In [77]:

```
from scipy.io.wavfile import read
import matplotlib.pyplot as plt
import numpy as np
import math
import soundfile as sf
from scipy import signal
import librosa
```

In [83]:

```
noise1,sr1=sf.read(r"TIMIT_full/TIMIT_full/train/dr1/fcjf0/sa1.wav")
noise = librosa.resample(noise1,sr1,16000)
```

In [79]:

```
len(noise)
```

Out[79]:

```
46797
```

In [80]:

```
def snr_calculate(speech_data,noise_data):
    speech_energy=np.sum(np.array(speech_data, dtype='int64')**2)
    noise_energy=np.sum(np.array(noise_data, dtype='int64')**2)
    ratio=speech_energy/noise_energy
    sound_level=10*math.log(ratio,10)
    return sound_level
# Function that creates noisy speech signal by combining noise and clean spee
def generate_signal(speech_data,noise_data,dsnr,outputfilename):
#     print(len(speech_data),len(noise_data))
    if(len(speech_data)>=len(noise_data)):
        temp_noise_data=noise_data.copy()
        diff=len(speech_data)-len(noise_data)
#         print("Initial Diff is :",diff)
        while diff!=0:
            if(diff>len(noise_data)):
                temp_noise_data=np.concatenate((temp_noise_data,noise_data))
#                 temp_noise_data.extend(noise_data)
                diff-=len(noise_data)
            else:
                temp_noise_data=np.concatenate((temp_noise_data,noise_data[0:
#                 temp_noise_data.extend(noise_data[0:diff])

                diff-=len(noise_data[0:diff])
#             print("diff is: ",diff)
    else:
        temp_noise_data=noise_data[0:len(speech_data)]
    speech_energy=sum(np.array(speech_data)**2)
    noise_energy=np.sum(np.array(temp_noise_data)**2)
    b=np.sqrt((speech_energy/noise_energy)*(10**(-dsnr/10)))
#     print(temp_noise_data)
#     print(b)
    temp_noise_data=np.asarray(temp_noise_data)
    updated_noise=np.array(b*temp_noise_data)
#     print("Noise : ",len(updated_noise))
#     print("Speech : ",len(speech_data))
    updated_noisy_signal=updated_noise+speech_data
#     print("Speech : ",len(updated_noisy_signal))
    sf.write(file=outputfilename,data=updated_noisy_signal,samplerate=16000)
    ss,sr = librosa.load(outputfilename,sr=None)
    S = librosa.stft(ss,n_fft=512,hop_length=160,win_length=320)
    return
```

In [81]:

```
import os
test_wav_files2=[]
for root, dirs, files in os.walk("TIMIT_full\\TIMIT_full\\test\\"):
    for file in files:
        if file.endswith(".wav"):
             test_wav_files2.append(os.path.join(root, file))
```

In [82]:

```
for filename in test_wav_files2:
    destination_file_name="PREPARED_DATASET\\"+"__".join(str(filename).split(
    speech_signal,sr=sf.read(filename)
    speech_signal = librosa.resample(speech_signal,sr,16000)
    len_speech_signal=len(speech_signal)

    for index,each_snr in enumerate([-5, 0, 10,25]):
#         print(index)
        generate_signal(speech_signal,noise,each_snr,destination_file_name+"_
```

In [89]:

```
print("There are total %s files in PREPARED DATASET FOLDER" % len(os.listdir(
```

```
There are total 6720 files in PREPARED DATASET FOLDER
```

In [91]:

```
os.listdir("PREPARED_DATASET")[:20]
```

Out[91]:

```
['dr1__faks0__sa1_0.wav',
 'dr1__faks0__sa1_1.wav',
 'dr1__faks0__sa1_2.wav',
 'dr1__faks0__sa1_3.wav',
 'dr1__faks0__sa2_0.wav',
 'dr1__faks0__sa2_1.wav',
 'dr1__faks0__sa2_2.wav',
 'dr1__faks0__sa2_3.wav',
 'dr1__faks0__si1573_0.wav',
 'dr1__faks0__si1573_1.wav',
 'dr1__faks0__si1573_2.wav',
 'dr1__faks0__si1573_3.wav',
 'dr1__faks0__si2203_0.wav',
 'dr1__faks0__si2203_1.wav',
 'dr1__faks0__si2203_2.wav',
 'dr1__faks0__si2203_3.wav',
 'dr1__faks0__si943_0.wav',
 'dr1__faks0__si943_1.wav',
 'dr1__faks0__si943_2.wav',
 'dr1__faks0__si943_3.wav']
```

In [92]:

```
os.listdir("PREPARED_DATASET")[-20:]
```

Out[92]:

```
['dr8__mslb0__sx113_0.wav',
 'dr8__mslb0__sx113_1.wav',
 'dr8__mslb0__sx113_2.wav',
 'dr8__mslb0__sx113_3.wav',
 'dr8__mslb0__sx203_0.wav',
 'dr8__mslb0__sx203_1.wav',
 'dr8__mslb0__sx203_2.wav',
 'dr8__mslb0__sx203_3.wav',
 'dr8__mslb0__sx23_0.wav',
 'dr8__mslb0__sx23_1.wav',
 'dr8__mslb0__sx23_2.wav',
 'dr8__mslb0__sx23_3.wav',
 'dr8__mslb0__sx293_0.wav',
 'dr8__mslb0__sx293_1.wav',
 'dr8__mslb0__sx293_2.wav',
 'dr8__mslb0__sx293_3.wav',
 'dr8__mslb0__sx383_0.wav',
 'dr8__mslb0__sx383_1.wav',
 'dr8__mslb0__sx383_2.wav',
 'dr8__mslb0__sx383_3.wav']
```