DetectRotationsV2

January 13, 2022

[27]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
from datetime import timedelta
from scipy import signal
```

[28]:
```
Player = 9
Game = 2
df_Player = pd.read_csv('matrix_Player_' + str(Player) + '_game_' + str(Game) +␣
↪'.csv')
df_Player.columns =␣
↪['frAcc','frRoAcc','frDispl','frRoAng','frSpeed','timeLine','frameRotationalSpeedX','frameR
df_Action = pd.read_csv('Ned_DUI_Game_' + str(Game) + '.csv')
df_Player
```

[28]:

	frAcc	frRoAcc	frDispl	frRoAng	frSpeed	timeLine	\
0	0.000000	0.0	0.0	0.00000	0.000000	0.01	
1	0.000000	0.0	0.0	0.00000	0.000000	0.02	
2	0.000000	0.0	0.0	0.00000	0.000000	0.03	
3	0.000000	0.0	0.0	0.00000	0.000000	0.04	
4	0.000000	0.0	0.0	0.00000	0.000000	0.05	
…	…	…	…	…	…	…	
754220	-0.030424	4981.9	6585.3	-0.11980	-0.001860	7542.20	
754221	0.062668	4981.9	6585.3	-0.11723	-0.002164	7542.20	
754222	0.019899	4981.9	6585.2	-0.13407	-0.001537	7542.20	
754223	NaN	NaN	NaN	NaN	NaN	NaN	
754224	NaN	NaN	NaN	NaN	NaN	NaN	

	frameRotationalSpeedX	frameRotationalSpeedY	frameRotationalSpeedZ	\
0	NaN	NaN	NaN	
1	NaN	NaN	NaN	
2	NaN	NaN	NaN	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	
…	…	…	…	
754220	0.97222	2.4578	-0.98778	

```
754221                0.91000                2.5480               -1.05000
754222                0.92400                2.6320               -1.05000
754223                0.96444                2.5200               -1.01890
754224                0.88375                2.5200               -0.93625

        wheelRotationalSpeedX  wheelRotationalSpeedY  wheelRotationalSpeedZ  \
0                         NaN                    NaN                    NaN
1                         NaN                    NaN                    NaN
2                         NaN                    NaN                    NaN
3                         NaN                    NaN                    NaN
4                         NaN                    NaN                    NaN
…                           …                      …                      …
754220              -0.000000                  4.690                  2.730
754221              -0.056000                  4.494                  2.597
754222               0.070000                  4.620                  2.660
754223               0.070000                  4.620                  2.660
754224               0.081667                  4.585                  2.625

        frRoSpeed
0         0.00000
1         0.00000
2         0.00000
3         0.00000
4         0.00000
…           …
754220    0.25656
754221   -1.68370
754222        NaN
754223        NaN
754224        NaN

[754225 rows x 13 columns]
```

[29]:
```
# ----------------------------------------------------------------------

#LowPass Filter


#Lowpass filter design for rotation and wheelspeed to improve accuracy of code␣
↪(Butterworth filter)
df_Filter = df_Player
df_Filter = df_Filter.fillna(0)

Order = 5
cutoff_freq = 1.5
sampling_freq = 100
sampling_duration = len(df_Filter.wheelRotationalSpeedX)/100
```

```
normalized_cutoff_freq = 2 * cutoff_freq / sampling_freq
numerator_coeffs, denominator_coeffs = signal.butter(Order,␣
↪normalized_cutoff_freq)

filtered_WheelRotationspeed = signal.lfilter(numerator_coeffs,␣
↪denominator_coeffs, df_Filter.wheelRotationalSpeedX)
filtered_FrameRotationspeed = signal.lfilter(numerator_coeffs,␣
↪denominator_coeffs, df_Filter.frameRotationalSpeedZ)

# ------------------------------------------------------------------------
        #Operations


#Play with different operations to see clearer patterns
Sub = filtered_WheelRotationspeed + filtered_FrameRotationspeed
#Old_Conv1 = filtered_FrameRotationspeed / filtered_WheelRotationspeed
Conv1 = (abs(filtered_FrameRotationspeed)+abs(filtered_WheelRotationspeed))/
↪filtered_WheelRotationspeed



#Diff = arr[i+1] - arr[i]
DiffFrame = np.diff(df_Filter.frameRotationalSpeedZ,n=1)
DiffFrame = np.insert(DiffFrame,0,0)

DiffWheel = np.diff(df_Filter.wheelRotationalSpeedX,n=1)
DiffWheel = np.insert(DiffWheel,0,0)

Multi = filtered_WheelRotationspeed + filtered_FrameRotationspeed
#This isn't used


# ------------------------------------------------------------------------


#Set all data into a datafram
Data = pd.DataFrame({'Time':df_Player.timeLine,'WheelRotationspeed':
↪filtered_WheelRotationspeed,
                     'FrameRotationspeed':filtered_FrameRotationspeed,
                     'Sub':Sub ,'Conv':Conv1, 'DiffFrame':DiffFrame,␣
↪'DiffWheel':DiffWheel},
                     columns=['Time',␣
↪'WheelRotationspeed','FrameRotationspeed','Sub','Conv','DiffFrame','DiffWheel'])

#Convert data into chunks of n/100 of a second
```

```
n = 50  #chunk row size
Data_chunks = [Data[i:i+n] for i in range(0,Data.shape[0],n)]



#Search for the sprints
Startsprint = []
Stopsprint = []
Sprinting = False
Stop = True

#Use Sub en Conv to detect sprints
for chunks in Data_chunks:
    if abs(chunks['Conv'].max()) < 3 and chunks['Sub'].mean() > 300 and␣
 ↪chunks['WheelRotationspeed'].min() > 200  and Sprinting == False:
        Startsprint.append(chunks['Time'].min())
        Sprinting = True
        Stop = False
    elif abs(chunks['Conv'].max() < 3) and chunks['Sub'].mean() > 300 and␣
 ↪chunks['WheelRotationspeed'].min() > 200:
        Sprinting = True
        Stop = False
    elif Stop == False:
        Stopsprint.append(chunks['Time'].min())
        Sprinting = False
        Stop = True

#Use wheelrotation, framerotation and conv to detect rotations
Startrotate = []
Stoprotate = []
Rotate = False
Stop = True

count = 0

for chunks in Data_chunks:
    if abs(chunks['FrameRotationspeed'].max()) > 75 and abs(chunks['Conv'].
 ↪mean()) > 1.15 and count!=29 and Rotate == False:
        #this condition count!=29 is to make sure that it doesn't start a␣
 ↪rotation in the last
        #point of the graph because that way it would never enc

        Startrotate.append(chunks['Time'].min())
        Rotate = True
        Stop = False
```

```
    elif abs(chunks['FrameRotationspeed'].max()) > 75 and abs(chunks['Conv'].
↪mean()) > 1.15 and count == 29 and Rotate == True:
        #this comes due to the need of matching sizes (between startsprint and␣
↪stopsprint), as in one fast defense
        #theres a rotation that never ends in the plot, so I'm forcing it (had␣
↪to add Rotate == True so that this condition only
        #happens when a rotation has started before the last point of the graph)


        Stoprotate.append(chunks['Time'].min())
        Rotate = False
        Stop = True
    elif abs(chunks['FrameRotationspeed'].max()) > 75 and abs(chunks['Conv'].
↪mean()) > 1.15:#si abs menor que num sigue siendo rotation
        #Rotate =True is implicit
        Rotate = True
        Stop = False
    elif Stop == False :#Rotate==True and the previous conditions aren´t met,␣
↪thats implicit
        Stoprotate.append(chunks['Time'].min())
        Rotate = False
        Stop = True
    count +=1

# ------------------------------------------------------------------------

    #Pop small sprints out



#Filter Sprints by lenght, if length is below 2 delete sprint
Deleted = 0

if len(Startsprint) > len(Stopsprint):
    Startsprint.pop(-1)

if len(Startsprint) == len(Stopsprint):
    for i in range(0,len(Startsprint)-1):
        if (Stopsprint[i-Deleted] - Startsprint[i-Deleted]) < 5:
            Startsprint.pop(i-Deleted)
            Stopsprint.pop(i-Deleted)
            Deleted = Deleted + 1

# ------------------------------------------------------------------------
```

```
/tmp/ipykernel_48350/3389962776.py:28: RuntimeWarning: invalid value encountered
```

```
in true_divide
  Conv1 = (abs(filtered_FrameRotationspeed)+abs(filtered_WheelRotationspeed))/fi
ltered_WheelRotationspeed
```

[30]:
```
Starting =[]
Stopping = []

df_Filter = df_Player
df_Filter = df_Filter.fillna(0)

Order = 5
cutoff_freq = 1.5
sampling_freq = 100
sampling_duration = len(df_Filter.wheelRotationalSpeedX)/100


normalized_cutoff_freq = 2 * cutoff_freq / sampling_freq
numerator_coeffs, denominator_coeffs = signal.butter(Order,␣
 ↪normalized_cutoff_freq)

filtered_WheelRotationspeedX = signal.lfilter(numerator_coeffs,␣
 ↪denominator_coeffs, df_Filter.wheelRotationalSpeedX)
filtered_FrameRotationspeedZ = signal.lfilter(numerator_coeffs,␣
 ↪denominator_coeffs, df_Filter.frameRotationalSpeedZ)


df_Player['Sum_WheelX_FrameZ']  = df_Player.wheelRotationalSpeedX + df_Player.
 ↪frameRotationalSpeedZ
df_Player['Div_FrameZ_WheelX']  = df_Player.frameRotationalSpeedZ / df_Player.
 ↪wheelRotationalSpeedX
df_Player['Filt_WheelX'] = filtered_WheelRotationspeedX
df_Player['Filt_FrameZ'] = filtered_FrameRotationspeedZ

print(Startsprint, Stopsprint)

df_Player['Action'] = ""

for i  in range(0,len(Startsprint)):
    df_Player['Action'].iloc[int(Startsprint[i]*100):int(Stopsprint[i]*100)] = 1


df_Player.to_csv('matrix_Player_' + str(Player) + '_game_' + str(Game) +␣
 ↪'_All_Action.csv')
```

```
[997.01, 1019.0, 1041.5, 1062.5, 1078.0, 1086.0, 1093.5, 1105.5, 1138.0, 1161.0,
1181.5, 1201.0, 1232.5, 1248.5, 1257.0, 1427.5, 1485.5, 1511.5, 1538.0, 1566.0,
1593.0, 1626.5, 1659.5, 1676.0, 1734.0, 1832.0, 1912.0, 2216.5, 2224.5, 2235.5,
```

```
2433.5, 2465.5, 2490.0, 2558.0, 2598.0, 2631.0, 2647.0, 2671.5, 2684.0, 2786.0,
2800.0, 2850.0, 2873.0, 2888.5, 2940.0, 2956.5, 3004.5, 3025.5, 3049.0, 3074.0,
3093.5, 3118.5, 3137.5, 3156.0, 3256.0, 3276.0, 3329.5, 3372.0, 3506.5, 3530.0,
3555.5, 3602.5, 3628.0, 3654.0, 3685.0, 3703.5, 5040.0, 5053.5, 5080.5, 5095.0,
5135.5, 5146.5, 5177.0, 5207.0, 5324.0, 5345.5, 5369.0, 5400.5, 5411.0, 5434.0,
5462.0, 5510.5, 5594.5, 5621.0, 5704.0, 5745.5, 5842.5, 5895.0, 5927.5, 5955.5,
5974.5, 5993.0, 6044.5, 6059.5, 6076.5, 6897.0, 6912.5, 6923.0, 6950.5, 6963.0,
6981.0, 6998.0, 7051.0, 7186.5, 7206.0, 7225.5, 7249.0, 7281.0] [1008.0, 1041.0,
1047.0, 1074.0, 1083.5, 1093.0, 1103.0, 1121.0, 1148.5, 1181.0, 1186.5, 1219.5,
1246.0, 1254.0, 1262.0, 1433.5, 1490.5, 1516.5, 1543.5, 1571.0, 1598.0, 1632.0,
1665.5, 1681.5, 1739.0, 1837.0, 1917.5, 2221.5, 2231.5, 2241.5, 2441.0, 2471.0,
2495.0, 2566.5, 2607.5, 2640.0, 2654.5, 2680.0, 2689.5, 2795.0, 2807.0, 2856.0,
2879.0, 2894.0, 2946.5, 2963.5, 3011.5, 3031.0, 3057.0, 3081.0, 3100.0, 3126.0,
3144.0, 3162.5, 3264.5, 3283.0, 3335.0, 3378.0, 3513.5, 3536.0, 3563.5, 3611.5,
3634.5, 3661.0, 3691.5, 3708.5, 5046.0, 5079.0, 5092.5, 5132.5, 5146.0, 5172.0,
5201.0, 5227.5, 5333.0, 5353.0, 5377.0, 5405.5, 5419.5, 5440.0, 5470.5, 5519.0,
5604.5, 5626.5, 5710.5, 5755.5, 5848.5, 5903.5, 5934.0, 5962.0, 5981.0, 6001.0,
6051.0, 6067.0, 6089.0, 6903.0, 6918.5, 6928.0, 6955.5, 6977.0, 6989.5, 7008.0,
7057.5, 7193.0, 7211.5, 7234.5, 7254.5, 7285.0]
```

```
/opt/jupyterhub/anaconda/lib/python3.8/site-
packages/pandas/core/indexing.py:1637: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
```

[31]:
```
print(len(Stoprotate))

print(len(Startrotate))
```

```
688
688
```

[ ]:
```

```