# Functies

def angle(df, s1, s2, s3):

return angle4(df, s1, s2, s3, s2)

Om een hoek te berekenen tussen 3 punten. Zie angle4 voor verduidelijking.

def angle4(df, s1, s2, s3, s4):

x1, y1, z1 = vnorm(df['x\_' + s1], df['y\_' + s1], df['z\_' + s1], df['x\_' + s2], df['y\_' + s2], df['z\_' + s2])

x2, y2, z2 = vnorm(df['x\_' + s3], df['y\_' + s3], df['z\_' + s3], df['x\_' + s4], df['y\_' + s4], df['z\_' + s4])

res = x1 \* x2 + y1 \* y2 + z1 \* z2

angle = np.arccos(res)

return 180 \* angle / math.pi

Berekent de hoek tussen twee 3d vectoren.

def diff(df, sl, sr):

return abs(df[sl] - df[sr])

Per frame wordt het absolute verschil tussen de linkersensor en de rechtersensor berekend.

def shoulder\_at\_max\_elbow(df, shoulder, elbow):

max\_elbow = np.max(df[elbow])

sh = df[df[elbow] == max\_elbow][shoulder]

return sh

Berekent de hoogte van de schouder wanneer de elleboog de maximale hoogte heeft bereikt.

def lowest\_max\_lr(df, sl, sr):

return min(np.max(df[sl]), np.max(df[sr]))

Voor de linkersensor en de rechtersensor wordt de maximale waarde berekend. Ook mogelijk om de hoek van de schouder hierin aan te roepen.

def get\_velocity(df, sensor):

velocity = df[[sensor]].diff(axis=0)

velocity = velocity.dropna()

return velocity

Berekent voor elk frame de snelheid van de sensor.

def get\_acceleration(df, sensor):

velocity = get\_velocity(df, sensor)

acceleration = get\_velocity(velocity, sensor)

acceleration = acceleration.dropna()

return acceleration

Berekent voor elk frame de versnelling van de sensor.

# Features

|  |  |  |  |
| --- | --- | --- | --- |
| **Featurenaam** | **Functie** | **Oefeningen** | **Operaties** |
| right\_shoulder | angle | AB, AF, RF | min, max |
| left\_shoulder | angle | AB, AF, RF | min, max |
| right\_arm | angle4 | EL | min, max |
| left\_arm | angle4 | EL | min, max |
| diff\_x\_wrist | diff | AF, RF, EL | mean, max |
| diff\_x\_elbow | diff | AF, RF, EL | mean, max |
| diff\_x\_shoulder | diff | AF, RF, EL | mean, max |
| diff\_y\_wrist | diff | AB, AF, RF, EL | mean, max |
| diff\_y\_elbow | diff | AB, AF, RF, EL | mean, max |
| diff\_y\_shoulder | diff | AB, AF, RF, EL | mean, max |
| diff\_z\_wrist | diff | AB, AF, RF | mean, max |
| diff\_z\_elbow | diff | AB, AF, RF | mean, max |
| diff\_z\_shoulder | diff | AB, AF, RF | mean, max |
| z\_left\_sh\_max\_el | shoulder\_at\_max\_elbow | AB, AF, RF | mean |
| z\_right\_sh\_max\_el | shoulder\_at\_max\_elbow | AB, AF, RF | mean |
| z\_elbow | lowest\_max\_lr | AB, AF, RF | max |
| z\_wrist | lowest\_max\_lr | AB, AF, RF | max |
| lowest\_max\_shoulder\_angle | lowest\_max\_lr | AB, AF, RF | max |
| vel\_elbows\_x\_l | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_x\_r | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_y\_l | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_y\_r | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_z\_l | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_z\_r | get\_velocity | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_x\_l | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_x\_r | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_y\_l | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_y\_r | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_z\_l | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_z\_r | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| angular\_vel\_elbow\_l | abs(self[‘vel\_elbow\_l’]/self[‘upper\_arm\_left’] | AB, AF, RF | mean, max, std |
| angular\_vel\_elbow\_r | abs(self[‘vel\_elbow\_r’]/self[‘upper\_arm\_right’] | AB, AF, RF | mean, max, std |
| angular\_acc\_elbow\_l | abs(self[[‘angular\_vel\_elbow\_l]].diff(axis=0) | AB, AF, RF | mean, max, std |
| angular\_acc\_elbow\_r | abs(self[[‘angular\_vel\_elbow\_r]].diff(axis=0) | AB, AF, RF | mean, max, std |
| angular\_vel\_wrist\_l | abs(self[‘vel\_wrist\_l’]/self[‘forearm\_left’] | EL | mean, max, std |
| angular\_vel\_wrist\_r | abs(self[‘vel\_wrist\_r’]/self[‘forearm\_right’] | EL | mean, max, std |
| angular\_acc\_elbow\_l | abs(self[[‘angular\_vel\_wrist\_l]].diff(axis=0) | EL | mean, max, std |
| angular\_acc\_elbow\_r | abs(self[[‘angular\_vel\_wrist\_r]].diff(axis=0) | EL | mean, max, std |

# Featureverantwoording

## Symmetrie links/rechts

Een gezond persoon zal een hoge mate van symmetrie vertonen bij het uitvoeren van een oefening. De symmetrie kan gemeten worden door per frame het absolute verschil tussen een sensor op de linkerarm en de rechterarm (polsen of ellebogen) te berekenen. Door al deze verschillen bij elkaar op te tellen en vervolgens te delen door het aantal frames, krijg je het gemiddelde absolute verschil per frame. Deze waarde kan gebruikt worden om de symmetrie te vergelijken. Hoe kleiner de waarde, hoe hoger de mate van symmetrie.

Ook kan het maximale verschil bepaald worden. Hoe groter dit verschil, hoe waarschijnlijker dat de proefpersoon een patiënt betreft.

|  |  |  |  |
| --- | --- | --- | --- |
| diff\_x\_wrist | diff | AF, RF, EL | mean, max |
| diff\_y\_wrist | diff | AF, RF, EL, AB | mean, max |
| diff\_x\_elbow | diff | AF, RF, EL | mean, max |
| diff\_y\_elbow | diff | AF, RF, EL, AB | mean, max |
| diff\_x\_shoulder | diff | AF, RF, EL | mean, max |
| diff\_y\_shoulder | diff | AF, RF, EL, AB | mean, max |

## Symmetrie hoogte

Bij een gezond persoon zullen de linkerarm en rechterarm zo goed als symmetrisch de armen omhoog (over de z-as) bewegen. Iemand met een schouderblessure zal de geblesseerde arm moeizaam omhoog bewegen. De niet-geblesseerde arm zal daarentegen op de normale manier naar boven bewegen. Dit betekent dat er een verschil zal zijn in de hoogte tussen de linker- en rechterarm.

|  |  |  |  |
| --- | --- | --- | --- |
| diff\_z\_wrist | diff | AF, RF, AB | mean, max |
| diff\_z\_elbow | diff | AF, RF, AB | mean, max |
| diff\_z\_shoulder | diff | AF, RF, AB | mean, max |

## Maximale hoogte/hoek (bewegingsrange)

Een gezond persoon zal een grotere bewegingsrange hebben dan een persoon met een schouderafwijking. De bewegingsrange is gemeten door de maximale waarde van een sensor op de linkerarm en de rechterarm te bepalen. Voor de elleboog en pols geldt dat de arm met de kleinste bewegingsrange de geblesseerde arm is.

|  |  |  |  |
| --- | --- | --- | --- |
| z\_elbow | lowest\_max\_lr | AB, AF, RF | *max* |
| z\_wrist | lowest\_max\_lr | AB, AF, RF | *max* |
| x\_wrist | lowest\_max\_lr | EL | *max* |
| lowest\_max\_shoulder\_angle | lowest\_max\_lr | AB, AF, RF | *max* |

## Beweging over de verkeerde as

Bij het uitvoeren van de bewegingen is er gewoonlijk 1 as waarover geen veranderingen zouden moeten zijn. Voor de oefeningen AF, RF en EL (schouder) is dit de y-as en voor de oefening AB is dit de x-as. Een grotere beweging over de verkeerde as kan een indicatie zijn voor een schouderblessure. Het bewegen over de verkeerde as kan gezien worden als het vermijden van een pijnpunt.

|  |  |  |  |
| --- | --- | --- | --- |
| diff\_x\_elbow | diff | AB | min, max, mean |
| diff\_y\_elbow | diff | AF, RF | min, max, mean |
| diff\_y\_shoulder | diff | EL | min, max, mean |

## Hoogte schouders bij maximale ellebooghoogte

Bij de oefeningen AB, RF en AF zou de schouder bij het optillen van de armen minimaal opgetrokken moeten worden. Iemand met een schouderblessure zal ter compensatie voor het missen van hoogte of om pijn te ontwijken, wel de schouder optrekken.

Hiervoor wordt de hoogte van de schouder berekend zodra de elleboog de maximale hoogte heeft bereikt.

|  |  |  |  |
| --- | --- | --- | --- |
| z\_left\_sh\_max\_el | shoulder\_at\_max\_elbow | AB, AF, RF | max |
| z\_right\_sh\_max\_el | shoulder\_at\_max\_elbow | AB, AF, RF | max |

## (Hoek)snelheid

Iemand met een schouderblessure zal de oefening op een lagere snelheid uitvoeren dan iemand zonder schouderblessure. Het kost meer moeite om de arm te bewegen, wat resulteert in een langzamere uitvoering.

|  |  |  |  |
| --- | --- | --- | --- |
| vel\_elbows\_x\_l | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_x\_r | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_y\_l | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_y\_r | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_z\_l | get\_velocity | AB, AF, RF, EL | mean, max, std |
| vel\_elbows\_z\_r | get\_velocity | AB, AF, RF, EL | mean, max, std |
| angular\_vel\_elbow\_l | abs(self[‘vel\_elbow\_l’]/self[‘upper\_arm\_left’] | AB, AF, RF | mean, max, std |
| angular\_vel\_elbow\_r | abs(self[‘vel\_elbow\_r’]/self[‘upper\_arm\_right’] | AB, AF, RF | mean, max, std |
| angular\_vel\_wrist\_l | abs(self[‘vel\_wrist\_l’]/self[‘forearm\_left’] | EL | mean, max, std |
| angular\_vel\_wrist\_r | abs(self[‘vel\_wrist\_r’]/self[‘forearm\_right’] | EL | mean, max, std |

## (Hoek)versnelling

Veel variatie in de versnelling kan een indicatie geven voor het hebben van een schouderblessure. Iemand zonder blessure zal de oefening vloeiend, met een redelijke constante snelheid, uitvoeren. (Een constante snelheid betekent geen versnelling). Iemand met een pijnlijke arm zal de oefening minder vloeiend, meer hakkelend, uitvoeren. Dit betekent dat er variatie is in de versnelling, de versnelling kan zowel positief als negatief zijn.

|  |  |  |  |
| --- | --- | --- | --- |
| acc\_elbows\_x\_l | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_x\_r | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_y\_l | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_y\_r | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_z\_l | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| acc\_elbows\_z\_r | get\_acceleration | AB, AF, RF, EL | mean, max, std |
| angular\_acc\_elbow\_l | abs(self[[‘angular\_vel\_elbow\_l]].diff(axis=0) | AB, AF, RF | mean, max, std |
| angular\_acc\_elbow\_r | abs(self[[‘angular\_vel\_elbow\_r]].diff(axis=0) | AB, AF, RF | mean, max, std |
| angular\_acc\_elbow\_l | abs(self[[‘angular\_vel\_wrist\_l]].diff(axis=0) | EL | mean, max, std |
| angular\_acc\_elbow\_r | abs(self[[‘angular\_vel\_wrist\_r]].diff(axis=0) | EL | mean, max, std |