In [1]:
```
from scipy import signal
from scipy.io import wavfile
import scipy.io.wavfile
from IPython.display import Audio
from matplotlib import pyplot as plt
import numpy as np
import glob
import librosa
import librosa.display
from librosa import frames_to_time
import IPython.display as ipd
import os
import numpy as np
import pandas as pd
```

In [2]:
```
os.getcwd()
print(os.listdir())
```

```
['.anaconda', '.conda', '.condarc', '.continuum', '.ipynb_checkpoints', '.ipython', '.jupy
ter', '.keras', '.matplotlib', '.ms-ad', '.spyder-py3', '1.png', '10f,10d,1000ss_AI 0.wa
v', '10f,10d,2000ss_AI 0.wav', '10f,1d,1000s_AI 0.wav', '10f,1d,2000ss_AI 0.wav', '10f,5d,
1000s_AI 0.wav', '10f,5d,2000ss_AI 0.wav', '10f,5d,3000spindlepseed_AI 0.wav', '1f,10d,100
0ss_AI 0.wav', '1f,1d,1000s_AI 0.wav', '1f,1d,2000ss_AI 0.wav', '1f,5d,1000ss_AI 0.wav',
'1f,5d,2000ss_AI 0.wav', '3D Objects', '5f,10d,1000ss_AI 0.wav', '5f,10d,2000ss_AI 0.wav',
'5f,10d,3000ss_AI 0.wav', '5f,1d,10000s_AI 0.wav', '5f,1d,2000ss_AI 0.wav', '5f,1d,3000ss_
AI 0.wav', '5f,5d,1000ss_AI 0.wav', '5f,5d,3000spindlespeed_AI 0.wav', 'abaqus_plugins',
'abaqus_v6.13.gpr', 'anaconda3', 'AppData', 'Application Data', 'barfig.png', 'CNN-Copy1.i
pynb', 'CNN.ipynb', 'comp2.csv', 'Contacts', 'converted.mp3', 'Cookies', 'Desktop', 'DFLUX
-subroutine-code', 'Documents', 'Downloads', 'Favorites', 'Figure.png', 'GANs.ipynb', 'Gra
phs.ipynb', 'Jedi', 'keras_detect_tool_wear', 'krish.csv', 'krish1.csv', 'krish2.csv', 'kr
ishna code.ipynb', 'Links', 'LoadMFCC.ipynb', 'Local Settings', 'Log_Spectrogram.ipynb',
'MetaStruct', 'MFCC.ipynb', 'MFCC_Code.ipynb', 'MFCC_mac.ipynb', 'milling1.csv', 'milling
1.xlsx', 'milling2.xlsx', 'model', 'Music', 'My Documents', 'my_model.h5', 'NetHood', 'NTU
SER.DAT', 'ntuser.dat.LOG1', 'ntuser.dat.LOG2', 'NTUSER.DAT{35b34783-ab7d-11eb-9f4f-aa0249
ddd545}.TxR.0.regtrans-ms', 'NTUSER.DAT{35b34783-ab7d-11eb-9f4f-aa0249ddd545}.TxR.1.regtra
ns-ms', 'NTUSER.DAT{35b34783-ab7d-11eb-9f4f-aa0249ddd545}.TxR.2.regtrans-ms', 'NTUSER.DAT
{35b34783-ab7d-11eb-9f4f-aa0249ddd545}.TxR.blf', 'NTUSER.DAT{35b34784-ab7d-11eb-9f4f-aa024
9ddd545}.TM.blf', 'NTUSER.DAT{35b34784-ab7d-11eb-9f4f-aa0249ddd545}.TMContainer00000000000
000000001.regtrans-ms', 'NTUSER.DAT{35b34784-ab7d-11eb-9f4f-aa0249ddd545}.TMContainer00000
000000000000002.regtrans-ms', 'ntuser.ini', 'OneDrive', 'Pictures', 'PrintHood', 'Recent',
'report.log', 'Saved Games', 'ScStore', 'Searches', 'SendTo', 'smart-industry', 'Srav Jump
ECG signal.txt', 'Start Menu', 'TD', 'Templates', 'tensorflow_datasets', 'Transfer-learnin
g.ipynb', 'Untitled Folder', 'Untitled.ipynb', 'Untitled1.ipynb', 'Untitled10.ipynb', 'Unt
itled11.ipynb', 'Untitled12.ipynb', 'Untitled13.ipynb', 'Untitled14.ipynb', 'Untitled2.ipy
nb', 'Untitled3.ipynb', 'Untitled4.ipynb', 'Untitled5.ipynb', 'Untitled6.ipynb', 'Untitled
7.ipynb', 'Untitled8.ipynb', 'Untitled9.ipynb', 'Videos', '~']
```

In [3]:
```
path = os.getcwd()
file_name = "TD"
new_path = os.path.join(path,file_name)
os.chdir(new_path)
```

In [4]:
```
os.listdir()
```

Out[4]:
```
['machinigstatus.csv', 'Test', 'Train', 'Validation']
```

In [5]:
```
df = pd.read_csv(r'C:\Users\HP.NK\TD\machinigstatus.csv')
df
```

Out[5]:

	S.No	fname	lable
0	1	in1_AI 0_AI 0.wav	Tool_in_contat
1	2	in2_AI 0_AI 0.wav	Tool_in_contat
2	3	in3_AI 0_AI 0.wav	Tool_in_contat
3	4	in4_AI 0_AI 0.wav	Tool_in_contat
4	24	ma1_AI 0_AI 0.wav	Machining
...	...	...	...
64	6	in6_AI 0_AI 0.wav	Tool_in_contat
65	7	in7_AI 0_AI 0.wav	Tool_in_contat
66	8	in8_AI 0_AI 0.wav	Tool_in_contat
67	9	in9_AI 0_AI 0.wav	Tool_in_contat
68	10	in10_AI 0_AI 0.wav	Tool_in_contat

69 rows × 3 columns

In [6]:
```
os.listdir()
```

Out[6]:
```
['machinigstatus.csv', 'Test', 'Train', 'Validation']
```

In [7]:
```
path = os.getcwd()
file_name = "Train"
new_path = os.path.join(path,file_name)
os.chdir(new_path)
```

In [8]:
```
os.listdir()
```

Out[8]:
```
['10f,10d,1000ss_AI 0.wav',
 'in10_AI 0_AI 0.wav',
 'in11_AI 0_AI 0.wav',
 'in12_AI 0_AI 0.wav',
 'in13_AI 0_AI 0.wav',
 'in14_AI 0_AI 0.wav',
 'in15_AI 0_AI 0.wav',
 'in16_AI 0_AI 0.wav',
 'in17_AI 0_AI 0.wav',
 'in18_AI 0_AI 0.wav',
 'in19_AI 0_AI 0.wav',
 'in1_AI 0_AI 0.wav',
 'in20_AI 0_AI 0.wav',
 'in21_AI 0_AI 0.wav',
 'in22_AI 0_AI 0.wav',
 'in23_AI 0_AI 0.wav',
 'in2_AI 0_AI 0.wav',
 'in3_AI 0_AI 0.wav',
 'in4_AI 0_AI 0.wav',
 'in5_AI 0_AI 0.wav',
 'in6_AI 0_AI 0.wav',
 'in7_AI 0_AI 0.wav',
 'in8_AI 0_AI 0.wav',
 'in9_AI 0_AI 0.wav',
 'ma10_AI 0_AI 0.wav',
 'ma11_AI 0_AI 0.wav',
 'ma12_AI 0_AI 0.wav',
 'ma13_AI 0_AI 0.wav',
 'ma14_AI 0_AI 0.wav',
 'ma15_AI 0_AI 0.wav',
 'ma16_AI 0_AI 0.wav',
 'ma17_AI 0_AI 0.wav',
 'ma18_AI 0_AI 0.wav',
 'ma19_AI 0_AI 0.wav',
 'ma1_AI 0_AI 0.wav',
 'ma20_AI 0_AI 0.wav',
 'ma21_AI 0_AI 0.wav',
 'ma22_AI 0_AI 0.wav',
 'ma23_AI 0_AI 0.wav',
 'ma2_AI 0_AI 0.wav',
 'ma3_AI 0_AI 0.wav',
 'ma4_AI 0_AI 0.wav',
 'ma5_AI 0_AI 0.wav',
 'ma6_AI 0_AI 0.wav',
 'ma7_AI 0_AI 0.wav',
 'ma8_AI 0_AI 0.wav',
 'ma9_AI 0_AI 0.wav',
 'st10_AI 0_AI 0.wav',
 'st11_AI 0_AI 0.wav',
 'st12_AI 0_AI 0.wav',
 'st13_AI 0_AI 0.wav',
 'st14_AI 0_AI 0.wav',
 'st15_AI 0_AI 0.wav',
 'st16_AI 0_AI 0.wav',
 'st17_AI 0_AI 0.wav',
 'st18_AI 0_AI 0.wav',
 'st19_AI 0_AI 0.wav',
 'st1_AI 0_AI 0.wav',
 'st20_AI 0_AI 0.wav',
 'st21_AI 0_AI 0.wav',
 'st22_AI 0_AI 0.wav',
 'st23_AI 0_AI 0.wav',
 'st2_AI 0_AI 0.wav',
 'st3_AI 0_AI 0.wav',
 'st4_AI 0_AI 0.wav',
 'st5_AI 0_AI 0.wav',
 'st6_AI 0_AI 0.wav',
 'st7_AI 0_AI 0.wav',
```

```
 'st8_AI 0_AI 0.wav',
 'st9_AI 0_AI 0.wav']
```

In [9]:
```
y, sr = librosa.load(os.listdir()[55], duration = 6)
ps = librosa.feature.melspectrogram(y=y, sr=sr)
ps.shape
```

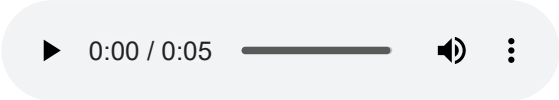
Out[9]:
```
(128, 219)
```

In [10]:
```
import wave
fname = os.path.join(new_path, 'in10_AI 0_AI 0.wav')   # Applause
# Open using wave library
wav = wave.open(fname)
print("Sampling (frame) rate = ", wav.getframerate())
print("Total samples (frames) = ", wav.getnframes())
print("Duration = ", wav.getnframes()/wav.getframerate())

ipd.Audio(fname)
```

```
Sampling (frame) rate =  10000
Total samples (frames) =  50849
Duration =  5.0849
```

Out[10]:



In [11]:
```
filtered_classes = ['Tool_in_contat','Tool_approch','Machining']
train_df_filtered = df[df["lable"].isin(filtered_classes)]

print("Number of training examples: %d"%(train_df_filtered.shape[0]))
print("Number of Classes: %d"%(train_df_filtered.lable.nunique()))

print("\nClasses: ",train_df_filtered.lable.unique())
```

```
Number of training examples: 69
Number of Classes: 3

Classes:  ['Tool_in_contat' 'Machining' 'Tool_approch']
```

In [12]:
```
os.getcwd()
```

Out[12]:
```
'C:\\Users\\HP.NK\\TD\\Train'
```

In [13]:
```
os.path.normpath(os.getcwd() + os.sep + os.pardir)
```

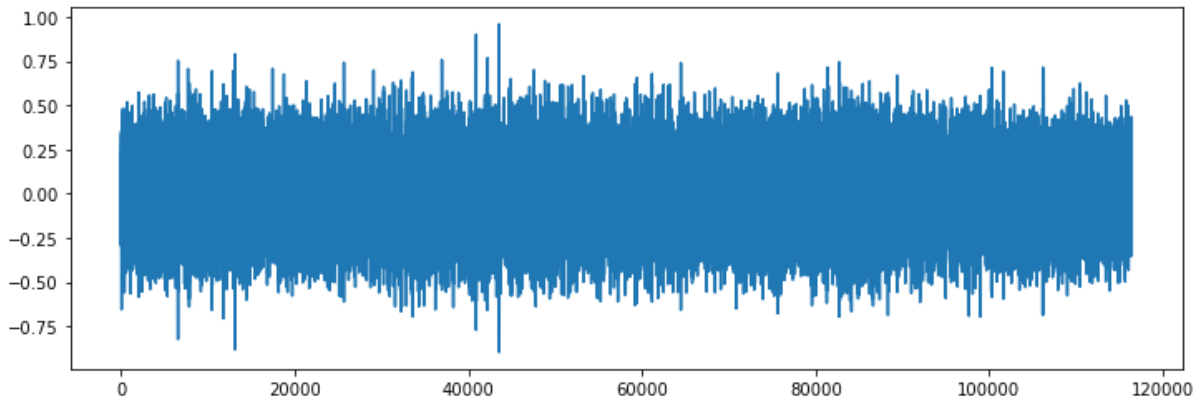
Out[13]:
```
'C:\\Users\\HP.NK\\TD'
```

In [14]:
```
audio_dataset_path = r'C:\\Users\\HP.NK\\TD\\Train'
```

In [15]:
```
audio_file_path = 'ma13_AI 0_AI 0.wav'
librosa_audio_data, librosa_sample_rate = librosa.load(audio_file_path)
print(librosa_audio_data)
import matplotlib.pyplot as plt
plt.figure(figsize = (12,4))
plt.plot(librosa_audio_data)
```

```
[-0.0261634   0.02733506  0.21072222 ...  0.02396899 -0.05842648
  0.        ]
```

Out[15]:
```
[<matplotlib.lines.Line2D at 0x15d26eb92e0>]
```



In [72]:
```
mfccs = librosa.feature.mfcc(y=librosa_audio_data, sr = sr, n_mfcc = 5)
np.set_printoptions(threshold=np.inf)  # Set printing options to display full array
```

In [63]:
```
import numpy as np
from tqdm import tqdm

max_frames = 216
mfcc_features = []
mfcc_labels = []
for index_num,row in tqdm(df.iterrows(), total=df.shape[0]):
    file_name = row['fname']
    labels = row["lable"]
    audio_path = os.path.join(audio_dataset_path,file_name)
    audio, sr = librosa.load(audio_path)
    mfcc = librosa.feature.mfcc(y = audio, sr=sr, n_mfcc = 13)

    if mfcc.shape[1] < max_frames:
        mfcc_padded = np.pad(mfcc, ((0,0),(0, max_frames - mfcc.shape[1]))).T
        mfcc_features.append(mfcc_padded)
    elif mfcc.shape[1] > max_frames:
        mfcc_truncated = mfcc[:, :max_frames].T
        mfcc_features.append(mfcc_truncated)
    else:
        mfcc_features.append(mfcc.T)




    mfcc_labels.append(labels)
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 6
9/69 [00:04<00:00, 17.09it/s]
```

In [64]:
```
mfcc_features = np.array(mfcc_features)
mfcc_labels = np.array(mfcc_labels)
```

In [54]:
```
mfcc_labels
```

Out[54]:
```
array(['Tool_in_contat', 'Tool_in_contat', 'Tool_in_contat',
       'Tool_in_contat', 'Machining', 'Machining', 'Machining',
       'Machining', 'Machining', 'Machining', 'Machining', 'Machining',
       'Machining', 'Machining', 'Machining', 'Machining', 'Machining',
       'Tool_approch', 'Tool_approch', 'Tool_approch', 'Tool_approch',
       'Tool_approch', 'Tool_approch', 'Tool_approch', 'Tool_approch',
       'Tool_approch', 'Tool_approch', 'Tool_approch', 'Tool_approch',
       'Tool_approch', 'Tool_approch', 'Tool_in_contat', 'Tool_in_contat',
       'Tool_in_contat', 'Tool_in_contat', 'Tool_in_contat', 'Machining',
       'Machining', 'Machining', 'Machining', 'Machining', 'Machining',
       'Machining', 'Machining', 'Machining', 'Machining',
       'Tool_in_contat', 'Tool_in_contat', 'Tool_in_contat',
       'Tool_in_contat', 'Tool_in_contat', 'Tool_in_contat',
       'Tool_in_contat', 'Tool_in_contat', 'Tool_approch', 'Tool_approch',
       'Tool_approch', 'Tool_approch', 'Tool_approch', 'Tool_approch',
       'Tool_approch', 'Tool_approch', 'Tool_approch', 'Tool_in_contat',
       'Tool_in_contat', 'Tool_in_contat', 'Tool_in_contat',
       'Tool_in_contat', 'Tool_in_contat'], dtype='<U14')
```

In [65]:
```
mfcc_features_flattened = mfcc_features.reshape(len(mfcc_features), -1)
```

In [67]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC


X_train, X_test, y_train, y_test = train_test_split(mfcc_features_flattened, mfcc_labels,
```

In [68]:
```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [69]:
```
svm = SVC()
svm.fit(X_train, y_train)
```

Out[69]:
```
SVC()
```

In [70]:
```
y_pred = svm.predict(X_test)
```

In [71]:
```
accuracy = np.mean(y_pred == y_test)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9090909090909091
```

In [ ]:

In [ ]: