RUTGERS UNIVERSITY
School of Engineering – ECE Department
332:447 – Digital Signal Processing Design
S. J. Orfanidis – Fall 2017

Project 1 – Dereverberation
Removing unwanted echoes and reverb from recorded audio

Assigned 09/14/17 – Due 09/21/17

1. Introduction

The purpose of this project is to design and implement an inverse filtering procedure for removing unwanted echoes and reverb from recorded audio. Echoes, reverb, and other audio effects are discussed in Section 8.2 of the I2SP text [1].

In writing your report, please adhere strictly to the report preparation guidelines mentioned in the syllabus. Your report should be written preferably in LaTeX, or Word, and converted to PDF.

2. Design Procedures

a. In recording a guitar piece, an inadvertent echo of the original was added to the recording. Let $x(n), s(n)$ denote the sampled recorded and original signals, related by,

$$x(n) = s(n) + a\,s(n-D) \tag{1}$$

where a, D are unknown and represent an attenuation coefficient and the echo delay D in samples. The recorded and original signals can be loaded into MATLAB from two wave files included in this assignment,

```
[x,fs] = audioread('echo.wav');         % signal with echo
[s,fs] = audioread('original.wav');     % original
```

Both have duration of 10 seconds and were sampled at 16 kHz sampling rate. The original is not needed in this part, but is included so that you can assess your results.

Devise a method to determine estimates of the unknown parameters a, D from $x(n)$, and without using $s(n)$.

Then, apply an inverse filter to $x(n)$ to recover an estimate, say, $y(n)$, of the original.

Listen to all three signals $x(n), y(n), s(n)$ and plot them versus sampled time in seconds.

Save $y(n)$ in a wave file and upload it to Sakai with your report.

Note: Before saving $y(n)$ to an audio file, make sure to normalize it so that it lies in the interval, $-1 \le y(n) \le 1$, otherwise it will be clipped.

b. In another attempt to record the same guitar piece, the recording was carried out in a highly reverberant room and saved into a wave file. The recorded signal can be loaded into MATLAB with the command,

```
[x,fs] = audioread('reverb.wav');         % signal with reverb
```

The original is the same as above, and is to be used only as reference to assess your results.

In order to deconvolve the effect of the room, one would need to measure the impulse response of the room. However, this proved to be difficult and instead the following procedure was used: A short 1 kHz sinusoidal pulse $p(n)$ of duration of 0.25 sec was generated and its reverberating version $r(n)$ was recorded over a period of 1.50 sec. This pulse and its recorded version can be loaded from the included wave files:

```
[p,fs] = audioread('pulse.wav');         % generated pulse p(n)
[r,fs] = audioread('pulse_rec.wav');     % recorded pulse r(n)
```

The signals $p(n), r(n)$ are depicted at the end.

From the known signals $p(n), r(n)$, determine an estimate of the impulse response $h(n)$ of the room and plot it versus time in seconds.

Use $h(n)$ to deconvolve the reverberation from the recording $x(n)$ and recover an estimate, say, $y(n)$, of the original signal $s(n)$.

Listen to all three signals $x(n), y(n), s(n)$ and plot them versus sampled time in seconds.

Save $y(n)$ in a wave file and upload it to Sakai with your report.

c. Having successfully recorded your original piece $s(n)$, now you decide to add a few effects to it, such as (i) FIR comb, (ii) multi-delay, (iii) multi-tap, and (iv) flange. Please see the attached file "effects-sum.pdf" for a summary of the block diagrams and sample processing algorithms of these effects.

   (i) Add an FIR comb effect that has three 250 msec delays, picking an appropriate value for the attenuation coefficient a. Using a circular buffer implementation, calculate and plot the resulting signal $y(n)$, save it into a wave file and upload it to Sakai with your report.

   Repeat the calculation of $y(n)$ using the built-in function **filter** and compare its execution speed with that of the circular-buffer version.

   (ii) Add a multi-delay effect that has delay parameters D_1, D_2 corresponding to 0.250 and 0.125 msec delays, and coefficients, $a_1 = 0.2$, $a_2 = 0.4$, $b_0 = b_1 = b_2 = 1$.

   Using a circular buffer implementation, calculate and plot the resulting signal $y(n)$, save it into a wave file and upload it to Sakai with your report.

   Repeat the calculation of $y(n)$ using the built-in function **filter** and compare its execution speed with that of the circular-buffer version.

   (iii) Repeat part (ii), for a multi-tap effect that now has the reversed values of the delays D_1, D_2, that is, corresponding to 0.125 and 0.250 delays, respectively.

   (iv) Load another guitar piece with the command:

   ```
   [x,fs] = audioread('noflange.wav');      % original
   ```

   Add a flanging effect in which the time-varying delay varies sinusoidally between 0 and 3 msec with a frequency of 2 Hz.

   Using a circular buffer implementation, calculate and plot the resulting signal $y(n)$, save it into a wave file and upload it to Sakai with your report.

Example wave files of what these effects sound like are included in this assignment.

3. References

These references have been placed on Sakai Resources.

[1] S. J. Orfanidis, *Introduction to Signal Processing*, online book, 2010.

[2] Udo Zölzer, ed., *DAFX – Digital Audio Effects*, Wiley, Chichester, England, 2003.

Example Graphs

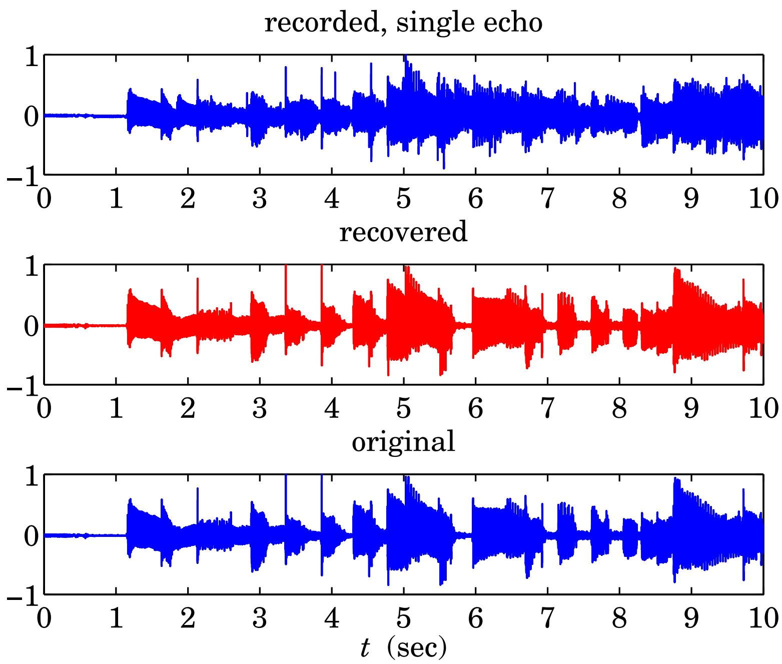


Fig. 1 Echo, restored, and original signals.

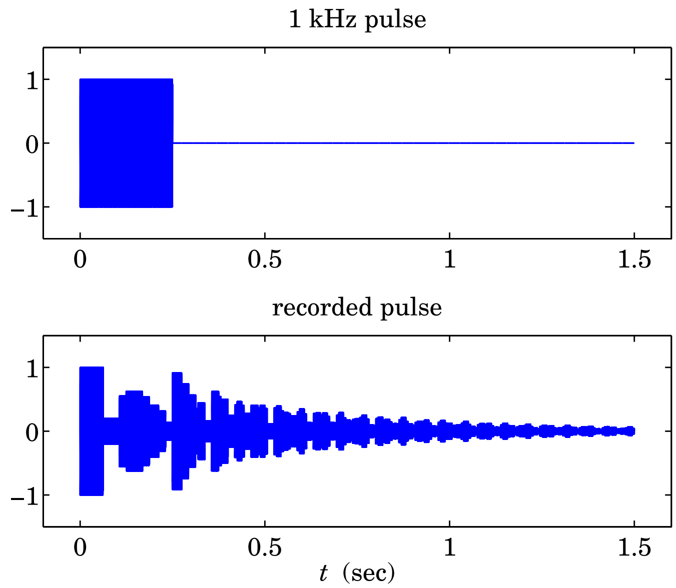


Fig. 2 Emitted and recorded pulse.

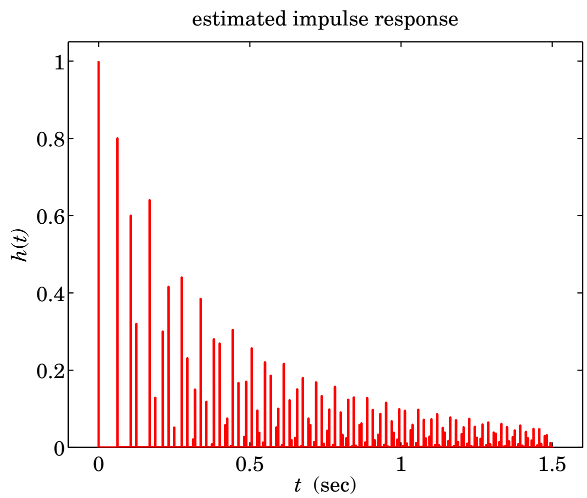


Fig. 3 Estimated room impulse response.

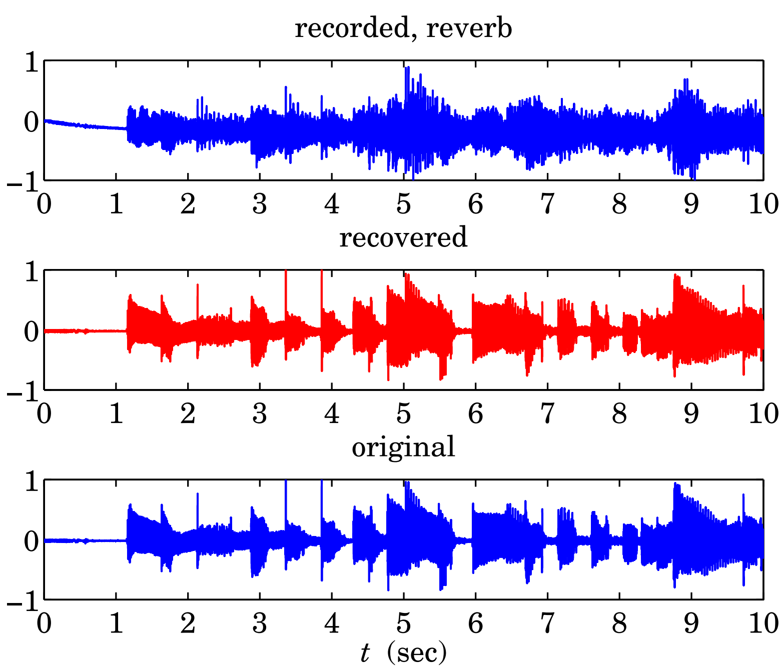
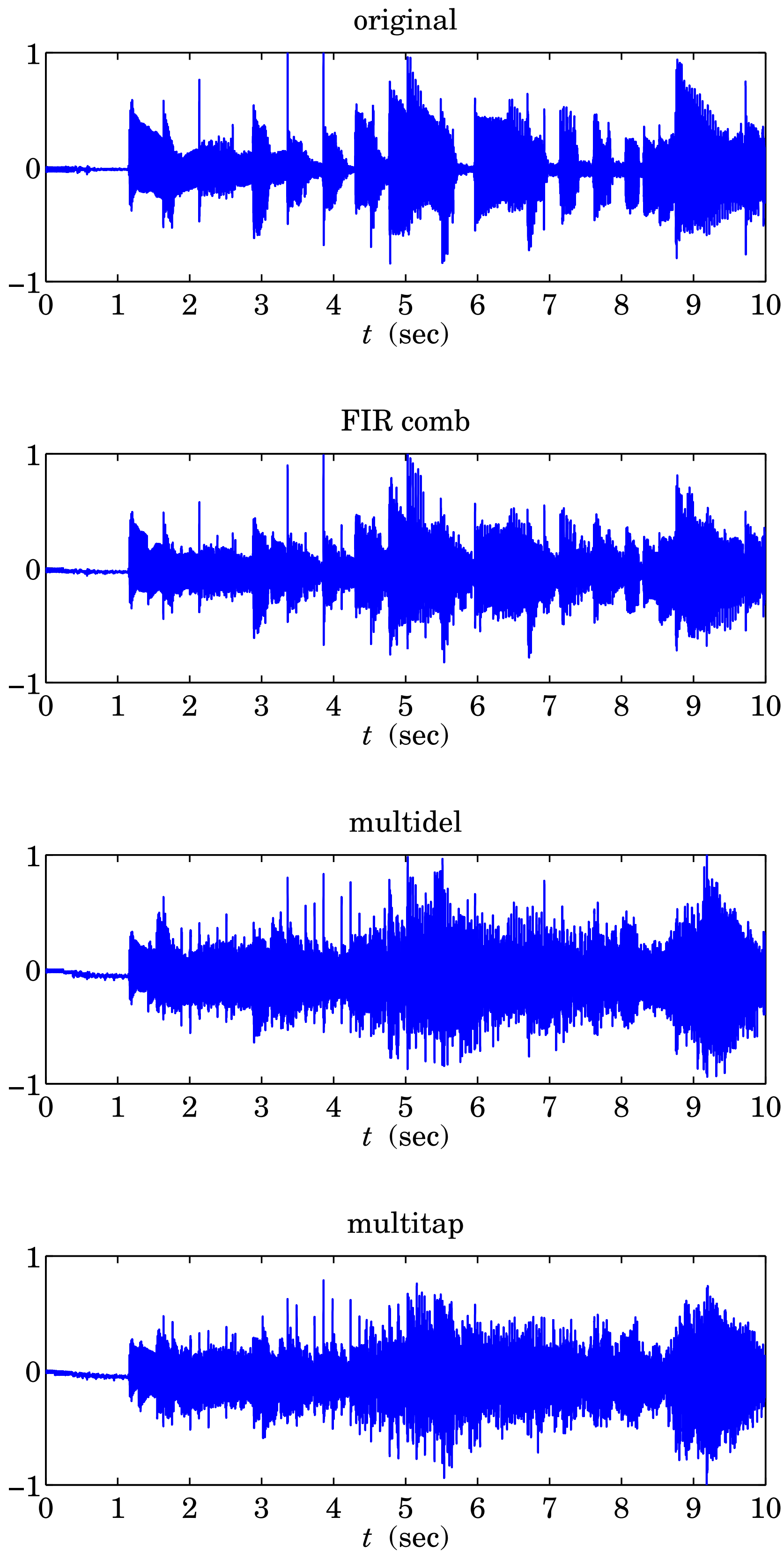
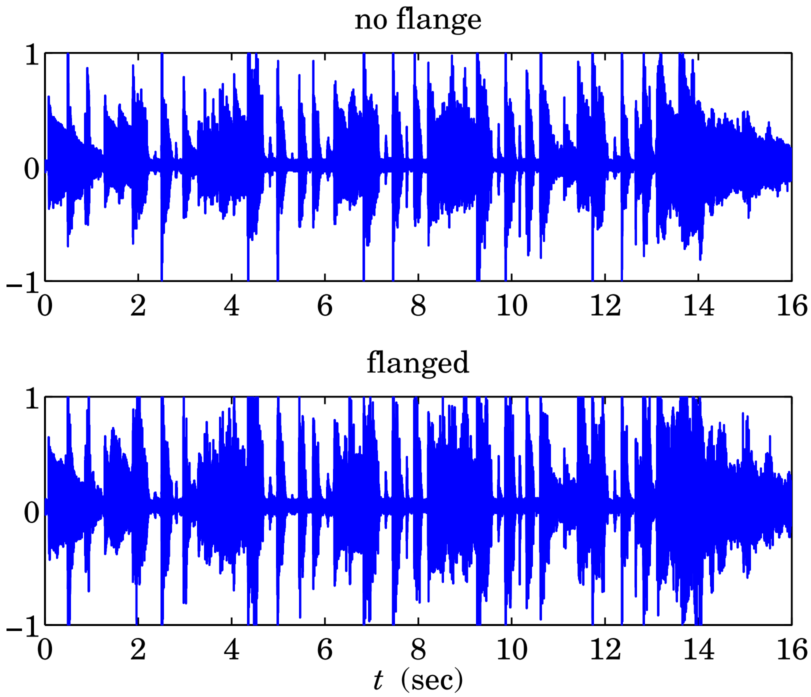


Fig. 4 Reverb, restored, and original signals.


original
FIR comb
multidel
multitap
1
0
−1
0 1 2 3 4 5 6 7 8 9 10
t (sec)


no flange
1
0
−1
0 2 4 6 8 10 12 14 16
flanged
1
0
−1
0 2 4 6 8 10 12 14 16
t (sec)