LAB-7

M.HARIVIRINCHI

BL.EN.U4AIE21077

In [56]:
```
import numpy as np
import librosa
from hmmlearn import hmm
```

In [57]:
```
def extract_features(file_path, n_mfcc=13, n_fft=2048, hop_length=512):
    # Load audio file
    y, sr = librosa.load(file_path)
    # Extract STFT features
    stft = np.abs(librosa.stft(y, n_fft=n_fft, hop_length=hop_length))
    mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=n_mfcc)
    # Concatenate STFT and MFCC features
    features = np.concatenate([stft, mfccs], axis=0)
    return features.T
file_path = "Hari1.wav"
features = extract_features(file_path)
n_components = 3  # Number of states in HMM
covariance_type = "full"
model = hmm.GaussianHMM(n_components=n_components, covariance_type=covariance_type, n_
model.fit(features)
```

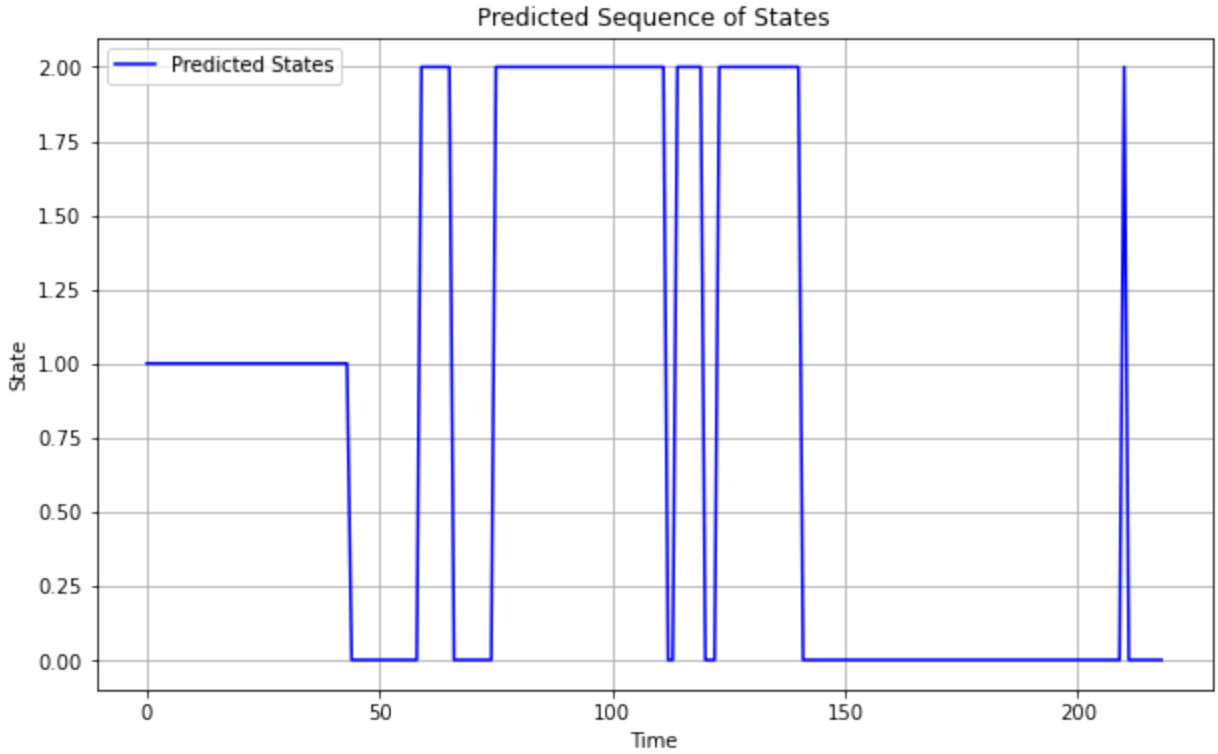
```
Fitting a model with 1620845 free scalar parameters with only 227322 data points will
result in a degenerate solution.
```

Out[57]:
```
GaussianHMM(covariance_type='full', n_components=3, n_iter=1000)
```

In [58]:
```
predicted_states = model.predict(features)
print("Predicted states sequence:", predicted_states)
```

```
Predicted states sequence: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0
 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 0 0 2 2 2 2 2 2 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0]
```

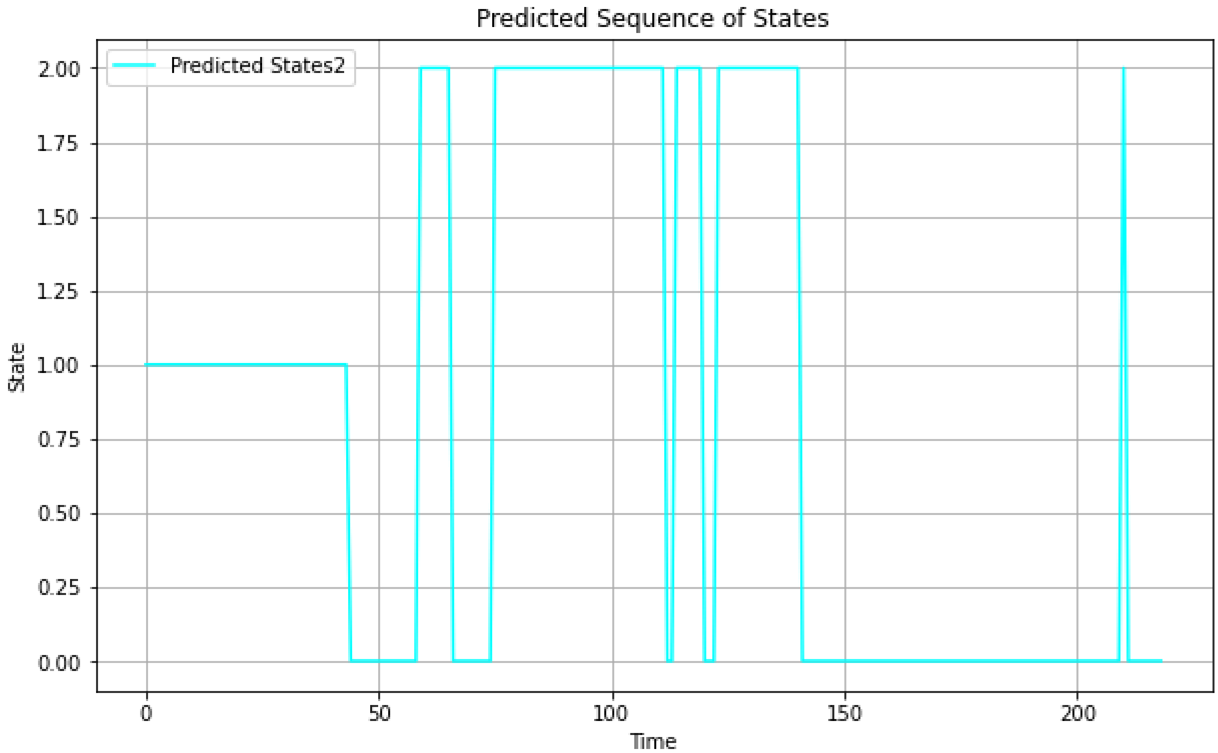
In [59]:
```
import matplotlib.pyplot as plot
plot.figure(figsize=(10, 6))
plot.plot(predicted_states, label='Predicted States', color='blue')
plot.xlabel('Time')
plot.ylabel('State')
plot.title('Predicted Sequence of States')
plot.legend()
plot.grid(True)
plot.show()
```



In [60]:
```
file_path2 = "Hari2.wav"
features2 = extract_features(file_path2)
n_components2 = 3  # Number of states in HMM
model2 = hmm.GaussianHMM(n_components=n_components2, covariance_type="diag", n_iter=10
model2.fit(features2)
predicted_states2 = model.predict(features2)
print("Predicted states sequence:", predicted_states2)
```

```
Predicted states sequence: [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0]
```

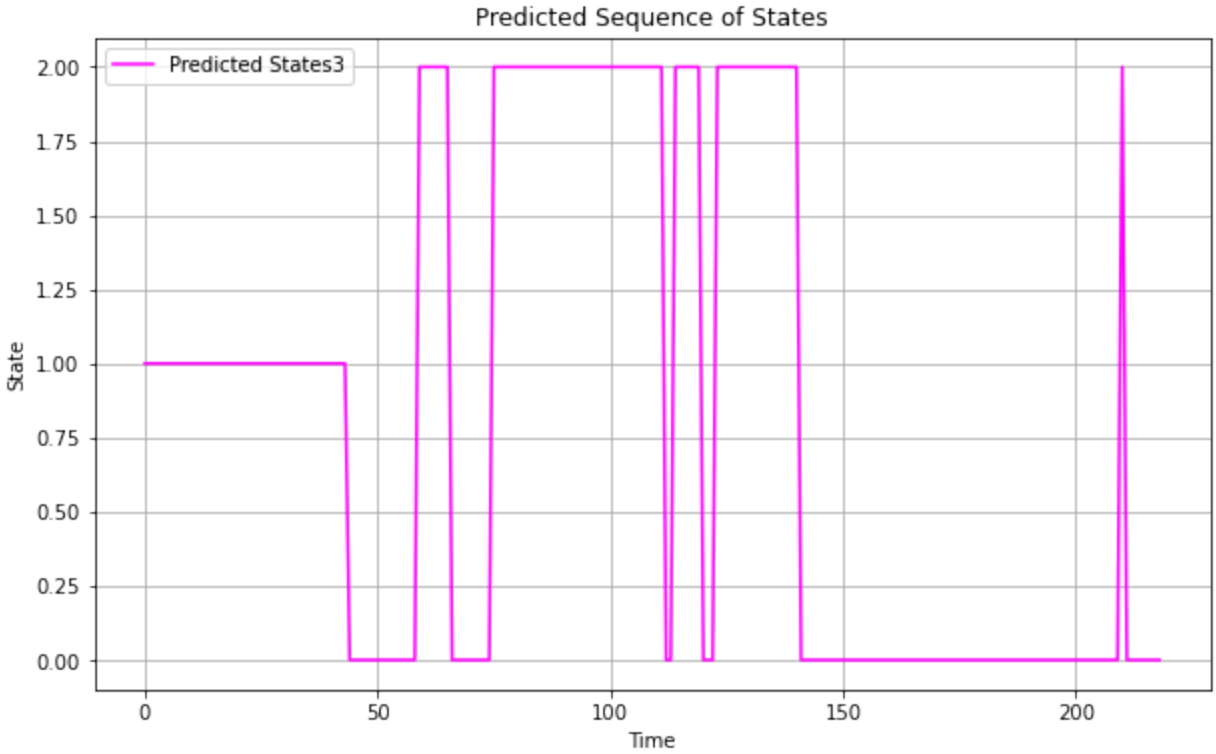
In [61]:
```
plot.figure(figsize=(10, 6))
plot.plot(predicted_states, label='Predicted States2', color='cyan')
plot.xlabel('Time')
plot.ylabel('State')
plot.title('Predicted Sequence of States')
plot.legend()
plot.grid(True)
plot.show()
```



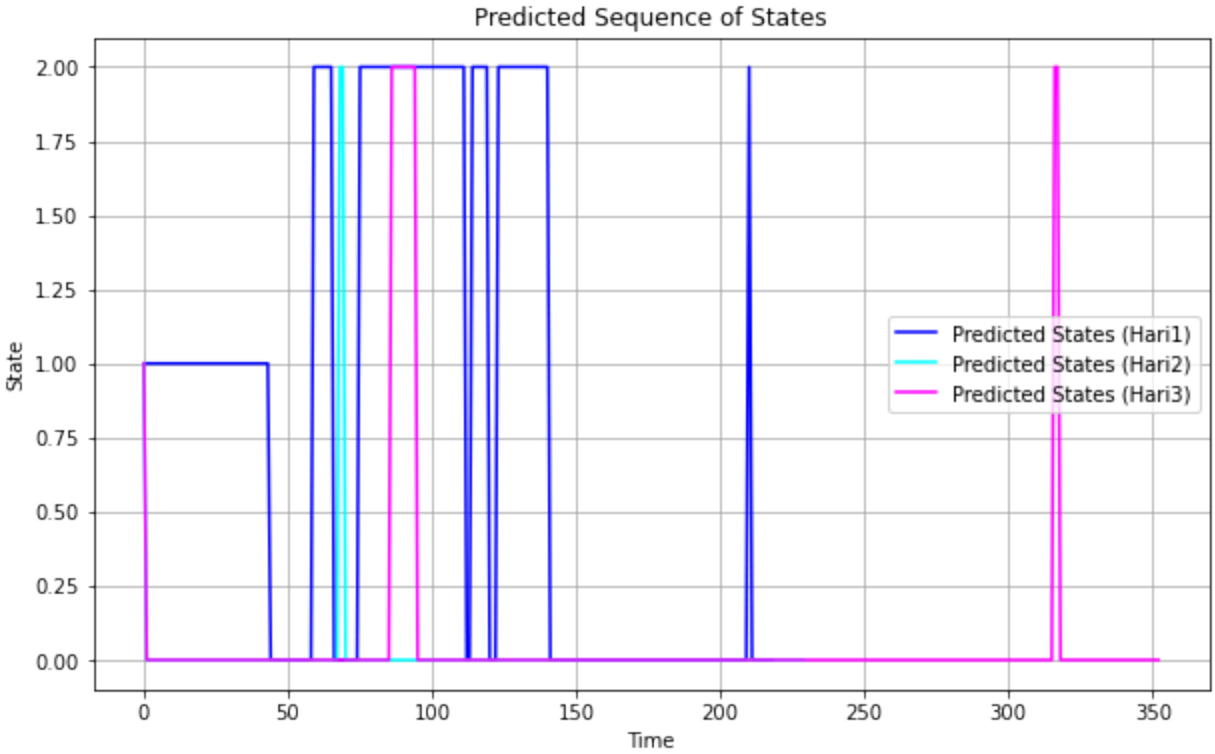
In [62]:
```
file_path3 = "Hari3.wav"
features3 = extract_features(file_path3)
n_components2 = 3  # Number of states in HMM
covariance_type3 = "full"
model3 = hmm.GaussianHMM(n_components=n_components2, covariance_type=covariance_type3,
model3.fit(features3)
predicted_states3 = model.predict(features3)
print("Predicted states sequence:", predicted_states3)
```

```
Fitting a model with 1620845 free scalar parameters with only 366414 data points will
result in a degenerate solution.
Predicted states sequence: [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

In [63]:
```
plot.figure(figsize=(10, 6))
plot.plot(predicted_states, label='Predicted States3', color='magenta')
plot.xlabel('Time')
plot.ylabel('State')
plot.title('Predicted Sequence of States')
plot.legend()
plot.grid(True)
plot.show()
```



In [64]:
```
plot.figure(figsize=(10, 6))
plot.plot(predicted_states, label='Predicted States (Hari1)', color='blue')
plot.plot(predicted_states2, label='Predicted States (Hari2)', color='cyan')
plot.plot(predicted_states3, label='Predicted States (Hari3)', color='magenta')
plot.xlabel('Time')
plot.ylabel('State')
plot.title('Predicted Sequence of States')
plot.legend()
plot.grid(True)
plot.show()
```


Predicted Sequence of States
State
Time
Predicted States (Hari1)
Predicted States (Hari2)
Predicted States (Hari3)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
50
100
150
200
250
300
350