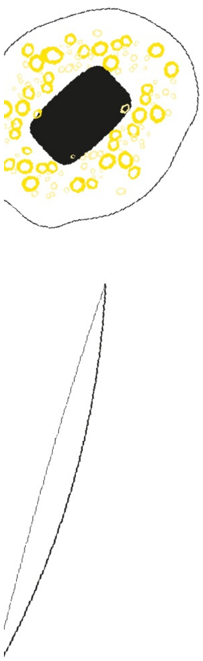
Using frame rotations based on movement direction to improve reaching event detection from IMU data.



Daniel Los

Universiteit Twente

June 13, 2024

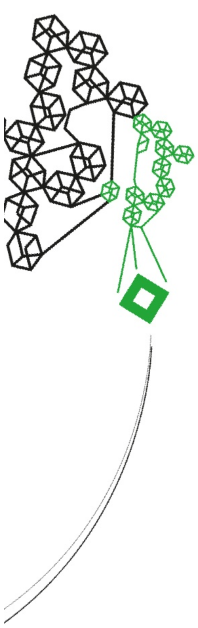


Table of Contents

1 Introduction

After stroke, many patients experience impaired motor function that hamper their Activities of Daily Life (ADL) in their upper limbs [4].
In order to optimize the rehabilitation procedure, an objective evaluation of movements during ADL is essential [5]. In order to facilitate this evaluation, an automated system that can be used at home is needed.
One way of realizing such a system is a set of three inertial measurement units (IMUs), worn by the subject on the sternum, left wrist and right wrist. These devices collect acceleration, angular velocity and magnetic field data. The data from these sensors can be used to reconstruct the movement of these patients. The data processing side of such a system would consist of four parts. The first part is recognition of movement events from the raw data. When this system is used in a home situation, no information is available regarding the actions the subject has taken, so these actions need to be identified. Second, every event is classified into one of a set of actions (e.g. reaching, waving, drinking), so movement events that are relevant for a certain subject can later easily be filtered out. Third, the quality of every movement needs to be assessed. Finally, a clear summary of this information needs to be presented to care-professionals so they require less time to interpret it[5].
Part 2, the classification of movement events, using an Artificial Neural Network (ANN), has some promise, but suffers from low recall and precision [1][2]. In a study classifying volleyball actions using

IMU data, the recall and precision of a machine learning method were improved by compensating the IMU data for the orientation of the players[3]. In this study we will reimplement the Matlab script used in [2] with the Keras library in python (see 3.4), and adapt this method and define a reaching frame for each reaching event and then rotate the IMU data of that event to that reaching frame. The reaching frame is based on the direction of the movement and should make the data from reaching events in different directions look more similar. We will compare the performance of an ANN classifier using data in the reaching frame under a variety of conditions with the results from [2], and see if the method used in this study outperforms the methods in [2].

2 Background

This section will describe the hardware used in this study, provide background information on the sensor and global frame, and give a basic introduction to neural networks. Then the previous research is discussed, and the rationale behind the chosen approach to address some shortcomings in those studies will be explained.

2.1 Hardware

The data used in this study was acquired from subjects wearing an Xsens MTw IMU sensor suit. This suit consists of a set of 17 IMUs distributed over the body. Every IMU samples the acceleration, angular velocity and magnetic field along 3 axes at 120 Hz. The Xsens MVN Analyze software provides orientation data for the sensors with respect to the global frame, based on a Kalman filter sensor fusion algorithm [10]. We will not use the data of all 17 IMUs, but only the data of the IMUs worn on the left and right wrist, and the sensor on the sternum. The reason for this is that such a set would be less cumbersome to wear during daily life than a full 17 sensor suit.

Figure 1: A woman wearing a set of MTw sensors [13]

2.2 Sensor frame and global frame

When data is recorded with an IMU, the data is in the sensor frame. This means the orientation of the x, y and z axes of the data align with the orientation of the sensor. For example, two identical movements will net different data in the sensor frame if the sensor is displaced between these two movements. This is a problem if the gravity-free acceleration is needed as it is unknown in what direction gravity is pointing relative to the sensor frame. As we want the data of a movement to be the same, irrespective of how the sensor was placed on the body, we need the gravity-free acceleration. To solve this problem, the sensor data can be rotated to a global frame. A global frame is defined by some factors outside the sensor that are relatively constant and independent of sensor orientation: for example the direction of gravity and the direction of magnetic north. If one can determine the orientation of the sensor with regard to this global frame, and then rotate the data in the sensor frame according to this orientation, the data will be in the global frame. Figure 2 Shows IMU acceleration and angular velocity data in the sensor frame, in the global frame, and in the global frame with gravity subtracted from the acceleration data. As we will see in 3.3.1, we will transform the data to a global frame in this study.

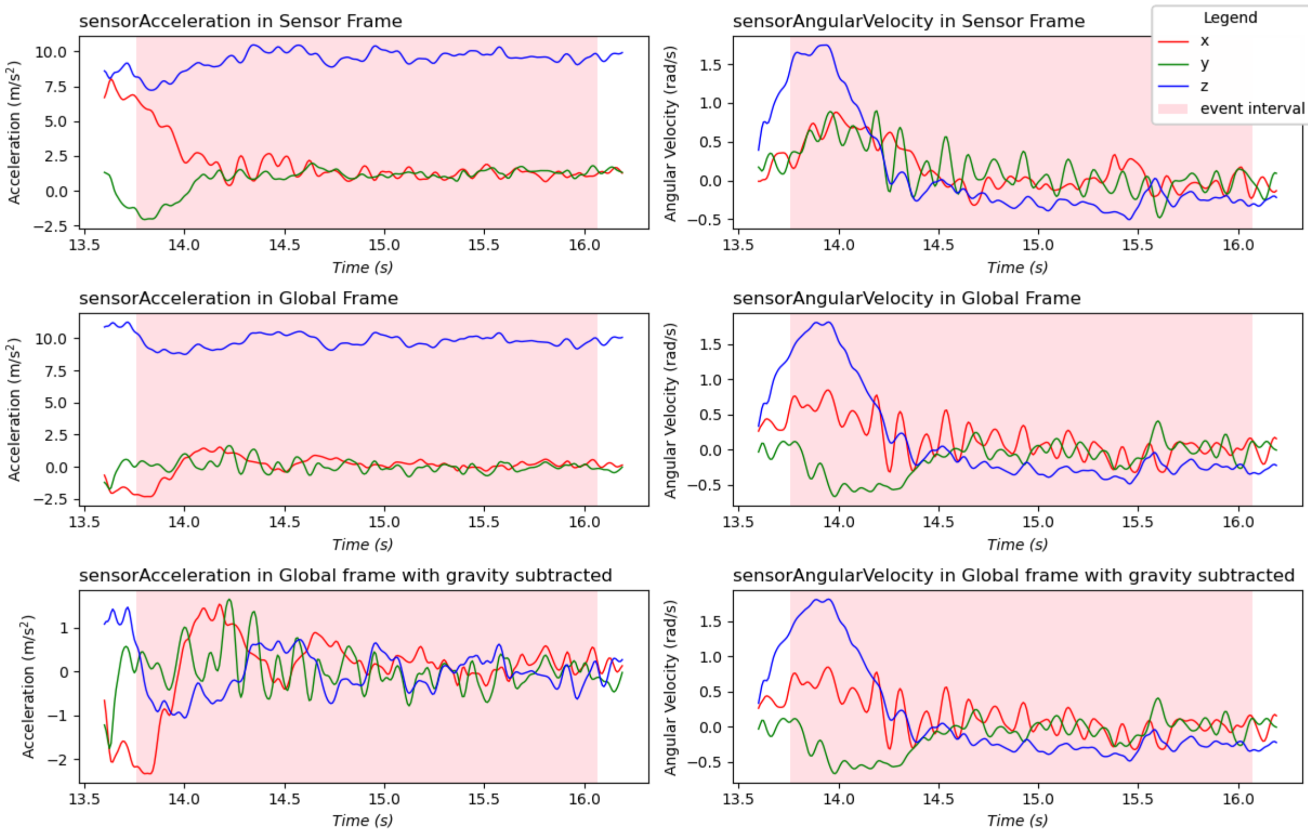


Figure 2: Plots of the acceleration and angular velocity data from a random event of data in this study in different frames

2.3 Basics of Artificial Neural Networks

The classifier used in this study is an Artificial Neural Network(ANN). There is a huge variety in the structure of neural nets, but all neural nets have the same basic structure. One component is the artifial neuron or node as shown in Figure 3.

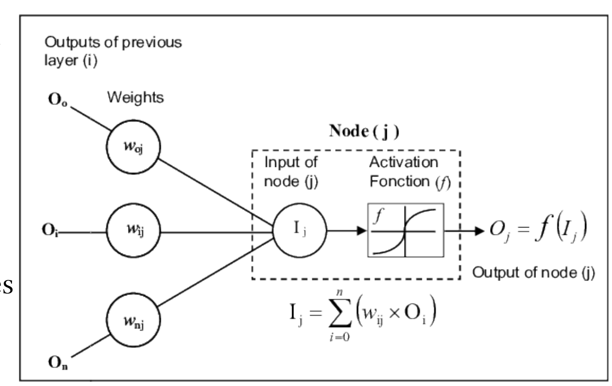


Figure 3: A node in an artificial neural network

A node receives some numbers as inputs, multiplies every input it receives with a weight, sums the results, and passes this result through some nonlinear function called the activation function to yield an output number.

An artificial neural network is structured in layers of nodes as seen in Figure 4. The input data is usually only fed to the first layer of nodes, and deeper layers receive the output of previous layers as input. The output of the last layer is considered the output of the net.

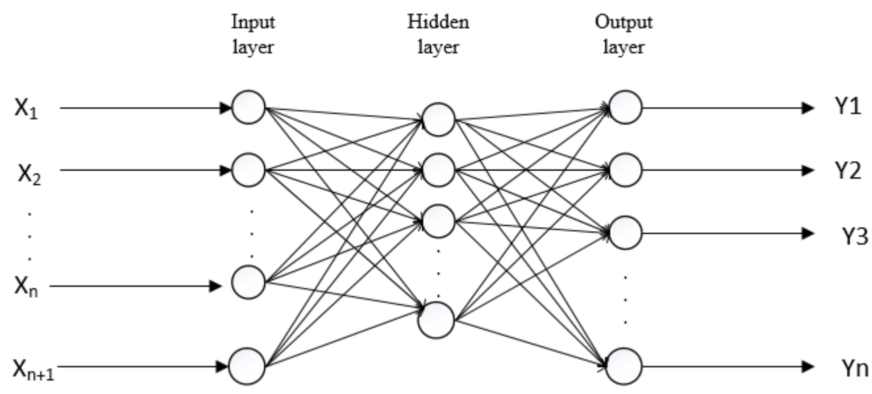


Figure 4: Example of an artificial neural network

The initial weights for every node are randomized, and then the network needs to be trained. In the case of supervised learning, a training set of (label, input data) pairs is used to train the network. The label holds the class the input data belongs to. The input data is presented to the network in small subsets called batches. Then, the output class of the network is compared to the ground truth class in the labels, and network weights are updated based on their difference, usually by some variation of gradient descent. When all the training data has been passed through the network once, this is called an epoch. After some number of epochs, training is stopped and the ANN is considered finished and can be used for classification.

In order to properly evaluate a machine learning classifier, the input samples need to be split into a training set and a test set. The test set is only used for evaluating the performance of the classifier after training has finished and is not used during training. The reasoning is that a classifier might learn to classify the samples it was trained on correctly, but it might not generalize very well to samples from outside its training set. This is called overfitting. By having a separate test set, accuracy of the model on unknown samples can be estimated.

2.4 Previous research

Van Gils [2] studied the performance of an ANN classifier for classifying planar reach events using a dataset obtained from post stroke subjects performing several motion tasks.

One issue with this research is that a separate model was trained for every subject. It was stated this was done because during rehabilitation, the severity of impairment changes, which might negatively impact the performance of a pre-trained classifier. This might be true, but the performance of a single model was not explored, even though this could save subjects a calibration run to train a subject specific model.

Another issue of the study was that the experiments were all performed with the subjects sitting at a desk oriented in one direction. After rotation to the global frame, had subjects been seated rotated randomly in the horizontal plane, IMU data of reaching events would vary more in the x- and y-components, as they would reach in a different direction. Because of this increase in variability in the data, classification results might suffer, but this was not explored.

The study also did not check performance of 3D reaching events, though [1] did.

An ANN classifier will be used, similar to the system used in [1] and [2] and, but with some modifications. The study in [3] found that compensating for the subjects' orientations yielded an improvement in accuracy for their classifier. Though the classifier used in [3] was not an ANN, it used input features very similar to the ANN classifier used in this study. For this reason it was deemed to be worth evaluating the performance impacts of this method in the context of an ANN classifier.

3 Method

This section will describe how this study was carried out. First, the dataset origin, the labeling process and the preprocessing stage will be described. After that the Reaching Frame Transformation (RFT) and Horizontal Plane Rotation (HPR) methods are specified.

3.1 Dataset

The data used in this study was acquired in the INTERACTION [6][7] project. During this project, several post stroke subjects performed clinical tests to assess their sensorimotor impairment, including Stroke Upper Limb Capacity Scale (SULCS), Fugl-Meyer Assesment, Berg Balance Scale (BBS) and several reaching tasks while wearing an Xsens MTw IMU sensor suit. For every subject, a number of different files with IMU data were available, each file containing the data of the subject performing tasks for 1-5 minutes.

3.2 Labeling

The ANN classifier used in this study is a machine learning method that uses supervised learning during its training phase. This means the input data used for training the classifier is annotated with a label, indicating what kind of movement (the class), such as "reach left planar" that data belongs to. The first step in labeling the data is determining what classes will be distinguished. After that the data needs to labeled according to which of the classes it belongs to.

In this study, planar reaching events and 3D reaching events will be considered. Reaching events were interpreted according to the definition of reaching by the International Classification of Functioning, Disability and Health: "Using the hands and arms to extend outwards and touch and grasp something, such as when reaching across a table or desk for a book."[8]. "Planar reaching" refers to a reaching motion in the horizontal plane, while "3D reaching" refers to a reaching motion with an upwards or downwards component. The following classes were used in this study: "reach left planar" and "reach right planar" for planar reaching with the left and right arms respectively; "reach left 3D", "reach right 3D" for 3D reaching with the left and right arm respectively; and "no event" for all other data.

In [2], planar reaching events were labeled. In this study, those labels are reused. There was also some unlabeled data from the same subjects while they were performing 3D reaching movements; these data were also labeled and used in this study. Labeling was done as in [2]: for every used data file, a corresponding video with a 3D representation of the subject produced by Xsens MVN Analyze was used as a reference. 3D reaching events were interpreted according to the ICF definition and the start and stop time of the event together with its label were saved to a text file for later use.

	Subject			
Label	QMS01	QMS02	QMS03	QMS04
right_hand_planar_reach	37	38	24	41
left_hand_planar_reach	36	39	17	36
right_hand_3d_reach	17	20	0	0
left_hand_3d_reach	17	18	0	0

Table 1: Number of events per labeled category, per subject.

3.3 Preprocessing

Preprocessing consists of two parts: the first part are changes made to the IMU data that preserve the time series nature of the data. The second part consist of using this processed IMU data to generate features suitable for use in the ANN classifier.

3.3.1 IMU time series preprocessing

First, a low pass filter of order 6 with its cutoff frequency at 24 Hz was applied to the acceleration and angular velocity data to reduce noise [9].

Second, the data was transformed from the sensor frame to the global earth frame. This was done using the orientation data supplied by the Xsens MVN Analyse software. The global frame used by the Xsens software is based on gravity and magnetic north [10].

Besides movement induced acceleration, the acceleration sensors will always measure the gravity of Earth. As the data is now in the global frame, gravity can be removed from the sensor acceleration data by subtracting a constant vector of 9.81 in the z-axis.

After this process, the norm of the acceleration and the norm of the angular velocity data was calculated and added to the data.

The data is now ready for the next step: feature generation.

3.3.2 Feature generation

In order to train the ANN, input values for the network are needed . These input data are called features. As described in [1], the IMU data will be divided into 1 second segments, each next segment overlapping 80% with the previous one. For every segment, the event associated with that segment was determined and its class was added to it as the label. All segments fully contained in the duration of a reaching event are labeled with that events class. The segment preceding the first segment fully contained in an event, and the segment following the last segment fully contained in an event are also considered part of that event. All other segments are labeled “no event”

An example of the labeling of segments around an event can be seen in Figure 5.

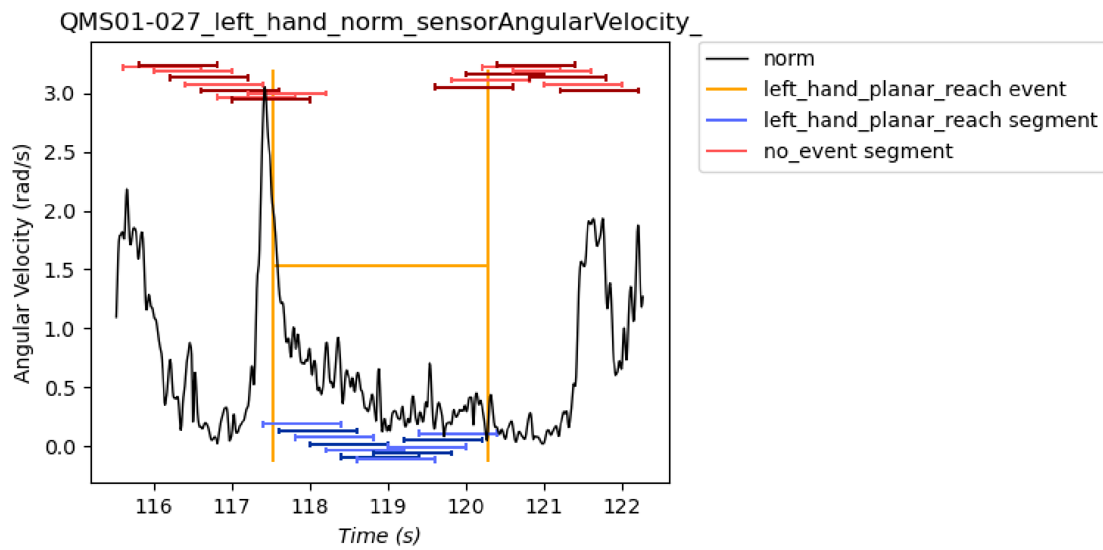


Figure 5: Data from an event used in this study. Segment intervals and their labels shown, with the interval of the event, and the norm of acceleration data.

Then, for every segment, the following metrics will be calculated: for every sensor, the mean and standard deviation of the x-, y- and z-component and the norm for both the angular velocity and the acceleration data. This results in 3 (sensors) * 2 (mean and stdev), * 4 (x, y, z, norm) * 2 (angular velocity and acceleration) = 48 features per segment.

This results in a collection of labeled features, called the feature set, which will be used to train the ANN classifier. One such set of 48 features with a label will be called a sample.

3.4 ANN classifier

The ANN used in this study uses the same structure as in [1][2]: A fully connected net with one hidden layer of 15 nodes using the tanh activation function, and an output layer with N outputs that uses the softmax activation function, where N is the number of classes present in the input data. The scaled conjugate gradient backpropagation optimizer used in [2] was not available in Keras, so the Adam optimizer was used, as it does not require much tuning of the hyperparameters [12].

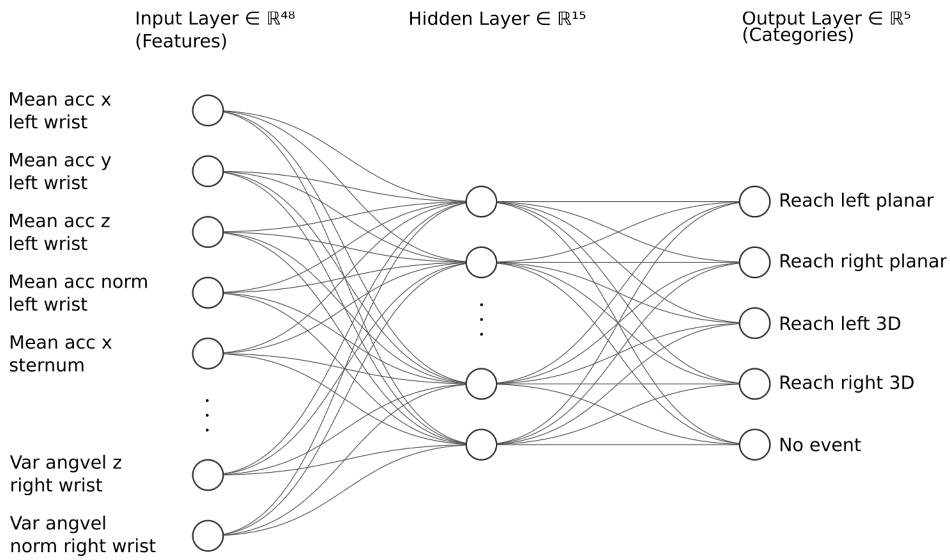


Figure 6: Topology of the network used in this study

Before training starts, the labels of the training set are one-hot encoded. The classes present in the feature set are numbered 0 to N-1 (N = the number of distinct classes), and the labels in the feature set are replaced by an array containing a 1 at the index corresponding to the label number.

Events			
Data	**Start**	**Stop**	**Label**
`0xfa89b3c…`	`0:04`	`0:7`	`No event`
`0x29ae765d…`	`0:22`	`0:24`	`Reach left planar`
`0x305009e…`	`0:31`	`0:36`	`No event`
`0x0d1e5a2…`	`0:46`	`0:48`	`Reach right planar`
`0x2a87e4f…`	`0:51`	`0:55`	`Reach right 3D`
`0x3037ea9…`	`1:02`	`1:10`	`No event`
`0xc678da3…`	`1:11`	`1:15`	`Reach left 3D`

⟹

Events							
Data	**Start**	**Stop**	**Reach left planar**	**Reach right planar**	**Reach left 3D**	**Reach right 3D**	**No Event**
`0xfa89b3c…`	`0:04`	`0:7`	`0`	`0`	`0`	`0`	`1`
`0x29ae6d5…`	`0:22`	`0:24`	`1`	`0`	`0`	`0`	`0`
`0x305009e…`	`0:31`	`0:36`	`0`	`0`	`0`	`0`	`1`
`0x0d1e5a2…`	`0:46`	`0:48`	`0`	`1`	`0`	`0`	`0`
`0x2a87e4f…`	`0:51`	`0:55`	`0`	`0`	`0`	`1`	
`0x3037ea9…`	`1:02`	`1:10`	`0`	`0`	`0`	`0`	`1`
`0xc678da3…`	`1:11`	`1:15`	`0`	`0`	`1`	`0`	`0`

Table 2: Example of conversion from text labels to one-hot encoded labels

This enables the network to be trained by comparing the network output (an array containing a number between 0 and 1 for every class) to the label (an array containing a 1 for the correct label and 0 otherwise). The softmax activation function in the output layer normalizes the output array to numbers between 0 and 1 which sum to 1. The output array can be interpreted as the probabilities the network assigns to the input data belonging to any of the classes. The class that is assigned the highest value is considered the prediction of the network for that input.

The network was implemented in python using the Keras library for neural network construction and training, PlaidML to enable running the training sequence on an AMD R9 270 graphics card, and some functions from scikit-learn to shuffle and divide the input data. The reason for reimplementing the network used in [1] and [2] was that the Matlab script there ran very slow. Running ANN models on a graphics card is quite a bit faster than running them on the CPU as the Matlab script does.

3.4.1 Training procedure

In order to be able to compare the effects of our modifications on the IMU data to the results of [2], the training procedure presented in that study is used. The training procedure consists of the following steps:

First, the feature set is randomly split into a training set containing 70% of the samples, a validation set containing 15% of the samples, and a test set containing the remaining 15% of the samples. The samples are distributed such that the proportion of classes in all sets is kept the same.

The network is trained on the training data. After every epoch, the accuracy of the network on the validation set is evaluated. If the validation set accuracy starts to drop, training is stopped.

The network is then evaluated on the test set. For every sample in the test set, the sample features are presented to the network, and the predicted class and the true class from the sample label are recorded. From these predictions and ground truth, the metrics discussed in 3.7 are calculated.

3.4.2 Per subject classifiers

The studies in [1] and [2] used one classifier per subject. For every subject, the features calculated from the data of that subject are collected. Then the training procedure is applied on a new ANN for every subject, using the data for only that subject.

In this study, we use that approach in order to properly compare the results of this study to procedure [1] and [2].

However, having to train a new classifier for a new subject involves extra work, as you need to do a calibration session for every new subject. Therefore, all the experiments were repeated using a single classifier trained using the data of all the subjects. If these models perform similar to the per-subject models, this would be a step in the direction of eliminating the need to train a new classifier for new subjects.

When training is completed, the test set is presented to the network. For every sample, the network processes the features. It will then assign a number to each output label corresponding to how likely the network believes the features to belong to that label; the higher the number, the higher the confidence of the model for a match of that class. The label with the highest number will be noted as the prediction. These predictions can then be compared with the known ground truth in order to calculate metrics such as accuracy.

3.5 Reaching frame

To adapt the orientation compensation from [3] to our use case, a “reaching frame” based on the direction of the reaching movement and gravity is defined. As the IMU data is in the global frame, the negative z axis is taken as the gravity axis. Then for every reaching event, Principal Component Analysis (PCA) is applied to the IMU acceleration data within the interval of that event. The first component of the PCA is the direction in which most of the acceleration is done. As the position of the arm is obtained by double integration of the acceleration, the first PCA component of acceleration is assumed to be close to the direction of movement [11]. This vector will be called the forward vector.

The reaching frame is then constructed: the cross product of the gravity vector with the forward vector will be called the auxiliary vector. Then the cross product of the auxiliary vector and the gravity vector is used to replace the forward vector, effectively removing any z-component from the forward vector. The magnitude of the forward vector, gravity vector, and auxiliary vector are set to 1, providing an orthonormal basis of the reaching frame. The reaching frame is now a rotation in the horizontal plane compared to the global frame. Finally, a rotation is calculated from the global frame to the reaching frame.

Before applying this rotation to the sensor data in the event interval, we extend the event interval to align with the start of the first segment and end of the last segment that are considered part of this event. After extending the event interval, the RFT is applied to the sensor data in the extended event interval. An example of the RFT can be seen in Figure 7. Features calculated from the segments in a reaching event use this rotated data. The features calculated for the other segments, which are all labeled "no event", use the original global frame data.

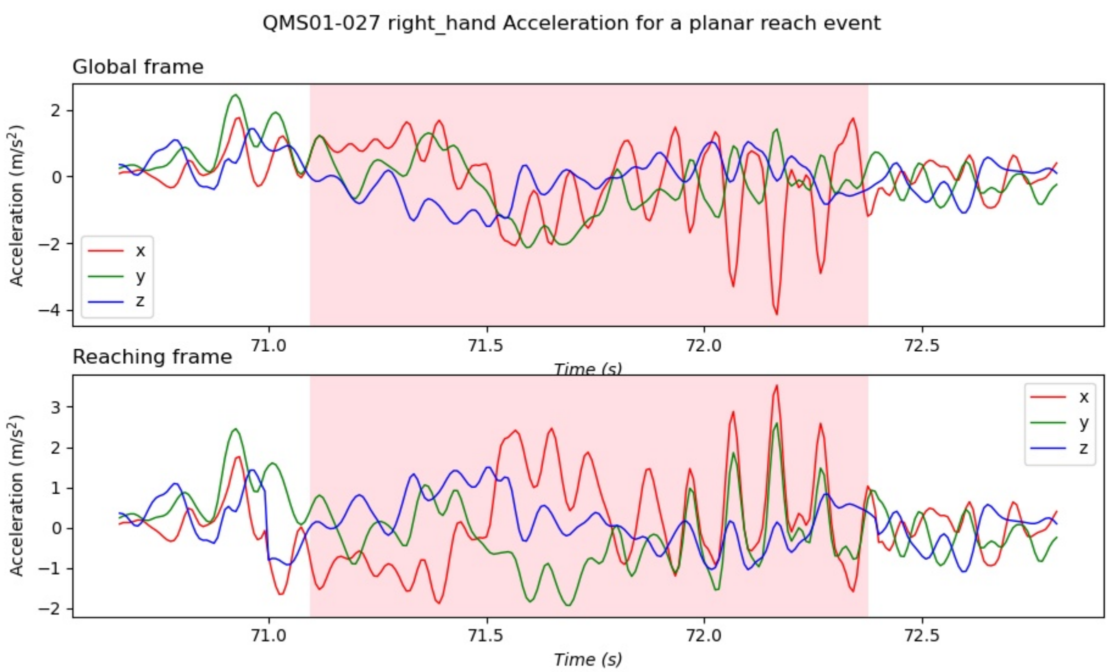


Figure 7: Acceleration data from an event in this study before and after RFT for a planar reach event. The shaded are is time interval of the event.

3.6 Horizontal plane rotation

As mentioned in 2.4.1, one of the limitations of the research in [1] and [2] was that the subjects all performing their tasks facing the same forward direction. In order to test the performance of the classifiers in a situation where subjects are not all seated in the same orientation in the horizontal plane, a random Horizontal Plane Rotation (HPR) was added to each experiment. When applying an HPR, a random angle is selected and all sensor data is rotated in the horizontal plane by that angle. This synthetically orients the subject in a different direction, allowing the evaluation of classifier performance on subjects doing tasks in different orientations.

3.7 Resulting metrics and comparison

After classification was done on the test set, the following metrics were calculated:

For every class, the F1 score will be calculated. The F1 score is defined as the harmonic mean of precision and recall: $2*\frac{precision*recall}{precision+recall}$.

Precision is defined as the ratio of True Positives (TP) and all positives (TP + False Positives(FP)):

$$\frac{TP}{TP+FP}$$

Recall is defined as the ratio between TP and all positive cases in the data, or equivalently, TP + False Negatives (FN):

$$\frac{TP}{TP+FN}$$

The accuracy, defined as $\frac{TP+TN}{TP+TN+FP+FN}$, is not used as the it includes the True Negatives (TN). The reason TN are problematic, is that the number of samples in the “no event” class is very large compared to the number of samples in all the other classes. If “no event” samples are identified correctly, then for all classes other than “no event”, this would result in an accuracy very close to 1 for all these labels, regardless of their TP, FP and FN results. For this reason, the F1 score is a better indicator of model performance in this context.

The macro average (MA) of the F1 score is also calculated: this is the mean of the F1 scores of all the classes in an experiment. The classes are all weighted evenly, as to not let the “no event” class distort the results too much.

Every experiment was ran 5 times. The average scores were noted.

3.8 Experiment design

This section describes the different settings used for every experiment. For each experiment, the metrics described in the previous section were calculated. The ANN classifiers were trained as described in 3.4.

Table 3 shows the settings used for every experiment for experiments using one classifier per subject. Experiment M0 is not an experiment done in this study, but contains the metrics described in 3.7 calculated from the table in Figure 9 of v. Gils [2]. With this data it is possible to compare the results from that study to the experiments done in this study.

Table 4 shows the experiment settings for the experiments where one classifier was trained using the data from all subjects per experiment.

Experiment M1 is essentially a replication of the experiment described in [2], apart from the different gradient optimizer used for the ANN.

Table 3: Overview of differences between the performed experiments using one classifier per subject (Multiple classifiers)

Experiment		Planar events	3D events	HPR	RFT
Experiment M0		x			
Experiment M1		x			
Experiment M2		x			x
Experiment M3		x		x	
Experiment M4		x		x	x
Experiment M5		x	x		
Experiment M6		x	x		x
Experiment M7		x	x	x	
Experiment M8		x	x	x	x

Table 4: Overview of differences between the performed experiments using a single classifier for all subjects (Single classifier)

Experiment		Planar events	3D events	HPR	RFT
Experiment S1		x			
Experiment S2		x			x
Experiment S3		x		x	
Experiment S4		x		x	x
Experiment S5		x	x		
Experiment S6		x	x		x
Experiment S7		x	x	x	
Experiment S8		x	x	x	x

4 Results

Table 5 and Table 6 show the results of the experiments described in 3.8.

Multiple classifiers (One per subject)	Macro average	Planar reach left	Planar reach right	No event	3D reach left	3D reach right
Exp M0 (v Gils)	0.90	0.83	0.85	0.98		
Exp M1 (Control, No HPR, No RFT)	0.78	0.66	0.72	0.96		
Exp M2 (RFT)	0.83	0.72	0.79	0.97		
Exp M3 (HPR)	0.71	0.58	0.57	0.96		
Exp M4 (HPR, RFT)	0.68	0.55	0.53	0.95		
Single classifier (One for all subjects)						
Exp S1 (No HPR, No RFT)	0.77	0.70	0.65	0.96		
Exp S2 (RFT)	0.81	0.72	0.73	0.97		
Exp S3 (HPR)	0.75	0.66	0.62	0.96		
Exp S4 (HPR, RFT)	0.73	0.58	0.64	0.96		

Table 5: F1 scores of the experiments with planar events only

Multiple classifiers (One per subject)	Macro average	Planar reach left	Planar reach right	No event	3D reach left	3D reach right
Exp M5 (No HPR, No RFT)	0.52	0.72	0.57	0.96	0	0.33
Exp M6 (RFT)	0.47	0.64	0.64	0.96	0	0.11
Exp M7 (HPR)	0.36	0.39	0.36	0.95	0	0.11
Exp M8 (HPR, RFT)	0.36	0.46	0.35	0.95	0.06	0
Single classifier (One for all subjects)						
Exp S5 (No HPR, No RFT)	0.53	0.67	0.69	0.96	0.35	0
Exp S6 (RFT)	0.66	0.76	0.68	0.97	0.46	0.43
Exp S7 (HPR)	0.47	0.60	0.44	0.95	0	0.38
Exp S8 (HPR, RFT)	0.64	0.60	0.61	0.96	0.43	0.60

Table 6: F1 scores of the experiments with planar and 3D events

Compared experiments	Planar reach left	Planar reach right
M5-M1	0.06	−0.15
M6-M2	−0.08	−0.15
M7-M3	−0.19	−0.21
M8-M4	−0.09	−0.18

Table 7: Differences of F1 scores Between M5-M8 and M1-M4

The results of the replication experiment M1 are far worse compared to the results of M0, which is calculated from data in[2]. The F1-scores of the planar reach events in M1 are at least 0.13 less than those in M0. As want to know the relative effectiveness of applying HPR, RFT, and/or using a multi-subject classifier compared to the control, we will compare the results from the experiments M2-M8 and S1-S8 with M1, our attempted replication of M0 from [2].

First we'll discuss experiments M1-M4. Introducing the RFT on the data in M2 seem to slightly increase performance, as we expected. The F1-scores in M2 for planar reach events is at least 0.06 higher than the scores in M1. Adding HPR in M3 decreases the F1-scores by at least 0.08 for reaching events, which shows some dependency of the recognition algorithm on reaching events taking place in the same orientation. The combination of RFT and HPR in M4 does slightly worse (a 0.03 decrease in F1-score compared to M3) than applying only HPR. This is unexpected: the point of the RFT is to realign all events to the reaching direction. As RFT data in M2 has a small positive effect on non-rotated data in M1, it was expected to solve the negative performance impact of the random rotations of the HPR, but it does not.

The experiments S1-S4, using a single classifier, show a similar pattern. Application of the RFT alone in S2 improves the F1 scores for planar reaching events by at least 0.02 compared to S1. Applying HPR in experiment S3 decreases the F1 scores for planar reaching events by at least 0.03 compared to S1, and HPR + RFT in S4 decreases the F1 scores for planar reaching events by at least 0.02 than HPR (S3) alone. Comparing M1-M4 with S1-S4 respectively, we see very similar performance. A single classifier trained on all patients thus seems to result in similar performance as per-patient trained classifiers.

Next we will look at the experiments including 3D reach events.

As can be seen in Table 7, experiments M5-M8 perform worse than M1-M4 for planar reaching, decreasing the F1-scores for planar reaching by at least 0.08, except the planar reach left for M5. Comparing M6 to M5, we see a 0.08 decrease in F1 score for planar reach left, but a 0.07 increase in F1 score for planar reach left. Applying the RFT is thus not strictly better or worse when 3D events are included. Applying only HPR in M7 reduces the F1 score by at least 0.21 compared to M5 for planar events, which is a lot more than the reduction we saw when comparing M3 to M1. Applying both RFT and HPR in M8 increases the F1 score for left planar reach events by 0.07 compared to M7, and decreases the F1 score for right planar reach events by 0.01. While not strictly better, M8, with an increase of 0.06 together with a decrease of only 0.01 could be considered slightly better than M7.

3D reaching event recognition is abysmal across the board for M5-M8. Left reach 3D has an F1 score of 0, except in experiment M8, where it reaches a whopping 0.06. Reach right 3D does less poorly; it has a top F1 score of 0.33, though it also reaches 0, in experiment M8.

The single classifier method S5-S8 have better performance than their M5-M8 counterparts. For planar reach events, we see a similar pattern to the experiment groups M1-M4, M5-M8 and S1-S4: experiment S6, with the RFT, slightly improves results compared to S5 with an increase of the F1 score for planar reach events of 0.09 and −0.01. Just like M8 vs M7, this is not a strictly better result, but it is still considered slightly better as the decrease in performance is so small. S7, with HPR reduces the F1 scores by 0.07 and 0.25 for left and right planar reach respectively compared to the base experiment S5, while S8 improves performance compared to S7 by increasing the F1 score for right planar reach by 0.17.

For S5-S8, 3D reach events also perform quite poor, though better than their M5-M8 counterparts. The F1 scores for S5-S8 are strictly better than M5-M8 by at least 0.32. An exception is for 3D reach left in S7 and M7, where the F1 score is equal to 0 for both, and for M5 and S5, where the 3D reach right for M5 has an F1 score that is 0.33 higher than S5.

For S5-S8, we can also see that applying RFT (S6, S8) has strictly better results than the experiments without RFT (S5, S7).

The effects of the RFT for planar reaching events are summarized in the table below.

M2 vs M1: increase F1 score by at least 0.06
M4 vs M3: decrease F1 score by at least 0.03
S2 vs S1: increase F1 score by at least 0.02
S4 vs S3: decrease F1 score by at least 0.02
M6 vs M5: inconclusive
M8 vs M7: increase F1 score by 0.08, decrease by 0.01
S6 vs S5: increase F1 score by 0.09, decrease by 0.01
S8 vs S7: increase F1 score by 0.17 or equal

Table 8: Effects of RFT compared to partner experiment without RFT. Green means a better results, red is a worse result, grey is inconclusive

The effects of the HPR for planar reaching events are summarized in the table below.

M3 vs M1: decrease F1 score by at least 0.08
M7 vs M5: decrease F1 score by at least 0.21
S3 vs S1: decrease F1 score by at least 0.03
S7 vs S5: decrease F1 score by at least 0.07

Table 9: Effects of HPR-only experiments compared to the control experiments. Red means a worse result.

5 Conclusion

As can be seen in Table 8, the RFT has a modest positive effect most of the time for planar reaching events. Applying HPR worsens performance in all cases as can be seen in Table 9, which shows that having similar reaching directions for a category is important for the performance of the system. This was expected; however, it was also expected that the RFT would remedy the random rotation introduced by the HPR by reorienting the IMU data of randomly rotated events to a common forward orientation. The results show that this is not the case. This is important to note, as the intended use case, ADL in a home situation, will in general not be performed in the same forward orientation.

3D reaching events have very poor F1 scores. For the experiments with one classifier per patient, it is lower than 0.33 in all cases, and 0 in half the cases. Results for 3D reaching of the single classifiers are a lot better: the F1 score is 0 in 2 cases, and larger than 0.35 in the other cases. Perhaps the reason for this is that the amount of 3D reach events is rather small. The single classifier combines the reaching of all patients for every reaching category, resulting in more data for the classifier to train on.

For planar reaching events, the series of experiments with a single classifier for all patients shows comparable results to the classifier-per-patient experiments.

6 Discussion

It was expected that results from M1 would be nearly identical to the results of M0, which is not the case. The reason for this is unclear; it might be different weight adjustment algorithm used in M1, as the algorithm used in the Matlab script from M0, “trainscg” (scaled conjugate gradient backpropagation) was unavailable in the keras library used for the experiments in this thesis.

The most unexpected results from this study is that the RFT does not seem to help reorienting reaching movements. Study [3] shows that reorienting movements to a common frame can have an effect. The reason why reorienting using PCA does not work in this study is unclear, and it might be worth looking into the reason why it does not work as expected. It might be that the assumption “direction of greatest acceleration is the direction of movement” is not correct. Maybe a different method of determining a forward direction vector is better suited, for example the method used in [1], that determines direction by estimating the start and ending position of a reaching event.

The machine learning method used in this study, a memory-less perpectron, might not be the optimal architecture for the job at hand. Mechanical movement of human limbs is a continous time process where acceleration at some time is constrained by what happened before; not all movements are possible at any moment. For every event, we calculate some features for several time windows in the event and then set a point in N-dimensional space. There is some coupling between the “past” of a movement and the current feature-set, as the features are calculated over a window that overlaps with the previous feature set; but I believe there are other architectures that will take better advantage of the constraints in human movement that is present in the IMU data.

7 References

[1] M. Esfahani 2019; Minimal sensing of quality of upper-limb movements

[2] H. van Gils 2020; Comparison between dynamic time warping and artificial neural networks for recognizing reaching events in post stroke subjects

[3] F. Dransfeld 2019; Transforming IMU Data To Global Frame For Improving Volleyball Action Recognition

[4] G. Kwakkel, B.J. Kollen, J. van der Grond, A.J.H Prevo 2003; Probability of Regaining Dexterity in the Flaccid Upper Limb

[5] F.B. van Meulen, B. Klaassen, J. Held, J. Reenalda, J.H. Buurke, B.J.F van Beijnum, A. Luft, P.H. Veltink; 2016, Objective Evaluation of the Quality of Movement in Daily Life after Stroke

[6] INTERACTION project news item; 2011 - 2014 - European project - Interaction - News 2014 - body sensor system | BSS: Biomedical Signals and Systems (utwente.nl)

[7] A. Tognetti, F. Lorussi, N. Carbonaro, D. De Rossi, G. De Toma, C. Mancuso, R. Paradiso, H. Luinge, J. Reenalda, E. Droog, P.H. Veltin 2014; Daily-life monitoring of stroke survivors motor performance: The INTERACTION sensing system

[8] https://icd.who.int/dev11/l-icf/en#/http://id.who.int/icd/entity/7 630 894

[9] F.B. van Meulen, B.J.F. van Beijnum, J.H. Burke, P.H. Veltink 2017; Assessment of Lower Arm Movements Using One Inertial Sensor

[10] https://www.xsens.com/hubfs/3 446 270/Downloads/Manuals/MTwAwinda_WhitePaper.pdf

[11] K. Kunze; P. Lukowicz; K. Partridge; B. Begole 2009; Which Way Am I Facing: Inferring Horizontal Device Orientation from an Accelerometer Signal

[12] Diederik P. Kingma, Jimmy Ba 2014; https://arxiv.org/abs/1412.6980

[13] Image source: https://shop.movella.com/thumbnail/39/d9/44/1 675 176 852/MVN%20Analyze%20-%20Awinda_1920x1920.jpg