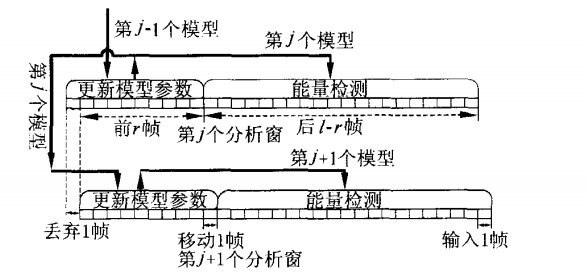
# EPD源代码概要说明

## 1 频域能量判断模块

首先分帧之后，由L帧语音构成一个分析窗，根据其中前r帧信号估计模型参数，并设定能量阈值C:\Users\thinkit\AppData\Local\Temp\1529025393(1).png，(其中C:\Users\thinkit\AppData\Local\Temp\1529025433(1).png是灵敏度系数，C:\Users\thinkit\AppData\Local\Temp\1529025457(1).png，可根据具体情况适当调节)，则可根据阈值，检验后l—r帧信号的能量是否平稳。每输入一帧信号，分析窗就后移一帧，并校正分布模型，重新计算C:\Users\thinkit\AppData\Local\Temp\1529025500(1).png。之后根据特定算法[1]对该模型进行初始化以及更新，这样就得到了一个背景噪声的模型。之后根据每一个分析窗的信号谐波特征，选取5条相邻的谐波，之后根据这五条谐波是否满足能量大于阈值的要求，且都大于噪声，如果满足则判定语音信号的起点在该分析窗中。大致流程如下图：

* 实现源文件：DetectEnergy.h DetectEnergy.cpp
* 类名：CDetectEnergy\_EPD
* 主要函数说明：·

（1）CDetectEnergy\_EPD(int SampleRate);

功能：

频域能量计算构造函数，初始化变量。

形参：

int SampleRate：输入语音采样率

返回值：无

（2）void InitNoise();void ReInit();

功能：

分别是初始化噪声模型以及重置能量模块。

形参：无

返回值：无

（3）int InitSensitivity(int FrameCount\_in);

功能：

初始化能量模块自适应灵敏度。

形参：

int FrameCount\_in：输入语音帧数

返回值：1

（4）double GetNewEnergy(float \*BinEnergy\_in, int FrameCount\_in);

功能：

计算各个子带的能量并存储,并返回该帧的总能量。

形参：

float \*BinEnergy\_in：输入语音频域能量

int FrameCount\_in：输入语音帧数

返回值：

该帧的总能量

（5）void UpdateNoise(int FrameCount\_in);

功能：

根据语音情况更新噪音均值、方差（寻找起点时）。

形参：

int FrameCount\_in：输入语音帧数

返回值：无

（6）void UpdateNoiseInSpeech(int FrameCount\_in);

功能：

根据语音情况更新噪音均值、方差（寻找尾点时）。

形参：

int FrameCount\_in：输入语音帧数

返回值：无

（7）int DetectRoughStart(int FrameCount\_in);

功能：

根据频域能量确定语音粗起点。

形参：

int FrameCount\_in：输入语音帧数

返回值：

语音起点

（8）int DetectRoughStart(int FrameCount\_in);

功能：

根据频域能量确定语音粗起点。

形参：

int FrameCount\_in：输入语音帧数

返回值：

语音起点

（9）int DetectRoughEnd(int FrameCount\_in);

功能：

根据频域能量确定语音尾点。

形参：

int FrameCount\_in：输入语音帧数

返回值：

语音尾点

## 2 浊音（谐波）判断

浊音是由周期性声门脉冲激励声道而产生的。对纯净浊音信号加窗，则加窗信号在基音及谐波频率存在能量峰起。因为基频变化较缓慢，此特性在窄带语谱图上表现为一系列均匀平行亮线。由于多数噪声不具有这种谐波特性，所以通过在输入信号中检测浊音谐波特性，可以检出浊音信号。

基音和谐波集中了浊音的主要能量，所以浊音在噪声干扰下仍能保持清晰的谐波特性。但是，语音的频域能量分布随基音和共振峰而变化，噪声的频域能量则随环境而改变。因此，在实际噪声环境中，语音的各频率分量所受干扰程度不同，并且处在不断变化中，浊音谐波特征较清晰的频段也随之变化。本算法根据人类浊音特性，在较宽的频率范围中自动搜索最清晰的谐波特征，并结合前后信息判断浊音是否存在。这样不仅可避免低频噪声和基音倍频的干扰，而且能自动适应噪声类型和强度的变化。

* 实现源文件：DetectVoice.h DetectVoice.cpp
* 类名：CDetectVoice\_EPD
* 主要函数说明：·

（1）CDetectVoice\_EPD(int SampleRate);

功能：

谐波判断构造函数，初始化变量。

形参：

int SampleRate：输入语音采样率

返回值：无

（2）int DetectVoice(float \*AllBinEnergy\_in, int VoiceNumber\_in);

功能：

接口函数，接受数据，返回是否有谐波。

形参：

float \*AllBinEnergy\_in：输入语音各频带能量

int VoiceNumber\_in：能组成谐波最短的语音帧数

返回值：

存在谐波返回1，否则返回0

（3）void UpdateBuf();

功能：

更新谐波检测的内部频带能量缓存。

形参：无

返回值：无

（4）void MatchPitch();

功能：

对当前一帧的局部极值做基本的匹配，看是否可以基本符合某些谐波。

形参：无

返回值：无

（5）int Find(int VoiceNumberRequest);

功能：

连接匹配出来的局部极值，如果超过阈值则认为语音中存在谐波。

形参：

int VoiceNumberRequest：谐波帧数阈值

返回值：

存在谐波返回1，否则返回0

## 3起点细查找

起点细查找，若在分析窗内检测到浊音，则结合能量检测的结果，通过能量细搜索寻找语音起点。从b开始，向前搜索语音起点帧b0，它满足如下条件：从b0的下一帧开始，信号能量逐帧递增。由于语音能量较大且变化迅速，所以语音起点附近的能量起伏很大，不会引起噪声模型的变化，能量阈值也不会升高。由于清音位于浊音前，且浊音能量远大于清音，所以根据前后h帧平均能量所找到的起点b一般处在比阈值能量更低的位置，即清音段或清浊音交界处，满足上述条件的b0就是语音起点。

* 实现源文件：DetectEnergy.h DetectEnergy.cpp
* 类名：CDetectVoice\_EPD
* 主要函数说明：·

（1）RefineStart(int FrameCount\_in, int FirstVoice\_in);

功能：

起点细查找

形参：

int FrameCount\_in：浊音开始位置

int FirstVoice\_in：当前搜索到的帧数

返回值：

真实的起点位置