

「適切な音量」を考える

~ユーザーの鼓膜を守ろう~

音割れしない程度に
ラウドネスを
既定の値にそろえよう

ラウドネス規制 (テレビ)

2012年10月1日 わかりやすく・新しいルール T032 導入開始 VU→vu+「ラウドネス LKFS」
テレビ放送における音声レベル運用規準

今の放送音声の現状…

下の図をご覧ください。残念ながらこれが今の放送音声の現状です。
これまではVUメータを使っていましたが、それだけでは番組やCMの音量感を揃えられません。

そこで登場したのが「ラウドネスメータ」! ✨

民放連では人の感じる音の大きさを数値で表す「ラウドネス」という新しい「ものさし」に着目し、
T032「テレビ放送における音声レベル運用規準」という新しいルールを作りました。

「平均ラウドネス値」を国際標準の
「-24LKFS」に揃えましょう。

-24
LKFS

現状の
放送

「平均ラウドネス値」がバラバラ

エッ!?
聞こえないよ!

聞きやすい!

これからの
放送

「平均ラウドネス値」が統一されて
とても聞きやすい!

音圧戦争（ラウドネス・ウォー）

テレビ以外にも...

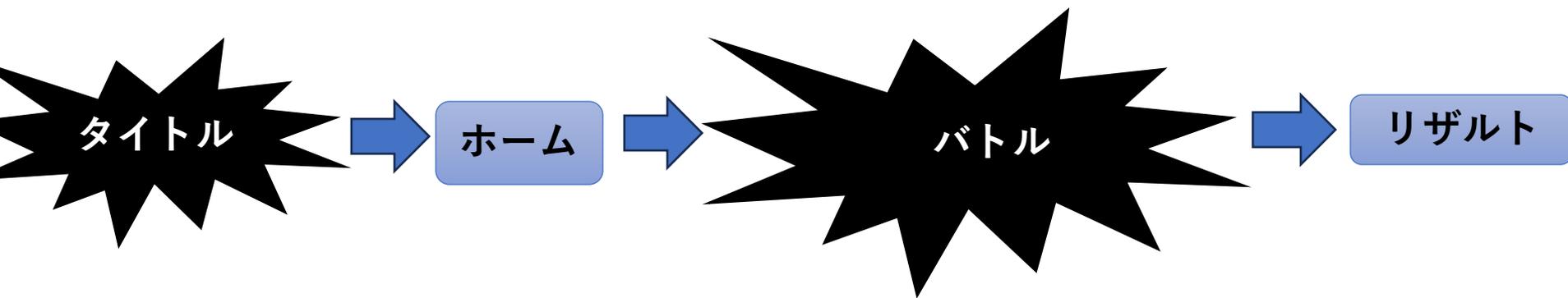
- 音圧が高いほど「良い音」に聞こえやすい
- 単純に目立たせたい

音圧戦争勃発！！

現在は各コンテンツ別でラウドネス規制を設けている場合がある

ゲームでも

- ・シーンごとにラウドネスが異なるとつらい
- ・音ゲーで曲ごとにラウドネスが異なるとつらい



- ・各社でラウドネスに制限をかけ、調整している（ラウドネスノーマライゼーション）

音の大きさ

- RMS (Root Mean Square, 二乗平均平方根)
 - 平均的な音の大きさ
- ラウドネス
 - 人が感じる音の心理的な大きさ
 - 周波数によって聞こえ方が異なる
 - 低い音は小さく、高い音は大きく聞こえる、超音波は聞こえない
- LKFS (=LUFS)
 - ラウドネス測定単位
 - 人間の聴覚特性によって補正をかけた音量
 - +1dB \cong +1LKFS

ラウドネスの種類

- **モーメンタリラウドネス値 (Momentary Loudness)**
 - 400msの瞬間的なラウドネス
- **ショートタームラウドネス値 (Short-Term Loudness)**
 - 3秒間のラウドネス
- **平均ラウドネス値 (Integrated Loudness)**
 - 最初から最後まで通したラウドネス
- **Integrated Loudnessでそろえることが多い**
 - 冒頭から末尾までを同じ音量でそろえることを意図しているわけではない
 - ダイナミックレンジ (音量の大小の幅) があるほうが自然
 - 目的は聞きやすくすること

ラウドネスメータ

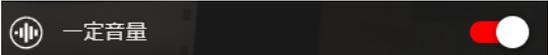
- **ライブ測定型メータ**
 - リアルタイムで測定
 - ソフトウェアの再生音量などによってラウドネスが変わってしまう
 - リファレンス音源を使って揃える
 - 最終的に出力される音量感を揃えるのに有用
 - Orban Loudness Meter (無料版) など
- **ファイルベース型メータ**
 - 完パケ状態で測定
 - 測定時間がリアルタイムより短い
 - DAW (音楽制作ソフト) のプラグイン、単体の測定プログラムなど
 - ffmpegでもできる

ラウドネスノーマライゼーション

Q. ラウドネスを無理やり変えると元の音楽を壊してしまわないか？

A. 壊すやり方と壊さないやり方がある

- 全体的に均一にする手法（ダイナミックレンジ圧縮(DRC)など）
 - 音楽のダイナミクスを損なう可能性がある
 - YouTubeの一部動画で使われる一定音量機能はこれ
 - 基本音楽ではオフですが、ライブのアーカイブとかでオンになっていることも



一定音量

- 1曲単位（もしくは1アルバム単位）での単純な音量変更
 - 音量を変えるだけだが、気を付けるべきこともある

ラウドネスノーマライゼーション (Spotify)

- **マイナスのゲイン**は、ラウドネスレベルが-14 dB LUFSになるように、大音量のマスタリングに適用されます。このプロセスでは、マスタリングと比較して音量が減少するだけで、追加の歪みは生じません。
- **プラスのゲイン**は、ラウドネスレベルが-14 dB LUFSになるように、小音量のマスタリングに適用されます。トラックのヘッドルームを考慮し、音質を維持できるようロッシェー符号化のために1 dBのヘッドルームを残しています。

例: トラックのラウドネスレベルが-20 dB LUFSで、そのトゥルーピークの最大値が-5 dB FSである場合、そのトラックのラウドネスは最大-16 dB LUFSまでしか上がりません。

<https://support.spotify.com/jp/artists/article/loudness-normalization/>

ラウドネスノーマライゼーション

ラウドネスを小さくする場合

⇒問題なし

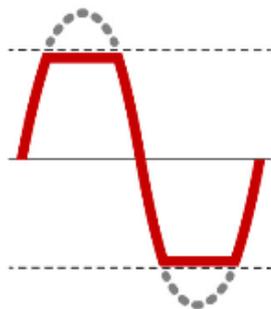
ラウドネスを大きくする場合

⇒音割れが起きる可能性がある

⇒ノイズが大きくなる（生音の場合）

音割れ（クリッピングノイズ）

- ビット深度が16, 24 bitの場合、表現できる上限値（0dbFS≒0LKFS）を超えると、強制的に上限値に抑制（クリップ）され、音の情報が失われてしまう
- この結果、音が歪んだりバチバチといったノイズが出る音割れが生じる

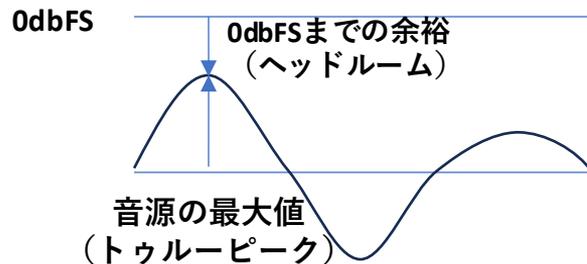


画像：

https://tascam.jp/jp/feature/32-bit_float

トゥルーピークとヘッドルーム

- 音割れが生じないように、音源の最大値（トゥルーピーク）を意識する必要がある
- トゥルーピークが0dbFSを超えないように余裕を持たせる（ヘッドルーム）



ターゲットラウドネス

	ターゲットラウドネス	トゥルーピークの最大値
Spotify	-14dB LUFS	-1dbFS
Amazon Alexa	-14dB LUFS/LKFS	-2dBFS
テレビ	-24LKFS ±1	-1dBTP
SCE (PS3)	-23LUFS ±2	
SCE (PSVita)	-18LUFS	
DeNA モバイルゲーム規定 (楽曲)	ショートタームで-13LKFS	
DeNA モバイルゲーム規定 (SE)	RMS -13dB(response 90ms)	

← テレビに近づけている

← Youtubeに近づけている

(Spotify) <https://support.spotify.com/jp/artists/article/loudness-normalization/>

(Alexa) <https://developer.amazon.com/ja-JP/docs/alexa/flashbriefing/normalizing-the-loudness-of-audio-content.html>

(Netflix) <https://partnerhelp.netflixstudios.com/hc/ja/articles/360001794307-Netflix-サウンドミックスの仕様と実践ガイド>

(テレビ) <https://www.j-ba.or.jp/category/t032>

(SCE) <https://www.slideshare.net/slideshow/sigaudio1-cedec2012/15240036>

(DeNA) https://qiita.com/ADX_kawaguchi/items/edd6ecb2f2e77ce33b52

そのほか：<https://av.watch.impress.co.jp/docs/series/higuchi/1142525.html>

ラウドネス、トゥルーピーク計測、ノーマライズ用ソフト

- DAW（音楽制作ソフト）、プラグイン
- 動画編集ソフト
- （計測のみ）Unityサウンド系ミドルウェア（CRI ADX、Wwise）

- フリーソフト
 - （ノーマライズのみ）Audacity
 - （計測のみ）Youlean Loudness Meter、Orban Loudness Meter
 - ffmpeg
 - CUIのみだが、CUIのおかげで自動化プログラムを作りやすい。
 - （計測のみ）Unity Technologies NativeAudioPlugins、Oculus Audio SDK

ラウドネス、トゥルーピーク計測、ノーマライズ (ffmpeg)

- ffmpegを使ってラウドネス計測を行う

```
ffmpeg -i ファイル名 -filter_complex ebur128=peak=true -f null -
```

- 見るのは次の2つ
 - Integrated loudnessの最初の数値 (I: の項)
 - True peak

https://pspunch.com/pd/article/measuring_loudness/

<https://ffmpeg.org/ffmpeg-filters.html#loudnorm>

https://nico-lab.net/ebur128_with_ffmpeg/

```
Integrated loudness:
I:                -6.1 LUFS
Threshold:        -16.1 LUFS

Loudness range:
LRA:              2.7 LU
Threshold:        -26.2 LUFS
LRA low:          -7.6 LUFS
LRA high:         -4.9 LUFS

True peak:
Peak:             1.0 dBFS
```

ラウドネス、トゥルーピーク計測、ノーマライズ (ffmpeg)

- ffmpegによって音量を変更する
 - ffmpegにはラウドネスノーマライゼーション用にloudnormやdynamdnorm, speecnormといった機能もあるが、場合によっては圧縮がかかる可能性があるなのでここでは使わない
 - 1dbあげるとおよそ1LUFS上がる
 - True peakが-1dbFSを超えない値になるよう気を付ける
 - 1度音割れしたら元に戻らない。再度やり直す

`ffmpeg -i 入力ファイル名 -af volume=-8dB 出力ファイル名`

- (dB数は上げたい、下げたい分を入力する)

<https://qiita.com/mml/items/c28d69ab889eb36dd522>

```
Integrated loudness:
I:          -14.2 LUFS
Threshold: -24.3 LUFS

Loudness range:
LRA:         2.6 LU
Threshold: -34.3 LUFS
LRA low:    -15.7 LUFS
LRA high:   -13.0 LUFS

True peak:
Peak:       -5.5 dBFS
```

補足

- 意図がある場合はターゲットラウドネスに縛られなくてもよい
 - 呟いた声と叫び声を同じラウドネスにしなくてもよい
 - とはいえ上限を超えてうるさすぎてはいけない
- 意図がない場合はできる限りそろえる
- 最終的には人の耳で判断する

効果音

- 効果音もアセット段階では音量をそろえる

- 参考.DeNAモバイルゲームの規定：

- 楽曲はショートタームで -13LKFS
 - SEはラウドネスではなく、RMS -13dB(response 90ms)
 - https://qiita.com/ADX_kawaguchi/items/edd6ecb2f2e77ce33b52
 - ラウドネスでなくRMS？
 - たぶんMomentary Loudnessの400msじゃ長すぎるからRMSを使ってる？
 - ターゲットラウドネスに近い値のdbFSにする？

- 効果音再生段階で音量を調節する

- 重要度など、意図によって調節する

https://www.youtube.com/watch?v=Zf8o_0J3p-o

<https://blog.audiokinetic.com/ja/loudness-processing-best-practice-chapter-3-scalable-loudness-processing-for-games/>



効果音

- 音は同時に鳴ると大きくなる
 - 物理の「重ねあわせの原理」
 - 同じ音が同時に2個鳴ると音量2倍、3000個鳴ると3000倍

■大きくなりすぎないようにするテクニック

- 同時になる音数を制限する
 - 先着優先と後着優先
- 2個目以降が鳴る場合は音量を小さくする
- ピークが重ならないようキューを使い少しずらす
 - 鳴らしたいタイミングからずれてしまう可能性あり
- 出力直前にコンプレッサー・リミッターをかける
 - 耳を守る

https://note.com/adx_kawaguchi/n/n2e28b57cafdd

https://game.criware.jp/manual/native/adx2/latest/criatom_feat_voice_pool.html

まとめ

- 音割れしない程度にラウドネスを規定値にそろえると聞きやすくなる
 - 音割れ防止にトゥルーピークに-1dbFSぐらいヘッドルームを持たせる
- 耳でも判断する
 - 人の耳に勝るセンサーはない
 - プレイ環境を想定した様々な視聴環境を想定・用意できるとよい