

「ディザリング」に関するメモ

■ディザリングにまつわる誤解

△:ディザリングとは、デジタル臭さをとるためにノイズを加えること

×:ディザリングとは、アナログ感を出すためにノイズを加えること

「アナログ機器固有のノイズを加えることで、アナログ感を演出する機能」という説明は、広く流布している誤解。数学的に有用性が説明できるもの。

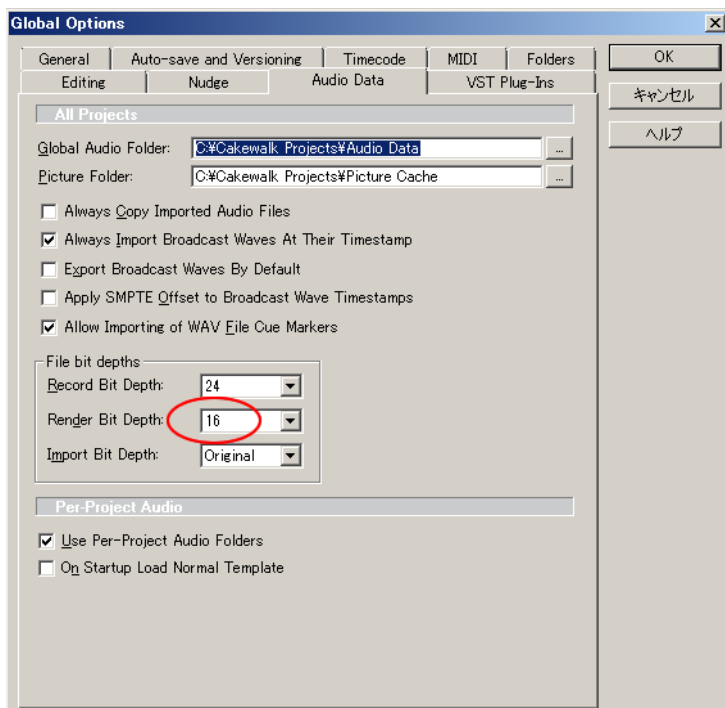
■実験

準備物

- ディザリング機能のある DAW(今回は Sonar8.53)
- トーン・ジェネレータ ([MeldaProduction](#) – Moscillator [フリー])
- スペクトラム・アナライザ ([Voxengo](#) – SPAN [フリー])

主な手順 (結果だけに興味のある方は次ページへ)

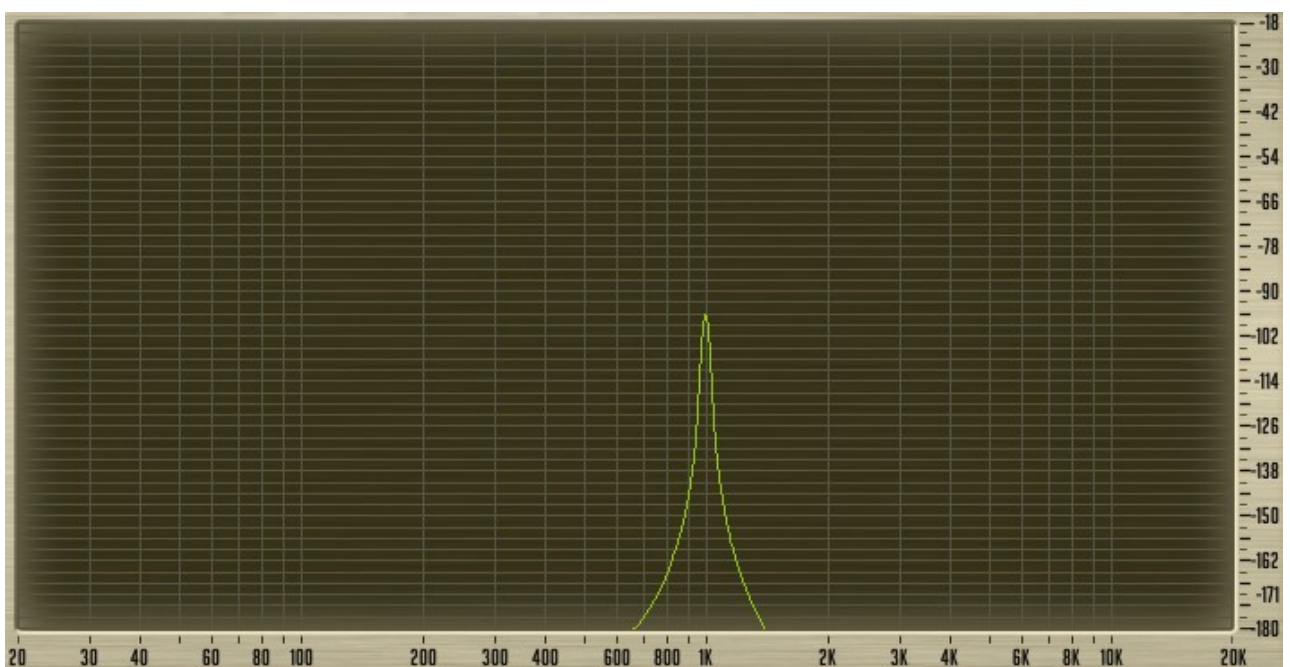
1. バウンスしたファイルが 16bit で出力されることを確認する。
Options → Global... より、”Audio Data”タブ下、”Render Bit Depth”を 16 に設定。(下図)
Sonar 以外の DAW を使用している場合は、適宜読み替え。
2. MOscillator で 1kHz のトーンを生成
3. 生成されたトーンを新規トラックにバウンス→16bit のトーンが生成される。
4. バウンス時の Dithering 種類を順次変更し(None を含む)、結果をスペアナで測定。
5. トラックのゲインをあげて、音を試聴比較。



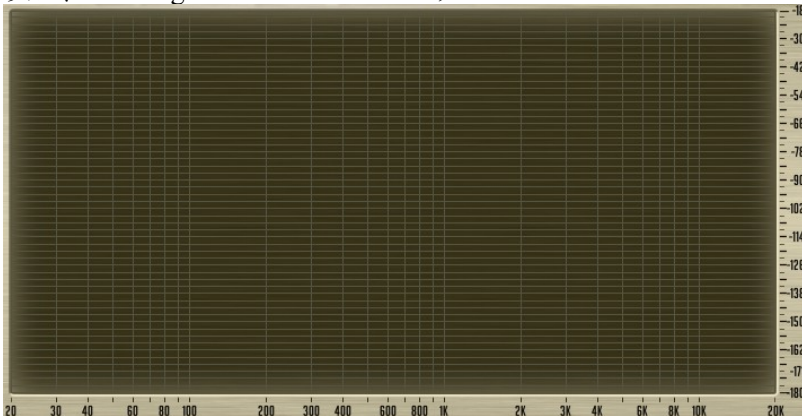
■実験

16bit で扱えるダイナミックレンジは 96dB。
つまり、正攻法で再現できる最小レベルは-96dBFS...であることを確認。

Moscillator で、1000kHz、-97dBFS のトーンを作成

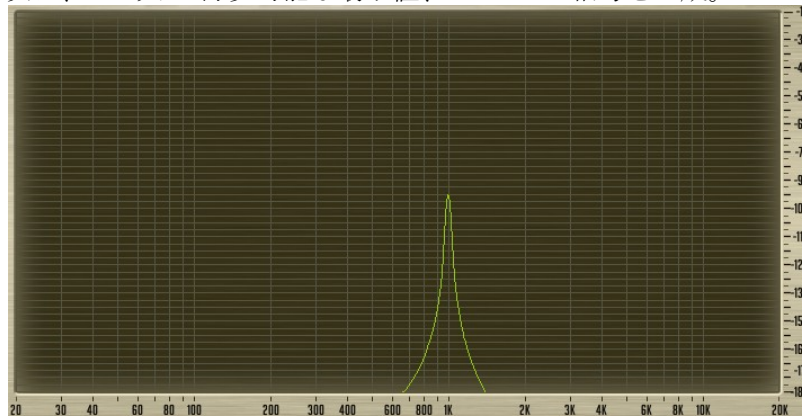


次に、Dithering = **None** で 16bit ヘバウンス

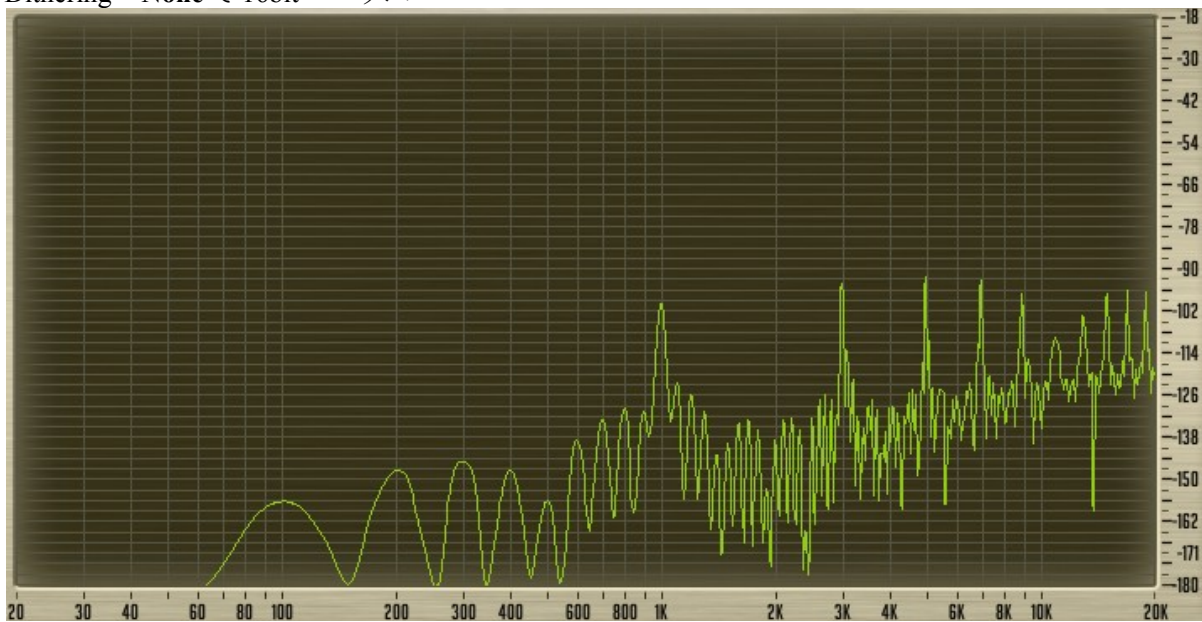


下位ビットは切り捨てられてしまうので、何もなくなる。

次に、16ビットで再現可能な最小値、-96dBFS の信号を生成。



Dithering = **None** で 16bit ヘバウンス



ぎゃー~~~~っ！！

(トラックのゲインを上げると、このノイズは聞き取り可能)

Dithering = **Rectangular** で 16bit へバウンス



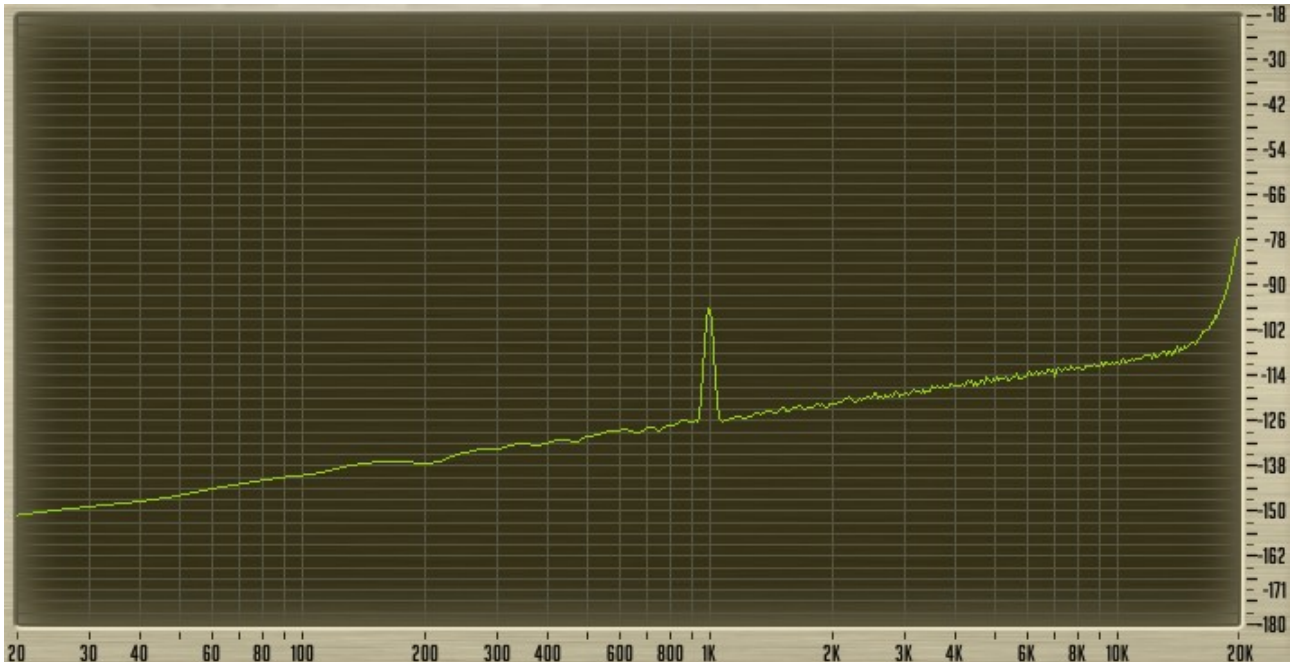
均等にノイズが付加される代わりに、Dithering 無しするときほど倍音が生じない。
(3kHz に立った倍音は、トラックのゲインを上げると聞き取り可能)

Dithering = **Triangular** で 16bit へバウンス



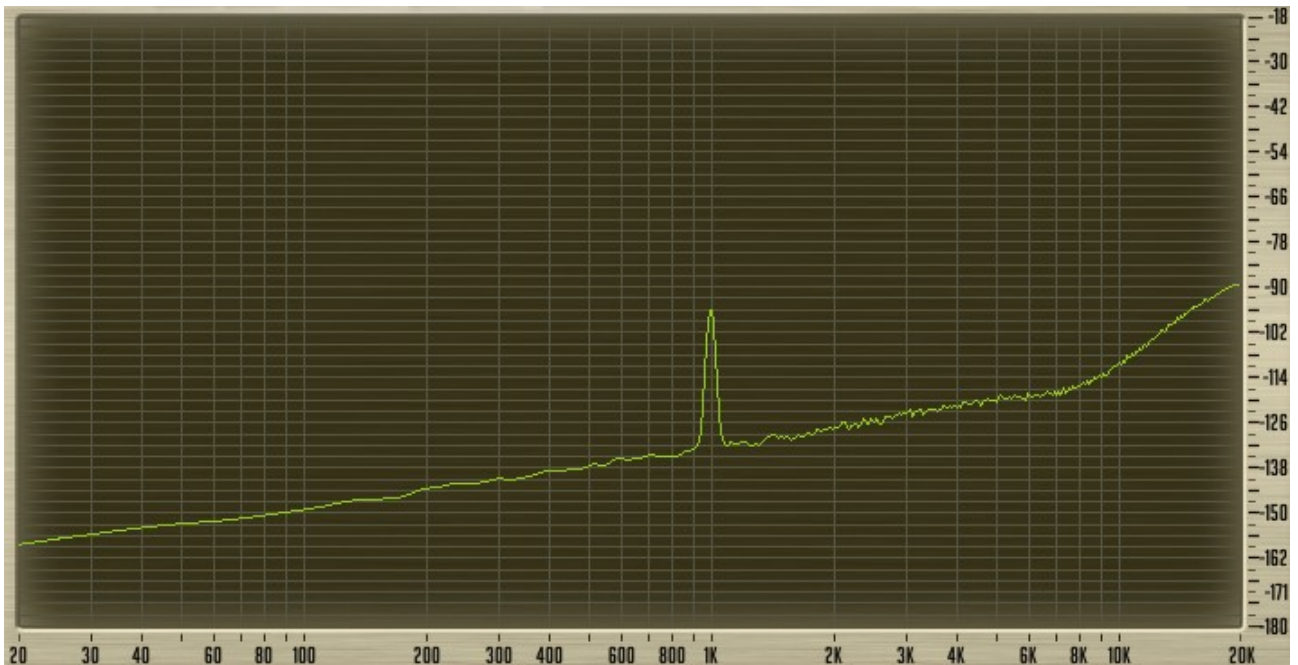
Dithering = Rectangular のときのような倍音が立たない。(その代わり CPU 負荷は若干高い)

Dithering = **Pow-r1** で 16bit へバウンス



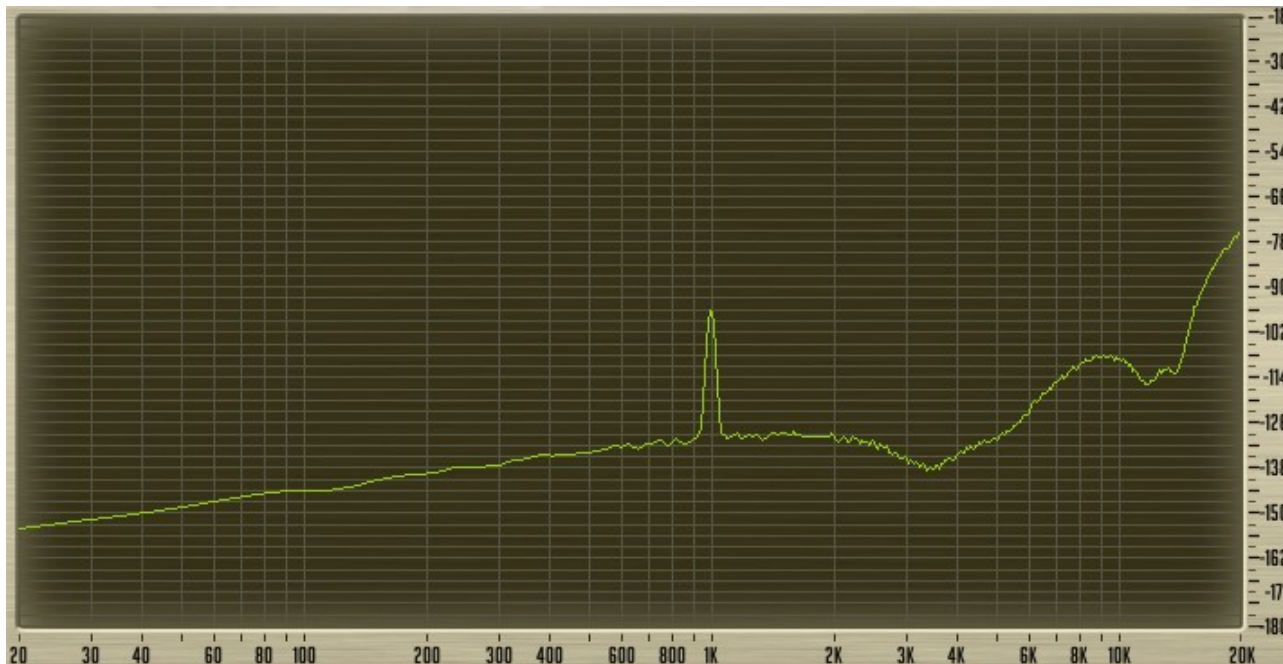
Triangular と比べて、ノイズが高域に集中している。代わりに、他の帯域のノイズ・フロアは総じて低い。

Dithering = **Pow-r2** で 16bit へバウンス



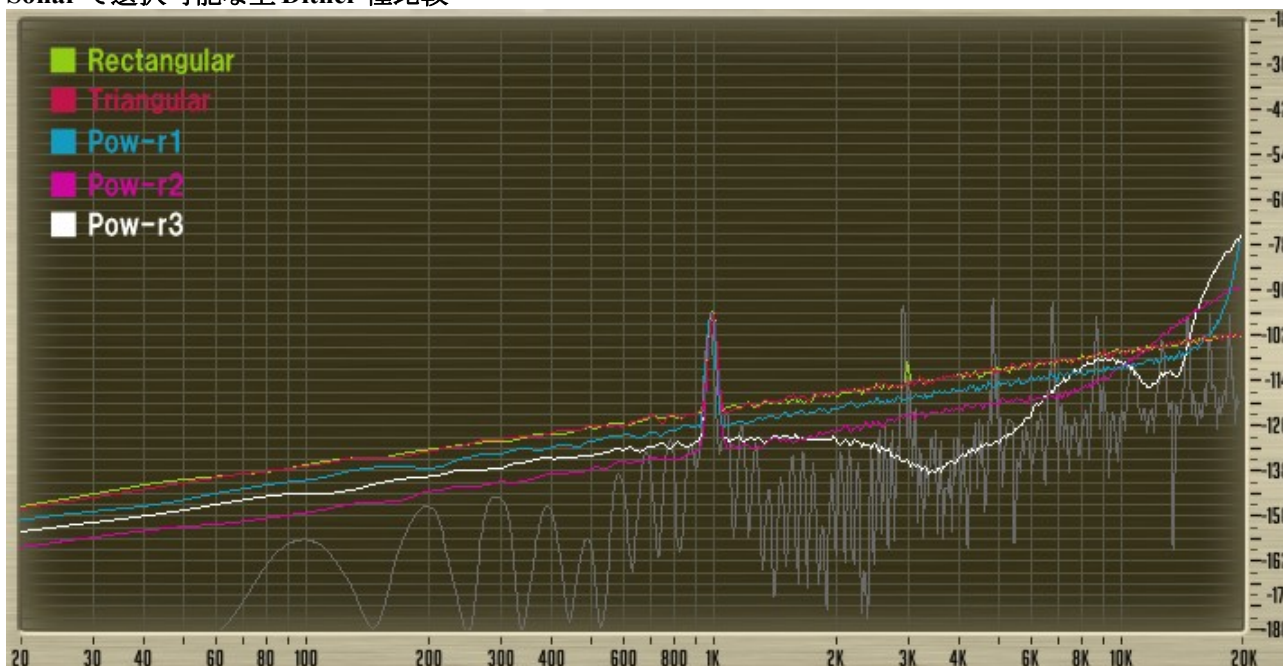
7kHzあたりから徐々にノイズ・フロアが上がる。代わりに、フロアのピーク値、高域以外のノイズ・フロアは Pow-r1 より低い。

Dithering = **Pow-r3** で 16bit へバウンス



1kHz 付近までノイズ・フロアは、Pow-r2と同様。高域のノイズ・フロアが高い代わりに、3.5kHz 付近にディップがある。→Vo など肉声の明瞭さを司る周波数帯が、Dither ノイズに埋もれにくい

Sonar で選択可能な全 Dither 種比較



おまけ

3.5kHz、-130dBFS のトーンを Dither = **Rectangular** で 16bit にバウンス



Dither ノイズに埋もれた…

3.5kHz、-130dBFS のトーンを Dither = **Pow-r3** で 16bit にバウンス



トラックのゲインを上げると聞き取り可能

→ ダイナミック・レンジ 96dB しかないフォーマットで、-130dBFS のトーンが再現可能

■まとめ

ビット数を落とす際、

Dither 無しでは原音に相関のある倍音が発生してしまうが、Dithering を行うことでこれを防げる。

※レベルの低い(ホワイトっぽい)ノイズは、レベルの高い倍音より自然に聞こえる。また倍音が生じないため原音のキャラクターを損なわない。

■ディザリングを使うべき場面

Mastering Audio (Bob Katz 著) より…

- ビットレートを落とすときは必ず使用。
- 後に 2MIX の一部として使用される素材を出力する場合など、複数回 Dithering が行われると判っている場合 Pow-r 系は避ける(後にマスターに同じ Dither を使用すると、ノイズが特定帯域に累積されるため)

■メモ

- UV22、UV22HR は、Pow-r 以上に高域に Dither ノイズを集中される、信号パターン(一見ノイズだが、ランダムではなく、循環する「パターン」になっている。) Apogee 社が特許を持っており、他社にライセンス提供している。
- Native の DAW は大抵バスが 32bit Float なので、16/24bit に出力する際は Dithering が必要。(信号バスのビット数は、32/64bit OS、32/64bit プラグインなどとは無関係)

■ デザインが効果的である仕組みの、大雑把な理由

一定間隔で数値をサンプリングする。(採取するのは、信号レベル/気温/気圧/株価、その他なんでも)
ここでは、100回の測定結果が常に「107」と想定。

```
01: 107
02: 107
03: 107
04: 107
   :
98: 107
99: 107
```

↓

さて、3桁の表示ではデータ量が多過ぎるので、1の位を切り落とすことにしました、と。

```
01: 100
02: 100
03: 100
04: 100
   :
98: 100
99: 100
```

タダの切捨てでは、一律に「100」となり 107 という数値は悠久の彼方へ…
ある意味、データが捏造される(≒Dithering 無しでビット数を落とした場合に立つ倍音)

↓

それなら、1の位を切り捨てる前に0~9のランダムな数字を足してみますか。
「0~9」は、切り捨てられる値が取り得る範囲。

```
01: 107 + 0 = 107          100
02: 107 + 1 = 108          100
03: 107 + 9 = 116      切捨て  110
04: 107 + 8 = 115        →      110
   :
98: 107 + 7 = 114          110
99: 107 + 1 = 108          100
```

100 か 110 を行ったり来たり。
しかし、切り捨て後 100 回の平均をとると、おおよそ「107」になる！

```
<?php
$sum = 0;

for ($i=0; $i<100; $i++) {
    $val = 107 + rand(0, 9);
    $sum += floor($val / 10) * 10;
}

echo "Average: " . ($sum / 100);
```

参考文献:

“Mastering Audio” Bob Katz

<http://www.digido.com/mastering-audio-book.html>

“Dither Explained” Nika Aldrich

<http://www.users.qwest.net/~volt42/cadenzarecording/DitherExplained.pdf>

Studio Gyokimae

<http://pspunch.com/pd/>

初版 2012/2/13