

純正律から生まれるより良い音律

東京都立小石川中等教育学校 大久保 幸太

要旨

本研究は、純正律(Cmajor)を基準にして、主要な和音(Cmaj, Em, Fmaj, Gmaj, Am)の良好さを保ちながら、DmとBdimの和音の不協さを改善する新しい律を求める目的とする。各和音の汚さを評価する関数を作成し、その関数に基づき最適化した。その結果、純正律に少し手を加えただけではより良い音律を作ることは不可能なことが分かった。

前提/背景

純正律…音の振動数が基音に対して単純な整数比によって生み出された音階。本研究では基音をC4とする。

e.g. E4=C4*5/4[Hz]、G4=C4*3/2[Hz]、C5=C4*2[Hz]

※アルファベットの後ろについている数字はオクターブを表す。

和音…複数の音を組み合わせて作った音。

e.g. Cmaj : [C, E, G]

Dm : [D, F, A]

Bdim : [B, D, F]

なお、例えばCmajにおいて、Cを根、Eを第三、Gを第五の音ということにする。

今回最適化を図る和音は

Cmaj, Dm, Em, Fmaj, Gmaj, Am, Bdimの7種である。

理由としてはこの7種はCを基音とする調で重要なコード（ダイアトニックコード）と言われるものだからである。

既存の純正律は、この7種の内、DmとBdimが綺麗に響かないコードとなっており、これを解決する。

c.f. Dmなどが綺麗に響かない理由について

純正律の定める比に従って和音の構成音の周波数比を観察すると以下のようになる。

Cmaj[CEG] : C:G=3:2

Dm[DFA] : D:A=40:27

Cmajは構成音を簡単な整数周波数比で表されるがDmは表されない。これがDm等が綺麗に響かない理由とされる。

汚さ評価関数について

本研究では和音の種類ごとの理想的な目標とする比を以下のように定める。

Major: 1:5/4:3/2

Minor: 1:6/5:3/2

Diminished: 1:6/5:45/32

Diminishedは学術的に理想とされる比が複数あるが、今回はこの比を採用する。少し分母が大きいが実用的に用いられる妥当な近似の一つである。

和音の汚さは、目標日からのセント(cent)差の二乗和として定義する。(=最小二乗法)

なお、セント差とは以下の事である。

任意の二音f-aとf-bのHz比:r=f-b/f-aがあり、目標比がr-targetであるとき、セント差は

$$\Delta_{cent} = 1200 \cdot \log_2 \left(\frac{r}{r_{target}} \right)$$

である。

和音について根→第三、根→第五のそれぞれのセント差の二乗和の合計をその和音の汚さとできる。

また、既に良く響く5つの和音に対しては高い重みを、汚い2つの和音に対しては小さい重みを与えて、7つの和音の汚さの重み付き和をとる。

この重み付き和を出力する関数が汚さ評価関数で、この値を最小化する。

```
# 全体目的関数
def objective(x, R, notes, triads, weights):
    # x: セントオフセット配列 (note順)
    freqs = {}
    for i, n in enumerate(notes):
        freqs[n] = f0 * R[n] * (2**((x[i]/1200)))
        total = 0.0
    for (root, third, fifth, t_type), w in zip([(t[:4] for t in triads], weights)):
        target = triad_target_ratios(t_type)
        d = triad_dissonance(freqs, root, third, fifth, target)
        total += w * d
    return total
```

最適化方法

未知量：C,D,E,F,G,A,Bの7つについて各音の純正値からのセントオフセットXiを導入する。(iはC~Bの7つの音)
ここで、純正値から大きくずれすぎても面白くないため、 $|Xi| <= 30\text{cent}$ とする。

このとき、未知量は

$$fi = f0 \cdot Ri \cdot 2^{\frac{Xi}{1200}}$$

で表せる。

f0は基音の周波数、Riは音iの純正律での基音との比である。
このようにして作られた7つの音を汚さ評価関数に入れて、出力が最小化されるようなXiの組を探す。

プログラミングの実装はpythonのscipy.optimize.minimizeのL-BFGS-Bで行うこととする。

結果/考察

今回のプログラムで最適とされたXiは以下の通り。

i	C	D	E	F	G	A	B
Xi	+30	+30	+30	-30	-30	-30	-30

全ての音がちょうど制約の端に張り付いてしまっており、最適化の局所解というより、制約一杯まで音を動かして苦しいバランスを取った状態になってしまった。

また、評価関数の値が非常に大きく、今回の分析の条件ではこの7和音が同時に綺麗な響きとなる音律を作れないことが分かった。

i	C	D	E	F	G	A	B
Xi	+228	+884	+129	-743	-0.194	-869	-43.2

また、制約を外した結果このようになった。

例えば、Dの+884centというのは元のDの音から約9半音分音を上昇させるという意味であり、あまりにも音が大きくなってしまっている。

結論/展望

純正律を基として音律を作成しても、主要7和音を同時に綺麗響かせることはできなかった。

今後の研究アプローチとして、

- ・制約を外した時の結果の解釈を深める
- ・純正律比からのずれ以外の和音の汚さの評価法を考える
- ・Bdimの理想比の再考