

AKADEMIA GÓRNICZO-HUTNICZA
WYDZIAŁ INŻYNIERII MECHANICZNEJ I ROBOTYKI
KIERUNEK: INŻYNIERIA AKUSTYCZNA



METODY I NARZĘDZIA PROGRAMOWE W AKUSTYCE

Projekt MES

Obliczenia charakterystyk straty wtrącenia akustycznego tłumika
refleksyjnego oraz optymalizacja geometrii kanału

Grupa 3, III rok

Maja Szydłowska,
Justyna Szymańska,
Dominika Woźniak,
Alexander Stefani

Kraków, 15 czerwca 2020

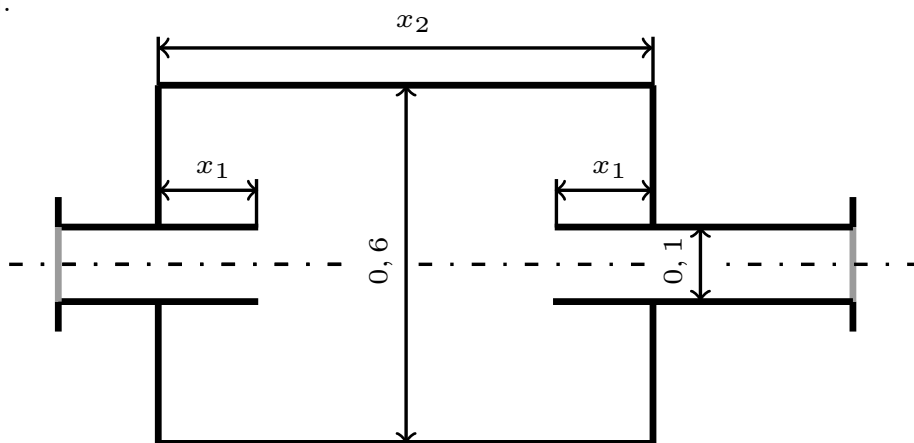
I. Wstęp

Przedmiotem projektu jest analiza tłumika refleksyjnego o wymiarach stałych i schemacie przekroju opisanym na [rysunku 1](#). Zakres wymiarów sparаметryzowanych jako zmienne wejściowe jest przedstawiony w [tabeli 1](#), wraz analizowanym pasmem częstotliwości.

Zadaniem grupy było dokonanie obliczeń charakterystyki częstotliwościowej straty wtrącenia opisanego tłumika kołowego w pakiecie obliczeniowym do metody elementów skończonych – Elmer FEM, który posiada moduł dla akustycznego równania Helmholtza.

Następnie, zgodnie z wybranym przez grupę planem eksperymentu opisanym poniżej, zostały wyznaczone charakterystyki straty wtrącenia dla zmiennych wejściowych, które metodą optymalizacji dadzą średnice kanałów: x_1 , x_2 tłumika o największej całkowitej stracie wtrącenia w zadanym paśmie częstotliwości, metodą omiatania częstotliwościami (scanning).

Porównawczo, zastosowano metodę omiatania parametrów wejściowych z małym krokiem dyskretyzacji (sweep) w celu uzyskania precyzyjnych wyników. Jest ona potrzebna do weryfikacji poprawności planu eksperymentu, który w naszej grupie uległ modyfikacji opisanej niżej.



Rys. 1.: Schemat przekroju tłumika refleksyjnego dla grupy 3.

Tabela 1.: Zakres parametrów wejściowych oraz kroki dyskretyzacji dla metody omiatania

Parametr	Minimum	Maksimum	Krok dyskretyzacji
x_1 [m]	0,05	0,4	0,05 (sweep)
x_2 [m]	0,2	1,0	0,1 (sweep)
f [Hz]	100	500	1 (CCI/ sweep ¹)

I.A. Problemy

I.A.1. Wybór planu eksperymentu

Jak widać na [rys. 1](#)., fizyczny warunek na istnienie tłumika zachodzi, gdy parametry spełniają nierówność: $x_2 > 2x_1$, inaczej współrzędne siatki MES nie miałyby rosnących wartości, a kanał tłumika zostałby zamknięty. To znacznie ogranicza zakres parametrów wejściowych, które rozplanowane na planie kompozycyjnym dadzą mniej punktów do interpolacji funkcji modelu zastępczego (powierzchnia odpowiedzi). To dało w rzeczywistości 6 z 9 możliwych punktów na planie kompozycyjnym zarówno CCI jak i CCC.

¹Dało to aż 17 644 par symulacji MES, ale dzięki temu, nie ma problemu z aliasingiem częstotliwości.

I.A.2. Wyznaczanie parametru modelu zastępczego

Dodatkowym problemem jest brak zdefiniowanej ogólnej funkcji celu dla powierzchni odpowiedzi, ponieważ strata wtrącenia $IL(f)$ jest charakterystyką częstotliwościową [1]. Nie można zatem wykonać jednego planu eksperymentu dla całego pasma częstotliwości, należy przyjąć jakąś funkcję zastępczą dającą opisać się na całym przedziale. Wykluczono natomiast możliwość filtrowania charakterystyki w pasmach tercjowych/oktawowych i wyznaczania n -planów eksperymentu (gdzie n to liczba pasm), ponieważ celem zadania nie było przeprowadzenie optymalizacji tłumika dla wybranych arbitralnie częstotliwości, a dla całego pasma.

I.A.3. Długość wlotu i wylotu oraz wektor współrzędnych X domen siatki

Jak widać z rysunku 1., zadana geometria kanału wlotu jak i wylotu nie jest przedstawiona w skali odnosząc się do x_1 i x_2 . Nie zostało jednak powiedziane jaka powinna być stała długość wlotu i wylotu, więc grupa przyjęła długość wlotu na **5 m**, a wylotu na **10 m**, proporcjonalnie do stosunku ich długości na rysunku. Aby wlot i wylot nie miały dużego wpływu na zjawiska falowe w tłumiku, ich długość musi być znacznie większa od największej długości fali rozpatrywanej w paśmie – (jest to dla 100 Hz: 3,43 m), więc warunek został spełniony. Zgodnie z [1], przyjęto fizyczną grubość ścianek = 1 cm.

Aby ułatwić obliczenia wartości ciśnienia dla wylotu, przyjęto środek układu współrzędnych siatki (0,0) jako koniec wylotu położony w środku symetrii kanału tłumika.

I.B. Rozwiązania:

I.B.1. Ad A.1.

Grupa zaproponowała jako rozwiązanie zastosowanie uboższego **planu eksperymentu CCI (6-punktowego)**, rozszerzonego o walidację z dokładnymi wartościami interpolowanej powierzchni odpowiedzi metodą omiatania po wszystkich parametrach wejściowych z małym krokiem. Odbywa się to kosztem czasu obliczeń symulacji MES, natomiast daje pełną informację o punktach pośrednich między punktami planu eksperymentu. Początkowo wybrano plan CCC, który zgodnie z [2]: „pozwala na uzyskanie wysokiej jakości dopasowania powierzchni odpowiedzi w całym badanym obszarze”, niestety zakres parametrów wejściowych przekracza przedział nieujemnych (fizycznych) wartości, więc porzucono ten plan.

I.B.2. Ad A.2.

Aby uzyskać funkcję całkowitej straty wtrącenia w artykułach o tłumieniu przegród dźwiękochłonnych znaleziono inną definicję na stratę wtrącenia IL (*insertion loss*) [3], [4]. Jest to **poziom stosunku mocy całkowitych**: źródła dźwięku do wypromieniowywanej za odgradą tłumiącą. Dzięki temu, definicja ta jest prostsza w naszym problemie, ponieważ należy wyznaczyć dla każdej pary tłumik-falowod estymaty całkowitych mocy akustycznych.

$$\begin{aligned} \text{Zgodnie z definicją mocy akustycznej: } P &= \frac{I}{S} = \frac{\langle p^2 \rangle}{ZS} \\ \text{oraz definicją na moc „fizyczną”} &: P = \frac{E_{tot}}{\Delta t} = E_{tot} \Delta f = \sum_i E(f[i]) \Delta f \end{aligned}$$

– można wywnioskować, że bardzo dobrą estymatą mocy całkowitej będzie całka (dyskretna suma) w zadanym paśmie częstotliwości, pod warunkiem, że źródło dźwięku będzie wypromieniowywać stałą gęstość mocy widmowej, ale tylko w tym paśmie częstotliwości.

Jako, że impedancja właściwa Z oraz średnica przekroju S nie zmieniają się, całkowita strata wtrącenia będzie poziomem stosunku mocy całkowitych wypromieniowywanych przez falowod płaski i tłumik akustyczny (całek z uśrednionych kwadratów amplitud ciśnienia $\langle p^2 \rangle$ przy wylocie, czyli energii):

$$IL_{tot} = 10 \log_{10} \frac{\sum_i \langle p_{bez}^2[i] \rangle \Delta f}{\sum_i \langle p_{tłum}^2[i] \rangle \Delta f} \quad (1)$$

II. Algorytm obliczeniowy

II.A. Opis działania

Jako, że grupa dostała tłumik geometrycznie podobny do tłumika opisanego na laboratorium, w niniejszej pracy nie będzie ponownie przedstawiana struktura domen obliczeniowych siatki, a grupa odsyła do rysunku w punkcie 4.1. konspektu do lab. 2. [1].

W celu wykonywania algorytmu i wywoływania podprogramów, wybrano napisanie głównego skryptu `runmuffler.sh` dla powłoki systemowej Linuxa – Bash oraz dokonywanie obliczeń macierzowych i interpolacji w skryptach pakietu Octave: `replace.m` – tworząca wektor siatki na osi X , `cci.m` – wyznaczająca punkty planu CCI i `optimize.m` – opisana dalej. Kluczem do prostoty obliczeń jest automatyzacja podmieniania parametrów wejściowych w plikach: **siatki tłumika** – `tlumik.grd`, **siatki płaskiego falowodu** – `plaski.grd` oraz **solwera ElmerSolver** – `case.sif`, dokonywana wyrażeniem regularnym `sed` na znaku `@`. Lokalizacja szablonów tekstowych dla plików `case.sif`, `tlumik.grd`, `plaski.grd` znajduje się podfolderze `./data`.

Punkty należące do zbioru punktów „fizycznych” zwraca skrypt `cci.m` do pliku tekstowego `cci-t1t2.txt`, następnie główny skrypt wykonawczy `runmuffler.sh` najpierw konwertuje pliki wyników symulacji MES – `output.dat` do plików:

`cci-tlumik.txt`, `cci-plaski.txt` z wartościami oddzielonymi przecinkami, ulokowane w folderach: `./output/tlumik`, `./output/plaski`, dopiero wtedy funkcja `optimize.m` pobiera wektory współrzędnych planu CCI z `cci-t1t2.txt`, sortuje wyniki symulacji MES z poprzednio sformatowanych plików tekstowych i przeprowadza optymalizację.

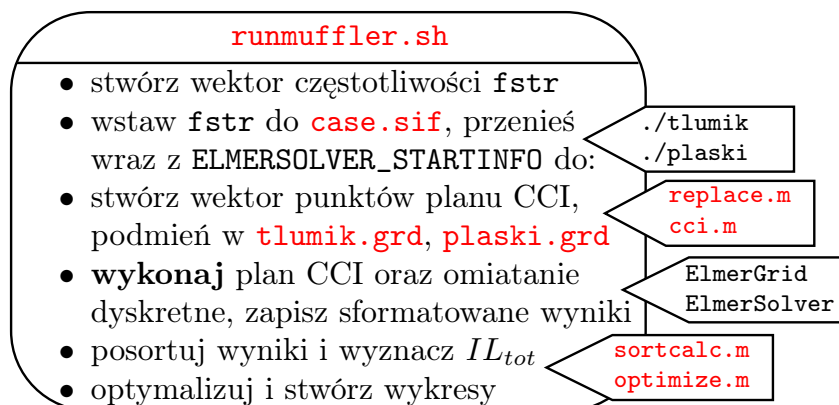
Dodatkowo stworzono podfunkcję `sortcalc.m` wykonującą bezpośrednio sortowanie wyników z zadanych plików do macierzy i wyznaczanie IL_{tot} , by zwrócić je do funkcji `optimize.m`.

II.B. Schemat algorytmu

Schemat procedury obliczeniowej znajduje się na [rysunku 2](#), a listingi plików pod nim. Jako poprawne wywołanie `runmuffler.sh` z poziomu Bash’a podaje się po nazwie argumenty kroków dyskretyzacji: $\Delta x_1, \Delta x_2, \Delta f$, zgodnie z notacją:

`./runmuffler.sh [krok_x1] [krok_x2] [krok_f] [CCI|SWEEP] (2 flagi boolean)`

Dodatkowo wstawiając logiczną flagę po argumentach, np. 01, można wybrać, że moduł MES ma nie obliczać danych z punktów CCI, a tylko z omiatania parametrami x_1, x_2 .



Rys. 2.: Schemat algorytmu głównego skryptu wykonującego

II.C. Listingi plików

Całość kodu źródłowego jest dostępna na platformie GitHub: [\[LINK\]](#)

Na następnych stronach wygenerowano listingi wszystkich kluczowych plików projektu.

Listing 1: Główny skrypt wykonujący (runmuffler.sh)

```

1 #!/bin/bash
2 if [[ $1 =~ "-h" ]]; then
3     echo "RUN-MUFFLER Syntax: ./'basename $0' [x1step] [x2step] [fstep] [CCI|SWEEP](2x bool)"
4     exit 0
5 fi
6 echo FEM started: "date +%H:%M" # mierzenie czasu dla przewidzenia złożoności
7 # stwórz wektor częstotliwości i przygotuj pliki:
8 read -r x1min x1max x2min x2max fmin fmax <<<$(echo 0.05 0.4 0.2 1 100 500);
9 echo "Freq vec:  [$fmin : $3 : $fmax] Hz"
10 freq=() # (fstep=$3)
11 flag=${4:-00}
12 for i in $(seq $fmin $3 $fmax); do freq+=($i); done
13 fstr=$(IFS=' '; echo "${freq[*]}")
14 mkdir -p output/tlumik output/plaski
15 cd output
16 rm -f tlumik/* plaski/*
17 sed "1s/0/$fstr/" ../data/case.txt > tlumik/case.sif
18 number=$(echo "($fmax-$fmin)/$3 + 1" | bc)
19 sed -i "15s/0/$number/" tlumik/case.sif
20 cp tlumik/case.sif plaski/case.sif
21 echo -e "case.sif\n1" > tlumik/ELMERSOLVER_STARTINFO
22 cp tlumik/ELMERSOLVER_STARTINFO plaski/ELMERSOLVER_STARTINFO
23 if [ $flag == 10 ] || [ $flag == 11 ]; then
24     # wykonaj plan eksperymentu:
25     echo "-----EXPERIMENT CCI-----"
26     echo "Calc for:  x1 | x2 | x-coords (meters)"
27     octave --silent --eval "cd ../scripts; cci($x1min,$x1max,$x2min,$x2max)"
28     while read -r x1 x2; do
29         # obliczenia dla tłumika:
30         xline=$(octave --silent --eval "cd ../scripts; replace($x1,$x2)")
31         echo "          $x1 | $x2 | " $xline
32         sed "4s/0/$xline/" ../data/tlumik.txt > tlumik.grd
33         ElmerGrid 1 2 tlumik.grd > /dev/null
34         cd tlumik
35         ElmerSolver > /dev/null
36         cd ..
37         # obliczenia dla płaskiego falowodu:
38         length=$(echo "-15-$x2" | bc) # wpisać zależnie od długości tłumika!
39         sed "4s/0/$length/" ../data/plaski.txt > plaski.grd
40         ElmerGrid 1 2 plaski.grd > /dev/null
41         cd plaski
42         ElmerSolver > /dev/null
43         cd ..
44     done < cci-x1x2.txt
45     # sformatuj wyniki do analizy zamieniając spacje na przecinki:
46     sed -e 's/\s\+/,/g' tlumik/output.dat > cci-tlumik.txt
47     sed -e 's/\s\+/,/g' plaski/output.dat > cci-plaski.txt
48     rm -f tlumik/*.dat plaski/*.dat
49 fi
50 if [ $flag == 01 ] || [ $flag == 11 ]; then
51     # wykonaj omiotanie z dyskretnym krokiem parametrów:
52     echo "-----DISCRETE SWEEP-----"
53     echo "Calc for:  x1 | x2 | x-coords (meters)"
54     rm -f sweep-x1x2.txt
55     for x1 in $(seq $x1min $1 $x1max);do # (x1step=$1; x2step=$2)
56         for x2 in $(seq $x2min $2 $x2max);do
57             if [ $(echo "$x2 > 2*$x1" | bc) -eq 1 ];then
58                 # obliczenia dla tłumika:
59                 echo $x1 $x2 >> sweep-x1x2.txt
60                 xline=$(octave --silent --eval "cd ../scripts; replace($x1,$x2)")
61                 echo "          $x1 | $x2 | " $xline
62                 sed "4s/0/$xline/" ../data/tlumik.txt > tlumik.grd
63                 ElmerGrid 1 2 tlumik.grd > /dev/null
64                 cd tlumik
65                 ElmerSolver > /dev/null
66                 cd ..
67                 # obliczenia dla płaskiego falowodu:
68                 length=$(echo "-15-$x2" | bc) # wpisać zależnie od długości tłumika!
69                 sed "4s/0/$length/" ../data/plaski.txt > plaski.grd
70                 ElmerGrid 1 2 plaski.grd > /dev/null
71                 cd plaski
72                 ElmerSolver > /dev/null
73                 cd ..
74             fi
75         done
76     done # sformatuj wyniki do analizy zamieniając spacje na przecinki:
77     sed -e 's/\s\+/,/g' tlumik/output.dat > sweep-tlumik.txt
78     sed -e 's/\s\+/,/g' plaski/output.dat > sweep-plaski.txt
79 fi # wyznacz IL, optymalizuj i zwróć wynik w konsoli
80 echo -e "FEM finished: 'date +%H:%M'\nPress Enter to close figures and print results in ./output/Optimum.txt"
81 octave --silent --eval "cd ../scripts; optimize($x1min,$1,$x1max,$x2min,$2,$x2max,$fmin,$fmax)" > Optimum.txt
82 cat Optimum.txt
83 cd ..

```

Listing 2: Funkcja replace.m zamieniająca parametry: x_1, x_2 na wektor współrzędnych

```

1 function replace(x1,x2)
2 % wektor o srodku współrzędnych (0,0) na osi symetrii konca wylotu
3 xvec = [-15-x2, -10-x2, -10-x2+x1, -10-x1, -10, 0, 0.1];
4 xstr = "";
5 for i = xvec
6     xstr = [xstr, ' ', num2str(i)];
7 end
8 disp(xstr);

```

Listing 3: Szablon modelu siatki ElmerGrid dla tłumika (tlumik.grd)

```

1 Version = 210903
2 Coordinate System = Cartesian 2D
3 Subcell Divisions in 2D = 6 7
4 Subcell Limits 1 = 0
5 Subcell Limits 2 = -0.31 -0.3 -0.06 -0.05 0.05 0.06 0.3 0.31
6 Material Structure in 2D
7   2 2 2 2 2 2
8   2 1 1 1 2 3
9   2 2 1 2 2 3
10  1 1 1 1 1 3
11  2 2 1 2 2 3
12  2 1 1 1 2 3
13  2 2 2 2 2 2
14 End
15 Materials Interval = 1 1
16 Boundary Definitions
17 ! type      out      int      double      of the boundaries
18 1           2         1         1
19 2           0         1         1
20 3           3         1         1
21 End
22 Numbering = Horizontal
23 Element Degree = 1
24 Element Innernodes = False
25 Triangles = False
26 Plane Elements = 51000

```

Listing 4: Szablon modelu siatki ElmerGrid dla płaskiego falowodu (plaski.grd)

```

1 Version = 210903
2 Coordinate System = Cartesian 2D
3 Subcell Divisions in 2D = 2 3
4 Subcell Limits 1 = 0 0 0.1
5 Subcell Limits 2 = -0.3 -0.05 0.05 0.3
6 Material Structure in 2D
7   2 3
8   1 3
9   2 3
10 End
11 Materials Interval = 1 1
12 Boundary Definitions
13 ! type      out      int      double      of the boundaries
14 1           2         1         1
15 2           0         1         1
16 3           3         1         1
17 End
18 Numbering = Horizontal
19 Element Degree = 1
20 Element Innernodes = False
21 Triangles = False
22 Plane Elements = 51000

```

Listing 5: Szablon pliku obliczeniowego case.sif dla solvera MES: ElmerSolver

```

1 $ freq = 1 0
2 Header
3 CHECK KEYWORDS Warn
4 Mesh DB "." "."
5 Include Path ""
6 Results Directory ""
7 End
8
9 Simulation
10 Max Output Level = 5
11 Coordinate System = Cartesian
12 Coordinate Mapping(3) = 1 2 3
13 Simulation Type = Scanning
14 Output Intervals (2) = 1 1
15 Timestep Intervals (2) = 0
16 Timestep Sizes (2) = 1.0 1.0
17 Steady State Max Iterations = 1
18 Timestepping Method = BDF
19 BDF Order = 1
20 Solver Input File = case.sif
21 !Post File = case.vtu - USPRAWNIE: brak tworzenia ciężkich plików ElmerPost
22 End
23
24 Constants
25 Gravity(4) = 0 -1 0 9.82
26 Stefan Boltzmann = 5.67e-08
27 Permittivity of Vacuum = 8.8542e-12
28 Boltzmann Constant = 1.3807e-23
29 Unit Charge = 1.602e-19
30 End
31
32 Body 1
33 Target Bodies(1) = 1
34 Name = "Body 1"
35 Equation = 1
36 Material = 1
37 End
38
39 Solver 1
40 Equation = Helmholtz Equation

```

```

41 Procedure = "HelmholtzSolve" "HelmholtzSolver"
42 Variable = -dofs 2 Pressure Wave
43 Exec Solver = Always
44 Stabilize = True
45 Bubbles = False
46 Lumped Mass Matrix = False
47 Optimize Bandwidth = True
48 Steady State Convergence Tolerance = 1.0e-5
49 Nonlinear System Convergence Tolerance = 1.0e-7
50 Nonlinear System Max Iterations = 20
51 Nonlinear System Newton After Iterations = 3
52 Nonlinear System Newton After Tolerance = 1.0e-3
53 Nonlinear System Relaxation Factor = 1
54 Linear System Solver = Direct
55 Linear System Direct Method = Umfpack
56 End
57
58 Solver 2
59 Exec Solver = always
60 !USPRAWNIENIE: automatyzacja dopisywania wyników do poprzednich symulacji
61 Equation = String SaveScalars
62 Procedure = File "SaveData" "SaveScalars"
63 Filename = File "output.dat"
64 Variable 1 = Pressure Wave
65 Save Coordinates (4,3) = Real -0.1 0 0\
66                               -0.07 0 0\
67                               -0.04 0 0\
68                               -0.01 0 0
69 File Append = Logical True
70 End
71
72 Equation 1
73 Name = "Equation 1"
74 Frequency = Variable time
75 Real MATC "freq(tx)"
76 Active Solvers(1) = 1
77 End
78
79 Material 1
80 Name = "Air (room temperature)"
81 Density = 1.205
82 Heat expansion Coefficient = 3.43e-3
83 Heat Conductivity = 0.0257
84 Relative Permittivity = 1.00059
85 Heat Capacity = 1005.0
86 Viscosity = 1.983e-5
87 Sound speed = 343.0
88 End
89
90 Boundary Condition 1
91 Target Boundaries(1) = 1
92 Name = "odbicie"
93 Wave Flux 1 = 0
94 Wave Flux 2 = 0
95 End
96
97 Boundary Condition 2
98 Target Boundaries(1) = 2
99 Name = "wymuszenie"
100 Wave Flux 1 = 10
101 End
102
103 Boundary Condition 3
104 Target Boundaries(1) = 3
105 Name = "wylot"
106 Wave impedance 1 = -343
107 End

```

Listing 6: Funkcja cci.m zapisująca ustandaryzowane punkty planu eksperymentu

```

1 function cci(x1min,x1max,x2min,x2max)
2 % numeracja od punktu planu CCI (1,0) przeciwie do wskazówek zegara
3 t1 = [1,1/sqrt(2),0,-1/sqrt(2),-1,-1/sqrt(2),0,1/sqrt(2),0];
4 t2 = [0,1/sqrt(2),1,1/sqrt(2),0,-1/sqrt(2),-1,-1/sqrt(2),0];
5 x1_0 = (x1max + x1min)/2;
6 dx1 = (x1max - x1min)/2;
7 x2_0 = (x2max + x2min)/2;
8 dx2 = (x2max - x2min)/2;
9 x1x2(1,:) = [x1max, x2_0];
10 x1x2(2,:) = [1/sqrt(2)*dx1 + x1_0, 1/sqrt(2)*dx2 + x2_0];
11 x1x2(3,:) = [x1_0, x2max];
12 x1x2(4,:) = [-1/sqrt(2)*dx1 + x1_0, 1/sqrt(2)*dx2 + x2_0];
13 x1x2(5,:) = [x1min, x2_0];
14 x1x2(6,:) = [-1/sqrt(2)*dx1 + x1_0,-1/sqrt(2)*dx2 + x2_0];
15 x1x2(7,:) = [x1_0, x2min];
16 x1x2(8,:) = [1/sqrt(2)*dx1 + x1_0, -1/sqrt(2)*dx2 + x2_0];
17 x1x2(9,:) = [x1_0, x2_0];
18 it = 1;
19 for i = 1:9
20 if x1x2(i,2) > 2*x1x2(i,1)
21 x1x2good(it,:) = x1x2(i,:);
22 t1t2good(it,:) = [t1(i), t2(i)];
23 it = it+1;
24 end
25 end
26 cd ../output
27 dlmwrite('cci-x1x2.txt',x1x2good,'delimiter',' ','precision', '%.2f');
28 save("-ascii","cci-t1t2.txt","t1t2good");

```

Listing 7: Funkcja sortcalc.m sortująca wyniki dla danego pliku i obliczająca $\sum_i \langle p_i^2 \rangle \Delta f$

```

1 function [int_p2,k] = sortcalc(filename,fmin,fmax)
2     cd ../output
3     M = dlmread(filename,',' );
4     [w,k] = size(M);
5     p = zeros(w,1);
6     for i = [2,7,12,17]
7         p = p + sqrt( M(:,i).^2 + M(:,i+1).^2 );
8     end
9     p2 = (p/4).^2;
10    % sortowanie wyników zgodnie z punktami
11    fstep = M(2,22) - M(1,22);
12    number = (fmax-fmin)/fstep + 1;
13    p2sort = zeros(number,w/number);
14    for i = 1:w/number
15        p2sort(1:number,i) = p2(1+(i-1)*number : i*number);
16    end
17    f = fmin:fstep:fmax;
18    [w,k] = size(p2sort);
19    int_p2 = zeros(1,k);
20    for i = 1:k
21        int_p2(i) = trapz(f',p2sort(:,i));
22    end
23    cd ../scripts

```

Listing 8: Funkcja optimize.m wyznaczająca optymalizację tłumika

```

1 function optimize(xlmin,xlstep,xlmax,x2min,x2step,x2max,fmin,fmax)
2     % posortuj wyniki CCI i wyznacz calk strate wtracenia plaskiego falowodu
3     cd ../scripts
4     CCI_intp2bez = sortcalc('cci-plaski.txt',fmin,fmax);
5     % posortuj wyniki CCI i wyznacz calk strate wtracenia tłumika
6     CCI_intp2tlum = sortcalc('cci-tlumik.txt',fmin,fmax);
7     % wyznaczenie calkowitej straty wtracenia
8     CCI_ILtot = 10*log10(CCI_intp2bez./CCI_intp2tlum);
9     cd ../output
10    [CCI_t1,CCI_t2] = textread('cci-t1t2.txt','%f %f');
11    % wykonanie siatki dla punktow z planu CCI
12    x1_0 = (xlmax + xlmin)/2;
13    dx1 = (xlmax - xlmin)/2;
14    x2_0 = (x2max + x2min)/2;
15    dx2 = (x2max - x2min)/2;
16    CCI_x1vec = xlmin:0.001:xlmax; % gesta siatka dla interpolacji
17    CCI_t1vec = (CCI_x1vec-x1_0)/dx1;
18    CCI_x2vec = x2min:0.001:x2max;
19    CCI_t2vec = (CCI_x2vec-x2_0)/dx2;
20    % wyznaczenie wspolczynnikow wielomianu dla planu CCI
21    W = zeros(6);
22    for i = 1:6
23        W(i,:) = [1,CCI_t1(i),CCI_t2(i),CCI_t1(i)*CCI_t2(i),CCI_t1(i)^2,CCI_t2(i)^2];
24    end
25    [CCI_T1,CCI_T2] = meshgrid(CCI_t1vec,CCI_t2vec);
26    % wyznaczenie i plotowanie plaszczyny odpowiedzi
27    a=W\(CCI_ILtot'); % rownanie macierzowe: x=A^(-1)*b'
28    ILsurf = a(1)+a(2)*CCI_T1+a(3)*CCI_T2+a(4)*CCI_T1.*CCI_T2+a(5)*CCI_T1.^2+a(6)*CCI_T2.^2;
29    px = [-1, (0.1-x1_0)/dx1, 1, 1, -1];
30    py = [-1, -1, (0.8-x2_0)/dx2, 1, 1];
31    inpts = inpolygon(CCI_T1,CCI_T2,px,py);
32    ILsurf('inpts' = nan; % wytnij niefizyczny fragment plaszczyny
33    figure('Position', [1000 300 750 600]);
34    surf(CCI_T1,CCI_T2,ILsurf);
35    shading interp;
36    hold on;
37    scatter3(CCI_t1,CCI_t2,CCI_ILtot,80,'r','filled');
38    set(gca,'FontSize',17);
39    xlabel('t_1');
40    ylabel('t_2');
41    zlabel('IL_{tot} [dB]');
42    view(170,20);
43    line = "ILtot values for CCI points:\nt1\tt2\tILtot\n";
44    for i=1:length(CCI_t1)
45        text(CCI_t1(i)+0.3,CCI_t2(i),CCI_ILtot(i)+1,['(',num2str(CCI_t1(i),2),',',',',num2str(CCI_t2(i),2),',')'], 'FontSize',17);
46        line = strcat(line,num2str(CCI_t1(i),2),"\t",num2str(CCI_t2(i),2),"\t",num2str(CCI_ILtot(i),2),"\n");
47    end
48    clb = colorbar(gca,'FontSize',17);
49    set(clb,'YTick',[round(min(min(ILsurf))*10)/10,-4:1,round(max(max(ILsurf))*10)/10]);
50    [CCI_ILtotmax,Idx] = max(ILsurf(:));
51    [CCI_ILtotmaxRow,CCI_ILtotmaxCol] = ind2sub(size(ILsurf), Idx);
52    Popt = [CCI_T1(1,CCI_ILtotmaxCol),CCI_T2(CCI_ILtotmaxRow,1),CCI_ILtotmax];
53    scatter3(Popt(1),Popt(2),Popt(3),'b','filled');
54    text(Popt(1)+0.3,Popt(2),Popt(3)+1,['(',num2str(Popt(1),2),',',',',num2str(Popt(2),2),',')'], 'FontSize',17);
55    contour3(CCI_T1,CCI_T2,ILsurf,15,'b');
56    titletext = "IL_{tot}(t_1,t_2)=";
57    coeffs = {'","t_1","t_2","t_1t_2","t_1^2","t_2^2"};
58    for i = 1:6
59        if a(i)<0
60            titletext=strcat(titletext,num2str(a(i),2),coeffs(i));
61        else
62            titletext=strcat(titletext,"+",num2str(a(i),2),coeffs(i));
63        end
64    end
65    title(titletext);
66    % posortuj wyniki sweepa i wyznacz calk strate wtracenia plaskiego falowodu
67    cd ../scripts
68    [SW_intp2bez, k] = sortcalc('sweep-plaski.txt',fmin,fmax);
69    % posortuj wyniki sweepa i wyznacz calk strate wtracenia tłumika

```



```

70 SW_intp2tlum = sortcalc('sweep-tlumik.txt',fmin,fmax);
71 SW_ILtot = zeros(1,k);
72 for i = 1:k
73     SW_ILtot(i) = 10*log10(SW_intp2bez(i)/SW_intp2tlum(i));
74 end
75 % wykonanie siatki dla punktów dyskretnych ze sweepowania
76 SW_x1vec = xmin:xstep:x1max;
77 SW_t1vec = (SW_x1vec-x1_0)/dx1;
78 SW_x2vec = x2min:x2step:x2max;
79 SW_t2vec = (SW_x2vec-x2_0)/dx2;
80 [SW_T1,SW_T2] = meshgrid(SW_t1vec,SW_t2vec);
81 cd ../output
82 [SW_x1,SW_x2] = textread('sweep-x1x2.txt','%f %f');
83 SW_t1 = (SW_x1-x1_0)/dx1;
84 SW_t2 = (SW_x2-x2_0)/dx2;
85 figure('Position',[600 100 750 600]);
86 scatter3(SW_t1,SW_t2,SW_ILtot,'b','filled');
87 hold on;
88 %SW_ILtot_int = mesh(SW_t1vec,SW_t2vec,interp2(SW_t1,SW_t2,SW_ILtot), SW_t1vec, SW_t2vec, "pchip");
89 %surf(SW_T1,SW_T2,SW_ILtot_int);
90 %shading interp;
91 x1opt = num2str(Popt(1)*dx1 + x1_0);
92 x2opt = num2str(Popt(2)*dx2 + x2_0);
93 disp("Optimized surface interpolant coeffs:");
94 disp(num2str(a));
95 input([line,"Optimum for CCI:\nx1 (m)\tx2 (m)\tILtot (dB)\n",x1opt,"\t",x2opt,"\t",num2str(CCI_ILtotmax,2),"\n"]);

```

Listing 9: Skrypt optimized.sh wykonujący obliczenia MES dla zoptymalizowanego tłumika

```

1 #!/bin/bash
2 line=$(sed -n '13p' ../output/Optimum.txt)
3 xopt=${line}
4 echo "RUN FOR OPTIMIZED MUFFLER: x1=" ${xopt[0]} "m, x2=" ${xopt[0]} "m"
5 echo "Freq vec: [100 : 1 : 500] Hz"
6 freq=()
7 for i in $(seq 100 1 500); do freq+=($i); done
8 fstr=${IFS=$' '; echo "${freq[*]}")
9 cd output
10 rm -f tlumik/* plaski/*
11 sed "1s/@/$fstr/" ../data/case.txt > tlumik/case.sif
12 number=401
13 sed -i "15s/@/$number/" tlumik/case.sif
14 cp tlumik/case.sif plaski/case.sif
15 echo -e "case.sif\n1" > tlumik/ELMERSOLVER_STARTINFO
16 cp tlumik/ELMERSOLVER_STARTINFO plaski/ELMERSOLVER_STARTINFO
17 # obliczenia dla tłumika:
18 xline=$(octave --silent --eval "replace(${xopt[0]},${xopt[0]})")
19 sed "4s/@/$xline/" data/tlumik.txt > tlumik.grd
20 ElmerGrid 1 2 tlumik.grd > /dev/null
21 cd tlumik
22 ElmerSolver > /dev/null
23 cd ..
24 # obliczenia dla płaskiego falowodu:
25 length=$(echo "-15-${xopt[1]}" | bc) # wpisać zależnie od długości tłumika!
26 sed "4s/@/$length/" data/plaski.txt > plaski.grd
27 ElmerGrid 1 2 plaski.grd > /dev/null
28 cd plaski
29 ElmerSolver > /dev/null
30 cd ..
31 sed -e 's/\s+/,./g' tlumik/output.dat > tlumik-optimized.txt
32 sed -e 's/\s+/,./g' plaski/output.dat > plaski-optimized.txt
33 octave ../scripts/optILchar.m
34 cd ..

```

Listing 10: Skrypt optILchar.m obliczający i wykreślający charakterystykę $IL(f)$ dla zoptymalizowanego tłumika

```

1 cd output
2 % posortuj wynik i wyznacz srednie IL płaskiego falowodu
3 M = dlmread("plaski-optimized.txt",',');
4 [w,k] = size(M);
5 pbez = zeros(w,1);
6 for i = [2,7,12,17]
7     pbez = pbez + sqrt( M(:,i).^2 + M(:,i+1).^2 );
8 end
9 pbez = pbez/4;
10 % posortuj wynik i wyznacz srednie IL tłumika
11 M = dlmread("tlumik-optimized.txt",',');
12 [w,k] = size(M);
13 ptlum = zeros(w,1);
14 for i = [2,7,12,17]
15     ptlum = ptlum + sqrt( M(:,i).^2 + M(:,i+1).^2 );
16 end
17 ptlum = ptlum/4;
18 % wyznacz charakterystykę częstotliwościową IL
19 IL = 20*log10(pbez./ptlum);
20 f = 100:1:500;
21 figure('Position',[500 300 1050 500]);
22 plot(f,IL,'LineWidth',2);
23 set(gca,'XGrid','on','LineWidth',1,'XMinorTick','on','YGrid','on','YMinorTick','on');
24 box(gca,'on');
25 xlabel("Częstotliwość [Hz]");
26 ylabel("Strata wtrącenia IL [dB]");
27 title("Charakterystyka częstotliwościowa straty wtrącenia IL_{tot} dla zoptymalizowanego tłumika");
28 input("Press Enter to close figure...");

```

III. Model zastępczy i optymalizacja tłumika

III.A. Plan eksperymentu i algorytm funkcji `optimize.m`

Badanie obiektu przeprowadzono z wykorzystaniem eksperymentu wielopoziomowego. Przy konstruowaniu planu zwracano uwagę na wymiary tłumika, tak aby kolejno przyjmowane wartości z podanych zakresów nie powodowały kolizji elementów. Zastosowano plan CCI wykorzystujący 6 z 9 punktów zlokalizowanych na okręgu wpisanym do kwadratu powierzchni [5]. Trzy punkty zostały wykluczone, ponieważ nie spełniały warunku koniecznego do odpowiedniej konstrukcji tłumika.

Za pomocą odpowiednich zależności zmienne wejściowe x_1, x_2 we współrzędnych naturalnych zostały ustandaryzowane tak, by nowe parametry t_1, t_2 przyjmowały wartości w zakresie $[-1; 1]$.

Następną procedurą było wykonanie obliczenia średniego kwadratu ciśnienia z **4 punktów położonych blisko przed wylotem** $\langle p^2[i] \rangle$ dla każdej zdyskretyzowanej częstotliwości oraz uzyskanie IL_{tot} poprzez aproksymację całek we wzorze (1) metodą trapezów.

Przechodząc do sprowadzania modelu tłumika do **modelu zastępczego** zauważono, że problem interpolacji punktów powierzchnią odpowiedzi, wyrażoną wielomianem 2-go stopnia i wyznaczaniem współczynników wielomianu metodą najmniejszych kwadratów [5], jest równoważny z rozwiązaniem układu poniższych równań liniowych:

$$IL_{tot_j}(t_1[j], t_2[j]) = a_0 + a_1 t_1[j] + a_2 t_2[j] + a_{12} t_1[j] t_2[j] + a_{11} t_1^2[j] + a_{22} t_2^2[j],$$

to układ oznaczony – (6 współczynników a_i , 6 punktów węzłowych z planu CCI), a poprzez zapisanie w równaniu macierzowym:

$$\{IL_{tot_j}\} = \mathbb{W}^{-1} \cdot \{a_j\} \quad (2)$$

– gdzie \mathbb{W} to macierz 6×6 parametrów po podstawieniu do równania powierzchni odpowiadających mu współrzędnych $t_1[j], t_2[j]$ planu CCI 6-punktowego.

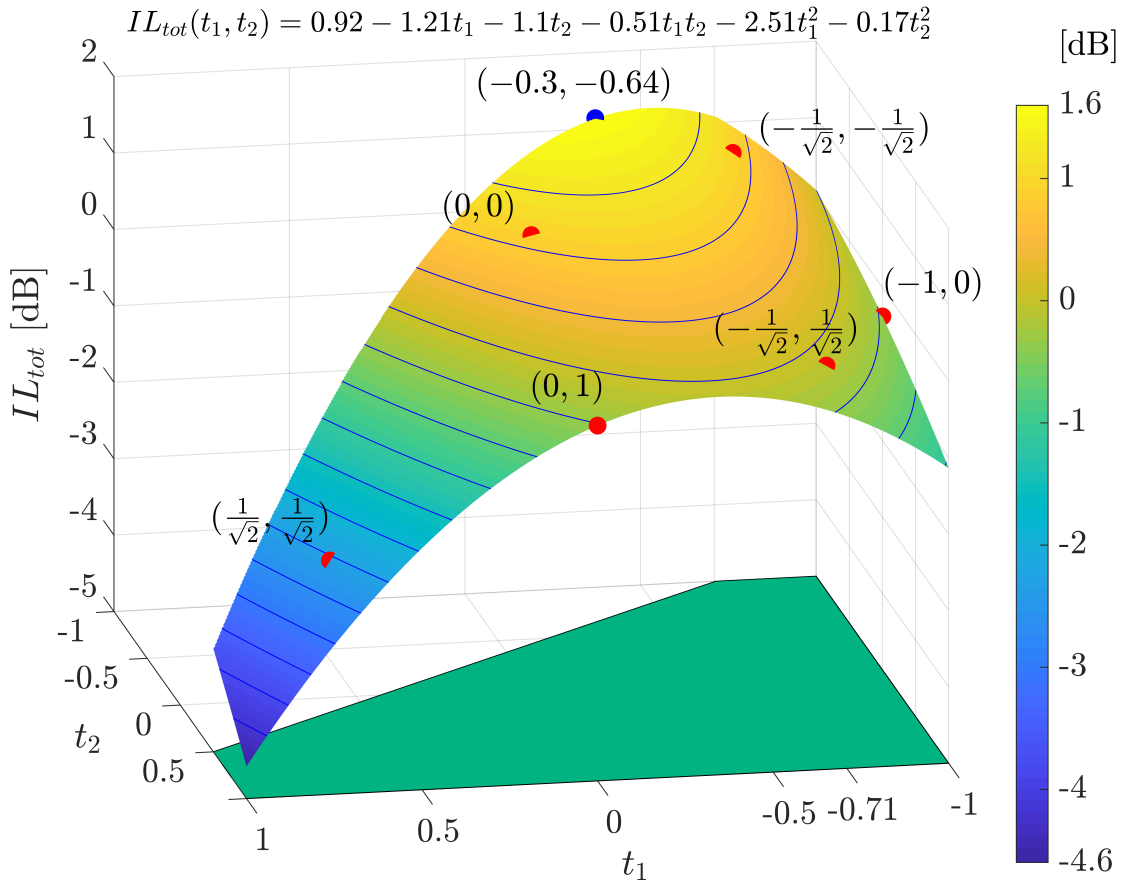
III.B. Wyniki optymalizacji

Wynik optymalizacji wyznaczania współczynników powierzchni odpowiedzi tłumika jest przedstawiony niżej na **wykresie 1**. **Dozwolony obszar powierzchni odpowiedzi**, ograniczony do pola trapezu został również na nim przedstawiony. W związku z obsługą interpretera L^AT_EX przez środowisko MATLAB, tylko do renderu wykresu grupa posłużyła się tym komercyjnym narzędziem, a „plotowanie” powierzchni wykonuje się także z poziomu pakietu Octave. (Octave, pomimo bycia darmowym pakietem, nie posiada wielu istotnych funkcji.)

Jako, że model zastępczy poddano optymalizacji, zweryfikowano adekwatność modelu zastępczego dla zmiennych wejściowych. Określono dopasowanie powierzchni odpowiedzi do punktów planu eksperymentu w przestrzeni odpowiedzi układu. **Zaznaczone na wykresie czerwone punkty planu CCI** idealnie pokryły się z interpolowaną powierzchnią odpowiedzi, ponieważ jak wcześniej wspomniano, 6 punktów daje wprost jednoznaczne rozwiązanie dwuwymiarowej powierzchni 2-go stopnia, więc różnica najmniejszych kwadratów błędu dla każdego punktu jest równa 0. Współczynniki wielomianu przedstawione są nad powierzchnią.

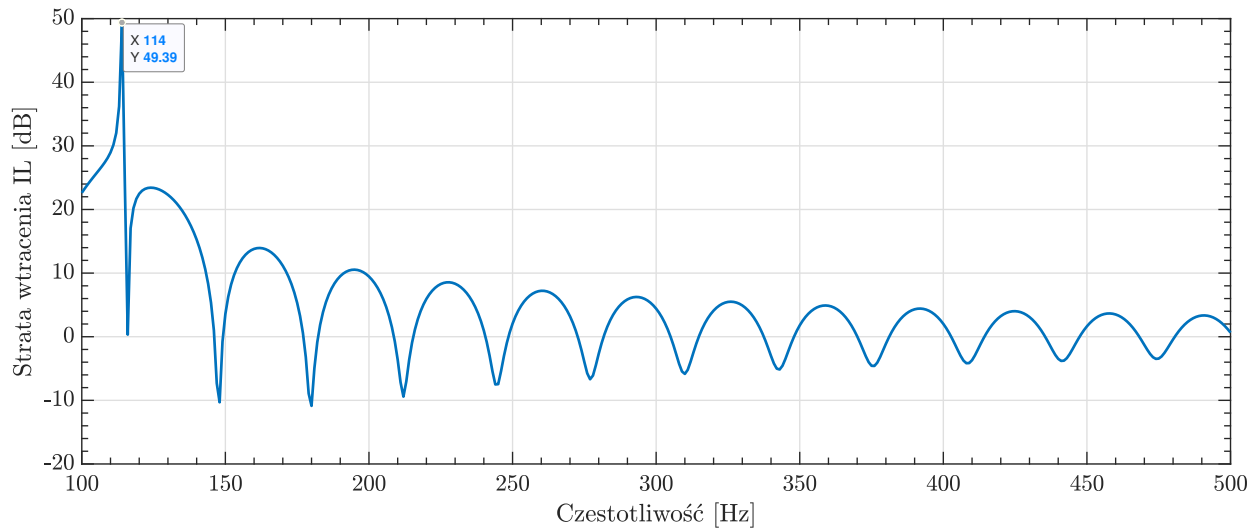
Funkcja `optimize.m` pozwoliła wyznaczyć maksymalną całkowitą wartość straty wtrącenia IL_{tot} dla rozpatrywanego obiektu. Za pomocą polecenia `max(max(ILtot))` przy zastosowaniu gęstej siatki uzyskaliśmy ekstremum funkcji z dokładnością rzędu 0,001 (brak konieczności obliczania hesjanu obrzeżonego). Grupa przyjęła, że maksimum warunkowe IL_{tot} zawierające się w przedziale powierzchni odpowiedzi **jest wystarczającą formą funkcji celu** zadaną w tym problemie.

Skrypt wykonawczy `runmuffler.sh` może także zwrócić do pliku `./output/Optimum.txt` wyniki optymalizacji, dokładne współczynniki wielomianu interpolującego oraz podać wartości x_1, x_2 w maksimum warunkowym, spełniające optymalizację tłumika.



Wykres 1.: Powierzchnia odpowiedzi dla modelu optymalizacji tłumika planem CCI 6-pkt

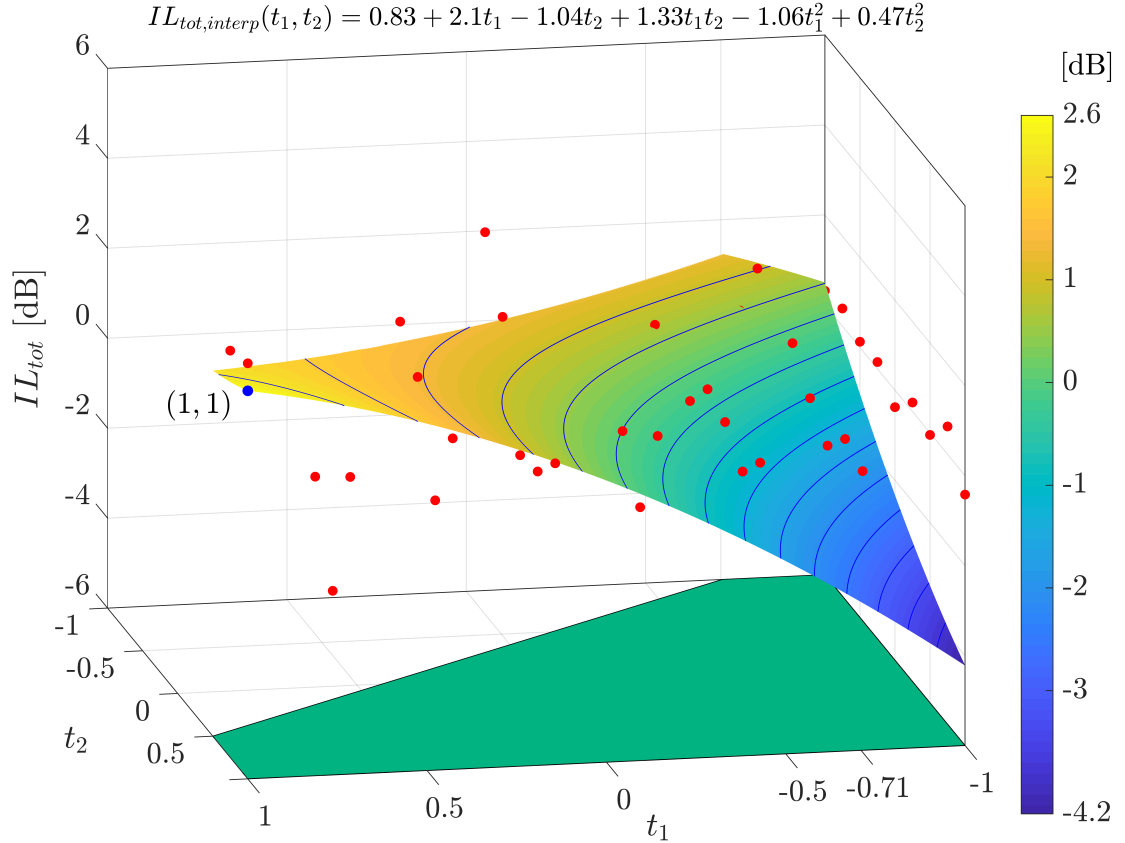
Wyznaczono też charakterystykę częstotliwościową straty wtrącenia $IL(f)$ dla zoptymalizowanego tłumika, której implementacja znajduje się w skryptach: `optimized.sh`, `optILchar.m`.



Wykres 2.: Charakterystyka straty wtrącenia $IL(f)$ dla zoptymalizowanego tłumika

III.C. Walidacja dopasowania powierzchni odpowiedzi z interpolowanymi wynikami uzyskanymi metodą omiatania dla x_1, x_2

Z metody omiatania wyznaczono 44 punkty wchodzące w dozwolony obszar fizyczny obliczonych z 17 644 par symulacji tłumik–płaski falowod (401 kroków $\Delta f \times 8 - \Delta x_1 \times 9 - \Delta x_2$). Wyniki następnie interpolowano powierzchnią 2-go stopnia z wyznaczaniem współczynników metodą najmniejszych kwadratów (błąd średniokwadratowy metody: 1,02 dB). Wynik interpolacji i współczynniki wielomianu przedstawione są poniżej na [wykresie 3](#). Maksimum warunkowe znajduje się w punkcie $t_1 = 1, t_2 = 1$.



Wykres 3.: Wykres interpolowanej powierzchni odpowiedzi dla metody omiatania

IV. Wnioski

- Maksymalną wartość całkowitej straty wtrącenia na powierzchni odpowiedzi uzyskano dla granicznego przypadku wymiarów tłumika przekraczającego dopuszczalny zakres ([wykres 1.](#) – [punkt niebieski](#)). Świadczy to o konieczności budowy tłumika z jak najmniejszą szerokością szczeliny pomiędzy wlotem a wylotem kanałów wewnątrz komory. Prawdopodobnie taki dobór wymiarów zaowocowałby również zjawiskiem dyfrakcji, które pozytywnie wpłynęłoby na skuteczność tłumienia.
- Jest to rozwiązanie gwarantujące najlepsze rezultaty, przy zachowaniu warunku granicznego $x_2 > 2x_1$. Całkowite usunięcie szczeliny upodobiłoby układ do falowodu i pozbawiło go jego właściwości. Przy spełnieniu tych założeń, maksymalna strata wtrącenia IL_{tot} wyniosłaby 1,6 dB, dla optymalnego wymiaru całkowitego komory tłumika $x_2 = 0,344$ m (po znormalizowaniu parametrów: $t_2 = -0,64$). Jest to bardzo słaby wynik, lecz należy zaznaczyć, że jest to poziom stosunku mocy akustycznych w pełnym paśmie.

- Na [wykresie 2](#). przedstawiono również charakterystykę częstotliwościową straty wtrącenia $IL(f)$ dla zoptymalizowanego tłumika po poprawce nadania szczeliny minimalnej sensownej grubości równej 1 cm. Dla prostoty wykonania realnego tłumika wartości parametrów zaokrąglono w górę do centymetrów: $x_1 = 0,17$ m, $x_2 = 0,35$ m.
- Na charakterystyce widać wyraźne okresowe wzmocnienia i osłabienia tłumienia oraz maksimum obwiedni znajdujące się w peaku tłumienia dla 114 Hz. Jest to typowy charakter dla tłumików akustycznych będących filtrami pasmowymi (dolnoprzepustowymi) i jednocześnie grzebieniowymi, ze względu na okresowy rezonans przy wystąpieniu fali stojącej. Niestety, w wyższym przedziale pasma od około 250 Hz wartość straty wtrącenia IL oscyluje stale wokół 0 dB. Tłumik ten zatem sprawdzi się do tłumienia niskich częstotliwości, szczególnie uciążliwych w podpaśmie 100 – 140 Hz.

V. Podsumowanie

- Stworzono autorskie środowisko uruchomieniowe ze składnią do obliczeń straty wtrącenia tłumików akustycznych metodą elementów skończonych wraz z zestawem skryptów pozwalających na wyznaczenie modelu zastępczego tłumika i uzyskanie najbardziej optymalnych parametrów.
Ma ono szerokie zastosowanie i może wykonać prawie każde z zadań grup (wszystkie osiowo symetryczne), pod warunkiem zmodyfikowania szablonów `ElmerGrid` ustalając: domeny siatki, współrzędne X, Y domen oraz wektor współrzędnych w funkcji zamiany współrzędnych od parametrów wejściowych: `replace.m`.
- Niestety, uzyskany model zastępczy tłumika i jego optymalizacja słabo pokrywa się z interpolowanymi wynikami uzyskanymi metodą omiatania zmiennymi x_1, x_2 .
Każde przybliżanie układu fizycznego modelem zastępczym jest podatne na błędy i uproszczenia, ponieważ nie przedstawia faktycznego rozkładu wartości funkcji celu na powierzchni odpowiedzi pomiędzy węzłami. Zatem, model zastępczy opisany dwuwymiarową powierzchnią 2-go stopnia jest tylko zgrubnym przybliżeniem złożoności obiektu opisanego równaniami różniczkowymi i warunkami brzegowymi.
- Grupa wybrała dobranie uboższego, ale jedynego możliwego fizycznie 6-punktowego planu eksperymentu CCI, ale pierwotnym pomysłem grupy było przesunięcie środka obserwacji dla planu CCI i wykonanie dzięki temu nawet planu CCC. Niestety, dobór przedziału parametrów wejściowych zadanych grupie uniemożliwiał taki zabieg.
- Całość projektu wykonano przy użyciu „wolnoźródłowych” narzędzi, ponadto, treść niniejszego sprawozdania, zgodnie z wymaganiami, została sporządzona w systemie składania tekstu $\text{\LaTeX} 2_{\epsilon}$, a rysunki w bibliotece graficznej `TikZ`.
- Obliczenia MES są bardzo „zasobożerne”, jednak starano się, by nie występowało w nich zjawisko aliasingu częstotliwości, stąd mały ich krok dyskretyzacji.
Dla przykładu, symulacja dla punktów planu CCI zajęła około 6 godzin dla komputera wyposażonego w procesor 4-rdzeniowy Intel Core i5-6300HQ 2,3 – 3,2 GHz. Każdy niezauważony błąd przed skończeniem symulacji wymagał uruchomienia ich ponownie. Stwierdzono, że nie ma sensu bardzo zagęszczać siatki dla metody omiatania, ponieważ obliczenia zajęłyby na użytym komputerze w przybliżeniu 2 tygodnie. Do takiego zadania wymagany by był klaster obliczeniowy, chociażby Cyfronetu AGH, niestety, grupa nie miała do niego dostępu.

- Oszacowano, że na projekt poświęcono około 60 godzin, co w zupełności wyczerpuje i przekracza wymagania ECTS dotyczące modułu MES w przedmiocie MiNPwA. Starano się, by na każdym poziomie projektu jego realizacja spełniała wymagania opisane w instrukcji, stąd tak długi poświęcony czas, pomimo problematycznego okresu pandemii.

VI. Bibliografia

- [1] [Konspekt do laboratorium nr 2 z MES MiNPwA](#)
- [2] [Wykład z MES dr inż. Ireneusza Czajki](#)
- [3] [R. Ming, J. Pan – Insertion loss of an acoustic enclosure](#)
- [4] [H. Kim, J. Kim, S. Lee, Y. Seo – A simple formula for insertion loss prediction of large acoustical enclosures using statistical energy analysis method](#)
- [5] [Konspekt do laboratorium nr 3 z MES MiNPwA](#)

**. Przydział zadań w grupie

- Maja Szydłowska – stworzenie planu eksperymentu, wyznaczenie przedziału dozwolonego dla parametrów wejściowych, obliczenia transformacji współrzędnych naturalnych na znormalizowane, edycja rozdziału III
- Justyna Szymańska – optymalizacja tłumika, wykonywanie równoległe obliczeń na maszynie wirtualnej Linuxa `VirtualBox`, napisanie skryptów do stworzenia planu CCI, edycja rozdziału II
- Dominika Woźniak – napisanie kodu do interpolacji wyników planu CCI powierzchnią odpowiedzi, walidacja wyników i ich analiza, zapisanie wniosków w rozdziale IV, stworzenie wykresów i rysunków
- Alexander Stefani – analiza teoretyczna, uruchamianie skryptów na komputerze z systemem Linux, implementacja głównego skryptu oraz metody omiatania, zarządzanie kodem i repozytorium, skład tekstu, autor I i V rozdziału, koordynowanie projektem.