常见琵琶技法的计算建模

戴舒琪， 夏光宇

北京大学 信息科学技术学院，北京 100871

上海纽约大学 工程与计算机科学学院，上海 200122

摘 要： 民族乐器电子化目前仍处于起步阶段，绝大多数的民乐合成器，即便有着丰富的乐器采样库，也因缺乏基本的技法控制模型而无法在计算机上演奏真实流畅的乐曲。本文以琵琶为例，对轮指、扫弦、推拉弦揉弦等主要技法，及其衍生或叠加而成的常见琵琶技法建立计算模型，同时加入泛音技法的采样，采用 MIDI 格式编码，analysis by synthesis 的研究方法调整参数，最后使用 GarageBand 中的琵琶合成器输出合成结果。经过对比实验，本文的技法模型在真实性和流畅性上都取得了显著效果，对民乐电子化及其计算机自动演奏有着重要意义。

关键词： 民乐；琵琶；技法模型；电子合成；MIDI 编程.

1 介绍

随着计算机音乐领域的发展，越来越多的乐器开始了电子化之路。在电子合成器的帮助下，人们可以在计算机和智能移动设备的软件上演奏乐器。然而，相比于钢琴、小提琴和小号这些西方乐器，我国民族音乐中的琵琶、古筝等弹拨乐器有着一套自己独特的、更为复杂的技法体系，即使有着出色的合成器音色，也难以在缺乏技法模型的情况下，于计算机上演奏出流畅而真实的乐曲。以素有“民乐之王”美誉的琵琶为例，它有着所有民族乐器中最为复杂的技法体系，即便是简单的琵琶曲目，除了弹挑、轮指这样的基本技法外，也会用到扫弦、拉弦、揉弦、泛音等一系列中级难度的技法。这不仅需要计算机理解掌握这些常见技法，还要学习模仿人的演奏习惯，才能“以假乱真”。

另一方面，智能化的计算机自动演奏，近年来日益瞩目。以钢琴为例，在计算机中输入一段钢琴乐谱，则能自动合成出相应的钢琴音乐。而对于琵琶这样的中国民族乐器而言，要想实现智能化自动演奏，必须要有更为精确而智能的技法控制模型。例如，想要在计算机上生成出一段由轮指弹奏的指定节奏下的 1/4 音符，则需要精确计算出轮指的时长，并根据时长限制和乐曲风格等智能控制轮指的速度。同样，琵琶的不同技法模型在叠加和衔接时也需要更为智能的控制，譬如，如何在左手揉弦的同时右手进行长轮，将扫弦紧接轮指从而形成扫轮等等。

不论是通过计算机和移动设备的软件演奏琵琶，还是智能化的计算机自动演奏，琵琶常见技法的建模控制都必不可少，其模型水准更是决定演奏效果的关键所在。由琵琶出发，迈出民族乐器技法智能电子化的第一步，将来更容易推广至古筝、阮、柳琴、三弦等技法相似的民族弹拨乐器，对于西方的吉他等弹拨乐器的技法电子化，也有着重要的借鉴意义。

本文对琵琶的轮指、扫弦、推拉弦揉弦等主要技法，及其衍生或叠加而成的常见技法进行计算建模，并加入了泛音技法的采样，输入乐谱音符序列和对应的技法标示后，程序将自动生成曲目的 MIDI 格式编码文件，最后使用 GarageBand 中的琵琶合成器对 MIDI 文件输出合成结果。在实验阶段，采用 analysis by synthesis 方法进行实验，最终确定各个模型的参数，并与 GarageBand ios 版琵琶的演奏进行对比，取得了显著效果。本文第二章介绍国内外的研究现状，第三章将详细描述技法模型，第四章叙述实验分析与对比结果，最后一章进行总结展望。

2 国内外研究现状

MIDI 键盘最初用来控制钢琴，而对于钢琴，并不需要复杂的技法控制，甚至根本不必对技法建模，就可以在计算机上演奏大部分乐曲。对吉他而言，它与琵琶在构造和技法上都有相似之处，然而目前的学术研究和业界软件中也只有基本的弹拨、推拉弦和扫弦技法，并且，没有根据乐曲要求来自动智能调节参数的模型。

在苹果公司 2016 年 5 月发布的新版 GarageBand ios 版中，加入了采样制作的琵琶合成器，并添加了弹、轮指、推拉弦和扫弦的技法，这也是当前唯一的琵琶技法模型。然而，这些技法模型有着致命缺陷：（1）技法太少，无法满足绝大多数琵琶乐曲的需求；（2）模型过于简单，导致演奏效果和人的真实演奏差距非常大，例如其中的轮指模型完全采用均匀分布，不符合人的演奏实际，且无法区分轮指和夹弹、滚奏、摇指之间的区别；（3）无法根据乐曲要求，智能精确地控制技法，衔接操作难度大，难以流畅演奏，例如无法控制轮指个数（琵琶曲目中常常有精确的五指轮或四指轮要求）， 更无法精确限制轮指的节奏与时长等。由于 GarageBand 的琵琶采样合成器的音色目前仍属业界最佳，尤其是推拉弦的音色，于是本文选择该合成器进行建模后的音色合成。

在声学方面，文献[1]针对琵琶的 11 种左手技法，进行了时长、能量、频谱的声学参数分析，这对本文 3.4 节中的推拉弦和揉弦模型的构建有所启发。除此之外，其他民族弹拨乐器，尚且没有任何从计算机或声学角度进行技法建模或分析的经验可以借鉴。

3 技法计算模型

琵琶的左右手各有约 30 种技法（文献[2]中附录三与四分别记录了琵琶左右手共 68 种技法）,两手互相组合，变化繁多。从计算机演奏的角度来看，其中的基本技法（即独立且有与其他技法明显不同的合成单位）有右手的弹挑、轮指、扫弦与左手的推拉弦、揉弦、泛音。大类中又有细分，左右手组合、不同技法衔接又衍生出新技法。本文的技法模型，以单音“弹”为最小的基本单元，四种基本技法模型为基础，涵盖了大部分琵琶曲目所需的技法。

为了更接近人的真实演奏效果和并实现精准控制，一方面,根据琵琶演奏经验和相关技法发声原理进行编程建模，输出结果为 MIDI 编码格式的文件；另一方面，借鉴文献[3]中提出的 analysis-by-synthesis 方法，根据合成实验的结果不断调整参数，验证并修改模型。

3.1 轮指模型

轮指是琵琶演奏中最为常见的技法，其中又可以细分为长轮、三指轮、四指轮、五指轮、轮双等等。现有的轮指模型使用各指均匀分布的方式，与真实演奏差距大，且无法精确控制轮指个数与时长。对此，采用全新的计算模型，从时间控制（速度）和力度控制两个维度对常用的四指轮、五指轮及长轮进行建模[1]。

(1) 四指轮

又称“半轮”，食、中、名、小四指依次“弹”出，连得四声。首先是速度，对于四指轮来说，除了第四声的延长时间由乐谱中下一个指法的开始时间所控制外，其余三指的弹奏时长都有一定的规律分布。且根据经验，演奏速度较快或较慢时，往往各声之间的时长差别较小。如公式(1)中所示，T 为速度(Tempo, 单位：每分钟四分音符的个数 BPM)，D 为四指轮的总时长（单位：秒），L 表示最后一声的拍长(单位:四分音符个数)，α_i是控制前三声时长关系的参数(浮点数)，B 为其余三声的基准拍长。公式(2)所示，α_i是关于前三声总时长的二次函数。可以通过两种方式控制速度，一是调整轮指的总时长 D，时间越短，速度越快；二是直接调整每一声的基准拍长 B，B越小，则速度越快。

$$L + \sum_{i=1}^{3}(\alpha_i * B) = 60TD, \tag{1}$$

$$\alpha_i = k_{i1}(D - \frac{60L}{T})^2 + k_{i2}(D - \frac{60L}{T}) + k_{i3}, \quad 0 \le \alpha_i \le 1, \ i = 1,2,3 \tag{2}$$

其次是力度，如公式(3)所示，设每一声的力度为v_i，为 1 到 127 的正整数。V 是基准力度，β_i为浮点参数。

$$v_i = \beta_i * V, \quad i = 1,2,3,4; \ 0 \le \beta_i \le 1 \tag{3}$$

(2) 五指轮

简称“轮”，食、中、名、小四指依次“弹”出，大指“挑”进，连得五声。其建模公式与四指轮基本相同，将公式(1)(2)中 i 的取值范围增加到 4,(3)中 i 的取值增加到 5 即可。

(3) 长轮

两个以上的“轮”连接在一起称为“长轮”。在人演奏长轮时，无法精确保证每个轮的速度和力度都相同，为了更接近真实演奏效果，在长轮模型中加入噪音参数n_t与n_v，分别用来模拟速度和力度的偏差。同时，将相邻轮之间的间隔也将加入考量。另外，由于一段长轮中连续轮的个数不定，无法由总时长确定其轮指速度，因此，加入 K 作为每个轮的拍长参数用于调节轮的速度，N 为轮的个数。模型公式如下。

$$NK = 60TD \tag{4}$$

$$K = \sum_{i=1}^{5}(\alpha_i * B + n_t), \quad 0 \le \alpha_i \le 1 \tag{5}$$

$$v_i = \beta_i * V + n_v, \quad i = 1,2,3,4,5; \ 0 \le \beta_i \le 1 \tag{6}$$

---

[1] 轮指有“上出轮”（即按照食指、中指、无名指、小指顺序弹奏）和“下出轮”（从小指、无名指、中指、食指顺序演奏）之分，目前在实际演奏中，下出轮几乎很少出现，因此本文只讨论上出轮。

3.2 扫弦模型

扫弦与弹的区别在于，食指自右向左弹出后，同时要弹多根弦，如得一声。由此，扫又可分为双弹（两根弦）、小扫（三根弦）、扫（四根弦）、拂（四根弦，自左向右）、琶音（慢速，得四声）、连续扫拂等等。[2]

根据经验和测评，扫弦时四声的间隔几乎没有差异，可认为是平均分布。公式(7)表示时间控制，与公式(1)类似，只是将α_i变为常数 1，L 表示最后一声弦音的拍长，B 表示其余三根弦的基准拍长，T 为速度（每分钟四分之一拍的个数），D 是以秒为单位的总时长。对于大多数扫弦（爬音除外），L 的值会远远大于 B，以获得四弦一声的效果。

$$L + 3B = 60TD \tag{7}$$

在力度上，由于琵琶四根弦的构造不同，用食指扫弦时各弦所发出的音量也有所区别，但一般来说，在同一把琴上四根弦的力度比例可近似认为是恒定的。其公式模型与公式(3)相同。

3.3 推拉弦与揉弦模型

(1) 推拉弦

推、拉弦分别是，左手按弦向右推进或向左拉出，使得弦音升高的指法。两者在拉高弦音的效果上并没有明显差异，由文献[1]的实验结果，也可从声学角度得知二者几乎没有差别。现有的推拉弦模型，无法精确控制速度和音高，本文的模型则能够很好地解决这些问题。

建模由两个维度进行控制，分别是推拉弦的速度和目标音高。在调整音高过程中，采用 MIDI 命令 Pitch Bend，即音符弯曲。如公式(8)与(9)所示，T 与 D 的定义与上文其他模型相同，L1 和 L2 分别是从弦发声开始时到推拉弦音高开始变化的所用拍长，与推拉弦音高回到原来音位后距离该音结束的所剩拍长。B 为常数参数，表示每次调用 Pitch Bend 命令的拍长间隔。mid 为从弦发声开始到拉至目标音高所需的总拍长。N 为 Pitch Bend 命令的调用次数。t 为关于 i 的线性函数，表示第 i 次调用 Pitch Bend 时所处拍数时间。n(t)是在拍数为 t 时，Pitch Bend 命令中的调用值，取值范围是位于[-63,63]区间内的整数（与 GarageBand 中琵琶采样合成器中 Pitch Bend 命令取值范围相同）。k 表示推拉弦速度,k 越大，推拉弦速度越慢，γ为控制推拉弦速度、生成 k 的常数参数。

令 $mid = 30TD + \frac{L1-L2}{2}$, $N = \frac{30TD - \frac{L1+L2}{2}}{B}$，$k = \gamma D/\left(note_{target} - note_{origin}\right)$，

$$t(i) = L1 + B * i,\ 1 \le i \le N \tag{8}$$

$$n\,(t) = \begin{cases} \left\lfloor \left(note_{target} - note_{origin}\right) * log_k(t - L1 + 1) \right\rfloor,\ L1 \le t \le mid \\ \left\lfloor \left(note_{target} - note_{origin}\right) * log_k(60TD - L2 - t + 1) \right\rfloor, mid \le t \le 60TD - L2 \end{cases} \tag{9}$$

---

[2] 本文讨论的扫弦模型，在应用于从低音到高音，或从高音到低音的扫弦时，可以通过调整音高和力度参数进行选择。

(2) 揉弦

揉弦指的是左手按弦在音位上作规律的小幅度摇动，发出波音效果，本文的揉弦模型是市面上迄今为止第一个揉弦模型。与推拉弦模型的不同之处在于，揉弦没有必须拉到某一指定音高的要求，只需小幅度的音高变化，且绝大多数情况下，揉弦的速度没有明显区别，只需按某一速度进行循环往复地摇动即可。如公式(10)所示，揉弦中一个推拉摇动所用拍长为常数参数 Dx，N 为重复个数。每一个小幅度推拉摇动的公式即为推拉弦的公式（8）和（9），目标音高需设置在较小范围内。

$$N * Dx = 60TD \tag{10}$$

3.4 泛音模型

泛音广泛存在于弦乐器中，这里讨论的是琵琶的自然泛音,其原理与吉他的自然泛音类似。根据文献[4]，当左手虚按在琴弦上（仅仅接触琴弦但不用力），琴弦的振动将受到限制，且只有以此触点为节点的振动才被保留下来。如果触点在琴弦的 1/2 处，那么泛音频率就是琴弦基频的 2 倍，如果触点在琴弦的 1/3 或 2/3 处，那么泛音频率就是琴弦基频的 3 倍，依此类推。由于目前市面上的琵琶合成器均没有泛音的合成选项，MIDI 命令也无法直接控制滤波器，因此，本文计算出琵琶每根弦各个泛音的频率，再根据计算结果使用滤波器在泛音所对应音高的弦音上进行滤波。

3.5 模型叠加与衔接

在琵琶的常见技法中，绝大多数都是由上述四种基本技法叠加、衔接衍生而来。而目前市面上还没有考虑这些衍生技法的模型。在此，分别以左右手配合的揉弦轮指，以及右手扫弦与轮指衔接的“扫带轮“两种最典型的技法为例，描述衍生技法的建模。

(1) 揉弦与轮指

在右手轮指的同时左手进行揉弦，是琵琶演奏中极为常用的技法。其组合建模的方式为，将公式(10)中的揉弦总时长 N*Dx 设为轮指的每一声时长，即公式(1)中的$\alpha_i * B$与 L；将普通轮指调用若干次“弹”换为调用若干次“揉弦”；同时，揉弦公式中的 L1 与 L2 参数需调至更小的值。

(2) 扫带轮

“轮”时首先用食指作“扫”，然后中、名、小、大指紧跟作轮单弦。联立五指轮和扫弦的时长公式，可得，

$$L + 3B_1 + \sum_{i=1}^{5}(\alpha_i * B_2) = 60TD \tag{11}$$

在力度方面，虽然扫弦的力度通常比轮指要大，但由于两者的力度公式形式相同，如公式(3),可直接沿用。

4 实验

除泛音模型直接使用 GarageBand 中的滤波器在 MIDI 音符上进行实验外，其余模型的实

验，均是先编程输出 MIDI 格式的编码文件，再由 GarageBand 琵琶采样合成器生成音频，并采用文献[3]中的 Analysis by synthesis(A-by-S)方法，与真实的演奏片段进行比对，分析差距，调节参数，如此循环往复。

在实验的参数调节中，除了依据专业的演奏经验和发音原理外，我们邀请了琵琶演奏家、琵琶爱好者、以及无琵琶背景的普通音乐爱好者，分别进行聆听，将他们的反馈意见应用到实验的 Analysis by synthesis(A-by-S)方法中。下面将逐一介绍四种指法模型的实验情况。

4.1 轮指实验

根据轮指的速度公式(1)(2)，有两种方式控制轮指速度。一是调整轮指的总时长 D，时间越短，速度越快； 二是直接调整每一声的基准拍长 B，B越小，则速度越快。L 作为最后一个音的总拍数，在轮指后紧接其他指法的情形下，由乐曲及下一个指法的要求来确定 L 的具体数值（通常较短）；在没有明确要求时，则 L 为琵琶弦音的自然延长拍数，根据经验，自然延长的时间约为 2 秒左右。

根据实验测试和调整，时间控制公式中的参数$\mathrm{k_i}$及力度控制公式中β_i的估计取值如表 1 所示，如此，既能保证每一声的间隔时长在轮速较快或较慢时差异更小，也能保持最佳的时长比例。长轮中，速度噪音n_t与力度噪音n_v分别为基准拍长 B 和基准力度 V 上下 5%范围内的随机数，从而更接近真实演奏。

实验中发现，即便在力度和速度分布上与人的演奏接近，本文的轮指模型在音乐性上显得过于机械。于是，进一步在长轮模型中加入了一些常见的音乐效果，如，通过在轮指各声的音量 V 上，按时序增加线性渐强或渐弱的包络，来控制整体的渐强与渐弱。

表 1　轮指模型参数的估计结果

Tab.1 Estimation results of parameters in Rotating Fingering Model

i	$\mathrm{k_{i1}}$	$\mathrm{k_{i2}}$	$\mathrm{k_{i3}}$	β_i
1	0.088	-0.169	0.955	1.00
2	0.170	-0.337	0.909	0.95
3	0.193	-0.385	0.945	0.89
4	0.031	0.054	0.979	0.87
5	0.000	0.000	1.000	1.00

注：在四指轮和五指轮模型中，只使用其公式中 i 取值范围以内的值。

4.2 扫弦实验

在扫弦的模型公式(7)中，L 值与轮指模型中的 L 相同，同时，与轮指模型类似，也通过控制时长 D 或直接调整 B 来控制扫弦速度，具体力度参数实验估计结果见表 2。

表 2　扫弦模型力度参数的估计结果

Tab.1 Estimation results of velocity parameters in Sweep Model

β_1	β_2	β_3	β_4
1	0.92	0.9	0.85

注：“拂”指法中，生成顺序为从β_4到β_1。

4.3 推拉弦与揉弦实验

推拉弦模型中的 L2 的取值参照轮指和扫弦模型的 L 参数，其余参数的实验计算估值如表 3 所示。模型中 target_note 与 origin_note 的取值分别为目标音高和原始音高的 MIDI 音高值，在揉弦模型中 target_note 默认为 origin_note 值加一。

表 3　推拉弦与揉弦模型参数的估计结果

Tab.1 Estimation results of parameters in bend and vibrato Model

	Dx	L1/T	L2/T	B/T	γ
推拉弦	-	0.3	-	0.0078	e
揉弦	0.13	0.02	0.02	0.005	e

注：L1/T, L2/T,B/T 的单位都是秒(s), e 为自然常数.

4.4 泛音实验

实验发现，计算频率并滤波的方法合成出的泛音，与真实的琵琶自然泛音相比，仍有较大的差距。由于琵琶每根弦的自然泛音数目很少，完全可以通过采样录音的方式将其记录下来，再封装出函数调用接口，加入技法模型中，由程序在合成音频时进行调用。

为了检验模型最终的效果，本文从琵琶传统古曲《高山流水》（浦东派演奏家，林石城编曲版）中改编出 7 小节的选段，涵盖拧弦、轮指、扫弦、泛音、揉弦、推拉弦等技法，分别用 GarageBand ios 版中的琵琶和真实琵琶进行演奏，并与本文中模型的合成结果进行对比分析。对于 GarageBand 琵琶中无法演奏的技法，则用简化版或相近技法代替，例如，用音高相同或相差一个八度的弦音代替泛音、不使用揉弦、扫轮拆分为扫弦和轮指等等。检验结果表明，本文中的模型不仅大大提高了计算机演奏的质量，与人的真实演奏更为相似；而且相比 GarageBand 中需要花费大量时间练习操作、错误率高、无法精准控制演奏的事实，本文中的模型生成简单便捷，可操作性强。相关合成结果见：http://shuqid.net/pipa。

5　总结与展望

第四章中的实验结果表明，本文所提出的琵琶常用技法计算模型，不论是在指法的种类数目，与人演奏的相似度，还是在演奏控制上都有着质的飞跃，可以实现大部分中等难度琵琶曲目的电子化合成，控制自由准确，生成便捷，并且保证了相当的演奏质量。同时，采用 MIDI 编码格式也保证了该模型的通用性和可扩充性。

针对模型中存在的问题，还有更多的改进空间：

(1) 琵琶指法组合千变万化，在现有模型基础上，增加更多的高级指法模型，包括更多的“噪

音”技法，如“摘”、“拍板”、“绞线”等（倘若难以实现物理模型，也可以直接使用类似本文“泛音”技法的采样方法来加入，毕竟这些技法没有音高，音色种类较为固定），则可以演奏更多的琵琶乐曲。

(2) 在建模时采用机器学习方法来假设和验证分布，或许可以提高模型的精确性。例如，公式(2)中α_i的二次函数的参数可以通过回归模型进行解决，或假设为其他分布类型；甚至可以不假设分布，采用生成对抗网络(Generative Adversarial Network，文献[5])，在建立生成指法的网络同时，再设计一个判别网络专门用来判断该指法是否由人演奏。

(3) 增加模型参数，采用一些机器学习中的监督学习算法，从合成的指法模型音频出发，以真实演奏的音频信号波形为训练样本，从而更准确地分析调整各个指法模型。

(4) 尝试学习同一指法在不同曲目中的音频样本，或许能够得到不同演奏家在不同演奏风格下的指法模型，进一步学习某个具体演奏家的演奏习惯。

(5) 可以适当加入一些触弦所产生的噪音，更加真实地模拟人的演奏效果。

(6) 设计模型的图形化操作界面，让非开发者用户也能自如演奏。

未来，若能够创建琵琶指法符号的电子识别系统，结合本文中的指法模型，将真正实现计算机上的琵琶智能读谱及自动演奏。琵琶中级指法智能模型，作为民族乐器智能电子化的第一步，不仅为其他民族乐器，也为部分西方乐器（如吉他）的指法电子化提供了很好的借鉴；对中西方乐器融入未来信息世界，特别是对于建立中国民族音乐的独特体系，有着重要而深远的意义。

参考文献

[1] 高珊，于洪志，徐慧. *琵琶左手指法演奏音色的声学浅析*, 第 11 届全国人机语音通讯学术会议, 2011.10.

[2] 中国民族管弦乐学会全国民族乐器演奏（业余）考级委员会琵琶专家委员会，*琵琶曲集*，北岳文艺出版社，1998.8 第 1 版.

[3] Halle, Morris and Kenneth N. Stevens (1962). *Speech Recognition: A Model and a Program for Research*. IRE Transactions of the PGIT IT-8: 155-159.

[4] Kelvin Bradley. *Synthesis of an Acoustic Guitar with a Digital String Model and Linear Prediction*, Phd Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1995

[5] Ian Goodfellow, Jean Pouget-Abadie,et.al. *Generative Adversarial Nets*, Advances in neural information processing systems, 2014

Computational models for common Pipa techniques based on a commercial sample library

DAI Shu-qi, Xia Guang-yu

(*School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China*

Department of Computer Science and Engineering, NYU Shanghai, Shanghai, 200122, China)

Abstract: The digitalization of Chinese traditional instruments is still in its infancy. Most synthesizers, though equipped with rich sample libraries, lack basic control models to intimate common performance techniques. In this paper, taking Pipa as an example, we have developed a series of computational models for common performance techniques by an "analysis-by-synthesis" method, leading to more realistic synthesized Pipa performances.

Keywords: Chinese Traditional Instrument; Pipa; Performance Technique Model; Digitalization; Synthesis; MIDI Programming.