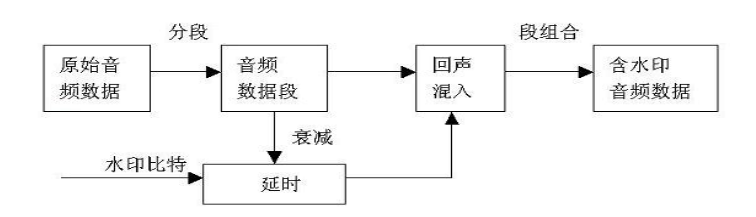
**基于回声的音频水印算法**

利用回升嵌入水印的算法是一种经典的音频水印算法。它利用了人类听觉系统的一个特性：音频信号在时域的后向掩码作用，即弱信号在强信号消失之后变得无法听清。

1. **嵌入算法**

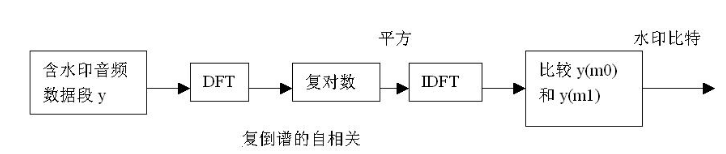
设音频序列，按照下式即可得到含有回声的音频序列：

其中m是信号和回声间的延时，一般取m≤N，λ为衰减系数。

在回声编码中通过修改m来嵌入水印信息，水印嵌入流程图如下：

对音频数据文件，先将其分成若干包含相同样点数的片段，每个片段时间约为几个到几十个毫秒，样点数记为N。每段用来嵌入一比特的水印信息。在水印嵌入阶段，对每段信号使用前面的回声序列构造方式，选择则在信号中嵌入比特0，选择则在信号中嵌入比特1。延时和是以人耳听不到回声信号为准则选取，最后将所有回声信号段串联成连续信号。

1. **检测算法**

回声编码中水印提取流程如下图：

对一个音频回声信号，水印提取关键在于确定回声的延时。利用复倒谱可将回声从原始信号中分离出来。设回声信号的复倒谱自相关为。由于引入回声的延时只有和两种可能，因此，只有比较在和处的取值，根据其中较大者即可判断回声延时，从而确定嵌入的一比特消息。

1. **程序实现**

本实验在win10，matlab2016平台下完成

* 1. **嵌入算法**

算法采用λ = 0.2，，作为嵌入参数，嵌入消息为”the final work is completed by helinfei”，在matlab中嵌入消息的长度为273位。

%% read audio

[aud, fs] = audioread('original.wav');

[M, N] = size(aud);

%% segmentation

leng = 4400;

aud\_s = cell(floor(M / leng),1);

for i = 1 : floor(M / leng)

aud\_s{i,1} = aud( leng \* ( i - 1 ) + 1: leng \* i, : );

end

%% embeding

lamda = 0.2;

m0 = 200;

m1 = 300;

msg= 'the final work is completed by helinfei';

msg\_bin = dec2bin(msg);

[R,Q] = size(msg\_bin);

k = 1;

for i = 1:R

for j = 1:Q

if(msg\_bin(i,j))

m = m1;

else

m = m0;

end

aud\_s{k,1}( m + 1 : leng, :) = aud\_s{k,1}( m + 1 : leng, :) + lamda \* aud\_s{k,1}( 1 : leng - m, :);

k = k + 1;

end

end

%% stitching

aud\_o = [];

for i = 1 : floor(M / leng)

aud\_o = cat(1,aud\_o, aud\_s{i,:});

end

aud\_o = [aud\_o; aud( floor(M / leng) \* leng + 1 : end, :)];

%% write/output

audiowrite('embeded.wav', aud\_o, fs);

* 1. **检测算法**

%% load embeded audio

load msg.mat

[aud,fs] = audioread('embeded.wav');

[M, N] = size(aud);

%% detect

L = 273; % length of the message

leng = 4400;

aud\_s = cell(L,1);

for i = 1 : L

aud\_s{i,1} = aud( leng \* ( i - 1 ) + 1: leng \* i, : );

end

msg = zeros(L, 1);

for i = 1 : L

XHAT=abs(ifft(log((abs(fft(aud\_s{i,1}'))).^2)));

if( XHAT(201) > XHAT(301) )

msg(i) = 1;

end

end

%% output message

k = num2str(msg);

s = reshape(msg\_bin, 273,1);

accuracy = length(find(k == s))/273

1. **实验结果**

嵌入算法输出embeded.wav文件，通过人耳测试，和原文件没差别，嵌入效果很好。

检测算法通过比较输出信息与原嵌入信息，再除以信号长度得到算法的精确度。精确度为accuracy = 0.9123，并不高。

1. **总结与展望**

嵌入算法效果很好，但检测算法并不能做到100%的提取，有待改进。