

# Datenanalyse MediaPipe und DeepMotion

## 1. DeepMotion Nicht Kompensiert 03

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.spatial import distance
from scipy.interpolate import interp1d
from matplotlib.offsetbox import OffsetImage, AnnotationBbox
```

```
In [ ]: df_deepM_03 = pd.read_csv('../Data/KeypointsBereinigtNichtKompensiert_03.
df_deepM_03.head(5)
```

```
Out[ ]:   compensation  frame          path      x_0
0            0     1 /Users/salomekoller/Library/CloudStorage/OneDr...  0.0000 -9
1            0     2 /Users/salomekoller/Library/CloudStorage/OneDr... -0.2305 -9
2            0     3 /Users/salomekoller/Library/CloudStorage/OneDr... -0.5451 -9
3            0     4 /Users/salomekoller/Library/CloudStorage/OneDr... -0.8911 -9
4            0     5 /Users/salomekoller/Library/CloudStorage/OneDr... -1.2253 -9
```

5 rows × 102 columns

### 1.1 Analyse Form

```
In [ ]: print('Dimension:', df_deepM_03.shape)
print('Number of rows:', df_deepM_03.shape[0])
print('Number of columns:', df_deepM_03.shape[1])
```

```
Dimension: (937, 102)
Number of rows: 937
Number of columns: 102
```

### 1.2 Splitting Frames

```
In [ ]: # Variablen initialisieren
frame_count = 0
frames_until_reset = []

# Iterieren über Dataframe, um Frames mit '1' zu finden
for index, row in df_deepM_03.iterrows():
    frame_count += 1

    if row["frame"] == 1 and frame_count > 1:
        frames_until_reset.append(frame_count - 1)
```

```

if frame_count > 0:
    frames_until_reset.append(frame_count)

print("Number of frames until reset for each cycle:", frames_until_reset)

```

Number of frames until reset for each cycle: [182, 373, 565, 740, 937]

In [ ]: # Einzelne Bewegungen werden in verschiedene Dataframes gepackt

```

dfs = []
start = 0
for end in frames_until_reset:
    dfs.append(df_deepM_03.iloc[start:end])
    start = end

print("Number of splits:", len(dfs))

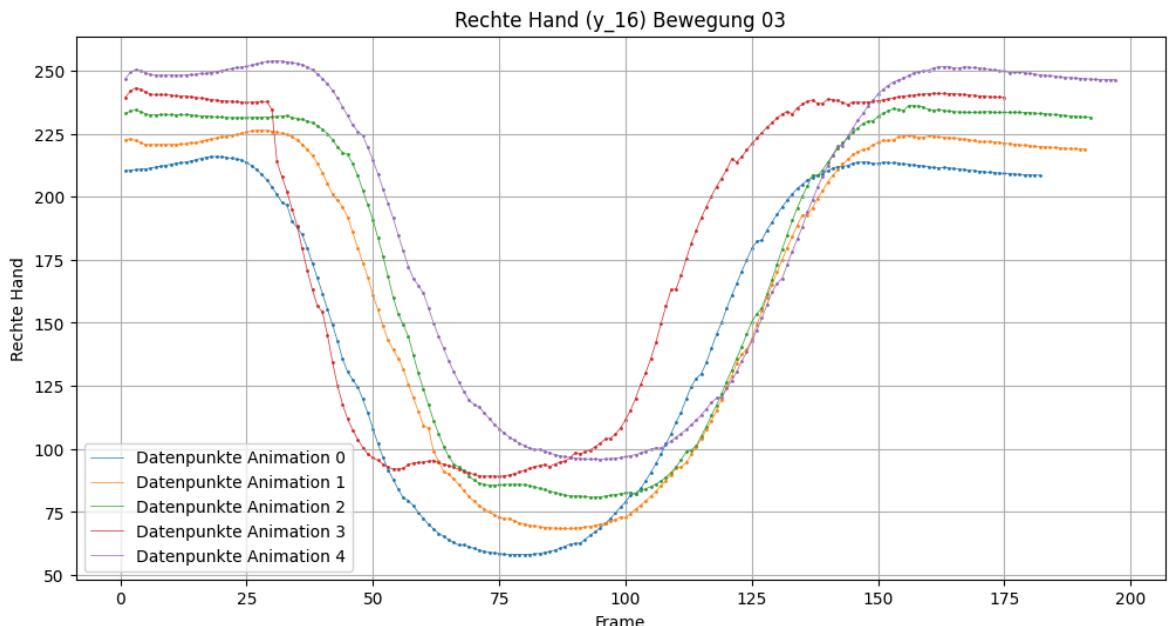
```

Number of splits: 5

## 1.3 Bewegungsanalyse anhand rechter Handbewegung (Nicht Kompensiert 03)

In [ ]: plt.figure(figsize=(12, 6))
for i, df in enumerate(dfs):
 plt.scatter(df["frame"], df["y\_16"], marker='o', color=f'C{i}', s=1)
 plt.plot(df["frame"], df["y\_16"], color=f'C{i}', linestyle='-', linewidth=2)

plt.title("Rechte Hand (y\_16) Bewegung 03")
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()



### 1.3.1 Berechnung des Mean

In [ ]: means03 = []
mean\_values\_03 = df\_deepM\_03[['frame', 'x\_0', 'y\_0', 'z\_0', 'x\_1', 'y\_1',
 'x\_4', 'y\_4', 'z\_4', 'x\_5', 'y\_5', 'z\_5', 'x\_6', 'y\_6',
 'x\_8', 'y\_8', 'z\_8', 'x\_9', 'y\_9', 'z\_9', 'x\_10', 'y\_10']]

```

'x_12', 'y_12', 'z_12', 'x_13', 'y_13', 'z_13', 'x_14'
'x_16', 'y_16', 'z_16', 'x_17', 'y_17', 'z_17', 'x_18'
'x_20', 'y_20', 'z_20', 'x_21', 'y_21', 'z_21', 'x_22'
'x_24', 'y_24', 'z_24', 'x_25', 'y_25', 'z_25', 'x_26'
'x_28', 'y_28', 'z_28', 'x_29', 'y_29', 'z_29', 'x_30'
'x_32', 'y_32', 'z_32']]].groupby('frame').mean()

means03.append(mean_values_03)
mean_values_03.head(5)

```

Out[ ]:

	x_0	y_0	z_0	x_1	y_1	z_1	x_
frame							
<b>1</b>	65.99356	18.27286	-171.467506	-993.7168	-312.4525	-60.97508	-993.716
<b>2</b>	66.85646	19.80700	-171.254204	-993.7168	-312.4525	-60.97508	-993.716
<b>3</b>	67.98854	20.18948	-170.908414	-993.7168	-312.4525	-60.97508	-993.716
<b>4</b>	68.69194	19.89282	-170.505374	-993.7168	-312.4525	-60.97508	-993.716
<b>5</b>	69.62770	19.35010	-170.065084	-993.7168	-312.4525	-60.97508	-993.716

5 rows × 99 columns

### 1.3.2 Bewegungsmuster mit Mean

In [ ]:

```

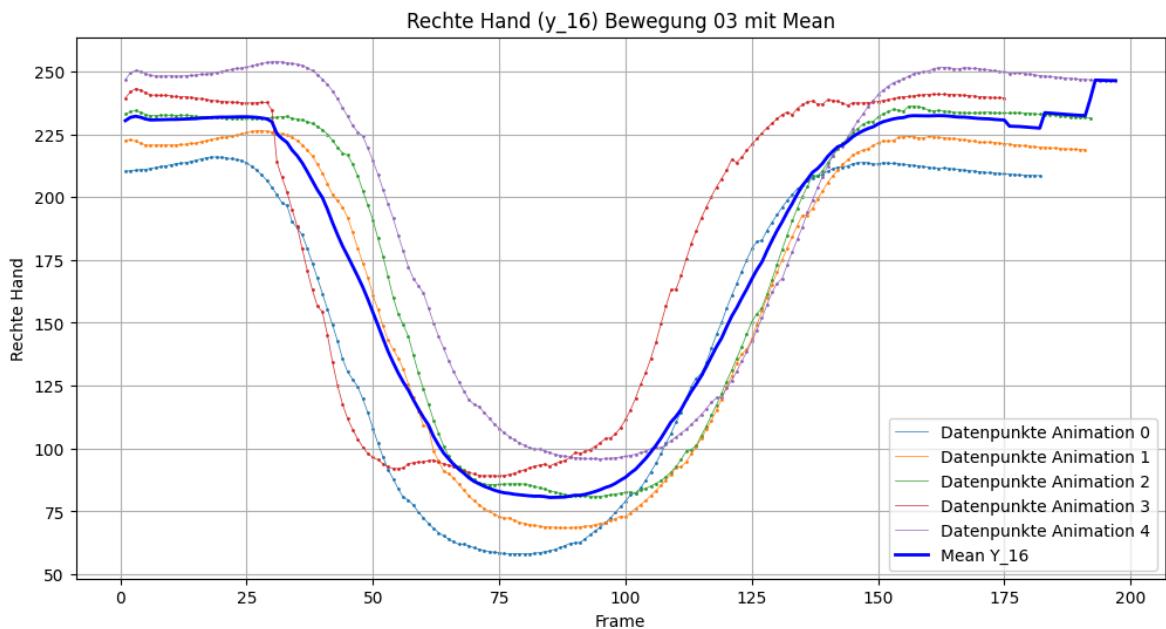
plt.figure(figsize=(12, 6))

for i, df in enumerate(dfs):
    plt.scatter(df["frame"], df["y_16"], marker='o', color=f'C{i}', s=1)
    plt.plot(df["frame"], df["y_16"], color=f'C{i}', linestyle='-', linewidth=2)

plt.plot(mean_values_03.index, mean_values_03["y_16"], color='b', linestyle='--', linewidth=2)

plt.title("Rechte Hand (y_16) Bewegung 03 mit Mean")
plt.xlabel("Frame")
plt.legend(loc='best')
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.show()

```



## 2. DeepMotion Kompensiert 01

```
In [ ]: df_deepM01 = pd.read_csv('../Data/KeypointsBereinigtKompensiert_01.csv', df_deepM01.head(5))
```

```
Out[ ]:
```

	compensation	frame	path	x_0	
0	1	1	/Users/salomekoller/Library/CloudStorage/OneDr...	0.000	-100
1	1	2	/Users/salomekoller/Library/CloudStorage/OneDr...	1.909	-100
2	1	3	/Users/salomekoller/Library/CloudStorage/OneDr...	2.832	-100
3	1	4	/Users/salomekoller/Library/CloudStorage/OneDr...	3.748	-100
4	1	5	/Users/salomekoller/Library/CloudStorage/OneDr...	5.316	-100

5 rows × 102 columns

### 2.1 Analyse Form

```
In [ ]: print('Dimension:', df_deepM01.shape)
print('Number of rows:', df_deepM01.shape[0])
print('Number of columns:', df_deepM01.shape[1])
```

Dimension: (1096, 102)  
Number of rows: 1096  
Number of columns: 102

### 2.2 Splitting Frames

```
In [ ]: # Variablen initialisieren
frame_count = 0
frames_until_reset = []
```

```

# Iterieren über Dataframe, um Frames mit '1' zu finden
for index, row in df_deepM01.iterrows():
    frame_count += 1

    if row["frame"] == 1 and frame_count > 1:
        frames_until_reset.append(frame_count - 1)

if frame_count > 0:
    frames_until_reset.append(frame_count)

print("Number of frames until reset for each cycle:", frames_until_reset)

```

Number of frames until reset for each cycle: [210, 431, 652, 873, 1096]

```

In [ ]: # Einzelne Bewegungen werden in verschiedene Dataframes gepackt
dfsKomp01 = []
start = 0
for end in frames_until_reset:
    dfsKomp01.append(df_deepM01.iloc[start:end])
    start = end

print("Number of splits:", len(dfsKomp01))

```

Number of splits: 5

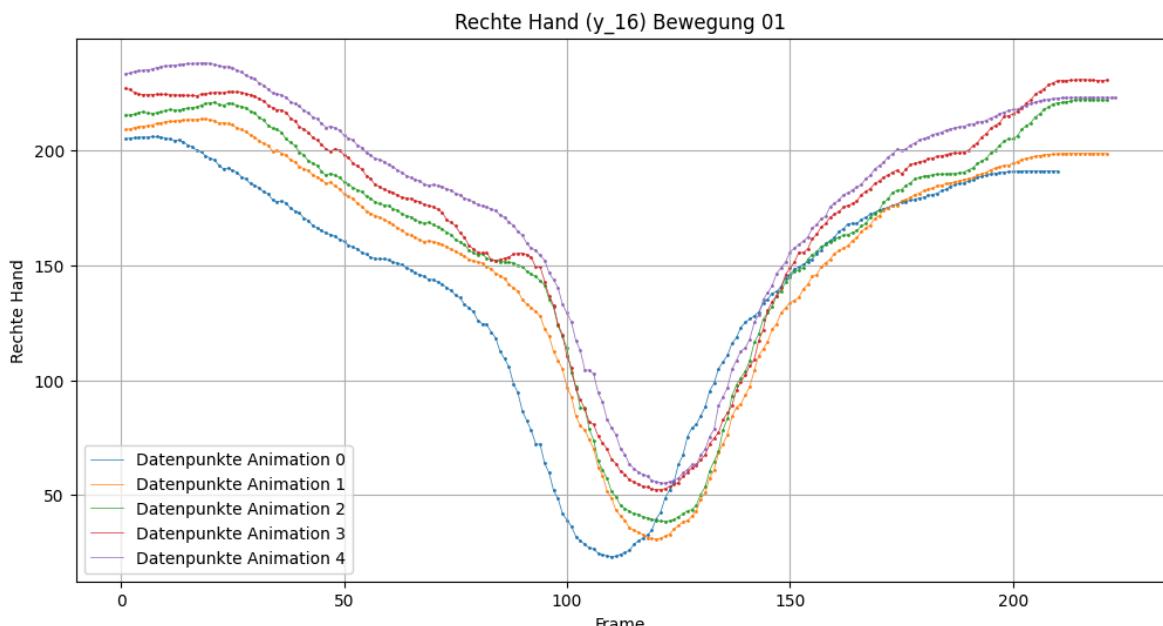
## 2.3 Bewegungsanalyse anhand rechter Hand (Kompenziert 01)

```

In [ ]: plt.figure(figsize=(12, 6))
for i, df in enumerate(dfsKomp01):
    plt.scatter(df["frame"], df["y_16"], marker='o', color=f'C{i}', s=1)
    plt.plot(df["frame"], df["y_16"], color=f'C{i}', linestyle='-', linewidth=2)

plt.title("Rechte Hand (y_16) Bewegung 01")
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()

```



### 2.3.1 Berechnung des Mean

```
In [ ]: means01 = []
mean_values_01 = df_deepM01[['frame', 'x_0', 'y_0', 'z_0', 'x_1', 'y_1',
                               'x_4', 'y_4', 'z_4', 'x_5', 'y_5', 'z_5', 'x_6', 'y_6',
                               'x_8', 'y_8', 'z_8', 'x_9', 'y_9', 'z_9', 'x_10', 'y_10',
                               'x_12', 'y_12', 'z_12', 'x_13', 'y_13', 'z_13', 'x_14',
                               'x_16', 'y_16', 'z_16', 'x_17', 'y_17', 'z_17', 'x_18',
                               'x_20', 'y_20', 'z_20', 'x_21', 'y_21', 'z_21', 'x_22',
                               'x_24', 'y_24', 'z_24', 'x_25', 'y_25', 'z_25', 'x_26',
                               'x_28', 'y_28', 'z_28', 'x_29', 'y_29', 'z_29', 'x_30',
                               'x_32', 'y_32', 'z_32']].groupby('frame').mean()
means01.append(mean_values_01)

mean_values_01.head(5)
```

```
Out[ ]:      x_0      y_0      z_0      x_1      y_1      z_1      x_2
frame
1  42.9238  16.61496 -200.431752 -1007.957 -310.477 -48.49379 -1007.957
2  44.8634  16.47286 -202.449376 -1007.957 -310.477 -48.49379 -1007.957
3  46.3038  16.14494 -204.223192 -1007.957 -310.477 -48.49379 -1007.957
4  47.5600  15.77780 -206.149102 -1007.957 -310.477 -48.49379 -1007.957
5  49.5454  15.59410 -208.155064 -1007.957 -310.477 -48.49379 -1007.957
```

5 rows × 99 columns

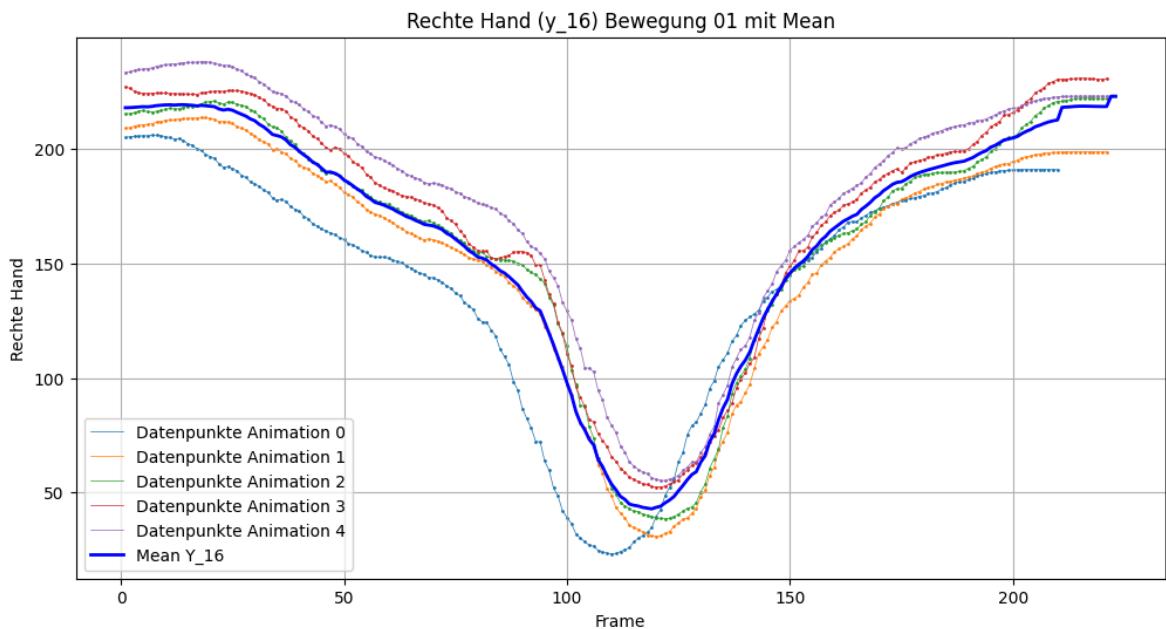


```
In [ ]: plt.figure(figsize=(12, 6))

for i, df in enumerate(dfsKomp01):
    plt.scatter(df["frame"], df["y_16"], marker='o', color=f'C{i}', s=1)
    plt.plot(df["frame"], df["y_16"], color=f'C{i}', linestyle='-', linewidth=2)

plt.plot(mean_values_01.index, mean_values_01["y_16"], color='b', linestyle='--', linewidth=2)

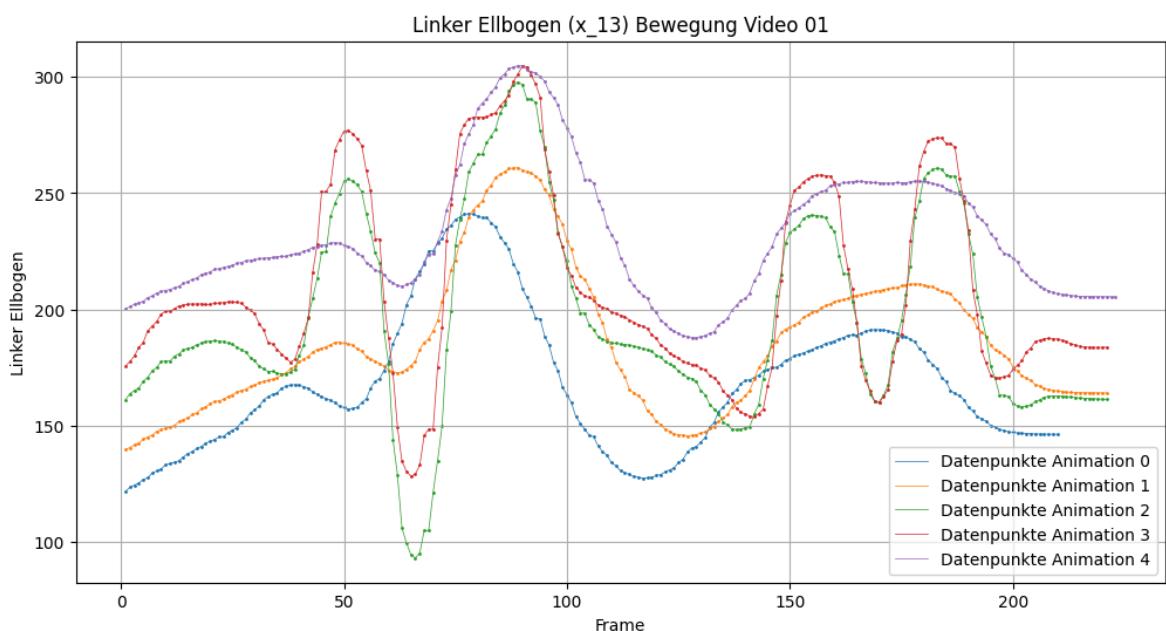
plt.title("Rechte Hand (y_16) Bewegung 01 mit Mean")
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()
```



## 2.4 Bewegungsanalyse linker Ellbogen (Kompensiert 01)

```
In [ ]: plt.figure(figsize=(12, 6))
for i, df in enumerate(dfsKomp01):
    plt.scatter(df["frame"], df["x_13"], marker='o', color=f'C{i}', s=1)
    plt.plot(df["frame"], df["x_13"], color=f'C{i}', linestyle='-', linewidth=2)

plt.title("Linker Ellbogen (x_13) Bewegung Video 01")
plt.xlabel("Frame")
plt.ylabel("Linker Ellbogen")
plt.grid(True)
plt.legend(loc='best')
plt.show()
```



## 2.5 Bewegungsanalyse linke Hand (Kompensiert 01)

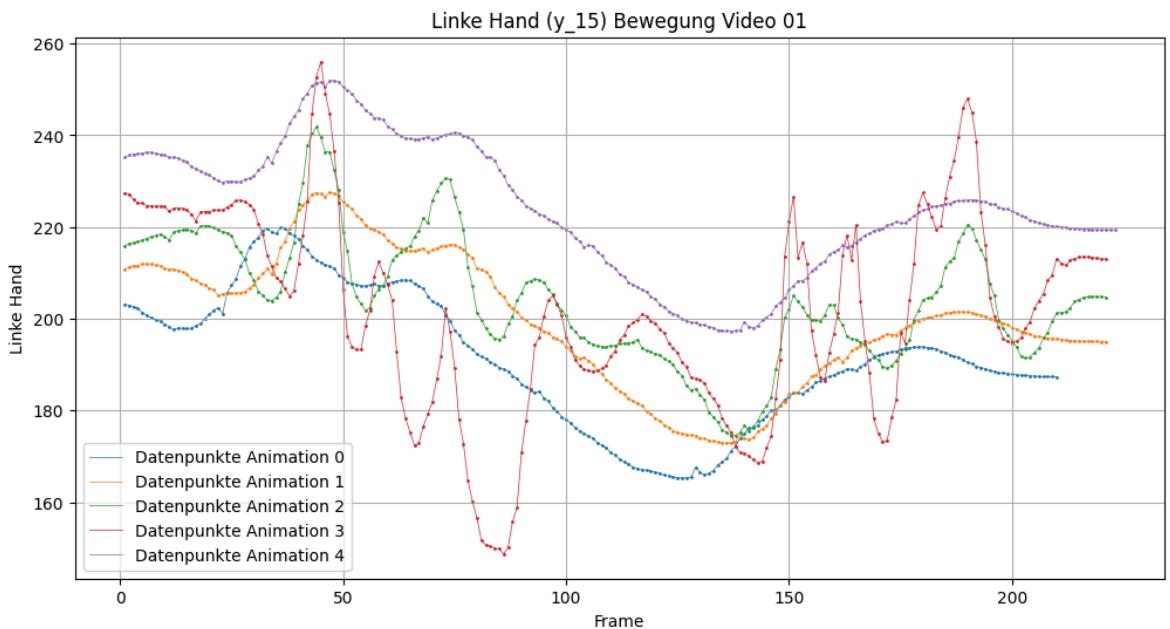
```
In [ ]: plt.figure(figsize=(12, 6))
for i, df in enumerate(dfsKomp01):
```

```

plt.scatter(df["frame"], df["y_15"], marker='o', color=f'C{i}', s=1)
plt.plot(df["frame"], df["y_15"], color=f'C{i}', linestyle='-', linewidth=2)

plt.title("Linke Hand (y_15) Bewegung Video 01")
plt.xlabel("Frame")
plt.ylabel("Linke Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()

```



### 3. DeepMotion Kompensiert 02

```
In [ ]: df_deepM02 = pd.read_csv('../Data/KeypointsBereinigtKompensiert_02.csv',
df_deepM02.head(5)
```

```
Out[ ]:
```

	compensation	frame	path	x_0	
0	1	1	/Users/salomekoller/Library/CloudStorage/OneDrive-Personal/DeepMotion/Keypoint...	0.000	-10.0
1	1	2	/Users/salomekoller/Library/CloudStorage/OneDrive-Personal/DeepMotion/Keypoint...	5.366	-10.0
2	1	3	/Users/salomekoller/Library/CloudStorage/OneDrive-Personal/DeepMotion/Keypoint...	8.136	-10.0
3	1	4	/Users/salomekoller/Library/CloudStorage/OneDrive-Personal/DeepMotion/Keypoint...	13.775	-10.0
4	1	5	/Users/salomekoller/Library/CloudStorage/OneDrive-Personal/DeepMotion/Keypoint...	15.967	-10.0

5 rows × 102 columns

#### 3.1 Analyse Form

```
In [ ]: print('Dimension:', df_deepM02.shape)
print('Number of rows:', df_deepM02.shape[0])
print('Number of columns:', df_deepM02.shape[1])
```

```
Dimension: (1715, 102)
Number of rows: 1715
Number of columns: 102
```

## 3.2 Splitting Frames

```
In [ ]: # Variablen initialisieren
frame_count = 0
frames_until_reset = []

# Iterieren über Dataframe, um Frames mit '1' zu finden
for index, row in df_deepM02.iterrows():
    frame_count += 1

    if row["frame"] == 1 and frame_count > 1:
        frames_until_reset.append(frame_count - 1)

if frame_count > 0:
    frames_until_reset.append(frame_count)

print("Number of frames until reset for each cycle:", frames_until_reset)
```

```
Number of frames until reset for each cycle: [337, 676, 1019, 1366, 1715]
```

```
In [ ]: # Einzelne Bewegungen werden in verschiedene Dataframes gepackt
dfsKomp02 = []
start = 0
for end in frames_until_reset:
    dfsKomp02.append(df_deepM02.iloc[start:end])
    start = end

print("Number of splits:", len(dfsKomp02))
```

```
Number of splits: 5
```

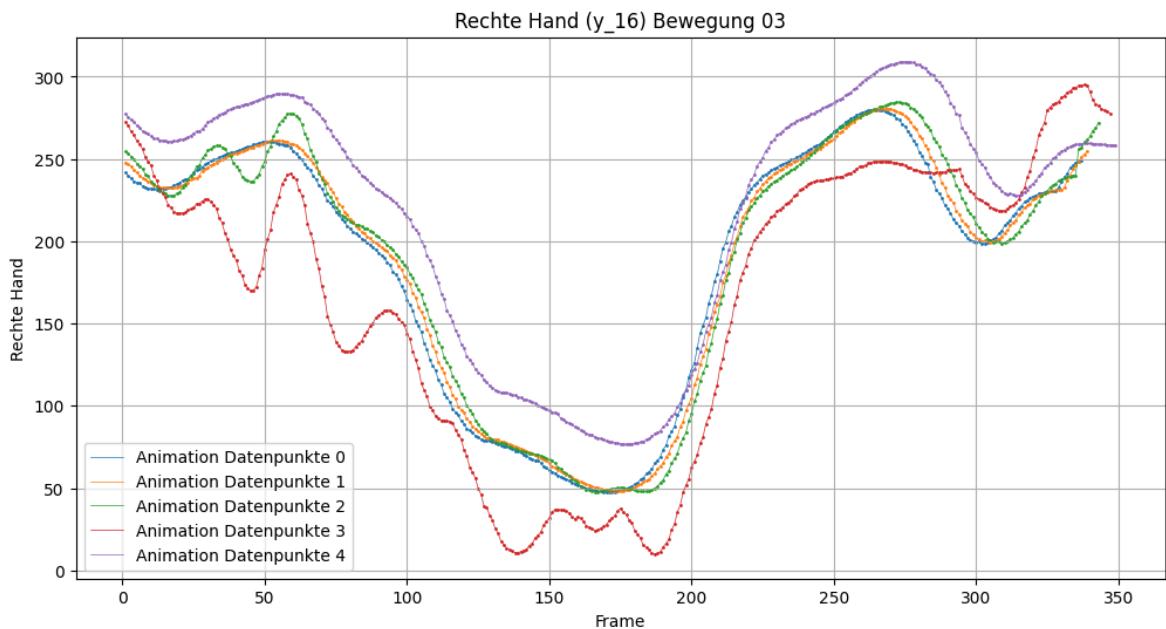
## 3.3 Bewegungsanalyse anhand rechter Handbewegung (Kompensiert 02)

```
In [ ]: plt.figure(figsize=(12, 6))

for i, df in enumerate(dfsKomp02):

    plt.scatter(df["frame"], df["y_16"], marker='o', color=f'C{i}', s=1)
    plt.plot(df["frame"], df["y_16"], color=f'C{i}', linestyle='-', linewidth=2)

    plt.title("Rechte Hand (y_16) Bewegung 03")
    plt.xlabel("Frame")
    plt.ylabel("Rechte Hand")
    plt.grid(True)
    plt.legend(loc='best')
    plt.show()
```



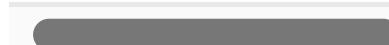
### 3.3.1 Berechnung des Mean

```
In [ ]: means02 = []
mean_values_02 = df_deepM02[['frame', 'x_0', 'y_0', 'z_0', 'x_1', 'y_1',
 'x_4', 'y_4', 'z_4', 'x_5', 'y_5', 'z_5', 'x_6', 'y_6',
 'x_8', 'y_8', 'z_8', 'x_9', 'y_9', 'z_9', 'x_10', 'y_10',
 'x_12', 'y_12', 'z_12', 'x_13', 'y_13', 'z_13', 'x_14',
 'x_16', 'y_16', 'z_16', 'x_17', 'y_17', 'z_17', 'x_18',
 'x_20', 'y_20', 'z_20', 'x_21', 'y_21', 'z_21', 'x_22',
 'x_24', 'y_24', 'z_24', 'x_25', 'y_25', 'z_25', 'x_26',
 'x_28', 'y_28', 'z_28', 'x_29', 'y_29', 'z_29', 'x_30',
 'x_32', 'y_32', 'z_32']].groupby('frame').mean()
means02.append(mean_values_02)

mean_values_02.head(5)
```

```
Out[ ]:      x_0      y_0      z_0      x_1      y_1      z_1      x_
frame
1 -80.50884  8.62712 -102.583294 -1045.738 -307.2164 -53.49067 -1045.7
2 -77.81076  8.94612 -98.155338 -1045.738 -307.2164 -53.49067 -1045.7
3 -75.21866  9.22386 -93.563912 -1045.738 -307.2164 -53.49067 -1045.7
4 -72.21568  9.81578 -88.551916 -1045.738 -307.2164 -53.49067 -1045.7
5 -69.32478 10.38222 -82.843140 -1045.738 -307.2164 -53.49067 -1045.7
```

5 rows × 99 columns



### 3.3.2 Bewegungsmuster mit Mean

```
In [ ]: plt.figure(figsize=(12, 6))

for i, df in enumerate(dfsKomp02):
```

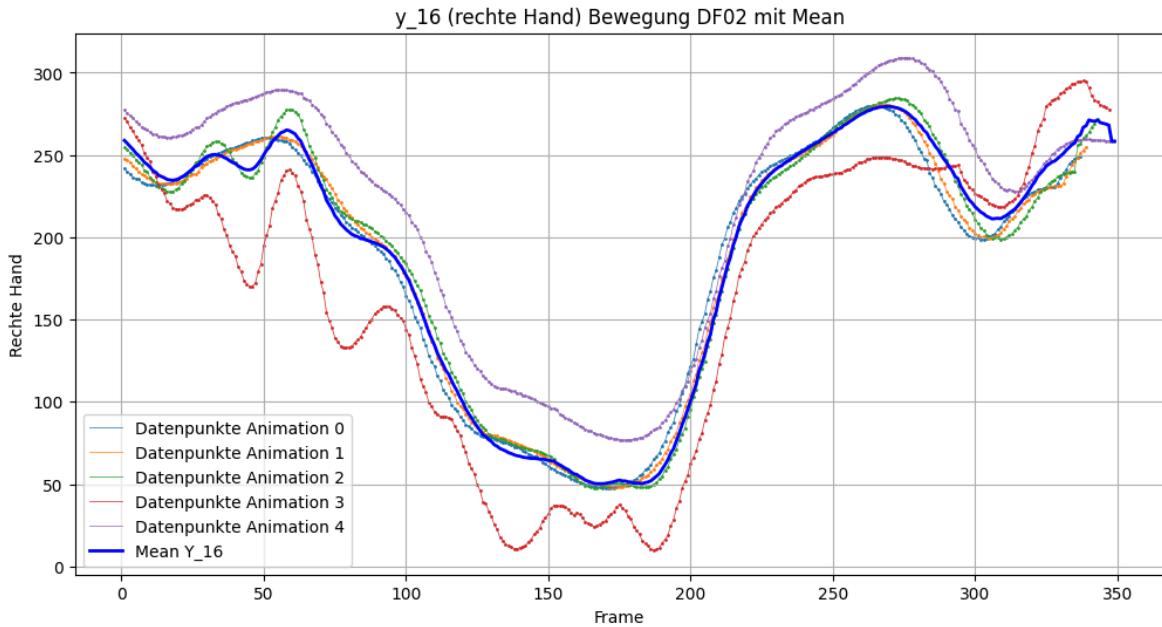
```

    plt.scatter(df["frame"], df["y_16"], marker='o', color=f'C{i}', s=1)
    plt.plot(df["frame"], df["y_16"], color=f'C{i}', linestyle='-', linewidth=2)

plt.plot(mean_values_02.index, mean_values_02["y_16"], color='b', linestyle='solid', linewidth=2)

plt.title("y_16 (rechte Hand) Bewegung DF02 mit Mean" )
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()

```



## 4. MediaPipe

```
In [ ]: df_MediaP = pd.read_csv('../Data/samples.csv', sep=',', encoding='utf-8')
df_MediaP.head(5)
```

Out[ ]:

		path	frame	compensation	x_0	
0	bachelorarbeiten/nils/data/videos/samples/drin...		1	False	0.00000	0.0
1	bachelorarbeiten/nils/data/videos/samples/drin...		2	False	-0.08766	0.
2	bachelorarbeiten/nils/data/videos/samples/drin...		3	False	-0.81886	0.9
3	bachelorarbeiten/nils/data/videos/samples/drin...		4	False	-1.46955	1.4
4	bachelorarbeiten/nils/data/videos/samples/drin...		5	False	-2.07000	1.4

5 rows × 102 columns



### 4.1 Analyse form

```
In [ ]: print('Dimension:', df_MediaP.shape)
print('Number of rows:', df_MediaP.shape[0])
print('Number of columns:', df_MediaP.shape[1])
```

```
Dimension: (623, 102)
Number of rows: 623
Number of columns: 102
```

## 4.2 Splitting Frames

```
In [ ]: # Variablen initialisieren
frame_count = 0
frames_until_reset = []

# Iterieren über Dataframe, um Frames mit '1' zu finden
for index, row in df_MediaP.iterrows():
    frame_count += 1

    if row["frame"] == 1 and frame_count > 1:
        frames_until_reset.append(frame_count - 1)

if frame_count > 0:
    frames_until_reset.append(frame_count)

print("Number of frames until reset for each cycle:", frames_until_reset)
```

```
Number of frames until reset for each cycle: [172, 471, 623]
```

```
In [ ]: # DataFrame entsprechend splitten
df_MediaP1 = df_MediaP.iloc[:frames_until_reset[0]]
df_MediaP2 = df_MediaP.iloc[frames_until_reset[0]:frames_until_reset[1]]
df_MediaP3 = df_MediaP.iloc[frames_until_reset[1]:]
```

```
In [ ]: df_MediaP1.head(5)
```

```
Out[ ]:
```

	path	frame	compensation	x_0	x_1
0	bachelorarbeiten/nils/data/videos/samples/drin...	1	False	0.00000	0.0
1	bachelorarbeiten/nils/data/videos/samples/drin...	2	False	-0.08766	0.
2	bachelorarbeiten/nils/data/videos/samples/drin...	3	False	-0.81886	0.9
3	bachelorarbeiten/nils/data/videos/samples/drin...	4	False	-1.46955	1.4
4	bachelorarbeiten/nils/data/videos/samples/drin...	5	False	-2.07000	1.4

```
5 rows × 102 columns
```

### 4.3.1 Vergleich Länge Mediapipe Frame zu DeepMotion Frame 01

```
In [ ]: print('Dimension MediaPipe02:', df_MediaP1.shape)

for i, df in enumerate(dfsKomp01):
    print(f"DataFrame {i+1} Shape:", df.shape)
```

```
Dimension MediaPipe02: (172, 102)
DataFrame 1 Shape: (210, 102)
DataFrame 2 Shape: (221, 102)
DataFrame 3 Shape: (221, 102)
DataFrame 4 Shape: (221, 102)
DataFrame 5 Shape: (223, 102)
```

```
In [ ]: df_MediaP2.head(5)
```

Out[ ]:

		path	frame	compensation	x_0
172	bachelorarbeiten/nils/data/videos/samples/03_t...		1	False	0.00000
173	bachelorarbeiten/nils/data/videos/samples/03_t...		2	False	0.13657
174	bachelorarbeiten/nils/data/videos/samples/03_t...		3	False	0.19873
175	bachelorarbeiten/nils/data/videos/samples/03_t...		4	False	1.98133
176	bachelorarbeiten/nils/data/videos/samples/03_t...		5	False	3.43781

5 rows × 102 columns

### 4.3.2 Vergleich Länge Mediapipe Frame zu DeepMotion Frame 02

```
In [ ]: print('Dimension MediaPipe02:', df_MediaP2.shape)

for i, df in enumerate(dfsKomp02):
    print(f"DataFrame {i+1} Shape:", df.shape)
```

```
Dimension MediaPipe02: (299, 102)
DataFrame 1 Shape: (337, 102)
DataFrame 2 Shape: (339, 102)
DataFrame 3 Shape: (343, 102)
DataFrame 4 Shape: (347, 102)
DataFrame 5 Shape: (349, 102)
```

```
In [ ]: df_MediaP3.head(5)
```

Out[ ]:

		path	frame	compensation	x_0
471	bachelorarbeiten/nils/data/videos/samples/drin...		1	False	0.0000
472	bachelorarbeiten/nils/data/videos/samples/drin...		2	False	-0.6283
473	bachelorarbeiten/nils/data/videos/samples/drin...		3	False	-0.7909
474	bachelorarbeiten/nils/data/videos/samples/drin...		4	False	-1.8517
475	bachelorarbeiten/nils/data/videos/samples/drin...		5	False	-1.7632

5 rows × 102 columns

### 4.3.3 Vergleich Länge Mediapipe Frame zu DeepMotion Frame 03

```
In [ ]: print('Dimension MediaPipe03:', df_MediaP3.shape)

for i, df in enumerate(dfs):
    print(f"DataFrame {i+1} Shape:", df.shape)
```

```
Dimension MediaPipe03: (152, 102)
DataFrame 1 Shape: (182, 102)
DataFrame 2 Shape: (191, 102)
DataFrame 3 Shape: (192, 102)
DataFrame 4 Shape: (175, 102)
DataFrame 5 Shape: (197, 102)
```

### 4.4 Rechter Arm (z\_16) Bewegung Mediapipe

```
In [ ]: plt.figure(figsize=(12, 6))

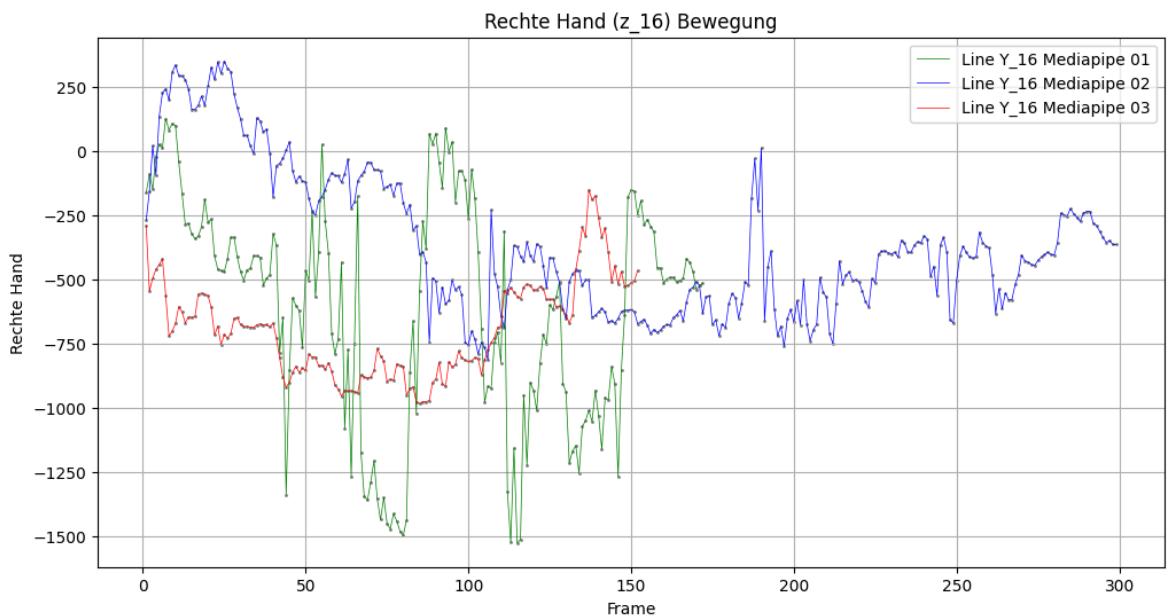
plt.scatter(df_MediaP1["frame"], df_MediaP1["z_16"], marker='o', color='green')
plt.plot(df_MediaP1["frame"], df_MediaP1["z_16"], color='green', linestyle='solid')

plt.scatter(df_MediaP2["frame"], df_MediaP2["z_16"], marker='o', color='blue')
plt.plot(df_MediaP2["frame"], df_MediaP2["z_16"], color='blue', linestyle='solid')

plt.scatter(df_MediaP3["frame"], df_MediaP3["z_16"], marker='o', color='red')
plt.plot(df_MediaP3["frame"], df_MediaP3["z_16"], color='red', linestyle='solid')

plt.title("Rechte Hand (z_16) Bewegung")
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')

plt.show()
```



### 4.5 Kopfbewegung (z\_0) Mediapipe

```
In [ ]: plt.figure(figsize=(12, 6))

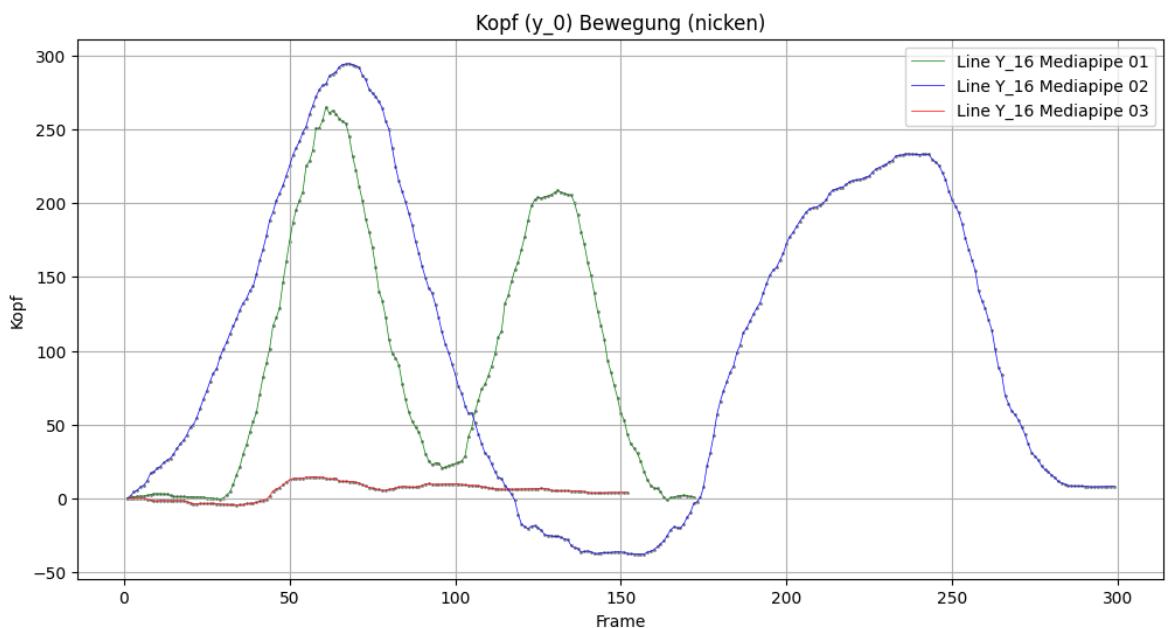
plt.scatter(df_MediaP1["frame"], df_MediaP1["y_0"], marker='o', color='green')
plt.plot(df_MediaP1["frame"], df_MediaP1["y_0"], color='green', linestyle='solid')

plt.scatter(df_MediaP2["frame"], df_MediaP2["y_0"], marker='o', color='blue')
plt.plot(df_MediaP2["frame"], df_MediaP2["y_0"], color='blue', linestyle='solid')

plt.scatter(df_MediaP3["frame"], df_MediaP3["y_0"], marker='o', color='red')
plt.plot(df_MediaP3["frame"], df_MediaP3["y_0"], color='red', linestyle='solid')

plt.title("Kopf (y_0) Bewegung (nicken)")
plt.xlabel("Frame")
plt.ylabel("Kopf")
plt.grid(True)
# Move the legend outside of the plot
plt.legend(loc='best')

plt.show()
```



## 4.6 Bewegungsanalyse anhand rechter Handbewegung

```
In [ ]: plt.figure(figsize=(12, 6))

plt.scatter(df_MediaP1["frame"], df_MediaP1["y_16"], marker='o', color='green')
plt.plot(df_MediaP1["frame"], df_MediaP1["y_16"], color='green', linestyle='solid')

plt.scatter(df_MediaP2["frame"], df_MediaP2["y_16"], marker='o', color='blue')
plt.plot(df_MediaP2["frame"], df_MediaP2["y_16"], color='blue', linestyle='solid')

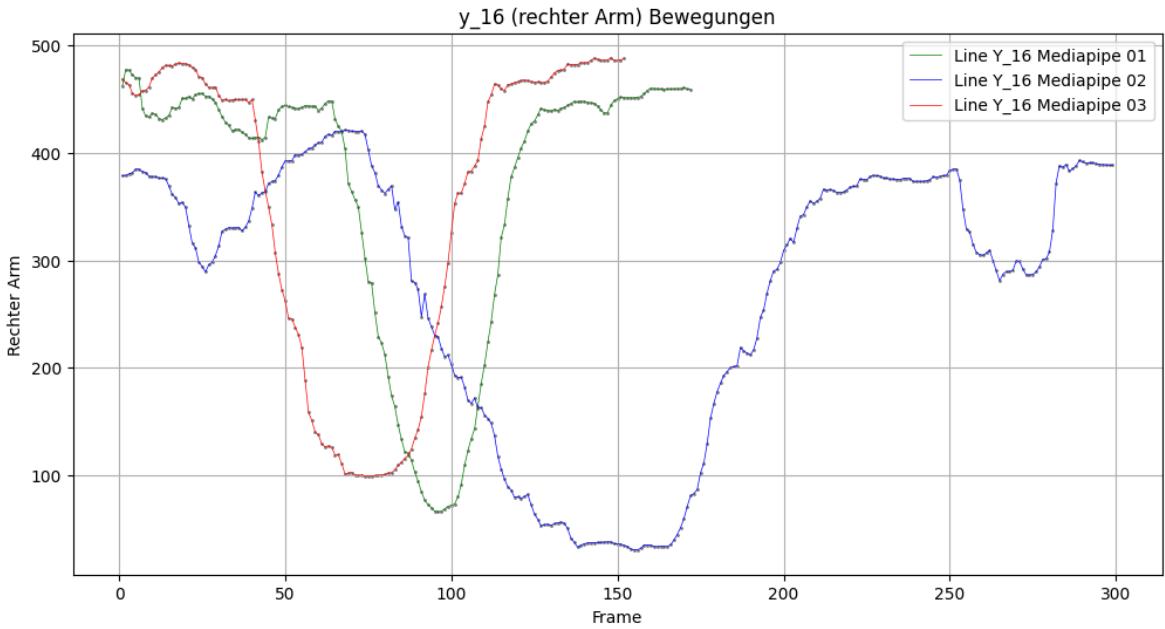
plt.scatter(df_MediaP3["frame"], df_MediaP3["y_16"], marker='o', color='red')
plt.plot(df_MediaP3["frame"], df_MediaP3["y_16"], color='red', linestyle='solid')

plt.title("y_16 (rechter Arm) Bewegungen")
plt.xlabel("Frame")
plt.ylabel("Rechter Arm")
```

```

plt.grid(True)
plt.legend(loc='best')
plt.show()

```



```

In [ ]: meansMP = []
df_MediaP12 = pd.concat([df_MediaP1, df_MediaP2])

mean_values_MP = df_MediaP12[['frame', 'x_0', 'y_0', 'z_0', 'x_1', 'y_1',
                               'x_4', 'y_4', 'z_4', 'x_5', 'y_5', 'z_5', 'x_6', 'y_6',
                               'x_8', 'y_8', 'z_8', 'x_9', 'y_9', 'z_9', 'x_10', 'y_10',
                               'x_12', 'y_12', 'z_12', 'x_13', 'y_13', 'z_13', 'x_14',
                               'x_16', 'y_16', 'z_16', 'x_17', 'y_17', 'z_17', 'x_18',
                               'x_20', 'y_20', 'z_20', 'x_21', 'y_21', 'z_21', 'x_22',
                               'x_24', 'y_24', 'z_24', 'x_25', 'y_25', 'z_25', 'x_26',
                               'x_28', 'y_28', 'z_28', 'x_29', 'y_29', 'z_29', 'x_30',
                               'x_32', 'y_32', 'z_32']].groupby('frame').mean()
meansMP.append(mean_values_MP)

mean_values_MP.head(5)

```

```

Out[ ]:      x_0      y_0      z_0      x_1      y_1      z_1
frame
1  0.000000  0.000000  0.000000  19.160885 -26.792770  61.691245  29.9
2  0.024455  1.189490  31.137870  19.269535 -26.128870  99.299155  29.92
3 -0.310065  2.638355  55.155305  18.940485 -24.881945 122.820275  29.4
4  0.255890  3.420960 150.906590  19.777185 -24.170795 216.327370  30.18
5  0.683905  4.665790 185.844140  20.488755 -22.907115 249.370338  30.9

```

5 rows × 99 columns

```

In [ ]: plt.figure(figsize=(12, 6))

plt.scatter(df_MediaP1["frame"], df_MediaP1["y_16"], marker='o', color='g'

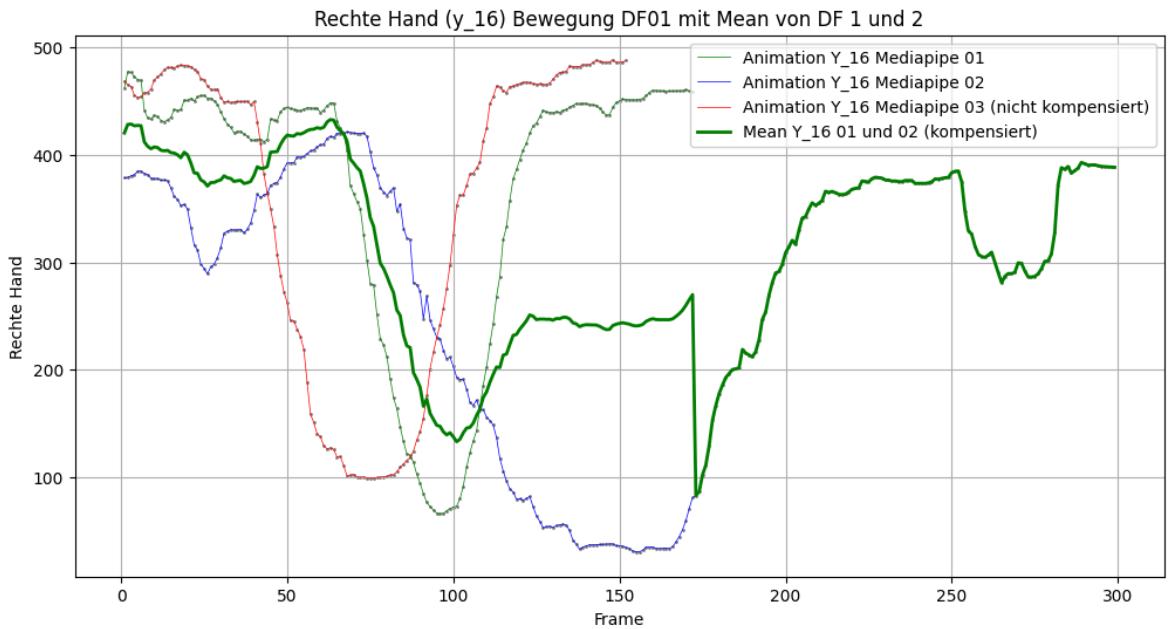
```

```

plt.plot(df_MediaP1["frame"], df_MediaP1["y_16"], color='green', linestyle='solid')
plt.scatter(df_MediaP2["frame"], df_MediaP2["y_16"], marker='o', color='green', linestyle='dashed')
plt.plot(df_MediaP2["frame"], df_MediaP2["y_16"], color='blue', linestyle='solid')
plt.scatter(df_MediaP3["frame"], df_MediaP3["y_16"], marker='o', color='red', linestyle='dashed')
plt.plot(df_MediaP3["frame"], df_MediaP3["y_16"], color='red', linestyle='solid')

plt.plot(mean_values_MP.index, mean_values_MP["y_16"], color='green', linestyle='solid')
plt.title("Rechte Hand (y_16) Bewegung DF01 mit Mean von DF 1 und 2")
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()

```



## Berechnung Euclidean Distances

### 1. Analyse der Mean Werte

```

In [ ]: plt.figure(figsize=(12, 6))

plt.plot(mean_values_01.index, mean_values_01["y_16"], color="#054907", linestyle='solid')
plt.plot(df_MediaP1["frame"], df_MediaP1["y_16"], color='green', linestyle='solid')
plt.plot(mean_values_02.index, mean_values_02["y_16"], color='darkblue', linestyle='solid')
plt.plot(df_MediaP2["frame"], df_MediaP2["y_16"], color='blue', linestyle='solid')
plt.plot(mean_values_03.index, mean_values_03["y_16"], color='darkred', linestyle='solid')
plt.plot(df_MediaP3["frame"], df_MediaP3["y_16"], color='red', linestyle='solid')

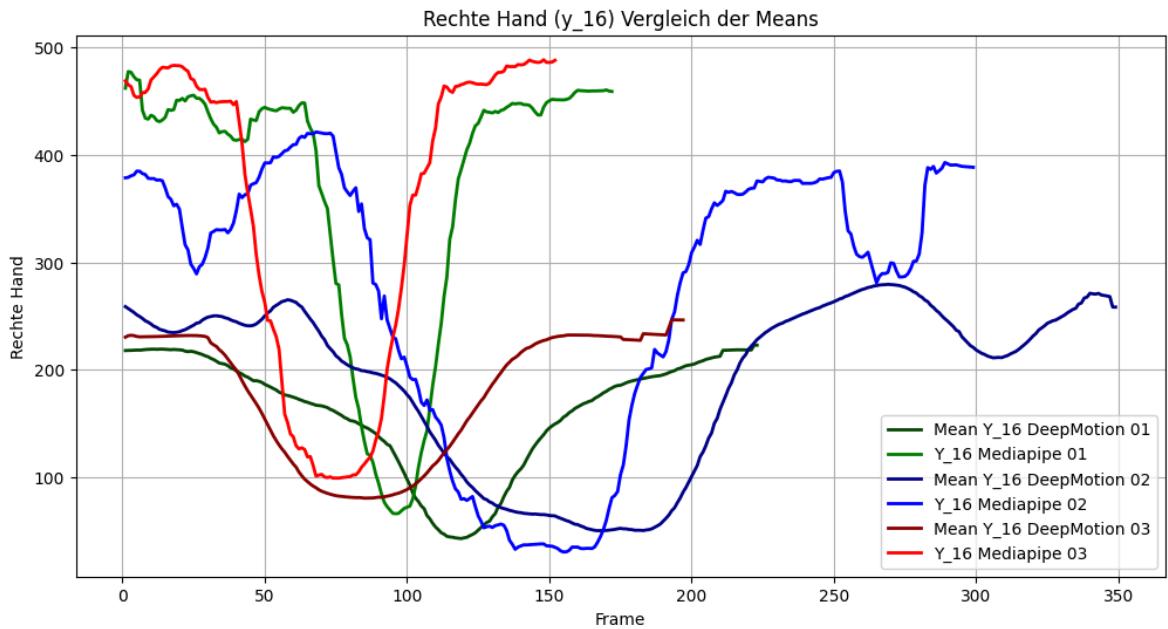
plt.title("Rechte Hand (y_16) Vergleich der Means")
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)

```

```

plt.legend(loc='best')
plt.show()

```



```

In [ ]: difference_in_mean_01 = df_MediaP1['y_16'].mean() - mean_values_01['y_16']
difference_in_mean_02 = df_MediaP2['y_16'].mean() - mean_values_02['y_16']
difference_in_mean_03 = df_MediaP3['y_16'].mean() - mean_values_03['y_16']

difference_in_mean_01_per = difference_in_mean_01 / df_MediaP1['y_16'].mean()
difference_in_mean_02_per = difference_in_mean_02 / df_MediaP2['y_16'].mean()
difference_in_mean_03_per = difference_in_mean_03 / df_MediaP3['y_16'].mean()

print(f"Mean Value in DeepMotion 01: {mean_values_01['y_16'].mean()}")
print(f"Mean Value in MediaPipe 01: {df_MediaP1['y_16'].mean()}")
print(f"Difference in Mean: {difference_in_mean_01}")
print(f"Difference in Mean in Percentage: {difference_in_mean_01_per}")
print("")

print(f"Mean Value in DeepMotion 02: {mean_values_02['y_16'].mean()}")
print(f"Mean Value in MediaPipe 02: {df_MediaP2['y_16'].mean()}")
print(f"Difference in Mean: {difference_in_mean_02}")
print(f"Difference in Mean in Percentage: {difference_in_mean_02_per}")
print("")

print(f"Mean Value in DeepMotion 03: {mean_values_03['y_16'].mean()}")
print(f"Mean Value in MediaPipe 03: {df_MediaP3['y_16'].mean()}")
print(f"Difference in Mean: {difference_in_mean_03}")
print(f"Difference in Mean in Percentage: {difference_in_mean_03_per}")

```

Mean Value in DeepMotion 01: 162.18050183856505  
 Mean Value in MediaPipe 01: 367.5381621511628  
 Difference in Mean: 205.35766031259774  
 Difference in Mean in Percentage: 55.87383337573999

Mean Value in DeepMotion 02: 191.1468976599809  
 Mean Value in MediaPipe 02: 271.1325231438127  
 Difference in Mean: 79.9856254838318  
 Difference in Mean in Percentage: 29.500564726204477

Mean Value in DeepMotion 03: 176.7090342639594  
 Mean Value in MediaPipe 03: 351.3075072368421  
 Difference in Mean: 174.59847297288272  
 Difference in Mean in Percentage: 49.6996134088228

## 2. Vergleich Werte MediaPipe und DeepMotion

Werte von 1 Original MediaPipe Frame werden zum dazugehörigen Animationsclip DeepMotion verglichen

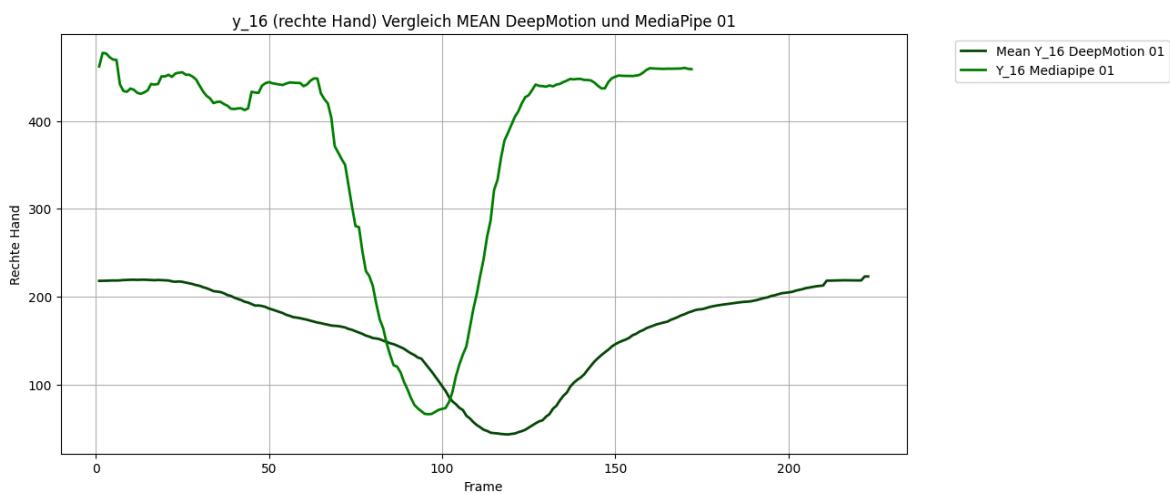
Es wird der Y-Wert der rechten Hand (y\_16) analysiert.

### 2.1 Kompensiert 01

```
In [ ]: plt.figure(figsize=(12, 6))

plt.plot(mean_values_01.index, mean_values_01["y_16"], color="#054907", l
plt.plot(df_MediaP1["frame"], df_MediaP1["y_16"], color='green', linestyle='solid'

plt.title("y_16 (rechte Hand) Vergleich MEAN DeepMotion und MediaPipe 01")
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



```
In [ ]: df_deepM01.head(5)
```

```
Out[ ]:   compensation  frame          path      x_0
0            1    1 /Users/salomekoller/Library/CloudStorage/OneDr...  0.000 -100
1            1    2 /Users/salomekoller/Library/CloudStorage/OneDr...  1.909 -100
2            1    3 /Users/salomekoller/Library/CloudStorage/OneDr...  2.832 -100
3            1    4 /Users/salomekoller/Library/CloudStorage/OneDr...  3.748 -100
4            1    5 /Users/salomekoller/Library/CloudStorage/OneDr...  5.316 -100
```

5 rows × 102 columns

#### 2.1.1 Euclidean Distances

```
In [ ]: # Definition von Kolonnen
columns = ['frame', 'x_0', 'y_0', 'z_0', 'x_11', 'y_11', 'z_11',
           'x_12', 'y_12', 'z_12', 'x_13', 'y_13', 'z_13', 'x_14', 'y_14',
           'x_16', 'y_16', 'z_16', 'x_17', 'y_17', 'z_17', 'x_18', 'y_18',
           'x_20', 'y_20', 'z_20', 'x_21', 'y_21', 'z_21', 'x_22', 'y_22',
           'x_24', 'y_24', 'z_24', 'x_25', 'y_25', 'z_25', 'x_26', 'y_26',
           'x_28', 'y_28', 'z_28', 'x_31', 'y_31', 'z_31', 'x_32', 'y_32']

# Kalkulieren von euclidean Distance
euclidean_distances_01 = {}

for frame in df_MediaP1["frame"]:
    euclidean_distances_01[frame] = []
    for col in columns[1:]:
        point_1 = (frame, mean_values_01[col].loc[frame])
        point_2 = (frame, df_MediaP1[col][df_MediaP1['frame'] == frame].v
        euclidean_distance_01 = distance.euclidean(point_1, point_2)
        euclidean_distances_01[frame].append(euclidean_distance_01)

euclidean_df_01 = pd.DataFrame.from_dict(euclidean_distances_01, orient='
euclidean_df_01.head(5)
```

Out[ ]:

	x_0	y_0	z_0	x_11	y_11	z_11	x_12
1	42.92380	16.61496	200.431752	69.3176	90.13566	758.638164	168.83520
2	44.95106	15.95476	263.983176	69.9716	89.87085	831.607521	171.16664
3	47.12266	15.15391	347.277852	70.6990	91.24217	862.063893	173.41177
4	49.02955	14.35974	555.240142	69.6152	91.58547	1001.308958	175.43172
5	51.61540	14.09434	658.537104	70.5362	92.12765	1103.323308	177.55982

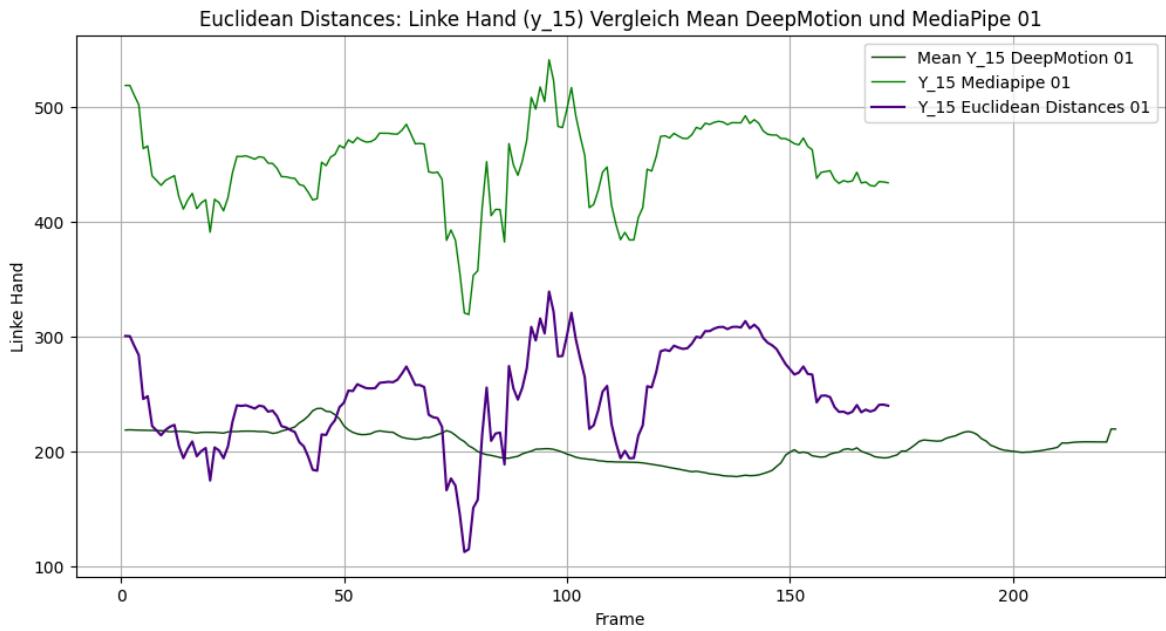
5 rows × 63 columns

```
In [ ]: plt.figure(figsize=(12, 6))

plt.plot(mean_values_01.index, mean_values_01["y_16"], color="#054907", l
plt.plot(df_MediaP1["frame"], df_MediaP1["y_16"], color='green', linestyle='

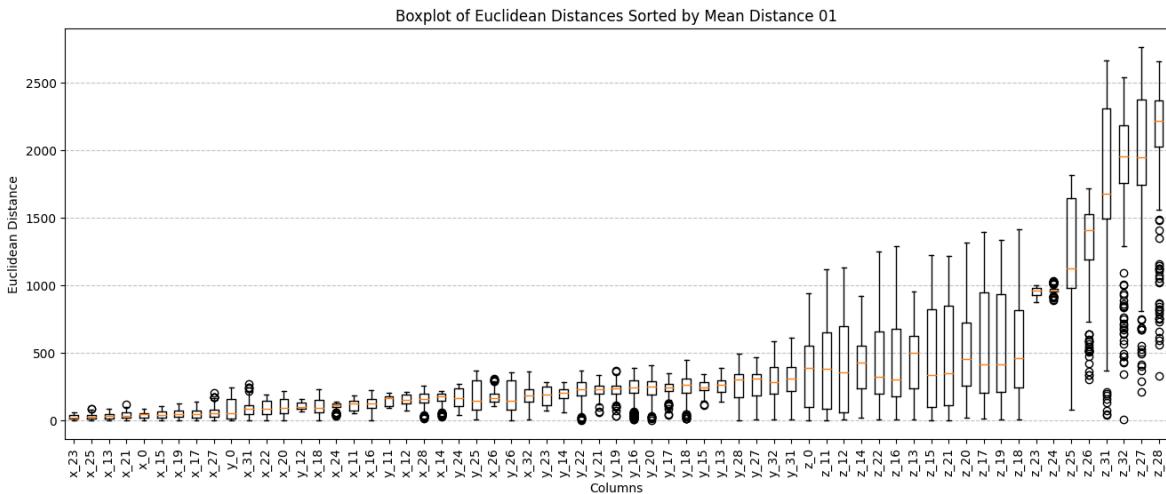
plt.plot(euclidean_df_01.index, euclidean_df_01["y_16"], color='indigo', l

plt.title("Euclidean Distances: Rechte Hand (y_15) Vergleich Mean DeepMot
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()
```



```
In [ ]: mean_distances = euclidean_df_01.mean()
sorted_df = euclidean_df_01[mean_distances.sort_values().index]

plt.figure(figsize=(16, 6))
plt.boxplot(sorted_df.values)
plt.xticks(range(1, len(sorted_df.columns) + 1), sorted_df.columns, rotation=90)
plt.title('Boxplot of Euclidean Distances Sorted by Mean Distance 01')
plt.xlabel('Columns')
plt.ylabel('Euclidean Distance')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



### 2.1.1.1 Ohne Z Werte

```
In [ ]: # Definition von Kolonnen
selected_columns = ['x_0', 'x_11', 'x_12', 'x_13', 'x_14', 'x_15', 'x_16',
                    'x_18', 'x_19', 'x_20', 'x_21', 'x_22', 'x_23', 'x_24',
                    'y_11', 'y_12', 'y_13', 'y_14', 'y_15', 'y_16', 'y_17',
                    'y_18', 'y_19', 'y_20', 'y_21', 'y_22', 'y_23', 'y_24']

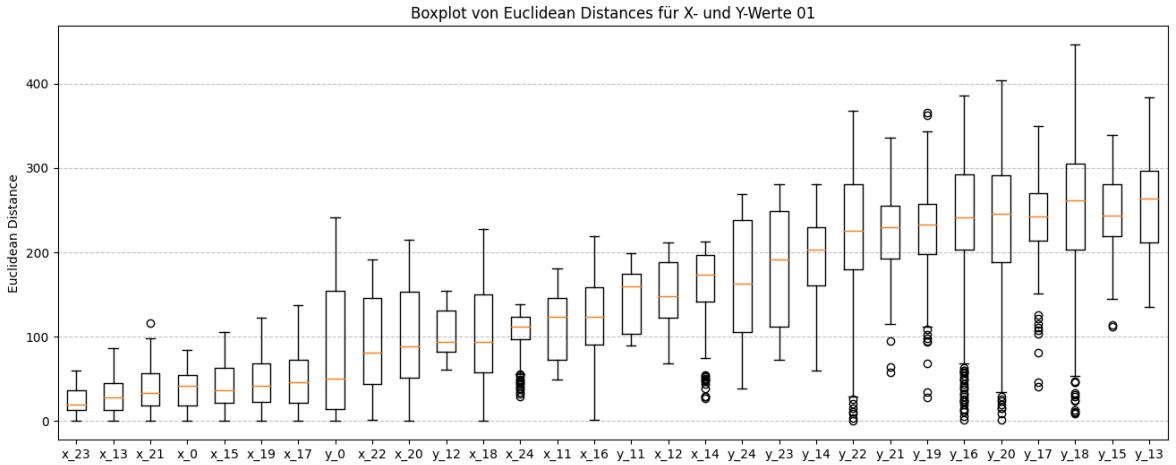
mean_distances = euclidean_df_01[selected_columns].mean()
sorted_df = euclidean_df_01[mean_distances.sort_values().index]

plt.figure(figsize=(16, 6))
```

```

plt.boxplot(sorted_df.values)
plt.xticks(range(1, len(sorted_df.columns) + 1), sorted_df.columns)
plt.title('Boxplot von Euclidean Distances für X- und Y-Werte 01')
plt.ylabel('Euclidean Distance')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

```



## 2.2 Kompensiert 02

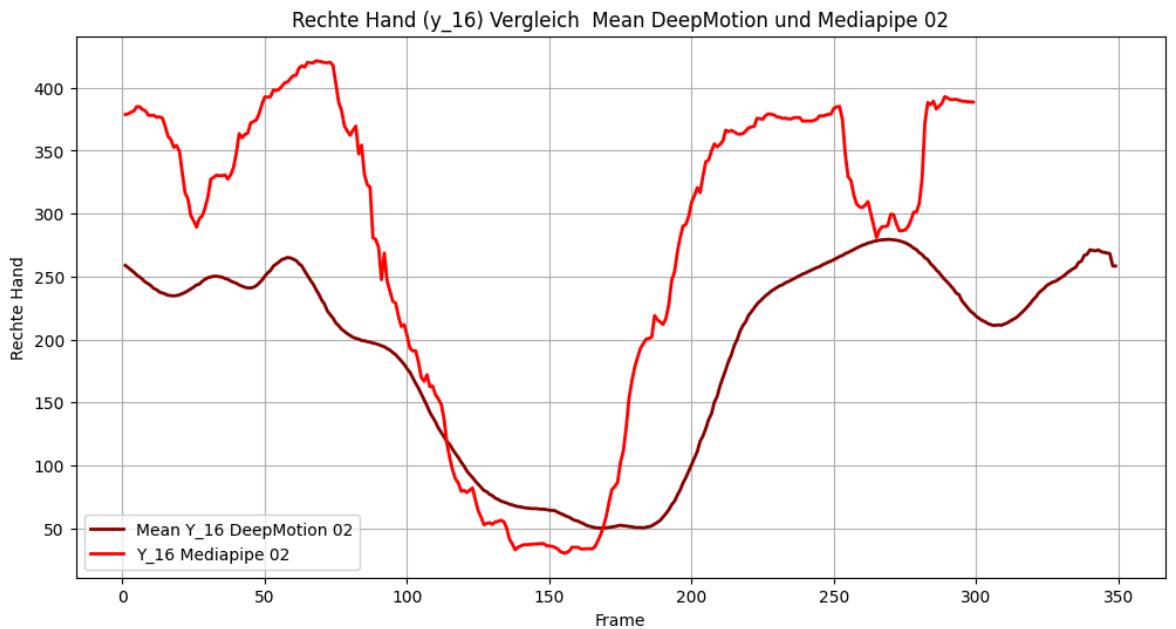
```

In [ ]: plt.figure(figsize=(12, 6))

plt.plot(mean_values_02.index, mean_values_02["y_16"], color='darkred', l
plt.plot(df_MediaP2["frame"], df_MediaP2["y_16"], color='red', linestyle='

plt.title("Rechte Hand (y_16) Vergleich Mean DeepMotion und Mediapipe 02")
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()

```



### 2.2.1 Euclidean Distances

```
In [ ]: # Definition von Kolonnen
columns = ['frame', 'x_0', 'y_0', 'z_0', 'x_11', 'y_11', 'z_11',
           'x_12', 'y_12', 'z_12', 'x_13', 'y_13', 'z_13', 'x_14', 'y_14',
           'x_16', 'y_16', 'z_16', 'x_17', 'y_17', 'z_17', 'x_18', 'y_18',
           'x_20', 'y_20', 'z_20', 'x_21', 'y_21', 'z_21', 'x_22', 'y_22',
           'x_24', 'y_24', 'z_24', 'x_25', 'y_25', 'z_25', 'x_26', 'y_26',
           'x_28', 'y_28', 'z_28', 'x_31', 'y_31', 'z_31', 'x_32', 'y_32']

# Kalkulieren von euclidean Distance
euclidean_distances_02 = {}

for frame in df_MediaP2["frame"]:
    euclidean_distances_02[frame] = []
    for col in columns[1:]:
        point_1 = (frame, mean_values_02[col].loc[frame])
        point_2 = (frame, df_MediaP2[col][df_MediaP2['frame'] == frame].v
        euclidean_distance_02 = distance.euclidean(point_1, point_2)
        euclidean_distances_02[frame].append(euclidean_distance_02)

euclidean_df_02 = pd.DataFrame.from_dict(euclidean_distances_02, orient='e
euclidean_df_02.head(5)
```

Out[ ]:

	x_0	y_0	z_0	x_11	y_11	z_11	x_12
1	80.50884	8.62712	102.583294	204.19991	75.73850	612.477154	33.43345
2	77.94733	7.08524	98.897278	202.00413	75.67390	621.639808	34.80006
3	75.41739	4.93818	60.819862	199.83341	74.17957	609.463960	36.16155
4	74.19701	4.39192	41.274056	197.32713	72.01682	591.316945	37.25993
5	72.76259	2.55040	4.149380	195.28057	69.82888	569.920596	37.51685

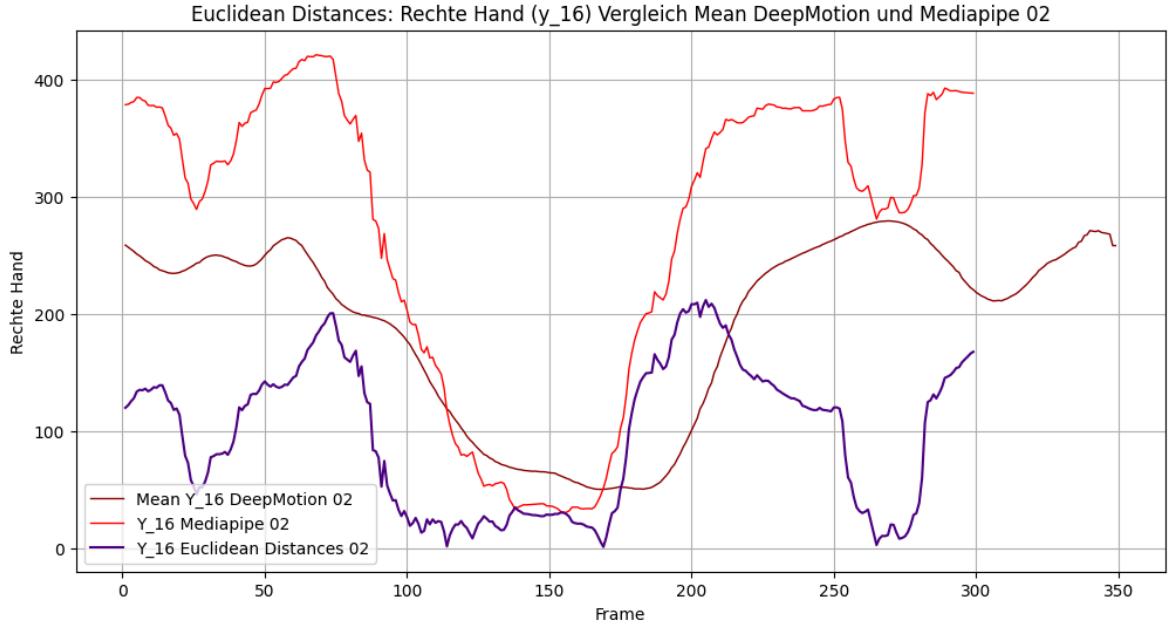
5 rows × 63 columns

```
In [ ]: plt.figure(figsize=(12, 6))

plt.plot(mean_values_02.index, mean_values_02["y_16"], color='darkred', l
plt.plot(df_MediaP2["frame"], df_MediaP2["y_16"], color='red', linestyle='

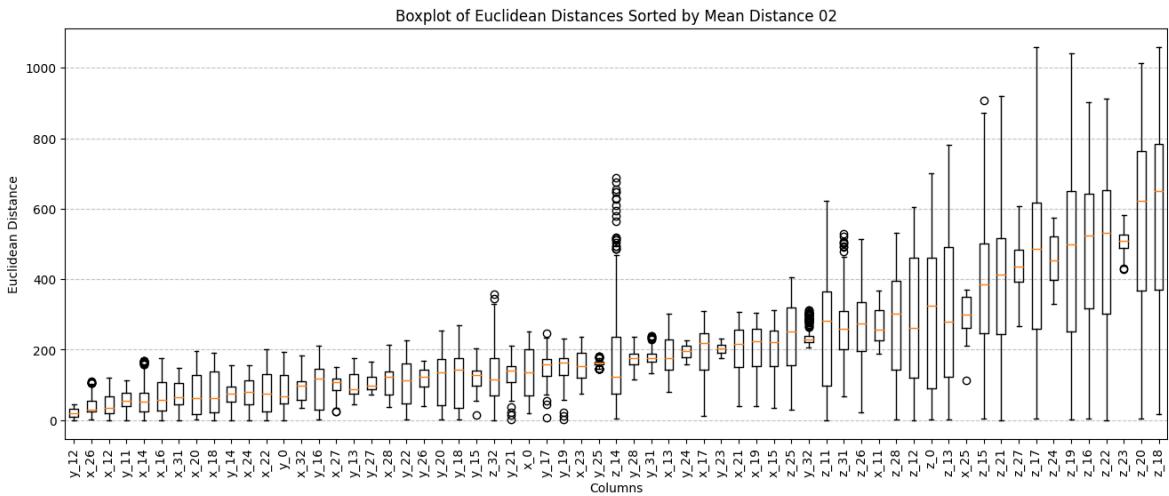
plt.plot(euclidean_df_02.index, euclidean_df_02["y_16"], color='indigo', 

plt.title("Euclidean Distances: Rechte Hand (y_16) Vergleich Mean DeepMot
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()
```



```
In [ ]: mean_distances = euclidean_df_02.mean()
sorted_df = euclidean_df_02[mean_distances.sort_values().index]

plt.figure(figsize=(16, 6))
plt.boxplot(sorted_df.values)
plt.xticks(range(1, len(sorted_df.columns) + 1), sorted_df.columns, rotation=90)
plt.title('Boxplot of Euclidean Distances Sorted by Mean Distance 02')
plt.xlabel('Columns')
plt.ylabel('Euclidean Distance')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



### 2.2.1.1 Ohne Z Werte

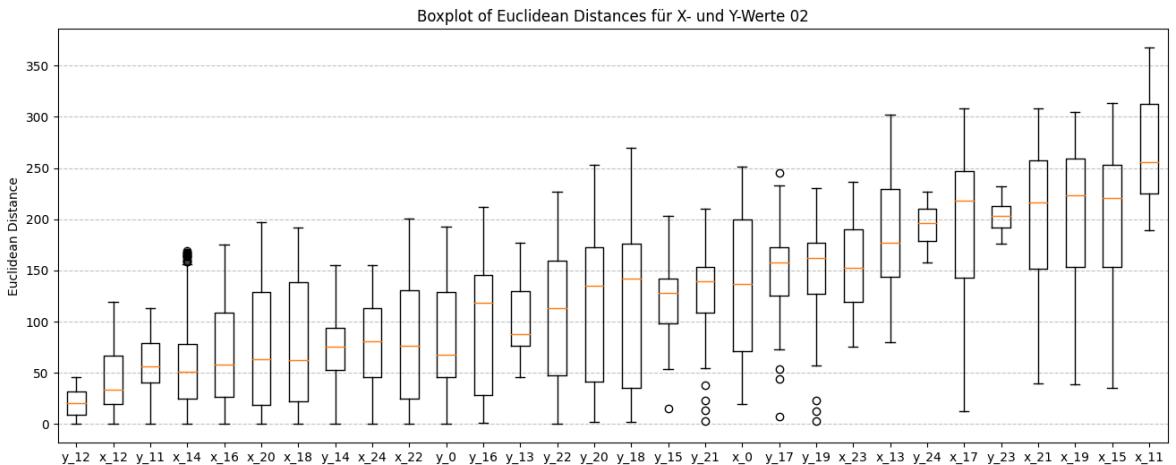
```
In [ ]: # Definition von Kolonnen
selected_columns = ['x_0', 'x_11', 'x_12', 'x_13',
                    'x_14', 'x_15', 'x_16', 'x_17',
                    'x_18', 'x_19', 'x_20', 'x_21',
                    'x_22', 'x_23', 'x_24', 'y_0',
                    'y_11', 'y_12', 'y_13',
                    'y_14', 'y_15', 'y_16', 'y_17',
                    'y_18', 'y_19', 'y_20', 'y_21',
                    'y_22', 'y_23', 'y_24']
```

```

mean_distances = euclidean_df_02[selected_columns].mean()
sorted_df = euclidean_df_02[mean_distances.sort_values().index]

plt.figure(figsize=(16, 6))
plt.boxplot(sorted_df.values)
plt.xticks(range(1, len(sorted_df.columns) + 1), sorted_df.columns)
plt.title('Boxplot of Euclidean Distances für X- und Y-Werte 02')
plt.ylabel('Euclidean Distance')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

```



## 2.3 Nicht Kompensiert 03

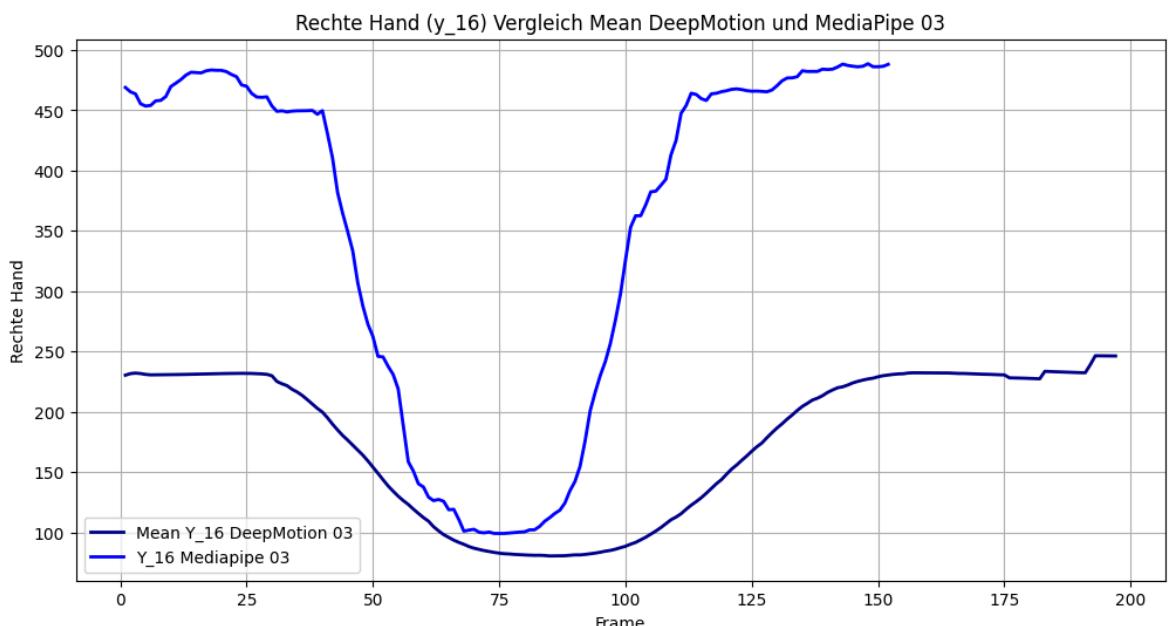
```

In [ ]: plt.figure(figsize=(12, 6))

plt.plot(mean_values_03.index, mean_values_03["y_16"], color='darkblue',
         plt.plot(df_MediaP3["frame"], df_MediaP3["y_16"], color='blue', linestyle='--')

plt.title("Rechte Hand (y_16) Vergleich Mean DeepMotion und MediaPipe 03")
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()

```



### 2.3.1 Euclidean Distances

```
In [ ]: # Definition von Kolumnen
columns = ['frame', 'x_0', 'y_0', 'z_0', 'x_11', 'y_11', 'z_11',
           'x_12', 'y_12', 'z_12', 'x_13', 'y_13', 'z_13', 'x_14', 'y_14',
           'x_16', 'y_16', 'z_16', 'x_17', 'y_17', 'z_17', 'x_18', 'y_18',
           'x_20', 'y_20', 'z_20', 'x_21', 'y_21', 'z_21', 'x_22', 'y_22',
           'x_24', 'y_24', 'z_24', 'x_25', 'y_25', 'z_25', 'x_26', 'y_26',
           'x_28', 'y_28', 'z_28', 'x_31', 'y_31', 'z_31', 'x_32', 'y_32']

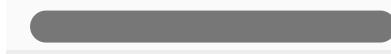
# Kalkulieren von euclidean Distance
euclidean_distances_03 = {}

for frame in df_MediaP3["frame"]:
    euclidean_distances_03[frame] = []
    for col in columns[1:]:
        point_1 = (frame, mean_values_03[col].loc[frame])
        point_2 = (frame, df_MediaP3[col][df_MediaP3['frame'] == frame].v
        euclidean_distance_03 = distance.euclidean(point_1, point_2)
        euclidean_distances_03[frame].append(euclidean_distance_03)

euclidean_df_03 = pd.DataFrame.from_dict(euclidean_distances_03, orient='
euclidean_df_03.head(5)
```

```
Out[ ]:      x_0      y_0      z_0      x_11      y_11      z_11      x_12
1  65.99356  18.27286  171.467506  49.6439  99.93474  668.763334  205.30202  7
2  67.48476  19.79484  130.008146  48.9014  95.24961  431.186842  206.08323  79
3  68.77944  20.10476  121.484736  48.5285  93.40679  434.953544  207.17242  78
4  70.54364  19.92736     2.222716  49.3687  93.34960  490.740326  208.43588  80
5  71.39090  19.31311  24.813634  49.5948  93.57797  519.740428  209.44843  83
```

5 rows × 63 columns

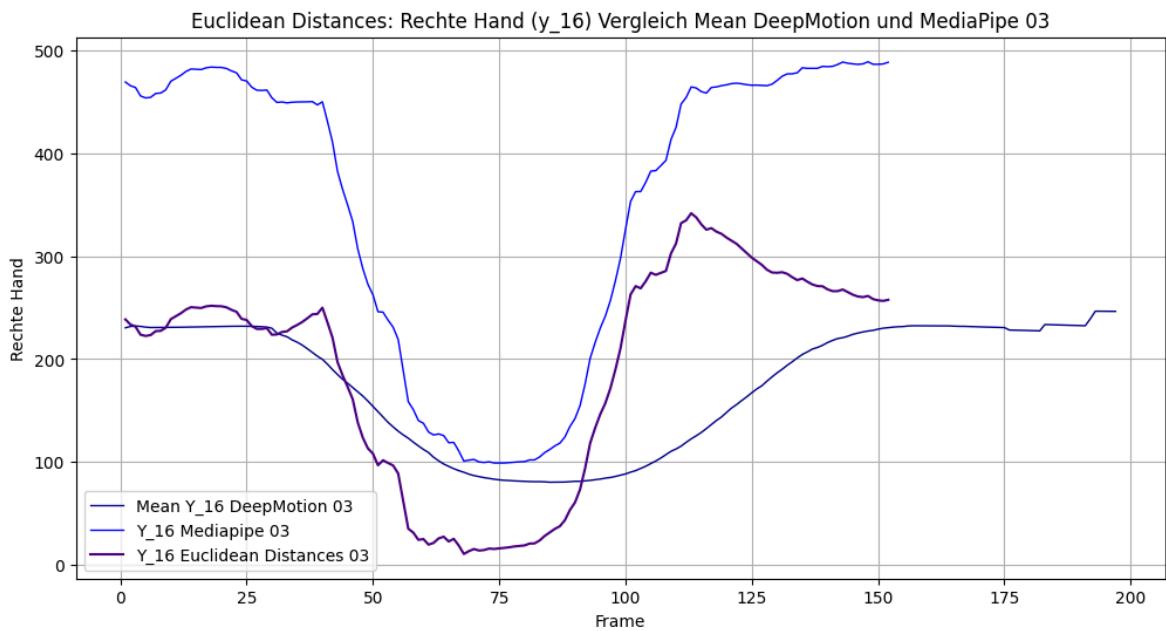


```
In [ ]: plt.figure(figsize=(12, 6))

plt.plot(mean_values_03.index, mean_values_03["y_16"], color='darkblue',
plt.plot(df_MediaP3["frame"], df_MediaP3["y_16"], color='blue', linestyle

plt.plot(euclidean_df_03.index, euclidean_df_03["y_16"], color='indigo',

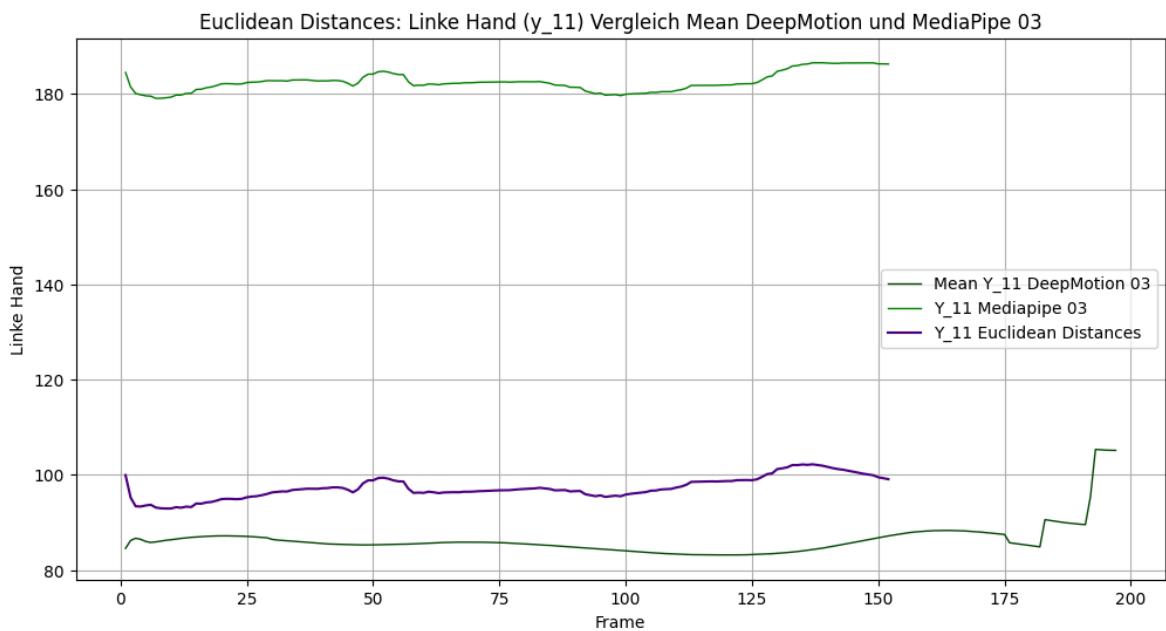
plt.title("Euclidean Distances: Rechte Hand (y_16) Vergleich Mean DeepMot
plt.xlabel("Frame")
plt.ylabel("Rechte Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()
```



```
In [ ]: plt.figure(figsize=(12, 6))

plt.plot(mean_values_03.index, mean_values_03["y_11"], color="#054907", l
plt.plot(df_MediaP3["frame"], df_MediaP3["y_11"], color='green', linestyle='dashed'
plt.plot(euclidean_df_03.index, euclidean_df_03["y_11"], color='indigo', l

plt.title("Euclidean Distances: Linke Hand (y_11) Vergleich Mean DeepMotio
plt.xlabel("Frame")
plt.ylabel("Linke Hand")
plt.grid(True)
plt.legend(loc='best')
plt.show()
```



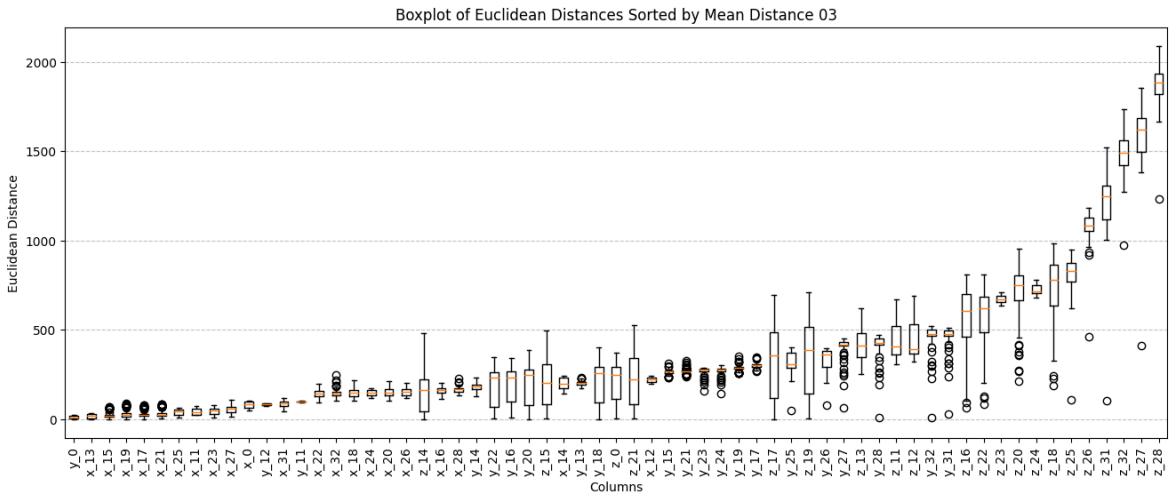
```
In [ ]: mean_distances = euclidean_df_03.mean()
sorted_df = euclidean_df_03[mean_distances.sort_values().index]

plt.figure(figsize=(16, 6))
plt.boxplot(sorted_df.values)
plt.xticks(range(1, len(sorted_df.columns) + 1), sorted_df.columns, rotat
```

```

plt.title('Boxplot of Euclidean Distances Sorted by Mean Distance 03')
plt.xlabel('Columns')
plt.ylabel('Euclidean Distance')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

```



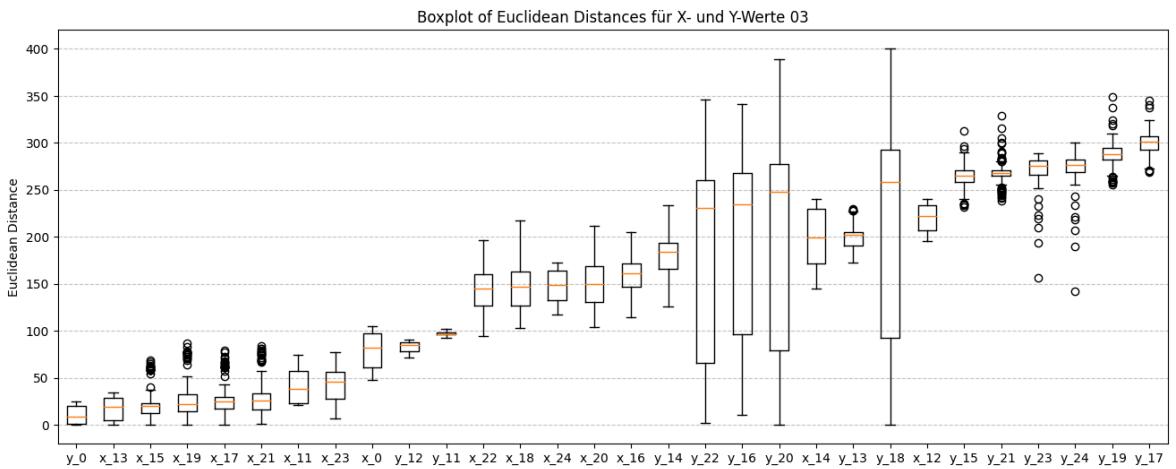
### 2.3.1.1 Ohne Z Werte

```

In [ ]: # Definition von Kolonnen
selected_columns = ['x_0', 'x_11', 'x_12', 'x_13',
                    'x_14', 'x_15', 'x_16', 'x_17',
                    'x_18', 'x_19', 'x_20', 'x_21',
                    'x_22', 'x_23', 'x_24', 'y_0',
                    'y_11', 'y_12', 'y_13',
                    'y_14', 'y_15', 'y_16', 'y_17',
                    'y_18', 'y_19', 'y_20', 'y_21',
                    'y_22', 'y_23', 'y_24']
mean_distances = euclidean_df_03[selected_columns].mean()
sorted_df = euclidean_df_03[mean_distances.sort_values().index]

plt.figure(figsize=(16, 6))
plt.boxplot(sorted_df.values)
plt.xticks(range(1, len(sorted_df.columns) + 1), sorted_df.columns)
plt.title('Boxplot of Euclidean Distances für X- und Y-Werte 03')
plt.ylabel('Euclidean Distance')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

```

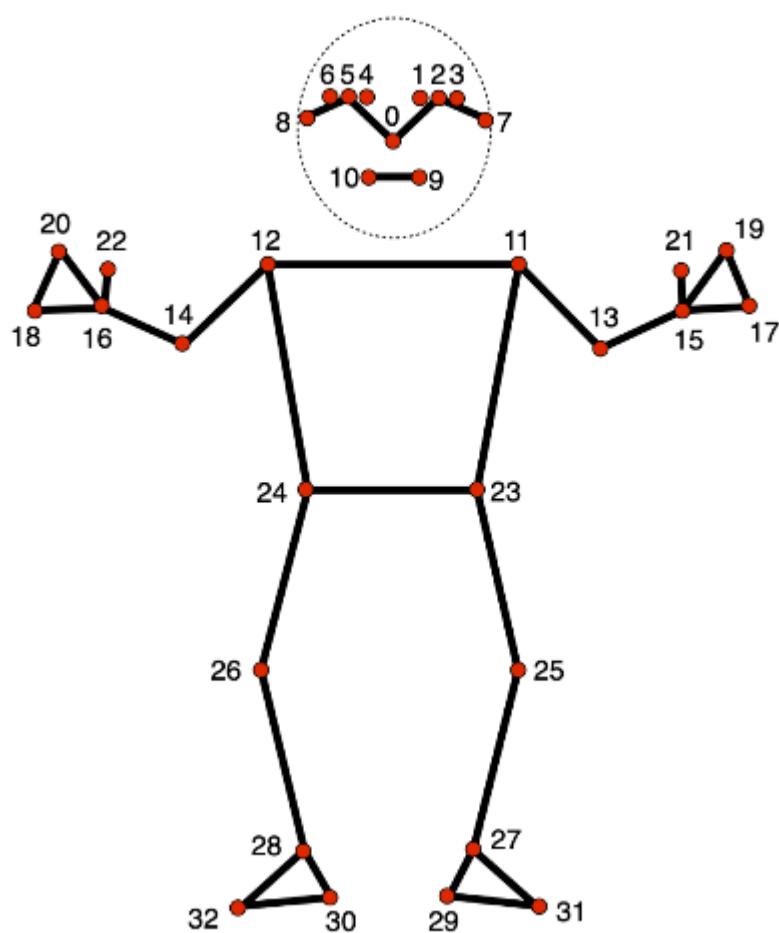


## Erkenntnisse

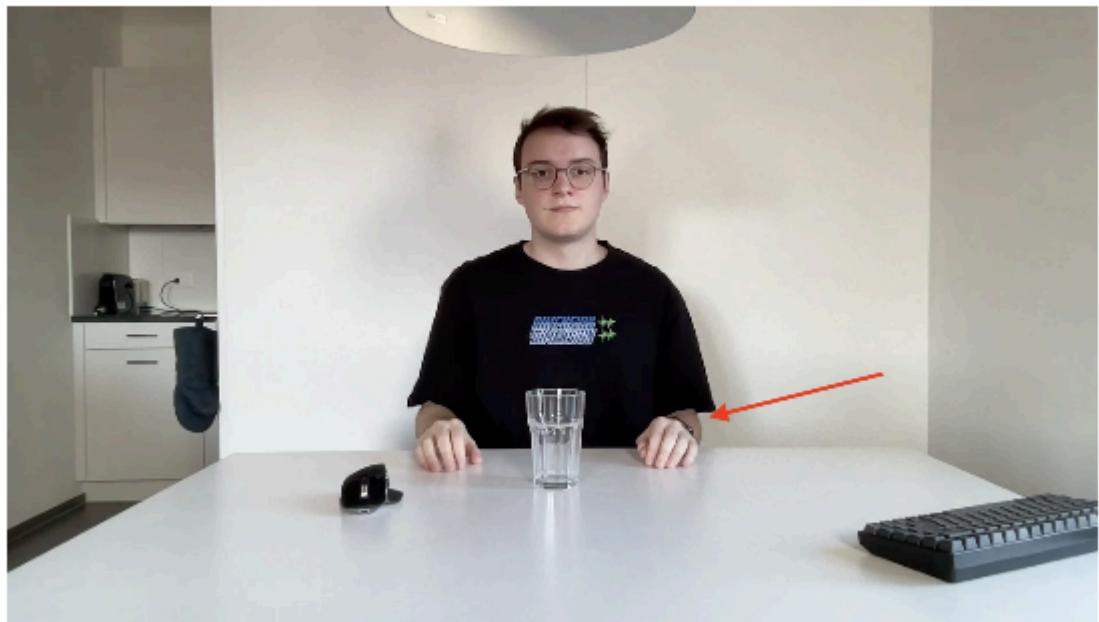
- Trinkarm ist viel ungenauer, als der ruhende Arm
- Videoaufnahmen: Arm ist wegen Shirt nicht ersichtlich
- Videoaufnahmen: Beine in Sitzposition können irritieren durch Tisch
- Z Werte können nicht verwendet werden für Analyse Performance
- Nicht Kompensierte Bewegung hat viel kleinere Ranges von Differenzen weil Bewegung sehr klar ist

## 3. Vergleich ganzer Oberkörper

```
In [ ]: img = plt.imread('Pictures/landmark.png')
im = OffsetImage(img, zoom=0.2)
ab = AnnotationBbox(im, (0.5, 0.5), frameon=False)
plt.gca().add_artist(ab)
plt.axis('off')
plt.show()
```

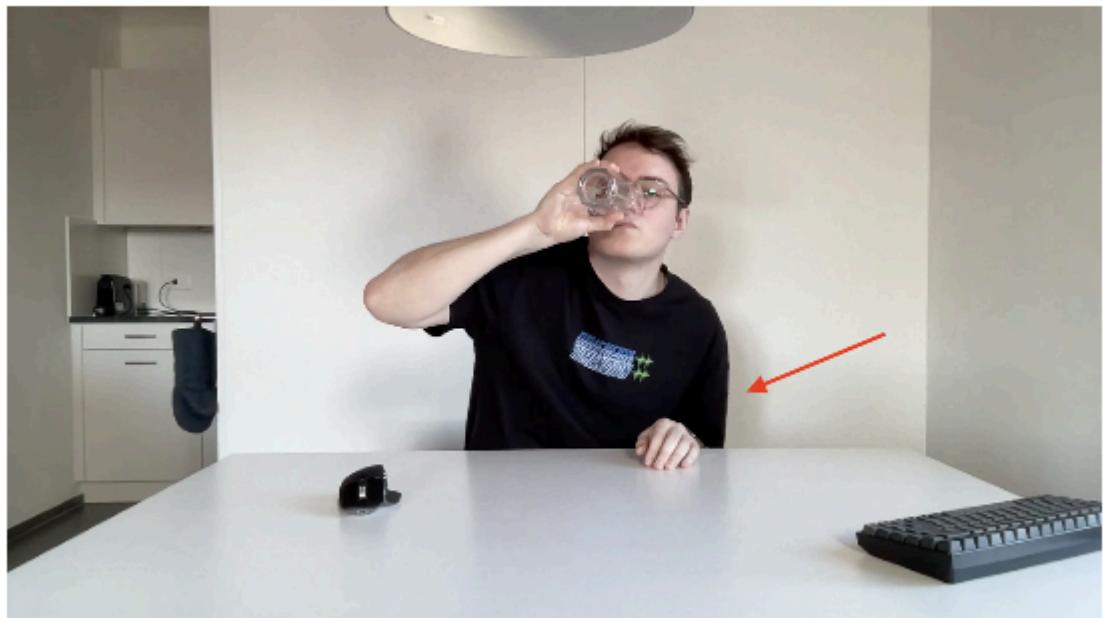


```
In [ ]: img = plt.imread('Pictures/ausgangsposition.jpg')
im = OffsetImage(img, zoom=0.2)
ab = AnnotationBbox(im, (0.5, 0.5), frameon=False)
plt.gca().add_artist(ab)
plt.axis('off')
plt.show()
```



```
In [ ]: img = plt.imread('Pictures/bewegung.jpg')
im = OffsetImage(img, zoom=0.2)
ab = AnnotationBbox(im, (0.5, 0.5), frameon=False)
plt.gca().add_artist(ab)

plt.axis('off')
plt.show()
```



```
In [ ]: img = plt.imread('Pictures/komp_1.jpg')
im = OffsetImage(img, zoom=0.4)
ab = AnnotationBbox(im, (0.5, 0.5), frameon=False)
plt.gca().add_artist(ab)
```

```
plt.axis('off')
plt.show()
```



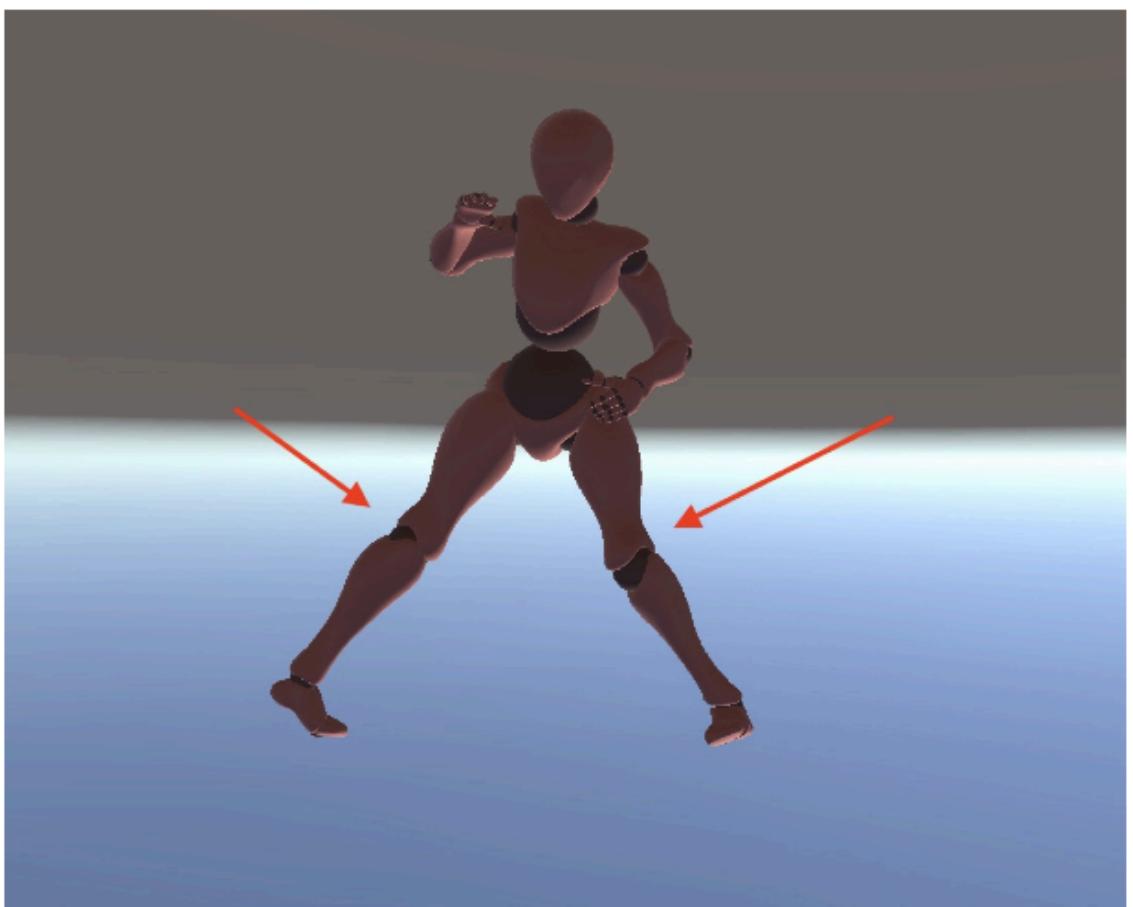
```
In [ ]: img = plt.imread('Pictures/komp2.jpg')
im = OffsetImage(img, zoom=0.2)
ab = AnnotationBbox(im, (0.5, 0.5), frameon=False)
plt.gca().add_artist(ab)

plt.axis('off')
plt.show()
```



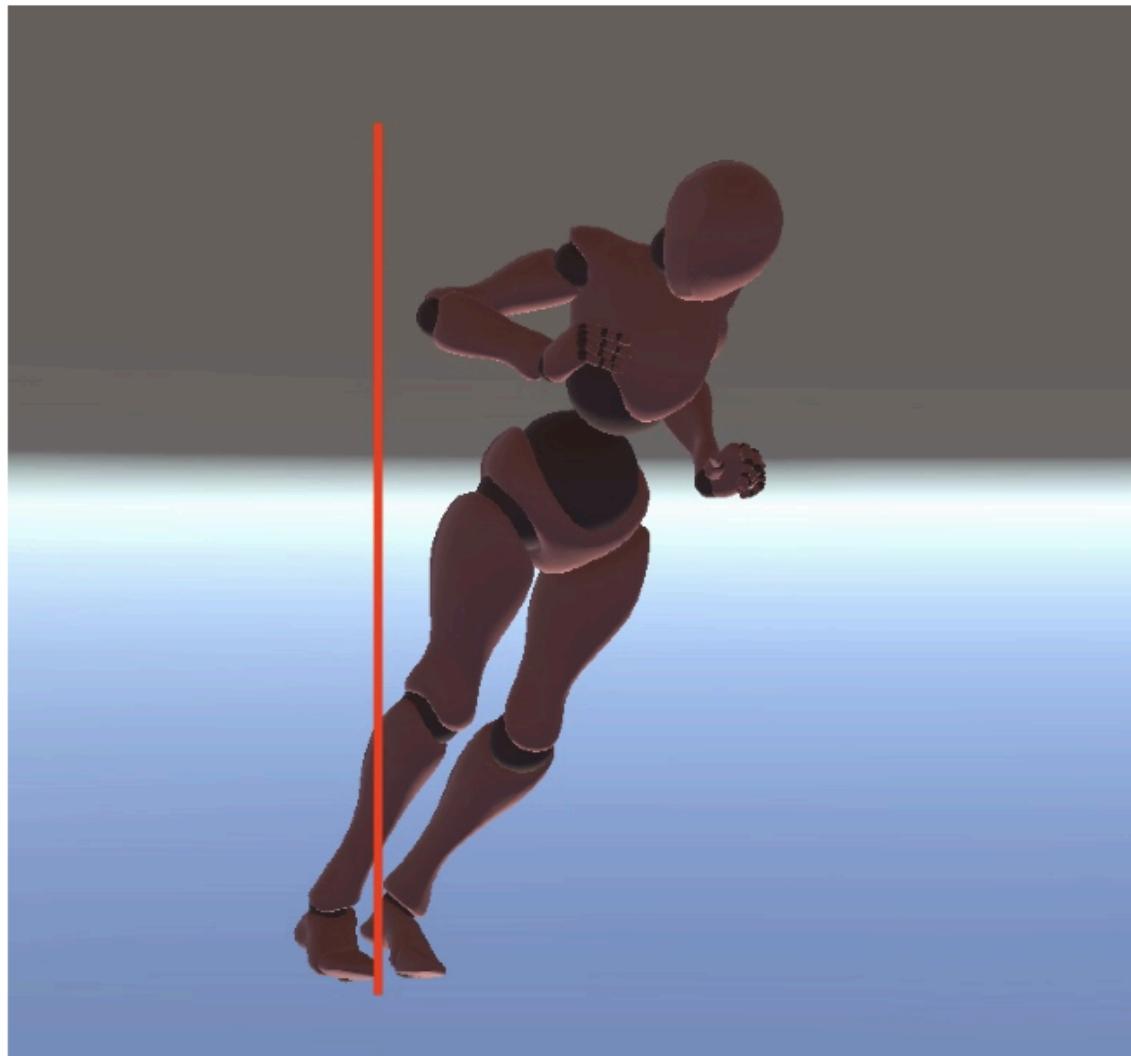
```
In [ ]: img = plt.imread('Pictures/x_2.jpg')
im = OffsetImage(img, zoom=0.4)
ab = AnnotationBbox(im, (0.5, 0.5), frameon=False)
plt.gca().add_artist(ab)

plt.axis('off')
plt.show()
```



```
In [ ]: img = plt.imread('Pictures/x_3.jpg')
im = OffsetImage(img, zoom=0.6)
ab = AnnotationBbox(im, (0.5, 0.5), frameon=False)
plt.gca().add_artist(ab)

plt.axis('off')
plt.show()
```



## 4. Difference in Mean Upper Body

- Jeglich die Oberarm Keypoints werden verglichen

### 4.1 Kompensation 01

```
In [ ]: selected_columns = ['x_0', 'y_0', 'z_0', 'x_11', 'y_11', 'z_11',
                         'x_12', 'y_12', 'z_12', 'x_13', 'y_13', 'z_13',
                         'x_14', 'y_14', 'z_14', 'x_15', 'y_15', 'z_15',
                         'x_16', 'y_16', 'z_16', 'x_17', 'y_17', 'z_17',
                         'x_18', 'y_18', 'z_18', 'x_19', 'y_19', 'z_19',
                         'x_20', 'y_20', 'z_20', 'x_21', 'y_21', 'z_21',
                         'x_22', 'y_22', 'z_22', 'x_23', 'y_23', 'z_23',
                         'x_24', 'y_24', 'z_24']
```

```
difference_in_mean_upper_body_01 = df_MediaP1[selected_columns].mean() -  
print(f"Mean Value in DeepMotion 01: {difference_in_mean_upper_body_01}")
```

```
Mean Value in DeepMotion 01: x_0      -3.319083  
y_0      73.683774  
z_0     -210.642338  
x_11    116.795024  
y_11    146.385612  
z_11    374.395101  
x_12   -149.673985  
y_12    104.762081  
z_12    374.873850  
x_13    11.958495  
y_13    257.475238  
z_13    353.311798  
x_14   -162.232370  
y_14    183.706348  
z_14    133.704332  
x_15    21.795657  
y_15    245.967297  
z_15    -91.969699  
x_16   -113.446733  
y_16    205.357660  
z_16   -389.343679  
x_17    16.749902  
y_17    238.363306  
z_17   -211.062132  
x_18    -94.916984  
y_18    212.191708  
z_18   -524.594197  
x_19    23.633374  
y_19    226.420394  
z_19   -202.265748  
x_20    -87.188510  
y_20    200.584724  
z_20   -479.274081  
x_21    11.851888  
y_21    223.093311  
z_21   -99.649882  
x_22    -76.151072  
y_22    188.579113  
z_22   -382.626055  
x_23    -19.884797  
y_23    183.582298  
z_23    957.603535  
x_24   -105.803264  
y_24    164.761015  
z_24    970.039118  
dtype: float64
```

```
In [ ]: selected_columns_X = ['x_0', 'x_11', 'x_12', 'x_13',  
                           'x_14', 'x_15', 'x_16', 'x_17',  
                           'x_18', 'x_19', 'x_20', 'x_21',  
                           'x_22', 'x_23', 'x_24']  
  
selected_columns_Y = ['y_0', 'y_11', 'y_12', 'y_13',  
                      'y_14', 'y_15', 'y_16', 'y_17',  
                      'y_18', 'y_19', 'y_20', 'y_21',  
                      'y_22', 'y_23', 'y_24']
```

```

selected_columns_Z = ['z_0', 'z_11', 'z_12', 'z_13',
                     'z_14', 'z_15', 'z_16', 'z_17',
                     'z_18', 'z_19', 'z_20', 'z_21',
                     'z_22', 'z_23', 'z_24']

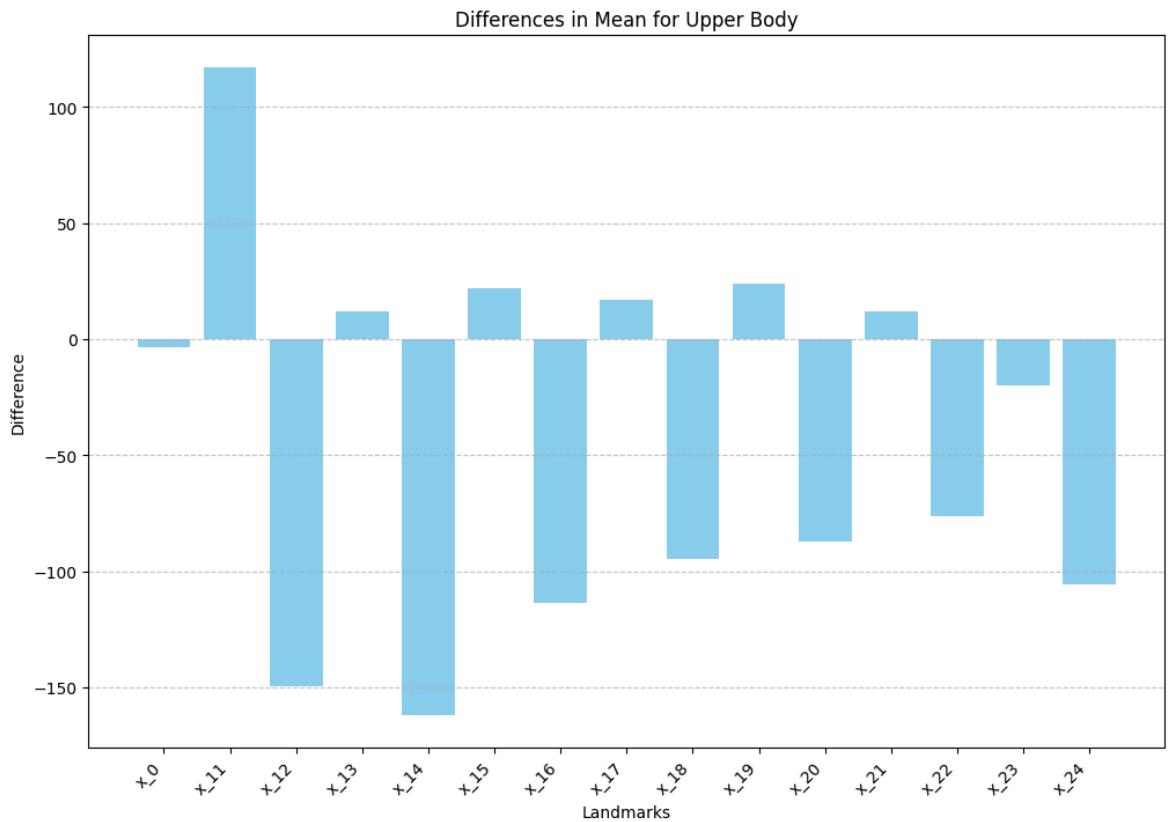
difference_in_mean_upper_body_01_X = difference_in_mean_upper_body_01[selected_columns_Z]
difference_in_mean_upper_body_01_Y = difference_in_mean_upper_body_01[selected_columns_Z]
difference_in_mean_upper_body_01_Z = difference_in_mean_upper_body_01[selected_columns_Z]

```

```

In [ ]: plt.figure(figsize=(12, 8))
plt.bar(difference_in_mean_upper_body_01_X.index, difference_in_mean_upper_body_01_X)
plt.title('Differences in Mean for Upper Body')
plt.xlabel('Landmarks')
plt.ylabel('Difference')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

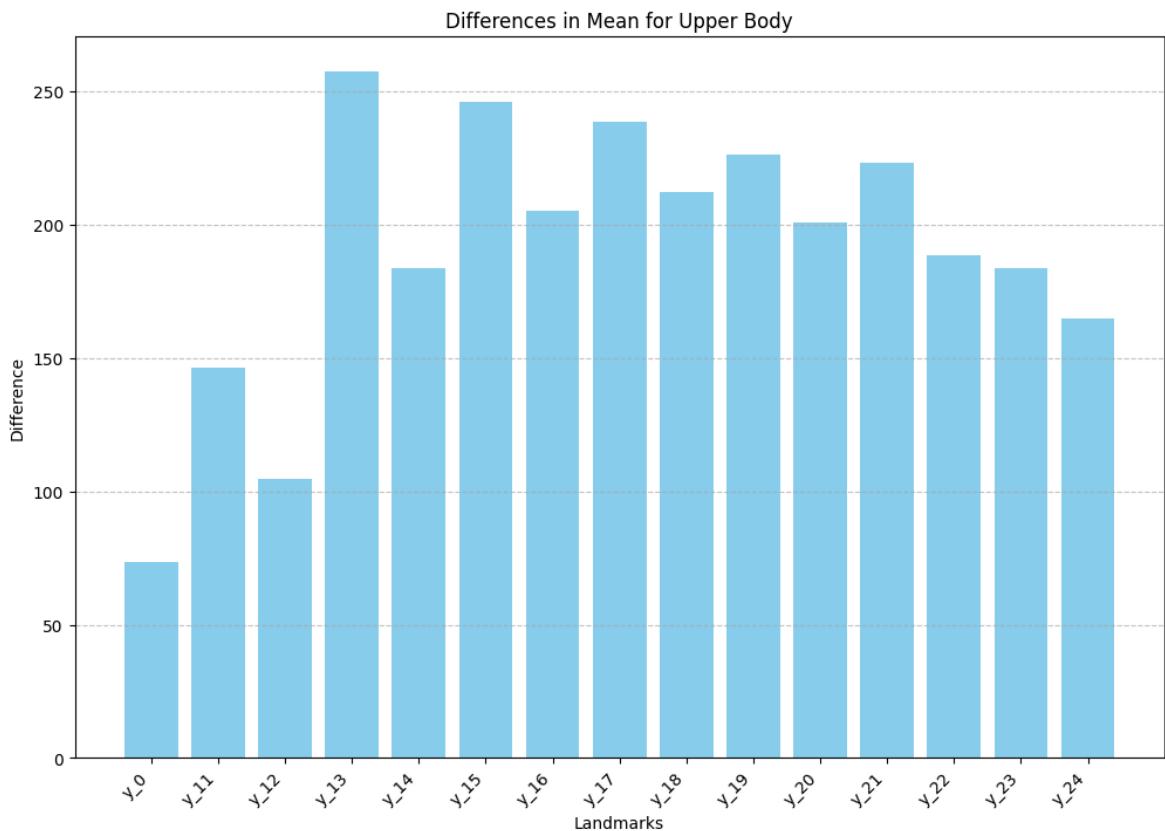
```



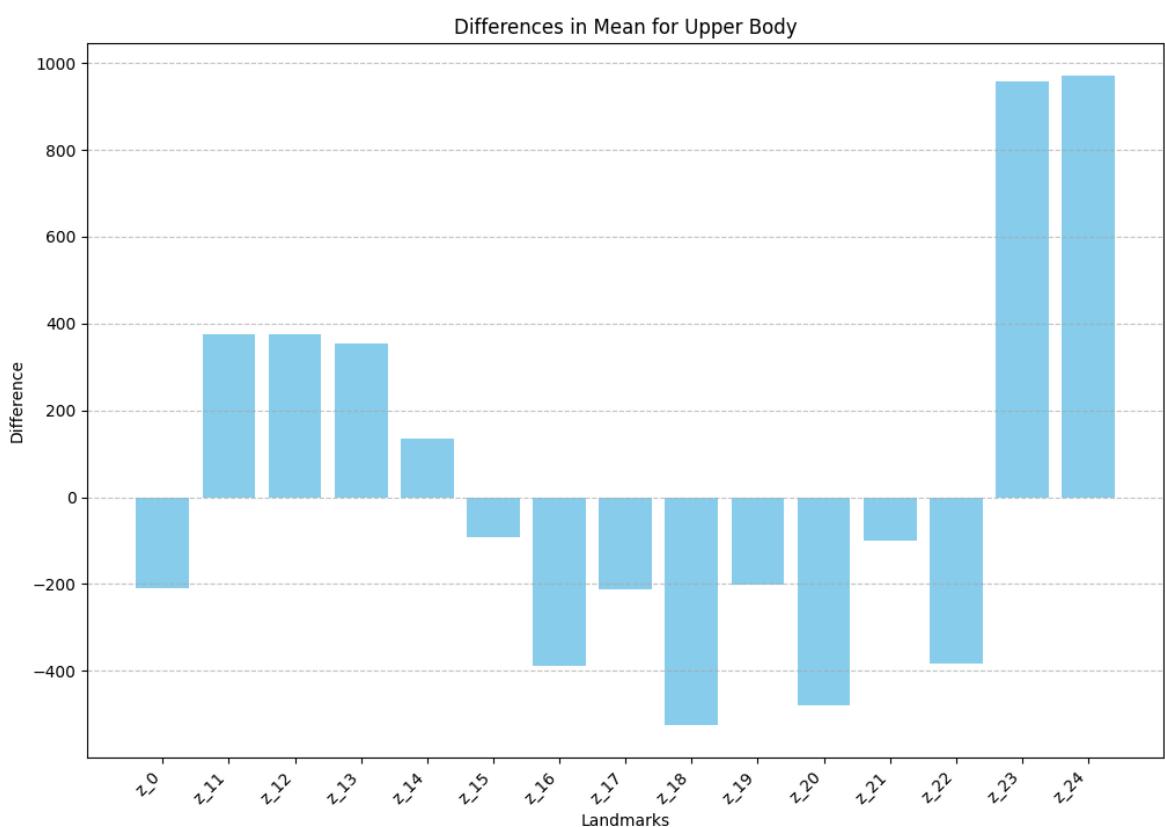
```

In [ ]: plt.figure(figsize=(12, 8))
plt.bar(difference_in_mean_upper_body_01_Y.index, difference_in_mean_upper_body_01_Y)
plt.title('Differences in Mean for Upper Body')
plt.xlabel('Landmarks')
plt.ylabel('Difference')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

```



```
In [ ]: plt.figure(figsize=(12, 8))
plt.bar(difference_in_mean_upper_body_01_Z.index, difference_in_mean_upper_body_01_Z)
plt.title('Differences in Mean for Upper Body')
plt.xlabel('Landmarks')
plt.ylabel('Difference')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



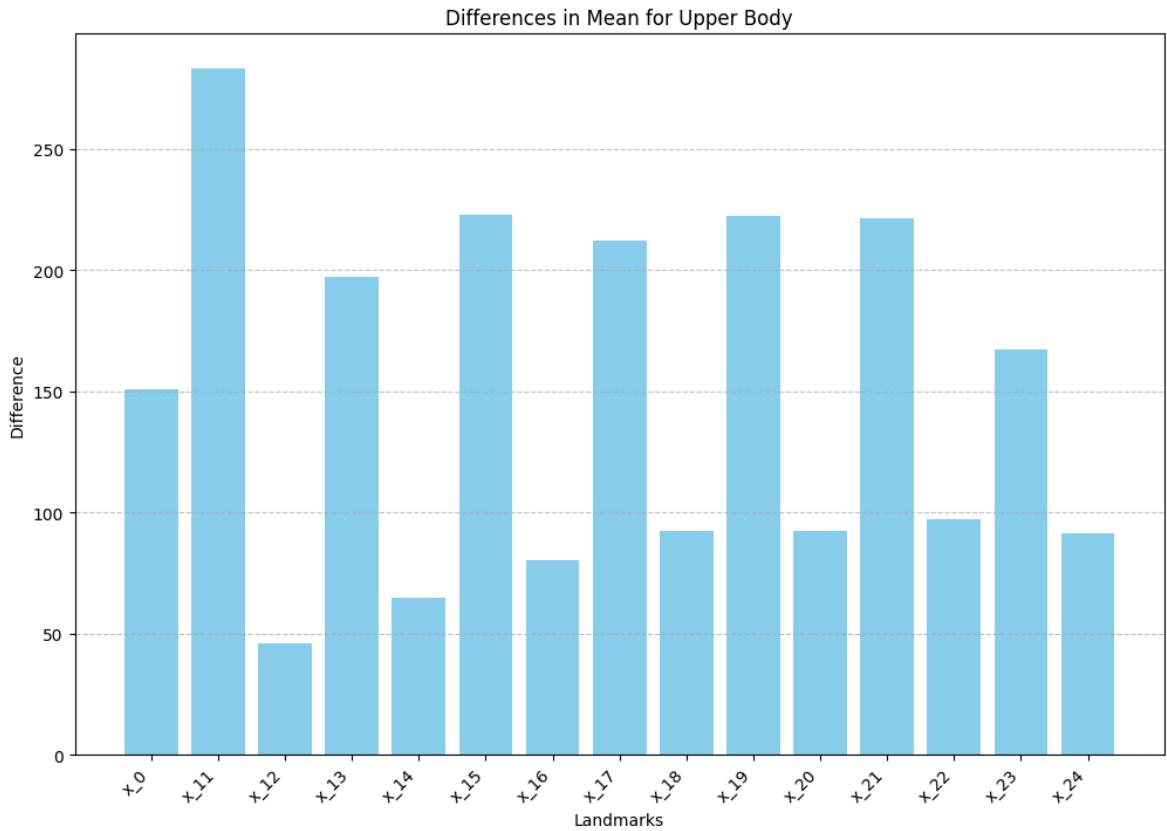
## 4.2 Kompensation 02

```
In [ ]: difference_in_mean_upper_body_02 = df_MediaP2[selected_columns].mean() -  
        print(f"Mean Value in DeepMotion 02: {difference_in_mean_upper_body_02}")  
  
Mean Value in DeepMotion 02: x_0      150.901215  
y_0      45.595955  
z_0     -227.528791  
x_11    283.252751  
y_11    61.692810  
z_11    285.009399  
x_12    46.207360  
y_12    15.589027  
z_12    262.641740  
x_13    197.326623  
y_13    102.834697  
z_13    315.100115  
x_14    65.072355  
y_14    74.540943  
z_14    132.690039  
x_15    222.777897  
y_15    126.753305  
z_15    -184.942157  
x_16    80.540867  
y_16    79.985625  
z_16    -418.256537  
x_17    212.372087  
y_17    155.749507  
z_17    -296.386476  
x_18    92.287134  
y_18    103.080864  
z_18    -534.293733  
x_19    222.254899  
y_19    158.058629  
z_19    -341.451427  
x_20    92.491910  
y_20    99.587247  
z_20    -536.114006  
x_21    221.153078  
y_21    136.507000  
z_21    -220.009200  
x_22    97.359988  
y_22    83.443998  
z_22    -434.171967  
x_23    167.449638  
y_23    204.225627  
z_23    529.435083  
x_24    91.554630  
y_24    196.521203  
z_24    480.946851  
dtype: float64
```

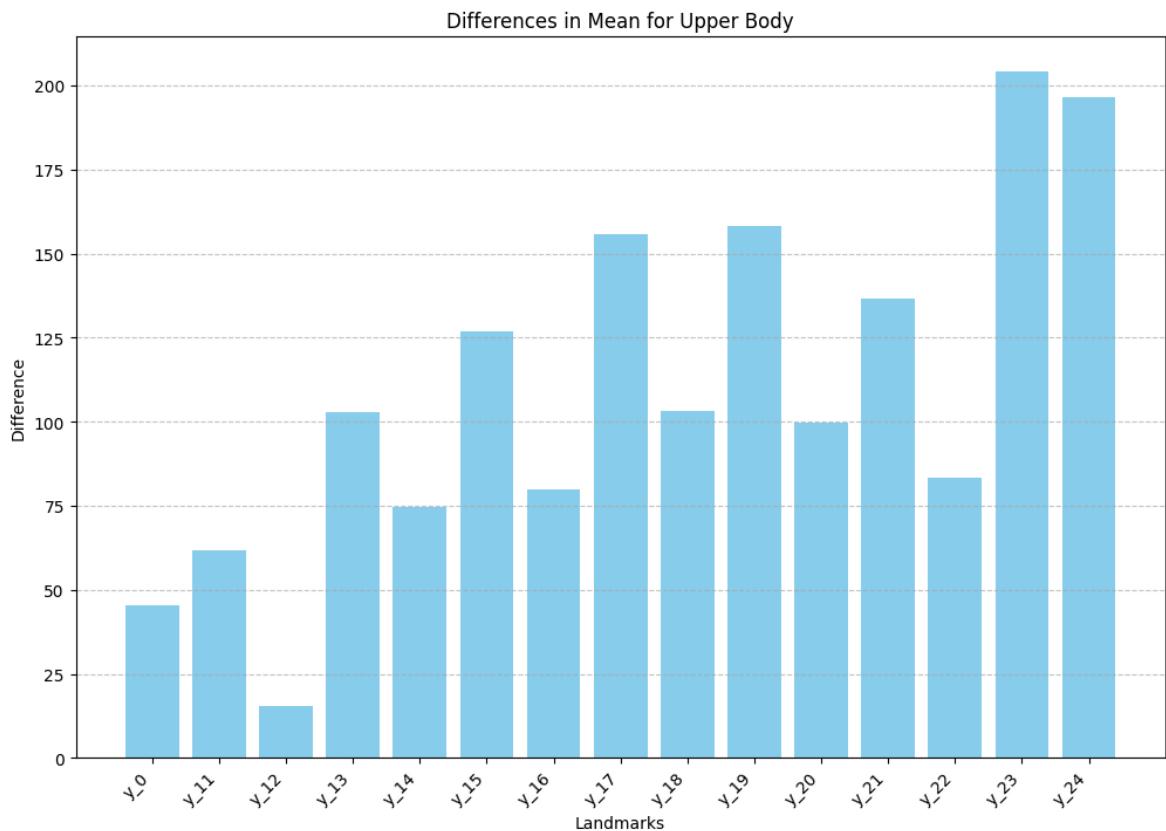
```
In [ ]: difference_in_mean_upper_body_02_X = difference_in_mean_upper_body_02[selected_columns[0]]  
difference_in_mean_upper_body_02_Y = difference_in_mean_upper_body_02[selected_columns[1]]  
difference_in_mean_upper_body_02_Z = difference_in_mean_upper_body_02[selected_columns[2]]
```

```
In [ ]: plt.figure(figsize=(12, 8))  
plt.bar(difference_in_mean_upper_body_02_X.index, difference_in_mean_upper_body_02_X)
```

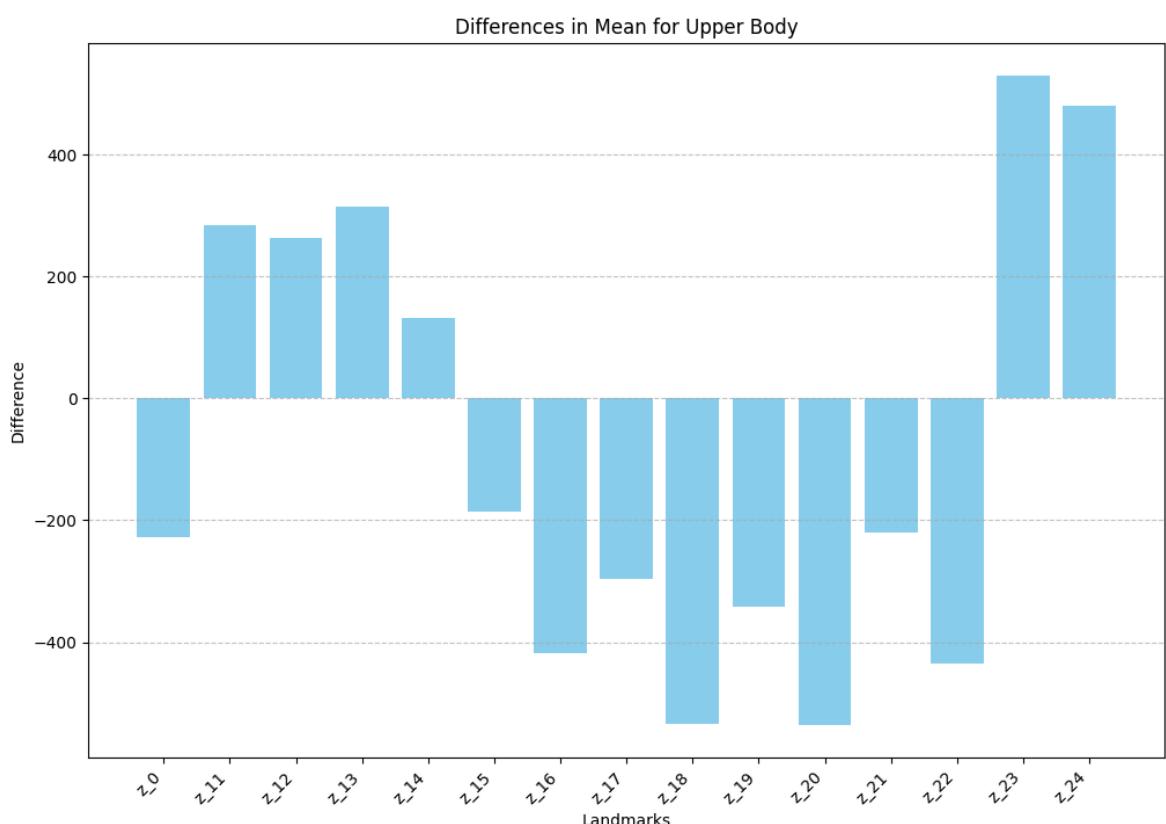
```
plt.title('Differences in Mean for Upper Body')
plt.xlabel('Landmarks')
plt.ylabel('Difference')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



```
In [ ]: plt.figure(figsize=(12, 8))
plt.bar(difference_in_mean_upper_body_02_Y.index, difference_in_mean_upper_body_02_Y)
plt.title('Differences in Mean for Upper Body')
plt.xlabel('Landmarks')
plt.ylabel('Difference')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



```
In [ ]: plt.figure(figsize=(12, 8))
plt.bar(difference_in_mean_upper_body_02_Z.index, difference_in_mean_upper_body_02_Z)
plt.title('Differences in Mean for Upper Body')
plt.xlabel('Landmarks')
plt.ylabel('Difference')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



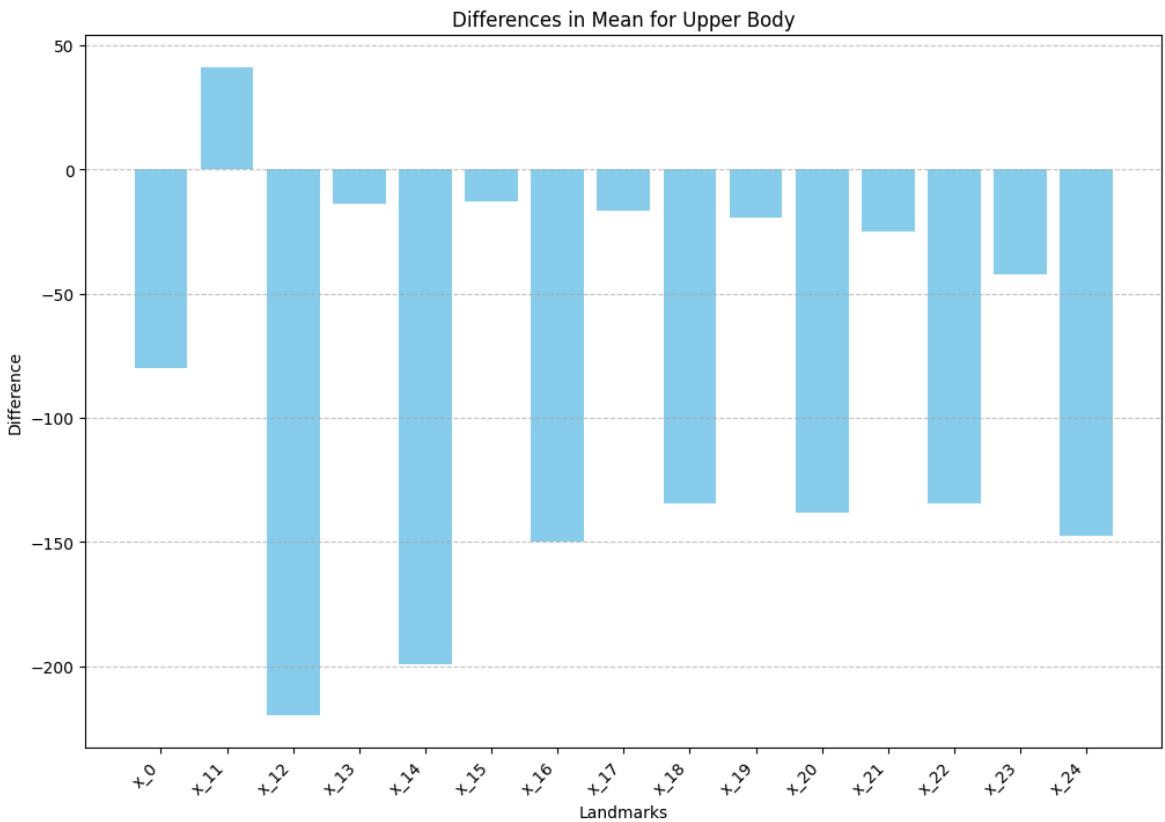
## 4.3 Nicht Kompensiert 03

```
In [ ]: difference_in_mean_upper_body_03 = df_MediaP3[selected_columns].mean() -  
        print(f"Mean Value in DeepMotion 03: {difference_in_mean_upper_body_03}")  
  
Mean Value in DeepMotion 03: x_0      -79.727432  
y_0      -12.878601  
z_0      -215.486841  
x_11     40.958609  
y_11     96.163076  
z_11     421.896135  
x_12     -219.902243  
y_12     81.928650  
z_12     427.840876  
x_13     -13.864860  
y_13     199.769663  
z_13     411.117255  
x_14     -198.952793  
y_14     176.548532  
z_14     145.939174  
x_15     -12.935756  
y_15     265.261322  
z_15     -157.637703  
x_16     -149.607376  
y_16     174.598473  
z_16     -581.207535  
x_17     -16.636216  
y_17     298.852667  
z_17     -310.012186  
x_18     -134.422691  
y_18     188.485895  
z_18     -749.404108  
x_19     -19.505289  
y_19     286.740996  
z_19     -335.861554  
x_20     -138.045069  
y_20     176.511838  
z_20     -732.746303  
x_21     -25.007999  
y_21     268.128056  
z_21     -187.053607  
x_22     -134.235969  
y_22     157.867234  
z_22     -592.268149  
x_23     -42.196064  
y_23     270.904955  
z_23     667.807884  
x_24     -147.509207  
y_24     273.400205  
z_24     721.503507  
dtype: float64
```

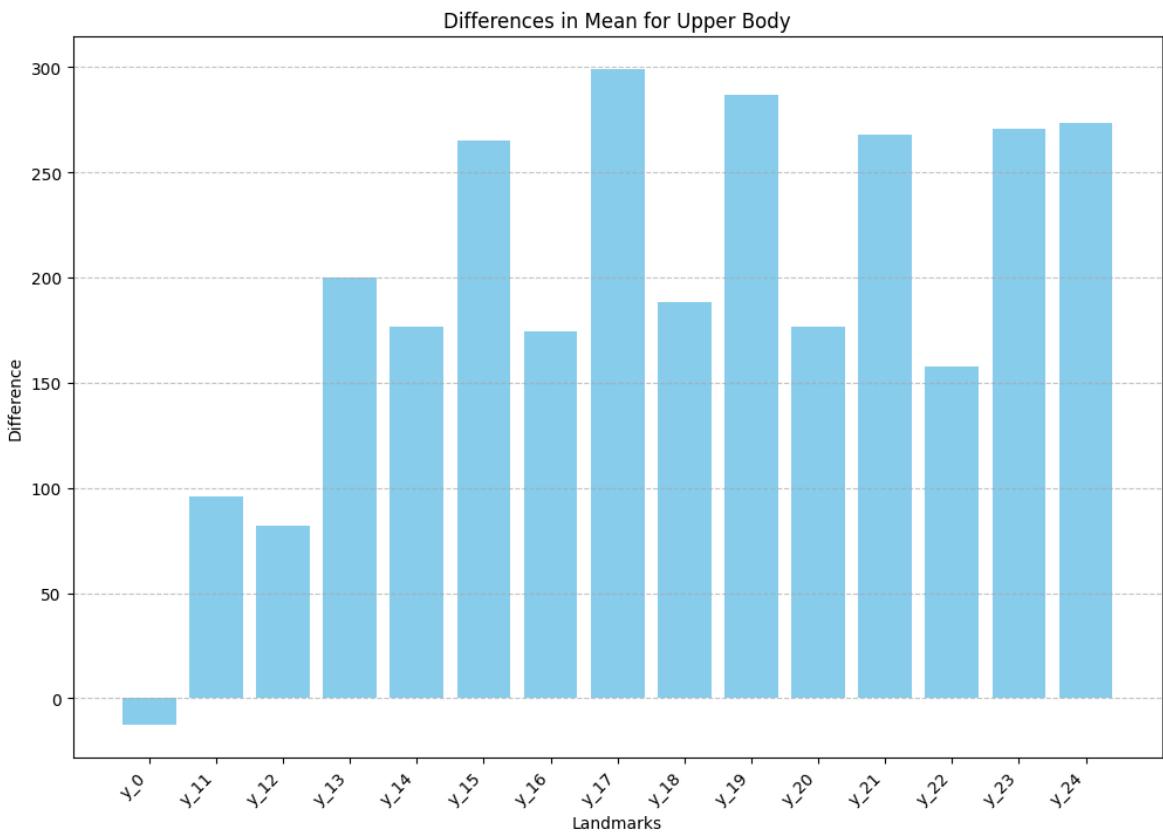
```
In [ ]: difference_in_mean_upper_body_03_X = difference_in_mean_upper_body_03[sel  
difference_in_mean_upper_body_03_Y = difference_in_mean_upper_body_03[sel  
difference_in_mean_upper_body_03_Z = difference_in_mean_upper_body_03[sel
```

```
In [ ]: plt.figure(figsize=(12, 8))  
plt.bar(difference_in_mean_upper_body_03_X.index, difference_in_mean_uppe
```

```
plt.title('Differences in Mean for Upper Body')
plt.xlabel('Landmarks')
plt.ylabel('Difference')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



```
In [ ]: plt.figure(figsize=(12, 8))
plt.bar(difference_in_mean_upper_body_03_Y.index, difference_in_mean_upper_body_03_Y)
plt.title('Differences in Mean for Upper Body')
plt.xlabel('Landmarks')
plt.ylabel('Difference')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



```
In [ ]: plt.figure(figsize=(12, 8))
plt.bar(difference_in_mean_upper_body_03_Z.index, difference_in_mean_upper_body_03_Z)
plt.title('Differences in Mean for Upper Body')
plt.xlabel('Landmarks')
plt.ylabel('Difference')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

