



(12) 发明专利申请

(10) 申请公布号 CN 119068741 A

(43) 申请公布日 2024. 12. 03

(21) 申请号 202411444959.4

(22) 申请日 2021.11.23

(30) 优先权数据

2020-203590 2020.12.08 JP

(62) 分案原申请数据

202111398182.9 2021.11.23

(71) 申请人 卡西欧计算机株式会社

地址 日本

(72) 发明人 石冈洋介 石冈雪奈

(74) 专利代理机构 中科专利商标代理有限责任

公司 11021

专利代理师 李逸雪

(51) Int. Cl.

G09B 15/00 (2006.01)

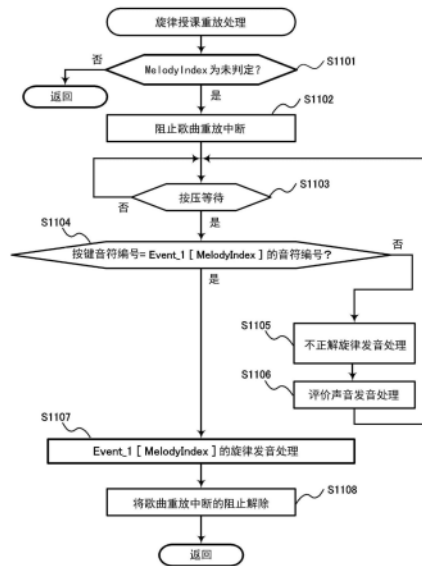
权利要求书1页 说明书15页 附图15页

(54) 发明名称

演奏装置、演奏授课方法以及程序

(57) 摘要

本发明提供演奏装置、演奏授课方法以及程序。演奏授课装置具备：多个键；检测所述键的操作的检测部；和控制部，所述控制部执行：判定处理，按照乐曲的和弦的进展来引导与所述和弦对应的键，基于所述和弦的构成音与被所述检测部检测到按压的键的关系来判定所述按压是否是正解；和发音处理，在所述判定处理的判定结果成为不正解的情况下，进行与所述不正解对应的发音。



1. 一种演奏装置,其特征在于,具备:
多个键;
检测部,检测所述键的操作;和
控制部,按照乐曲的和弦的进展,基于所述和弦的构成音与被所述检测部检测到按压的键的关系来判定所述按压是否是正解,
所述控制部在通过所述检测部检测到按压的情况下,执行关于所述按压是否是正解的判定和无论所述按压是否是正解都以被所述检测部检测到按压的键的音高来发出乐音。
2. 根据权利要求1所述的演奏装置,其特征在于,
在所述按压成为正解的情况下,执行在将发音中的所述乐音消音后发出所述和弦的构成音的发音处理。
3. 根据权利要求1所述的演奏装置,其特征在于,
所述控制部在所述按压成为不正解的情况下,执行将发音中的所述乐音消音的发音处理。
4. 根据权利要求1所述的演奏装置,其特征在于,
所述控制部在所述按压成为不正解的情况下,进一步执行发出与所述不正解的程度相应的评价声音的处理。
5. 根据权利要求1至4中任一项所述的演奏装置,其特征在于,
被所述检测部检测到按压的键是位于和弦键域的键。
6. 根据权利要求1至4中任一项所述的演奏装置,其特征在于,
以被所述检测部检测到按压的键的音高发出的乐音和所述和弦的构成音不同。
7. 根据权利要求1至4中任一项所述的演奏装置,其特征在于,
所述控制部在通过所述检测部检测到按压的情况下,以被所述检测部检测到按压的键的音高发出乐音之后,执行关于所述按压是否是正解的判定。
8. 一种演奏授课方法,其特征在于,
执行如下处理:
发音处理,发出乐音;和
判定处理,按照乐曲的和弦的进展,基于所述和弦的构成音与被检测部检测到按压的键的关系来判定所述按压是否是正解,
在所述发音处理中,在通过所述检测部检测到按压的情况下,执行无论所述按压是否是正解都以被所述检测部检测到按压的键的音高来发出乐音。
9. 一种程序,其特征在于,
用于使计算机执行如下处理:
发音处理,发出乐音;和
判定处理,按照乐曲的和弦的进展,基于所述和弦的构成音与被检测部检测到按压的键的关系来判定所述按压是否是正解,
在所述发音处理中,在通过所述检测部检测到按压的情况下,执行无论所述按压是否是正解都以被所述检测部检测到按压的键的音高来发出乐音。

演奏装置、演奏授课方法以及程序

[0001] 本申请是申请号为“202111398182.9”，申请日为2021年11月23日，发明名称为“演奏授课装置、方法以及存储介质”之申请的分案申请。

技术领域

[0002] 本公开涉及演奏者通过基于被引导的按压指示信息进行按压操作来接受演奏授课的演奏授课装置、方法以及存储介质。

背景技术

[0003] 如下技术广为人知：关于乐曲的例如和弦声部、旋律声部引导应进行按压的键、演奏者按照该引导进行了正确的按压操作，在该情况下，通过发出乐曲的和弦声部的和弦音、旋律声部的旋律音，来接受演奏授课。

[0004] 并且，过去，还已知如下技术：在进行授课的声部，在直到通过引导决定了被引导的应进行按压的键的按压定时为止都未由演奏者正确进行按压的情况下，通过使引导的进展停止，使得演奏者不再推进授课（例如专利第2707853号公报）。

[0005] 进而还已知如下技术：即使由演奏用的声部的演奏数据指定的演奏操作件到给定的定时为止都未进行演奏操作，从而将遵循演奏数据的乐曲的进展的读出中断，也不会发生由于继续进行一部分乐音信号的产生而吓到演奏者、或误以为在电子乐器中发生异常的情况（例如特开2004-101979号公报）。

发明内容

[0006] 方案的一例的演奏装置具备：多个键；检测部，检测键的操作；和控制部，按照乐曲的和弦的进展，基于和弦的构成音与被检测部检测到按压的键的关系来判定按压是否是正解，控制部在通过检测部检测到按压的情况下，执行无论按压是否是正解都以被检测部检测到按压的键的音高来发出乐音和关于按压是否是正解的判定。

附图说明

- [0007] 图1是实施方式的电子键盘乐器的外观图。
- [0008] 图2是表示实施方式的电子键盘乐器系统的框图。
- [0009] 图3是表示乐曲数据的数据例的数据结构图。
- [0010] 图4是表示实施方式的主处理的示例的流程图。
- [0011] 图5A是表示初始化处理的详细例的流程图。
- [0012] 图5B是表示拍速变更处理的详细例的流程图。
- [0013] 图5C是表示歌曲开始处理的详细例的流程图。
- [0014] 图6是表示开关处理的详细例的流程图。
- [0015] 图7是表示键处理的详细例的流程图。
- [0016] 图8是表示歌曲重放中断处理的详细例的流程图（其一）。

- [0017] 图9是表示歌曲重放中断处理的详细例的流程图(其二)。
- [0018] 图10是表示乐曲重放处理的详细例的流程图。
- [0019] 图11是表示旋律授课重放处理的详细例的流程图。
- [0020] 图12是表示和弦授课重放处理的第1实施方式的详细例的流程图。
- [0021] 图13是表示和弦授课重放处理的第2实施方式的详细例的流程图。
- [0022] 图14是表示和弦授课重放处理的第3实施方式的详细例的流程图。

具体实施方式

[0023] 以下参考附图来详细说明用于实施本公开的方式。图1是实施方式的电子键盘乐器100的外观图,图2是表示实施方式的电子键盘乐器100的系统200的结构例的图。

[0024] 如图1所示那样,本实施方式所涉及的电子键盘乐器100具备具有多个键101的键盘110、LCD显示部102和操作部103。此外,各键101在内部具备LED(Light Emitting Diode:发光二极管)等发光元件,能使各键101的表面发光。

[0025] LCD显示部102进行与操作部103的操作相伴的消息的显示、用于后述的授课模式的选择的显示等。

[0026] 操作部103具有演奏者进行各种设定等的操作按钮和切换电子键盘乐器100的电源的接通/断开的电源开关,操作按钮用于进行演奏模式的选择、音量调整等各种设定操作等。

[0027] 作为演奏模式,有通常演奏模式、自动演奏模式、授课模式(右声部授课模式、左声部授课模式、左右声部授课模式)、授课模式中的和弦输入方法等设定。

[0028] 通常演奏模式是演奏者不依赖于和弦进展等地进行通常的乐器演奏的模式。

[0029] 自动演奏模式是按照从ROM202依次读出的乐曲数据、没有演奏者的参与地进行自动演奏的模式。

[0030] 授课模式是按照从ROM202依次读出的乐曲数据使应演奏的键盘110上的各键101发光来进行引导的同时由演奏者接受演奏授课的模式。右声部授课模式是通过在主要用右手演奏的旋律键域点亮键101的LED来进行旋律引导、由演奏者依次追随光来对发光的键101进行按压的同时接受演奏授课的模式。左声部授课模式是通过在主要用左手演奏的和弦键域使构成和弦音的多个(3或4个)键101的LED闪烁(根音的情况)或点亮(这以外的和弦构成音的情况)来进行和弦引导、由演奏者依次追随发光的多个键101并同时按压的同时接受演奏授课的模式。左右声部授课模式是能同时进行右声部授课模式下的旋律授课和左声部授课模式下的和弦授课的模式。

[0031] 接下来,在图2所示的电子键盘乐器系统200中,CPU(中央运算处理装置)201、读取专用的存储器即ROM(只读存储器)202、能读写的存储器即RAM(随机存取存储器)203、检测部204、LED控制器206、LCD控制器207、开关接口208和音源LSI210通过系统总线213相互连接。

[0032] 开关接口208检测操作部103中的开关设定状态,通知给CPU201。

[0033] ROM202存储CPU201载入到RAM203并执行的控制程序、前述的自动演奏模式、授课模式中的乐曲数据。

[0034] RAM203存储从ROM202转发的控制程序、乐曲数据,此外存储程序执行的各种变量

数据等。

[0035] CPU201按照从ROM202载入到RAM203的控制程序来进行相音源LSI210的乐音的发音/消音指示、用于自动演奏或演奏授课的各种控制。该控制动作的详细使用图3的数据结构图和图4到图14的流程图之后叙述。

[0036] 计时器209在后述的歌曲重放中断处理中使用。

[0037] 图2的LED205与各个键101(白键或黑键)的操作件对应而设,各操作件的与LED205对置的部分能透过LED205的光。LED205在授课模式时,在接下来应按压的键101闪烁(和弦的根音的情况)或点亮(旋律音或根音以外的和弦的构成音的情况)。

[0038] 图2的LED控制器206基于来自CPU201的指示来进行各键101所具备的各LED205的闪烁或点亮以及熄灭。

[0039] 音源LSI (LSI:大规模集成电路) 210按照来自CPU201的指示产生乐音数据。

[0040] D/A变换部211将音源LSI210所产生的数字的乐音数据变换成模拟乐音信号。将该模拟乐音信号经由放大器212从扬声器213放音。

[0041] 对具有以上的结构的本实施方式的概略动作进行说明。图2的CPU201通过执行载入到RAM203的控制程序,来将演奏者选曲且从ROM202载入到RAM203的乐曲数据依次读出并执行,由此,按照乐曲中的各个和弦的进展,将与和弦的构成音对应的键101用例如LED205进行光引导,同时基于和弦的构成音与检测部204中检测到按压的1个以上的键101的按键音的关系来执行判定按压是否是正解的判定处理。然后,CPU201在判定处理的判定结果成为不正解的情况下,对音源LSI210发行指示与不正解对应的发音的发音控制数据,音源LSI210执行对应的发音处理。CPU201在判定处理的判定结果成为正解的情况下,对音源LSI210发行将和弦的构成音同时发音的多个发音控制数据,音源LSI210对和弦音进行发音处理。由此,在演奏者例如在左声部授课模式下,在将和弦判定不是正解的键101按压的情况下,演奏者能认识到不正解,从而不会给演奏者带来不协调感、故障等的误解。

[0042] 作为一个实施方式,CPU201在上述的发音处理中,例如将与不正解对应的发音经由音源LSI210以与按压对应地决定的不正解和弦的静止和弦(Static Chord)进行发音处理。在此,所谓静止和弦,是指不管哪个键被按压,在正按压的期间都以专用的音色持续地鸣响所决定的和弦的构成音的全部的和弦。该处理的详细,使用图12的流程图作为和弦授课重放处理的第1实施方式进行说明。由此,演奏者能在和弦判定为不正解的情况下,自然地根据静止和弦认识到不正解。

[0043] 作为其他一个实施方式,CPU201在上述的发音处理中,在检测部204中检测到按压的1个以上的键101所表示的各个音高未含在按照乐曲中的各个和弦的进展而决定的正解和弦的构成音中的情况下,将与检测到按压的1个以上的键101所表示的各个音高对应的乐音经由音源LSI210进行发音处理。由此,演奏者能认识到按压了不是正解和弦的构成音的和弦键域的乐音的键101。

[0044] 作为再其他一个实施方式,CPU201在上述的发音处理中将与检测部204中各自被检测到按压的键101所表示的各音高对应的乐音经由音源LSI210进行发音处理,并且在判定处理的判定结果成为正解的情况下,在将发音中的乐音消音后,发出正解和弦的各构成音。由此,能将不正解时的不需要的发音消音,并授课演奏正解的和弦构成音,能防止演奏者感到不快感。

[0045] 在此,CPU201也可以在判定处理的判定结果成为不正解的情况下,经由音源LSI210进一步执行与不正解的程度相应的评价声音、例如“还有差距!”、“可惜了!”,或者在完全不行的情况下,经由音源LSI210进一步执行发出爆炸音等的评价声音的发音处理。由此,演奏者能愉快地接受演奏授课。

[0046] 此外,也可以除了键101的LED205的光引导以外,还在图1的LCD显示部102显示与键盘110相同的图像,将与和弦的构成音对应的多个键101各自作为引导显示进行显示指示。由此,演奏者除了能用发光的键101以外,还能用LCD显示部102来确认授课引导。

[0047] 此外,也可以是,除了键101的LED205的光引导以外,还将用于将与和弦的构成音对应的多个键101各自进行按压的指编号经由音源LSI210作为引导声音进行指示发音。由此,演奏者除了能根据发光的键101以外,还能根据引导声音来确认指法。

[0048] 此外,也可以是,除了键101的LED205的光引导以外,还将与和弦的构成音对应的多个键101各自所对应的音符音名经由音源LSI210作为引导声音进行指示发音。由此,演奏者除了能根据发光的键101以外,还能根据引导声音来确认应按压的键101。

[0049] 图3是表示在本实施方式中从图2的ROM202读入到RAM203的乐曲数据的数据结构例的图。该数据结构例遵循MIDI (Musical Instrument Digital Interface,乐器数字接口)用文件格式之一的标准MIDI文件的格式。该乐曲数据由被称作块(chunk)的数据块构成。具体地,乐曲数据由表示数据尺寸、格式信息的头部块、与其接续的存放旋律声部用的旋律演奏数据的音轨块1和存放和弦声部用的和弦演奏数据的音轨块2构成。

[0050] 音轨块1、2分别由ChunkID、ChunkSize、和DeltaTime_1[i]以及Event_1[i] (音轨块1/旋律声部的情况)或DeltaTime_2[i]以及Event_2[i] (音轨块2/和弦声部的情况)所构成的演奏数据组($0 \leq i \leq L$:音轨块1/旋律声部的情况, $0 \leq i \leq M$:音轨块2/和弦声部的情况)构成。

[0051] DeltaTime_1[i]是表示其近前的Event_1[i-1]的执行时刻起的等待时间(相对时间)的1~4字节的可变长度数据。同样地,DeltaTime_2[i]是表示其近前的Event_2[i-1]的执行时刻起的等待时间(相对时间)的1~4字节的可变长度数据。Event_1[i]是在音轨块1/旋律声部中指示旋律的发声定时和音高的元事件。Event_2[i]是在音轨块2/和弦声部中指示音符开(“否” te-On)或音符关(“否” te-Off)的MIDI事件、或指示拍子的元事件。通过对音轨块1/旋律声部在各演奏数据组DeltaTime_1[i]以及Event_1[i]中在其近前的Event_1[i-1]的执行时刻起等待DeltaTime_1[i]基础上执行Event_1[i],来实现旋律的发音进展。另一方面,通过对音轨块2/和弦声部在各演奏数据组DeltaTime_2[i]以及Event_2[i]中在其近前的Event_2[i-1]的执行时刻起等待DeltaTime_2[i]的基础上执行Event_2[i],来实现和弦的发音进展。

[0052] 图4是表示本实施方式中的电子乐器的控制处理例的主流程图。该控制处理例如在图2中是CPU201执行从ROM202载入到RAM203的控制处理程序的动作。

[0053] CPU201首先在执行初始化处理后(步骤S401),重复执行步骤S402到S407的一系列处理。

[0054] 在该重复处理中,CPU201首先执行开关处理(步骤S402)。在此,CPU201基于来自图2的开关接口208的中断,来执行与图1的操作部103中的开关操作对应的处理。对开关处理的详细例,使用图6的流程图之后叙述。

[0055] 接下来,CPU201执行键处理,基于来自图2的检测部204的中断来判定图1的键盘110的哪个键101被操作并进行处理(步骤S403)。在此,CPU201对应于演奏者所进行的任意的键的按键或离键的操作,来对图2的音源LSI204输出指示音符开(发音开始)或音符关(发音停止)的发音控制数据。关于键处理的详细例,使用图7的流程图之后叙述。

[0056] 接下来,CPU201执行显示处理,对应显示于图1的LCD显示部102的数据进行处理,将该数据经由图2的LCD控制器207显示到LCD显示部102(步骤S404)。作为显示于LCD显示部102的数据,例如有所演奏的授课模式等中使用的乐曲的乐谱、与键盘110对应的键盘图像上的各键101所对应的各键图像的授课引导显示、以及各种设定信息。

[0057] 接下来,CPU201执行乐曲重放处理(步骤S405)。在该处理中,执行旋律授课重放处理、和弦授课重放处理、自动演奏重放处理等。关于乐曲重放处理的详细例,使用图10的流程图之后叙述。

[0058] 接着,CPU201执行音源处理(步骤S406)。在音源处理中,CPU201执行音源LSI204中的发音中的乐音的包络控制等处理。

[0059] 最后,CPU201判定演奏者是否按下未特别图示的关机开关而进行了关机(步骤S407)。若步骤S407的判定成为“否”,CPU201就回到步骤S402的处理。若步骤S407的判定成为“是”,CPU201就结束图4的流程图中所示的控制处理,将电子键盘乐器100的电源切断。

[0060] 图5A、图5B、以及图5C分别是表示图4的步骤S401的初始化处理的详细例、图4的步骤S402的开关处理中的后述的图6的步骤S602的拍速变更处理的详细例、以及相同的图6的步骤S606的歌曲开始处理的详细例的流程图。

[0061] 首先,在表示图4的步骤S401的初始化处理的详细例的图5A中,CPU201执行TickTime的初始化处理。在本实施方式中,旋律的进展以及和弦的进展以TickTime这样的时间为单位来进展。被指定为图3的乐曲数据的头部块内的TimeDivision值的时基值表示4分音符的分辨率,若该值成为例如480,则4分音符具有480TickTime的时间长度。此外,图3的乐曲数据的音轨块内的等待时间DeltaTime_1[i]值以及DeltaTime_2[i]值也通过TickTime的时间单位进行计数。在此,1TickTime实际成为几秒,根据对乐曲数据指定的拍速而不同。现在,若将拍速值设为Tempo[节拍/分],将上述时基值设为TimeDivision,则TickTime的秒数通过下式算出。

[0062]
$$\text{TickTime}[\text{秒}] = 60 / \text{Tempo} / \text{TimeDivision} \quad (1)$$

[0063] 为此,在图5A的流程图所例示的初始化处理中,CPU201首先通过与上述(1)式对应的运算处理来算出TickTime[秒](步骤S501)。另外,拍速值Tempo在初始状态下在图2的ROM202中存储给定的值例如60[拍/秒]。或者,也可以在非易失性存储器中存储前次结束时的拍速值。

[0064] 接下来,CPU201对图2的计时器209设定基于步骤S501中算出的TickTime[秒]的计时器中断(步骤S502)。其结果,在计时器209中每当经过上述TickTime[秒],就对CPU201产生用于旋律进展以及和弦进展的中断(以下记载为“歌曲重放中断”)。因此,在基于该歌曲重放中断在CPU201中执行的歌曲重放中断处理(后述的图8以及图9)中,每1TickTime执行使旋律以及和弦进展的控制处理。

[0065] 接着,CPU201执行图2的RAM203的初始化等其他初始化处理(步骤S503)。之后,CPU201结束图5A的流程图所例示的图4的步骤S401的初始化处理。

[0066] 关于图5B以及图5C的流程图,之后叙述。

[0067] 图6是表示图4的步骤S402的开关处理的详细例的流程图。CPU201首先判定是否通过图1的操作部103内的拍速变更开关变更了旋律进展以及和弦进展的拍速(步骤S601)。若该判定成为“是”,CPU201就执行拍速变更处理(步骤S602)。该处理的详细使用图5B之后叙述。若步骤S601的判定成为“否”,CPU201就跳过步骤S602的处理。

[0068] 接下来,CPU201判定在操作部103中是否选择了某首歌曲(步骤S603)。若该判定成为“是”,CPU201就执行歌曲读入处理(步骤S604)。该处理是将具有图3中说明的数据构造的乐曲数据从图2的ROM202读入到RAM203的处理。这以后,对读入到RAM203的乐曲数据执行对图3例示的数据构造内的音轨块1或2的数据访问。若步骤S603的判定成为“否”,CPU201就跳过步骤S604的处理。

[0069] 接着,CPU201判定是否在图1的操作部103中操作了歌曲开始开关(步骤S605)。若该判定成为“是”,CPU201就执行歌曲开始处理(步骤S606)。该处理的详细使用图5C之后叙述。若步骤S605的判定成为“否”,CPU201就跳过步骤S606的处理。

[0070] 接着,CPU201判定是否在图1的操作部103中通过将授课模式开关以及右声部开关接通而选择了表示旋律授课的右声部授课(步骤S607)。若该判定成为“是”,CPU201就执行将右声部授课开启的处理(步骤S608)。具体地,CPU201将作为RAM203上的变量的右声部授课标记设定为开启(值“1”)。若步骤S607的判定成为“否”,CPU201就跳过步骤S608的处理。

[0071] 进而,CPU201判定是否在图1的操作部103中通过将授课模式开关以及左声部开关接通而选择了表示和弦授课的左声部授课(步骤S609)。若该判定成为“是”,CPU201就执行将左声部授课开启的处理(步骤S610)。具体地,CPU201将作为RAM203上的变量的左声部授课标记设定为开启(值“1”)。若步骤S609的判定成为“否”,CPU201就跳过步骤S610的处理。

[0072] 另外,与旋律授课对应的右声部授课和与和弦授课对应的左声部授课能同时开启。即,能同时进行右声部授课和左声部授课。

[0073] 最后,CPU201判定是否在图1的操作部103中操作了其他开关,执行与各开关操作对应的处理(步骤S611)。在此,探测例如演奏者在图1的操作部103中将自动演奏开关接通的状态和选择了所期望的自动演奏乐曲等,将作为RAM203上的变量的自动演奏标记设为开启(值“1”),进而将与所选择的自动演奏乐曲对应的图3中例示的数据结构的乐曲数据从ROM202转发到RAM203。之后,CPU201结束图6的流程图所例示的图4的步骤S402的开关处理。

[0074] 图5B是表示图6的步骤S602的拍速变更处理的详细例的流程图。如前述那样,若变更拍速值,则TickTime[秒]也变更。在图5B的流程图中,CPU201执行与该TickTime[秒]的变更相关的控制处理。

[0075] 首先,CPU201与图4的步骤S401的初始化处理中执行的图5A的步骤S501的情况同样地,通过与前述的(1)式对应的运算处理来算出TickTime[秒](步骤S511)。另外,拍速值Tempo将通过图1的操作部103内的拍速变更开关变更后的值存储到RAM203等。

[0076] 接下来,CPU201与图4的步骤S401的初始化处理中执行的图5A的步骤S502的情况同样地,对图2的计时器209设定基于步骤S511中算出的TickTime[秒]的计时器中断(步骤S512)。之后,CPU201结束图5B的流程图所例示的图6的步骤S602的拍速变更处理。

[0077] 图5C是表示图6的步骤S606的歌曲开始处理的详细例的流程图。

[0078] 首先,CPU201在自动演奏或授课模式的进展中,以TickTime为单位,将用于对近前

的事件的发生时刻起的相对时间进行计数的RAM203上的变量DeltaT_1(音轨块1)以及DeltaT_2(音轨块2)的值都初始设定为0。接下来,CPU201将用于指定图3中例示的乐曲数据的音轨块1内的演奏数据组DeltaTime_1[i]以及Event_1[i] ($1 \leq i \leq L-1$) 各自的i的RAM203上的变量AutoIndex_1、用于指定相同的音轨块2内的演奏数据组DeltaTime_2[i]以及Event_2[i] ($1 \leq i \leq M-1$) 各自的i的RAM203上的变量AutoIndex_2的各值都初始设定为0(以上、步骤S521)。由此,在图3的示例中,作为初始状态,首先,分别参考音轨块1内的排头的演奏数据组DeltaTime_1[0]和Event_1[0]、以及音轨块2内的排头的演奏数据组DeltaTime_2[0]和Event_2[0]。

[0079] 接下来,CPU201将指示乐曲数据的音轨块1内的当前的旋律事件的RAM203上的变量MelodyIndex的值、和表示乐曲数据的音轨块2内的当前的和弦事件群的RAM203上的排列变量CodeIndex()的各要素值分别初始设定为Null值(未定义值)(步骤S522)。另外,本实施例中的和弦设为是由3个或4个音构成的不含所谓的延伸音的和弦。因此,CodeIndex()是具有3个或4个要素值的变量排列。

[0080] 进而,CPU201将表示进行旋律以及和弦的进展(值“1”)或不进行旋律以及和弦的进展(值“0”)的RAM203上的变量SongStart的值初始设定为“1”(进展)(步骤S523)。之后,CPU201结束图5C的流程图所例示的图6的步骤S606的歌曲开始处理。

[0081] 图7是表示图4的步骤S403的键处理的详细例的流程图。首先,CPU201通过检查RAM203上的各变量右声部授课标记或左声部授课标记的值,来判定右声部授课或左声部授课是否开启(参考步骤S608、S610)(步骤S701)。

[0082] 在右声部授课以及左声部授课均未开启(步骤S701的判定为“否”)的情况下,CPU201进一步经由检测部204来判定是否在任何的键101发生按压(步骤S702)。

[0083] 若步骤S702的判定成为“是”,CPU201就对音源LSI210发行指示基于经由检测部204检测到的被按压的音符编号(按键音符)的音符开(note on)的发音控制数据,来指示发音(步骤S703)。即,在授课模式以外的模式下,在演奏者按压了任何的键101的情况下,按照演奏者的按压发出乐音。

[0084] 在步骤S702中未检测到按压的产生(步骤S702的判定为“否”)的情况下,或在步骤S703的处理后,CPU201经由检测部204来判定是否在任何的键101产生了离键(步骤S704)。

[0085] 若步骤S704的判定成为“是”,CPU201就对音源LSI210发行指示关于经由检测部204检测到的离键的音符编号(离键音符)的音符关(note off)的发音控制数据,来指示消音(步骤S705)。

[0086] 在步骤S705的处理后,或步骤S704的判定为“否”的情况下,CPU201结束图7的流程图所例示的图4的步骤S403的键处理。此外,在右声部授课或左声部授课开启而指定了授课模式的情况下,由于步骤S701的判定成为“是”,演奏者的按键、离键操作,根据键处理的不同不会被实质地处理,而是通过后述的图4的步骤S405的乐曲重放处理被处理。

[0087] 图8以及图9是表示基于在图2的计时器209中每TickTime[秒]产生的中断(参考图5A的步骤S502或图5B的步骤S512)来对图4的主处理施加中断来执行的歌曲重放中断处理的详细例的流程图。以下的处理对图3中例示的乐曲数据的音轨块1以及2的演奏数据组执行。

[0088] 首先,CPU201执行与音轨块1对应的一系列处理(步骤S801到S813)。最初CPU201判

定作为RAM203上的变量的SongStart值是否是1,即,是否指示了旋律以及和弦的进展(步骤S801)。

[0089] 在判定为未指示旋律以及和弦的进展(步骤S801的判定为“否”)的情况下,CPU201不进行旋律以及和弦的进展,将图8的流程图所例示的歌曲重放中断处理就这样结束。

[0090] 在判定为指示了旋律以及和弦的进展(步骤S801的判定为“是”)的情况下,CPU201判定右声部授课自动演奏是否开启(步骤S802)。

[0091] 在右声部授课以及自动演奏均未开启(步骤S802的判定为“否”)的情况下,CPU201不执行步骤S803到S813所示的与音轨块1的旋律声部相关的控制处理,移转到图9的步骤S814的与音轨块2的和弦声部相关的处理。

[0092] 在右声部授课或自动演奏的任一者开启(步骤S802的判定为“是”)的情况下,CPU201执行与音轨块1相关的步骤S803到S813的一系列处理。

[0093] 首先,CPU201判定表示与音轨块1相关的前次的事件的发生时刻起的相对时刻的DeltaT_1值是否与AutoIndex_1值所表示的今后要执行的演奏数据组的等待时间DeltaTime_1[AutoIndex_1]一致(步骤S803)。

[0094] 若步骤S803的判定成为“否”,CPU201就关于音轨块1,使表示前次的事件的发生时刻起的相对时刻的DeltaT_1值+1递增,来使时刻进展与本次的中断对应的1TickTime单位相应量(步骤S813)。之后,CPU201移转到后述的图9的步骤S814的与音轨块2的和弦声部相关的处理。

[0095] 若步骤S803的判定成为“是”,CPU201就判定AutoIndex_1值所表示的演奏数据组的事件Event[AutoIndex_1]是否是表示音轨块1的旋律乐曲数据的最后的EOT(音轨尾端)(参考图3)(步骤S804)。

[0096] 若步骤S804的判定成为“是”,CPU201就结束旋律乐曲数据的处理,并移转到与图9的步骤S814的与音轨块2的和弦声部相关的处理。

[0097] 在步骤S804的判定为“否”的情况下,CPU201关于音轨块1,判定AutoIndex_1值所表示的演奏数据组的事件Event[AutoIndex_1]是否是音符开事件(步骤S805)。

[0098] 若步骤S805的判定成为“否”,CPU201就立即执行AutoIndex_1值所表示的演奏数据组的事件Event[AutoIndex_1](步骤S810)。之后,CPU201移转到步骤S811的处理。

[0099] 若步骤S805的判定成为“是”,CPU201就将与音轨块1相关的AutoIndex_1值存放到RAM203上的变量MelodyIndex(步骤S806)。CPU201在这里不执行该音符开事件,在后述的图4的步骤S405的乐音重放处理内,在进行演奏者的旋律键域中的判定的基础上,基于变量MelodyIndex来进行发音处理(参考后述的图10的步骤S1002内的图11的步骤S1107)。

[0100] 接下来,CPU201使与存放于AutoIndex_1值所表示的当前的演奏数据组的事件Event[AutoIndex_1]的音符开的音符编号对应的键101的LED205经由图2的LED控制器206熄灭(步骤S807)。

[0101] 接着,CPU201检索与AutoIndex_1值的下一个发生的音符开事件对应的 α 事件目标($\alpha \geq 1$)的演奏数据组,使与存放于该事件Event[AutoIndex_1+ α]的音符开的音符编号对应的键101的LED205经由图2的LED控制器206点亮(步骤S808)。在没有符合条件的事件的情况下,CPU201不执行步骤S808的点亮动作。

[0102] 在本实施方式中,演奏者为了使演奏不会进展迟缓,对演奏者当前要按压的键101

的接下来应按压的键101提前进行光引导。为此,演奏者当前要按压的键101的LED205由于当前的前一个定时的按压时已经点亮,不再需要由于本次的按压定时的到来而进一步点亮,因此,首先在步骤S807,使演奏者当前要按压的键101的LED205熄灭。之后,在步骤S808,使演奏者当前要按压的键101的接下来应按压的键101的LED205提前点亮。

[0103] 接着,CPU201与步骤S808的情况同样地,执行与存放于 α 事件目标的演奏数据组的事件Event[AutoIndex_1+ α]的音符开对应的引导音的发音处理(步骤S809)。具体地,CPU201例如可以对音源LSI210指示与应按压存放于Event[AutoIndex_1+ α]的音符开对应的键101的指法信息对应的声音(“一”“二”“三”“四”“五”的任一者的声音)的发音。也可以取而代之,例如,CPU201对音源LSI210指示与存放于Event[AutoIndex_1+ α]的音符开的音符编号对应的音符名(“C”(C)到“B”(B)的任一者)的发音。音源LSI210例如将这些声音数据作为例如PCM声音保持,能通过将这些声音数据读出来进行发音。

[0104] 在上述步骤S809的处理后,或在Event[AutoIndex_1]不是音符开事件(步骤S805的判定为“否”)的情况下执行的步骤S810的处理后,CPU201使用于参考音轨块1内的演奏数据组的AutoIndex_1值+1递增(步骤S811)。

[0105] 此外,CPU201关于音轨块1,将表示本次参考的歌曲事件的发生时刻起的相对时刻的DeltaT_1值重置成0(步骤S812)。之后,CPU201移转到图9的步骤S814的与音轨块2的和弦声部相关的处理。

[0106] 接着,转移到图9的流程图的说明,CPU201判定左声部授课或自动演奏是否开启(步骤S814)。

[0107] 在左声部授课以及自动演奏均未开启(步骤S814的判定为“否”)的情况下,CPU201不执行步骤S815到S828所示的与音轨块2的和弦声部相关的控制处理,结束图8以及图9的流程图所示的本次的歌曲重放中断处理,进行返回,从而回到图4的主处理的中断时点的处理的执行。

[0108] 在左声部授课或自动演奏的任一者开启(步骤S814的判定为“是”)的情况下,CPU201执行与音轨块2相关的步骤S815到S828的一系列处理。

[0109] 首先,CPU201判定表示与音轨块2相关的前次的事件的发生时刻起的相对时刻的DeltaT_2值是否与AutoIndex_2值所表示的今后要执行的演奏数据组的等待时间DeltaTime_2[AutoIndex_2]一致(步骤S815)。

[0110] 若步骤S815的判定成为“否”,CPU201就关于音轨块2,使表示前次的事件的发生时刻起的相对时刻的DeltaT_2值+1递增,来使时刻进展与本次的中断对应的1TickTime单位的量(步骤S828)。之后,CPU201结束图8以及图9的流程图所示的本次的歌曲重放中断处理,进行返回,从而回到图4的主处理的中断时点的处理的执行。

[0111] 若步骤S815的判定成为“是”,CPU201就判定AutoIndex_2值所表示的演奏数据组的事件Event[AutoIndex_2]是否是表示音轨块2的和弦乐曲数据的最后的EOT(参考图3)(步骤S816)。

[0112] 若步骤S816的判定成为“是”,CPU201就结束和弦乐曲数据的处理,结束图8以及图9的流程图所示的本次的歌曲重放中断处理,进行返回,从而回到图4的主处理的中断时点的处理的执行。

[0113] 在步骤S816的判定为“否”的情况下,CPU201关于音轨块2,判定AutoIndex_2值所

表示的演奏数据组的事件Event[AutoIndex_2]是否是音符开事件(步骤S817)。

[0114] 若步骤S817的判定成为“否”,CPU201就立即执行AutoIndex_2值所表示的演奏数据组的事件Event[AutoIndex_2](步骤S824)。之后,CPU201移转到步骤S825的处理。

[0115] 若步骤S817的判定成为“是”,CPU201就将与音轨块2相关的AutoIndex_2值存放到RAM203上的排列变量CodeIndex()的末尾要素(步骤S818)。CPU201在这里不执行该音符开事件,在后述的图4的步骤S405的乐音重放处理内,在进行演奏者的和弦键域中的判定的基础上,基于排列变量CodeIndex()来进行发音处理(参考后述的图10的步骤S1004内的图12、图13或图14的步骤S1207)。

[0116] 接下来,CPU201使与存放于AutoIndex_2值所表示的当前的演奏数据组的事件Event[AutoIndex_2]的音符开的音符编号对应的键101的LED205经由图2的LED控制器206熄灭(步骤S819)。该处理和与旋律声部相关的步骤S807的处理同样。

[0117] 接着,CPU201检索与AutoIndex_2值的下一个发生的音符开事件对应的 β 事件目标($\beta \geq 1$)的演奏数据组,判定存放于该事件Event[AutoIndex_2+ β]的音符开的音符编号是否是某和弦的根音(步骤S820)。将表示是否是根音的信息存放在Event[AutoIndex_2+ β]。

[0118] 在步骤S820中判定为是根音的情况下,CPU201使与存放于Event[AutoIndex_2+ β]的音符开的音符编号对应的键101的LED205经由图2的LED控制器206闪烁(步骤S821)。在没有符合条件的事件的情况下,CPU201不执行步骤S821的闪烁动作。

[0119] 在步骤S820中判定为不是根音的情况下,CPU201使与存放于Event[AutoIndex_2+ β]的音符开的音符编号对应的键101的LED205经由图2的LED控制器206点亮(步骤S822)。在没有符合条件的事件的情况下,CPU201不执行步骤S821的闪烁动作。

[0120] 如此地,在本实施方式中,由于能在键101上区别根音和这以外的和弦构成音地引导,因此,演奏者对复杂的和弦也能进行意识到转位的合适的按压。上述的步骤S821以及步骤S822中的“+ β ”的动作与旋律声部中的图8的步骤S808的“+ α ”的动作同样。

[0121] 在步骤S821或步骤S822的处理后,CPU201执行与存放于 β 事件目标的演奏数据组的事件Event[AutoIndex_2+ β]的音符开对应的引导音的发音处理(步骤S823)。该动作与旋律声部中的图8的步骤S809的处理同样。

[0122] 在上述步骤S823的处理后,或者在Event[AutoIndex_2]不是音符开事件(步骤S817的判定为“否”)的情况下执行的步骤S824的处理后,CPU201使用于参考音轨块2内的演奏数据组的AutoIndex_2值+1递增(步骤S825)。

[0123] 此外,CPU201关于音轨块2,将表示本次参考的和弦事件的发生时刻起的相对时刻的DeltaT_2值重置成0(步骤S826)。

[0124] 然后,CPU201判定AutoIndex_2值所表示的接下来执行的音轨块2上的演奏数据组的等待时间DeltaTime_2[AutoIndex_2]是否是0,即,判定是否是与本次的事件同时执行的和弦所相关的事件(步骤S827)。

[0125] 若步骤S827的判定成为“否”,CPU201就结束和弦乐曲数据的处理,并结束图8以及图9的流程图所示的本次的歌曲重放中断处理,进行返回,从而回到图4的主处理的中断时点的处理的执行。

[0126] 若步骤S827的判定成为“是”,CPU201就回到步骤S816,重复与AutoIndex_2值所表示的在音轨块2上接下来执行的演奏数据组的事件Event_2[AutoIndex_2]相关的控制处

理。CPU201按照构成本次同时执行的和弦的构成音的数量的量,重复执行步骤S816到步骤S827的处理。以上的处理序列在本次的和弦声部的处理那样多个音符开事件在同时定时发音的情况下执行。

[0127] 通过以上的步骤S816到步骤S827的处理的重复,来在RAM203上的排列变量CodeIndex () 得到参考表示在本次的定时进行和弦判别的和弦的构成音的事件群的指标群。

[0128] 图10是表示图4的步骤S405的乐曲重放处理的详细例的流程图。首先,CPU201通过检查作为RAM203上的变量的右声部授课标记的值,来判定右声部授课是否开启(步骤S608)(步骤S1001)。

[0129] 在右声部授课开启(步骤S1001的判定为“是”)的情况下,CPU201执行旋律授课重放处理(步骤S1002)。关于该处理的详细例,使用图11的流程图之后叙述。

[0130] 在右声部授课未开启(步骤S1001的判定为“否”)的情况下,CPU201跳过步骤S1002的旋律授课重放处理。

[0131] 接下来,CPU201通过检查作为RAM203上的变量的左声部授课标记的值,来判定左声部授课是否开启(步骤S610)(步骤S1003)。

[0132] 在左声部授课开启(步骤S1003的判定为“是”)的情况下,CPU201执行和弦授课重放处理(步骤S1004)。关于该处理的详细例,使用图12到图14的流程图之后叙述。

[0133] 在左声部授课未开启(步骤S1003的判定为“否”)的情况下,CPU201跳过步骤S1004的和弦授课重放处理。

[0134] 之后,CPU201判定自动演奏是否开启(步骤S1005)。演奏者通过在图1的操作部103将自动演奏开关接通,进而选择自动演奏乐曲,来将这些信息存储到RAM203(参考图6的步骤S611)。在步骤S1005,CPU201判定作为RAM203上的变量的自动演奏标记是否成为开启。

[0135] 若步骤S1005的判定成为“是”,CPU201就通过RAM203上的变量MelodyIndex来取得存储于RAM203的自动演奏的乐曲数据的音轨块1(旋律声部)(参考图3)上的音符开事件Event_1[MelodyIndex](参考图8的步骤S805、S806),对音源LSI210(图2)发行该事件所表示的发音控制数据,由音源LSI210执行自动演奏旋律的发音处理(步骤S1006)。

[0136] 接着,CPU201通过RAM203上的排列变量CodeIndex () 来取得存储于RAM203的自动演奏的乐曲数据上的音轨块2(和弦声部)(参考图3)上的多个音符开事件群Event_2[CodeIndex ()](图9的步骤S817、S818参考),对音源LSI210(图2)发行指定这些事件群所表示的和弦音的多个发音控制数据(步骤S1007)。CodeIndex () 由于是排列变量,因此,实际上是从CPU201对音源LSI210发行与表示和弦构成音的3个或4个音符开事件Event_2[CodeIndex (0)]、Event_2[CodeIndex (1)]、Event_2[CodeIndex (3)]、Event_2[CodeIndex (4)]等各自对应的表示音符开的各发音控制数据,由音源LSI210执行自动演奏和弦的发音处理。

[0137] 在自动演奏开启的情况下,通过上述的步骤S1006和S1007,就经由音源LSI210自动演奏演奏者所指定的乐曲的旋律声部和演奏声部。演奏者通过配合该自动演奏,在键101上进行按键或离键的演奏操作,图4的步骤S403的键处理的详细例即图7的步骤S701的判定成为“否”,从而执行按键音符的发音处理(步骤S703)或离键音符的消音处理(步骤S705)。其结果,演奏者能配合自动伴奏来享受演奏。

[0138] 若步骤S1005的判定成为“否”,CPU201就跳过步骤S1006和S1007的处理。之后,CPU201结束图10的流程图所例示的图4的步骤S405的乐曲重放处理。

[0139] 图11是表示图10的步骤S1002的旋律授课重放处理的详细例的流程图。该处理是为了演奏者进行旋律声部的授课而执行的控制处理。

[0140] 首先,CPU201通过图8的步骤S805到S806的处理,来对存储于RAM203的变量MelodyIndex所表示的指标值判定旋律判定处理是否是未处理(步骤S1101)。

[0141] 在RAM203上的变量中记录有变量MelodyIndex所表示的指标值是处理完毕(步骤S1101的判定为“否”)的情况下,CPU201结束图11的流程图所例示的图10的步骤S1002的旋律授课重放处理。

[0142] 在RAM203上的变量中未记录变量MelodyIndex所表示指标值是处理完毕而是未处理(步骤S1101的判定为“是”)的情况下,CPU201在RAM203的适当的变量中记录了变量MelodyIndex是处理完毕后,阻止用于图8以及图9的流程图所例示的歌曲重放中断处理的计时器209的歌曲重放中断,使得暂时不执行歌曲重放中断处理(步骤S1102)。其结果,直到演奏者进行正确的旋律音的按压操作为止,授课模式的进展都停止。

[0143] 之后,CPU201成为经检测部204等待来自旋律键域的键101的由演奏者进行的按压的状态(步骤S1103)。在旋律键域中发生按压的期间,步骤S1103的判定成为“否”,从而重复执行步骤S1103的判定处理。

[0144] 若在旋律键域发生按压(若步骤S1103的判定成为“是”),CPU201取得通过RAM203上的变量MelodyIndex参考的、存储于RAM203的自动演奏的乐曲数据上的音轨块1(旋律声部)(参考图3)上的音符开事件Event_1[MelodyIndex](参考图8的步骤S805、S806),判定该事件所表示的音符编号是否与从检测部204通知的按键音符相等(步骤S1104)。即,在此,判定演奏者是否正确地按压了与在键101上光引导显示的旋律声部的键101相同的键。

[0145] 在步骤S1104的判定为“否”的情况下,即,在演奏者的旋律键域中的按压是不正解的情况下,CPU201经由音源LSI210,以不正解专用的音色发出演奏者所按压的音高的音(步骤S1105)。由此,演奏者能在旋律判定为不正解的情况下自然地认识到不正解。

[0146] 接着步骤S1105,CPU201执行评价声音发音处理(步骤S1106)。在该处理中,CPU201对音源LSI210发行与旋律声部的不正解的程度相应的评价声音,例如“旋律还有差距呢!”、“旋律可惜了!”,或者在完全不行的情况下指示爆炸音等的发音的发音控制数据,音源LSI210执行这些评价声音的发音处理。由此,演奏者能愉快地进行旋律声部的演奏授课。之后,CPU201回到步骤S1103的按压等待的状态。

[0147] 在步骤S1104的判定为“是”的情况下,即,在演奏者的旋律键域中的按压是正解的情况下,CPU201取得通过RAM203上的变量MelodyIndex参考的、存储于RAM203的自动演奏的乐曲数据上的音轨块1(旋律声部)(参考图3)上的音符开事件Event_1[MelodyIndex](图8的步骤S805、S806参考),对音源LSI210(图2)发行该事件所表示的音符开的发音控制数据,音源LSI210执行旋律发音处理(步骤S1107)。

[0148] 之后,CPU201将用于图8以及图9的流程图所例示的歌曲重放处理的计时器209的歌曲重放中断的阻止解除,恢复歌曲重放中断处理(步骤S1108)。其结果,恢复授课模式的进展。然后,CPU201结束图11的流程图所例示的图10的步骤S1002的旋律授课重放处理。

[0149] 图12是表示图10的步骤S1004的和弦授课重放处理的第1实施方式的详细例的流

程图。该处理是为了演奏者进行和弦声部的授课而执行的控制处理。

[0150] 首先,CPU201通过图9的步骤S817到S818的处理来对存储于RAM203的变量CodeIndex所表示的指标值判定和弦判定处理是否是未处理(步骤S1201)。

[0151] 在RAM203上的变量中记录有变量CodeIndex所表示的指标值是处理完毕(步骤S1201的判定为“否”)的情况下,CPU201结束图12的流程图所例示的图10的步骤S1004的和弦授课重放处理。

[0152] 在RAM203上的变量中未记录变量CodeIndex所表示的指标值是处理完毕而是未处理(步骤S1201的判定为“是”)的情况下,CPU201在RAM203的适当的变量中记录了变量CodeIndex是处理完毕后,阻止用于图8以及图9的流程图所例示的歌曲重放中断处理的基于计时器209的歌曲重放中断,使得暂时不执行歌曲重放中断处理(步骤S1202)。其结果,直到演奏者进行正确的和弦的按压操作为止,授课模式的进展都停止。

[0153] 之后,CPU201成为经检测部204等待来自和弦键域的键101的演奏者的按压的状态(步骤S1203)。在和弦键域中未发生按压的期间,步骤S1203的判定成为“否”,重复步骤S1203的判定处理。

[0154] 若发生和弦键域中的按压(若步骤S1203的判定成为“是”),CPU201就取得通过RAM203上的排列变量CodeIndex() 参考的、存储于RAM203的授课乐曲数据上的音轨块2(和弦声部)(参考图3)上的多个音符开事件群Event_2[CodeIndex()] (参考图9的步骤S817、S818),对是否能够通过本次到目前位置发生的按压的各按键音符确定以这些事件群确定的正解和弦进行和弦判定(步骤S1204)。作为和弦判定的手法,能采用各种判定等级的判定技术,例如,对应于演奏者的演奏授课的熟练度,仅按压根音就能确定大三和弦,或者若按压了根音和另1个音就能确定和弦,或者若不按压全部的和弦构成音,就不能确定和弦。为此,演奏者能在操作部103指定和弦授课的难易度的等级。作为和弦判定方式,例如能采用“特开2018-54713号公报”记载的技术,因此省略其详细的说明。

[0155] 步骤S1204中的和弦判定处理的结果,CPU201判定是否确定了正解和弦(步骤S1205)。

[0156] 在未通过演奏者到当前为止的按压操作确定正解和弦(步骤S1205的判定为“否”)的情况下,CPU201经由音源LSI210以静止和弦发出能通过到当前为止的按压操作确定的不正解和弦(步骤S1206)。在此,所谓静止和弦,是指不管哪个键被按压,都在按压着的期间以专用的音色持续鸣响所决定的和弦构成音的全部。由此,演奏者在和弦判定为不正解的情况下,能自然地认识不正解。这里的不正解和弦可以设为以演奏者按压的未构成正解和弦的音高为根音的大三和弦,在有能对某些和弦确定的多个按压的情况下,也可以设为该能确定的和弦。

[0157] 接着步骤S1206,CPU201执行评价声音发音处理(步骤S1207)。在该处理中,CPU201对音源LSI210发行与和弦声部的不正解的程度相应的评价声音,例如“和弦还有差距呢!”、“和弦可惜了!”,或者在完全不行的情况下是指示爆炸音等的发音的发音控制数据,音源LSI210执行这些评价声音的发音处理。由此,演奏者能愉快地进行和弦声部的演奏授课。之后,CPU201回到步骤S1203的按压等待的状态。

[0158] 在步骤S1205的判定为“是”的情况下,即,在演奏者的和弦键域中的按压能确定正解和弦的情况下,CPU201取得通过RAM203上的排列变量CodeIndex() 参考的、存储于

RAM203的授课乐曲数据上的音轨块2(和弦声部)(参考图3)上的多个音符开事件群Event_2[CodeIndex ()](图9的步骤S817、S818参考),对音源LSI210(图2)发行指定这些事件群所表示的和弦音的多个发音控制数据(步骤S1208)。与图10的步骤S1007的情况同样,由于CodeIndex ()是排列变量,实际上是从CPU201对音源LSI210发行与表示和弦构成音的3个或4个音符开事件Event_2[CodeIndex (0)]、Event_2[CodeIndex (1)]、Event_2[CodeIndex (3)]、Event_2[CodeIndex (4)]等各自对应的表示音符开的各发音控制数据,由音源LSI210执行正解和弦的发音处理。

[0159] 之后,CPU201将用于图8以及图9的流程图所例示的歌曲重放处理的基于计时器209的歌曲重放中断的阻止解除,恢复歌曲重放中断处理(步骤S1209)。其结果,恢复授课模式的进展。然后,CPU201结束图12的流程图所例示的图10的步骤S1004的和弦授课重放处理。

[0160] 图13是表示图10的步骤S1004的和弦授课重放处理的第2实施方式的详细例的流程图。该处理是为了演奏者进行和弦声部的授课而执行的控制处理。关于进行与图12所例示的和弦授课重放处理的第1实施方式的详细例的情况相同处理的步骤,赋予与图12的情况相同的步骤编号。

[0161] 因此,首先,图13的步骤S1201、S1202以及S1203的各步骤的处理与图12的第1实施方式的情况同样。

[0162] 若发生和弦键域中的按压(若图13的步骤S1203的判定成为“是”),CPU201就取得通过RAM203上的排列变量CodeIndex () 参考的、存储于RAM203的授课乐曲数据的音轨块2(和弦声部)(参考图3)上的多个音符开事件群Event_2[CodeIndex ()](参考图9的步骤S817、S818)。然后,CPU201在步骤S1203中的按压检测中,判定以上述事件群确定的正解和弦的构成音的任一者是否在本次被按压(步骤S1301)。

[0163] 若步骤S1301的判定成为“否”,CPU201就对音源LSI210发行指示步骤S1203中经由检测部204检测到的按键音符的音符开的发音控制数据(步骤S1302)。其结果,音源LSI210执行不是正解和弦的构成音的乐音的发音处理。

[0164] 在步骤S1302的发音处理后,CPU201执行与图12的步骤S1207的情况同样的评价声音发音处理(图13的步骤S1207)。之后,CPU201回到图13的步骤S1203的按压等待状态。

[0165] 若步骤S1301的判定成为“是”,CPU201就判定本次到目前为止以多个音符开事件群Event_2[CodeIndex ()]确定的正解和弦的构成音以外的乐音的键101是否未被按压(步骤S1303)。

[0166] 在正解和弦的构成音以外的乐音的键101被按压(步骤S1303的判定为“否”)的情况下,CPU201在执行步骤S1207的评价声音发音处理后,回到图13的步骤S1203的按压等待状态。

[0167] 在正解和弦的构成音以外的乐音的键101未被按压(步骤S1303的判定为“是”)的情况下,CPU201与图12的情况同样地执行正解和弦的和弦判定处理(图13的步骤S1204)以及确定判定处理(图13的步骤S1205)。

[0168] 其结果,在不能确定正解和弦(图13的步骤S1205的判定为“否”)的情况下,CPU201在执行步骤S1207的评价声音发音处理后,回到图13的步骤S1203的按压等待状态。

[0169] 在能确定正解和弦(图13的步骤S1205的判定为“是”)的情况下,CPU201与图12的

情况同样地执行正解和弦的发音处理(图13的步骤S1208)。

[0170] 然后,CPU201在与图12的情况同样地将歌曲重放中断的阻止解除后(图13的步骤S1209),结束图13的流程图所例示的图10的步骤S1004的和弦授课重放处理。

[0171] 如以上那样,根据和弦授课重放处理的第2实施方式,演奏者能认识到按压了不是正解和弦的构成音的和弦键域的乐音的键101。

[0172] 图14是表示图10的步骤S1004的和弦授课重放处理的第3实施方式的详细例的流程图。该处理是为了演奏者进行和弦声部的授课而执行的控制处理。关于进行与图12所例示的和弦授课重放处理的第1实施方式的详细例的情况相同的处理的步骤,赋予与图12的情况相同的步骤编号。

[0173] 因此,首先,图13的步骤S1201、S1202以及S1203的各步骤的处理与图12的第1实施方式的情况同样。

[0174] 若发生和弦键域中的按压(若图14的步骤S1203的判定成为“是”),CPU201就对音源LSI210发行指示从检测部204通知的按键音符的音符开的发音控制数据(步骤S1401)。其结果,音源LSI210对与被按压的键101对应的乐音进行发音处理。

[0175] 接下来,CPU201与图12的情况同样地,对是否能通过本次到目前为止检测到的按压来确定正解和弦进行和弦判定(图14的步骤S1204),判定是否能确定正解和弦(图14的步骤S1205)。

[0176] 在不能确定正解和弦(步骤S1205的判定为“否”)的情况下,CPU201在执行与图12的步骤S1207的情况同样的评价声音发音处理后(图14的步骤S1207),回到图14的步骤S1203的按压等待状态。

[0177] 在能确定正解和弦(步骤S1205的判定为“是”)的情况下,CPU201对音源LSI210发行通过本次到目前为止的按压基于步骤S1401来指示发音中的乐音的消音的发音控制数据。

[0178] 接着,CPU201与图12的情况同样地执行正解和弦的发音处理(图14的步骤S1208)。

[0179] 然后,CPU201在与图12的情况同样地将歌曲重放中断的阻止解除后(图14的步骤S1209),结束图14的流程图所例示的图10的步骤S1004的和弦授课重放处理。

[0180] 如以上那样,根据和弦授课重放处理的第3实施方式,能将不正解时的不需要的发音消音并对正解的和弦构成音进行授课演奏,能防止演奏者感到不快感。

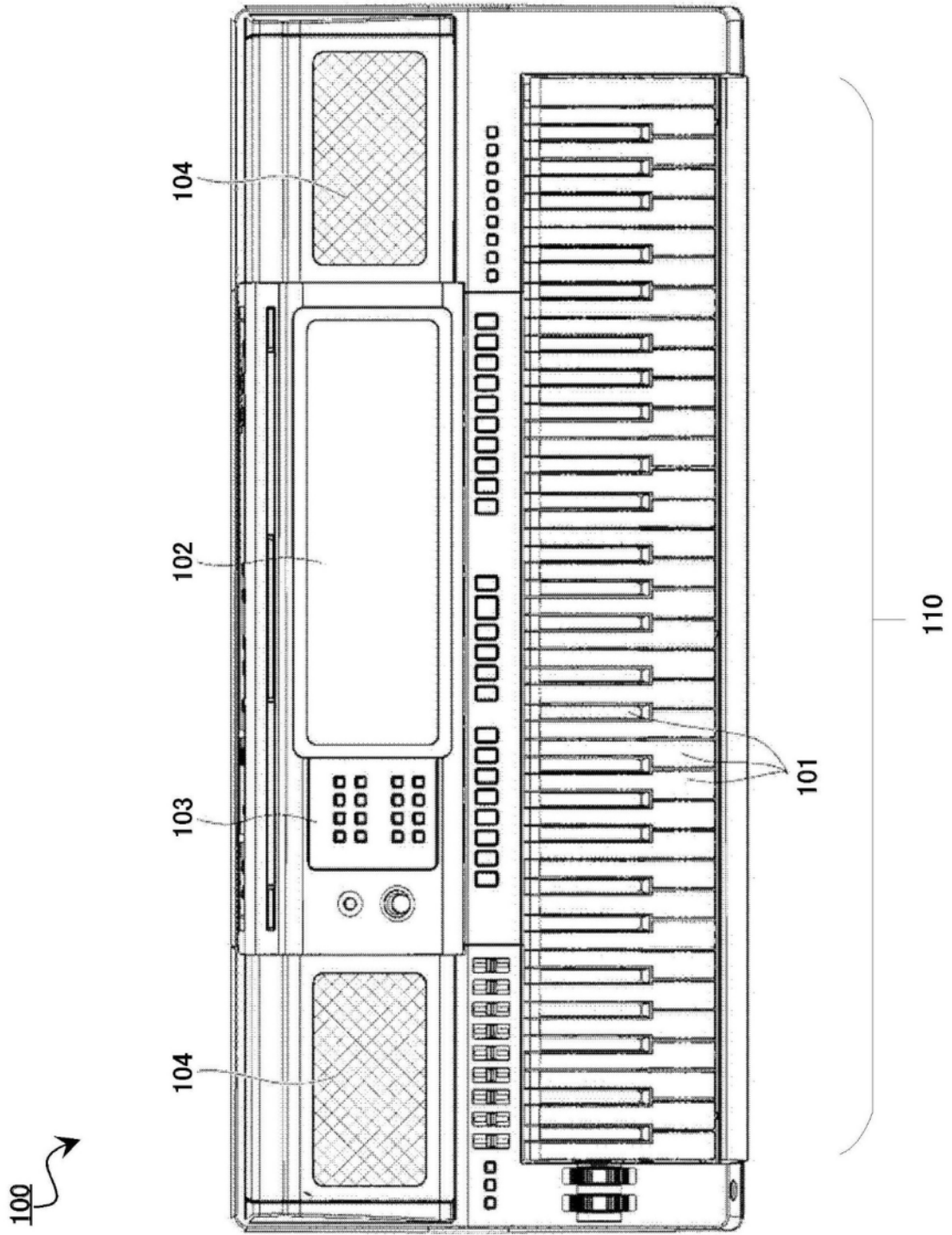


图1

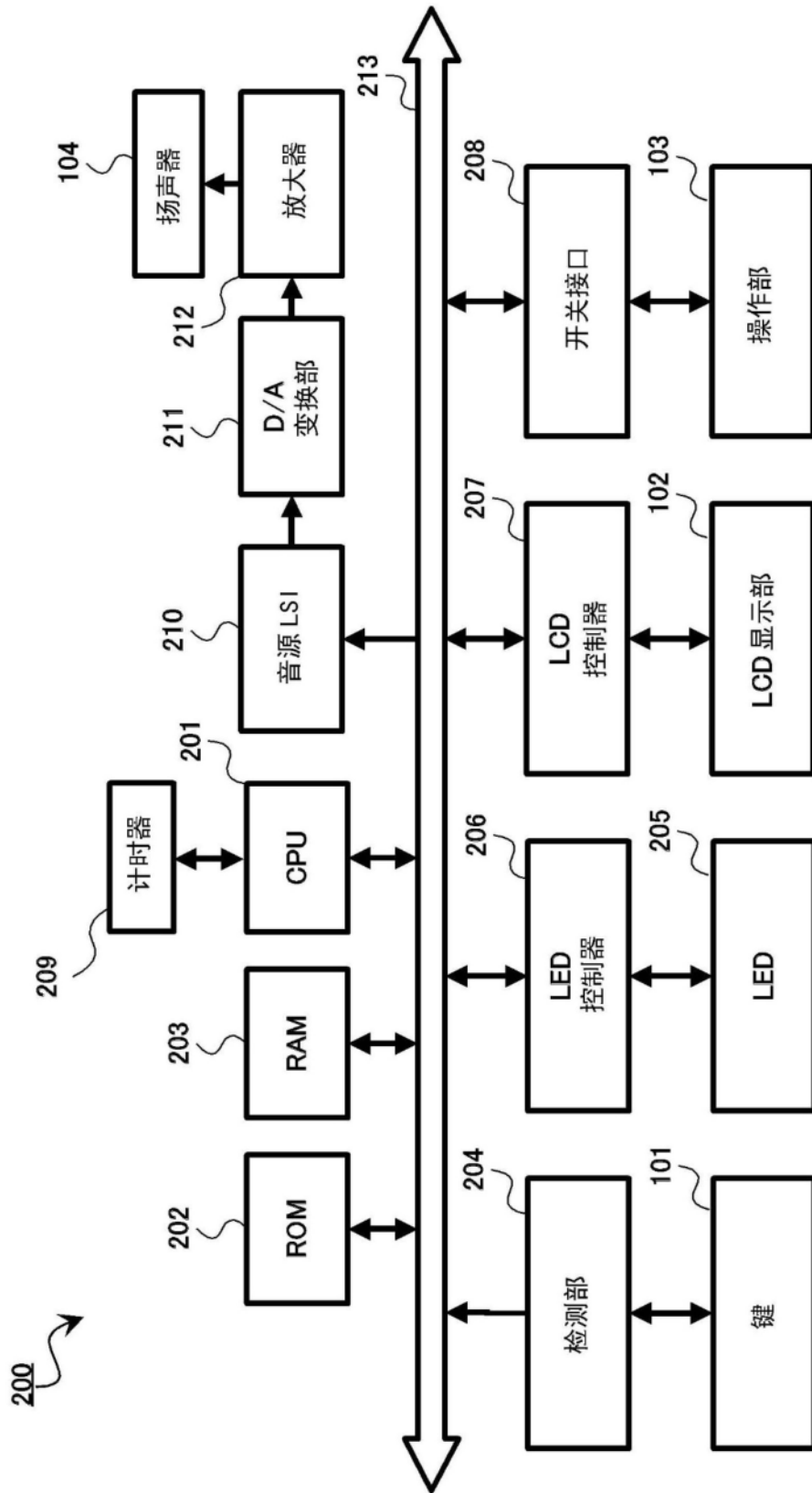


图2

头部块		ChunkID	固定字符串"MTkd"=0x4d546864
		ChunkSize	头部块的长度=0x00000006
		FormatType	格式：例如 0x0001
		NumberOfTrack	音轨数：例如 0x0002
		TimeDivision	时基：例如 0x1e0
音轨块 1 (旋律声部)	DeltaTime_1[0] Event_1[0]	ChunkID	固定字符串"MTrk"=0x4d54726b
		ChunkSize_1	音轨块 1 的长度
	DeltaTime_1[1] Event_1[1]	DeltaTime	近前事件起的等待时间
	Event	事件	
	
	DeltaTime_1[L-1] Event_1[L-1]	DeltaTime	近前事件起的等待时间
	Event	事件	
			音轨的最后必定是[EOT]
音轨块 2 (和旋律声部)	DeltaTime_2[0] Event_2[0]	ChunkID	固定字符串"MTrk"=0x4d54726b
		ChunkSize_2	音轨块 2 的长度
	DeltaTime_2[1] Event_2[1]	DeltaTime	近前事件起的等待时间
	Event	事件	
	
	DeltaTime_2[M-1] Event_2[M-1]	DeltaTime	近前事件起的等待时间
	Event	事件	
			音轨的组后必定是[EOT]

图3

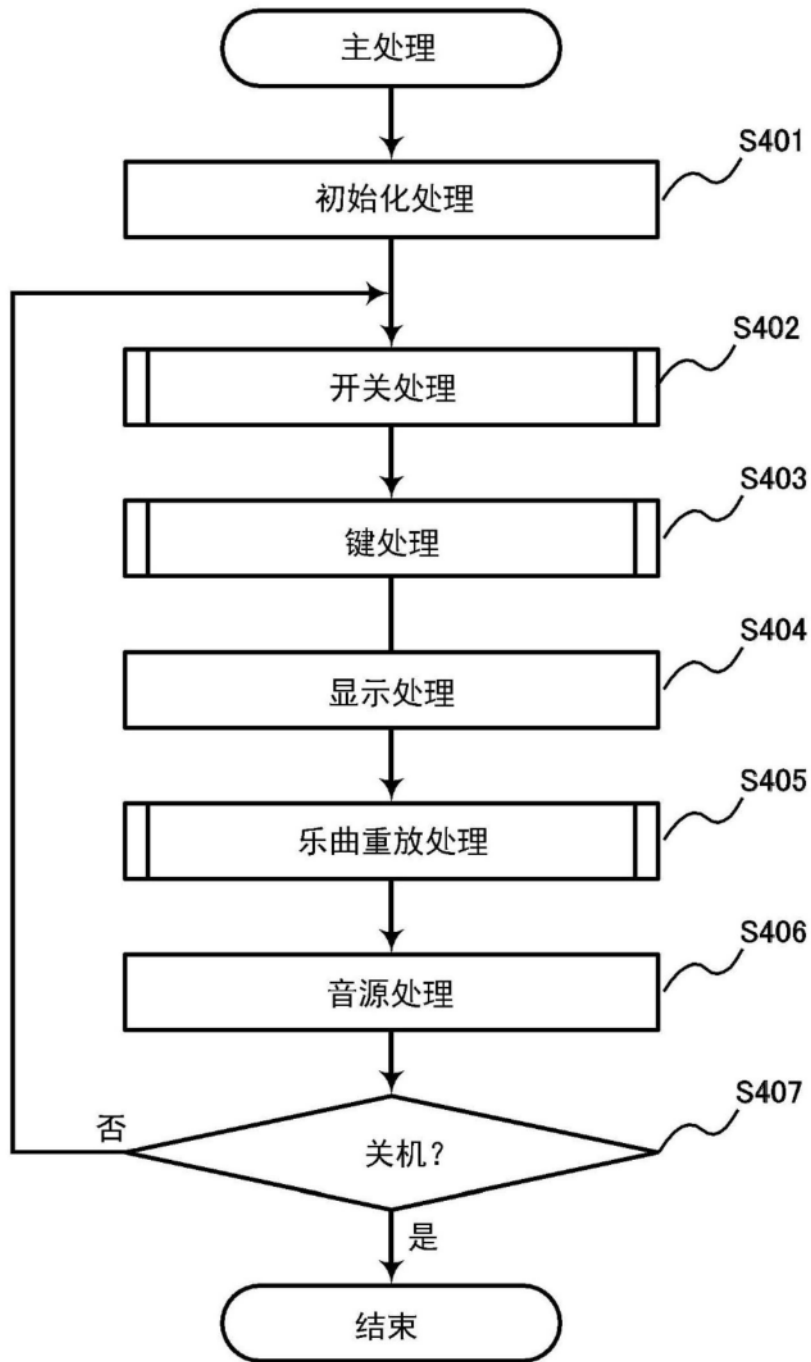


图4

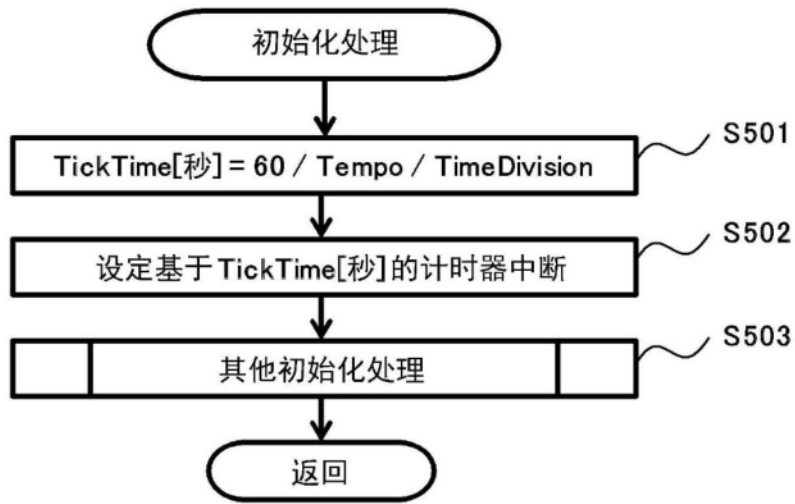


图5A

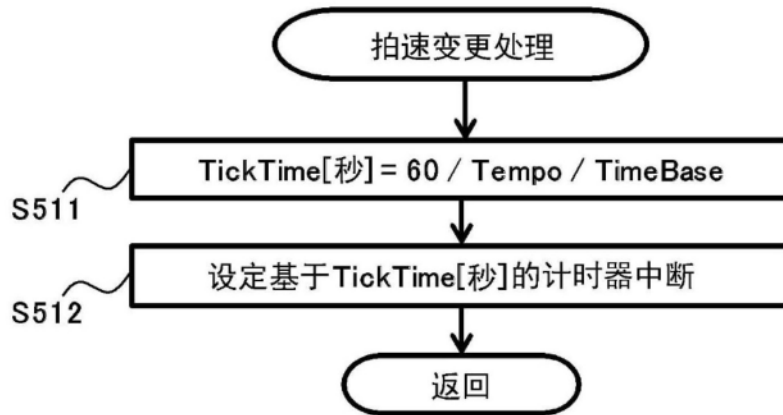


图5B

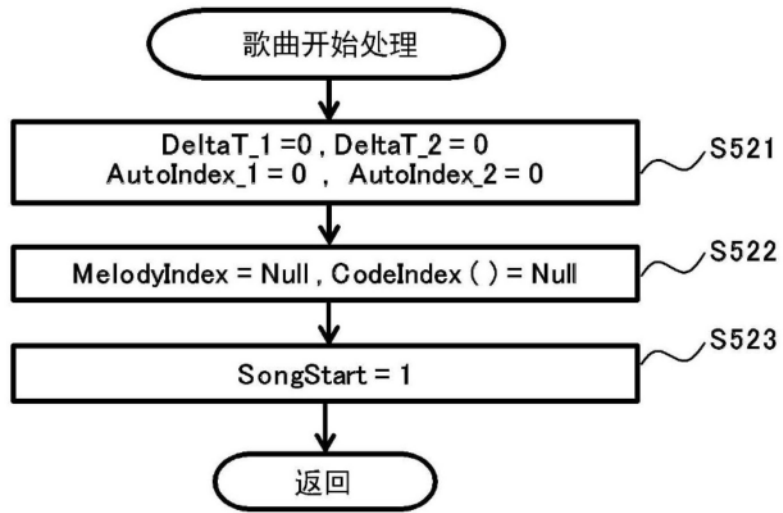


图5C

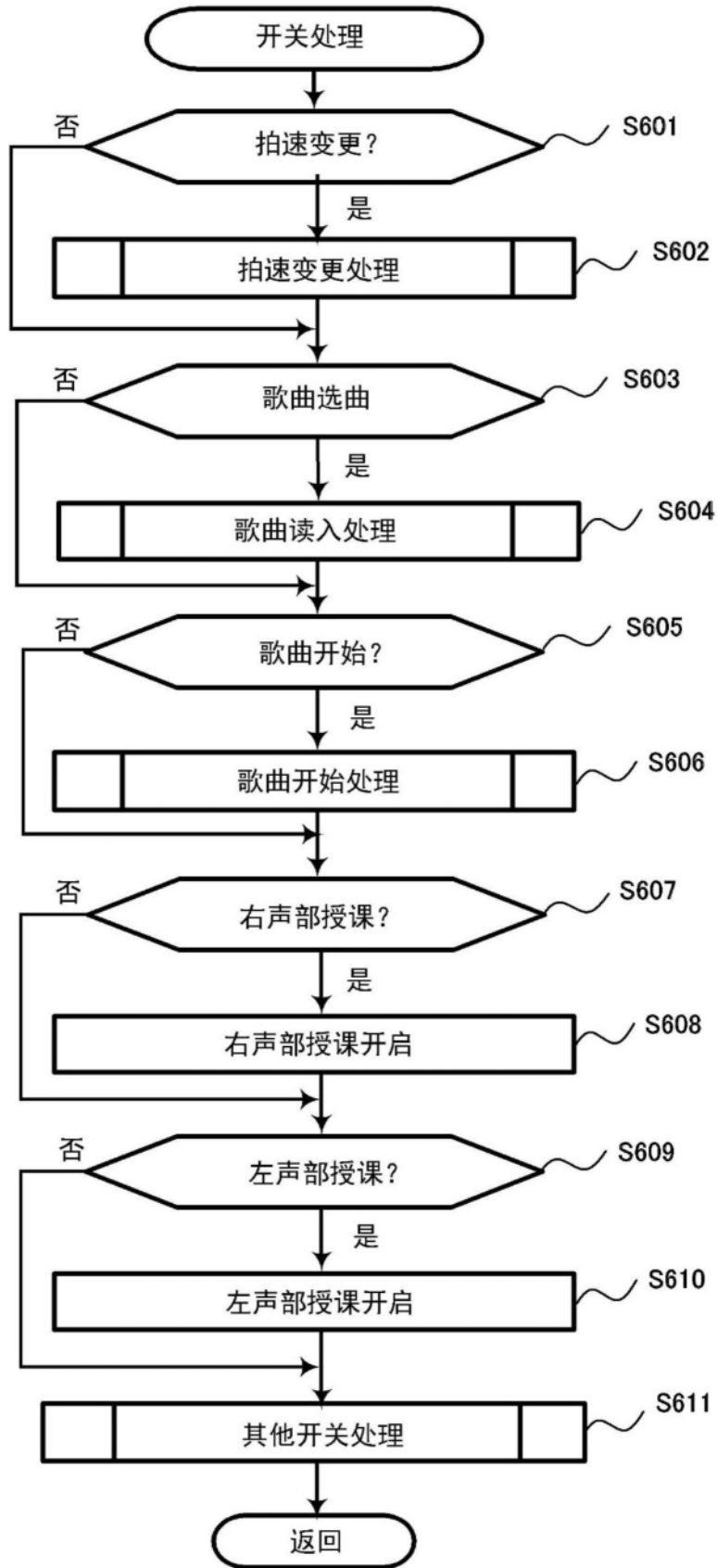


图6

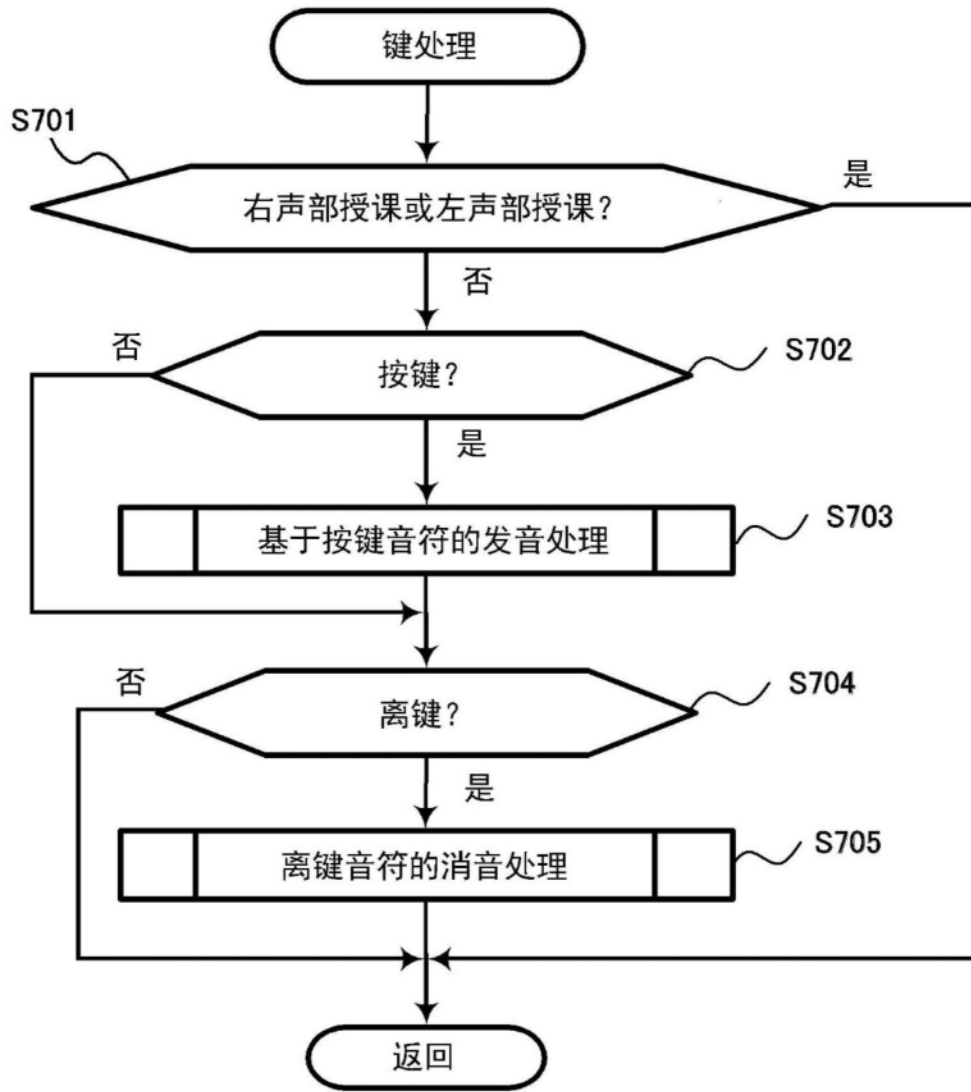


图7

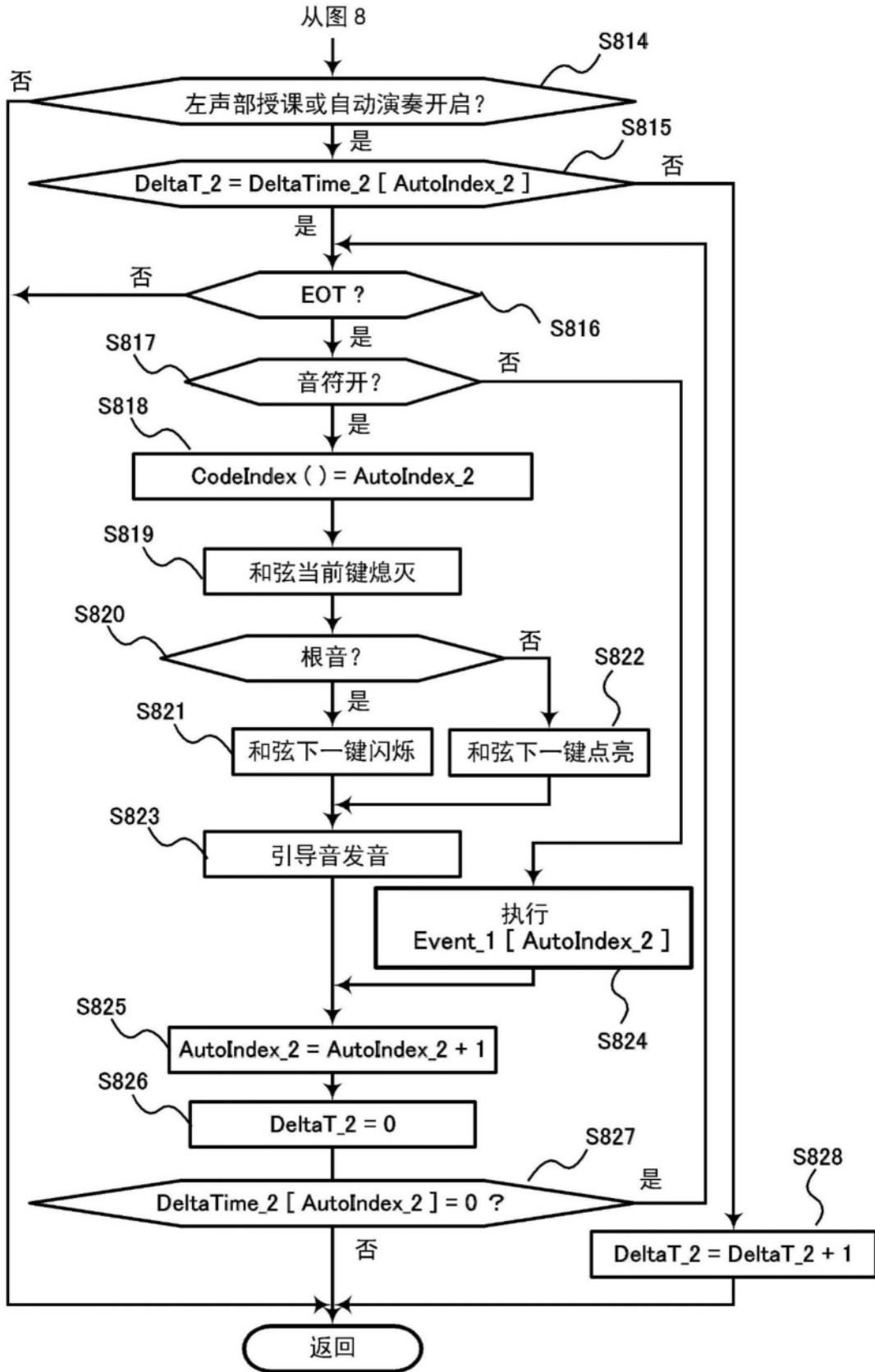


图9

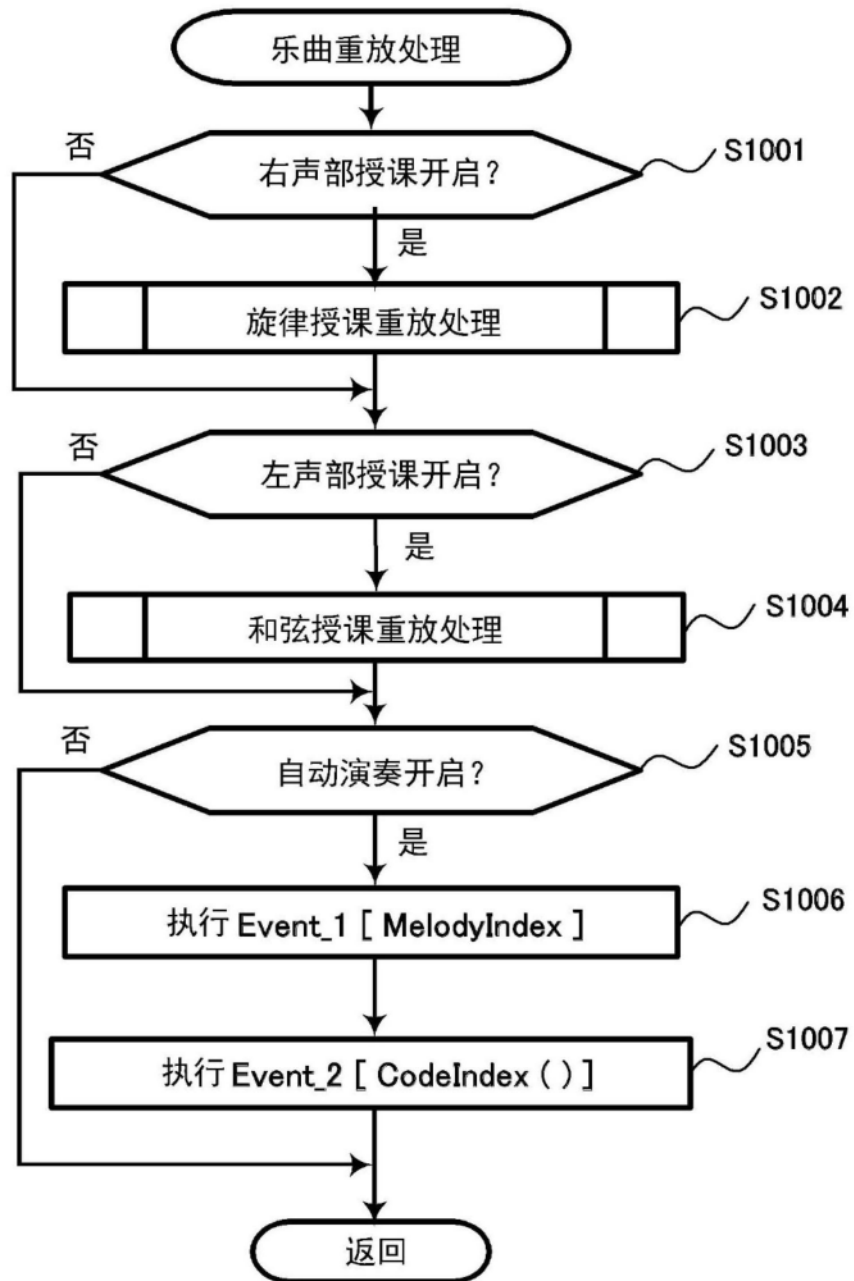


图10

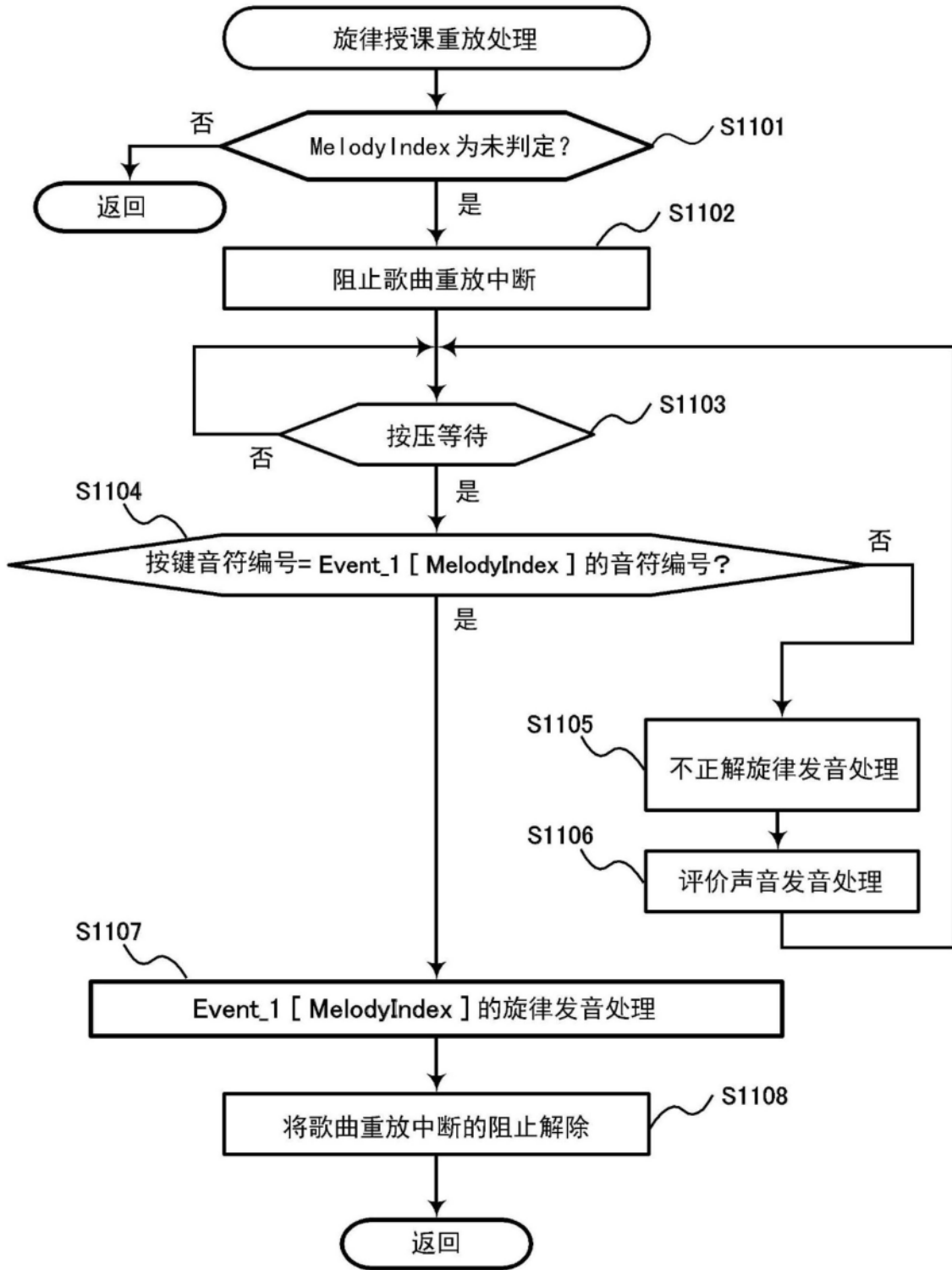


图11

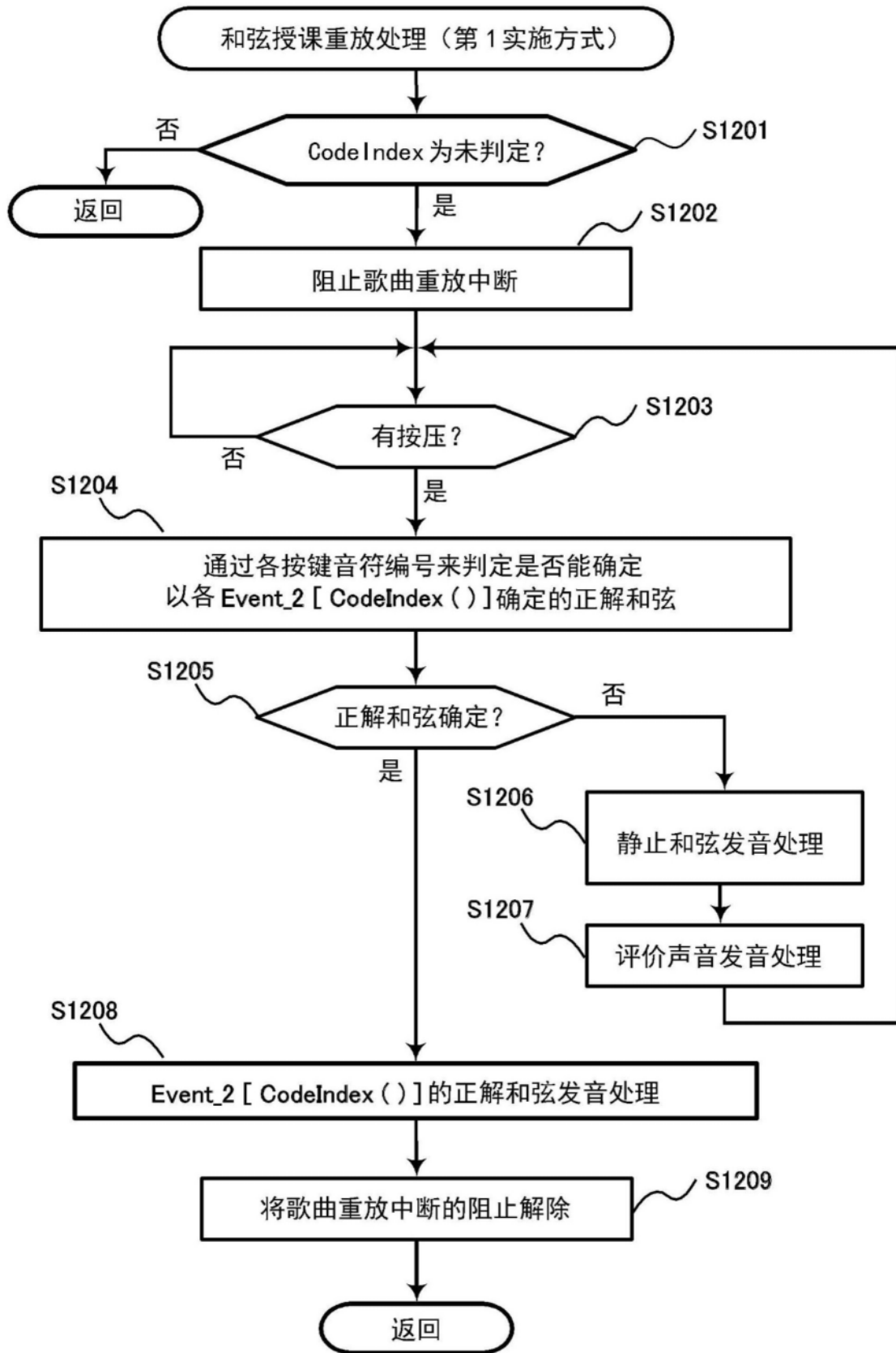


图12

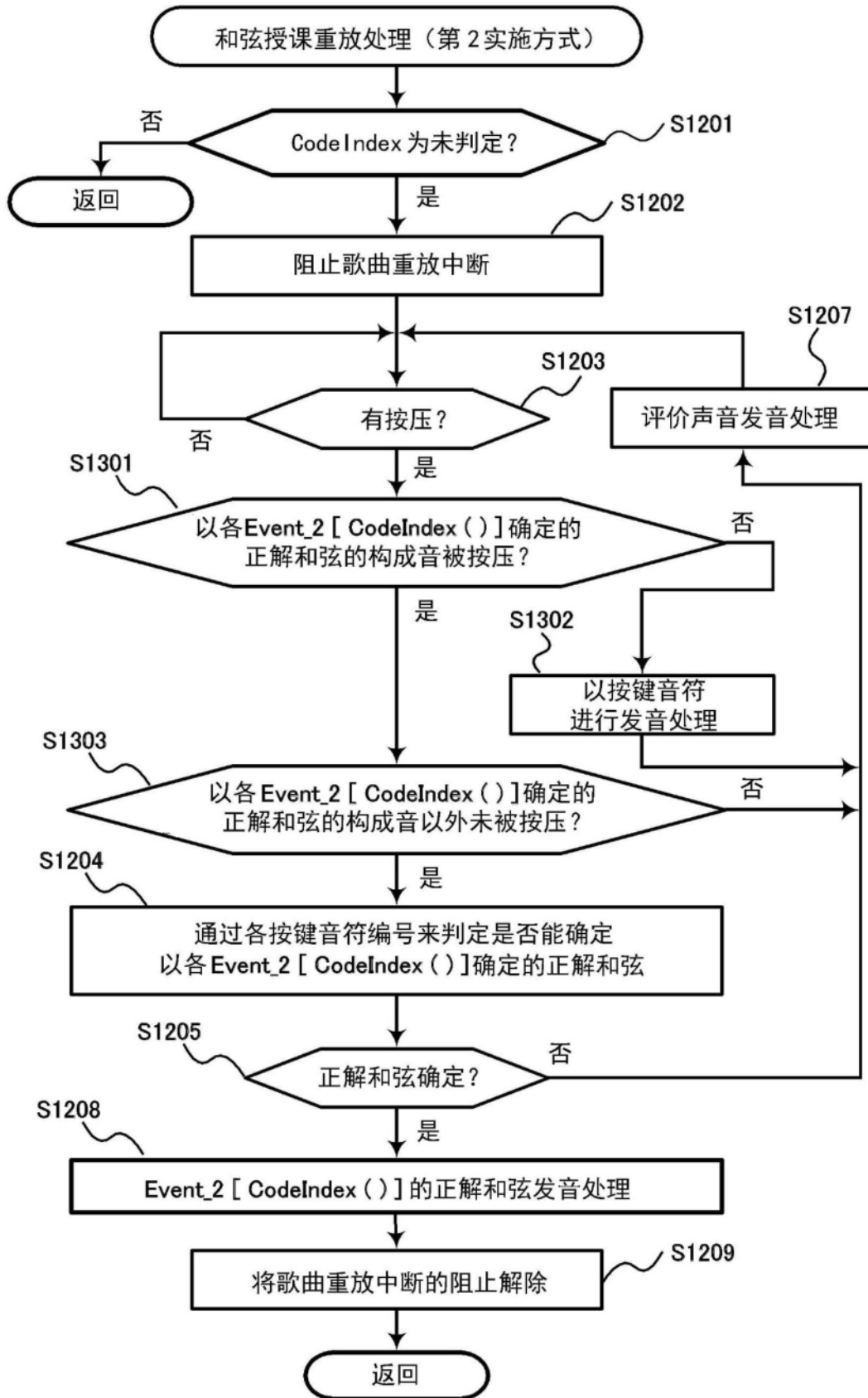


图13

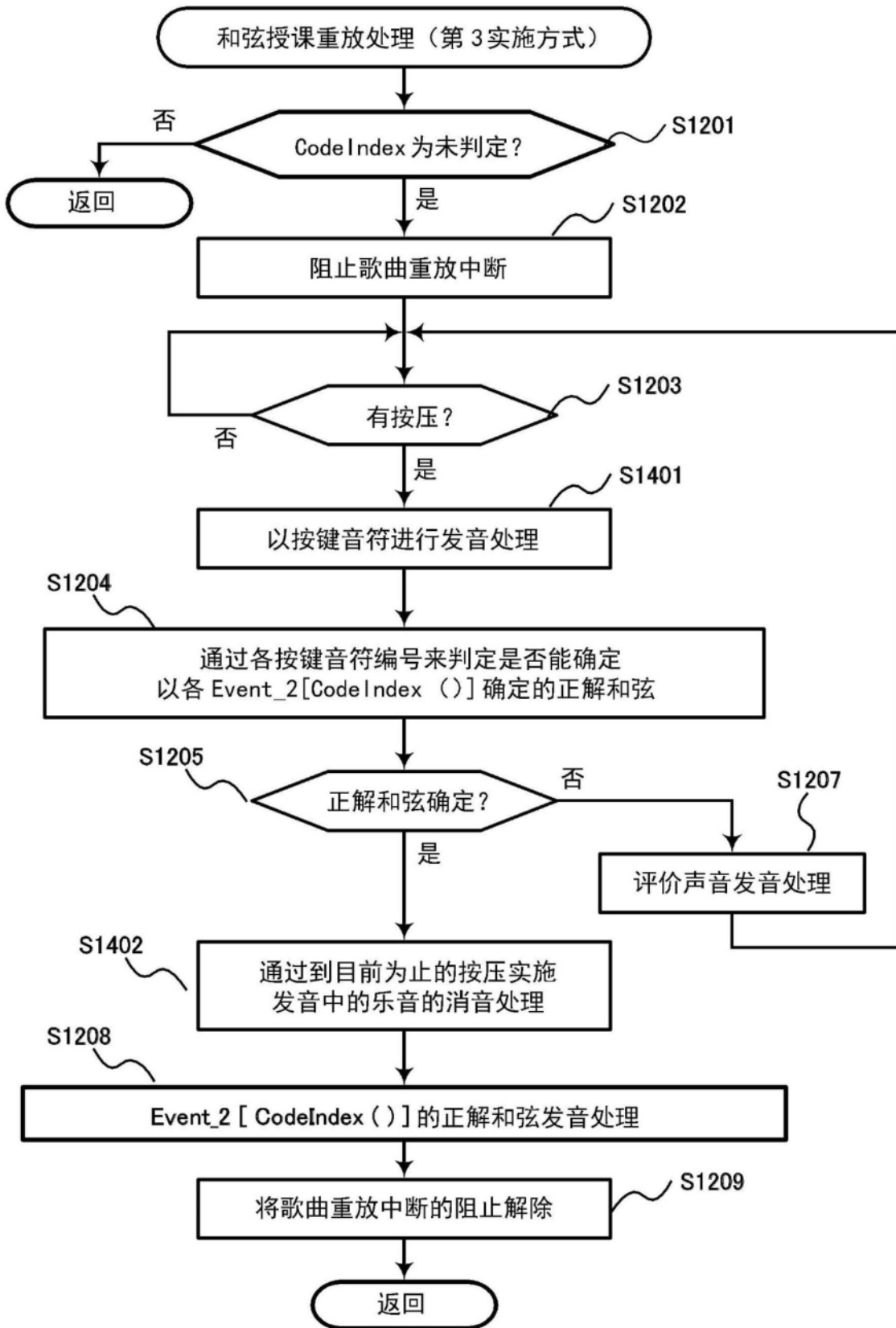


图14