

Petersen 五行阴阳音乐系统—底层音律说明

Abe.Chua (初稿 B)

2025-08-25

本文件为底层音律说明的升级版，补充了可复现的参数表、实现/导出实践、测试协议与最小化生成脚本。目标：便于工程实现 (Python/SC/SuperCollider)、听觉验证与 DAW/合成器集成。

1 符号与常量 (概览)

简要重述主要符号：

- 黄金比例: $\varphi = \frac{1 + \sqrt{5}}{2} \approx 1.6180339887$ 。
- 基频: F_0 (默认 220 Hz)。
- 五行角度基位: $\theta_e = 72^\circ \cdot e$, $e \in \{0, \dots, 4\}$ (0: 金, 1: 木, 2: 水, 3: 火, 4: 土)。
- 阴阳极性: $p \in \{-1, 0, +1\}$; 阴阳偏移: $\Delta\theta$ (默认 5°)。
- 折叠基准比 (八度选择): $R_{\text{oct}} \in \{2, \varphi\}$ 。

2 标准参数表 (默认值与可搜索范围)

为保证复现性与方便扫参，建议在文档或配置文件明确列出默认值与实验范围：

- $F_0 = 220$ Hz (可试 160–440 Hz)
- $\Delta\theta = 5^\circ$ (可试 0° – 18° , 或特殊值如 10° , 18°)
- $R_{\text{oct}} = 2$ (可选 φ)
- σ (和弦相容性宽度) = 50 cents (可试 10–200 cents)
- α_e (音行跳跃惩罚) = 1.0 (可试 0.1–5.0)
- BPM 基准 B (实验值按需要)
- MPE pitch_{bend} range 建议初期 ± 200 cents (根据最大偏移调整)

3 角度到频率—实现细节 (复述与边界)

核心映射公式与折叠策略如前文所述。实现时注意：

- 使用高精度浮点 (double) 计算 ϕ 的幂与对数，避免累积误差。
- 折叠到区间 $[F_0, F_0 \cdot R_{\text{oct}})$ ，通过

$$k = - \left\lfloor \log_{R_{\text{oct}}} \frac{f_{\text{raw}}}{F_0} \right\rfloor$$

保证单一标准区间。

- 输出 cents 时采用基准 F_0 的相对 cents: $\text{cents} = 1200 \log_2(F/F_0)$ 。

4 示例：15 方位 ($F_0=220\text{Hz}$, $\Delta\theta = 5^\circ$, $R_{\text{Oct}} = 2$)

下表为示例计算（折叠后频率均落入 $[220, 440)$ ）。可将此表纳入 README 或导出文件以便听评。

- 金 ($e=0$)
 - $p = -1$: $F \approx 425.07 \text{ Hz}$, $\text{cents} \approx 1140$
 - $p = 0$: $F = 220.00 \text{ Hz}$, $\text{cents} = 0$
 - $p = +1$: $F \approx 227.47 \text{ Hz}$, $\text{cents} \approx 58$
- 木 ($e=1$)
 - $p = -1$: $F \approx 344.28 \text{ Hz}$, $\text{cents} \approx 775$
 - $p = 0$: $F \approx 355.99 \text{ Hz}$, $\text{cents} \approx 833$
 - $p = +1$: $F \approx 368.20 \text{ Hz}$, $\text{cents} \approx 891$
- 水 ($e=2$)
 - $p = -1$: $F \approx 278.70 \text{ Hz}$, $\text{cents} \approx 410$
 - $p = 0$: $F \approx 287.98 \text{ Hz}$, $\text{cents} \approx 468$
 - $p = +1$: $F \approx 297.66 \text{ Hz}$, $\text{cents} \approx 523$
- 火 ($e=3$)
 - $p = -1$: $F \approx 225.27 \text{ Hz}$, $\text{cents} \approx 42$
 - $p = 0$: $F \approx 232.98 \text{ Hz}$, $\text{cents} \approx 99$
 - $p = +1$: $F \approx 242.75 \text{ Hz}$, $\text{cents} \approx 170$
- 土 ($e=4$)
 - $p = -1$: $F \approx 364.25 \text{ Hz}$, $\text{cents} \approx 871$
 - $p = 0$: $F \approx 376.98 \text{ Hz}$, $\text{cents} \approx 933$
 - $p = +1$: $F \approx 388.78 \text{ Hz}$, $\text{cents} \approx 986$

45 音区通过对上表分别乘以 $s \in \{1/\varphi, 1, \varphi\}$ （或 $2^{-1}, 1, 2$ ）生成；建议导出 CSU/Scala/.tun 三种格式。

5 导出格式与 DAW/合成器集成实践

建议同时支持三类输出：

1. Scala (.scl)：用于静态音阶比较、格式通用。
2. MIDI Tuning Standard (.tun) 或 SysEx：用于支持自定义微分音调律的合成器。
3. MPE + per-voice pitch-bend fallback：当合成器支持 MPE 时，每声部使用独立 pitch-bend；否则用 base MIDI note + pitch-bend。建议策略：
 - 计算目标频率与最邻近 MIDI note（等分十二平均律）差值（cents）。
 - 若绝对差值 $\leq 100 \text{ cents}$ ，分配该 MIDI note 并通过 pitch-bend 调整；否则选择更合适 octave / base note。
 - 默认 pitch-bend range = $\pm 200 \text{ cents}$ ；如果系统需要更大范围，动态调整或选择不同基准 note 组合以压缩到范围内。

6 和弦评分、搜索与约束（实现细化）

建议实现要点：

- 在计算 δ_{ij} 时同时记录对应的小整数比 ($n:m$)，用于后续分析与可视化。
- 对于多音和弦，引入基频对齐惩罚（若两个音的基频接近简单倍频关系则加分）。
- 搜索时采取分层策略：先在同一五行/相邻五行中枚举候选，再按和弦分数 $S(C)$ 排序，最后对时间窗口内的平滑性进行二次筛选（避免瞬时 cluster）。

7 实时合成与抗失真注意

实现时需注意：

- 频率变化平滑（滑音/portamento）：对频率包络应用分段线性或低通滤波，避免瞬时大跳导致 aliasing。
- 采样率与插值：使用至少 48 kHz，带线性相位低通滤波器的 resampler 用于 pitch bend 引起的频率变化。
- 预计算 LUT：在实时环境中预先生成 15/45 音频频率表与 pitch bend 值，减少计算负担。

8 可视化、调试与听觉评估流程

最小可行试验（MVP）流程：

1. 使用默认参数生成 15/45 的 Scala 与 MIDI 文件。
2. 在受控条件下进行主观 A/B 听测（建议 20-30 位受试者），并记录偏好。
3. 同步记录客观指标：平均偏差 (cents)、最大偏差、与常见小整数比的分布。
4. 用热力图显示 Petersen 图上节点被访问频度、和弦评分分布与路径转移矩阵。

9 音色补丁建议（工程化）

为便于快速试验，提供五行到合成参数的映射示例（仅参数示范，实际数值按合成器调整）：

- 金: bell FM / high-q resonant filter, bright harmonic boost。
- 木: plucked sample 或 physical model, transient + body noise。
- 水: pad + portamento, low-pass with subtle chorus。
- 火: brass/lead with saturation, shorter release, high harmonic content。
- 土: low oscillator, heavy LP filter, long decay。

阴阳可映射为: brightness, harmonic content, attack/decay 微调。

10 测试与参数搜索建议（自动化）

实现时建议：

- 网格或贝叶斯优化搜索 $\Delta\theta$, σ , α_e , R_{oct} 。
- 每组参数自动生成一组 MIDI/Audio，记录客观指标并做批量 A/B（或对照 12-TET）听测。
- 保存所有结果与元数据（参数 JSON + timestamp + render 文件名），便于后续分析与复现。

11 元数据、引用与许可证

请在项目中添加术语表与参考文献 (Bohlen-Pierce, Sevish, microtonal tuning 文献), 并声明代码与数据许可证 (建议 MIT/CC-BY 兼容组合)。

12 附录 A: 生成与导出最小 Python 脚本 (工程化示例)

下列脚本为最小可运行示例: 生成 15 音位表, 导出 CSV 与 Scala (.scl)。请将脚本保存在 tools/generate_tuning.py 并在 macOS 终端运行 python3。

```
# filepath: tools/generate_tuning.py
# 运行: python3 tools/generate_tuning.py
import math, csv

phi = (1+5**0.5)/2
F0 = 220.0
dtheta = 5.0
R_oct = 2.0

elements = ['金','木','水','火','土']
rows = []

def raw_freq(theta):
    return F0 * (phi ** (theta/72.0))

def fold_freq(f):
    k = -math.floor(math.log(f / F0, R_oct))
    return f * (R_oct ** k)

def cents(f):
    return 1200*math.log2(f / F0)

for e_idx, name in enumerate(elements):
    theta_e = 72.0 * e_idx
    for p in (-1,0,1):
        theta = theta_e + p*dtheta
        f_raw = raw_freq(theta)
        F = fold_freq(f_raw)
        rows.append({
            'element': name,
            'e': e_idx,
            'p': p,
            'theta': theta,
            'freq': round(F,6),
            'cents': round(cents(F),3)
        })

# CSV
with open('tools/15_positions.csv','w', newline='') as f:
    w = csv.DictWriter(f, fieldnames=['element','e','p','theta','freq','cents'])
    w.writeheader()
    w.writerows(rows)

# Scala .scl (single octave relative to F0)
with open('tools/petersen_15.scl','w') as f:
    f.write("! petersen_15.scl\n")
    f.write("Petersen 五行阴阳 15-tone scale (F0=220Hz, dtheta=5deg)\n")
    f.write("15\n")
```

```
for r in rows:
    cents_val = r['cents']
    f.write(f"{cents_val}\n")
print("Generated tools/15_positions.csv and tools/petersen_15.scl")
```

13 结语

本升级版补充了工程实现所需的参数表、导出实践、最小脚本与听觉测试流程。下一步建议：

1. 在 tools 中运行脚本并验证 15/45 导出；
2. 在 DAW（支持 MPE 或 .tun）中做对照听测；
3. 启动参数扫描并记录结果。