

標準 MCAP-CR スピーカーシステムシミュレーションプログラム説明書

2013 年 5 月 1 日

鈴木 茂

0. 配布の目的

このプログラムは、バスレフ型スピーカーシステムの技術に興味のある方のために配布します。このプログラムに使われているアルゴリズムは、ばね－質点系の運動方程式モデルです。振動板の変形動作を考慮した連続体の力学モデルではありません。このため、シミュレーション計算の結果を評価するためには、物理学の知識が必要です。

物理学や技術に興味がなく、音質の評価に使用することだけを目的としてこのプログラムを使用することはご遠慮ください。

1. プログラムの使用許諾条件

このプログラムを使用する方は、以下の条件に同意したものとします。

- i. このプログラムに使われている数式モデルとアルゴリズムには、制約があり、絶対的に正しい結果を計算することが不可能であることを理解していること。
- ii. このプログラムには、未知または既知のバグが存在することを理解していること。
- iii. このプログラムを使用することによる、いかなる、損失やその他の問題について、苦情を申し立てたり損害賠償を請求しないこと。
- iv. このプログラムを商用目的に使用しないこと。
- v. このプログラムの著作権者の許可無く、このプログラムそのもの、または、改変したコードを配布しないこと。
- vi. このプログラムがプラットフォームに依存しないことを理解し、Microsoft Visual Studio など特定のプラットフォームでしか使用できないコードに書き換えて使用しないこと。
- vii. スピーカーシステムの音質の評価には使用しないこと。

2. このプログラムの機能

- このプログラムは、MCAP-CR 型構造、ダブルバスレフ型、(また、その応用として、シングルバスレフ型、密閉型、バッフル型)スピーカーシステムの質点(振動板及びバスレフダクト内の空気塊)の過渡変位を計算し、結果を ASCII ファイルに出力します。
- 計算したシミュレーション結果を、離散フーリエ変換し、その結果を ASCII ファイルに出力します。
- 電流や電圧ではなく、力を入力とし、その信号形態として、正弦波、リニアスイープ、ランダムを選択することができます。また、リニアスイープの発展とそと、正弦波入力の周波数を 1Hz ずつ逐次計算し、これらの結果を ASCII ファイルに出力することができます。

この説明書に記載されていない内容について、知る必要がある場合には、添付のソースコード“`mcapcr-cui-dft_release01.0.cpp`”を参照して下さい。

注: ユーザによるプログラムの改変

ユーザは、第1章に記載された使用許諾条件の範囲内で、ソースコードを自由に改変して使用することができます。

3. このプログラムの標準的な使用方法

このプログラムは、ユーザー自身で、コンパイルし、実行ファイルを作成して使用します。Windows の x86 系 CPU の 32 ビット実行ファイルが付属していますので、その実行ファイルを使用することもできます。付属の実行ファイルは、64 ビットの Windows 7 でも使用出来ることを確認しています。

プログラムコードは、C 言語(一部に C++ の用法を含みます)で記述されています。このため、コード自体は、C コンパイラが使用可能な環境であれば、OS に依存すること無く使用することが可能です。コンパイラによっては、文法の扱いが異なることがありますが、その場合には、記述方法を修正してください。

このプログラムコードは、GCC (Linux) および MinGW (Windows) コンパイラでコンパイルできることが確認されています。

Mac OS で使用する場合には、GCC などの適切なコンパイラを使用して下さい。

Microsoft Visual C++ を使用すると、標準的な、コンパイラで使用できないコードに改変することが必要となります。このため、Visual C++は、お勧めできません。

Free BSD, UNIX または VMS 等の環境でも、このプログラムをコンパイルして使用できるはずですが、テストはしていません。

コンパイラは以下で入手できます。

GNU Compiler Collection	http://gcc.gnu.org/
MinGW Projects	http://www.mingw.org/

3.1 コンパイルの準備

Linux

GNU C コンパイラ (GCC)が、大抵の場合、標準でインストールされています。RPM 系のディストリビューションの場合は、以下のコマンドで、インストールされているかどうか確認できます。

```
$ rpm -qa|grep gcc
```

Windows

プログラムコードを改変する場合には、ユーザ自身で MinGW インストールする必要があります。コンパイラを使用したくない場合には、付属の実行ファイルを使用して下さい。

3.2 ソースコードのコンパイル

Linux

- すべてのディストリビューションに共通の内容

配布された圧縮ファイルを解凍し、作業フォルダにコピーします。

X Term などのターミナルウィンドウを起動し、作業フォルダに、移動します。

GCC のコマンドを利用し、ソースファイルをコンパイルします。

- 例 -

Vine Linux 6.0 の場合 (大抵のディストリビューションで同じコマンドを使用します)

```
$ g++ mcapcr-cui-dft_release01.0.cpp
```

この結果“a.out”という実行ファイルが生成されます。別な名前にしたい場合には、OS の制約内で、適当なファイル名に変更してください。

Windows

圧縮ファイルを解凍し、作業フォルダにコピーします。

コマンドプロンプトを起動し、作業フォルダに移動します。

MinGW のコマンド

```
g++ mcapcr-cui-dft_release01.0.cpp
```

この結果、“a.exe”という実行ファイルが生成されます。

3.3 プログラムの実行 (Linux/Windows)

3.3.1 シミュレーション条件設定ファイルの作成

作業フォルダに、“parameters.txt”という名称のテキストファイルを作成します。このファイルは、以下のフォー

マツで作成して下さい。以前のバージョンのシミュレーションソフトウェアとは、フォーマットが違っているので注意して下さい。

parameters.txt のフォーマット

注:

入力値の前には、いかなる文字、タブ、改行等も置くことができません。

改行コードは、コンパイラが認識できるものを使用して下さい。Windows の場合には、メモ帳を使用して作成することができます。

入力値の仕切りには、タブを使用します。コンマやセミコロン等タブ以外の区切り文字は使用出来ません。

入力には、ASCII コードのみが使用できます。

7 行目以降には、好きな内容を書き込むことができます。

例:

```
7.0 5.47 45
3 32 1 0.1
15 10 14 16
21.16 21.16 21.16 15.21 15.21 15.21
50 92 110 120 150 240
0 0 0 0 0 0 0
// -- Below this line does not affect calculation.
// 1st row: m0[g], radius of membrane[cm], f0[Hz]
// 2nd row: number of chambers(n), division, thermal condition, amplitude of force
// 3rd row: V[0], ..., V[2n] in litre
// 4th row: A[1], ..., A[2n] in sq-cm
// 5th row: L[1], ..., L[2n] in mm
// 6th row: C[0], ..., C[2n] in kg/s
```

1 行目	m0 (単位:グラム), 振動板の有効半径 (単位:cm), f0 (単位:Hz)
2 行目	副空気室の数 (1 以上 12 以下の整数), 1 周期の分割数 (自然数), 熱力学的条件 (1.0:等温, 1.4:断熱), 入力の振幅 (単位:N)
3 行目	各空気室の容積(単位:リットル)
4 行目	各ダクトの断面積(単位:平方センチメートル)
5 行目	各ダクトの実効長(単位:ミリメートル)
6 行目	振動板とダクトの減衰係数(単位:kg/s)

アルゴリズムは、以下の文書を参照して下さい。ただし、その後の改変がありますので、必要な場合には、ソースコードを参照して下さい。

http://mcap.web.fc2.com/documents/MCAP008J_S-MCAP-CR_simulation_r0.pdf

注:

分割数は、32(5ビット), 64(6ビット),または 128(7ビット)が推奨値です。通常は 32 で十分ですが、シミュレーション結果が大きく振動している場合には、64 以上に変更してください。この数字を大きくすると計算が遅くなります。

3.3.2 プログラムの実行

ソースコードをコンパイルすると、その環境で実行可能なファイルが生成されます。

実行ファイルを作業フォルダにコピーし、コマンドラインで、実行ファイル名をタイプしてプログラムを実行して下さい。

Linux (実行ファイル名が“a.out”の場合)

\$./a.out↵

Windows (実行ファイル名が“a.exe”の場合)

a↵

解析オプション

解析にあたっては、以下のオプションがあります。

(1) vector sum option

フーリエ解析を実行する場合に、内部ダクトの効果を含めるか、無視するかを選択肢ます。0を選択した場合には、振動板と全てのダクトの効果を加算します。1を選択すると、内部ダクトの効果は加算されません。通常は、0を選択して下さい。

```
[0] values and parameters caluculation done!  
Choose vector sum option  
      0: All ducts, 1: ignore internal ducts  
Your choice =
```

(2) Processing Option

“Sequential Processing(10Hz から400Hz までを1Hz 刻みで計算する):0”または “One Time Processing(単一周波数での正弦波、ニアスweep、またはランダム)”を選択肢ます。

```
Choose Sequential Processing or One time Processing  
      Sequential Processing:  0  
      One Time Processing:    1  
You choose :
```

0: Sequential:

このオプションを選択すると下記の計算を実行します。

1. 最初に 10Hz での応答を計算し、その周波数において、離散フーリエ変換します。すなわち、
$$f_j = \sum_{k=0}^{n-1} x_k e^{\frac{-2\pi i}{n} jk}$$
 (10Hz においての離散フーリエ係数)を計算し、その結果を “dft_sequential_calculation_result.csv” に書き込みます。
2. 次に 1Hz を加算した、11Hz において上記の計算を実行し、結果を同じファイルに追記します。
3. これを 400Hz になるまで、繰り返し実行します。

1: One Time Processing

このオプションを選択すると以下の処理を実行します。

(3) Input Signal Option

正弦波、リニアスweep、ランダムのいずれかの入力信号を選択肢ます。

0: 正弦波 (周波数はこの後に指定する)

1: リニアスweep (範囲はこの後に指定する)

2: ランダム入力

0: Sinusoidal

解析したい周波数(自然数)を入力し、エンターキーを押します。

1:Linear Sweep

スweepする最高周波数(Hz)とスweep速度(Hz/s)を指定します。周波数の最大値は 200Hz で、それより大きな数字を入力すると、200 にセットされます。

2:Random

このオプションを選択すると、サンプリング周波数は、100Hz に時間解像度(分割数)を掛けた値にセットされます。

ひつような場合は、ソースコードを改変して使用して下さい。

3.3.3 計算結果の確認

出力ファイル

計算の結果、出力されるファイルの一覧を Table-1 に示します。

Table-1 出力ファイル一覧

ファイル名	詳細	備考
dft_sequential_calculation_result.csv	周波数[Hz], SPL[dB] (動圧[Pa]をそのまま変換したもの), 各周波数での位相[度], DFT 結果の実部, 同じく虚部	“Sequential Processing”オプションを選択した場合の出力
dft_result.csv	周波数[Hz], SPL[dB] (動圧[Pa]をそのまま変換したもの), 各周波数での位相[度], DFT 結果の実部, 同じく虚部	“One Time Processing”オプションを選択した場合の出力
normalized_x_vectors.csv	時間[s], 変位の荷重和[mm], 振動板の変位[mm], ダクト内の空気塊の面積で重み付けされた変位[mm],...	
normalized_v_vectors.csv	時間[s], 速度の荷重和[m/s], 振動板の速度[mm/s], ダクト内の空気塊の面積で重み付けされた速度[m/s],...	
matrices.csv	Standard stiffness matrix, actual stiffness matrix and intermediate matrix are stored.	Used for diagnostics.
dftv.txt	1 st line: number of datapoints 2 nd line: time resolution 3 rd line and below: normalized sum of velocities	Used for Fourier Transform

規格化(重み付け)の定義

$$x_j^* = r_j x_j = \frac{a_j}{a_0} x_j \quad \text{変位}$$

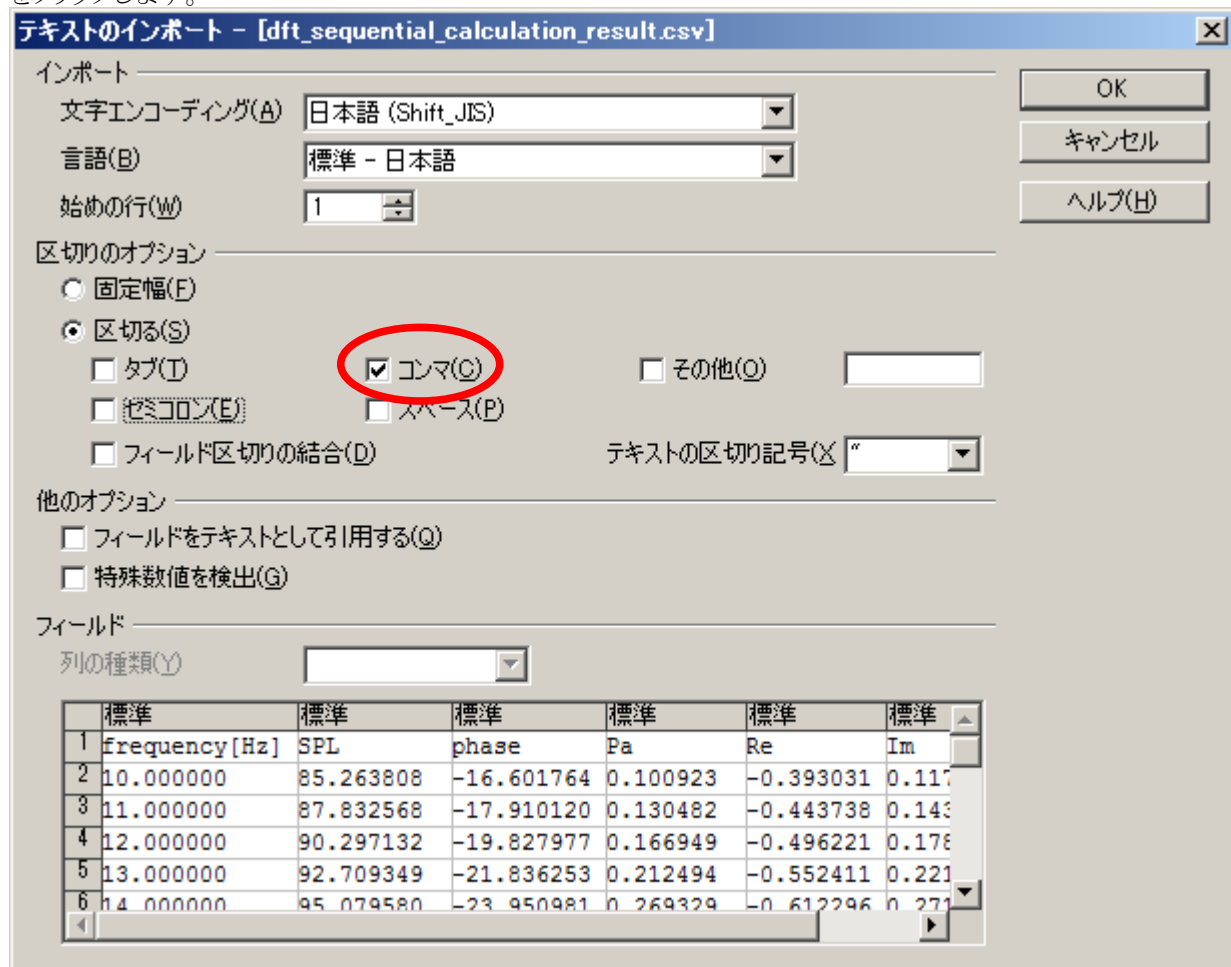
$$v_j^* = r_j v_j = \frac{a_j}{a_0} v_j \quad \text{速度}$$

データのプロット

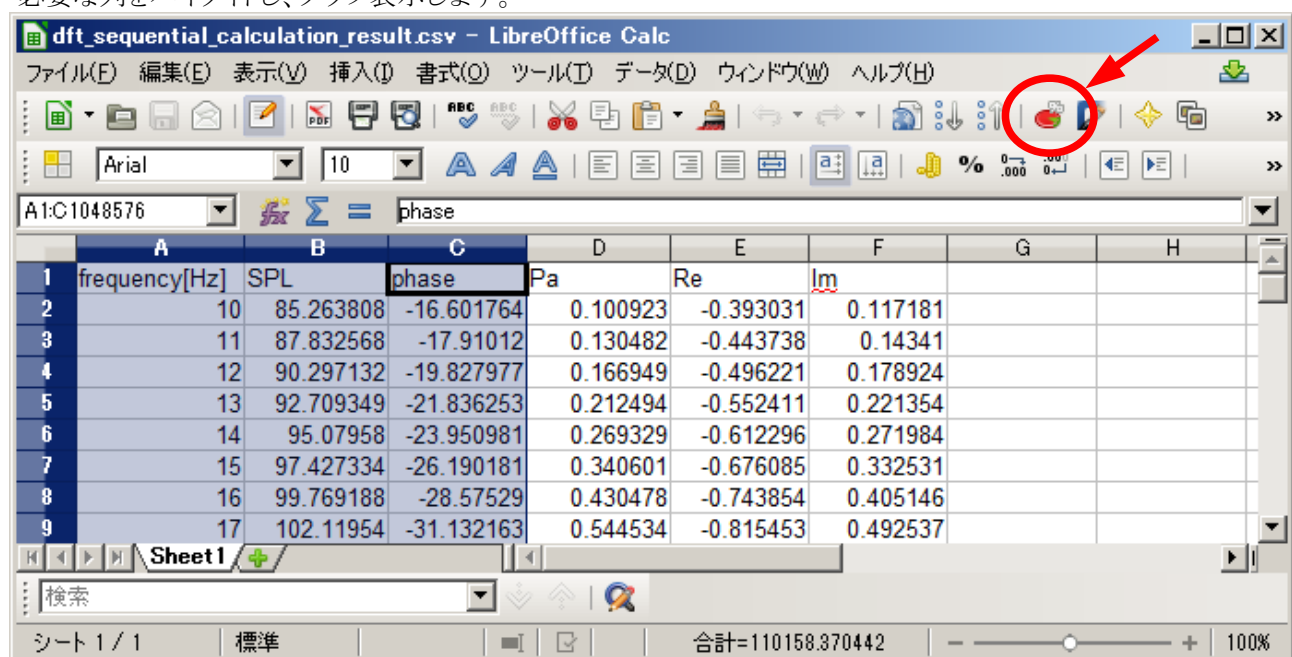
このプログラムには、結果をプロットする機能はありませんので、LibreOffice Calc 等を使用してプロットしてください。

LibreOffice Calc を使用した例

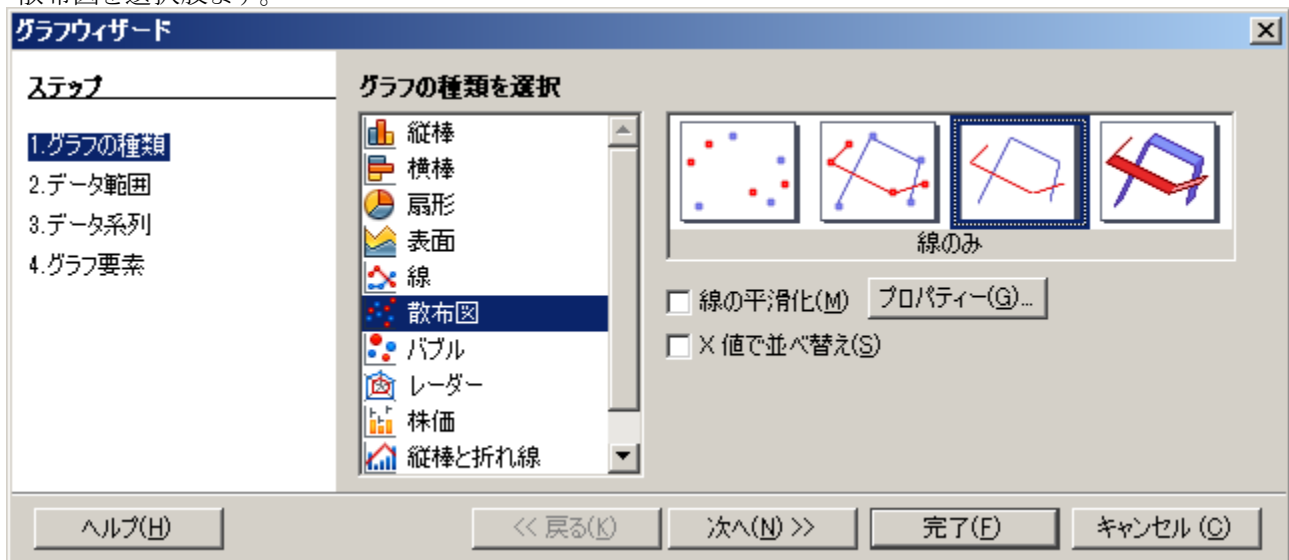
Windows エクスプローラ等で作業フォルダを開き“normalized_v_vectors.csv”をダブルクリックします。Calc が起動(別なソフトを関連付けしていない場合)します。ここで、区切りオプションに、コンマを選択し、OK をクリックします。



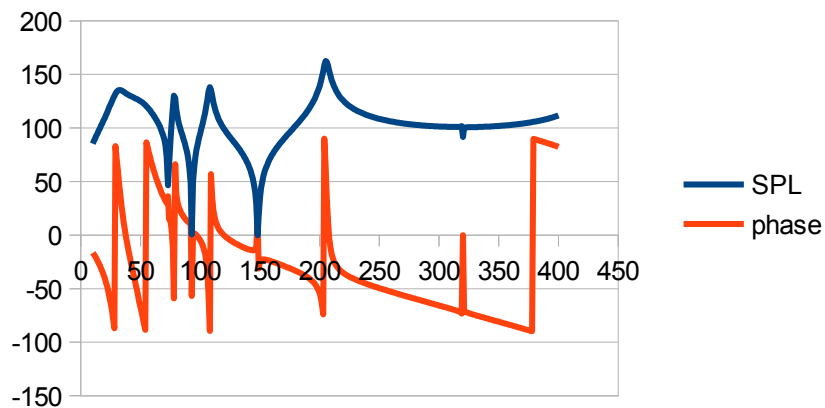
必要な列をハイライトし、グラフ表示します。



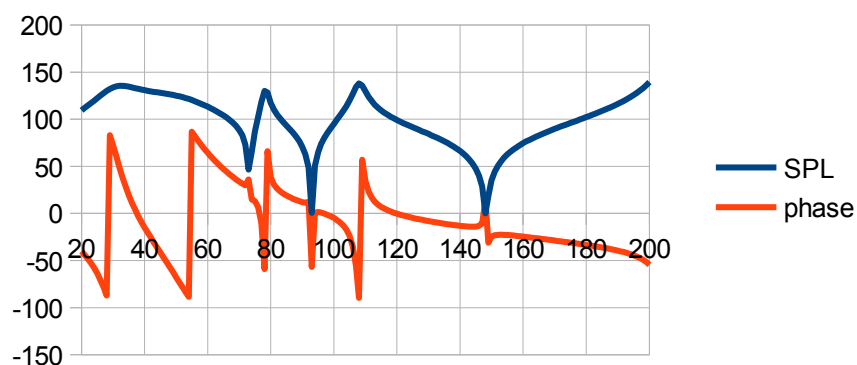
散布図を選択肢ます。



以下のようなプロットが表示されます。



必要に応じ、範囲などを変更します。



このプログラムは、共振周波数を発見するのには有効ですが、周波数応答を実用的に計算することは出来ません。

LibreOffice の使い方はソフトウェアのヘルプ等を参照して下さい。慣れれば、高価な商用ソフトよりも使いやすくなります。

バグを発見した場合には、下記までご一報ください(義務ではありません)。

mcapspeakers@gmail.com