

基于MATLAB的语音信号去 噪

杨跃浙 WA2214014

杨昀川 WA2214022



实验原理

IIR 滤波器设计原理

FIR 滤波器设计原理

谱减法滤波

卡尔曼滤波

实验记录

IIR 滤波器

自带的IIR 滤波器

FIR 滤波器

自带的FIR 滤波器

谱减法滤波

卡尔曼滤波



技术指标

- 在设计 IIR 滤波器之前，需要明确滤波器的设计目标和性能要求。以下是常见的设计技术指标：
- 通带截止频率 f_p ， 阻带截止频率 f_s ： 滤波器允许信号通过或被衰减的频率界限。
- 通带波动 (Passband Ripple, δ_p) ： 通带范围内信号允许的最大幅度波动
- 阻带衰减 (Stopband Attenuation, δ_s) ： 阻带范围内信号的最小衰减量
- 过渡带宽 (Transition Bandwidth, Δf) ： 通带和阻带之间的频率范围，越窄意味着更高的滤波器阶数。
- 采样频率 (Sampling Frequency, F_s) ： 离散信号的采样频率，影响滤波器设计的归一化频率。技术指标通常通过以下参数表示：



巴特沃斯滤波器计算阶数

- 滤波器阶数 n 决定了滤波器的陡峭程度，其计算公式为：

$$n = \left\lceil \frac{\log \left(\sqrt{\frac{10^{0.1\delta_s} - 1}{10^{0.1\delta_p} - 1}} \right)}{2 \log \left(\frac{\omega_s}{\omega_p} \right)} \right\rceil$$

- 其中：
- $\omega_p = 2 f_p / F_s$ ：归一化通带频率。
- $\omega_s = 2 f_s / F_s$ ：归一化阻带频率。
- $\lceil \cdot \rceil$ ：向上取整，确保阶数为整数。



巴特沃斯滤波器设计步骤

■ 1. 确定技术指标

- 通带截止频率 $f_p = 3000$ Hz。

- 通带波动 $\delta_p = 1$ dB。

■ 2. 计算归一化频率:

- 阻带截止频率 $f_s = 5000$ Hz。

- 阻带衰减 $\delta_s = 70$ dB。

- 采样频率 $F_s = 44100$ Hz。

$$\omega_p = \frac{f_p}{F_s/2} \approx 0.136, \quad \omega_s = \frac{f_s}{F_s/2} \approx 0.227$$

■ 3. 使用公式计算滤波器阶数:

$$n = \left\lceil \frac{\log \left(\sqrt{\frac{10^{0.1 \cdot 70} - 1}{10^{0.1 \cdot 1} - 1}} \right)}{2 \log \left(\frac{0.227}{0.136} \right)} \right\rceil = 4$$

■ 4. 设计滤波器原型。

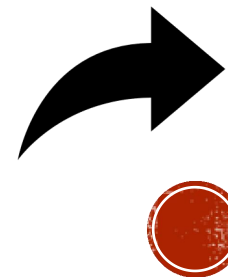
■ 5. 根据频率要求完成滤波器变换，生成最终的低通滤波器。



巴特沃斯滤波器的MATLAB 实现

```
elseif strcmp(filter_type, 'IIR滤波器')
    % 使用 `buttord` 计算IIR滤波器的阶数和截止频率
    [iir_order, wn] = buttord(passband_cutoff/(app.fs/2), stopband_cutoff/(app.fs/2), passband_attenuation, stopband_attenuation);

    % 使用计算出的阶数和截止频率设计 Butterworth 低通滤波器
    [b, a] = butter(iir_order, wn); % 使用计算的阶数和归一化截止频率
    app.filtered_audio = filter(b, a, app.noisy_audio);
```



技术指标

- 采样频率 (f_s): 信号的采样频率, 决定信号频谱的范围 ($-f_s/2$ 到 $f_s/2$) 。
- 通带截止频率 (f_p): 滤波器在通带内信号无损通过, 截止频率 f_p 定义通带的上限。
- 阻带截止频率 (f_s): 滤波器对高于 f_s 的频率信号进行完全衰减。
- 通带波动 (δ_p): 滤波器在通带内允许的幅度波动 (以 dB 表示) 。
- 阻带衰减 (δ_s): 滤波器在阻带内要求的最小衰减 (以 dB 表示) 。
- 过渡带宽 (Δf): 通带和阻带之间的过渡宽度, 定义为 $\Delta f = f_s - f_p$ 。



窗函数法设计 FIR 低通滤波器

窗函数	过渡带宽	阻带衰减
矩形窗	$1.8\pi/N$	21 dB
汉宁窗	$6.2\pi/N$	44 dB
海明窗	$6.6\pi/N$	53 dB
布莱克曼窗	$11\pi/N$	74 dB

表 1: 常见窗函数的特性比较



滤波器阶数计算

滤波器的阶数 N 与窗函数的特性及过渡带宽直接相关。为避免单位冲突，应首先对过渡带宽 Δf 进行归一化。归一化的过渡带宽定义为：

$$\Delta f_{\text{norm}} = \frac{\Delta f}{f_{\text{sampling}}/2}$$

其中： f_{sampling} 是采样频率； Δf 是通带和阻带之间的过渡带宽。滤波器阶数的公式应修正为：

$$N \geq \frac{k}{\Delta f_{\text{norm}}}$$

其中 k 是窗函数的过渡带特性常数。



FIR 低通滤波器设计步骤

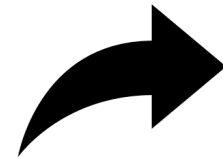
- 1. 确定技术指标，包括采样频率、通带截止频率、阻带截止频率、过渡带宽、通带波动和阻带衰减。
- 2. 对过渡带宽 Δf 进行归一化
- 3. 选择窗函数，根据阻带衰减和过渡带宽平衡性能与复杂度。例如：高阻带衰减要求时，选择布莱克曼窗。适中性能要求时，选择汉宁窗或海明窗。简单设计场景可选择矩形窗。
- 4. 根据窗函数的特性计算滤波器阶数 N 。
- 5. 生成理想的低通滤波器冲激响应：

$$h_d[n] = \begin{cases} \frac{\sin(2\pi f_c n)}{\pi n}, & n \neq 0 \\ 2f_c, & n = 0 \end{cases}$$

- 6. 使用窗函数 $w[n]$ 对冲激响应 $h_d[n]$ 进行加权修正：
$$h[n] = h_d[n] \cdot w[n]$$
- 7. 验证滤波器性能，绘制频率响应并检查是否满足设计要求。



FIR 滤波器的MATLAB 代码实现



```
if strcmp(filter_type, 'FIR滤波器')
    % 根据阻带衰减选择最简单的窗函数
    if stopband_attenuation <= 21
        window_type = 'rectwin';
        transition_band = 1.8 * pi;
    elseif stopband_attenuation <= 44
        window_type = 'hann';
        transition_band = 6.2 * pi;
    elseif stopband_attenuation <= 53
        window_type = 'hamming';
        transition_band = 6.6 * pi;
    else
        window_type = 'blackman';
        transition_band = 11 * pi;
    end

    % 计算滤波器阶数
    delta_f = (stopband_cutoff - passband_cutoff) / app.fs; % 归一化过渡带宽
    order = ceil(transition_band / delta_f); % 计算阶数
    if mod(order, 2) == 0
        order = order + 1; % 确保阶数为奇数
    end

    % 理想低通滤波器的冲激响应
    n = 0:order;
    center = floor(order / 2);
    h_low = (sin(2 * pi * passband_cutoff * (n - center) / app.fs) ./ (pi * (n - center)));
    h_low(center + 1) = 2 * passband_cutoff / app.fs; % 修正中心点, 避免除以0
```



谱减法的基本原理

- 谱减法是一种经典的语音信号降噪方法，尤其适用于宽带噪声（如高斯白噪声）的抑制。其基本假设是，语音信号和噪声在频域上是相互独立的。带噪信号可以建模为语音信号与噪声的叠加：

$$y(t) = x(t) + d(t)$$

其中：

- $y(t)$: 观测到的带噪信号；
- $x(t)$: 目标语音信号；
- $d(t)$: 噪声。



- 通过傅里叶变换，将信号转换到频域：

- $$Y(f) = X(f) + D(f)$$

- 其中：

- $Y(f)$: 带噪信号的频谱；

- $X(f)$: 目标信号的频谱；

- $D(f)$: 噪声的频谱。

- 谱减法的核心思想是，通过减去估计的噪声频谱 $|D(f)|$ ，恢复语音信号的频谱幅值：

- $$|X(f)| = |Y(f)| - |D(f)|$$

对于功率谱形式，有：

$$|X(f)|^2 = |Y(f)|^2 - |D(f)|^2$$

- 最终，通过对估计的 $X(f)$ 进行反傅里叶变换，恢复时域的去噪信号。



谱减法的基本算法

- (1) 带噪信号的短时傅里叶变换 (STFT)
- 将带噪信号划分为多个短时帧，利用短时傅里叶变换 (STFT) 将每一帧转换到频域：

$$Y(f, t) = \text{STFT}(y(t))$$

- 其中 t 表示时间帧的索引。
- (2) 噪声估计 在语音信号的静音段或非语音段中，估计噪声频谱 $|D(f)|$ 。假设此时语音信号为零，则 $|Y(f)|$ 可直接作为噪声频谱的估计。



- (3) 幅度谱减法 对每一帧 t , 执行以下减法:

$$|X(f, t)| = |Y(f, t)| - \alpha |D(f)|$$

- 其中, α : 噪声权重因子, 通常取值 $1 \sim 2$, 用于调整降噪程度。
- 当结果为负值时, 进行截断处理:

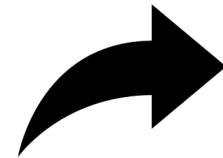
$$|X(f, t)| = \max(|Y(f, t)| - \alpha |D(f)|, 0)$$

- (4) 相位恢复
- 利用带噪信号的相位信息恢复目标信号频谱:

$$X(f, t) = |X(f, t)| e^{j\angle Y(f, t)}$$

- (5) 反短时傅里叶变换 (ISTFT)
- 对每帧的 $X(f, t)$ 执行逆变换, 叠加得到时域的去噪信号。





谱减法的MATLAB 实现

```
elseif strcmp(filter_type, '谱减法')
    % 谱减法降噪
    noisy_audio_fft = fft(app.noisy_audio);
    noise_estimate = mean(abs(noisy_audio_fft)); % 简单估计噪声频谱
    noise_subtracted_fft = max(abs(noisy_audio_fft) - noise_estimate, 0); % 谱减
    phase = angle(noisy_audio_fft); % 保留相位信息
    noise_subtracted_signal = noise_subtracted_fft .* exp(1i * phase); % 还原复数形式
    app.filtered_audio = real(ifft(noise_subtracted_signal)); % 还原时域信号
```



卡尔曼滤波基本原理

- 卡尔曼滤波器的核心思想是利用递归方法，在每一时刻结合系统的先验信息和测量信息，通过加权平均计算系统的最优状态估计。
- 系统的状态可以描述为以下形式的离散线性状态空间模型：

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

$$z_k = H_k x_k + v_k$$

- 其中：
- x_k ：系统在时刻 k 的状态向量；
- F_k ：状态转移矩阵，描述系统从时刻 $k-1$ 到 k 的状态变化；
- B_k ：控制输入矩阵，作用于控制量 u_k ；
- w_k ：过程噪声，假设服从零均值高斯分布 $w_k \sim N(0, Q_k)$ ，其中 Q_k 是过程噪声的协方差矩阵；
- z_k ：时刻 k 的观测值；
- H_k ：观测矩阵，将状态向量映射到测量空间；
- v_k ：测量噪声，假设服从零均值高斯分布 $v_k \sim N(0, R_k)$ ，其中 R_k 是测量噪声的协方差矩阵。



卡尔曼滤波器的递归过程

- (1) 预测阶段
- 根据状态转移方程和控制输入，预测系统在时刻 k 的状态 X_k 和误差协方差 P_k :

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

- 其中:
- $\hat{x}_{k|k-1}$ 时刻 k 的状态预测值;
- $P_{k|k-1}$ 状态预测的协方差矩阵。



- (2) 更新阶段

- 根据观测值 z_k , 结合预测值, 更新状态估计和协方差:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H_k \hat{x}_{k|k-1})$$

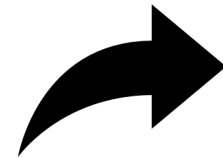
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

其中:

- K_k : 卡尔曼增益矩阵, 平衡预测与观测的权重;
- $\hat{x}_{k|k}$: 时刻 k 的状态估计值;
- $P_{k|k}$: 状态估计的协方差矩阵;
- $z_k - H_k \hat{x}_{k|k-1}$: 观测残差 (或称为创新)。



卡尔曼滤波器的MATLAB 实现



```
elseif strcmp(filter_type, '卡尔曼滤波')
    % 卡尔曼滤波降噪
    Q = 0.041; % 过程噪声
    R = 0.06; % 测量噪声
    N = length(app.noisy_audio); % 确保使用原始信号长度
    x_est = zeros(N, 1); % 初始化滤波后的信号数组
    P = 1; % 初始误差协方差
    x_est(1) = app.noisy_audio(1); % 使用第一个输入值初始化

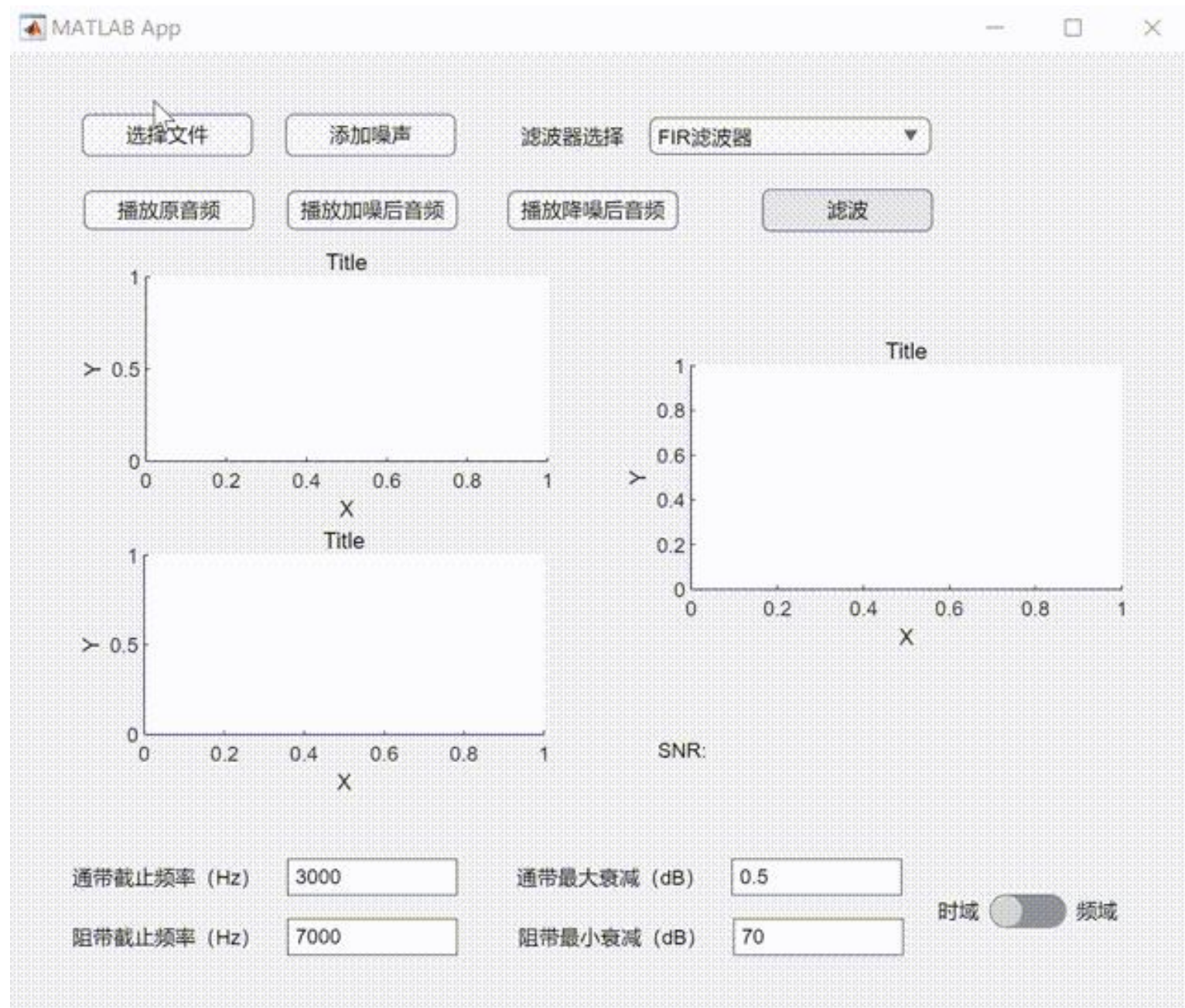
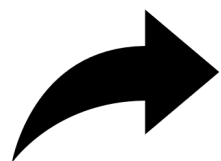
    for i = 2:N
        % 预测步骤
        x_pred = x_est(i - 1);
        P_pred = P + Q;

        % 更新步骤
        K = P_pred / (P_pred + R); % 计算卡尔曼增益
        x_est(i) = x_pred + K * (app.noisy_audio(i) - x_pred); % 更新估计值
        P = (1 - K) * P_pred; % 更新误差协方差
    end

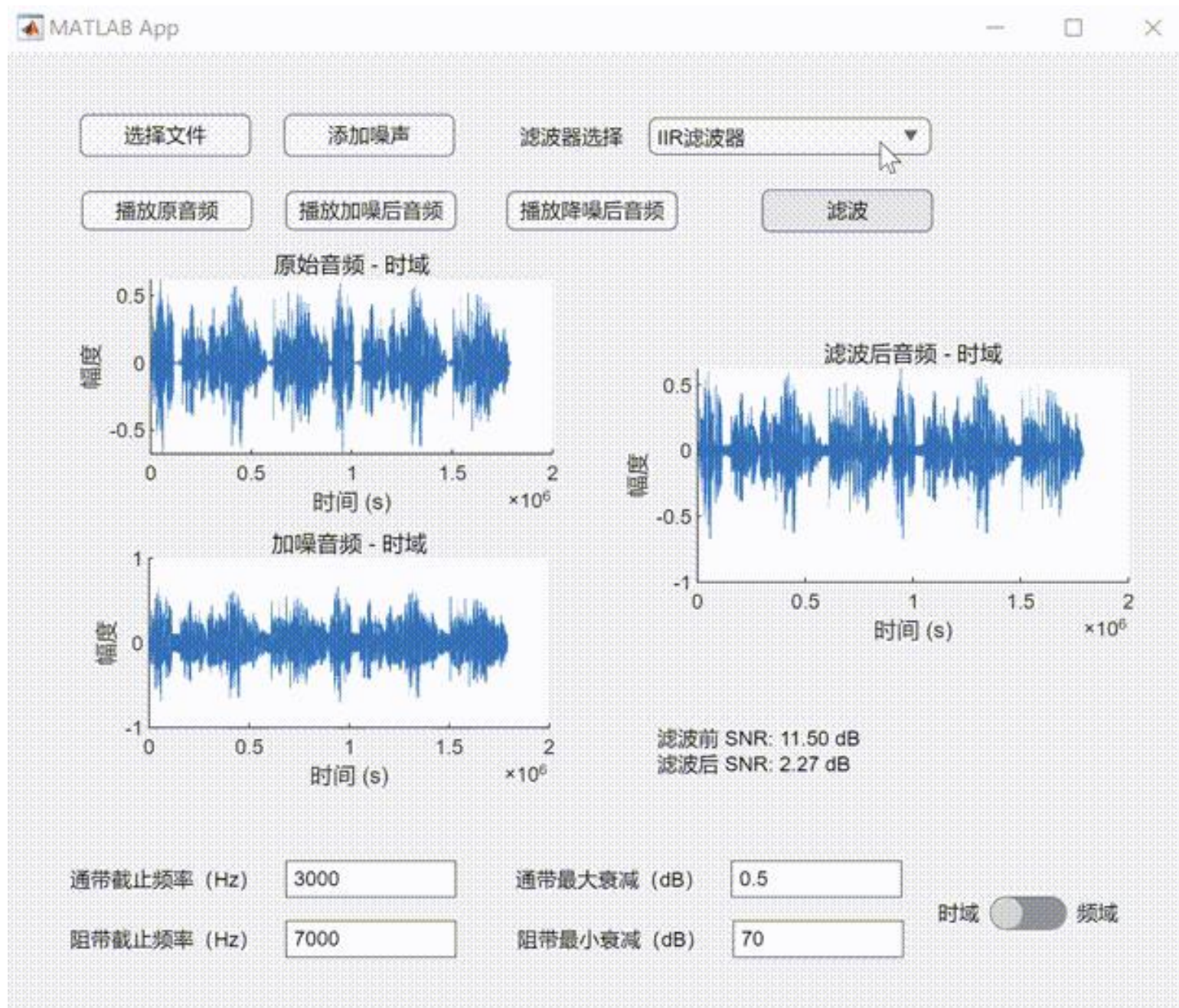
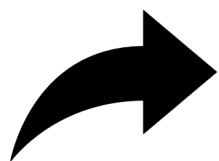
    % 确保输出长度与输入相同
    app.filtered_audio = x_est;
```



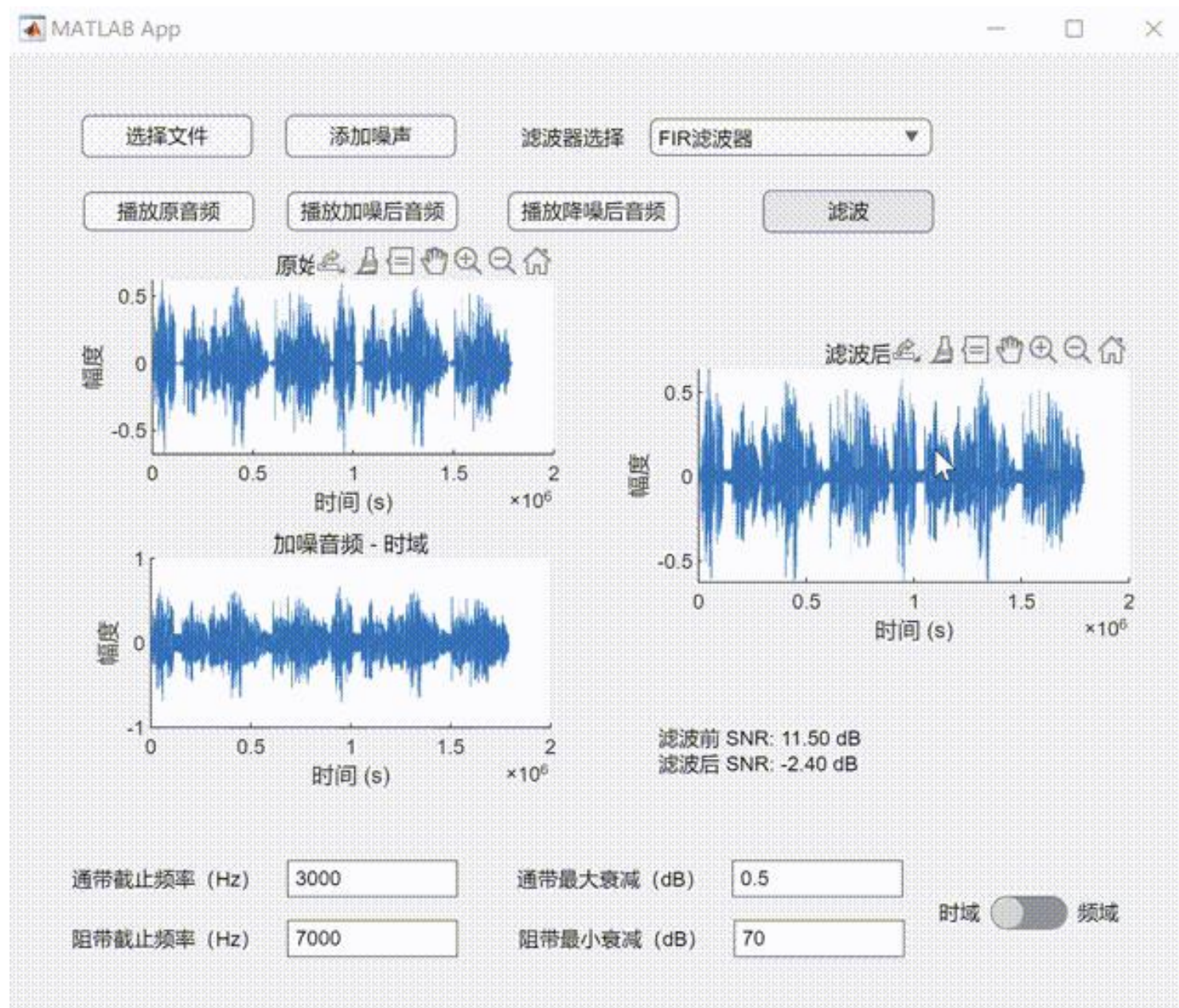
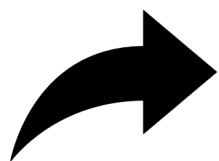
FIR 滤波器



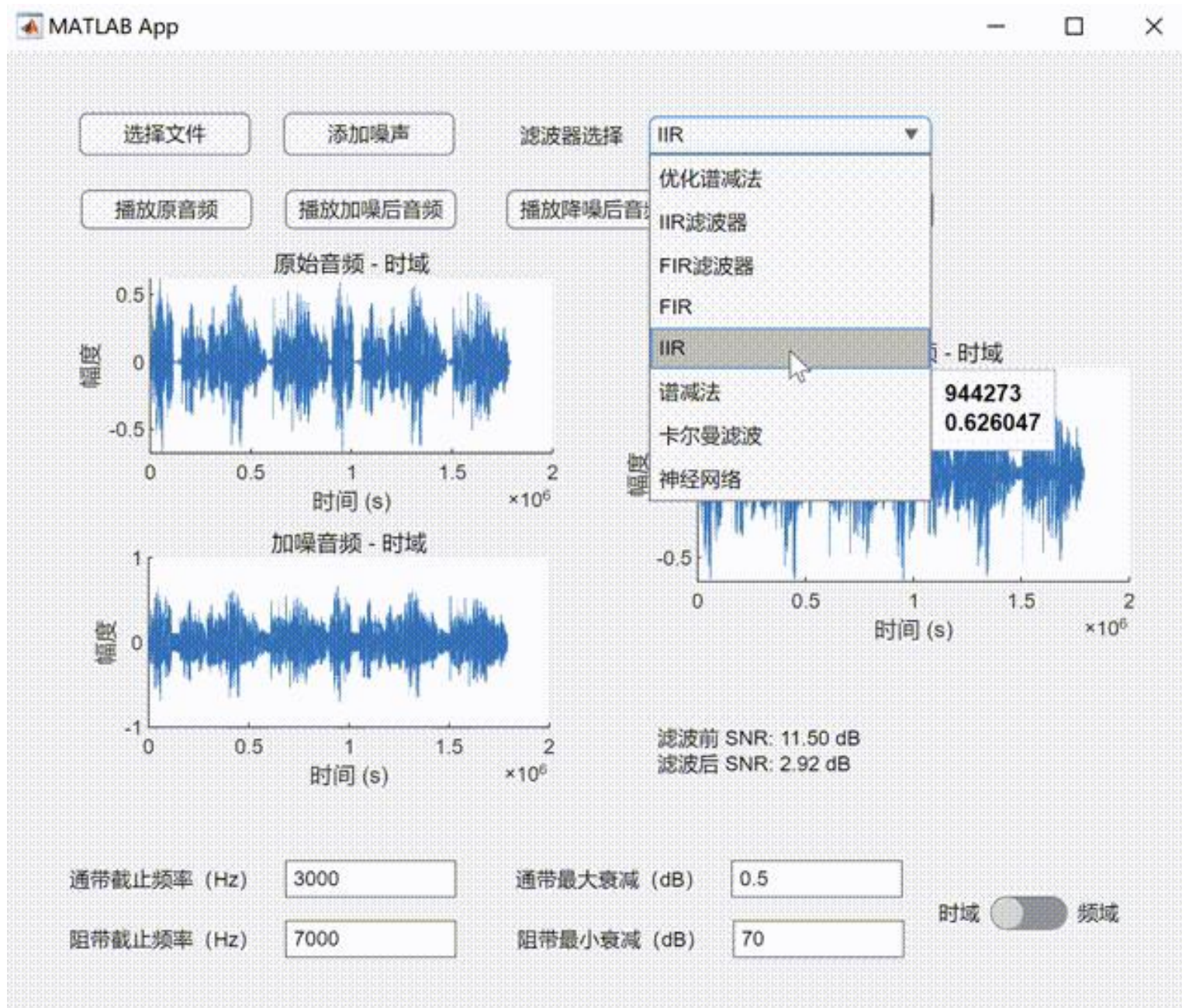
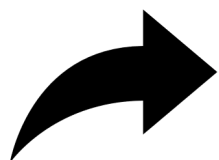
自带的 FIR 滤波器



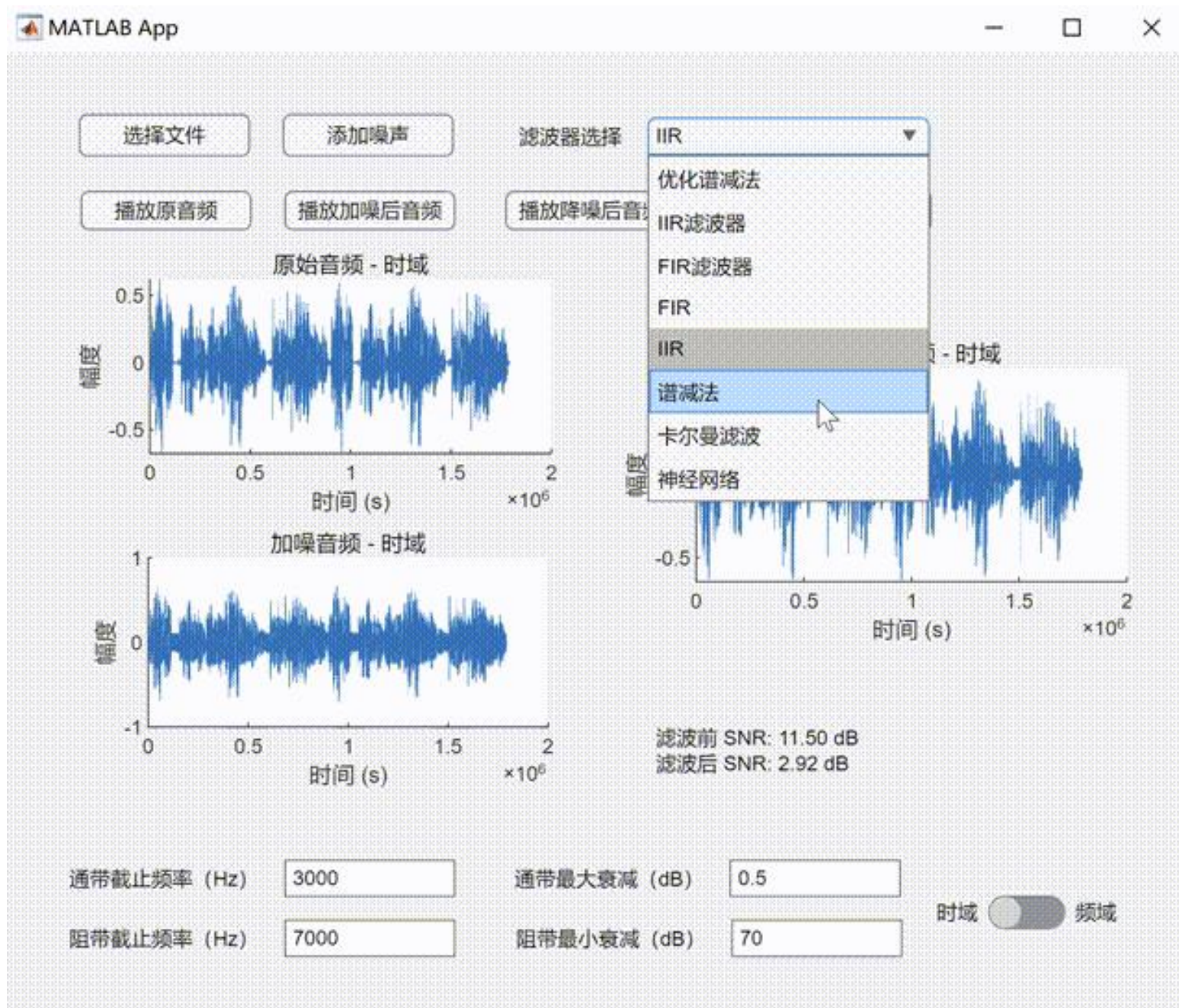
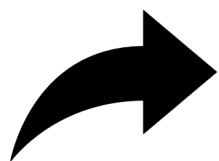
IIR 滤波器



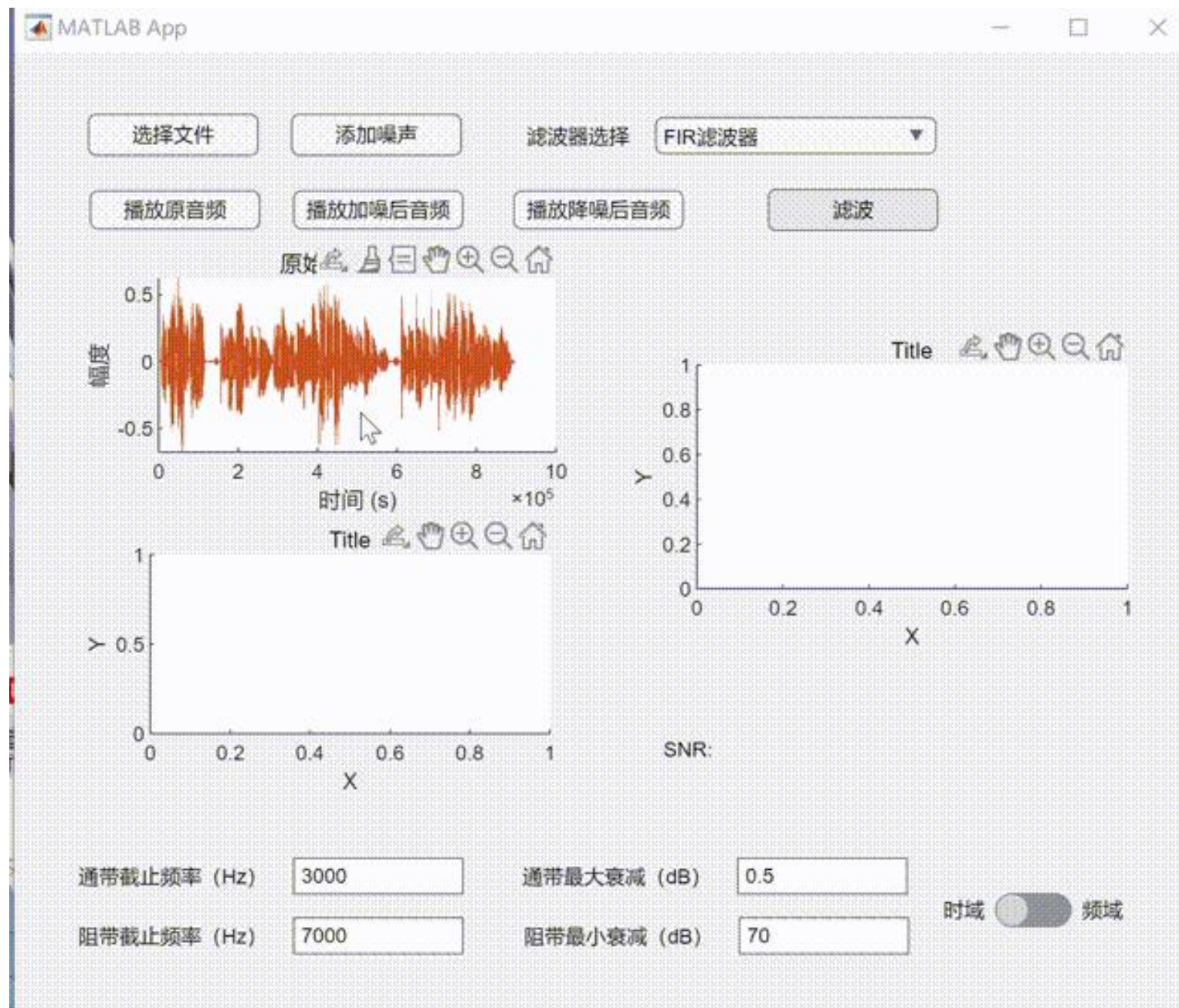
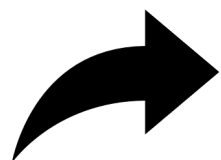
自带的 IIR 滤波器



谱减法 滤波器



卡尔曼 滤波器



THANKS

