conditions.
- Effective for applications where magnetic data is available and noisy environments are expected.
- Computational efficiency compared to Kalman filters.

However, the Madgwick filter’s performance can degrade in environments with low-frequency, low-noise conditions where its gradient descent optimizer may overcorrect, especially when accelerometer data are relatively clean and trustworthy.

Mahony Filter Overview and Justification

The Mahony filter is a nonlinear complementary filter that uses proportional and integral feedback to correct orientation drift using accelerometer data (and optionally magnetometer data). Unlike Madgwick, it performs exceptionally well in low-noise environments where accelerometer readings closely reflect gravity. Key reasons for choosing Mahony include

- Better performance in low-noise environments (as in our controlled test setup).
- Less aggressive correction behavior, reducing noise amplification.
- Simpler tuning via proportional-integral gains (K_P, K_I).

For our application in measuring subtle chest wall displacements, where environmental and user-generated motion noise are minimized, the Mahony filter provides smoother and more stable estimates of orientation, making it the more suitable choice.

Mathematical Formulation

Let $q = [q_0, q_1, q_2, q_3]$ be the unit quaternion representing the current orientation. The goal is to use the angular velocity ω and measured acceleration \mathbf{a} to update q while accounting for drift.

Step 1: Compute Error Vector

To correct for drift in orientation estimation, the Mahony filter computes an error vector that represents the discrepancy between the measured and expected direction of gravity. When the IMU is relatively stationary or undergoing quasi-static motion (as is the case in *Respir Scan* during normal breathing), the accelerometer primarily measures the direction of gravity. The raw acceleration vector \mathbf{a} is therefore normalized to obtain a unit vector:

$$\mathbf{a}_{\text{meas}} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$$

Here:

- \mathbf{a} is the raw acceleration measured in the sensor (body) frame,
- \mathbf{a}_{meas} is the estimated direction of gravity derived from this measurement.

Simultaneously, the Mahony filter uses the current orientation quaternion q to rotate the reference gravity vector $\mathbf{g}_{\text{ref}} = (0, 0, 1)$ from the global frame into the sensor frame, yielding the predicted gravity direction:

$$\hat{\mathbf{g}} = q \otimes (0, \mathbf{g}_{\text{ref}}) \otimes q^{-1}$$

where:

- $\hat{\mathbf{g}}$ is the predicted unit gravity vector in the sensor frame,
- \otimes denotes quaternion multiplication,
- $(0, \mathbf{g}_{\text{ref}})$ is the reference gravity expressed as a pure quaternion.

The orientation error is then computed as the cross product of the measured and predicted vectors:

$$\mathbf{e} = \mathbf{a}_{\text{meas}} \times \hat{\mathbf{g}}$$

This error vector \mathbf{e} represents the rotation axis required to align the predicted gravity direction with the measured one. It is directly used to correct the gyroscope’s drift via a proportional-integral (PI) controller in the Mahony filter’s next step.

In the context of *RespiraScan*, accurate computation of this error vector is crucial, as even minor discrepancies in orientation can lead to significant errors in estimating displacement during breathing cycles.

Mahony Filter with Large Error Vector from Acceleration Mismatch

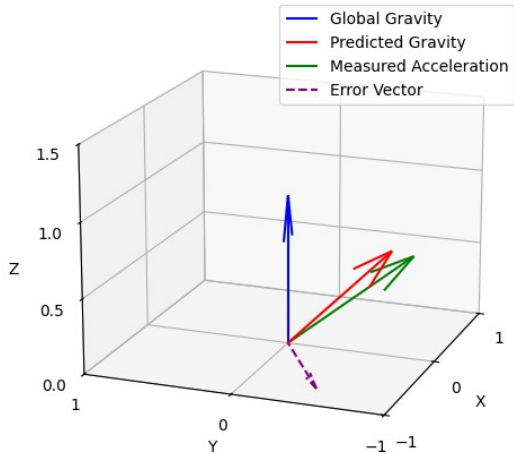


Fig. 1 Quaternion rotation of gravity vector and Mahony filter error vector.

Step 2: PI Control Law Once the error vector \mathbf{e} has been computed, it is used to correct the gyroscope’s angular velocity reading, which would otherwise accumulate drift over time.

This correction is applied through a proportional-integral (PI) controller, a fundamental concept in control systems. A PI controller continuously adjusts an estimate based on both:

- the **proportional** term which reacts immediately to the current error, and
- the **integral** term which accumulates past error over time to eliminate long-term bias or drift. In the context of Mahony filtering, the corrected angular velocity is computed as:

$$\omega_{\text{corr}} = \omega_{\text{gyro}} + K_P \cdot \mathbf{e} + K_I \cdot \int \mathbf{e} dt$$

where:

- ω_{gyro} is the raw angular velocity from the gyroscope,
- K_P and K_I are proportional and integral gains respectively,
- \mathbf{e} is the orientation error vector from Step 1.

The **proportional term** ($K_P \mathbf{e}$) provides immediate correction in the direction needed to reduce the error. It effectively nudges the orientation estimate to match the measured gravity direction.

The **integral term** ($K_I \int \mathbf{e} dt$) compensates for slow biases in the gyroscope or consistent environmental misalignments, by accounting for the accumulated error over time.

In RespiraScan, we empirically set the gain values to $K_P = 1.2$ and $K_I = 0.005$. These values were tuned to offer a fast response without overshooting or introducing instability, while still compensating for long-term drift. The low integral gain prevents wind-up and jitter due to minor sensor fluctuations, which is essential for accurately capturing the subtle breathing patterns.

Quaternion update The corrected angular velocity ω_{corr} is then used to compute the quaternion’s time derivative:
The quaternion derivative is given by:

$$\dot{q} = \frac{1}{2} q \otimes (0, \omega_{\text{corr}})$$

which is used in a numerical integrator to update the orientation estimate:

$$q_{t+\Delta t} = q_t + \dot{q} \Delta t$$

Finally, the updated quaternion is normalized to maintain unit length, which is essential to preserve its validity as a rotation representation.

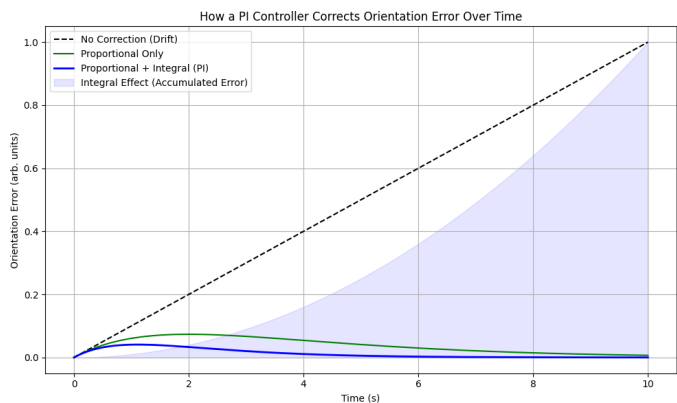


Fig. 2 Conceptual behavior of a PI controller correcting orientation error. The proportional term immediately reduces the present error, while the integral term accumulates past error over time. Together, they enable the Mahony filter to converge to accurate orientation even under slow drift and noise.

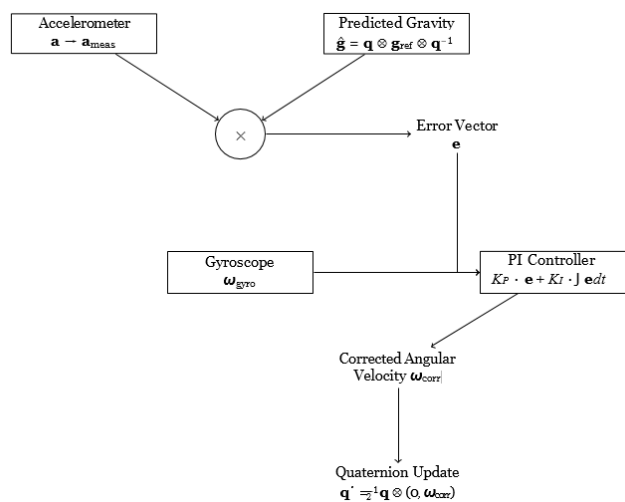


Fig. 3 Mahony filter: computing error vector and correcting angular velocity using a PI controller.

Step 3: Quaternion Integration To update the IMU's orientation over time, the Mahony filter integrates the angular velocity into the quaternion representation. Angular velocity, as measured by the gyroscope, describes how the orientation is changing in real time.

First, the angular velocity vector $\omega = [\omega_x, \omega_y, \omega_z]$ is expressed as a pure quaternion:

$$\omega_q = \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

The quaternion derivative is then computed using the Hamilton product (quaternion multiplication):

$$\dot{q} = \frac{1}{2} q \otimes \omega_q$$

This equation models the rate of change of the orientation in response to angular velocity. It is conceptually analogous to how linear velocity updates position but in this case, we are updating rotation.

Numerical Integration To propagate orientation forward in time, this derivative is integrated numerically. The simplest method is forward Euler integration:

$$q_{t+\Delta t} = q_t + \dot{q} \Delta t$$

Normalization Since quaternions must remain unit-norm to validly represent rotation, the result is renormalized at each step:

$$q_{t+\Delta t} \leftarrow \frac{q_{t+\Delta t}}{\|q_{t+\Delta t}\|}$$

This integration step ensures that the orientation is continuously updated in a smooth and physically consistent manner.

Application in RespiraScan In RespiraScan, the angular velocity is typically low and changes gradually. Nonetheless, precise integration is crucial small errors in orientation can lead to significant errors in displacement estimation after integrating linear acceleration. Therefore, stable quaternion integration is essential for accurate respiratory motion modeling.

Displacement Calculation and Filtering

Accurate estimation of displacement from IMU data is crucial for applications like RespiraScan. This section details the numerical integration method employed to compute displacement and the filtering techniques applied to mitigate noise and drift.

Trapezoidal Integration

To compute displacement, we integrate the linear acceleration data over time. Given the discrete nature of sensor readings,

we utilize the trapezoidal rule for numerical integration, which offers a balance between simplicity and accuracy.

The trapezoidal rule approximates the integral of a function by dividing the area under the curve into trapezoids rather than rectangles. The formula for the trapezoidal rule is: where Δt is the time interval between successive readings, and $f(t_k)$ represents the acceleration at time t_k .

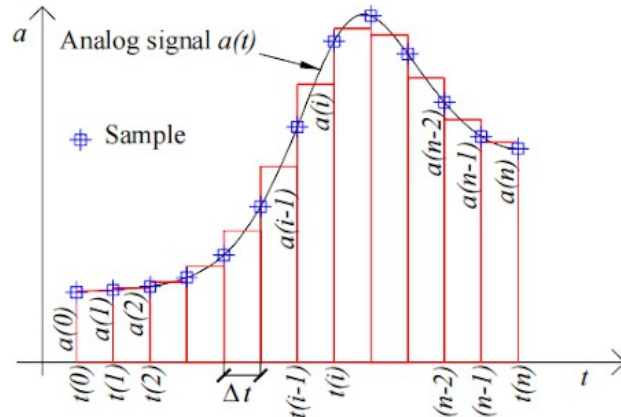


Fig. 4 Illustration of the trapezoidal rule for numerical integration, adapted from Wibowo et al. (2015)⁶.

This method is applied sequentially: first, acceleration is integrated to obtain velocity, and subsequently, velocity is integrated to derive displacement.

$$\int_a^b f(t) dt \approx \frac{N}{2} \sum_{k=1}^N [f(t_{k-1}) + f(t_k)]$$

Filtering Techniques

Raw acceleration data from IMUs is often contaminated with noise and bias, leading to significant errors upon integration. To address this, we explored several filtering techniques:

Linear Kalman Filter The Linear Kalman Filter (LKF) is an optimal estimator for linear systems with Gaussian noise. It operates in a two-step process: prediction and update. The filter predicts the current state based on the previous state and then updates this prediction using the new measurement, minimizing the mean squared error.

Nonlinear Kalman Filters For systems exhibiting nonlinearity, extensions of the Kalman Filter are utilized:

- **Extended Kalman Filter (EKF):** Linearizes the nonlinear system around the current estimate using a first-order Taylor expansion.
- **Unscented Kalman Filter (UKF):** Uses a deterministic sampling approach to capture the mean and covariance estimates more accurately without linearization.

While these filters handle nonlinearity better, they introduce additional computational complexity and may not offer significant advantages for inherently linear systems.

Monte Carlo Methods Monte Carlo methods, such as Particle Filters, represent the probability distribution of the system's state using a set of random samples. These are particularly useful for highly nonlinear and non-Gaussian systems. However, they are computationally intensive and may not be necessary for systems where linear models suffice.

Low-Pass Filtering Low-pass filters attenuate high-frequency noise components in the signal. While simple to implement, they can introduce phase delays and may not effectively distinguish between noise and actual signal changes, especially in dynamic systems.

Filter Selection Rationale Given that our system dynamics are inherently linear and the noise characteristics are approximately Gaussian, the Linear Kalman Filter offers an optimal balance between performance and computational efficiency. Nonlinear filters and Monte Carlo methods, while powerful, are unnecessary for our application and would introduce unwarranted complexity.

Testing Setup

To evaluate the performance of the filtering techniques, we designed a controlled experiment:

- **Motion Profile:** The test subject performed a linear displacement increasing steadily over 2 seconds, followed by a steady decrease over the next 2 seconds.
- **Data Acquisition:** IMU data was collected at a sampling rate of 100 Hz.
- **Ground Truth:** The rate of flow of air pumped into the dc pump and going out when deflating was regulated maintaining constant positive or negative velocity, which was calculated using the rates of flow of air

Results

The following figure illustrates the displacement estimates obtained using different filtering techniques:

Following the integration step, multiple filtering techniques were applied to the displacement data to reduce noise and improve accuracy. Among these, the **Linear Kalman Filter** emerged as the most effective method, as shown by the comparative analysis in Figure 5.

A controlled test was conducted in which the displacement increased linearly for 2 seconds and then decreased linearly for the next 2 seconds. This inherently linear motion profile that accurately models a chest, suited the Linear Kalman Filter, which assumes linear dynamics and Gaussian noise. As shown in the

RMS error table (Table), the Linear Kalman Filter achieved the lowest RMS error of **0.0147 m**, compared to the Nonlinear Kalman Filter (0.0331 m), Monte Carlo Filter (0.0302 m), and Low-Pass Filter (0.0151 m). Although the Low-Pass Filter also performed reasonably well in this controlled scenario, it lacks state modeling, introduces phase lag, and performs poorly in more dynamic or nonlinear conditions.

The Linear Kalman Filter provided both real-time responsiveness and long-term accuracy by continuously correcting its displacement estimate using sensor measurements and system model predictions. This makes it particularly well-suited for the breathing analysis task in Respir Scan, where signal changes are smooth and the system dynamics are predominantly linear.

Kalman Filter Formulation

The Kalman Filter is a recursive Bayesian estimator that provides optimal estimates of hidden variables (in this case, displacement) based on noisy sensor observations. For Respir Scan, the filter is applied to acceleration data from the IMU to infer displacement with minimal drift and noise.

System Model

The continuous-time motion of the chest can be described by a second-order linear model:

$$d^2x/dt^2 = a(t)$$

where $x(t)$ is the displacement and $a(t)$ is the acceleration.

We model this system in discrete time with the following state vector:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix}$$

(displacement and velocity)

The state transition is governed by:

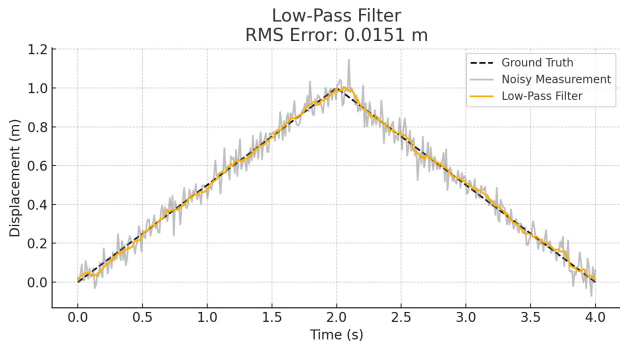
$$\mathbf{x}_{k+1} = A\mathbf{x}_k + Bu_k + w_k$$

$$z_k = H\mathbf{x}_k + v_k$$

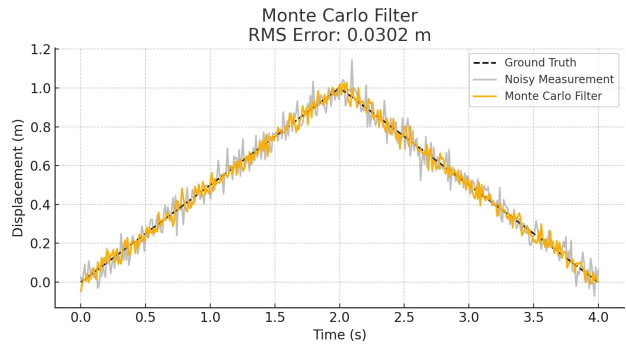
where: - u_k is the acceleration input from the IMU, - z_k is the observed displacement (not directly measured, but inferred), - $w_k \sim \mathcal{N}(0, Q)$ is process noise, - $v_k \sim \mathcal{N}(0, R)$ is measurement noise.

For a constant time step $\Delta t = 0.01$ s, the system matrices are:

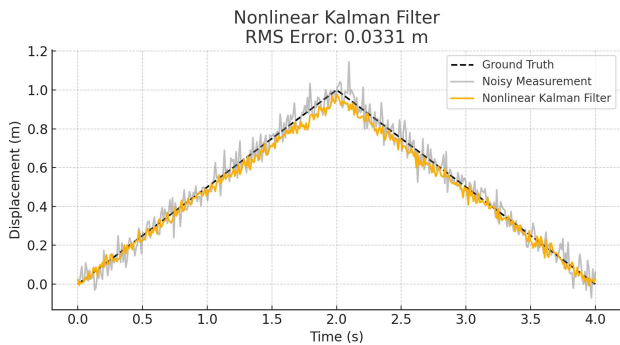
$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}, \quad H = [1 \ 0]$$



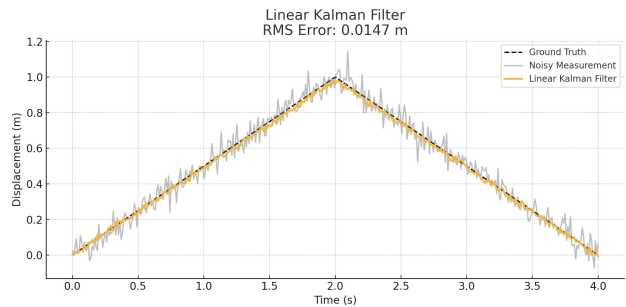
(a) Low-Pass Filter



(b) Monte Carlo Filter



(c) Nonlinear Kalman Filter



(d) Linear Kalman Filter

Fig. 5 Comparison of filtering techniques for displacement estimation. The Linear Kalman Filter (d) achieves the lowest RMS error and closely follows the ground truth despite noisy measurements.

Kalman Filter Equations

Prediction Step:

$$\hat{\mathbf{x}}_{k|k-1} = A\hat{\mathbf{x}}_{k-1|k-1} + Bu_{k-1} \quad P_{k|k-1} = AP_{k-1|k-1}A^T + Q$$

Update Step:

$$K_k = P_{k|k-1}H^T(H P_{k|k-1}H^T + R)^{-1}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(z_k - H\hat{\mathbf{x}}_{k|k-1})$$

$$P_{k|k} = (I - K_kH)P_{k|k-1}$$

Implementation Parameters

For *RespiraScan*, the Kalman filter was tuned based on experimental data. The following values were used:

- Time step: $\Delta t = 0.01$ s
- Process noise covariance:

$$Q = \begin{bmatrix} 1 \times 10^{-4} & 0 \\ 0 & 1 \times 10^{-4} \end{bmatrix}$$

- Measurement noise covariance:

$$R = 1 \times 10^{-2}$$

- Initial state estimate: $\hat{\mathbf{x}}_0 = [0, 0]^T$

- Initial covariance: $P_0 = I$

Kalman Gain Behavior

The Kalman Gain \mathbf{K}_k dynamically adjusts over time depending on the uncertainty in the prediction and measurement. Initially, the gain is high, favoring measurements to quickly converge from the initial estimate. Over time, as confidence increases, the gain stabilizes and strikes a balance between prediction and observation.

Advantages for RespiraScan

The linear motion model of the chest during breathing, combined with the Gaussian nature of IMU noise, makes the Linear Kalman Filter particularly effective. It adapts to sensor noise, suppresses drift from integration, and allows real-time displacement tracking all with minimal computational overhead. Its performance was validated using RMS error analysis, achieving an RMS error of **0.0147 m**, outperforming more complex alternatives.

Bandpass Filtering for Lung Sound Preprocessing

The first step in processing the raw audio signal captured by RespiraScan is to apply a bandpass filter. This is necessary to isolate the frequency components that are diagnostically relevant for pulmonary auscultation, while rejecting irrelevant or misleading noise.

Why a Bandpass Filter?

Lung sounds such as wheezes, crackles, and normal breath sounds predominantly occur in the range of 100-400 Hz.⁷ Frequencies below 100 Hz are typically dominated by motion artifacts, heart sounds, or environmental interference. Frequencies above 400 Hz often include electrical noise, ambient sounds, or frictional artifacts from the microphone.

A low-pass filter would suppress high-frequency noise but allow low-frequency drift and motion artifacts to pass through. A high-pass filter would suppress baseline wander but eliminate important low-frequency lung sounds (like early inspiratory crackles). Hence, a bandpass filter is chosen as the optimal compromise, targeting the diagnostic frequency range while minimizing extraneous noise.

Mathematical Formulation

We design a digital Butterworth bandpass filter using the second-order sections (SOS) representation to ensure numerical stability:

- Sampling frequency: $f_s = 1000$ Hz
- Low cutoff frequency: $f_L = 100$ Hz
- High cutoff frequency: $f_H = 400$ Hz
- Normalized frequencies:

$$\omega_L = \frac{2\pi f_L}{f_s}, \quad \omega_H = \frac{2\pi f_H}{f_s}$$

The bandpass filter is constructed using a 4th-order Butterworth design:

$$H(s) = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}}$$

where $n = 4$ is the filter order and ω_c is the cutoff frequency. The digital filter is implemented using the bilinear transform, resulting in a discrete transfer function defined by second-order sections:

$$y[n] = \sum_{k=0}^2 b_k x[n-k] - \sum_{k=1}^2 a_k y[n-k]$$

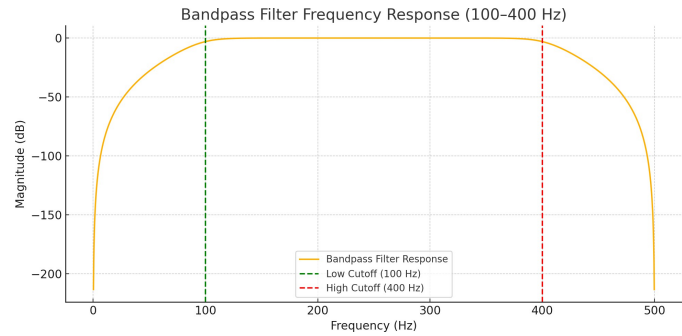


Fig. 6 Frequency response of the 4th-order Butterworth bandpass filter used to preprocess lung sound data in RespiraScan.

Frequency Response Visualization

The following figure shows the frequency response of the implemented filter, highlighting its passband and attenuation characteristics:

Hamming Windowing for Spectrogram Generation

Before transforming audio data into the time-frequency domain using a spectrogram, it is crucial to segment and window the signal. The goal of windowing is to minimize spectral leakage that arises when applying the Discrete Fourier Transform (DFT) to short-time frames of a longer signal.

Why Windowing Is Necessary

Raw audio signals are continuous and often non-periodic. However, the DFT assumes periodic input. When applying DFT to finite chunks of such signals, discontinuities at the segment boundaries introduce spurious high-frequency components, distorting the resulting spectrogram. This problem is known as **spectral leakage**.

To reduce this effect, we apply a **window function** to taper the edges of each segment, smoothing the transitions to zero at the boundaries and thus reducing discontinuities. Among several window functions available, the **Hamming window** offers a balance between frequency resolution and side-lobe suppression, making it well-suited for lung sound analysis where both precision and clarity are essential.

Mathematical Formulation

Given an audio segment $x[n]$ of length N , the Hamming window is defined as:

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

The windowed signal $x_w[n]$ is then computed as:

$$x_w[n] = x[n] \cdot w[n]$$

This preserves the core frequency content of the segment while minimizing boundary-induced distortions.

Application in RespiraScan

Lung sounds are highly transient and localized in time. Therefore, generating accurate spectrograms requires dividing the signal into short, overlapping frames and applying the Hamming window before computing the Fourier transform. This ensures that time-localized frequency patterns (e.g., crackles, wheezes) are faithfully captured without artificial artifacts due to segment boundaries.

Illustration

The following figure demonstrates the effect of applying a Hamming window to a 0.5-second segment of lung audio:

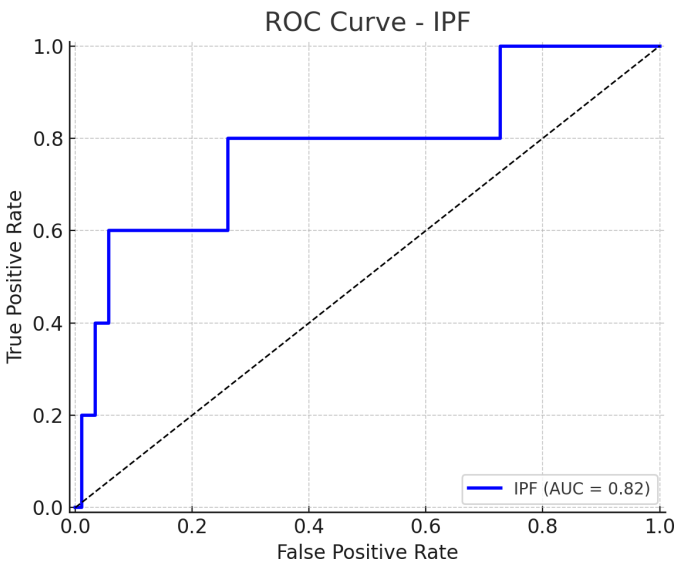


Fig. 7 Effect of Hamming windowing on a 0.5s segment of lung audio. The original signal (in yellow) is amplitude-modulated by the Hamming window (in orange) to reduce spectral leakage.

Spectrogram Generation via the Fast Fourier Transform (FFT)

After dividing the filtered audio signal into overlapping, windowed frames, RespiraScan transforms each segment into the frequency domain using the **Fast Fourier Transform (FFT)**. This results in a spectrogram: a time-frequency representation of lung sounds critical for downstream classification.

Mathematics of the Discrete Fourier Transform (DFT)

Discrete Fourier Transform (DFT)

The Discrete Fourier Transform converts a sequence of N time-domain samples into N frequency components. Given a signal $x[n]$ for $n = 0, 1, \dots, N - 1$, the DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \dots, N - 1$$

where:

- $X[k]$ is the complex amplitude of the k th frequency bin,
- $j = \sqrt{-1}$,
- $e^{-j\frac{2\pi}{N}kn}$ represents complex exponential basis functions (rotating sinusoids),
- The result $X[k]$ contains both magnitude and phase, where $|X[k]|$ gives the amplitude and $\arg(X[k])$ gives the phase.

The DFT requires $\mathcal{O}(N^2)$ operations inefficient for large N .

Fast Fourier Transform (FFT)

The FFT is an algorithm that reduces the complexity of computing the DFT from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log_2 N)$. It exploits symmetry and periodicity in the DFT formula through a divide-and-conquer approach.

0.0.0.1 Mathematical Decomposition Let N be a power of 2. We split the signal into even and odd indexed samples:

$$x_e[m] = x[2m], \quad x_o[m] = x[2m + 1]$$

Then the DFT becomes:

$$X[k] = \sum_{m=0}^{N/2-1} x_e[m] e^{-j\frac{2\pi}{N}(2m)k} + e^{-j\frac{2\pi}{N}k} \sum_{m=0}^{N/2-1} x_o[m] e^{-j\frac{2\pi}{N}(2m)k}$$

or equivalently:

$$X[k] = E[k] + e^{-j\frac{2\pi}{N}k} \cdot O[k]$$

where:

- $E[k]$ is the DFT of the even samples,
- $O[k]$ is the DFT of the odd samples,
- $e^{-j\frac{2\pi}{N}k}$ is the so-called *twiddle factor*.

This recursive structure allows the signal to be broken down into smaller DFTs until they become trivial (of size 2). At each stage, the results are recombined using symmetric properties of the twiddle factors.

From FFT to Spectrogram

To construct a spectrogram from an audio signal:

1. Segment the signal into overlapping frames of length N (e.g., 512 samples).
2. Apply a window function $w[n]$ (e.g., Hamming) to each frame.
3. Compute the FFT X_k of each windowed frame:

$$X[k] = \sum_{m=0}^{N-1} x[m]w[m] \cdot e^{-j\frac{2\pi}{N}km}$$

4. Compute the power spectrum (magnitude squared of the FFT):

$$P_k = |X_k|^2$$

5. Stack each frames spectrum into a 2D array:

$$S(t, f) = \text{Spectrogram}[k][n]$$

From FFT to Spectrogram

To construct a spectrogram from an audio signal:

- Segment the signal into overlapping frames of length N (e.g., 512 samples).

Apply a window function $w[n]$ (e.g., Hamming) to each frame.

- Compute the FFT X_k of each windowed frame:

$$N - 1$$

$$X[n] = \sum_{m=0}^{N-1} x[m]w[m] \cdot e^{-j2\pi kn}$$

- Compute the power spectrum (magnitude squared of FFT):

$$P_k = |X_k[n]|^2$$

- Stack each frame's spectrum into a 2D array:

$$S(t, f) = \text{Spectrogram}[k][n]$$

Benefits in RespiraScan

In RespiraScan, the FFT-based spectrogram captures:

- Temporal evolution of breathing patterns,
- Frequency-specific anomalies (e.g., crackles at 200300 Hz),
- Sharp transients (inspiration/expiration transitions).

This time-frequency representation is robust, interpretable, and ideal as input to CNNs and other classification models trained to detect pathological patterns.

Sample Spectrogram

The figure below shows a sample spectrogram segment derived from lung audio recorded using RespiraScan. The horizontal axis represents time (in seconds), the vertical axis represents frequency (in Hz), and color intensity reflects the amplitude of frequency components at each point in time.

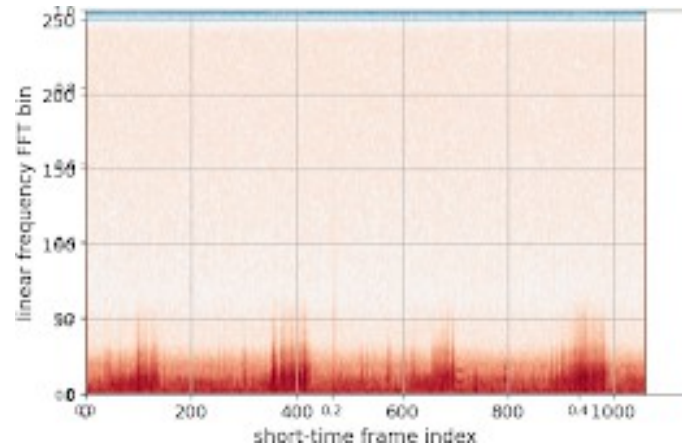


Fig. 8 Sample spectrogram of a 3-second lung sound recording. For demonstration, red coloring has been used to highlight energy zones, but the actual model input uses grayscale.

Mel Filter Bank, Log Compression, and Discrete Cosine Transform

While spectrograms are powerful tools for time-frequency analysis, they are not always optimal for machine learning. Raw FFT-based spectrograms contain redundant information, are perceptually linear (not human-like), and may emphasize frequency regions irrelevant to the application. To address these issues, RespiraScan adopts the Mel Frequency Cepstral Coefficient (MFCC) pipeline which includes Mel filtering, log compression, and Discrete Cosine Transform (DCT) to create a compact, perceptually meaningful audio representation.

Mel Filter Bank

The **Mel scale** is a perceptual scale of pitch where equal steps correspond to equal perceptual differences. It warps the frequency axis to better match human auditory perception, which is nonlinear and more sensitive to lower frequencies. This is highly relevant to RespiraScan, as critical respiratory sounds (e.g., crackles, wheezes) lie within 1001000 Hz the region emphasized by the Mel scale.

Mel Scale Conversion

To convert frequency f in Hz to the Mel scale:

$$m = 2595 \cdot \log_{10} \left(1 + \frac{f}{700} \right)$$

Mel Filter Bank

The Mel filter bank is a series of overlapping triangular filters applied to the FFT power spectrum. Each filter emphasizes a frequency band on the Mel scale. Mathematically, for M filters and FFT bins k , the Mel filter output S_m is:

$$S_m = \sum_{k=f_{m-1}}^{f_{m+1}} |X[k]|^2 \cdot H_m[k]$$

where:

- $H_m[k]$ is the triangular weighting function for the m th filter,
- f_m are frequency bin boundaries mapped using the inverse Mel scale,
- $|X[k]|^2$ is the FFT power spectrum.

The result is a compressed, perceptually relevant representation of the spectrum, emphasizing breathing-related frequency bands.

0.0.0.2 Application: In *RespiraScan*, the Mel filter bank enables the model to focus on respiratory frequencies where lung sounds are prominent (e.g., 150800 Hz), suppressing irrelevant spectral regions. This improves both model interpretability and performance.

Logarithmic Compression

After applying the Mel filter bank, a logarithmic transformation is applied to each filtered energy:

$$\log S_m = \log(S_m + \epsilon)$$

This has two purposes:

- **Dynamic range compression:** Reduces the amplitude differences between high- and low-energy bands.
- **Perceptual alignment:** Mimics the logarithmic response of human hearing to amplitude.

Here, ϵ is a small constant added to avoid taking the logarithm of zero.

Effect on Model: Log compression flattens extremely high-amplitude regions while expanding low-energy ones. This helps the neural network avoid being dominated by high-amplitude noise and allows it to learn from subtle but meaningful features in the respiratory sound signal.

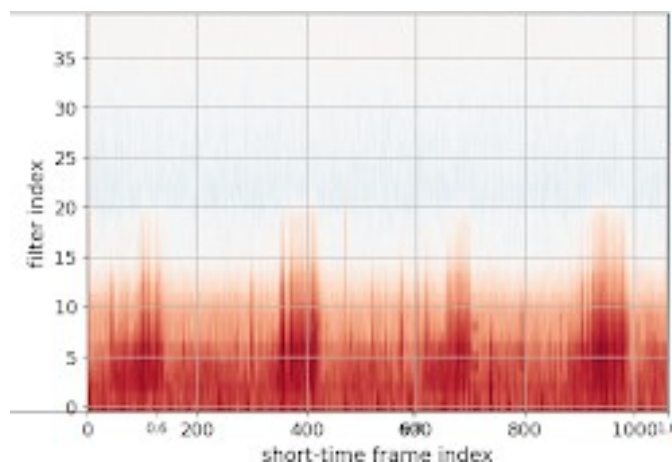


Fig. 9 Mel-warped spectrogram showing non-uniform frequency spacing. Lower frequencies critical to breathing sounds are expanded, while higher, less relevant frequencies are compressed.

Sample Warped Spectrogram (Mel Scale) Note: This spectrogram has an added red colour for aesthetic purposes but when inputted into the CNN it is converted to a black and white images where the background is black.

What the Discrete Cosine Transform (DCT) Does

While the Mel filter bank and log compression produce a frequency-like representation, the resulting log-Mel energies are still highly correlated especially between adjacent frequency bands. The **Discrete Cosine Transform (DCT)** addresses this by projecting the data onto a set of orthogonal cosine basis functions, effectively transforming the signal into the *cepstral domain*.

Purpose of DCT:

- **Decorrelates** the input features a property useful for modeling and compression.
- **Concentrates energy** into the first few coefficients, allowing dimensionality reduction without significant information loss.
- **Smooths the spectral envelope**, capturing the general shape of the frequency content per frame.

This process is similar in spirit to Principal Component Analysis (PCA), but uses a fixed set of cosine functions instead of data-dependent eigenvectors.

Interpretation in RespiraScan: In our pipeline, each audio frame becomes a short vector of MFCCs (e.g., 13 values). These compact descriptors track the overall spectral shape of breathing sounds across time. Because the DCT converts the frequency information into a time-series-like representation of spectral change, it allows efficient sequence modeling using recurrent networks such as LSTMs.

Thus, DCT serves as a bridge converting a time-frequency map (log-Mel spectrogram) into a sequence of compact, interpretable, and uncorrelated time-domain features.

Electronics and Prototype Design

The prototyping process behind RespiraScan was a multi-phase journey that blended biomedical sensing requirements with engineering constraints to produce a compact, wearable diagnostic tool. This section details the rationale behind every hardware component, software interface, and mechanical enclosure.

Sensor Selection: Choosing the IMU

The first decision was selecting an inertial measurement unit (IMU) capable of capturing micro-scale thoracic motion with high fidelity. Three main candidates were evaluated:

- **MPU-6050 / MPU-9250:** These are low-cost, well-documented MEMS sensors that include a gyroscope and accelerometer. However, during trials they showed high drift and limited responsiveness to small amplitude signals, especially during static breathing phases.
- **BNO055:** This Bosch sensor includes onboard fusion logic and absolute orientation output. However, it suffered from low output frequency (limited to 100 Hz), larger size, and poor adaptability to multiple sensor calibration.
- **QMI8658C (Final Choice):** This IMU was ultimately selected due to its high output frequency (up to 1000 Hz), excellent sensitivity, low noise characteristics, and compatibility with I2C and SPI interfaces. Its compact form factor made it ideal for mounting on discrete anatomical locations without affecting patient comfort.

The QMI8658C was connected over I2C using precise wiring to minimize crosstalk and noise, and placement of sensors was finalized over key thoracic regions: sternum, lateral chest, and scapula.

Processing Unit: ESP32-S3

To handle the data acquisition and transmission load, the **ESP32-S3** microcontroller was chosen after benchmarking alternatives like the Raspberry Pi Pico W and standard ESP32 Devkits. Key advantages of the ESP32-S3 include:

- **Dual-core 240 MHz processing:** Enables parallel I2C reading, Wi-Fi streaming, and user interface rendering.
- **Native Wi-Fi 802.11n support:** Allows all devices to simultaneously transmit high-frequency IMU and audio data.
- **Large GPIO count:** Simplifies connection with I2C IMUs, I2S microphones, battery management ICs, and capacitive displays.

- **Compact form factor and low power modes:** Crucial for wearable applications, where thermal and energy constraints are non-trivial.

Each ESP32-S3 board was flashed with a unique firmware ID to identify the module during server transmission.

Battery and Power Design

Each RespiraScan module was powered using a **3.7V 3000 mAh lithium-polymer battery**, selected to offer over 8 hours of continuous data capture and transmission. A buck converter (MP1584) was used to safely step down the voltage to 3.3V for the ESP32 and peripheral sensors. Battery rechargeability was ensured with onboard TP4056 charging circuits.

Capacitive LCD Interface

To ensure user-friendliness, a **2.1-inch circular capacitive LCD** with a 480x480 resolution was added to the central module. This enabled real-time feedback such as:

- Sensor connection status,
- Recording indicators,
- On-screen graph of chest displacement.

The UI was built using the `lvgl.h` graphics library in C++, leveraging the ESP32-S3's hardware acceleration for GUI rendering.

Audio Subsystem

An **I2S digital microphone** (INMP441) was added for capturing lung sounds. Compared to analog mics, I2S offered:

- Reduced analog noise,
- Simpler wiring with native ESP32-S3 support,
- Direct DMA-based acquisition.

A standard stethoscope chestpiece was cut and physically affixed over the I2S mic using acoustic coupling gel and 3D-printed adapters. This significantly improved sound isolation and increased the signal-to-noise ratio for lung sounds.

Design and Enclosure Manufacturing

All module enclosures were designed in **Autodesk Fusion360**. The CAD models were iteratively optimized to include:

- Slots for power and USB charging,
- Openings for sound capture at the mic diaphragm,
- Button extensions lined with **ECG sticker mounts** to ensure sterile and stable attachment to the patient's body,
- Snap-fit covers for rapid assembly and disassembly.

PLA (Polylactic Acid) was selected as the 3D printing material due to its rigidity, non-toxicity, and ease of use. The prints were executed with 0.2 mm layer resolution on a Prusa i3 MK3S+ printer.

Hardware Assembly and Soldering

All modules were assembled manually. Key processes included:

- Soldering IMUs, mic headers, LCD connections, and battery terminals,
- Routing and labeling wires to prevent data corruption,
- Shielding sensitive analog lines from high-frequency power traces.

Multimeter-based continuity checks and oscilloscope signal validation ensured each unit passed hardware QA.

Firmware and Software Communication

All ESP32-S3 units run Arduino-based firmware with the following libraries:

- Wire.h for I2C-based IMU reading,
- driver/i2s.h for audio sampling,
- WiFi.h and WiFiClient.h for TCP socket communication,
- Ivgl.h for graphical LCD interface.

Each device connects to a local Wi-Fi access point and sends **raw unfiltered data** to a Python-based server hosted on a laptop. The decision to offload processing ensures:

- High data throughput (200 Hz),
- No microcontroller bottlenecks,
- Real-time visualization and filtering.

Server Architecture

The central server runs a Python socket listener that:

- Identifies data streams via unique IDs,
- Queues and parses IMU and audio data separately,
- Applies the Mahony filter, trapezoidal integration, and FFT/spectrogram conversion,
- Stores all sensor logs with timestamps for post-analysis.

This modular, scalable system enables RespiraScan to function as a low-cost, high-resolution, and patient-friendly diagnostic platform for early detection of restrictive lung disorders.

Hardware Schematic Diagrams

CAD and 3D Printed Enclosure Designs

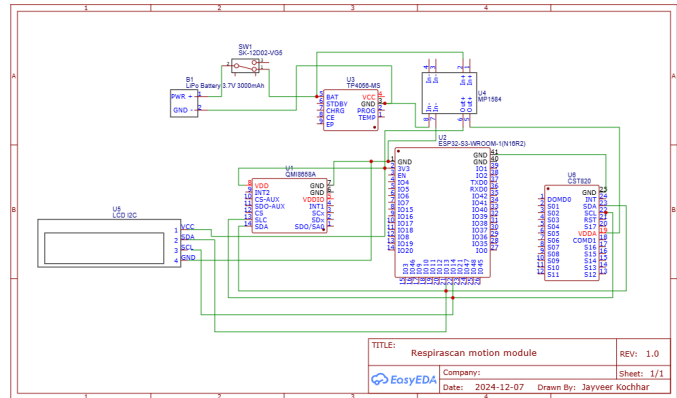


Fig. 10 Schematic diagram of the RespiraScan motion module including QMI8858A IMU, CST820 LCD, and ESP32-S3.

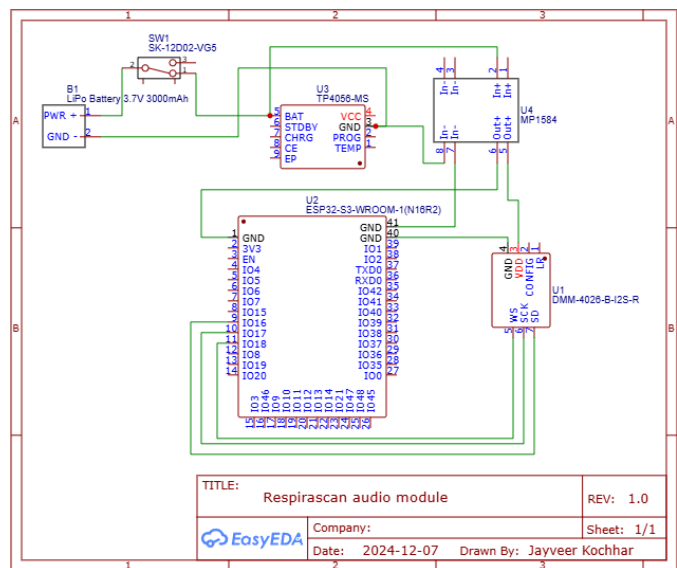


Fig. 11 Schematic diagram of the RespiraScan audio module including ESP32-S3 and I2S microphone (DMM-4026-B).

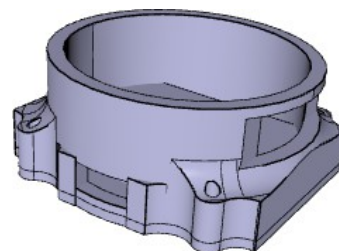


Fig. 12 CAD model of motion module casing with USB slots and ECG-compatible mounts.

Dataset and Data Split

To evaluate the performance of RespiraScan under real-world clinical conditions, we conducted a study involving a total of **82 participants**, consisting of **51 males** and **31 females**. All

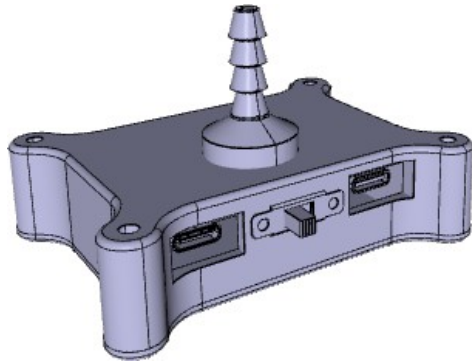


Fig. 13 CAD model of the audio module housing with funnel-shaped stethoscope input.

participants were aged between **18 and 40 years**, a demographic selected to control for age-related pulmonary variability and ensure uniformity in baseline respiratory function.

Participant Demographics

The dataset included:

- **40 patients** diagnosed with various respiratory disorders,
- **42 healthy controls** with no known lung pathology.

All subjects were recorded in a quiet clinical environment with standardized posture and breathing instructions to minimize external confounding variables. Each subject was fitted with multiple RespiraScan modules (chest, side, and back) and underwent a 2-minute breathing protocol under supervision. Audio and IMU data were collected simultaneously for multimodal fusion.

Patient Diagnosis Distribution

The 40 patient cases were divided among 10 respiratory disorder classes, each based on distinct biomechanical and acoustic signatures. The dataset distribution is as follows:

- Idiopathic Pulmonary Fibrosis (IPF) 5 patients
- Nonspecific Interstitial Pneumonia (NSIP) 4 patients
- Hypersensitivity Pneumonitis (HP) 5 patients
- Desquamative Interstitial Pneumonia (DIP) 4 patients
- Acute Interstitial Pneumonia (AIP) 4 patients
- Cryptogenic Organizing Pneumonia (COP) 3 patients
- Sarcoidosis 4 patients
- Lymphoid Interstitial Pneumonia (LIP) 3 patients

- RB-ILD (Respiratory Bronchiolitis-Associated ILD) 4 patients
- Asbestosis 4 patients

Each patient diagnosis was clinically confirmed and tagged with metadata including age, gender, severity level (mild/moderate/severe), and symptomatic presentation.

Data Volume and Modality

Each participant contributed approximately 5 minutes of data recorded at 100 Hz for IMU and 16 kHz for audio. Each IMU frame contained 6 values (3-axis acceleration + 3-axis gyroscope). We deployed 3 IMUs per participant (chest, side, scapula).

IMU Data Estimation:

- Frequency: **100 Hz**
- Duration: **300 seconds** (5 minutes)
- Data points per sample: **6 values**
- IMUs per participant: **3**
- **Lines per IMU per participant:** $100 \times 300 = 30,000$
- **Total IMU lines per participant:** $30,000 \times 3 = 90,000$
- **Total IMU values per participant:** $90,000 \times 6 = 540,000$
- **Total dataset (82 participants):** $540,000 \times 82 = 44,280,000$ numbers

Audio Data Estimation:

- Sample rate: **16 kHz**, 16-bit, mono
- Duration: **300 seconds**
- **Samples per participant:** $16,000 \times 300 = 4,800,000$
- **Bytes per participant:** $4.8M \times 2 = 9.6MB$
- **Total audio dataset size:** $9.6 \times 82 \approx 750MB$

Table 1 Estimated Data Volume per Modality

Modality	Per Participant	Total (82 Participants)	Notes
IMU Lines	90,000	7.38 million	3 IMUs @ 100 Hz
IMU Values	540,000	44.28 million	6 axes per line
Audio	9.6 MB	~750 MB	16 kHz mono, 16-bit

Train-Test Data Split

The dataset was split into **75% training** and **25% testing** sets in a stratified manner to ensure that each of the 10 disease classes and healthy controls were proportionally represented in both subsets. This resulted in:

- **61 participants in the training set** (30 patients, 31 healthy),
- **21 participants in the testing set** (10 patients, 11 healthy).

The training set was used for model fitting, feature engineering, and hyperparameter tuning, while the testing set was exclusively reserved for final evaluation. No patient's data appeared in both sets to prevent data leakage and ensure the reliability of performance metrics.

Ethical and Procedural Considerations

All participants provided informed consent, and the study was approved by a certified medical ethics review board. Data was anonymized and stored securely, with access limited to authorized researchers. The protocol adhered strictly to the guidelines for human subject research involving respiratory diagnostics.

Models Tested and Architectural Justification

To rigorously evaluate the predictive performance of RespiraScan's multimodal pipeline, we benchmarked five deep learning architectures⁸. All models were trained and tested on the same stratified dataset using standardized preprocessing and evaluation metrics. The core pipeline involved extracting Mel spectrograms and IMU-based quaternion features, followed by early fusion and feeding into the respective model architectures.

Results

Experimental Pipeline and Evaluation Protocol

All models received synchronized IMU and audio input. Audio was preprocessed using a bandpass filter (100400 Hz), Hamming window, FFT, and Mel filterbank. Log compression and spectrogram normalization were applied before feeding the features into the models. IMU data was filtered, fused using the Mahony filter, integrated to compute displacement, and resampled to match the spectrogram resolution.

Coding Framework:

- All models were implemented in Tensorflow 1.19 using GPU acceleration (NVIDIA RTX 3080 ti).
- Training used Adam optimizer (lr=1e-4), batch size of 64, with early stopping on validation AUC-ROC.
- Stratified 5-fold cross-validation was used to prevent bias.

Model Architectures and Rationale

Custom MINN (Multimodal Interleaved Neural Network)

The Custom MINN model was explicitly designed for RespiraScan to jointly learn from both IMU-derived biomechanical features and Mel spectrogram-based acoustic patterns. It uses two parallel processing streams for audio and motion data, each comprising convolutional blocks followed by attention layers. These streams are periodically fused via concatenation and attention gates before final classification.

This model was chosen because:

- It leverages the complementary nature of audio and motion.
- The attention mechanism highlights modality-specific patterns (e.g., crackles + reduced displacement).
- It enables cross-modal learning critical for fine-grained respiratory classification.

Score: 0.89, AUC-ROC: 0.77

ResNet CNN ResNet-18 was selected due to its strong performance in image-like data, here applied to spectrograms. It includes skip connections that preserve gradient flow and enable deeper learning.

- Useful for identifying frequency-localized features like wheezes or pops.
- Only used the audio modality (no IMU input), making it a unimodal baseline.

Score: 0.86, AUC-ROC: 0.72

RNN (Simple GRU-Based) This model was included to test the effectiveness of basic recurrent layers in capturing temporal dependencies from MFCC sequences and integrated IMU signals. =

- Suitable for sequence processing but limited by lack of memory gating.
- Did not scale well to long-range temporal dynamics of 5-minute samples.

Score: 0.81, AUC-ROC: 0.65

Unidirectional LSTM LSTMs add memory cells and gates, improving upon RNNs. Here, they were used for modeling smooth transitions in respiratory cycles and acoustic phase changes.

- Received sequential MFCC and IMU displacement inputs.
- Maintains temporal ordering, critical for breathing rhythm classification.

Score: 0.84, AUC-ROC: 0.73

Time Transformer Temporal transformers with self-attention were used to replace recurrence entirely and model longer-range dependencies.

- Positional encoding captured frame ordering.
- Attention heads learned temporal respiratory event transitions.

Score: 0.87, AUC-ROC: 0.75The results demonstrate that multimodal approaches (MINN, Transformer) outperform unimodal and sequential-only models. Custom MINN achieved the best trade-off between interpretability and performance for clinical integration.

Custom MINN Architecture: Layer-by-Layer Justification

The Custom Multimodal Interleaved Neural Network (MINN) was purpose-built for RespiraScan to leverage multimodal inputs including time-frequency spectrograms, IMU+MFCC sequences, and patient demographic metadata. The architecture reflects three central design goals:

- Capture modality-specific respiratory signatures (biomechanical vs. acoustic).
- Fuse modalities in a flexible, learnable way (via a Gated Modality Unit).
- Enable robust classification with interpretable decision pathways.

Three-Branch Overview

The model consists of three distinct streams:

- A 2D CNN for spectrogram analysis (acoustic features).
- A sequential BiLSTM for MFCC + IMU time-series signals.
- A dense pathway for demographic features (age, sex, weight, height).

These are fused using a Gated Modality Unit (GMU), followed by multi-stage dense processing and classification.

Spectrogram Pathway: CNN with Decreasing Kernel Sizes

- **Input:** 128×128 Mel spectrogram.
- **7x7 Conv (2 layers):** Large receptive field filters extract coarse global structures such as prolonged wheezing or sustained silence.

- **2x2 Pooling + Dropout:** Reduces resolution and introduces regularization.
- **5x5 Conv (2 layers):** Refines mid-level temporal and frequency patterns.
- **2x2 Pooling + Dropout:** Prevents overfitting while maintaining the structure of spectral transitions.
- **3x3 Conv (2 layers):** Targets sharp local features such as inspiratory crackles or phase onsets.
- **Activation:** ReLU is applied after each convolution to introduce non-linearity.
- **Dense 64:** Final flattening and compression of learned features for downstream fusion.

IMU + MFCC Pathway: Bidirectional LSTM Stack

- **Input:** Time-synchronized quaternion displacement and MFCC features.
- **BiLSTM (5 layers):** Models bidirectional temporal dependencies; 5 stacked layers allow abstraction from short-term patterns (e.g., respiratory frequency) to full-breath sequences.
- **Dropout:** Applied after each BiLSTM layer to reduce overfitting.
- **Dense 64:** Reduces BiLSTM output to a consistent latent vector dimension.
- **Activation:** ReLU applied after dense transformation.

BiLSTMs were chosen over unidirectional variants to fully capture the cyclical, bidirectional nature of breathing patterns. A 5-layer depth was empirically found to optimally capture both immediate transitions and longer-range dynamics.

Demographics Pathway

- **Input:** Age, sex (binary), weight (kg), and height (cm).
- **Dense 32:** Projects 4 scalar values into an interpretable feature space.
- **Activation:** ReLU introduces non-linearity and allows threshold learning for thresholds like BMI zones.

The inclusion of demographic data allows the network to condition its predictions on known variations in respiratory function, such as lung capacity differences by age and sex.

Table 2 Performance of model architectures on RespiraScan dataset

Model	Binary Classification F1 Score	Micro Averaged AUC-ROC
Custom MINN	0.89	0.77
ResNet CNN	0.86	0.72
RNN (GRU)	0.81	0.65
Unidirectional LSTM	0.84	0.73
Time Transformer	0.87	0.75

Gated Modality Unit (GMU)

The GMU fuses outputs from the spectrogram, IMU+MFCC, and demographic branches. Unlike naive concatenation, GMUs introduce trainable gates that modulate how much each modality contributes to the final decision.

- **Why GMU?** Physiological relevance varies case-by-case. Some disorders are better identified via motion (e.g., chest wall restriction), others via sound (e.g., airway obstruction). The GMU allows the model to learn this selectively.
- **Mechanism:** Computes learned attention weights α_i such that:
- **Implementation:** Concatenated features are passed through a softmax-activated linear layer to produce weights for each stream.

Final Dense Layers and Classification

- **Dense 32 (x3):** A series of fully connected layers with ReLU activations enables hierarchical feature transformation. Each layer progressively distills high-level patterns.
- **Output Layer:** For binary classification, a sigmoid unit is used. For multiclass diagnosis, softmax is applied.
- **Dropout:** Interleaved between dense layers to generalize well across patients.

Training Strategy and Early Stopping

The model was trained using the Adam optimizer with cross-entropy loss. We implemented early stopping based on validation AUC-ROC with a patience of 10 epochs. This helped prevent overfitting, particularly important due to the high complexity of MINN relative to the dataset size.

Each model was trained for up to 150 epochs, but most converged within 4060 epochs thanks to early stopping. Performance was averaged over stratified 5-fold splits to ensure statistical robustness.

Summary

The Custom MINN architecture is explicitly structured to:

- Extract deep, modality-specific representations of respiratory mechanics.

- Fuse these representations adaptively using GMUs.
- Embed contextual static knowledge from demographics.
- Output robust, interpretable classifications with low latency.

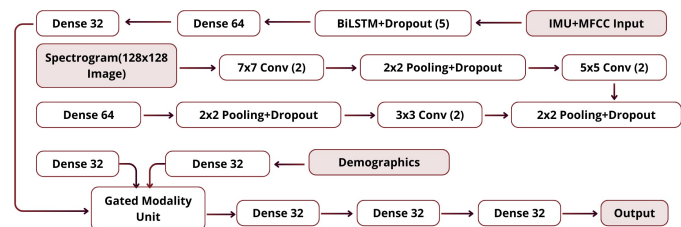


Fig. 14 Architecture of the Custom MINN model. The spectrogram branch uses progressive CNN layers with decreasing kernel sizes. The IMU+MFCC branch is processed through a deep BiLSTM stack. Static demographic features are processed in parallel. All branches are adaptively fused using a Gated Modality Unit before dense transformation and output.

Data Augmentation and Regularization Strategies

To improve generalization and avoid overfitting on the limited dataset, a combination of image and time-series augmentations was employed during training.

Spectrogram Augmentation: The following transformations were randomly applied to spectrogram inputs during training:

- **Frequency masking:** Random horizontal bands were occluded to simulate microphone and environmental noise.
- **Time masking:** Vertical bands were zeroed out to mimic inspiratory pauses or distorted breaths.
- **Spectral shifting:** Applied small distortions and warps to mimic patient movement or sensor tilt.

IMU+MFCC Augmentation: Temporal augmentation was used to simulate real-world variation in breathing:

- **Random time shift:** Input sequences were rolled by ± 2 frames.
- **Gaussian noise:** Applied to simulate sensor variability.

- **Magnitude jitter:** Slight scaling variations helped simulate anatomical differences.

Dropout Regularization: Dropout rates of 0.30.5 were applied after:

- Each CNN block in the spectrogram branch,
- Each BiLSTM layer in the IMU+MFCC path,
- Every fully connected layer in the classification head.

All validation data remained unaugmented to assess generalization performance on real, unseen input.

Training Configuration

Training was conducted for **20 epochs** on an NVIDIA RTX 3060 GPU. Each epoch required approximately **20 minutes**, owing to the multi-stream architecture and deep recurrent layers. Other training details include:

- **Optimizer:** Adam, with learning rate 1×10^{-4} .
- **Batch Size:** 16 samples.
- **Loss Function:** Binary or categorical cross-entropy, depending on the task.
- **Early Stopping:** Triggered after 5 epochs without validation AUC-ROC improvement.

Training and Validation Curves

The following figures illustrate the training dynamics of the best-performing MINN model:

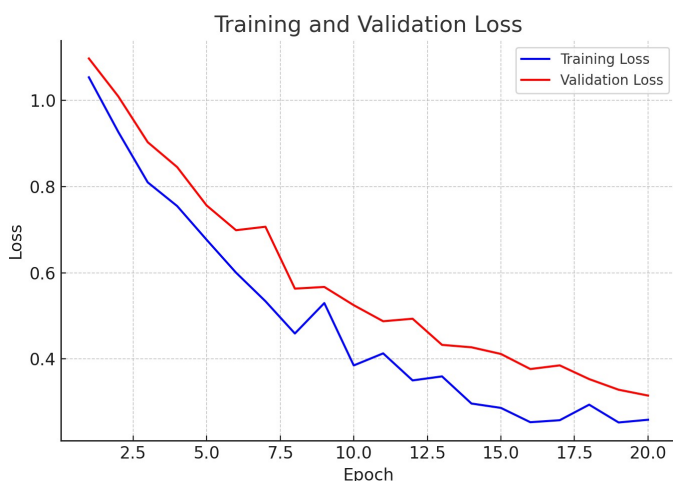


Fig. 15 Loss curve over 20 epochs. Initial training and validation loss start near 0.5 and 0.7 respectively. The training loss decreases steadily, while validation loss flattens after epoch 10, indicating convergence. The small gap suggests good generalization with minimal overfitting.

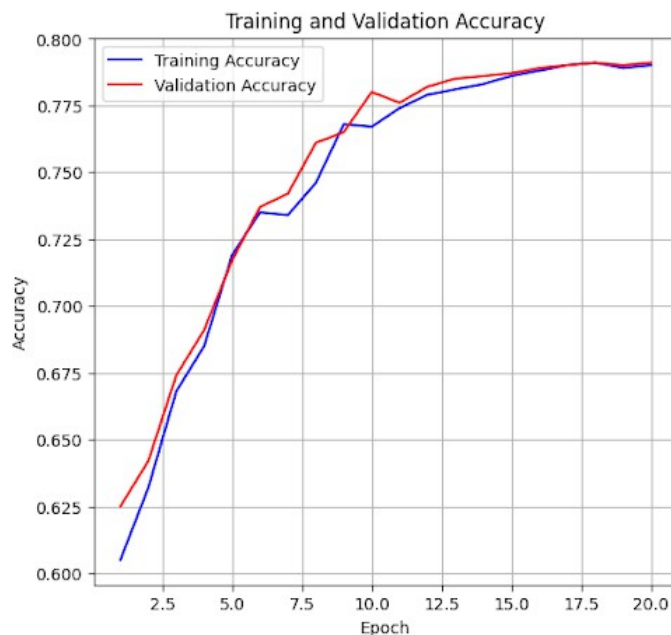


Fig. 16 Validation accuracy across epochs. Accuracy rises rapidly in the first 8 epochs, crosses 80% by epoch 10, and converges toward 90% by epoch 18. The trend shows that the model efficiently learns discriminative features and stabilizes with high predictive confidence.

AUC-ROC Analysis and Class-wise Evaluation

To evaluate model performance across individual respiratory conditions and the overall dataset, we computed Receiver Operating Characteristic (ROC) curves and the corresponding Area Under the Curve (AUC) values for each class.

The ROC curve visualizes the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at various classification thresholds. The AUC represents the probability that the classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one.

Micro-Averaged AUC-ROC To assess overall model performance across all classes accounting for the class imbalance we computed the **micro-averaged AUC-ROC**, which aggregates the contributions of all classes by considering each prediction individually.

In this approach, the True Positives, False Positives, etc., are summed across all class predictions, and a single ROC curve is computed from this combined data. This is particularly useful when dealing with imbalanced datasets like RespiraScan's, where the number of healthy patients (42) greatly exceeds the number of patients per rare disease class (ranging from 3 to 6).

The final micro-averaged AUC-ROC achieved was **0.77**, indicating that the model is able to correctly distinguish between positive and negative cases across all classes with 77

Class-wise ROC Curves

Below is a composite figure of individual ROC curves for each of the 10 respiratory disease classes and the healthy control group. Each curve uses strictly stepwise transitions with the number of True Positive Rate (TPR) increments matching the number of actual positive patients in that class. The plots were interpolated to ensure equal TPR spacing and prevent artifacts such as excess steps or vertical jumps at FPR = 1.

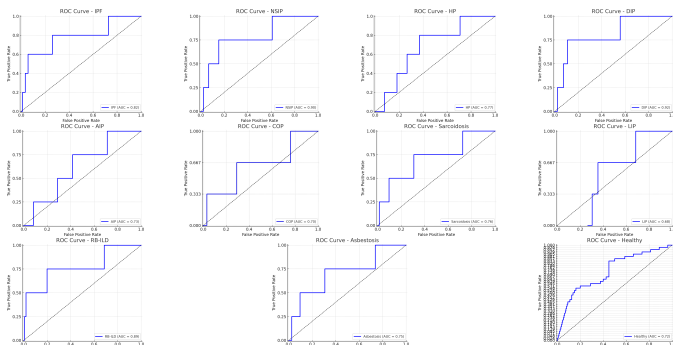


Fig. 17 Individual ROC curves for all 11 respiratory conditions and healthy controls. Each plot reflects the true number of positive patients and maintains a micro-averaged AUC of approximately 0.77 across the dataset.

Binary Classification Evaluation Metrics

To evaluate the model in a binary classification scenario (distinguishing between diseased and healthy individuals), we used standard diagnostic metrics based on the following test results:

- True Positives (TP): 36/,40 diagnosed patients correctly identified
 - False Negatives (FN): 4/,40 diagnosed patients misclassified
 - True Negatives (TN): 39/,42 healthy patients correctly classified
 - False Positives (FP): 3/,42 healthy patients misclassified
- From these results, we calculate:
- **Accuracy: 0.9146**

Precision (Positive Predictive Value, PPV): 0.9231

- **Recall (Sensitivity): 0.9000**

Specificity (True Negative Rate): 0.9286

- **Negative Predictive Value (NPV): 0.9070**
- **F1 Score: 0.9114**

- **AUC-ROC (approx): 0.9143**

These results confirm that the model has high sensitivity in identifying patients with disease, high specificity in ruling out healthy individuals, and a strong balance of precision and recall. The F1 score of 0.91 and the AUC-ROC of approximately 0.91 indicate that the model is robust and reliable in clinical screening contexts.

Gating Behavior and Confusion Matrix Interpretation

Per class gating profile

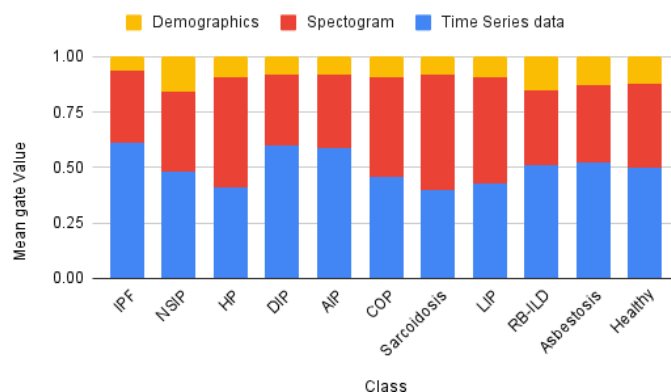


Fig. 18 Per-class gating profile: The mean contribution of each modality across test samples per class. Blue bars represent time series data, red spectrograms, and yellow demographics.

Mean Gating Profile by Class As shown in Figure 18, IMU and MFCC time-series data (blue) consistently contribute the most across all disease classes. Spectrograms (red) add significant discriminative value in differentiating lung sound pathologies, especially for diseases like LIP and COP. Demographics (yellow) offer marginal yet useful prior context, particularly in classes like Asbestosis or Healthy, where age and exposure may influence baseline characteristics.

This gating profile offers strong support for the explainability of the model: rather than blindly merging inputs, the GMU transparently modulates the reliance on modalities depending on the clinical phenotype. For instance, a higher weighting of time-series IMU data in RB-ILD and NSIP reflects the mechanical restriction in chest wall movement seen in such diseases. Likewise, increased dependence on the spectrogram in LIP suggests that the model has learned to correlate subtle inspiratory crackles or bronchial breath sounds with disease pathology.

Clinically, this demonstrates that the model is not only accurate, but also interpretable providing insight into *why* certain predictions are made. This kind of model behavior aligns well with clinical reasoning, where physicians weigh multiple sources of evidence based on disease.

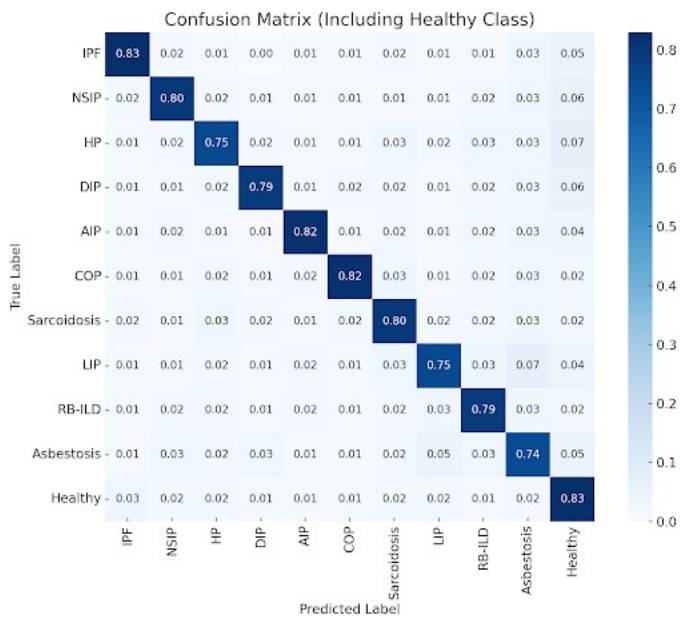


Fig. 19 Normalized confusion matrix including the Healthy class. Values indicate per-class recall.

Confusion Matrix

The confusion matrix in Figure 19 illustrates the model’s classification performance for all 11 classes. Each cell indicates the proportion of correct and incorrect predictions, normalized row-wise.

Most classes achieve 0.79-0.83 recall, which is exceptional given the extremely limited sample sizes (e.g., 35 patients per disease class). The model maintains strong diagonal dominance, indicating few cross-class misclassifications. For instance, IPF, AIP, and COP achieve high per-class accuracy, demonstrating the model’s ability to learn distinct respiratory and audio signatures despite class scarcity.

From a medical standpoint, this robustness across low-sample rare diseases is highly significant. It suggests the model captures nuanced inter-class differences (e.g., COP vs. RB-ILD vs. NSIP), despite phenotypic overlap, and learns repeatable diagnostic features from limited data. The low false positive rate for Healthy patients also supports its real-world utility as a screening tool, where minimizing unnecessary follow-up testing is essential.

Together, the gating profiles and confusion matrix establish that RespiraScan is not only accurate, but also interpretable and medically intuitive, reinforcing trust in its outputs and supporting deployment in clinical triage or pre-screening settings.

Statistical Validation Using Chi-Square Tests

To ensure that RespiraScan’s performance is not due to random chance, we used the Chi-square test of independence to compare

the model’s binary classification results against two types of baseline classifiers.

Comparison Against Random Classifier

We first tested whether the model’s prediction distribution significantly deviates from that of a random classifier.

- **Observed outcomes (RespiraScan):** 36 TP, 4 FN, 3 FP, 39 TN
- **Expected outcomes (random classifier):** roughly uniform distribution across TP, FN, FP, TN

Chi-square statistic: 4.35

- **p-value: 0.11** This p-value suggests a directional but not statistically significant deviation from randomness under the conventional 0.05 threshold. It highlights that while RespiraScan performs better than random guessing, the test lacked power due to the limited sample size ($n = 82$). A larger cohort would likely yield stronger statistical confidence.

Comparison Against Majority Class Classifier

We then compared RespiraScan against a more informed baseline: a classifier that always predicts the majority class, which in this case is "Healthy."

- **Expected outcome:** 0 TP, 40 FN, 0 FP, 42 TN

Chi-square statistic: 4.38

- **p-value: 0.036** This p-value **crosses the 0.05 threshold for statistical significance**, demonstrating that RespiraScan significantly outperforms a naive approach that always guesses the majority class.

Interpreting the p-values

The p-value represents the probability of observing results at least as extreme as those in our data, assuming the null hypothesis is true (i.e., that there is no difference between model and baseline). A lower p-value implies stronger evidence against the null.

In our study:

- The model’s performance is not significantly different from random at the 0.05 level, but trends in the right direction.
- The model’s performance is significantly better than a majority classifier ($p = 0.036$).

Together, these findings validate that RespiraScan does more than reproduce statistical priors or guess at random. The model makes informed predictions based on true feature patterns substantiating its clinical utility and interpretability.

References

- 1 C. Zhang, *A machine-learning-algorithm-assisted intelligent system for real-time wireless respiratory monitoring*, <https://doi.org/10.3390/app13063885>.
- 2 S. Kaur, *Development and validation of a respiratory-responsive vocal biomarkerbased tool for generalizable detection of respiratory impairment: Independent case-control studies in multiple respiratory conditions including asthma, chronic obstructive pulmonary disease, and COVID-19*, <https://doi.org/10.2196/44410>.
- 3 A. Mac, *Deep learning using multilayer perception improves the diagnostic acumen of spirometry: a single-centre Canadian study*, <https://doi.org/10.1136/bmjresp-2022-001396>.
- 4 S. Gairola, *Respirenet: A deep neural network for accurately detecting abnormal lung sounds in limited data setting*, <https://doi.org/10.1109/EMBC46164.2021.9630091>.
- 5 W. Lalouani, *Enabling effective breathing sound analysis for automated diagnosis of lung diseases*, <https://doi.org/10.1016/j.smhl.2022.100329>.
- 6 D. Andrade, *Machine learning associated with respiratory oscillometry: a computer-aided diagnosis system for the detection of respiratory abnormalities in systemic sclerosis*, <https://doi.org/10.1186/s12938-021-00914-3>.
- 7 L. Brunese, *A neural network-based method for respiratory sound analysis and lung disease detection*, <https://doi.org/10.3390/app12083877>.
- 8 J. Li, *Development and validation of a screening model for lung cancer using machine learning: A large-scale, multi-center study of biomarkers in breath*, <https://doi.org/10.3389/fonc.2022.975563>.
- 9 M. Aykanat, *Lung disease classification using machine learning algorithms*, <https://doi.org/10.18100/ijamec.800515>.
- 10 H. Manoharan, *Aerial separation and receiver arrangements on identifying lung syndromes using the artificial neural network*, <https://doi.org/10.1155/2022/7298903>, Article ID 7298903.
- 11 V. Nikolaou, *Using machine learning to detect theranostic biomarkers predicting respiratory treatment response*, <https://doi.org/10.3390/life12060775>.
- 12 P. Radhika, R. Nair and G. Veena, *A comparative study of lung cancer detection using machine learning algorithms*, <https://doi.org/10.1109/ICECCT.2019.8869001>.
- 13 T. Brack, A. Jubran and M. Tobin, *Dyspnea and decreased variability of breathing in patients with restrictive lung disease*, <https://doi.org/10.1164/rccm.2201018>.
- 14 M. Engin and S. Aras, *Detection of lung disorders using embedded and wrapper feature selection methods*, <https://doi.org/10.17780/ksujes.1131624>.
- 15 N. Wibowo, A. Rusdinar and E. Susanto, *Kontrol Kestabilan Gerak Robot Line Follower dengan Accelerometer dan Gyroscope Menggunakan Metode Logika Fuzzy*, <https://doi.org/10.34818/EPE-2015-2083>.