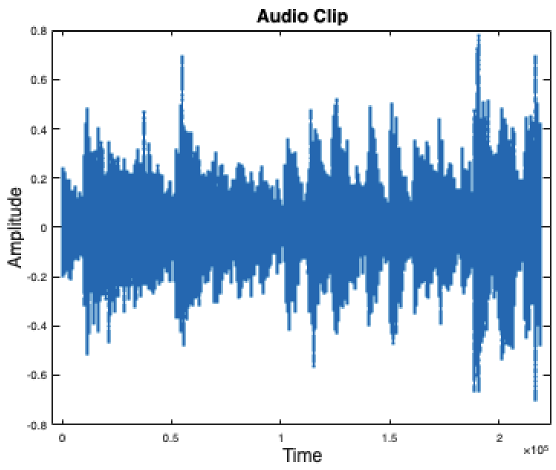
Lab 2

Problem 1: Simulating reverberations

In this problem, we will simulate reverberations using an audio clip.

a) Loading the clip and plotting the signal, we observe the plot below:



```
>> [audio_clip , fs] = audioread('ee110b.m4a');
>> audio_clip = mean(audio_clip, 2);  % stereo to mono
>> sound(audio_clip, fs);
>> stem(audio_clip, ('.');
>> title("Audio Clip", "FontSize", 16);
>> xlabel("Time", "FontSize", 16);
>> ylabel("Amplitude, "FontSize", 16);
```

b) Given that audio reverberations can be written as a sum of delayed and scaled copies of the original signal,

$$y[n] = \sum_{k=0}^{N_{\text{reverb}}-1} a_{qk} x[n - q_k], \tag{1}$$

by definition, we can conclude the impulse response must be

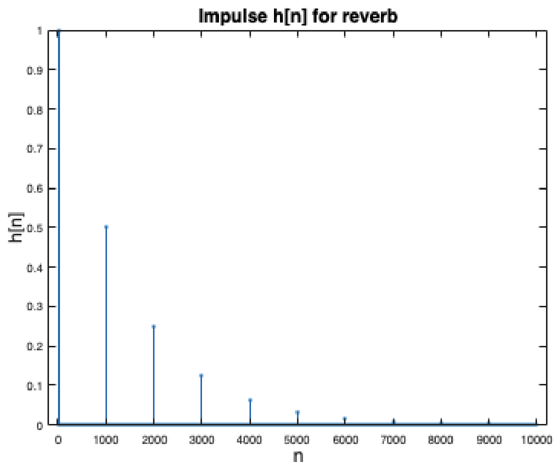
$$h[n] = \sum_{k=0}^{N_{\text{reverb}}-1} a_{qk} \delta[n - q_k] \tag{2}$$

as this is a discrete-time LTI system. This is a series of time-shifted and scaled impulses. As $h[n]$ is a finite sum of scaled impulses, finite values, the condition for stability,

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty, \rightarrow \sum_{k=0}^{N_{\text{reverb}}-1} a_{qk} < \infty, \tag{3}$$

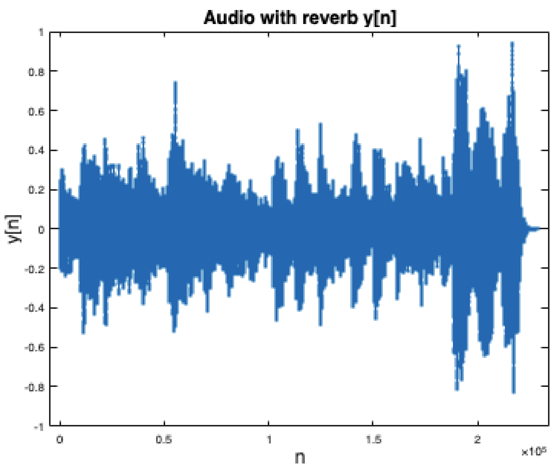
is true. Therefore, we can conclude that the system is stable.

c) The plot of an impulse response with $N_{\text{reverb}} = 10$, $y[n]$ of length $10,000$, $q_k = 1000k$, and $a_{qk} = [\frac{1}{2}]^k$ is shown below:



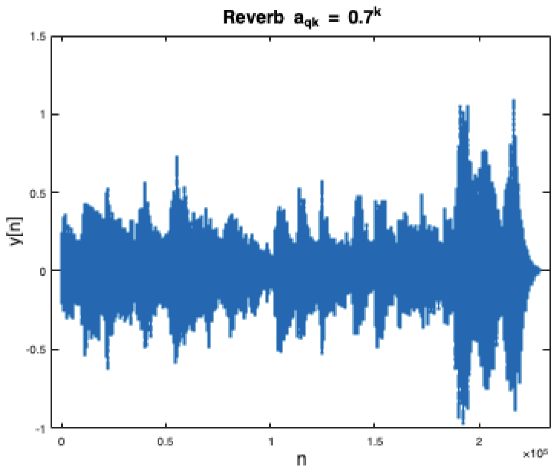
```
>> h_reverb = zeros(10000, 1);

>> for k = 0:9
>> qk = 1000 * k;      % delay pos
>> ak = 1 / (2^k);
>> h_reverb(qk+1) = ak;  % index adjust
>> end

>> stem(h_reverb, '.');
>> title('Impulse h[n] for reverb', 'FontSize', 16);
>> xlabel('n', 'FontSize', 16);
>> ylabel('h[n]', 'FontSize', 16);
```

d) The output for this system, $y[n] = x[n] * h[n]$, is shown below:



```
>> reverb_audio = conv(audio_clip, h_reverb);
>> sound(reverb_audio, fs);
>> stem(reverb_audio, '.');
>> title('Audio with reverb y[n]', 'FontSize', 16);
>> xlabel('n', 'FontSize', 16);
>> ylabel('y[n]', 'FontSize', 16);
```

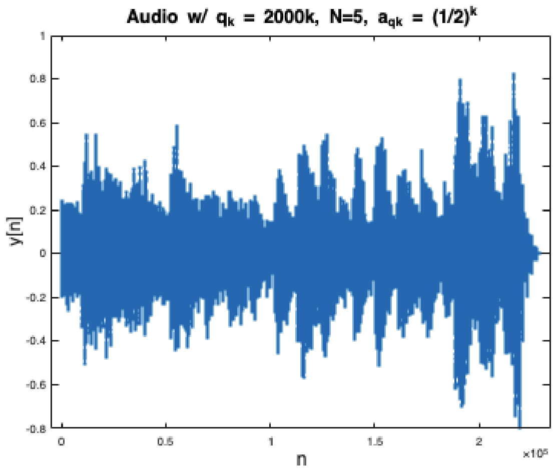
e) The output for the system with $a_{qk} = 0.7^k$ is shown below:



```
>> h_reverb = zeros(10000, 1);

>> for k = 0:9
>> qk = 1000 * k;
>> ak = 0.7^k;
>> h_reverb(qk+1) = ak;
>> end

>> reverb_audio = conv(audio_clip, h_reverb);
>> sound(reverb_audio, fs);
>> stem(reverb_audio, '.');
>> title('Reverb a_qk = 0.7^k', 'FontSize', 16);
>> xlabel('n', 'FontSize', 16);
>> ylabel('y[n]', 'FontSize', 16);
```

When compared to the original audio with reverb in part **d)**, we find that the signal has higher peaks in amplitude. It almost looks like the reverberations closely follow the original signal, even though they should, in practice, be further in distance. When listening to this sample, it sounds much more distorted as well, and is much more unpleasant to listen to — it more closely resembles a "chorus" effect as opposed to a reverb one, though it is important to consider that they are very similar, only different in the parameter that was changed for this example.

f) Repeating the experiment with $q_k = 2000k, N_{\text{reverb}} = 5$ and $a_{qk} = [\frac{1}{2}]^k$, we achieve the plot below:



```
>> h_reverb = zeros(10000, 1);

>> for k = 0:4
>> qk = 2000 * k;
>> ak = 1 / (2^k);
>> h_reverb(qk+1) = ak;
>> end

>> reverb_audio = conv(audio_clip, h_reverb);
>> sound(reverb_audio, fs);
>> stem(reverb_audio, '.');
>> title('Audio w/ q_k = 2000k, N=5, a_qk = (1/2)^k',
'FontSize', 16);
```

```
>> xlabel('n', 'FontSize', 16);
>> ylabel('y[n]', 'FontSize', 16);
```

This audio clip looks and sounds very similar to the original we had produced. The reason for this is that while q_k has increased by a factor of 2, N_{reverb} has decreased by a factor of 2. In practice, this means that they essentially cancel each other out, and we are left with the same signal that we started with. That is, the signal sounds more like a realistic replication of what reverberations in a larger room would sound compared to the previous example.

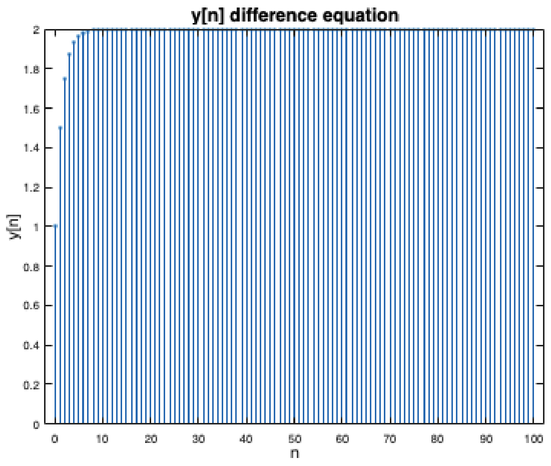
Problem 2: Difference equations

In this problem, we consider a system described by the following difference equation:

$$y[n] = \frac{1}{2}y[n-1] + x[n]. \tag{4}$$

We let $x[n] = 1$, and assume the initial condition $y[0] = 0$.

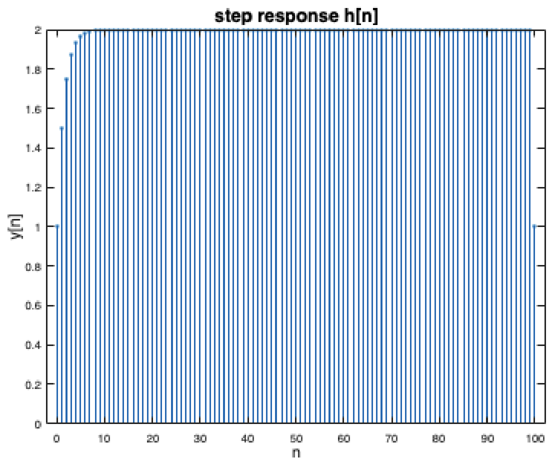
a) Evaluating $y[n]$ for $n = 0, 1, 2, ..., 100$, we obtain the following plot of output $y[n]$ for all values of n:



```
>> n = 0:100; % Time axis
>> x = ones(1, 101); % Input signal
>> y = zeros(1, 101); % Output
>>
>> for idx = 1:length(n)
>>     if idx == 1
>>         y(idx) = x(idx);
>>     else
>>         y(idx) = 0.5 * y(idx - 1) + x(idx);
>>     end
>> end
```

```
>>
>> stem(n, y, '.');
>> title('Output y[n] from difference equation', 'FontSize', 16);
>> xlabel('n', 'FontSize', 14);
>> ylabel('y[n]', 'FontSize', 14);
```

b) Considering an LTI system with the impulse response, $h[n] = \frac{1}{2^n}u[n]$, we will evaluate and plot the step response of the system below:



```
>> n = 0:100; % time axis
>> h = (1/2).^n; % impulse response
>> x = ones(1, 100); % step input

>> y_step = conv(x, h);
>> y_step = y_step(1:101); % length match

>> stem(n, y_step, '.');
>> title('step response h[n]', 'FontSize', 16);
>> xlabel('n', 'FontSize', 14);
>> ylabel('y[n]', 'FontSize', 14);
```

c) The systems described in **a)** and **b)** are the same. This is primarily indicated by their resulting output being the same. We can prove this by setting the input $x[n] = \delta[n]$ and analyzing the output $y[n]$. Applying this, we see

$$\begin{aligned}
y[0] &= \delta[0] = 1 \\
y[1] &= \frac{1}{2}y[0] + \delta[1] = \frac{1}{2} \\
y[2] &= \frac{1}{2}y[1] + \delta[2] = \frac{1}{4} \\
y[3] &= \frac{1}{2}y[2] + \delta[3] = \frac{1}{8} \\
&\vdots \\
y[n] &= \frac{1}{2^n}u[n].
\end{aligned}$$

This indicates that the systems are equal, as we can obtain the impulse response from the difference equation.