Acoustic Enclosure Optimizer – Master User Manual

Version 1.0

This manual consolidates beginner to expert-level documentation for the Acoustic Enclosure Optimizer tool.

# Table of Contents

1. 1. Introduction2. User Guide (Quick Start)3. Intermediate User Guide4. Advanced User Guide5. Expert Code Guide6. Appendix A: Full MATLAB Script7. Appendix B: References
2. 2. Quick Start Guide
3. 3. Intermediate User Guide
4. 4. Advanced User Guide
5. 5. Expert Code Guide
6. 6. Appendix A: Full MATLAB Script
7. 7. Appendix B: References

# 1. Introduction

This document serves as a complete manual for the Acoustic Enclosure Optimizer, a MATLAB-based tool designed to reduce noise breakout from industrial enclosures. It uses physics-based models and optimization algorithms to recommend optimal acoustic treatments and enclosure designs. The manual includes tutorials, technical documentation, and implementation notes for all user levels.

# 6. Appendix A: Full MATLAB Script

Below is the full source code for the MATLAB-based Acoustic Enclosure Optimizer. Each function is included for transparency and customization.

% Acoustic Enclosure Optimizer with Proper Membrane Damping (Target 40 dBA)

clear; clc;

%% === USER-DEFINED ENCLOSURE SIZE ===

L = 9; W = 5; H = 3;

enclosure\_dims = [L, W, H];

%% === FIXED INPUTS ===

panel\_density = 2710; % kg/m³ (steel)

membrane\_density = 1800; % kg/m³ (butyl-like)

frequencies = [125 250 500 1000 2000 4000 8000];

A\_weighting = [-16.1 -8.6 -3.2 0 1.2 1.0 -1.1];

L\_internal\_raw = [94 84 85 83 80 75 66];

%% === TARGET SETTINGS ===

dBA\_target = 40;

penalty\_weight = 1000;

%% === DESIGN VARIABLES (11 total) ===

x\_default = [0.03, 30, 0.12, 1e-3, 0.015, 1.5e-3, 1.5e-3, 100, 0.05, 1e-3, 2e-3];

x0 = x\_default;

lb = [0.01, 10, 0.05, 0.0005, 0.000, 1e-3, 1e-3, 30, 0.025, 0.5e-3, 0];

ub = [0.10, 100, 0.30, 0.003, 0.050, 5e-3, 5e-3, 150, 0.25, 3e-3, 5e-3];

%% === OPTIMIZATION ===

opt\_fun = @(x) cost\_function\_with\_target(x, L\_internal\_raw, ...

panel\_density, membrane\_density, enclosure\_dims, frequencies, A\_weighting, dBA\_target, penalty\_weight);

options = optimset('Display','iter','TolX',1e-3);

[x\_opt, cost\_min] = fmincon(opt\_fun, x0, [], [], [], [], lb, ub, [], options);

%% === BREAKOUT CALCULATION ===

breakout\_after = compute\_L\_breakout(x\_opt, L\_internal\_raw, ...

panel\_density, membrane\_density, enclosure\_dims, frequencies);

L\_A = breakout\_after + A\_weighting;

final\_dBA = 10 \* log10(sum(10.^(L\_A / 10)));

%% === OUTPUT ===

fprintf('\n=== OPTIMIZATION RESULTS ===\n');

fprintf('Target breakout: %.1f dBA\n', dBA\_target);

fprintf('Final breakout : %.1f dBA\n\n', final\_dBA);

labels = {...};

for i = 1:length(x\_opt)

fprintf('%-35s: %.4f\n', labels{i}, x\_opt(i));

end

% (Functions omitted here for brevity but included in actual file)

# 7. Appendix B: References

- Delany, M.E. and Bazley, E.N. (1970). Acoustical properties of fibrous absorbent materials.

- ISO 9613-1: Acoustics – Attenuation of sound during propagation outdoors.

- MATLAB Documentation on `fmincon` and optimization toolboxes.

- Kinsler, L.E. et al., Fundamentals of Acoustics.

Acoustic Enclosure Optimizer – User Guide

# 1. Introduction

This tool is designed to help engineers and designers optimize the soundproofing performance of an acoustic enclosure. The program models the transmission loss, absorption, and breakout noise from an enclosure and compares it to a user-defined target A-weighted sound level (dBA).

# 2. Quick Start for Novice Users

Follow these steps to get started with minimal effort:

1. Open the MATLAB script `Acoustic\_Enclosure.m`.

2. Run the script. It will automatically start the optimization process.

3. Once complete, the optimized parameters and breakout noise levels will be printed to the Command Window.

4. A plot will be generated showing breakout noise and transmission loss.

5. A figure window will also display the breakout noise values per frequency in table form.

# 3. Acoustic Path Overview

The diagram below outlines the typical sequence of acoustic components from internal noise to breakout, modeled in the program:

![](data:application/vnd.openxmlformats-officedocument.wordprocessingml.header+xml;base64,)

# 4. Technical Breakdown of the Script

The script performs an optimization of the enclosure's multi-layer design based on acoustic and physical properties. Below is an explanation of the key sections:

## 4.1 User-Defined Inputs

- `L`, `W`, `H`: Dimensions of the enclosure.

- `L\_internal\_raw`: Internal noise levels at standard octave frequencies.

- `dBA\_target`: The desired A-weighted breakout noise level.

## 4.2 Design Variables

These 11 variables define the materials and structure of the enclosure (thicknesses, densities, porosities, etc.). They are optimized to achieve the target breakout noise.

## 4.3 Optimization Process

The objective function (`cost\_function\_with\_target`) calculates the breakout noise after transmission through the enclosure. It computes the internal sound power, applies absorption, and evaluates transmission loss using physics-based models. A penalty is applied if the breakout level exceeds the target.

## 4.4 Calculation Functions

- `compute\_TL`: Calculates transmission loss using panel mass, damping, and infill properties.

- `compute\_L\_breakout`: Estimates breakout noise by combining internal SPL and TL.

- `delany\_bazley\_absorption`: Models absorption of the rockwool infill.

- `multilayer\_lining\_absorption`: Computes absorption from perforated layers, foam, and air gaps.

## 4.5 Output and Visualization

After optimization, results are printed to the Command Window. A plot and a figure-based table show raw and A-weighted breakout noise levels by frequency.

# 5. For Advanced Users

Advanced users can modify material properties, change the optimization bounds, or extend the model to include additional frequency bands or materials. Functions are modular and can be extended to interface with measurement data or more advanced acoustic models.

Acoustic Enclosure Optimizer – Intermediate User Guide

# 1. Introduction

This intermediate guide is intended for users who are comfortable with basic MATLAB scripting and wish to customize the acoustic enclosure optimizer further. It focuses on adjusting parameters such as enclosure size, material properties, frequency bands, and optimization targets.

# 2. Parameter Overview

The script is structured to allow direct modification of variables at the beginning of the file. These parameters control geometry, source levels, material characteristics, and optimization constraints.

# 3. Changing Enclosure Dimensions

To change the size of the enclosure, locate the following lines at the top of the script:

```matlab  
L = 9; W = 5; H = 3;  
enclosure\_dims = [L, W, H];  
```

Update the values of `L`, `W`, and `H` (in meters) to reflect your enclosure's length, width, and height. These values influence the internal volume, surface area, and room acoustics modeling.

# 4. Adjusting Internal Noise Levels

The vector `L\_internal\_raw` defines the baseline internal SPL at each octave band:

```matlab  
L\_internal\_raw = [94 84 85 83 80 75 66];  
```

You can modify this array to reflect different equipment or conditions inside the enclosure. Ensure that the number of entries matches the number of frequency bands.

# 5. Customizing Frequency Bands

The script uses standard octave bands. To change them, update:

```matlab  
frequencies = [125 250 500 1000 2000 4000 8000];  
```

You must also update the corresponding `A\_weighting` and `L\_internal\_raw` arrays to match.

# 6. Material Property Variables

Each variable in `x` corresponds to a physical property:

- `x(1)` – Foam thickness (m)  
- `x(2)` – Foam density (kg/m³)  
- `x(3)` – Perforation ratio (φ)  
- `x(4)` – Hole diameter (m)  
- `x(5)` – Air gap thickness (m)  
- `x(6)` – Inner steel panel thickness (m)  
- `x(7)` – Outer steel panel thickness (m)  
- `x(8)` – Rockwool density (kg/m³)  
- `x(9)` – Rockwool thickness (m)  
- `x(10)` – Perforated panel thickness (m)  
- `x(11)` – Membrane thickness (m)

# 7. Changing Optimization Bounds

To constrain the optimizer, adjust the lower (`lb`) and upper (`ub`) bounds:

```matlab  
lb = [0.01, 10, 0.05, 0.0005, 0.000, 1e-3, 1e-3, 30, 0.025, 0.5e-3, 0];  
ub = [0.10, 100, 0.30, 0.003, 0.050, 5e-3, 5e-3, 150, 0.25, 3e-3, 5e-3];  
```

These vectors limit the search space for each design variable.

# 8. Adjusting the Target Breakout Level

To change the goal for breakout noise, modify:

```matlab  
dBA\_target = 40;  
```

This defines the optimization goal in A-weighted decibels.

# 9. Running the Optimizer

Once all parameters are set, simply run the script. The optimizer will attempt to meet the target dBA breakout level by tuning the design variables within the specified bounds.

# 10. Interpreting the Results

The Command Window will display the optimized parameters and a table showing breakout levels at each frequency. A plot and interactive table figure are also shown. These outputs help verify performance against the target.

Acoustic Enclosure Optimizer – Advanced User Guide

# 1. Introduction

This advanced guide is for users with experience in acoustic modeling and MATLAB programming. It details the internal structure of the optimizer, mathematical models used, and potential extensions for integrating measurement data, advanced materials, or multi-objective optimization.

# 2. Functional Architecture

The script consists of the following major components:

- \*\*Input Section\*\*: Defines dimensions, material properties, frequency bands, target breakout level.  
- \*\*Design Variable Bounds\*\*: Lower and upper bounds for each design variable.  
- \*\*Optimization Setup\*\*: Uses MATLAB's `fmincon` to minimize a penalty-based cost function.  
- \*\*Acoustic Modeling Functions\*\*: Includes internal SPL, transmission loss (TL), absorption, and breakout noise.  
- \*\*Output Section\*\*: Plots, tables, and Command Window summaries.

# 3. Cost Function: `cost\_function\_with\_target`

The cost function evaluates the performance of a given enclosure configuration. It calculates breakout noise and applies a quadratic penalty for exceeding the A-weighted breakout target:

```matlab  
cost = penalty^2 \* penalty\_weight;  
```

Where `penalty = max(0, dBA\_actual - dBA\_target)` ensures only excess breakout levels contribute to the cost.

# 4. Acoustic Models Used

## 4.1 Internal Sound Power

Calculated using room volume, surface area, and average absorption to estimate reverberation time (RT60), room gain, and modal boost:

```matlab  
RT60 = 0.161 \* V / (alpha\_avg \* S + eps);  
```

## 4.2 Transmission Loss (TL)

Based on a simplified mass law model with resonance correction and membrane damping factor:

```matlab  
TL\_lin = 1 + (ω² \* m1 \* m2) / ((m1 + m2)² \* resonance\_term);  
```

## 4.3 Delany-Bazley Rockwool Model

Implements empirical formulas to estimate impedance and propagation constants for fibrous materials:

```matlab  
Zc = ρc(1 + 0.0571 \* X^-0.754 - j0.087 \* X^-0.732);  
```

## 4.4 Multilayer Absorber Model

Combines perforated panel impedance, air gap, and foam layer using series impedance formulation to compute total reflection and absorption coefficient.

# 5. Extending the Model

## 5.1 Frequency Resolution

To increase frequency resolution, extend the `frequencies` array to include 1/3 octave bands or any custom set. Ensure all related arrays (e.g., `A\_weighting`, `L\_internal\_raw`) are updated accordingly.

## 5.2 Data Integration

Replace `L\_internal\_raw` with real measurement data or integrate with a sound source model. The optimizer will still operate as designed.

## 5.3 Multi-objective Optimization

To balance cost, weight, or thickness along with acoustic performance, modify the `cost\_function\_with\_target` to incorporate additional terms. Use techniques such as weighted sums or Pareto front analysis.

## 5.4 Visualization and Export

Save results using MATLAB’s `writetable`, `savefig`, or export plots and data to Excel for reporting. Combine with the figure-based breakout table for presentations.

# 6. Conclusion

The optimizer provides a flexible framework for acoustic design. With knowledge of acoustic modeling and MATLAB scripting, users can extend it to new applications and configurations. This guide should support confident modification and enhancement of the tool.

Acoustic Enclosure Optimizer – Expert Line-by-Line Code Guide

# 1. Introduction

This expert guide provides a detailed line-by-line explanation of the MATLAB script `Acoustic\_Enclosure.m`, which performs acoustic optimization for a multi-layer soundproofing enclosure. Each line is annotated to describe its purpose, function, and role within the overall algorithm.

# 2. Script Sections and Explanations

## 2.1 User Inputs

```matlab  
clear; clc;  
```

Clears the workspace and command window for a fresh run.

```matlab  
L = 9; W = 5; H = 3;  
enclosure\_dims = [L, W, H];  
```

Defines the enclosure dimensions and combines them into a vector.

```matlab  
panel\_density = 2710;  
membrane\_density = 1800;  
```

Sets material densities for steel panels and membrane layers.

```matlab  
frequencies = [125 250 500 1000 2000 4000 8000];  
A\_weighting = [-16.1 -8.6 -3.2 0 1.2 1.0 -1.1];  
L\_internal\_raw = [94 84 85 83 80 75 66];  
```

Defines standard octave bands, corresponding A-weighting, and raw internal noise levels.

```matlab  
dBA\_target = 40;  
penalty\_weight = 1000;  
```

Specifies the optimization target and the penalty multiplier for excess breakout noise.

## 2.2 Design Variables and Bounds

```matlab  
x\_default = [...];  
x0 = x\_default;  
```

Initial guess for the optimizer based on expected material and thickness values.

```matlab  
lb = [...];  
ub = [...];  
```

Lower and upper bounds for design variables (thicknesses, densities, porosities).

## 2.3 Optimization Call

```matlab  
opt\_fun = @(x) cost\_function\_with\_target(...);  
options = optimset(...);  
[x\_opt, cost\_min] = fmincon(...);  
```

Defines the cost function handle, sets optimization options, and performs constrained optimization.

## 2.4 Post-Processing

```matlab  
breakout\_after = compute\_L\_breakout(...);  
L\_A = breakout\_after + A\_weighting;  
```

Computes the final breakout noise (raw and A-weighted).

```matlab  
final\_dBA = 10 \* log10(sum(10.^(L\_A / 10)));  
```

Calculates overall dBA level from frequency-weighted levels.

```matlab  
fprintf(...) and for loop  
```

Displays optimized variable values in the Command Window.

```matlab  
fprintf table loop (Breakout by Frequency)  
```

Prints raw and A-weighted breakout levels for each frequency.

## 2.5 Plotting Results

```matlab  
semilogx(...) and legend(...)  
```

Creates the comparison plot of internal noise, TL, breakout (dB and dBA).

## 2.6 Function Definitions

```matlab  
function cost = cost\_function\_with\_target(...)  
```

Main cost function that integrates TL, internal level, surface area, and penalty.

```matlab  
function TL = compute\_TL(...)  
```

Transmission loss model including damping and resonance correction.

```matlab  
function L\_breakout = compute\_L\_breakout(...)  
```

Breakout noise calculated using TL, internal levels, and enclosure surface area.

```matlab  
function L\_internal = compute\_L\_internal(...)  
```

Computes room-corrected internal sound pressure levels using RT60, gain, and modal boost.

```matlab  
function alpha = delany\_bazley\_absorption(...)  
```

Absorption coefficient for porous materials based on empirical Delany-Bazley equations.

```matlab  
function alpha = multilayer\_lining\_absorption(...)  
```

Models multilayer surface absorption using impedance-based formulation.

# 3. Conclusion

This guide details the script logic and acoustic principles embedded in the MATLAB code. Expert users can leverage this structure to integrate new materials, advanced damping models, measurement datasets, or custom acoustic constraints.