

(オシロスコープ100MHzタイプを例にした)

## 操作手順書 (簡易版) Type2

音圧測定解析システム「超音波テスターNA」



### <オシロスコープ>

USBオシロスコープ **P i c o s c o p e 2 2 0 7 A**

分解能 8 b i t   チャンネル数   2 c h

- ・帯域幅(-3dB) : 100MHz (100MHzタイプ)
- ・最大サンプリングレート : 1G
- ・バッファメモリ : 40k サンプル

USBオシロスコープ **P i c o s c o p e 2 2 0 4 A**

分解能 8 b i t   チャンネル数   2 c h

- ・帯域幅(-3dB) : 10MHz (超音波テスター標準タイプ)
- ・バッファメモリ : 8k サンプル

説明書は、必要な時に再読して下さい。

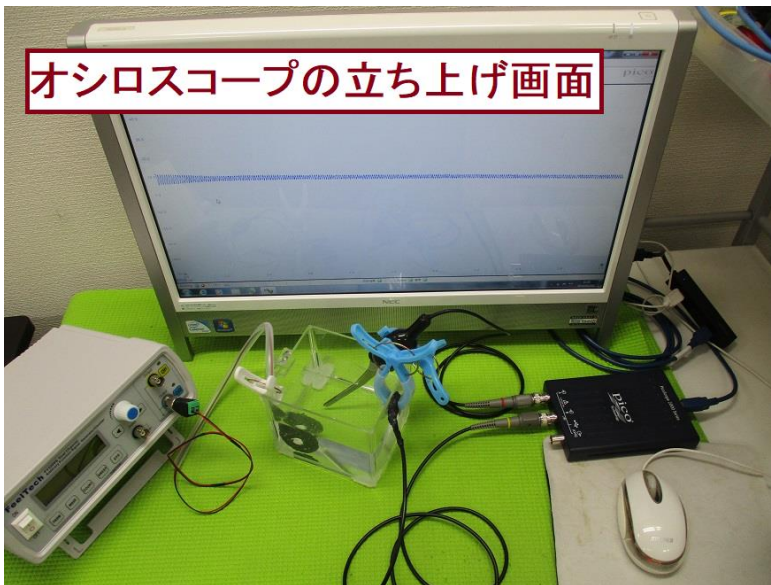
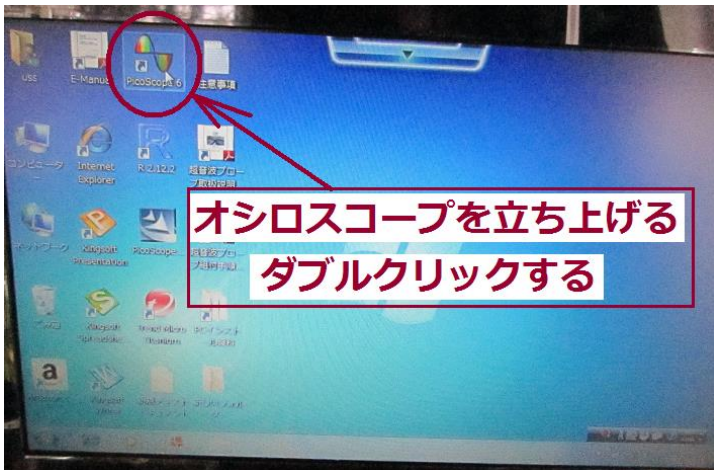
## 超音波システム研究所

Ver 2.0

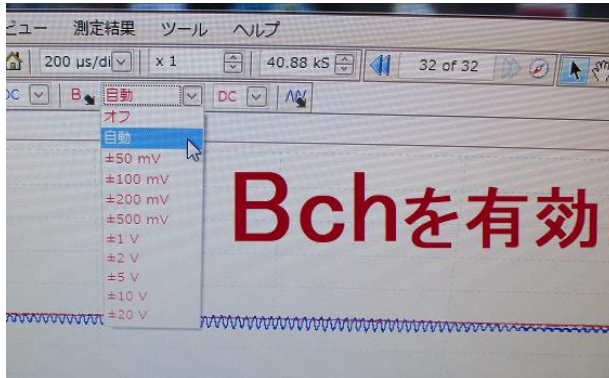
## 1 : 接続



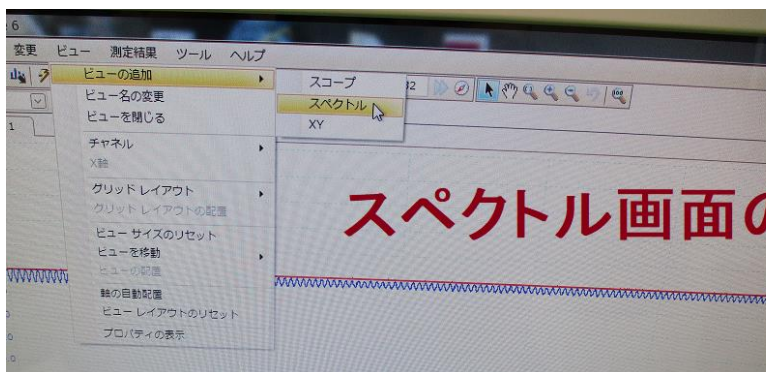
## 2 : オシロスコープを立ち上げる



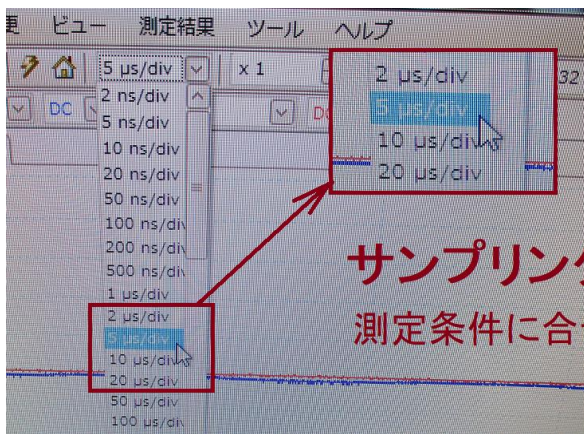
### 3 : 設定



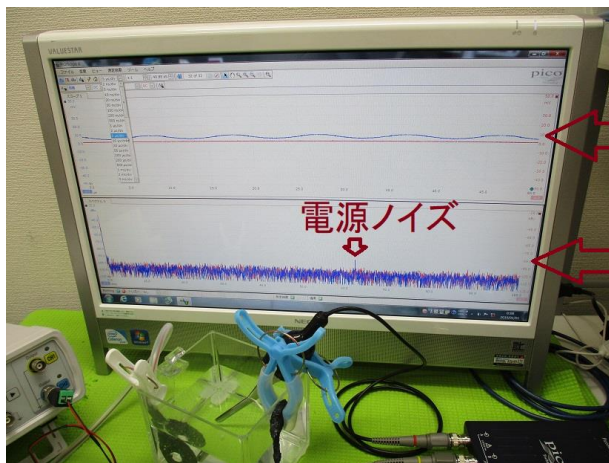
Bchを有効にする



スペクトル画面の追加



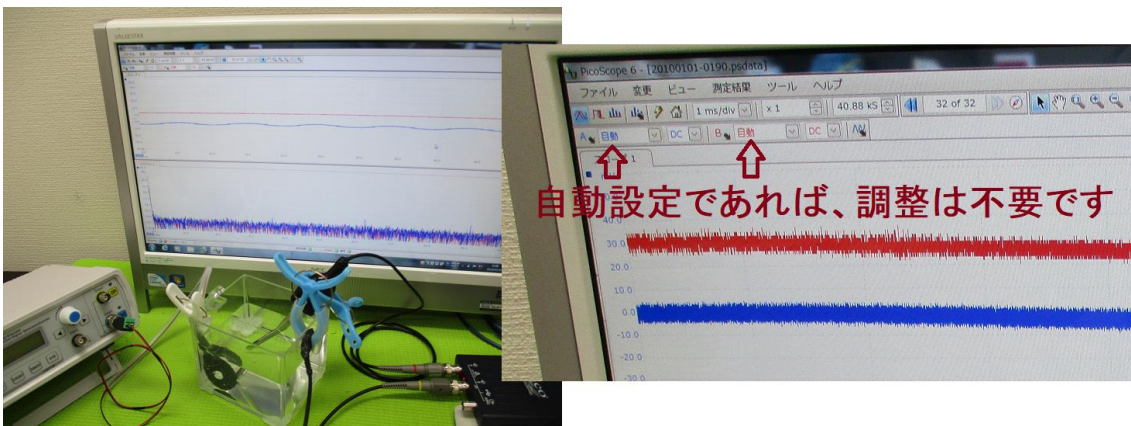
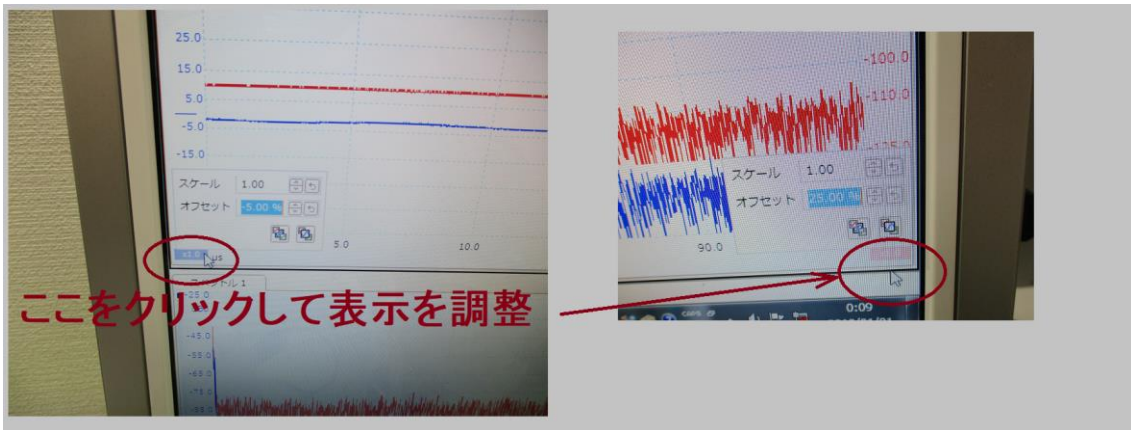
サンプリング時間の設定  
測定条件に合わせて設定



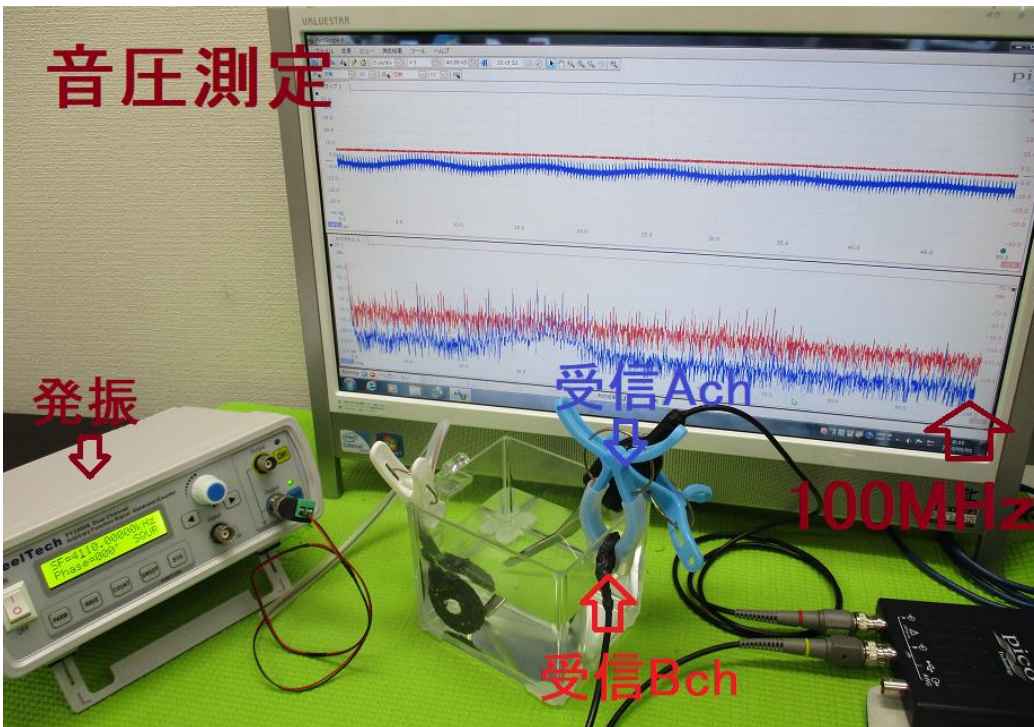
時間-電圧(音圧)グラフ

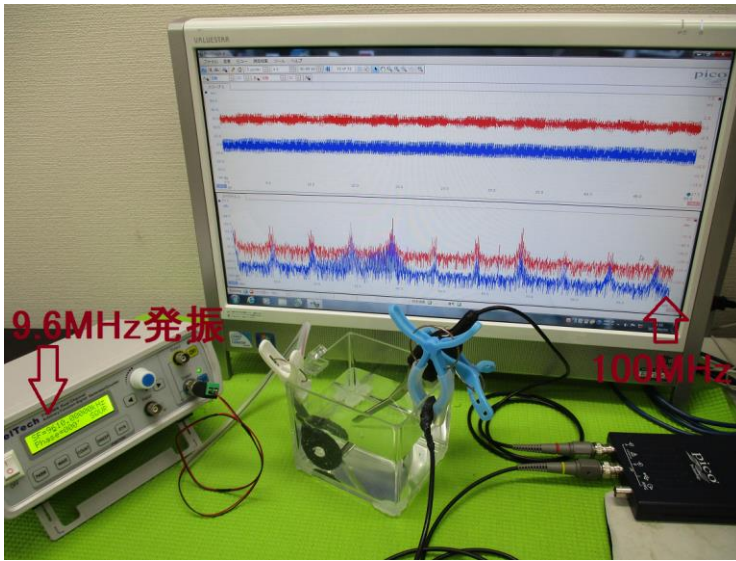
追加された  
周波数-エネルギー(dBu)  
グラフ

電源ノイズ



#### 4 : 測定





☰ YouTube JP 検索 **データの保存**

超音波の音圧測定解析システム（オシロスコープ100MHzタイプ） ultrasonic-labo

<https://youtu.be/0cQa9RcWQnc>

<https://youtu.be/EyJY5tHPq1s>

ファイル>全波形の保存

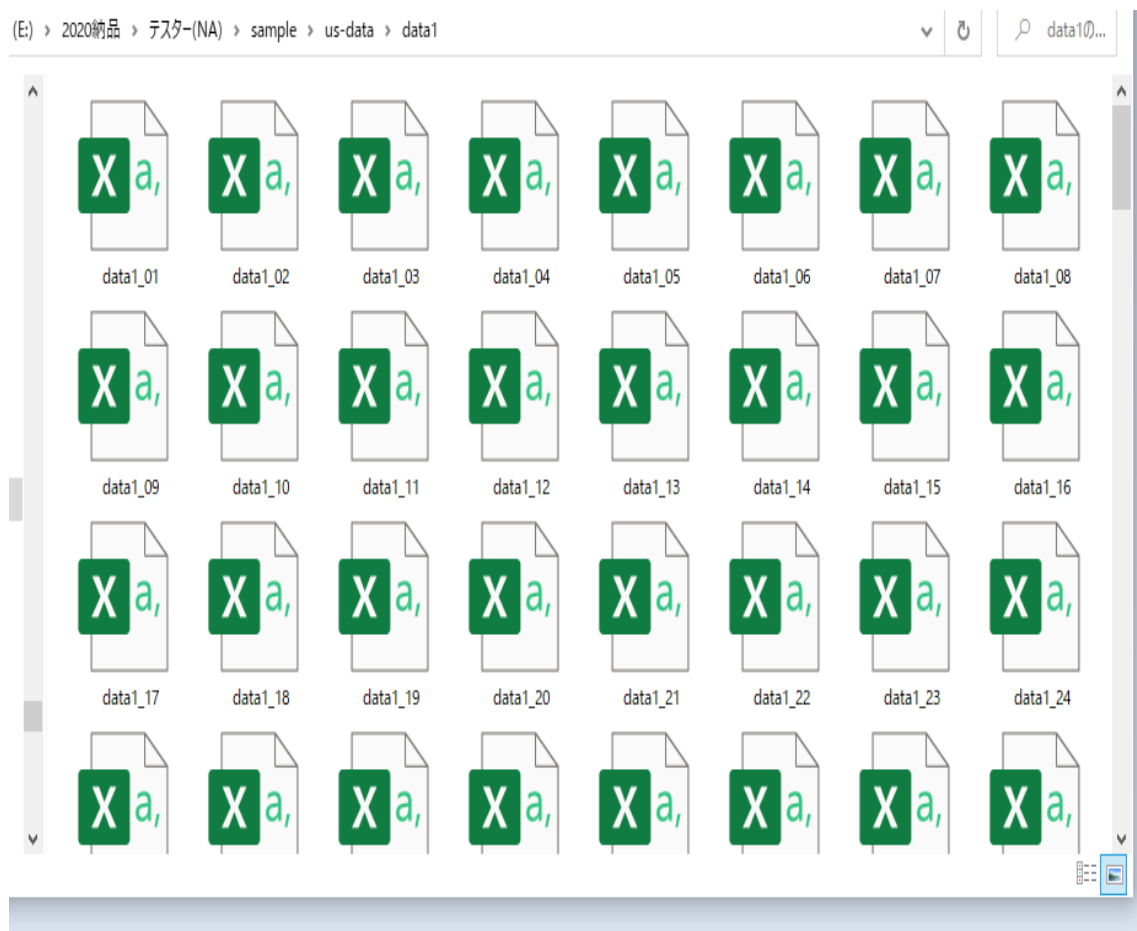
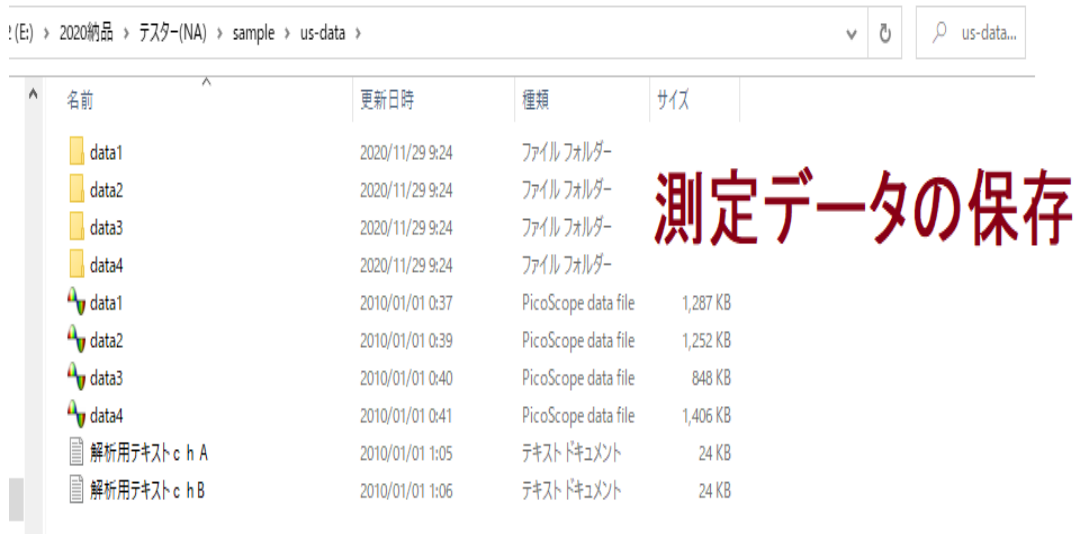
PicoScope data file (.psdata)

ファイル>全波形に名前をつけて保存（解析用）

Microsoft Excel CSV ファイル (.csv)

保存データは、最大32画面のデータを含みます

Excel CSV ファイルは、ホルダーに最大32個作成されます



## 参照

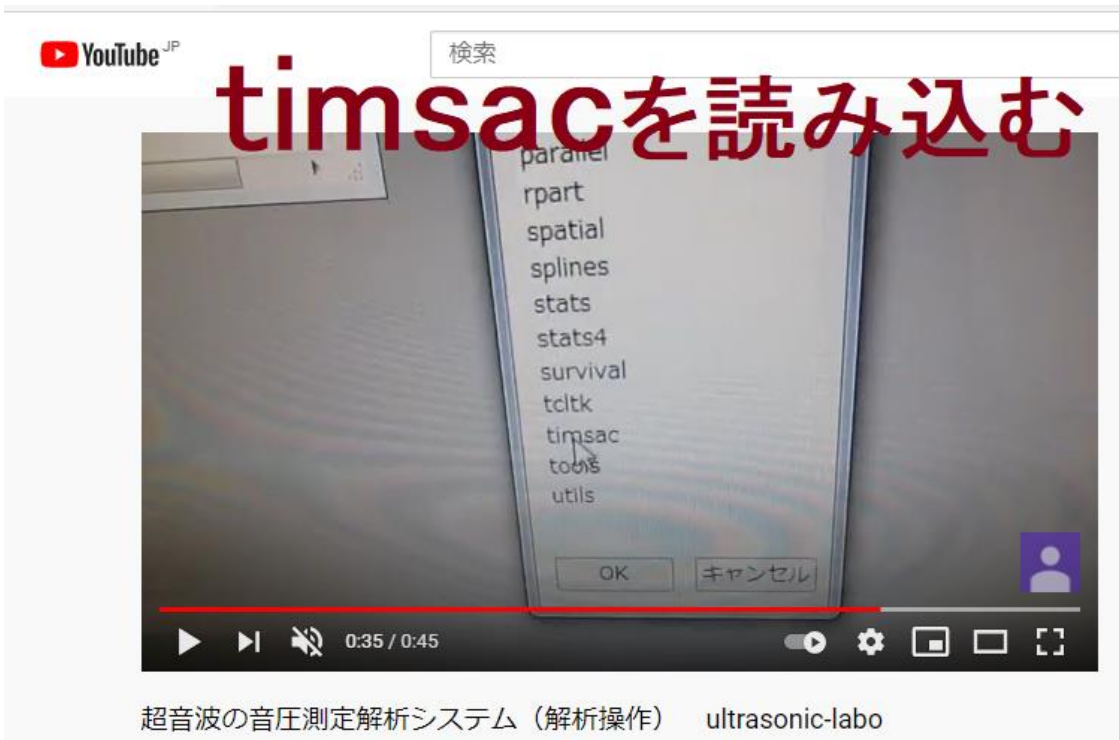
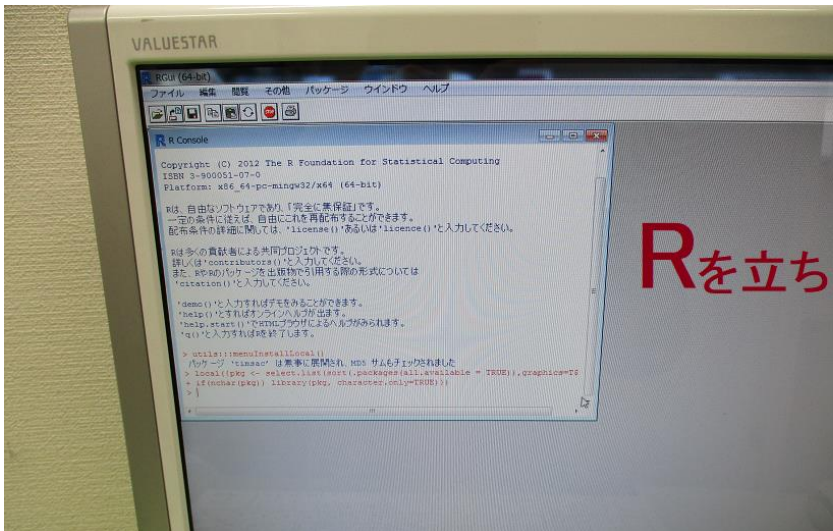
超音波の音圧測定解析システム (オシロスコープ100MHzタイプ)

<http://ultrasonic-labo.com/?p=17972>

## 5 : 解析

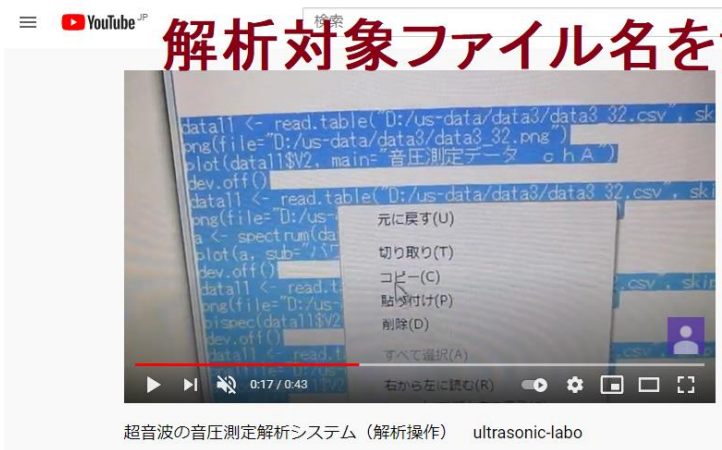
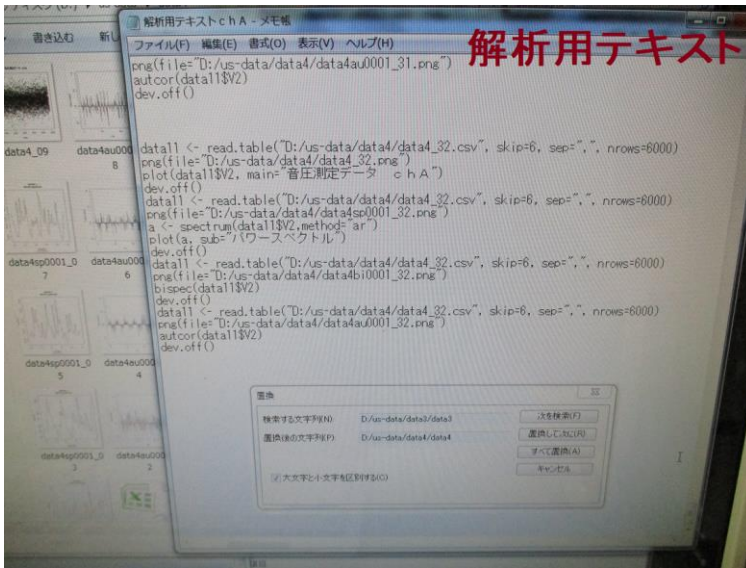


ダブルクリックして立ち上げる



[https://youtu.be/2RcXz\\_xtNu4](https://youtu.be/2RcXz_xtNu4)

## 解析用テキストファイル（解析用テキスト c h A . t x t）を開く



コピー  
ペースト

<https://youtu.be/uo2PCRF2xIo>

<https://youtu.be/GR7NLxUfU4A>





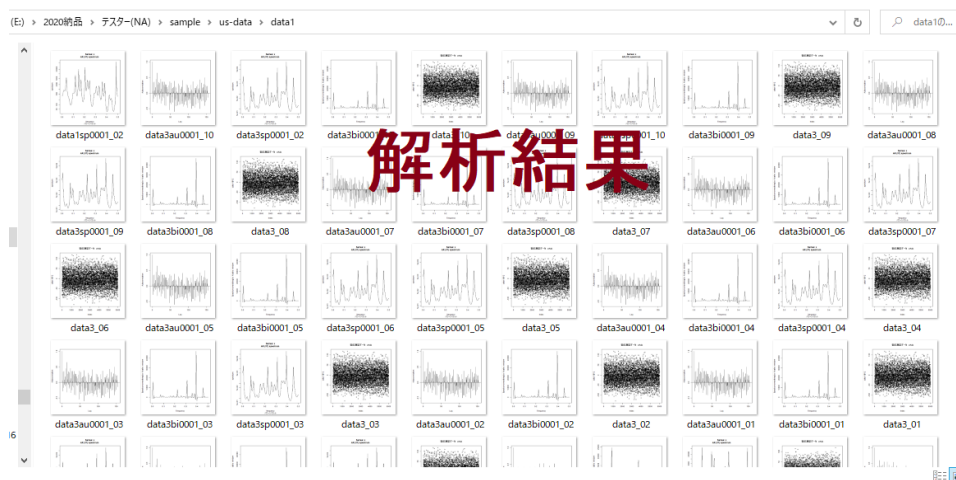
## 解析用テキスト c h A (テキスト ドキュメント (.txt))

### ファイル名「D:/us-data/data2/data2」の場合

```
data11 <- read.table("D:/us-data/data2/data2_01.csv", skip=6, sep=",", nrows=6000)
png(file="D:/us-data/data2/data2_01.png")
plot(data11$V2, main="音圧測定データ c h A")
dev.off()
data11 <- read.table("D:/us-data/data2/data2_01.csv", skip=6, sep=",", nrows=6000)
png(file="D:/us-data/data2/data2sp0001_01.png")
a <- spectrum(data11$V2, method="ar")
plot(a, sub="パワースペクトル")
dev.off()
data11 <- read.table("D:/us-data/data2/data2_01.csv", skip=6, sep=",", nrows=6000)
png(file="D:/us-data/data2/data2bi0001_01.png")
bispec(data11$V2)
dev.off()
data11 <- read.table("D:/us-data/data2/data2_01.csv", skip=6, sep=",", nrows=6000)
png(file="D:/us-data/data2/data2au0001_01.png")
autcor(data11$V2)
dev.off()
```

```
data11 <- read.table("D:/us-data/data2/data2_02.csv", skip=6, sep=",", nrows=6000)
png(file="D:/us-data/data2/data2_02.png")
plot(data11$V2, main="音圧測定データ c h A")
dev.off()
data11 <- read.table("D:/us-data/data2/data2_02.csv", skip=6, sep=",", nrows=6000)
png(file="D:/us-data/data2/data2sp0001_02.png")
a <- spectrum(data11$V2, method="ar")
plot(a, sub="パワースペクトル")
dev.off()0:49 2010/01/01
data11 <- read.table("D:/us-data/data2/data2_02.csv", skip=6, sep=",", nrows=6000)
png(file="D:/us-data/data2/data2bi0001_02.png")
bispec(data11$V2)
dev.off()
data11 <- read.table("D:/us-data/data2/data2_02.csv", skip=6, sep=",", nrows=6000)
png(file="D:/us-data/data2/data2au0001_02.png")
autcor(data11$V2)
dev.off()
```

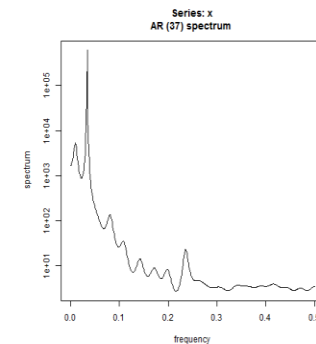
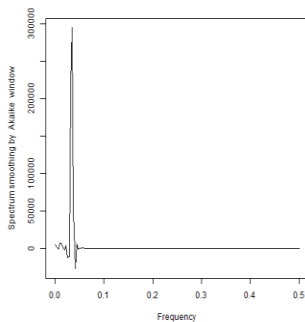
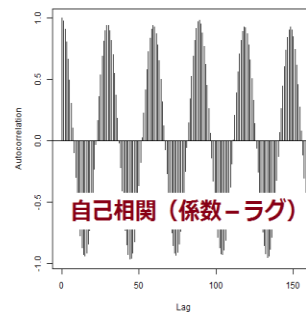
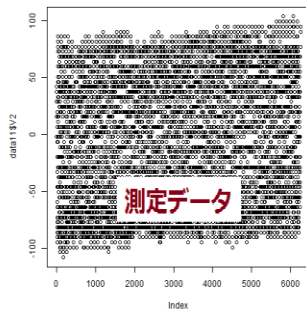
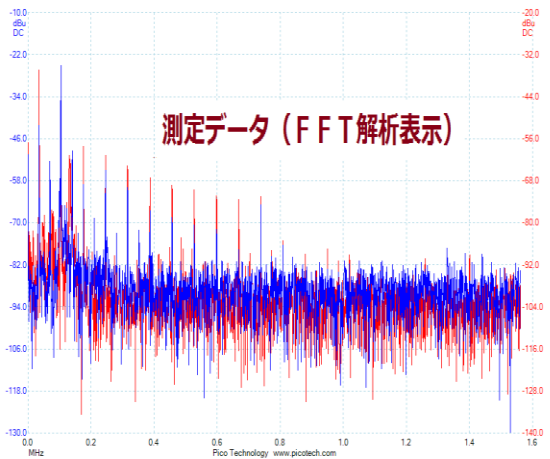
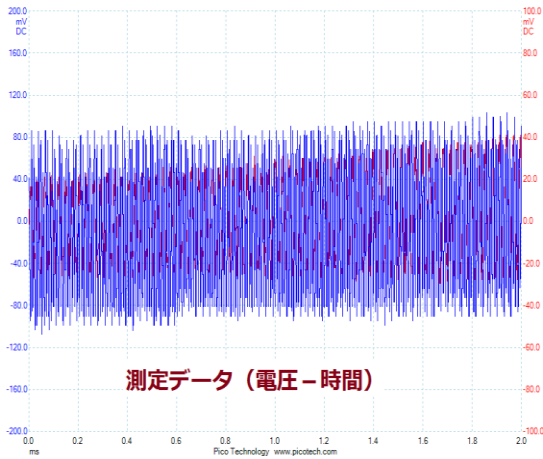
• • •



## 結果

測定データに対して、

以下の解析（自己相関、バイスペクトル、パワースペクトル）により  
超音波の伝搬状態を評価（超音波システム研究所 オリジナル技術）します



<https://youtu.be/q0C58L-yiWg>

<https://youtu.be/1hLsBHtg0tg>

注:解析には下記ツールを利用します

注:OML(Open Market License)

<https://www.ism.ac.jp/ismlib/jpn/ismlib/license.html>

注:TIMSAC(TIME Series Analysis and Control program)

<https://jasp.ism.ac.jp/ism/timsac/>

[Japanese/English]

## TIMSAC for R package

統計数理研究所

2020年07月

### 1. はじめに

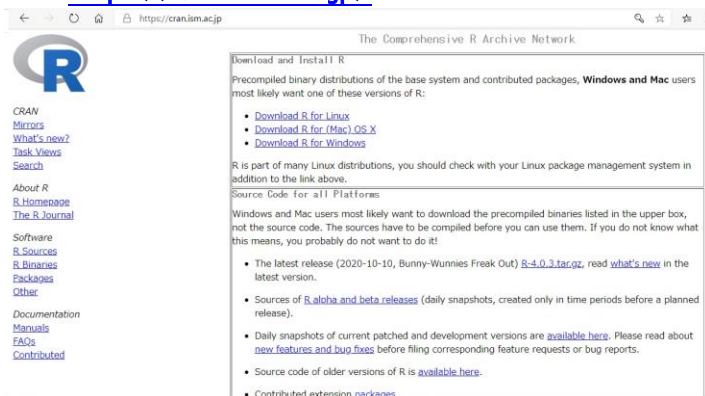
TIMSAC(TIME Series Analysis and Control program)は、統計数理研究所で開発された時系列データの解析、予測、制御のための総合的プログラム/パッケージである。オリジナルTIMSAC(TIMASAC-72)は1972年に発表され、その後、TIMSACシリーズとしてTIMSAC-74、TIMSAC-78、TIMSAC-84が Computer Science Monograph に発表された。工業プロセスの最適制御、経済変動の分析等広い分野で実際に利用されている。TIMSACの特徴としては、情報量規準の考え方をを用いた時系列解析プログラムであることが挙げられる。TIMSAC-72ではFPE(Final Prediction Error)、TIMSAC-74以降ではAIC(Akaike Information Criterion)、TIMSAC-78のベイズ型モデルではABIC(Akaike Bayesian Information Criterion)も用いられている。

一方、Rはフリーな統計処理言語かつ環境である。Rは配列演算を基本に設計されているため多次元配列の扱いに適している上にグラフィック関数も豊富であり、かつFORTRANやCのサブルーチンを簡単に呼び出せるインタフェースを備えている。そこで、FORTRANで書かれているオリジナルプログラムの計算処理機能のみをライブラリ化し、R関数を通して入出力を行い、必要であればその解析結果等をRでグラフィック表示することによりデータ解析を容易にした。

なお、バージョン 1.2.8のパッケージに含まれていた七つの関数 `armaimp()`、`lsar2()`、`ngsmth()`、`tsmooth()`、`tvvar()`、`tvvar()`、`tvspc()` は、バージョン 1.3.0では削除した。これらの関数は「FORTRAN77 時系列解析プログラミング」(北川源四郎著/岩波書店)のプログラムをソースとした関数であり、同書を基にした他の関数とともにRパッケージ `ISSS` として公開している。

注:「R」フリーな統計処理言語かつ環境

<https://cran.ism.ac.jp/>



## 参考: バイスペクトル

バイスペクトルは

以下のように

周波数  $f_1$ 、 $f_2$ 、 $f_1 + f_2$  のスペクトルの積で表すことができる。

$$B(f_1, f_2) = X(f_1)Y(f_2)Z(f_1 + f_2)$$

主要周波数が  $f_1$  であるとき、

$f_1 + f_1 = f_2$ 、 $f_1 + f_2 = f_3$  で表される

$f_2$ 、 $f_3$  という周波数成分が存在すれば バイスペクトルは値をもつ。

これは主要周波数  $f_1$  の

整数倍の周波数成分を持つことと同等であるので、

バイスペクトルを評価することにより、高調波の存在を評価できる。

# 理論的背景

## 超音波の検出方法

### 1：超音波の基礎

#### やさしい超音波工学

— 広がる新応用の開拓 (ケイブックス)

川端 昭 (著), 高橋 貞行 (著), 一ノ瀬 昇 (著)

出版社: 工業調査会; 増補版 (1998/01)

### 2：非線形性の解析

叩いて超音波で見る—非線形効果を利用した計測

佐藤 拓宋 (著) 出版社: コロナ社 (1995/06)

#### ダイナミックシステムの統計的解析と制御

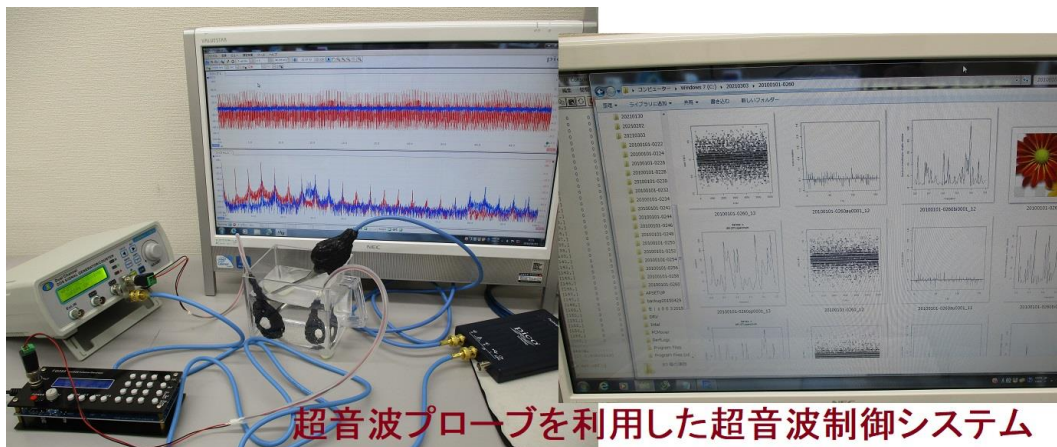
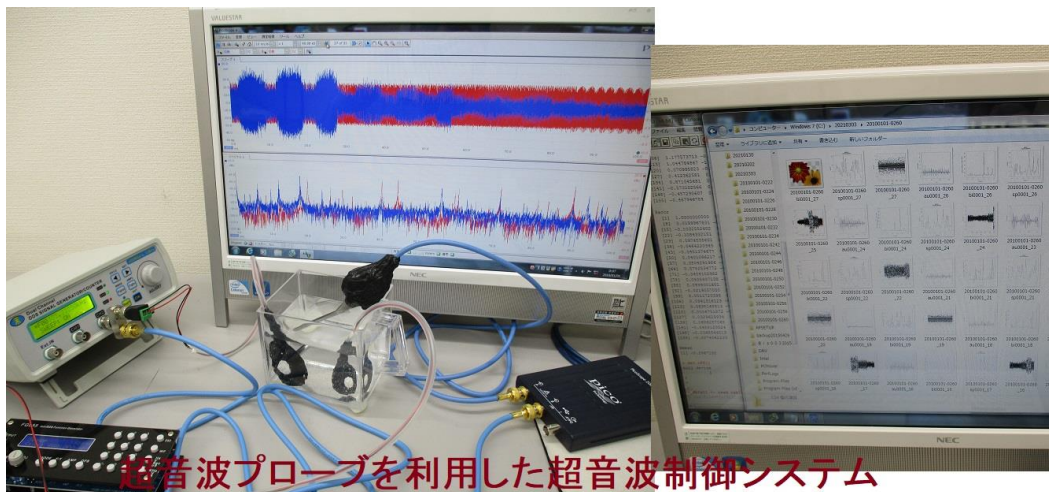
赤池 弘次 (著), 中川 東一郎 (著)

出版社: サイエンス社 (1972)

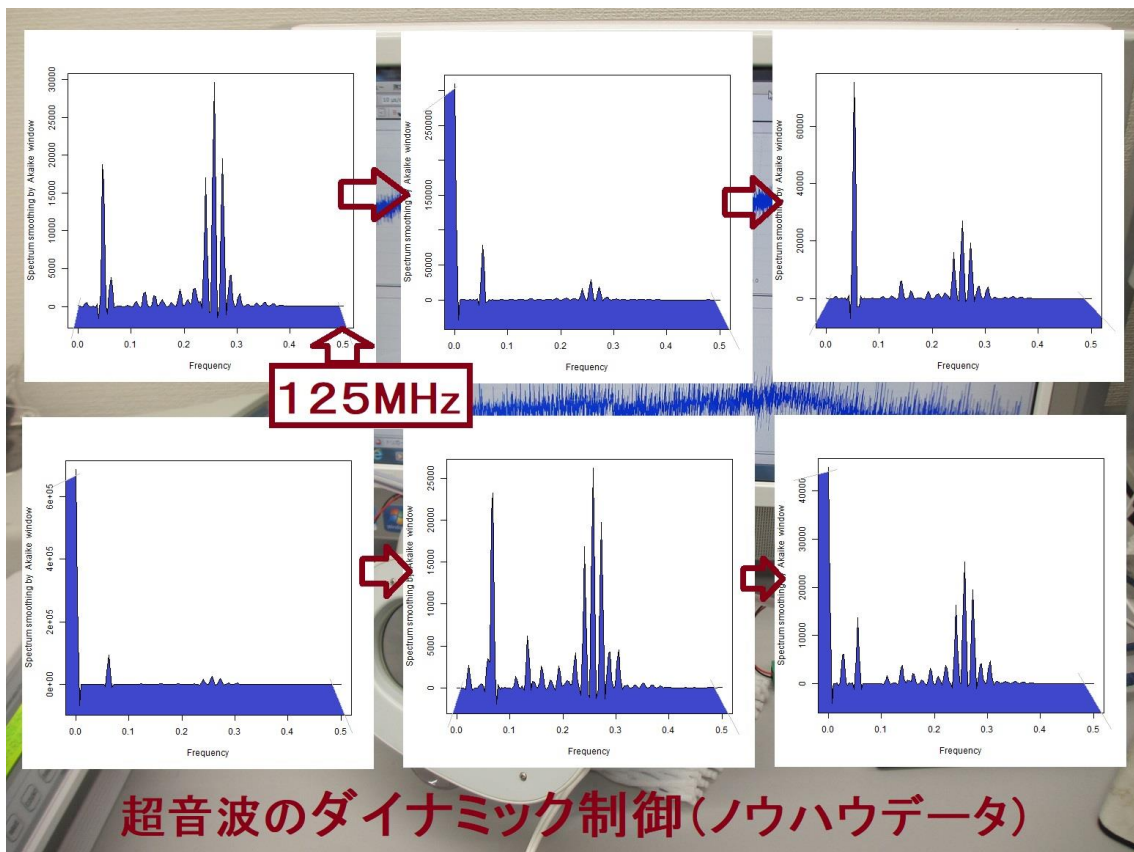
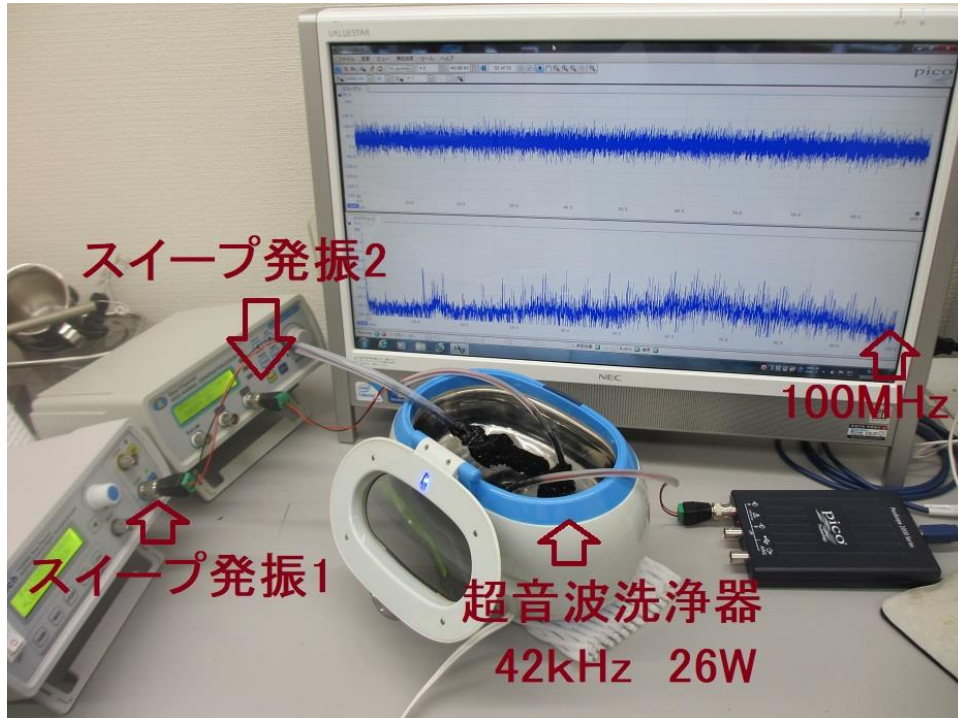
### 3：弾性波動への適用

「弾性波動論の基本」 田治米 鏡二 (著) 槇書店 (1994/10)

「弾性波動論」 佐藤 泰夫 (著) 岩波書店 (1978/03)



# 音圧解析の初歩



## サンプリング時間の表示方法

### 入力

```
data11 <- read.table("C:/2011/201101.csv", skip=0, sep=",", nrows=10)
data11
```

### 応答 (パソコンの画面表示)

	V1	V2	V3	
1	Time	Channel A	Channel B	
2	(ms)	(V)	(V)	
3	0.00000000	0.33310950	-0.07290872	
4	<b>0.00256000</b>	0.07199316	0.03616443	グラフ <b>0.5 = 195 kHz</b>
5	0.00512000	-0.16211430	-0.00927763	
6	0.00768000	0.06299020	-0.09109775	
7	0.01024000	0.05398724	0.13611260	
8	0.01280000	-0.16211430	0.15430160	
9	0.01536000	0.10800500	-0.22745450	
10	0.01792000	0.27008880	-0.23654900	

>

1 秒 / **0.00256000ms** = **390.625 kHz**    **390 / 2 = 195 kHz**

```
data11 <- read.table("C:/2011/2011103.csv", skip=0, sep=",", nrows=10)
data11
```

	V1	V2	V3	
1	Time	Channel A	Channel B	
2	(ms)	(V)	(V)	
3	0.00000000	-0.03604236	-0.11838130	
4	<b>0.00064000</b>	-0.04504532	-0.06381420	グラフ <b>0.5 = 781 kHz</b>
5	0.00128000	-0.05404828	-0.02746666	
6	0.00192000	-0.07205420	0.02706992	
7	0.00256000	-0.05404828	0.09067049	
8	0.00320000	-0.03604236	0.10885950	
9	0.00384000	-0.01803644	0.09976501	
10	0.00448000	-0.03604236	0.07248146	

1 秒 / **0.00064000ms** = **1562.5 kHz**    **1562 / 2 = 781 kHz**

注：解析には下記ツールを利用します

注：OML(Open Market License)

<https://www.ism.ac.jp/ismlib/jpn/ismlib/license.html>

注：TIMSAC(TIME Series Analysis and Control program)

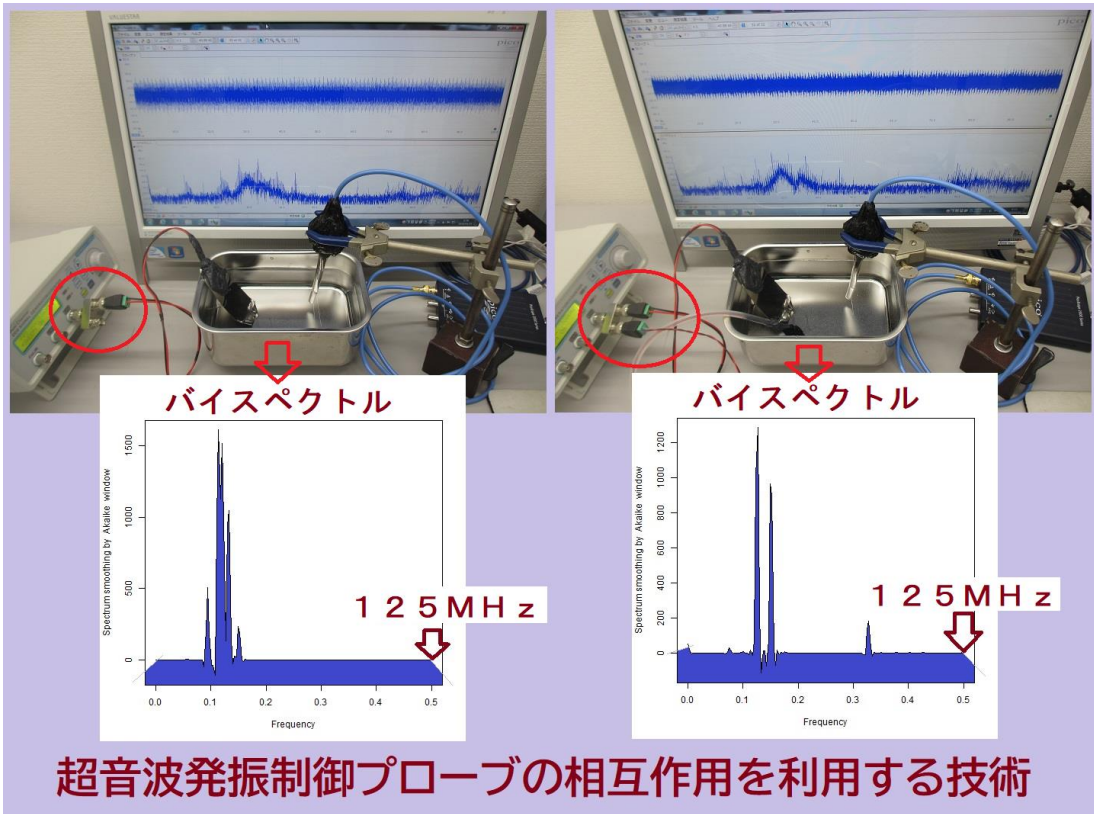
<https://jasp.ism.ac.jp/ism/timsac/>

注：「R」フリーな統計処理言語かつ環境

<https://cran.ism.ac.jp/>

## <サンプリング時間の設定は自動調整されます>

サンプリング時間	解析グラフ 0.5 の周波数
1 秒 / <b>1s</b> = 1Hz	1 / 2 = <b>0.5Hz</b>
...	
1 秒 / <b>0.02ms</b> = 50 k Hz	50 / 2 = <b>25 k Hz</b>
1 秒 / <b>0.01ms</b> = 100 k Hz	100 / 2 = <b>50 k Hz</b>
1 秒 / <b>0.0050048ms</b> = 200 k Hz	200 / 2 = <b>100 k Hz</b>
1 秒 / <b>0.0020032ms</b> = 500 k Hz	500 / 2 = <b>250 k Hz</b>
1 秒 / <b>0.0010048ms</b> = 995 k Hz	995 / 2 = <b>497 k Hz</b>
1 秒 / <b>0.0005056ms</b> = 1977 k Hz	1977 / 2 = <b>988 k Hz</b>
1 秒 / <b>0.0002048ms</b> = 4882 k Hz	4882 / 2 = <b>2441 k Hz (2.4MHz)</b>
1 秒 / <b>0.0001024ms</b> = 9765 k Hz	9765 / 2 = <b>4882 k Hz (4.8MHz)</b>
1 秒 / <b>0.0000512ms</b> = 19531 k Hz	19531 / 2 = <b>9765 k Hz (9.7MHz)</b>
1 秒 / <b>0.0000256ms</b> = 39062 k Hz	39062 / 2 = <b>19531 k Hz (20MHz)</b>
1 秒 / <b>0.0000128ms</b> = 78125 k Hz	78125 / 2 = <b>39062 k Hz (39MHz)</b>
...	
1 秒 / <b>0.016 μs</b> = 62.5MHz	62.5 / 2 = <b>31.25MHz</b>
1 秒 / <b>0.008 μs</b> = 125MHz	125 / 2 = <b>62.5MHz</b>
1 秒 / <b>0.004 μs</b> = 250MHz	250 / 2 = <b>125MHz</b>
1 秒 / <b>0.002 μs</b> = 500MHz	500 / 2 = <b>250MHz</b>
...	



## 音圧レベルの表示

入力

```
data11 <- read.table("C:/20111/2011102.csv", skip=6, sep=",")
mean(data11$V2)
mean(data11$V3)
var(data11$V2)
var(data11$V3)
range(data11$V2)
range(data11$V3)
```

### 応答 (パソコンの画面表示)

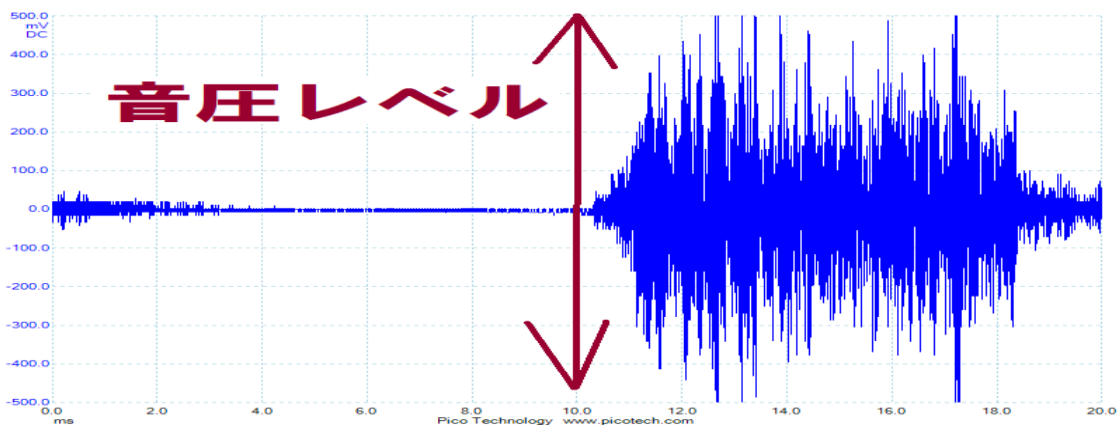
```
> data11 <- read.table("C:/20111022w/20111022-0412.csv", skip=6, sep=",")
> mean(data11$V2)      CH1 の平均値
[1] -0.001047526
> mean(data11$V3)     CH2 の平均値
[1] 3.430622e-05
> var(data11$V2)      CH1 の分散値
[1] 0.009286384
> var(data11$V3)     CH2 の分散値
[1] 0.001448241
> range(data11$V2)    CH1 の最小・最大値
[1] -0.4412366  0.4141362
> range(data11$V3)    CH2 の最小・最大値
[1] -0.1547288  0.1361126
>
```

注意

統計処理を行うために、測定値が自動的に、規格化（正規化）されています  
バイスペクトルについて理解が深まるまでは

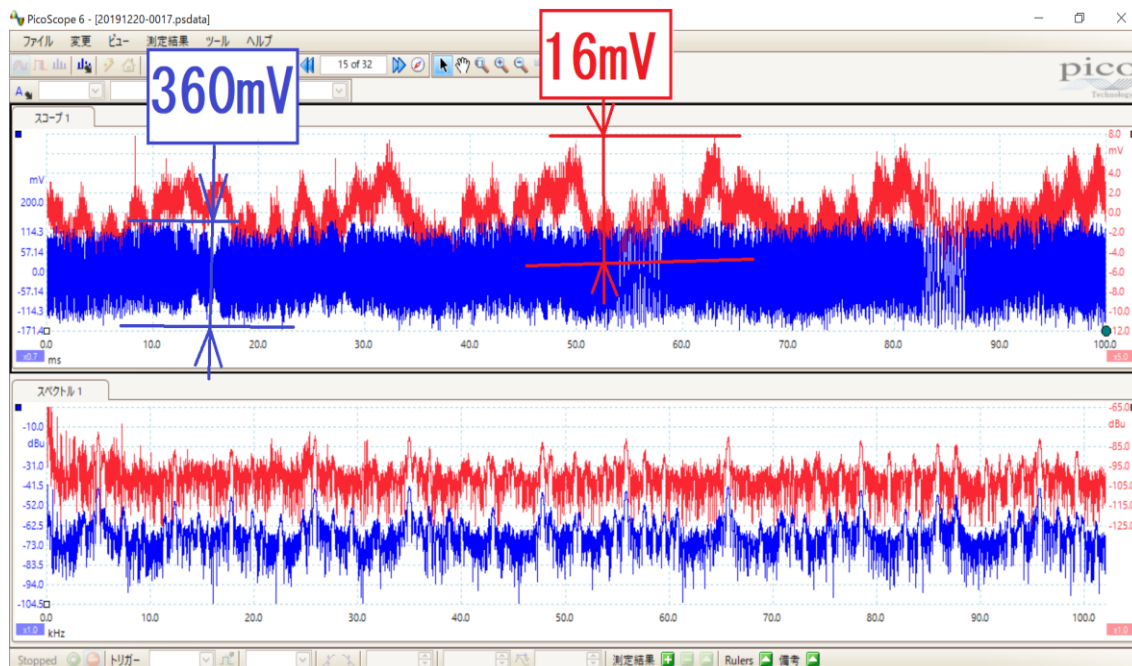
**最大・最小値、分散値、平均値** を利用することを推奨します

絶対値としての音圧は、測定データのグラフから読み取ってください  
その値に対する平均や分散を上記の処理で推定して利用します





## 具体例



```
data11 <- read.table("C:/20191220/191220-0018/20191220-0018_15.csv", skip=6,
sep=",")
mean(data11$V2)
mean(data11$V3)
var(data11$V2)
var(data11$V3)
range(data11$V2)
range(data11$V3)
>
> data11 <- read.table("C:/20191220/20191220-0018/20191220-0018_15.csv",
> mean(data11$V2)
> mean(data11$V3)
> var(data11$V2)
> var(data11$V3)
> range(data11$V2)
> range(data11$V3)
>
```

グラフ青 音圧レベル 360 mV  
グラフ赤 音圧レベル 16 mV

以上