diceface_corrected_data_analysis

January 12, 2021

Adnan Akbas # Diceface corrected data analysis

[6]:
```
from helpers import pandas_helper as ph
from helpers import math_helper as mh
from sensors.activpal import *

import matplotlib.pyplot as plt

activpal_utilities = Activpal()
```

[7]:
```
#initialization
correspondent = 'BMR002'

activities = ph.read_csv_activiteiten(correspondent)

normalized_activpal = ph.read_activpal_20_diceface(correspondent)
original_activpal = activpal_utilities.read_data(correspondent)
```

[8]:
```
original_activpal.tail()
```

[8]:
```
                                          pal_time  pal_accX  pal_accY  \
pal_time
2019-09-24 12:45:02.500001 2019-09-24 12:45:02.500001       118       116
2019-09-24 12:45:02.550001 2019-09-24 12:45:02.550001       121       117
2019-09-24 12:45:02.600000 2019-09-24 12:45:02.600000       121       116
2019-09-24 12:45:02.650000 2019-09-24 12:45:02.650000       119       116
2019-09-24 12:45:02.700000 2019-09-24 12:45:02.700000       121       117

                           pal_accZ
pal_time
2019-09-24 12:45:02.500001        70
2019-09-24 12:45:02.550001        73
2019-09-24 12:45:02.600000        69
2019-09-24 12:45:02.650000        70
2019-09-24 12:45:02.700000        73
```

[9]:
```
normalized_activpal.tail()
```

[9]:
```
                            pal_accX  pal_accY  pal_accZ  pal_diceFace
pal_time
2019-09-24 12:45:02:500001     118.0     116.0      70.0           NaN
2019-09-24 12:45:02:550001     121.0     117.0      73.0           NaN
2019-09-24 12:45:02:600000     121.0     116.0      69.0           NaN
2019-09-24 12:45:02:650000     119.0     116.0      70.0           NaN
2019-09-24 12:45:02:700000     121.0     117.0      73.0           NaN
```

[10]:
```
original_activpal['pal_accX'] = mh.
 ↪convert_value_to_g(original_activpal['pal_accX'])
original_activpal['pal_accY'] = mh.
 ↪convert_value_to_g(original_activpal['pal_accY'])
original_activpal['pal_accZ'] = mh.
 ↪convert_value_to_g(original_activpal['pal_accZ'])

normalized_activpal['pal_accX'] = mh.
 ↪convert_value_to_g(normalized_activpal['pal_accX'])
normalized_activpal['pal_accY'] = mh.
 ↪convert_value_to_g(normalized_activpal['pal_accY'])
normalized_activpal['pal_accZ'] = mh.
 ↪convert_value_to_g(normalized_activpal['pal_accZ'])
```

[11]:
```
original_activpal.head()
```

[11]:
```
                                          pal_time  pal_accX  pal_accY  \
pal_time
2019-09-16 12:45:06.999999 2019-09-16 12:45:06.999999 -0.682540  0.095238
2019-09-16 12:45:07.049999 2019-09-16 12:45:07.049999 -0.746032  0.095238
2019-09-16 12:45:07.099998 2019-09-16 12:45:07.099998 -0.682540  0.047619
2019-09-16 12:45:07.149998 2019-09-16 12:45:07.149998 -0.682540  0.031746
2019-09-16 12:45:07.199997 2019-09-16 12:45:07.199997 -0.682540  0.063492

                            pal_accZ
pal_time
2019-09-16 12:45:06.999999  1.015873
2019-09-16 12:45:07.049999  0.952381
2019-09-16 12:45:07.099998  0.936508
2019-09-16 12:45:07.149998  1.000000
2019-09-16 12:45:07.199997  1.015873
```

[12]:
```
normalized_activpal.head()
```

[12]:
```
                            pal_accX  pal_accY  pal_accZ  pal_diceFace
pal_time
2019-09-16 12:45:06:999999  0.682540 -1.015873 -0.095238           5.0
2019-09-16 12:45:07:049999  0.746032 -0.952381 -0.095238           5.0
2019-09-16 12:45:07:099998  0.682540 -0.936508 -0.047619           5.0
```

```
2019-09-16 12:45:07:149998  0.682540 -1.000000 -0.031746           5.0
2019-09-16 12:45:07:199997  0.682540 -1.015873 -0.063492           5.0
```

[13]:
```
#helper functions

def generate_filters(start, stop):
    return [
        # activity filter for normalized_activpal dataframe
        (normalized_activpal.index >= start) & (normalized_activpal.index <=␣
↪stop),
        # activity filter for original_activpal dataframe
        (original_activpal.index >= start) & (original_activpal.index <= stop)
    ]
```

0.0.1 activities

[14]:
```
activities
```

[14]:
```
                          start                stop
activiteit
springen       2019-09-16 14:29:20  2019-09-16 14:30:18
traplopen      2019-09-16 14:31:18  2019-09-16 14:32:04
fietsen licht  2019-09-16 14:41:29  2019-09-16 14:46:29
fietsen zwaar  2019-09-16 14:46:29  2019-09-16 14:51:29
lopen          2019-09-16 15:12:00  2019-09-16 15:17:00
rennen         2019-09-16 15:17:00  2019-09-16 15:22:00
zitten         2019-09-16 15:31:00  2019-09-16 15:36:00
staan          2019-09-16 15:45:00  2019-09-16 15:50:00
```

0.0.2 plotting

[15]:
```
def plot_orignal_and_normalized_activpal_on_activity(activity):
    specifick_activity = activities.loc[activity]

    activpal_filters = generate_filters(specifick_activity.
↪start,specifick_activity.stop)

    normalized_activpal_activity = normalized_activpal[activpal_filters[0]].
↪copy()
    original_activpal_activity = original_activpal[activpal_filters[1]].copy()


    #plot original_activpal
    plt.figure(figsize=(20,5))
```

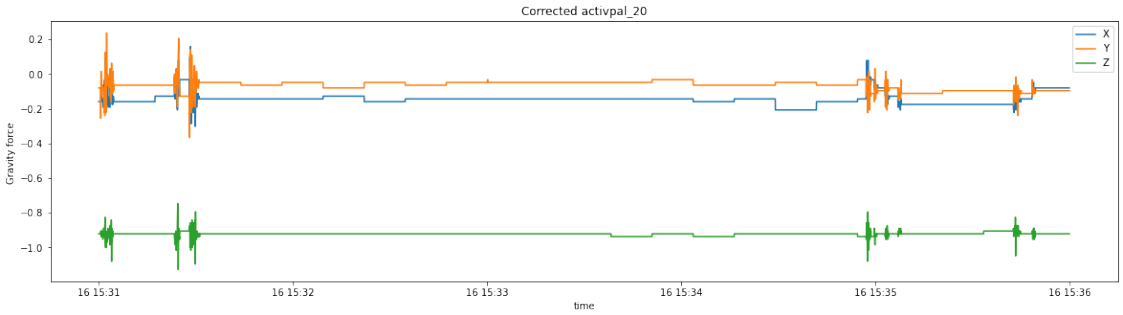
```
   plt.plot(original_activpal_activity.index, original_activpal_activity.
↪pal_accX, label='X')
   plt.plot(original_activpal_activity.index, original_activpal_activity.
↪pal_accY, label='Y')
   plt.plot(original_activpal_activity.index, original_activpal_activity.
↪pal_accZ, label='Z')

   plt.xlabel('time' )
   plt.ylabel('Gravity force ')
   plt.title('Corrected activpal_20')
   plt.legend()

   plt.show()

   #plot normalized_activpal
   plt.figure(figsize=(20,5))

   plt.plot(normalized_activpal_activity.index, normalized_activpal_activity.
↪pal_accX, label='X')
   plt.plot(normalized_activpal_activity.index, normalized_activpal_activity.
↪pal_accY, label='Y')
   plt.plot(normalized_activpal_activity.index, normalized_activpal_activity.
↪pal_accZ, label='Z')

   plt.xlabel('time' )
   plt.ylabel('Gravity force ')
   plt.title('Corrected activpal_20')
   plt.legend()

   print(normalized_activpal_activity.pal_diceFace.unique())
   plt.figure(figsize=(20,5))


   plt.plot(normalized_activpal_activity.index, normalized_activpal_activity.
↪pal_diceFace, label='X')

   plt.xlabel('time' )
   plt.ylabel('Diceface')
   plt.legend()


   plt.show()
```

[16]:
```
def plot_orignal_and_normalized_activpal_on_time(start,stop):
```

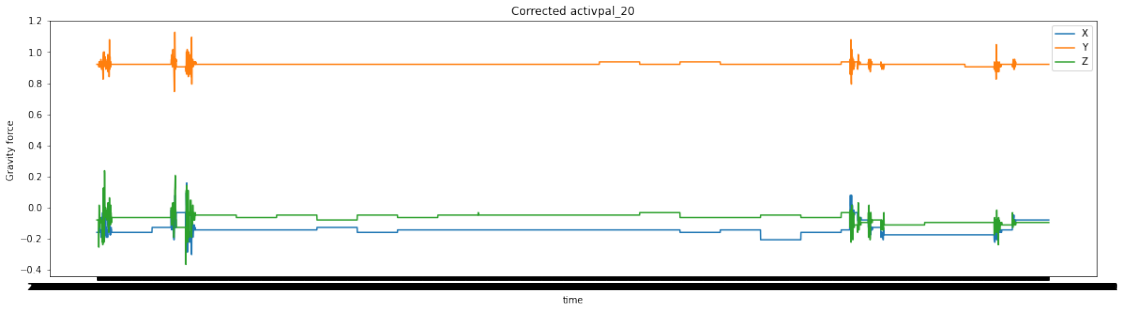
```
   activpal_filters = generate_filters(start,stop)

   normalized_activpal_filtered = normalized_activpal[activpal_filters[0]].
↪copy()
   original_activpal_filtered = original_activpal[activpal_filters[1]].copy()

   #plot original_activpal
   plt.figure(figsize=(20,5))

   plt.plot(original_activpal_filtered.index, original_activpal_filtered.
↪pal_accX, label='X')
   plt.plot(original_activpal_filtered.index, original_activpal_filtered.
↪pal_accY, label='Y')
   plt.plot(original_activpal_filtered.index, original_activpal_filtered.
↪pal_accZ, label='Z')

   plt.xlabel('time')
   plt.ylabel('Gravity force')
   plt.title('Corrected activpal_20')
   plt.legend()

   plt.show()

   #plot normalized_activpal

   plt.figure(figsize=(20,5))

   plt.plot(normalized_activpal_filtered.index, normalized_activpal_filtered.
↪pal_accX, label='X')
   plt.plot(normalized_activpal_filtered.index, normalized_activpal_filtered.
↪pal_accY, label='Y')
   plt.plot(normalized_activpal_filtered.index, normalized_activpal_filtered.
↪pal_accZ, label='Z')

   plt.xlabel('time' )
   plt.ylabel('Gravity force ')
   plt.title('Corrected activpal_20')
   plt.legend()

   print(normalized_activpal_filtered.pal_diceFace.unique())


   plt.show()
```

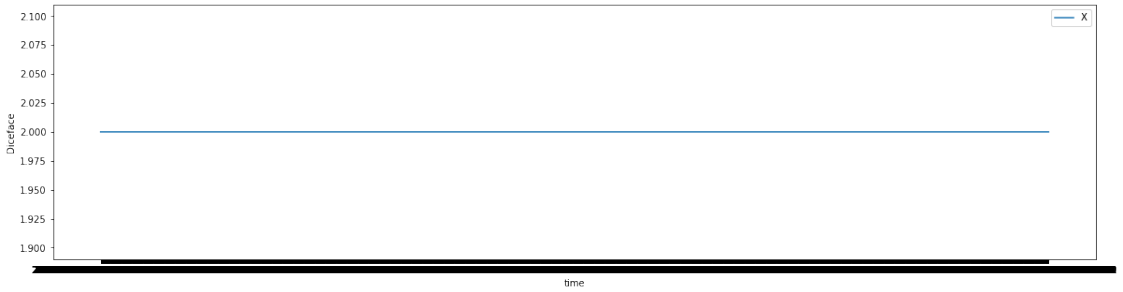
sitting

[17]:
```
plot_orignal_and_normalized_activpal_on_activity('zitten')
```



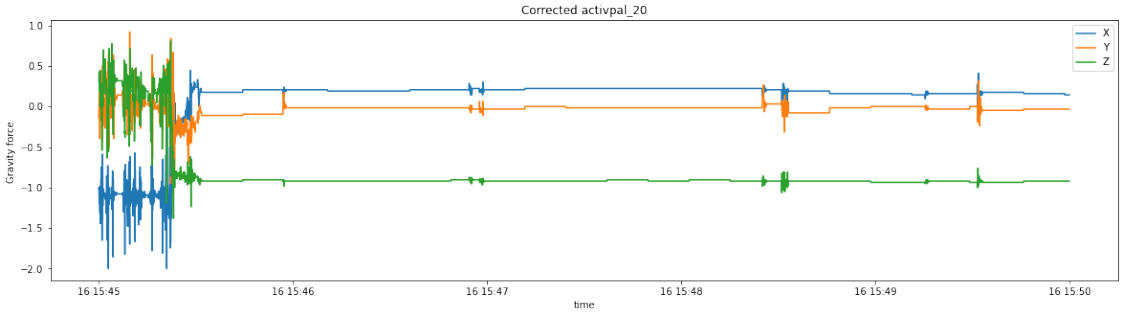
[2.]



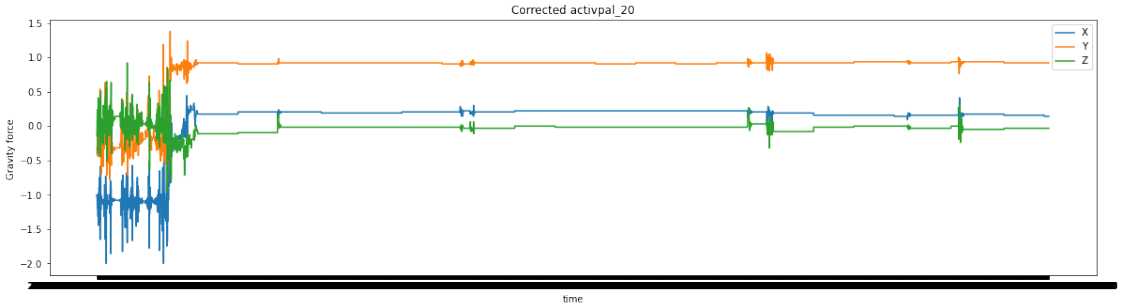


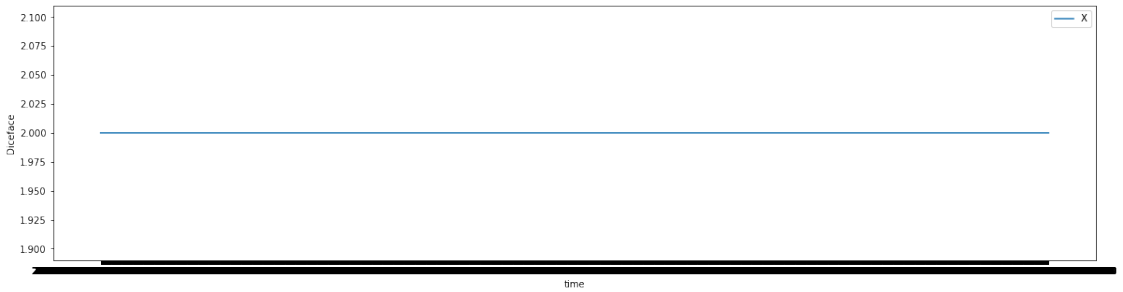
Standing

[18]:
```
plot_orignal_and_normalized_activpal_on_activity('staan')
```



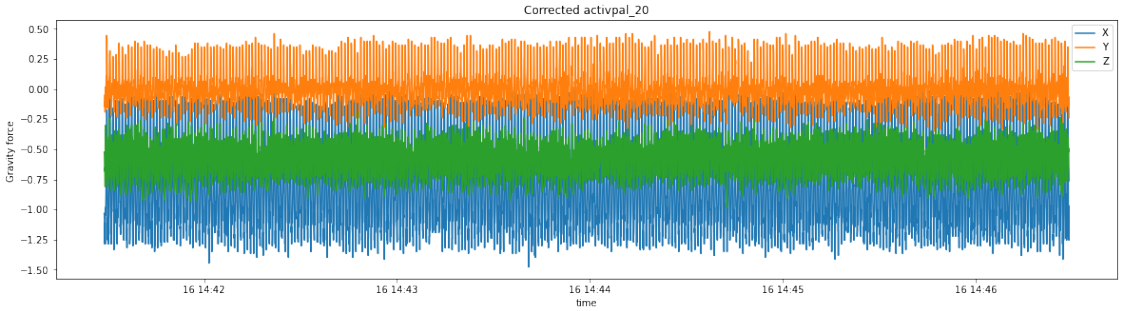
[2.]



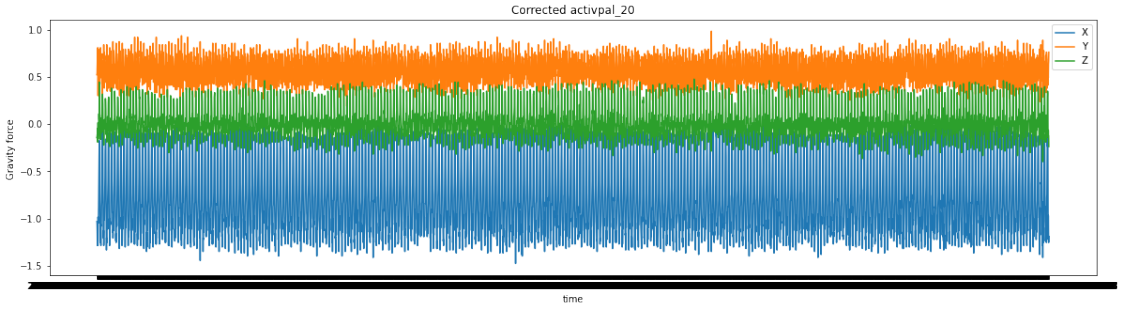


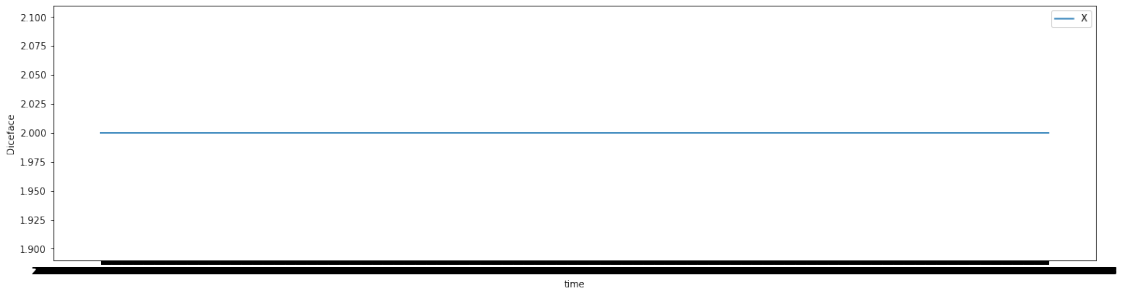
cycling

[19]:
```
plot_orignal_and_normalized_activpal_on_activity('fietsen licht')
```



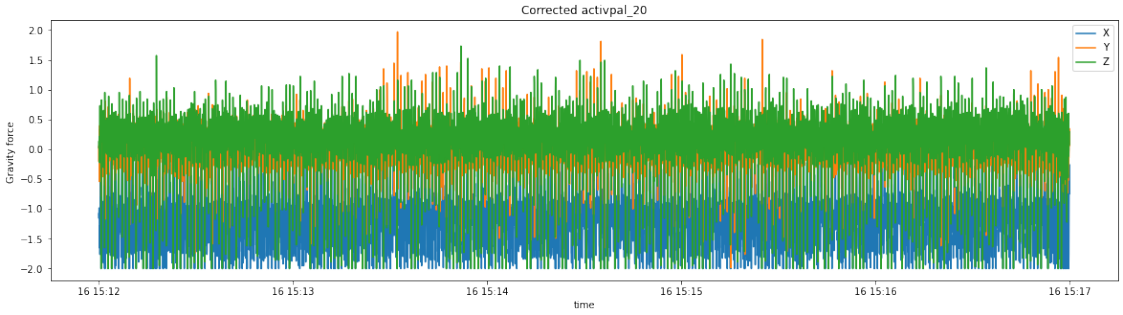
[2.]



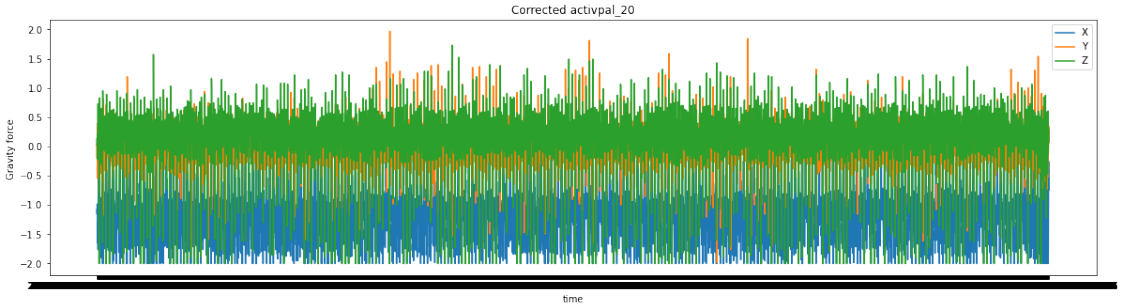


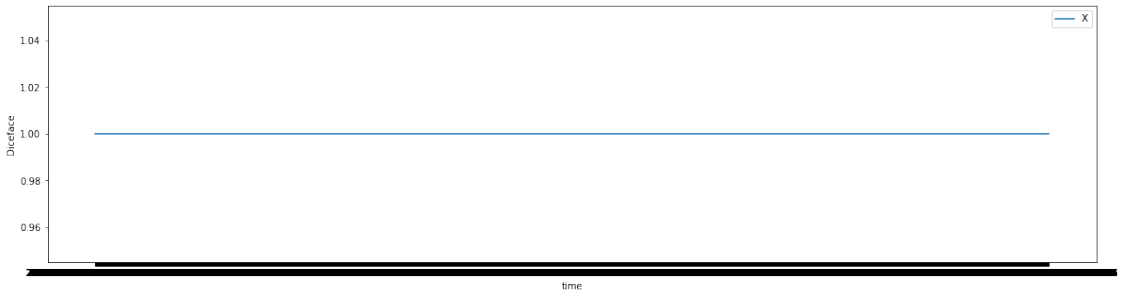
walking

[20]:
```
plot_orignal_and_normalized_activpal_on_activity('lopen')
```



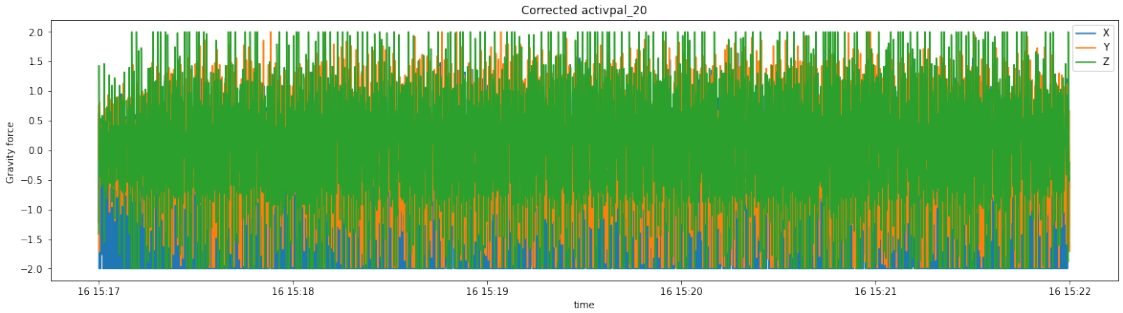
[1.]



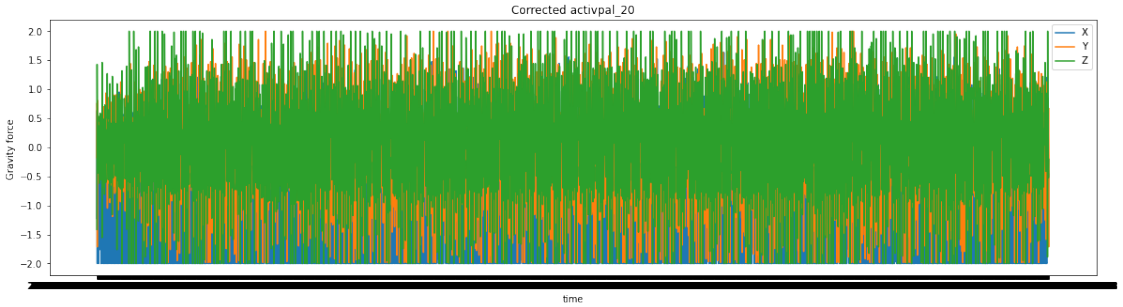


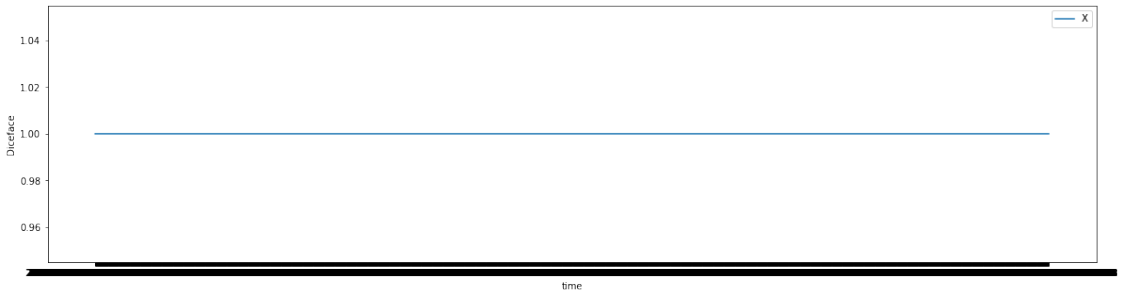
running

[21]:
```
plot_orignal_and_normalized_activpal_on_activity('rennen')
```



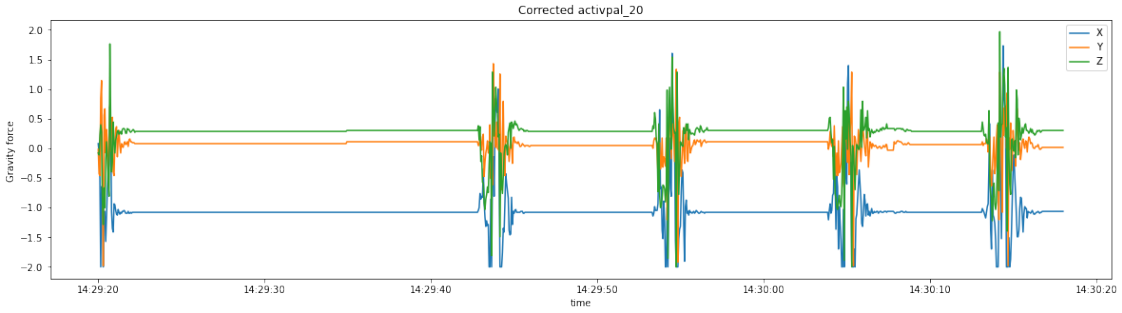
[1.]



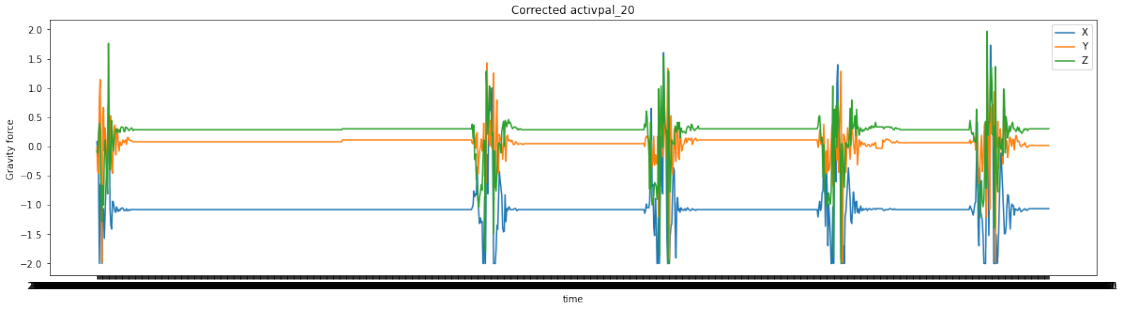


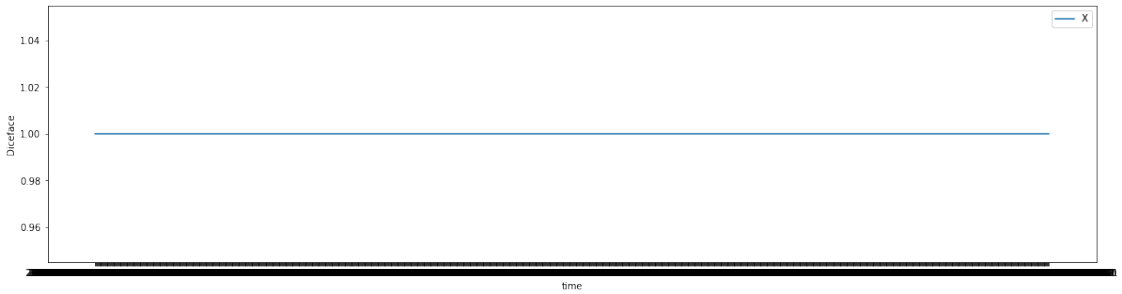
Jumping

[22]:
```
plot_orignal_and_normalized_activpal_on_activity('springen')
```



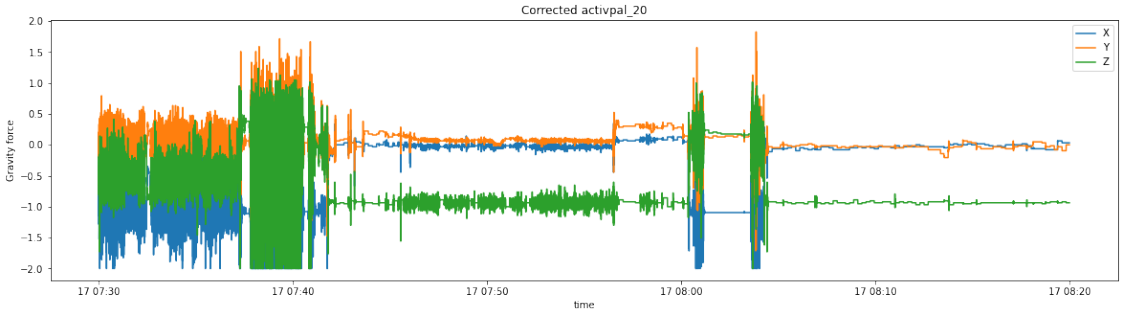
[1.]



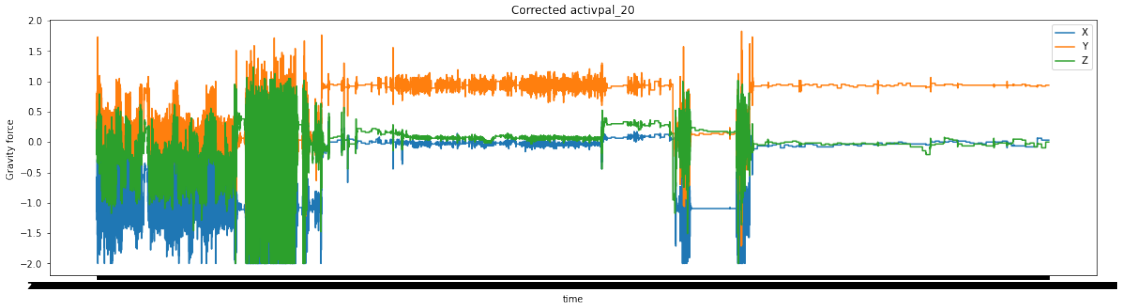


sleeping and waking up

[23]:
```
plot_orignal_and_normalized_activpal_on_time('2019-09-17 07:30:00', '2019-09-17␣
↪08:20:00')
```


Corrected activpal_20
Gravity force
time
X
Y
Z
17 07:30
17 07:40
17 07:50
17 08:00
17 08:10
17 08:20

```
[2. 1.]
```


Corrected activpal_20
Gravity force
time
X
Y
Z

[ ]:

[ ]: