# Overdrive, Distortion and Fuzz

## Distortion:

- plays an important part in electric guitar music, especially rock music and its variants.
- can be applied as an effect to other instruments including vocals.

## Three broad classes of distortion:

**Overdrive** — Audio at a low input level is driven by higher input levels in a non-linear curve characteristic

**Distortion** — a wider tonal area than overdrive operating at a higher non-linear region of a curve

**Fuzz** — complete non-linear behaviour, harder/harsher than distortion

# Overdrive

## Achieving Overdrive:

- **Symmetrical soft clipping** of input values is performed.
- A simple three layer *non-linear soft saturation* scheme may be:

$$f(x) = \begin{cases} 2x & \textbf{for} \quad 0 \leq x < 1/3 \\ \frac{3-(2-3x)^2}{3} & \textbf{for} \quad 1/3 \leq x < 2/3 \\ 1 & \textbf{for} \quad 2/3 \leq x \leq 1 \end{cases}$$

  - In the lower third the output is liner — multiplied by 2.
  - In the middle third there is a non-linear (quadratic) output
    response
  - Above 2/3 the output is set to 1.

# MATLAB Overdrive Example

p

## Symmetrical soft clipping, symclip.m:

```
function y=symclip(x)

N=length(x);
y=zeros(1,N); % Preallocate y
th=1/3; % threshold for symmetrical soft clipping
        % by Schetzen Formula
for i=1:1:N,
   if abs(x(i))< th, y(i)=2*x(i);end;
   if abs(x(i))>=th,
     if x(i)> 0, y(i)=(3-(2-x(i)*3).^2)/3; end;
     if x(i)< 0, y(i)=-(3-(2-abs(x(i))*3).^2)/3; end;
   end;
   if abs(x(i))>2*th,
     if x(i)> 0, y(i)=1;end;
     if x(i)< 0, y(i)=-1;end;
   end;
end;
```

# MATLAB Overdrive Example (Cont.)

An **overdriven signal** looks and sounds like this :

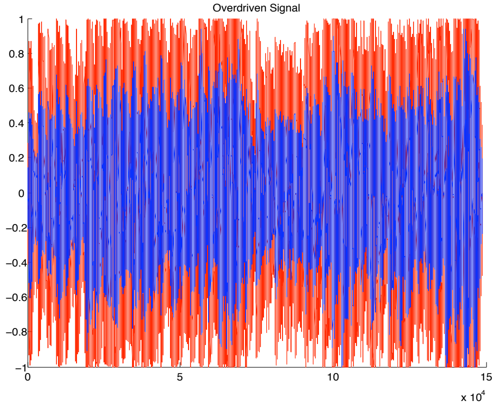
overdrive_eg.m:

```
% read the sample waveform
filename='acoustic.wav';
[x,Fs] = audioread(filename);

% call symmetrical soft clipping
% function
y = symclip(x);

% write output
audiowrite('out_overdrive.wav', y,Fs);

figure(1); hold on;
plot(y,'r');
plot(x,'b');
title('Overdriven Signal');
```




Click image or here to hear: original audio, overdriven audio.

# Distortion/Fuzz

## Distortion/Fuzz implementation:

- Apply non-linear **amplification function**.
- A non-linear function commonly used to simulate **distortion/fuzz** is given by:

$$f(x) = \frac{x}{|x|}(1 - e^{\alpha x^2/|x|})$$

  - This a non-linear exponential function:
  - The gain, $\alpha$, controls **level** of distortion/fuzz.
  - Common to **mix** part of the **distorted signal** with **original signal** for **output**.

# MATLAB Fuzz Example

fuzzexp.m:

```
function y=fuzzexp(x, gain, mix)
% y=fuzzexp(x, gain, mix)
% Distortion based on an exponential function
% x    - input
% gain - amount of distortion, >0->
% mix  - mix of original and distorted sound, 1=only distorted

q=x*gain/max(abs(x));
z=sign(-q).*(1-exp(sign(-q).*q));
y=mix*z*max(abs(x))/max(abs(z))+(1-mix)*x;
y=y*max(abs(x))/max(abs(y));
```

**Note**: **function allows to `mix` input and fuzz signals at output**

# MATLAB Fuzz Example (Cont.)

An **fuzzed up signal** looks and sounds like this:

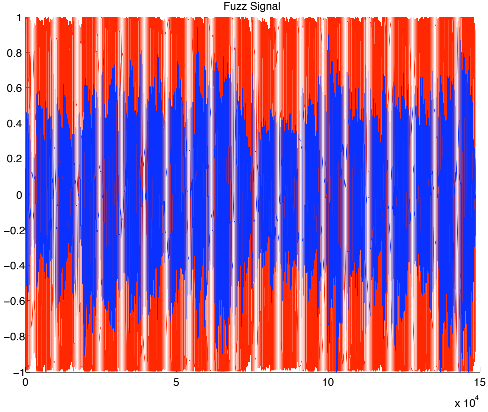
fuzz_eg.m:

```
filename='acoustic.wav';

% read the sample waveform
[x,Fs] = audioread(filename);

% Call fuzzexp
gain = 11; % Spinal Tap it
mix = 1; % Hear only fuzz
y = fuzzexp(x,gain,mix);

% write output
audiowrite('out_fuzz.wav', y,Fs);
```




Click image or here to hear: original audio, Fuzz audio.

# Exciter and Enhancers

### Exciter:

A signal processor that emphasises or de-emphasises certain frequencies in order to change a signal's timbre. It can bring extra brightness without necessarily adding in equalisation.

- Frequently used Fourier domain.

### Enhancers:

Combine equalisation with non-linear processing.

- introduce a small amount ('just noticeable') of distortion.

# Exciters

## Achieving Excitation:

- Basic signal processing is achieved by subtle amounts of high frequency distortion and possible phase shifting.
- Performed using the **Short-Time** (Windowed) **Fourier Transform** (STFM) (see **Phase Vocoder**)
- **Compression** often **employed** to non-linear frequency processed element before mixing with the original signal
- Effect can bring *more presence* and clarity to a single instrument in a mix
- Can add natural brightness to a stereo signal
- Can aid intelligibility to speech and vocals.
- Best applied to signals which lack high frequency content unless some odd special effects are required.

# Enhancers

## Achieving Enhancment:

- Enhancers comprise of a filter network and harmonic generator.
- At least a three band filter is used and an equaliser will boost or cut the frequencies in these bands — independently therefore **non-linearly**.
- Input signal usually mixed with enhanced signal to form output.
- Used in place of equalisers on some mixing consoles.
- Stereo enhancement for radio broadcast and sound reinforcement are also common applications.