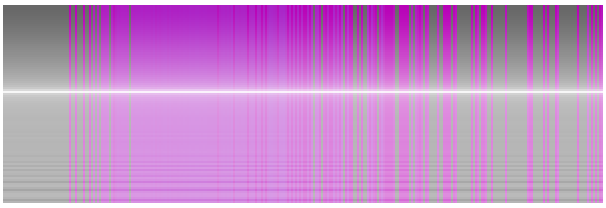
Diffrogram examples with code
Version 1.0 (diffrogram v2.2, Matlab 2014b)

1. Identical sine waves 1000 Hz as input and output

```
% Sine wave 1000 Hz
% 30s, -10 dBFS (rms), 2 channels, 44100 Hz, 32 bit
Fs = 44100; % [Hz]
L = 30; % [s]
G = -10; % [dB]
F = 1000; % [Hz]
Ls = round(L*Fs);
t = (0:Ls-1)';
ref = 10^(G/20) * sin(t*2*pi/(Fs/F));
ref = [ref,ref];
audiowrite('ref_sine1k_44.wav', ref, Fs, 'BitsPerSample',32)

diffrogram('ref_sine1k_44.wav', 'ref_sine1k_44.wav', 100, 'Mono NoWarp');
```



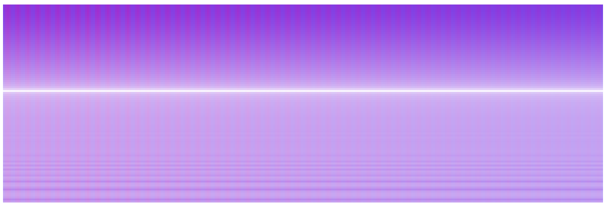
File: `ref_sine1k_44.wav(44)_ref_sine1k_44.wav(44)_mono_100-Inf-150.0000-143.6353.png`

The input and output signals are identical, no warping was applied, the diffrogram shows errors of Difference level computation without warping, Grey bars correspond to -Inf [dB] Df values.

2. Constant deviation of sample rate

```
% Sin wave 1000 Hz
% 30s, -10 dBFS (rms), 2 channels, 44104 Hz, 32 bit
Fs = 44104; % [Hz] +4 Hz increase sample rate
L = 30; % [s]
G = -10; % [dB]
F = 1000; % [Hz]
Ls = round(L*Fs);
t = (0:Ls-1)';
ref = 10^(G/20) * sin(t*2*pi/(Fs/F));
ref = [ref,ref];
audiowrite('ref_sine1k_44+.wav', ref, Fs, 'BitsPerSample',32)

diffrogram('ref_sine1k_44.wav', 'ref_sine1k_44+.wav', 100, 'Mono NoLowpass');
```



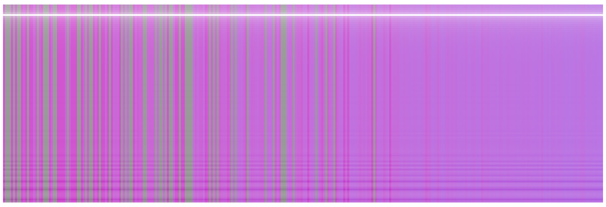
File: `ref_sine1k_44+.wav(44)_ref_sine1k_44.wav(44)_mono_100-141.2851-134.6323-130.3708.png`

Warping is applied and it compensated deviation of sample rate of output signal; the diffrogram shows errors of Difference level computation with warping.

3. Identical sine waves 15 000 Hz as input and output

```
% Sin wave 15000 Hz
% 30s, -10 dBFS (rms), 2 channels, 44100 Hz, 32 bit
Fs = 44100; % [Hz]
L = 30; % [s]
G = -10; % [dB]
F = 15000; % [Hz]
Ls = round(L*Fs);
t = (0:Ls-1)';
ref = 10^(G/20) * sin(t*2*pi/(Fs/F));
ref = [ref,ref];
audiowrite('ref_sine15k_44.wav', ref, Fs, 'BitsPerSample',32)

diffrogram('ref_sine15k_44.wav', 'ref_sine15k_44.wav', 100, 'Mono WarpMargin:1');
```



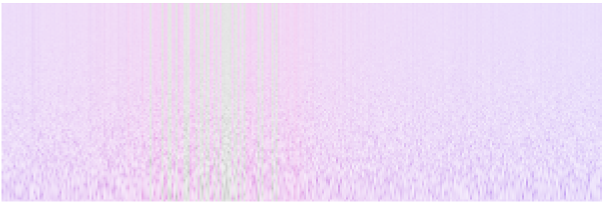
File: `ref_sine15k_44.wav(44)_ref_sine15k_44.wav(44)_mono_100-Inf-144.7747-137.4507.png`

The input and output files are identical but before calculation of Df values the output signal was up-sampled (x4) and time warped; the diffrogram shows errors of Difference level computation with up-sampling and warping.

4. Identical white noise signals as input and output

```
% White noise
% 30s, -10 dBFS (rms), 2 channels, 44100 Hz, 32 bit
Fs = 44100; % [Hz]
L = 30; % [s]
PdB = -10; % [dB]
Ls = round(L*Fs);
ref = randn(Ls,1);
Po = sqrt(mean(ref.^2));
Pd = 10^(PdB/20)/sqrt(2);
Kp = Pd/Po;
ref = ref .* Kp;
ref = [ref,ref];
audiowrite('ref_wn.wav', ref, Fs, 'BitsPerSample',32)

diffrogram('ref_wn.wav', 'ref_wn.wav', 100, 'Mono');
```



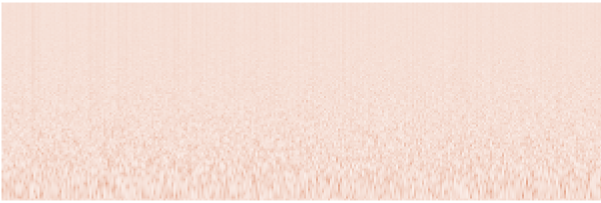
File: `ref_wn.wav(44)_ref_wn.wav(44)_mono_100-Inf-140.3291-132.0951.png`

The same example as in #3 but with wite noise insted of sine wave.

5. White noise with all-pass filter

```
% Allpass filtered white noise
% 30s, -10 dBFS (rms), 2 channels, 44100 Hz, 32 bit
PdB = -10; % [dB]
[out,Fs] = audioread('ref_wn.wav');
a = [1 1/5 1/4 1/3];
b = [1/3 1/4 1/5 1];
% fvtool(b,a)
out = filter(b,a,out);
Pd = 10^(PdB/20)/sqrt(2);
Po1 = sqrt(mean(out(:,1).^2));
Po2 = sqrt(mean(out(:,2).^2));
Kp1 = Pd/Po1;
Kp2 = Pd/Po2;
out(:,1) = out(:,1) .* Kp1;
out(:,2) = out(:,2) .* Kp2;
audiowrite('out_wn_af.wav', out, Fs, 'BitsPerSample',32)

diffrogram('ref_wn.wav', 'out_wn_af.wav', 100, 'Mono');
```



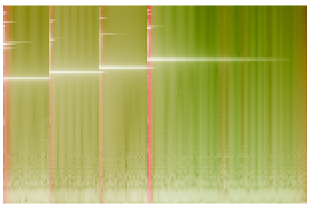
File: `out_wn_af.wav(44)_ref_wn.wav(44)_mono_100-8.5881-8.1375-7.8126.png`

White noise from example #4 was processed by all-pass filter and compared with the unprocessed one. Original waveform degraded substantially.

6. Glockenspiel sample with all-pass filter

```
% Allpass filtered Glockenspiel sample (SE)
% ~15s, -23.7/-24.3 dBFS (rms), 2 channels, 44100 Hz
% 16bit(ref), 32bit(out)
[out,Fs] = audioread('GLK.flac');
Pd1 = sqrt(mean(out(:,1).^2));
Pd2 = sqrt(mean(out(:,2).^2));
a = [1 1/5 1/4 1/3];
b = [1/3 1/4 1/5 1];
out = filter(b,a,out);
Po1 = sqrt(mean(out(:,1).^2));
Po2 = sqrt(mean(out(:,2).^2));
Kp1 = Pd1/Po1;
Kp2 = Pd1/Po2;
out(:,1) = out(:,1) .* Kp1;
out(:,2) = out(:,2) .* Kp2;
audiowrite('out_GLK_af.wav', out, Fs, 'BitsPerSample',32)

diffrogram('GLK.flac', 'out_GLK_af.wav', 100, 'Mono');
```

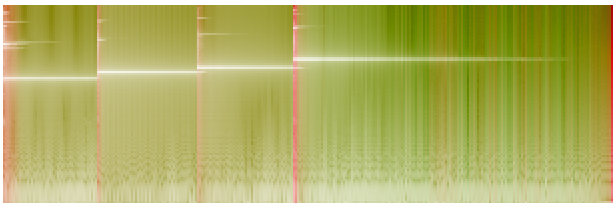


File: `out_GLK_af.wav(44)_GLK.flac(44)_mono_100-40.3269-32.3450-0.4874.png`

The same example as in #5 but with Glockenspiel sample instead of white noise. Some parts of the sample degraded higher than others.

The warped signal `out_GLK_af.wav(44)_warp_mono_30000.wav` returned by `diffrogram` function in this example can be used to produce diffrogram with different time window `Wdiff` in order to have more detailed view of the degradation. No need to perform computationally intensive warping once again:

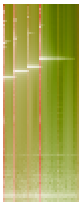
```
diffrogram('GLK.flac', 'out_GLK_af.wav(44)_warp_mono_30000.wav', 50, 'Mono NoWarp');
```



File:
out_GLK_af.wav(44)_warp_mono_30000.wav(44)_GLK.flac(44)_mono_50-42.3870-32.0029-0.3980.png

The `diffrogram` function also returns vector of Df values which can be used for further analysis:

```
df = diffrogram('GLK.flac', 'out_GLK_af.wav(44)_warp_mono_30000.wav', 400, 'Mono NoWarp');
```



File:
out_GLK_af.wav(44)_warp_mono_30000.wav(44)_GLK.flac(44)_mono_400-39.2668-33.4869-6.7296.png

```
df =

  -10.7272
  -29.9253
  -31.4534
  -31.6421
  -31.8515
  -14.4573
  -26.9624
  -30.9259
  -31.0878
  -31.2539
  -31.4248
  -31.5924
  -23.2240
  -30.6522
  -31.5928
  -31.9766
  -32.2699
  -32.5196
   -6.7296
  -34.4541
  -36.4194
  -37.1242
  -37.6369
  -37.7988
  -38.6454
  -39.0245
  -38.7044
  -34.4705
  -37.7599
  -38.4597
  -36.8821
  -39.2668
  -36.8222
  -37.0665
  -37.0849
  -36.8206
  -34.8118
  -29.4956
```

More details can be found in the article “Diffrogram: visualization of signal differences in audio research” -
http://soundexpert.org/news/-/blogs/visualization-of-distortion