

知乎



首发于

Mathematica 还能这样玩



Mathematica 歌姬计划（2）—— 听力训练



yxl1lc
物理研究生，业余程序员

已关注

45 人赞同了该文章

欢迎来到 Mathematica 歌姬计划，这一计划的目标是用 MMA 像 Vocaloid 那样调教自己想要的人声电子音乐。

▲ 赞同 45 ▼

● 6 条评论

➦ 分享

★ 收藏

知乎



首发于

Mathematica 还能这样玩

在上篇文章中，我们研究了如何在 Mathematica 中调用 Cortana 音源：

yxllc: Mathematica 歌姬计划 (1)
—— 与 Cortana 的联合

zhuanlan.zhihu.com



本文为本系列的第二篇文章，在前一篇的基础上，介绍如何运用 Mathematica 测量并获取 Cortana 音源的基频与共振峰信息。2.1 节介绍将 Speak 函数产生的音频流记录为波形的的方法，2.2 节介绍基频曲线 (PIT) 的测量方法，2.3 节介绍正弦模型以及共振曲线的测量方法，以及最后总结。

由于本系列的核心重点是 MMA 编程，本文假设读者对 DSP (即 Digital signal processing, 数字信号处理) 以及物理声学相关的基础知识有所了解，将不会涉及理论部分的一系列推导。本文会尽量避免数学公式的出现，将重点放在编程实现的过程上面。

2.1 录下 Cortana 的声音

在上篇文章最后，我们新定义了 MySpeak 函数，使得 Cortana 可以成功说中文，解决了 MMA 自带的 Speak 函数不支持中文输入的问题。然而，不论是原来的 Speak 函数还是新定义的 MySpeak 函数，它们的输出结果均是音频流的形式。也就是说，这种情况下 Cortana 姐姐的声音是转瞬即逝的，就像英语听力考试一样，说完了，就没了，然后麦酱一脸懵逼。



考虑到麦酱的计算力有限（相比于 C 语言来说），要做歌声合成的话，基于流的音频实时处理是一件几乎不可能完成的事情。所以一个直接的想法就是把音频流产生的声音“存起来”，变成波形数据的形式，可以被程序实时调用并做下一步计算。

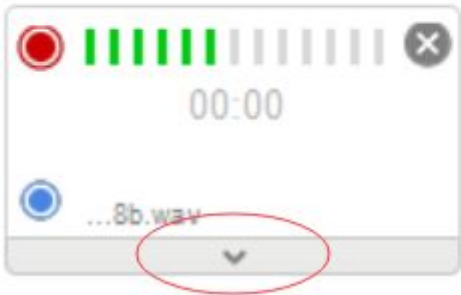
赞同 45

6 条评论

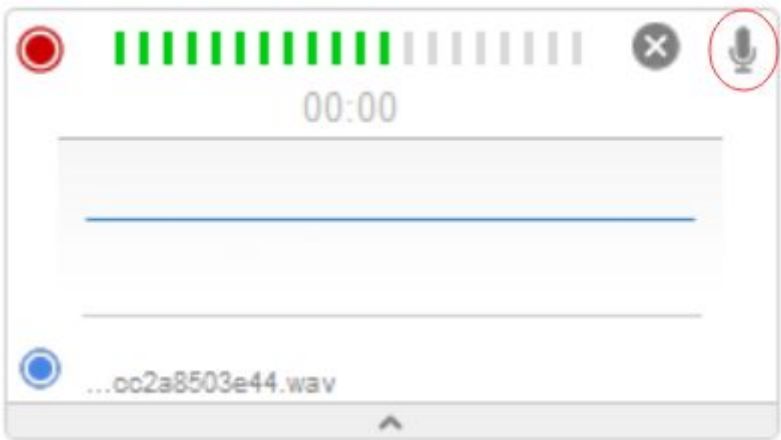
分享

收藏

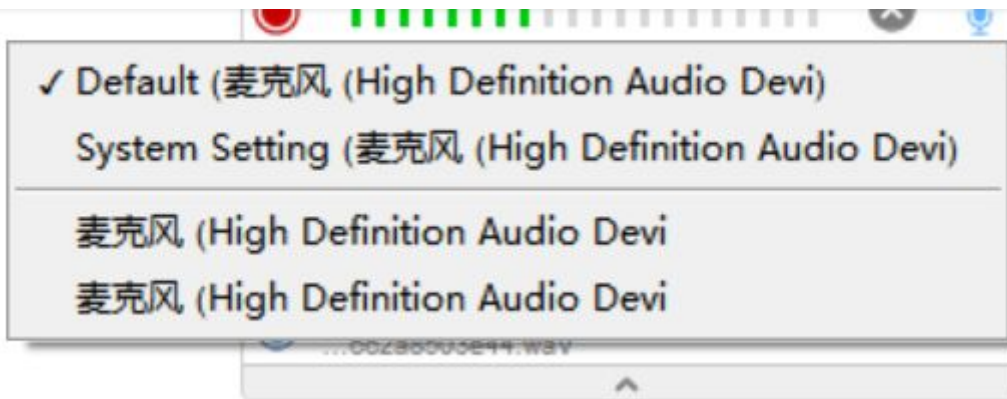
In[1]:= **AudioCapture[]**
[捕获音频]



展开上面画圈的箭头：



点击这个“话筒”图标可以设置音频输入设备：



我的电脑目前只能支持外录，也就是利用麦克风硬件录制环境中的声音。有些电脑是支持内录的，即定位声卡上临时存放音频流波形信息的缓冲区，在播放时不断将它们提取出来，相当于做一个“音频流 -> 文件流”的操作。就算电脑默认不支持内录，我们也可以学习各种屏幕录制软件所采用的方法，配置虚拟声卡欺骗操作系统，假装外录，实际上是内录。这里就不介绍具体怎么配置虚拟声卡了，总之配置好了之后，刚才的音频输入设备中会多出专门用来内录的“虚拟设备”。选择这个设备，然后在Cortana 发声时点击红色按钮进行录制就行了。

正当麦酱调好录音机准备开始给 Cortana 姐姐录音时，Cortana 姐姐表示：“不用麻烦你翻录了，我这有现成的录音带”。

原来，微软校长早已考虑到像麦酱这样的学生对听力素材有着巨大的需求，于是在设计Cortana 这样的 TTS 平台之初就提供了文件流输出的功能。让我们回顾一下上篇文章提到的微软官方说明文档：

SpeechSynthesizer 类

msdn.microsoft.com



注意看这一行：

<code>SetOutputToWaveFile(String^, SpeechAudioFormatInfo^)</code>	配置 SpeechSynthesizer 对象将输出追加到指定的格式的波形音频格式文件。
---	--

也就是说我们直接调用这个函数就可以生成 Cortana 声音的波形文件，直接拿到“录音带”！

于是我们稍微修改一下上次的 MySpeak 函数

▲ 赞同 45



● 6 条评论

➦ 分享

★ 收藏

知乎



首发于

Mathematica 还能这样玩

```
Options[VoiceData] = {"Rate" -> 0, "Volume" -> 100,
  "SampleRate" -> 44100, "SampleDepth" -> 16, "Channels" -> 1};
VoiceData[string_, OptionsPattern[]] :=
Module[{synth, format, tmpfile, data},
  synth = NETNew["System.Speech.Synthesis.SpeechSynthesizer"];
  synth@Rate = OptionValue["Rate"];
  synth@Volume = OptionValue["Volume"];
  format =
    NETNew["System.Speech.AudioFormat.SpeechAudioFormatInfo",
      OptionValue["SampleRate"], OptionValue["SampleDepth"],
      OptionValue["Channels"]];
  tmpfile = Close@OpenWrite[] <> ".wav";
  synth@SetOutputToWaveFile[tmpfile, format];
  synth@Speak[ToString[string]];
  synth@Dispose[];
  data = First@AudioData@AudioTrim[Import[tmpfile]];
  DeleteFile[tmpfile];
  SampledSoundList[data, OptionValue["SampleRate"]]
]
```

大概思路就是利用 NetLink 调用上述 SetOutputToWaveFile 方法指定 Cortana 输出音频文件的地址 (Close@OpenWrite[] <> ".wav")，作为一个临时文件 (tmpfile)，然后用 Import 函数导入这个 WAV 波形至 MMA 中，成功导入后将它删掉。同样像上次的 MySpeak 函数那样，我们可以分别用 "Rate" 和 "Volume" 选项设置发音速度和音量，另外增加的选项是 "SampleRate"，可用来指定采样率，比如输入以下代码：

```
ssl = VoiceData["你好", "Rate" -> 0, "Volume" -> 100,
  "SampleRate" -> 44100]
```

运行后变量 ssl 被赋值为一个 SampledSoundList 对象：

```
In[11]= ssl = VoiceData["你好", "Rate" -> 0, "Volume" -> 100, "SampleRate" -> 44100]
```

Out[11]=

```
SampledSoundList[{0.00100711, 0.00112918, 0.00122074, 0.0013123,
  ... 24963 ..., 0.00103763, 0.00103763, 0.00103763, 0.00100711}, 44100]
```

大型输出 显示更少 显示更多 显示全部 设定大小限制...

在 MMA 中，SampledSoundList 对象可以被

▲ 赞同 45



● 6 条评论

➤ 分享

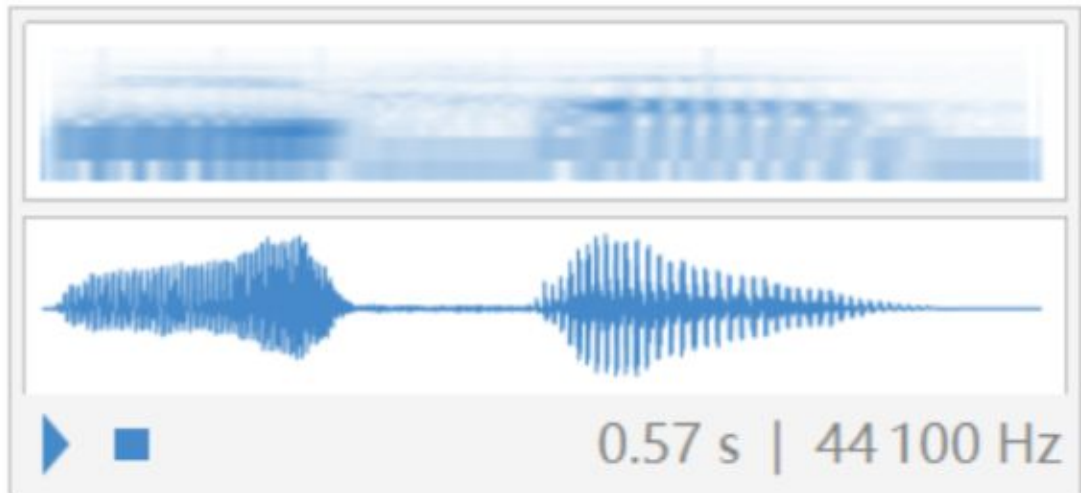
★ 收藏



```
In[12]:= Sound[ss1]
```

声音

Out[12]=



点击左下角的三角形播放键就可以听到 Cortana 姐姐说“你好”了。

2.2 抑扬顿挫

声音的三要素是音调、音色和响度，人类语音自然也不例外。虽然麦酱现在获得了录音带，直接听还是听不懂 Cortana 姐姐在说什么，哪怕是汉语中最简单的“你好”，于是开始思考如何把三要素信息从这些数据中提取出来。

我们首先考虑音调。无论什么语言，“抑扬顿挫”都是体现说话者语气和情绪的关键。在汉语中，语调的重要性相对更高，因为不同的“平仄”几乎直接对应不同的汉字，表示完全不同的信息。在物理中，音调对应的是频率，更准确的说，是基本频率，简称**基频 (fundamental frequency)**。在 Vocaloid 中，这便是 PIT 参数。


在 MMA 中，AudioLocalMeasurements 函数为这种测量提供了可能，首先打开帮助页面：

`AudioLocalMeasurements[audio, "prop"]`
为 `audio` 分区计算局部属性 `"prop"`.

`AudioLocalMeasurements[audio, {"prop1", "prop2", ...}]`
计算数个属性 `"propi"`.

`AudioLocalMeasurements[audio, "prop", format]`
在指定输出 `format` 中返回测量.

➤ 更多信息和选项



展开"更多信息和选项"，找到"频率方面属性"：

• 频率方面属性：

"FundamentalFrequency"	估算基础频率
"Formants"	信号的共振峰频率
"HighFrequencyContent"	功率谱的线性加权平均值
"MFCC"	梅尔频率倒谱系数

果然提供了基频的测量选项，再往下还有个说明：

- 使用 `{"FundamentalFrequency", t, minfreq, maxfreq}`，仅返回频率范围在 `minfreq` 和 `maxfreq` 之间置信区间为 `t` 或更高的频率. 默认值对包括语音和乐器的信号优化.

其中参数 `t` 的含义比较令人费解，经过研究，它体现的是灵敏度，值越小对快速变化的信号等越不敏感，一般取 0.5 左右比较合适。对于 Cortana 这样的女声音源，`minfreq` 一般取 100 左右，`maxfreq` 一般取 400 左右。

于是我们把这些经验性的参数都封装一下，得到如下的 FBase 函数：

```
Options[FBase] = {"Range" -> {0.4, 110, 440},
  "Partition" -> {1024/44100, 512/44100, HannWindow}};
FBase[ssl_, OptionsPattern[]] :=
  AudioLocalMeasurements[ssl,
    Flatten[{"FundamentalFrequency",
```

▲ 赞同 45

▼

6 条评论

分享

★ 收藏

知乎



首发于

Mathematica 还能这样玩

其中 "Range" 选项的三个参数分别依次对应上述的 t , minfreq 和 maxfreq. "Partition" 参数表示对音频的划分和加窗, 经过大量实验, 它们的默认值基本可以达到最好的效果, 因此调用时可不做任何修改。比如对之前那段"你好" (变量ssl), 我们直接调用函数:

```
fbase=FBase[ssl]
```

运行后变量 fbase 被赋值一个 TimeSeries 对象:

```
In[20]:= fbase = FBase[ssl]
```

```
Out[20]= TimeSeries[  Time: 0. to 0.557  
Data points: 49 ]
```

在 MMA 中, TimeSeries 对象可以被 ListPlot 或 ListLinePlot 等函数直接作用, 直接生成图像, 这便是基频 (Hz) - 时间 (s) 曲线 (Vocaloid里的PIT曲线)

```
ListLinePlot[fbase]
```

▲ 赞同 45 ▼

● 6 条评论

➤ 分享

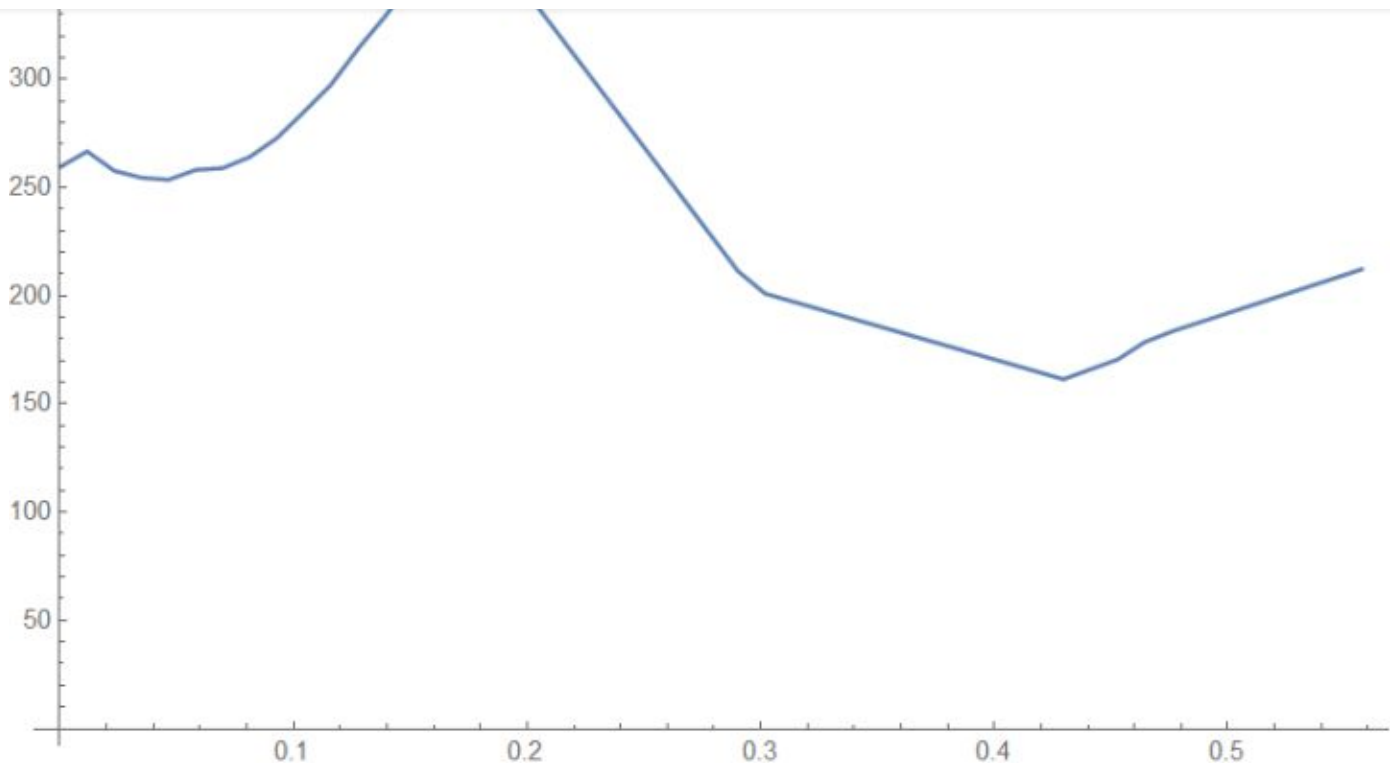
★ 收藏

知乎



首发于

Mathematica 还能这样玩



"你好" (Cortana) 的 PIT 曲线

从上图可以明显看出一个双“3声”结构，跟汉语拼音长得是一模一样。注意到“你”字“3声”开头的下降段非常短促乃至非常接近“2声”，这是汉语口语的连读习惯。

2.3 听音识人识字

音调考虑完了，我们来考虑音色。对于人类语言来说，音色有两个最大的作用：

1. 区分不同的讲话者
2. 区分不同的音素

其中第2条直接决定我们为什么能听到“你好”。不同的元音或辅音，在频域上各个频段的能量分配差别是很大的。在激励源（比如声带）不变的情况下，这种能量分配主要由发声腔体的物理结构决定（比如是张大嘴还是抿着嘴），可以近似认为与激励源本身的频率（语调）无关。

现在我们不得不用数学来精确描述这个模型了。设 t 时刻激励源的频谱系数为关于频率 f 的两个函数 $A(t, f)$ 和 $\phi(t, f)$ ，分别表示振幅和相位。再设 t 时刻时腔体结构对频率 f 的放大作用（共振）为 $g(t, f)$ ，最终的波形是 $a(t)$ ，则可以认为：

▲ 赞同 45



● 6 条评论

➤ 分享

★ 收藏



相当于认为跟腔体共振有关的 g 与跟激励源有关的 A 和 ϕ 是独立的，可以分别写开。这个近似对于元音是非常好的，因为根据语音学的定义，元音在发音过程中产生的气流不会摩擦腔体，意味着激励源与腔体之间不会产生无规则的相互作用（即辅音中的噪声成分，比如齿音）。

由于任一小段时间内激励源产生的是有周期的信号，傅里叶变换的结果 A 应该是离散的，也就是说：

$$A(t, f) = \sum_{k=1}^{\infty} a_k(t) \delta(f - k f_{base}(t))$$

这里的 f_{base} 就是上一节测出来的 PIT 曲线， $a_1, a_2 \dots a_n$ 描述的是离散化后的傅里叶级数中不同倍频的振幅大小。

将它代入原式，在狄拉克 δ 函数的作用下连续的积分会被转化为离散求和：

$$a(t) = \sum_{k=0}^{\infty} g(t, k f_{base}(t)) a_k(t) \cos[2\pi f_{base}(t)t + \phi(t, k f_{base}(t))]$$

激励信号一般具有很强的随机性。为了进一步简化模型，我们假设激励源在不同倍频处的平均能量相等（类似于白噪声），即给定 t 时刻不同的 a_k 都当成一个定值，这样我们可以直接把所有的 a_k 都可以吸收至前面的 g 函数中，记为 G 函数：

$$a(t) = \sum_{k=0}^{\infty} G(t, k f_{base}(t)) \cos[2\pi f_{base}(t)t + \phi(t, k f_{base}(t))]$$

这便是语音的**正弦模型**。给定 t 时刻，关于 f 的函数 $G(t, f)$ 称为该时刻的**共振曲线**。

在 MMA 中，并没有测量共振曲线相关的函数，我们只能自己构造了。

```
WindowList>windowfunction_, n_] :=
  WindowList>windowfunction, n] =
    Array>windowfunction, n, {-0.5, 0.5}];
LSFKernel[scanf_, n_, samplerate_] :=
  LSFKernel[scanf, n, samplerate] =
    LeastSquaresFilterKernel[{"Bandpass", (2 \[Pi] scanf)/samplerate},
      n];
Formant[ssl_, fbase_, nwidth_: 2048,
  windowfunction_: HannWindow, fwidth_: 1024]
```

[赞同 45](#)

[6 条评论](#)
[分享](#)
[★ 收藏](#)

知乎



首发于

Mathematica 还能这样玩

```

LineFit = Interpolation[#, InterpolationOrder -> 1] &;
datalist = Partition[ssl[[1]], nwidth, nshift, {-nwidth, 1}, 0];
wdatalist = WindowList>windowfunction, nwidth]*# & /@ datalist;
amplist =
  MapThread[
    Sqrt[2/nwidth*Total[#1^2]/Total[#2^2]] &, {datalist, wdatalist}];
tlist = (nshift (Range@Length[datalist] - 1) + nwidth/2)/ssl[[2]];
func = LineFit@MapThread[{#1, LineFit@
  Table[{f, #2 Total[
    ListConvolve[
      LSFKernel[
        If[f == 0, {0, fwidth/2},
          fshift*Floor[(f - fwidth/2)/fshift + 1/2] + {0,
            fwidth}], nwidth, ssl[[2]]], #3,
        Floor[(nwidth + 1)/2], 0]^2]^0.5}, {f, 0, flimit,
          Quiet@fbase[#1]]}] &, {tlist, amplist, wdatalist}];
Quiet[Expand[func[#1]][#2] /. (a_*func1_ + b_*func2_)[#2] :=>
  a*func1[#2] + b*func2[#2]] &;

```

构造出的 Formant 函数将直接计算得到的正弦模型中的 G，大概思路是首先对音频输入信号划分（Partition函数）并加窗（乘上 windowfunction），然后根据前面计算过的基频利用最小方均 FIR 滤波器（LeastSquaresFilterKernel 函数）分离各个频段的信号（带通滤波），计算能量值并拟合出共振曲线。

你可能注意到 Formant 函数有非常多的参数，不过跟上一节一样，经过大量实验，它们的默认值基本可以达到最好的效果，因此调用时可不做任何修改。因此只需要前两个变量作为输入，分别是对应语音信号 SampledSoundList 对象（ssl）和一个 PIT 曲线（fbase），比如我们将“你好”的 ssl 和 fbase 作为输入：

```
G = Formant[ssl, fbase];
```

然后我们可以创建个动态来观察共振曲线随时间的变化情况：

```

Manipulate[
  Plot[20 Log10@G[t, f], {f, 0, 8000},
    PlotRange -> {{0, 8000}, {-100, 0}}, {t, 0, 0.57}]

```

为了更好的观察，建议采用的分贝坐标来观察共振曲线的变化，即做一个 G 的 20 倍对数操作，这里就不上传 GIF 结果了，只给一个截图作为举例：

▲ 赞同 45



● 6 条评论

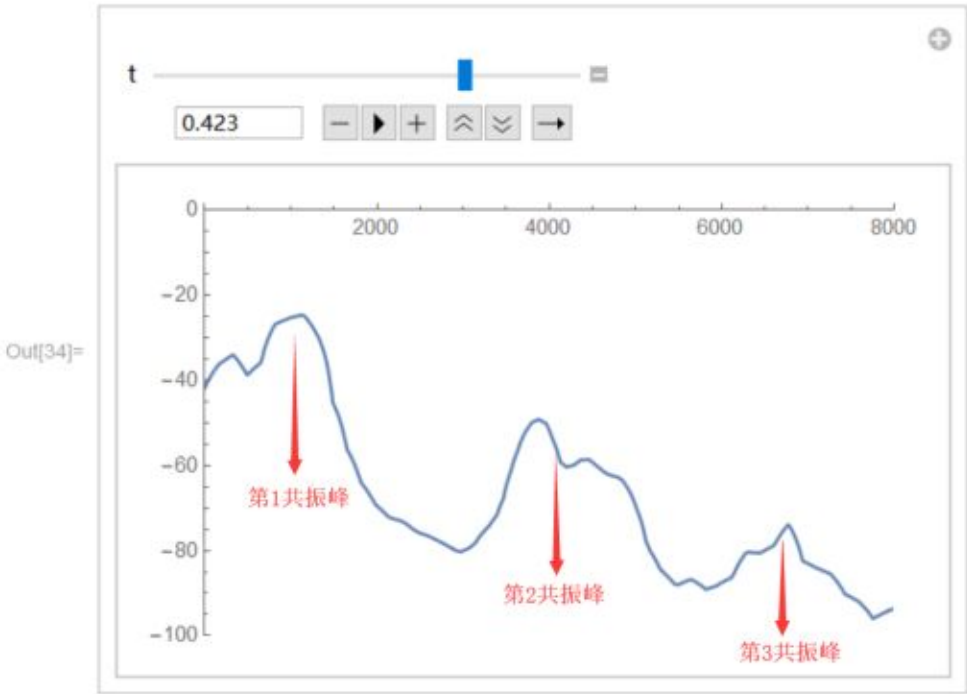
➦ 分享

★ 收藏

知乎



首发于
Mathematica 还能这样玩



元音“ao”(Cortana) 的共振曲线

上图取自“你好”音频的第 0.423 秒，此时 Cortana 正在发“ao”这个元音，可以看出有3个共振峰，分别在 1000 Hz, 4000 Hz 和 7000 Hz 左右。

总结

经过大量的听力训练，麦酱现在已经大体上熟悉了 Cortana 姐姐的声音（建模完成），终于能听懂讲话的内容了！这为之后的首次歌唱做出了关键铺垫。

发布于 2018-02-12

- Wolfram Mathematica
- 语音合成
- DSP（数字信号处理）

文章被以下专栏收录



Mathematica 还能这样玩
Mathematica（mma、麦酱），宇宙第一

已关注

▲ 赞同 45 ▼

● 6 条评论

➤ 分享

★ 收藏

知乎



首发于

Mathematica 还能这样玩



👍 赞

▲ 赞同 45



💬 6 条评论

➦ 分享

★ 收藏