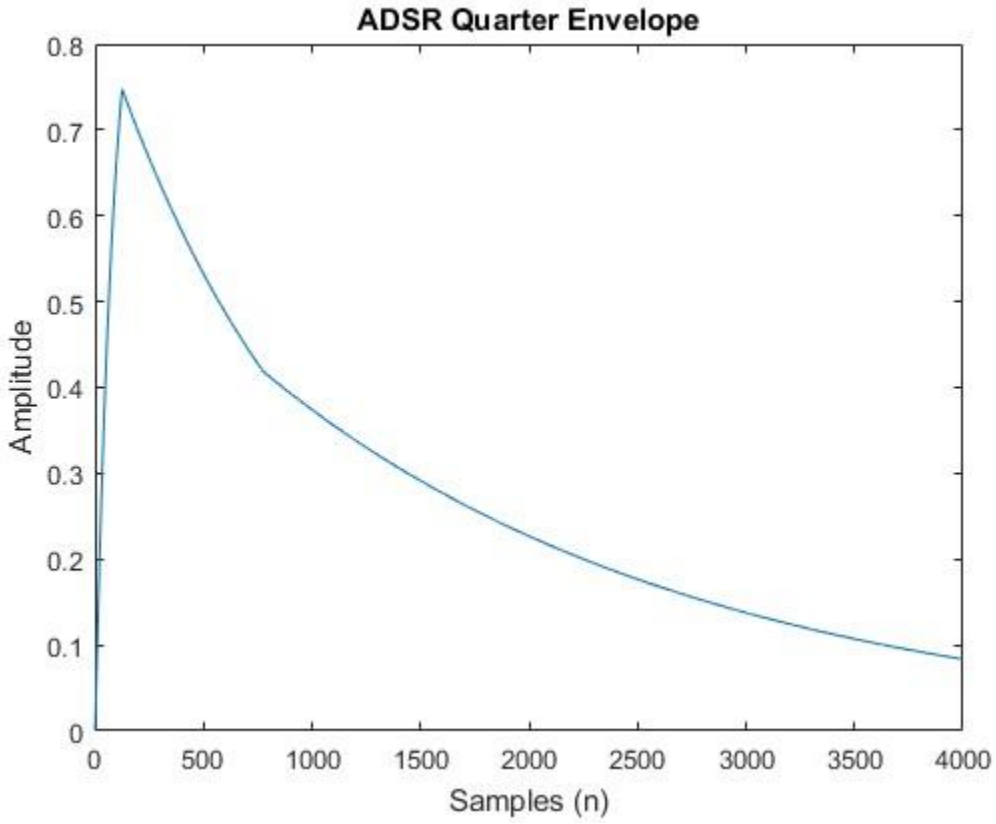
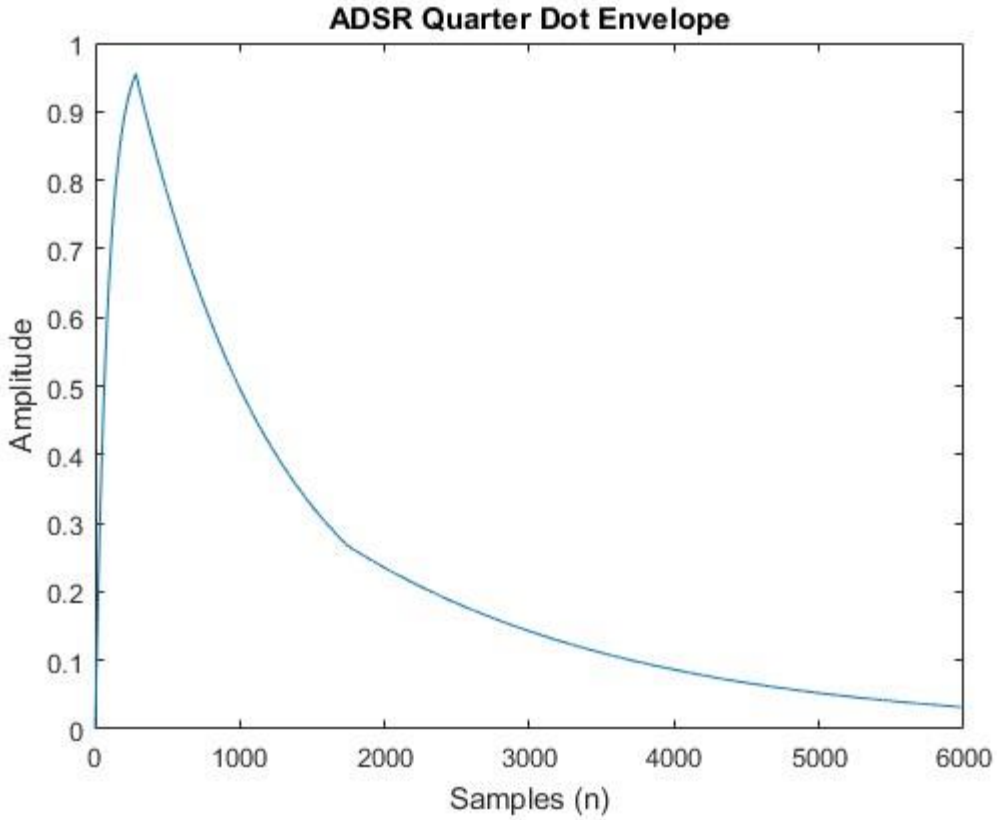
Project 3 ADSR Envelope music synthesis

By Christopher W. Fingers and Alexander Bolton
Professor Moussavi
EE 427 Speech Signal Processing

```
%  The music used was the first 10 seconds of the audiotrack "His Theme" %  from the video
game Undertale.

% Define our ADSRs' target, gain and duration(whole) values.
target = [ 1, 0.06, 0.00]; whole = [125, 650 , 225];
gain_whole = [.03*exp(-1), 0.001, 0.0005];

%  Dependent on quarter and dot quarter notes from song, the duration of
%  the signal will decrease from the length of a whole note by the length %  of the note.
quarter = whole/4; dot_quart = quarter*(3/2);

% Define the sampling frequency and the frequencies used in the song. fs = 16000;
frequency = [369.994, 554.365, 493.883, 466.164, 622.254];
%  Adsr is created for both quarter and dot quarter note with a new %  variable, a,
used to represent the type of note that is being %  represented.  adsr_quarter =
adsr_gen(target,gain_whole,quarter,4);                    adsr_quarter_dot          =
adsr_gen(target,gain_whole,dot_quart,8/3);
 % using singen the note values are calculated out with the correct
 % duration for each signal. fsharp = singen(frequency(1),
fs, 0.25); c = singen(frequency(2),fs,0.25); cdot =
singen(frequency(2),fs,3/8); b =
singen(frequency(3),fs,0.25); bdot =
singen(frequency(3),fs,3/8); a =
singen(frequency(4),fs,3/8); dsharp =
singen(frequency(5),fs,0.25);



%  The envelope and signal are multiplied. fsound =
adsr_quarter.*fsharp; csound = adsr_quarter.*c; cdotsound =
adsr_quarter_dot.*cdot; bsound = adsr_quarter.*b; bdotsound =
adsr_quarter_dot.*bdot; asound = adsr_quarter_dot.*a; dsound =
adsr_quarter.*dsharp;

% Define each verse Appropriately.
verse1 = [fsound; csound; bsound; fsound; asound; asound; bsound];
```

```
verse2 = [bsound; fsound; bsound; fsound; asound; asound; bsound];
verse3 = [bsound; fsound; bsound; dsound;
cdotsound;bdotsound;csound];
% Standard plot for the quarter note adsr and the quarter dot note
adsr % envelope. figure(1) plot(adsr_quarter); title('ADSR Quarter
Envelope'); xlabel('Samples (n)'); ylabel('Amplitude'); figure(2)
plot(adsr_quarter_dot); title('ADSR Quarter Dot Envelope');
xlabel('Samples (n)'); ylabel('Amplitude');


% Play out the audio in 2 second blocks.
 soundsc(verse1,fs)
pause(2)
soundsc(verse2,fs)
pause(2)
soundsc(verse1,fs)
pause(2)
soundsc(verse3,fs)
```





Published with MATLAB® R2016a

```
function a = adsr_gen(target,gain,duration,a)
  %
Input
% target - vector of attack, sustain, release target values
% gain - vector of attack, sustain, release gain values
% duration - vector of attack, sustain, release durations in ms
% Output
% a - vector of adsr envelope values
  fs =
16000; fsd
= fs/a;
a = zeros(fsd,1); % assume 1 second duration ADSR envelope duration
= round(duration./1000.*fsd); % envelope duration in samp
% Attack phase
  start = 2; stop
= duration(1);

for n = [start:stop]
a(n) = target(1)*gain(1) + (1.0 - gain(1))*a(n-1);
end
% Sustain phase    start =
stop + 1; stop = start +
```

```
duration(2); for n =
[start:stop]
a(n) = target(2)*gain(2) + (1.0 - gain(2))*a(n-1);
end

% Release phase
start = stop + 1;
stop = fsd; for n =
[start:stop]
a(n) = target(3)*gain(3) + (1.0 - gain(3))*a(n-1);
end;
```

Singen:

```
function x = singen( f,fs,duration )
% Generating a sine wave.
n = [0:(fs*duration)-1]';
x = sin(2*pi*n*f/fs);

end
```