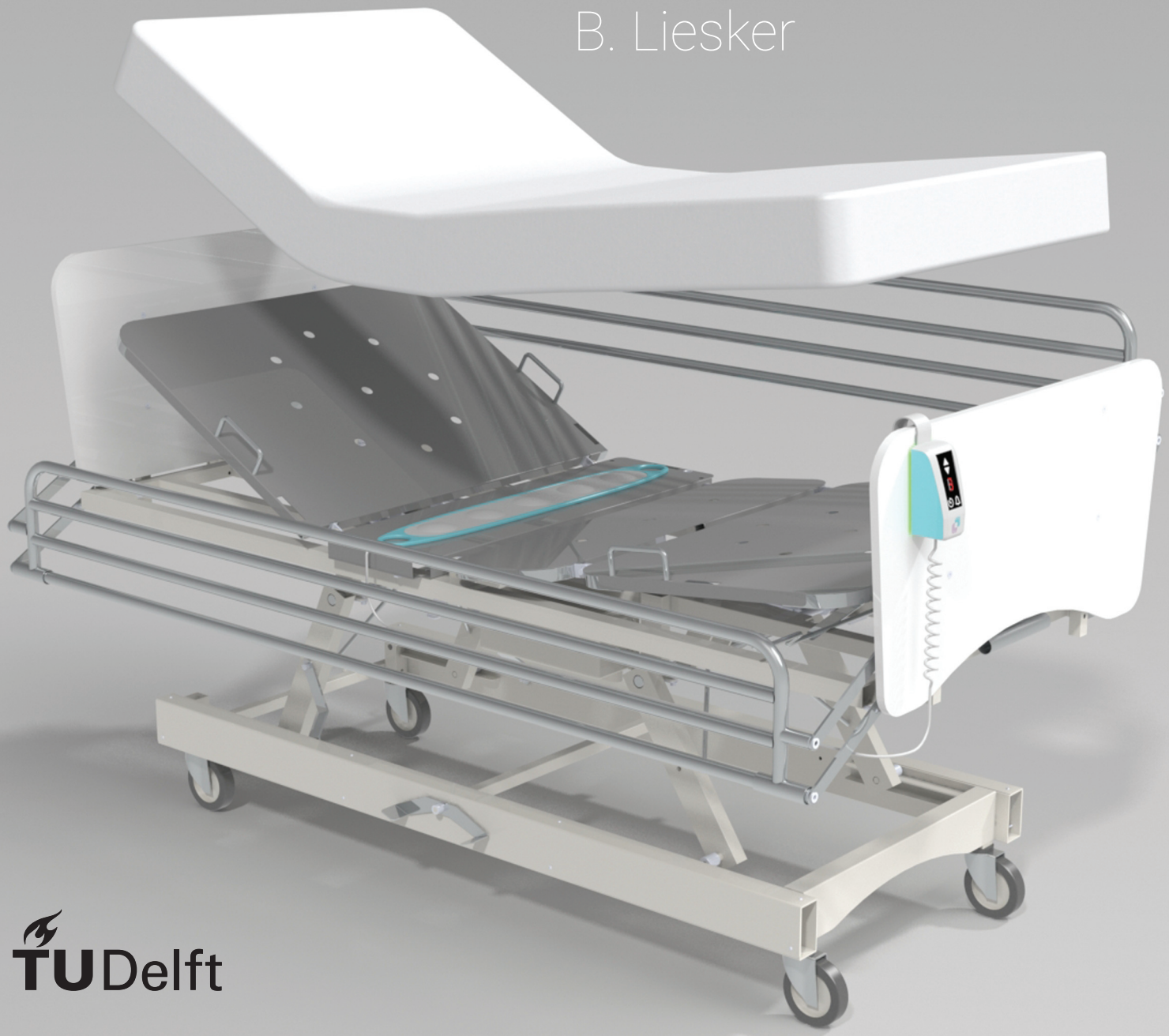
Momo Medical

# Non-intrusive patient monitoring to prevent pressure ulcers

## Algorithm Subgroup

D. Brinkhorst
B. Liesker



Technische Universiteit Delft

TU Delft

# Non-intrusive patient monitoring to prevent pressure ulcers

## Algorithm Subgroup

by

D. Brinkhorst
B. Liesker

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended on Wednesday July 5, 2017 at 11:00 AM.

| | | |
|---|---|---|
| Student numbers: | 4323742 and 4195590 | |
| Date: | July 12, 2017 | |
| Project duration: | April 24, 2017 – July 7, 2017 | |
| Thesis committee: | Prof. dr. ir. G. J. T. Leus, | TU Delft, Supervisor |
| | Dr. N. Llombart, | TU Delft, Chair |
| | Dr. A. R. Mor, | TU Delft, Jury Member |
| | M. L. Gravemaker, | Momo Medical, BAP Proposer |

*This thesis is confidential and cannot be made public until July 5, 2022.*



TU Delft

# Executive summary

The founders of Momo Medical envisioned a health care product that would help nurses worldwide with pressure ulcer prevention. Pressure ulcers are a chronic wound that affects the skin of patients who do not regularly change bed posture. As it currently stands, nurses lack the manpower and time to change the posture of patients in a timely manor. Thus, a goal was set to create a product that would reduce this strain on nurses.

The goal was to design a non-intrusive product that keeps track of the posture of a patient on a hospital bed, and send an alarm if this has not changed in a set amount of time. A constraint with this was that the product has to lay directly underneath the mattress of the patient. Furthermore, the product has to make use of as few piezoresistive and piezoelectric sensors as possible. With these constrains, the project group was divided in three subgroups: the sensor, the algorithm, and the testing subgroup.

This report will focus on the design, implementation, and evaluation of the algorithm. The algorithm has to be designed to process the piezoresistive and piezoelectric sensor data and return, with 90% accuracy, the posture of the patient. It also has to be able to determine with 99% accuracy whether or not there is a patient on the bed. An optional objective was to design an algorithm to determine the heart and respiration rate of the patient, with 80% accuracy, but this was of lower priority. It was soon decided to give priority to posture detection and not further investigate detection of the heart and respiration rate.

In the final product, the information about the patient's posture will be displayed in a graphical user interface (GUI). This GUI has to be user-friendly and intuitive to use by the nurses. During the prototyping phase, however, a different GUI was used, one that displayed various debugging information. This prototype GUI was necessary to test the various algorithms.

Several different mathematical models have been investigated, implemented, and tested. These models range from taking the variance of the piezoresistive data, to deploying a Fourier transform on a polynomial curve based on the sensor data. With preliminary test data, the neural network achieved 93% accuracy, with the algorithm based on sensorwise variance almost making the design requirements, at 88.7%. The Fourier transform-based algorithm achieved 77.4% accuracy, and the second derivative-based method 75.4%. With the final testing data however, none of the methods got an accuracy higher than 72.2%, except for one algorithm, a neural network.

In the end, only the neural network reaches the required accuracy of 90%, though only just, at 90.8%. With future iterations of the algorithm, more research and data collection is advised, as neural networks work better with larger data sets.

# Contents

**Executive summary** iii

**1 Problem description** 1
- 1.1 Problem scope . . . 1
- 1.2 Technical review . . . 1
  - 1.2.1 Background of the field . . . 2
  - 1.2.2 State-of-the-art analysis . . . 2
- 1.3 Project overview . . . 3
- 1.4 Design requirements . . . 3

**2 Design** 5
- 2.1 Data preprocessing . . . 5
- 2.2 Patient detection . . . 5
- 2.3 Posture detection . . . 7
  - 2.3.1 Problem description . . . 7
  - 2.3.2 Approach . . . 7
  - 2.3.3 Signal transformations . . . 7
    - 2.3.3.1 Raw data . . . 9
    - 2.3.3.2 Polynomial fit . . . 10
    - 2.3.3.3 Second derivative . . . 10
    - 2.3.3.4 Fourier analysis . . . 10
    - 2.3.3.5 Variance . . . 12
  - 2.3.4 Decision making . . . 12
    - 2.3.4.1 Threshold based on distance to mean . . . 12
    - 2.3.4.2 Neural networks . . . 13
    - 2.3.4.3 Posture change detection . . . 14
- 2.4 Heart rate and respiration rate sensing . . . 14

**3 Implementation** 15
- 3.1 Overview . . . 15
- 3.2 Data acquisition . . . 15
- 3.3 Algorithm software . . . 15
- 3.4 GUI . . . 16

**4 Evaluation** 17
- 4.1 Overview . . . 17
- 4.2 Data collection . . . 17
- 4.3 Testing and results . . . 18
  - 4.3.1 Patient detection . . . 18
  - 4.3.2 Posture detection . . . 18

**5 Conclusion** 21
- 5.1 Assessment . . . 21
  - 5.1.1 Human input constraints . . . 21
  - 5.1.2 Patient detection . . . 21
  - 5.1.3 Posture detection . . . 21
- 5.2 Recommendations . . . 21

**A Datasets** **23**

**B Reference signals** **25**

**C Posture change** **27**

**D Old results** **29**
D.1 . . . . . 29

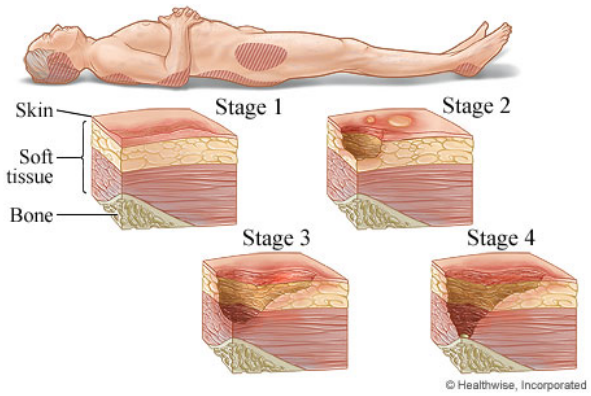
**E Code** **31**
E.1 Data acquisition . . . . . 31
E.2 Matlab example . . . . . 32

**Bibliography** **33**

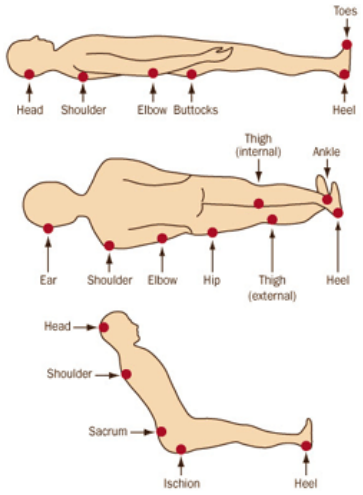
# 1 Problem description

## 1.1 Problem scope

Pressure ulcers (also known as bedsores, or medically as decubitus ulcers) are a medical condition that affects patients worldwide. By sitting or laying with the same posture for an extended period of time (two or more hours, depending on the health condition of the patient), pressure ulcers start forming on the skin layer. If these ulcers are left untreated, the pressure ulcer wound expands to the soft tissue layer (see figure 1.1a). Particularly, the pressure points, as indicated in figure 1.1b, are susceptible to formation of such chronic wounds.




(a) The four stages of pressure ulcers [1].




(b) Sensative pressure points are indicated with a red dot [2].

Figure 1.1: The formation of pressure ulcers.

In the Netherlands alone, it is estimated that the treatment of pressure ulcers costs between 362 million to 2.8 billion U.S. dollar every years, which is around one percent of the Dutch health care budget [3]. This problem also plagues the United Kingdom, where, according to Posnett and Franks (2008), one in five British hospital patients are estimated to have had pressure ulcers [4].

Despite training of hospital staff in pressure ulcer prevention, the problem continues to affect patients worldwide, mainly due to lack of time and manpower of nurses [5]. This information makes clear that a solution needs to be found for this problem.

## 1.2 Technical review

The problem scope provided a short overview of the effects of pressure ulcers. This part focusses on the causes and the state of the art concerning the prevention of pressure ulcers.

### 1.2.1 Background of the field

Pressure ulcers are caused by many factors. This project focusses on three factors related to forces, that are exposed to the skin. The first factor is pressure. Continuous pressure on the same place of the skin cuts off the blood vessels and decreasing the oxygen and nutrient supply to the cells under the skin, which is damaging to the cells and increases the chance of formation of pressure ulcers. Pressure is the main reason that patients who are immobilized or can not move very well without help get pressure ulcers. The second factor is friction. Friction between the skin and the patient's clothes or bedding damages the skin and can cause pressure ulcers. The last factor is shear. This occurs when the skin moves in an opposite direction compared to the underlying bone or muscle tissue. This may cause stretching and even breaking of cells and blood vessels [6].

There are several positions that can minimize pressure while lying down [7]. In case patients are lying on their back, they should lie flat on their back or preferably lie in a 30 degree Semi-Fowler Position (see figure 1.2). Pressure and friction will be minimized in this position. In case the patients are lying on their sides they should preferably lie in a 30 degree lateral tilt position (see figure 1.3). This technique involves turning a patient to his/her side, and placing supportive pillows behind their back. Although this technique is simple, it is very effective, though time-consuming for nursing staff at the same time.

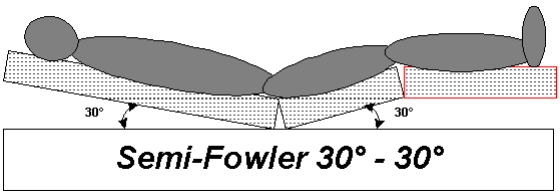



Figure 1.2: 30 degree Semi-Fowler Position [7].

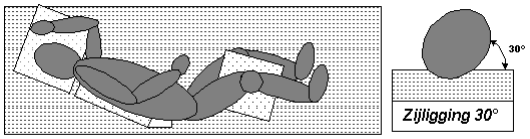



Figure 1.3: 30 degree lateral tilt Position [7].

### 1.2.2 State-of-the-art analysis

The technique for preventing pressure ulcers is basically very simple; by turning the patient regularly (for example, every four hours [8]) from one position to another, the active pressure points are shifted, thus preventing pressure ulcer forming. This is called *wisselligging* in Dutch. As it currently stands, this is the most common preventive measure [9]. In practice however, turning every patient individually at the hospital ward is a time-consuming task. Continuous tracking of the patient's body position can offer a technical solution for preventing pressure ulcers. With tracking information, the patient only needs to be turned around if the patient did not turn by himself. As a result, the patient would not need to be disturbed by unnecessary routine shifts and is much better protected from pressure ulcers [10]. One of these tracking systems is the BodiTrak [11]. This is a mattress cover containing thousands of pressure sensors that create a heat map of the pressure distribution. Pressure sensor mats which consist of an array of flexible force sensors [12] and continuous pressure mapping devices [13] are similar technical solutions, specifically developed and deployed to support pressure ulcer prevention. While very effective in tracking the patient's body position, they are very expensive, up to several thousands of euros.

Besides these mattress covers, also contact-free solutions exist, such as a system with a few piezo sensors placed beneath the mattress. Although not specifically developed for preventing pressure ulcers, such a system is capable of tracking patient's body position [14]. However, this system is also very expensive. Another system consists of high-resolution force sensors at each of the four bedposts [15]. This system is not ideal to use in a hospital setting where the beds have to be mobile. A solution is to build these sensors inside the bed frame, which would require hospitals to replace beds, a very expensive operation.

## 1.3 Project overview

This final bachelor project is an assignment given by the startup Momo Medical to use their existing ideas and technology to create a device that lowers the incidence of pressure ulcers due to continuous pressure in hospitals and other care institutions. During this final bachelor project the group was divided into three subgroups.

1. Sensors:
   The creation of a non-intrusive position monitoring system that collects data of the patient in the bed.

2. Algorithm:
   A piece of software that can determine the posture of the patient in bed using the data obtained by the sensors.

3. Test setup:
   A system able to verify the results of the algorithm group with a high degree of accuracy.

The global overview of the project is shown in figure 1.4, this diagram illustrates the different parts. It also shows the areas where cooperation between the different groups is required.

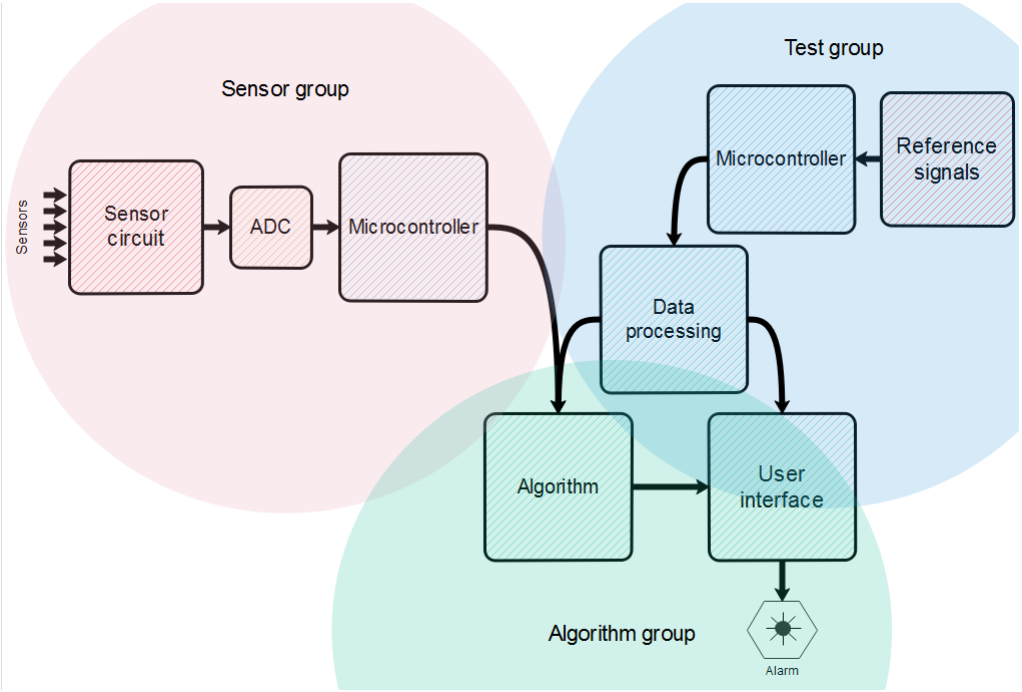



Figure 1.4: Project overview diagram

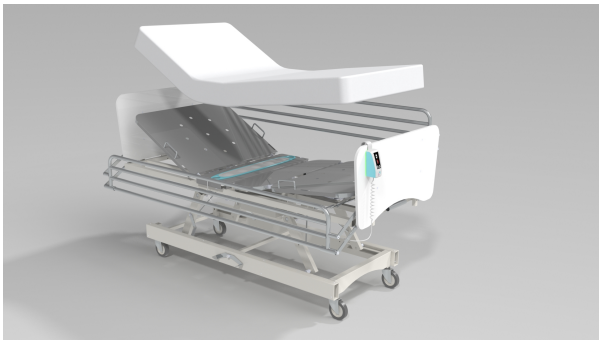
## 1.4 Design requirements

The main product is an algorithm that can determine the posture of a person lying in a hospital bed in real-time. The available inputs for the algorithm(s) are 5 piezoresisitve and 2 piezoelectric sensors, located under the mattress of the bed. Figure 1.5 shows a render of the bed, together with the placement of the sensor mat. The design requirements are as follows:

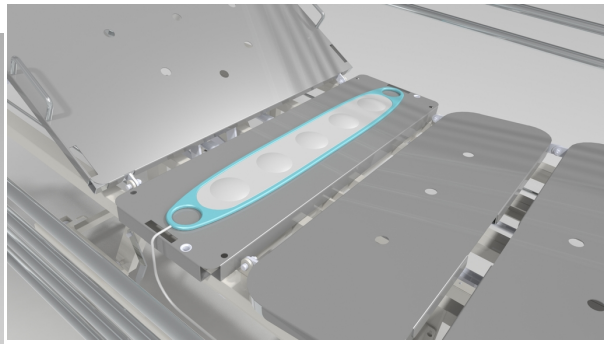
- The algorithm should operate without requiring human input. The only exception is the entering of patient information, such as weight and height.
- The algorithm must detect whether there is a person on the bed or not. This decision must be correct 99% of the time.
- The algorithm must distinguish the following postures. This decisions must be correct 90% of the time.

  - The person is on his/her back.
  - The person is on his/her left side.
  - The person is on his/her right side.

- (Optional) An algorithm that detects the heart rate of the person. 80% of the time, it must be within 10% deviation from the reference heart rate.
- (Optional) An algorithm that determines the respiration rate of the person. 80% of the time, it must be within 10% deviation from the reference respiration rate.



(a) A render of a hospital bed with the sensor mat underneath it.



(b) The sensor mat with 5 piezoresistive and 2 piezoelectric sensors. Under each dome is a piezoresistive sensor, with the second and fourth dome also containing a piezoelectric sensor.

Figure 1.5: Renders of the sensor mat and its position.

# 2

# Design

An algorithm has been designed that processes the raw sensor data, and return several key pieces of information about the patient. The input consists of seven sensor signals, five of which are piezoresistive sensors, and two piezoelectric sensors. The piezoresistive sensors return a potential difference between zero and 3.3 volts, while the piezoelectric sensors return a potential difference between zero and 3.6 volts. The algorithm processes this, and returns three pieces of information: whether or not there is a patient on the bed, if the patient is moving, and with which posture the patient is on the bed. Figure 2.1 gives an overview of the system. The output is then displayed on a graphical user interface (GUI).

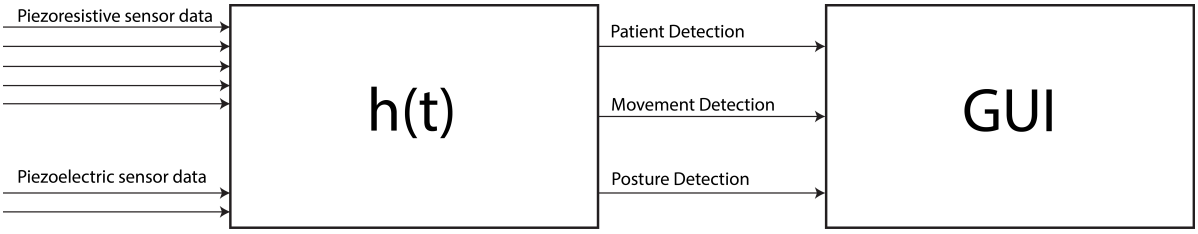



Figure 2.1: Overview of the algorithm system. h(t) is the algorithm.

## 2.1 Data preprocessing

Several hardware inconsistencies such as the form and size of the domes (which lay on top of the sensors) lead to differences in measured potential difference between the piezoresistive sensors when applying the same downward force (weight) to each sensor (see table 2.1). According to the datasheet, the sensors are linear within ± 3% [16]. However, in testing this margin was exceeded. Testing of the sensors has indicated that the potential difference scales quadratically. This is shown in figure 2.2 which plots the data from table 2.2. In the plot, the 2nd-order polynomial fit has a root mean squared error (RMSE) of 0.09 while the linear fit has a RMSE of 0.13.

With the quadratic nature of the sensors now known, the sensors have to have relative scaling instead of absolute scaling. The percentage with which a sensor value needs to be increased or decreased is listed in the fourth column of table 2.1. After this relative offset is applied, the potential difference of each sensor is normalised in order to ensure that they stay within the same range (between zero and one). The reason this scaling was necessary, was that in the prototype, the domes were not consistent in size, causing the different sensors to respond differently to the same mass put on top of them.

## 2.2 Patient detection

Before determining the posture of a patient, it is crucial to detect if a patient is actually present. Without this, the system would constantly send a false alarm that the patient (who is not present) has not recently changed posture. A small part of the algorithm has thus been allocated for this task.

| Sensor | Potential Difference [V] (± 0.05) | Offset [V] (± 0.05) | Offset [%] (± 3) |
|---|---|---|---|
| 1 | 1.36 | 0.34 | 25 |
| 2 | 1.74 | -0.04 | -2 |
| 3 | 1.16 | 0.54 | 47 |
| 4 | 2.29 | -0.59 | -26 |
| 5 | 1.95 | -0.25 | -13 |

Table 2.1: Measured potential difference for each piezoresistive sensor under the same mass (659 grams ± 1 gram) before adjustments. The fourth column represents the relative offset needed to equalise the behaviour of each sensor at this specific point (weight). After applying the offset and normalising the sensor values, they all assume the normalised value of 0.52 ± 0.03.

| Mass [g] | Potential Difference [V] (± 0.05) |
|---|---|
| 0 | 0.02 |
| 79 | 0.13 |
| 145 | 0.20 |
| 224 | 0.31 |
| 276 | 0.54 |
| 331 | 0.60 |
| 411 | 0.63 |
| 475 | 0.71 |
| 527 | 0.71 |
| 607 | 1.05 |
| 659 | 1.19 |
| 751 | 1.61 |
| 803 | 1.58 |
| 882 | 1.87 |

Table 2.2: Measured potential difference for a piezoresistive sensor with various applied masses.

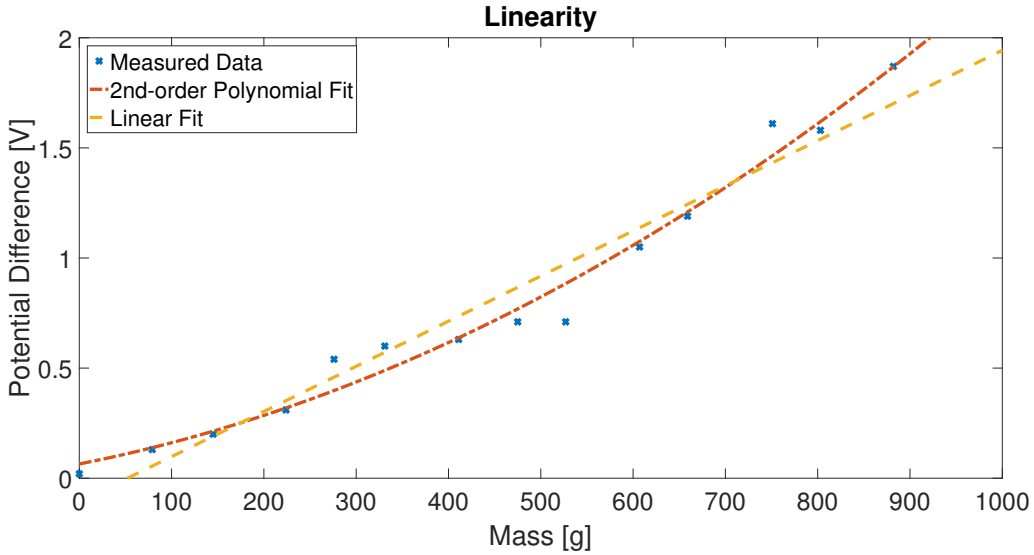



Figure 2.2: The measured data from table 2.1 is plotted together with a linear and a quadratic fit. The root mean squared error (RMSE) of the qudratic fit is 0.09 while the linear fit has a RMSE of 0.13.

The algorithm has a calibration function. It takes a sample of the five piezoresistive sensors and determines the mean. A small, constant value is added to this, and the combined value is set as a threshold. Now, if at any point the mean of the five piezoresistive sensors raises above this threshold, the patient detection Boolean is set to true. In every other case, this Boolean is set to false.

## 2.3 Posture detection

### 2.3.1 Problem description

Many factors of the patient's position had an effect on the sensor values. The posture of a patient (left side, right side, back, stomach) can be determined from the roll angle, as was shown by the test group. Unfortunately, this roll angle can not be directly measured by the pressure sensors under the mattress. Figure 2.3 shows the histograms of piezoresistive data collected for the 4-sensor setup (dataset 1.1 through 1.3. See also appendix A). These histograms show the difference in sensor values between a person on his back and a person on his side. It can be seen that when the person was on his side, the two middle sensors often have a larger value. An explanation for this is that when someone was on his/her side, the pressure was more focused on these sensors, instead of spread out. This difference was the focus in finding ways to detect the patient's posture.

It was expected that the difference in left side and right side would be smaller than that between back and side. Figure 2.4 shows the histogram of piezoresistive sensor 2, but now the data is split between the person being on his left side and on his right side. It supports the hypothesis: there is significant of overlap.

Other parts of patient position were not of interest, but still influenced sensor values. The lateral position of the patient had the largest impact. Having a person on the left side of the bed will result in very different sensor values than having the same person on the right side of the bed, even if the person is on his/her back in both cases. The position along the length of the bed was not expected to change much (due to a fixed pillow location). For people of different lengths, it was observed that for some, their lower back was on the sensors, while for others, it was their bottom. The second case resulted in larger pressure than the first. Also, the patient's weight influenced the magnitude of the sensor readout. To remove these influences, it was desirable to find a signal transformation that was less sensitive to them than the raw data.

### 2.3.2 Approach

The two most important situations to consider are the following:

- steady-state: patient is not moving. This situation was expected to be the most frequent in real-life situation.
- transient: patient is moving, possibly to a different posture. This was expected to be infrequent and short in duration.

The decision was made to not account for transient data. While it may be possible in to determine the change in posture from this data, the amount of ways a person could turn in bed, or be turned, was thought to be massive - with variations in timing, position and movement. To properly account for that, a massive amount of data would be needed - beyond the scope of this project. Furthermore, designing an algorithm that could handle all those situations would be too large an undertaking.

The main approach to determining posture thus only used a single (the most recent) time sample. An alternative way of doing steady-state analysis was to consider a series of time samples, perhaps by averaging over a longer time period. However, analysing a longer time window meant the influence of movement also persisted for longer. Since in testing, the time between a person turning was very short (to keep testing reasonably fast), analysing a larger time window was thought to do more harm than good. For this reason, only single-sample analysis was used during this project and in most prototypes.

Both piezoresistive and piezoelectric sensors were available. Piezoresistive sensors are designed to handle a constant load (weight), while piezoelectric sensors are designed to detect changes in applied pressure. Due to this fundamental nature of the piezoelectric sensors, and the need to asses steady-state situations, they were not used in the design.

As seen in figure 2.3 and 2.4 in subsection 2.3.1, the difference between back and side is larger than that between left side and right side. For this reason, the focus was put on reliably deciding between the back and side positions. Determining the difference between left side and right side, as well as the difference between back and stomach, would only be looked into once the design requirement was reached for distinguishing back and side.

### 2.3.3 Signal transformations

In determining patient posture, it may be helpful to first perform a transformation on the signal. A signal transformation may be able to remove one or more undesired influences of the patient position, while leaving

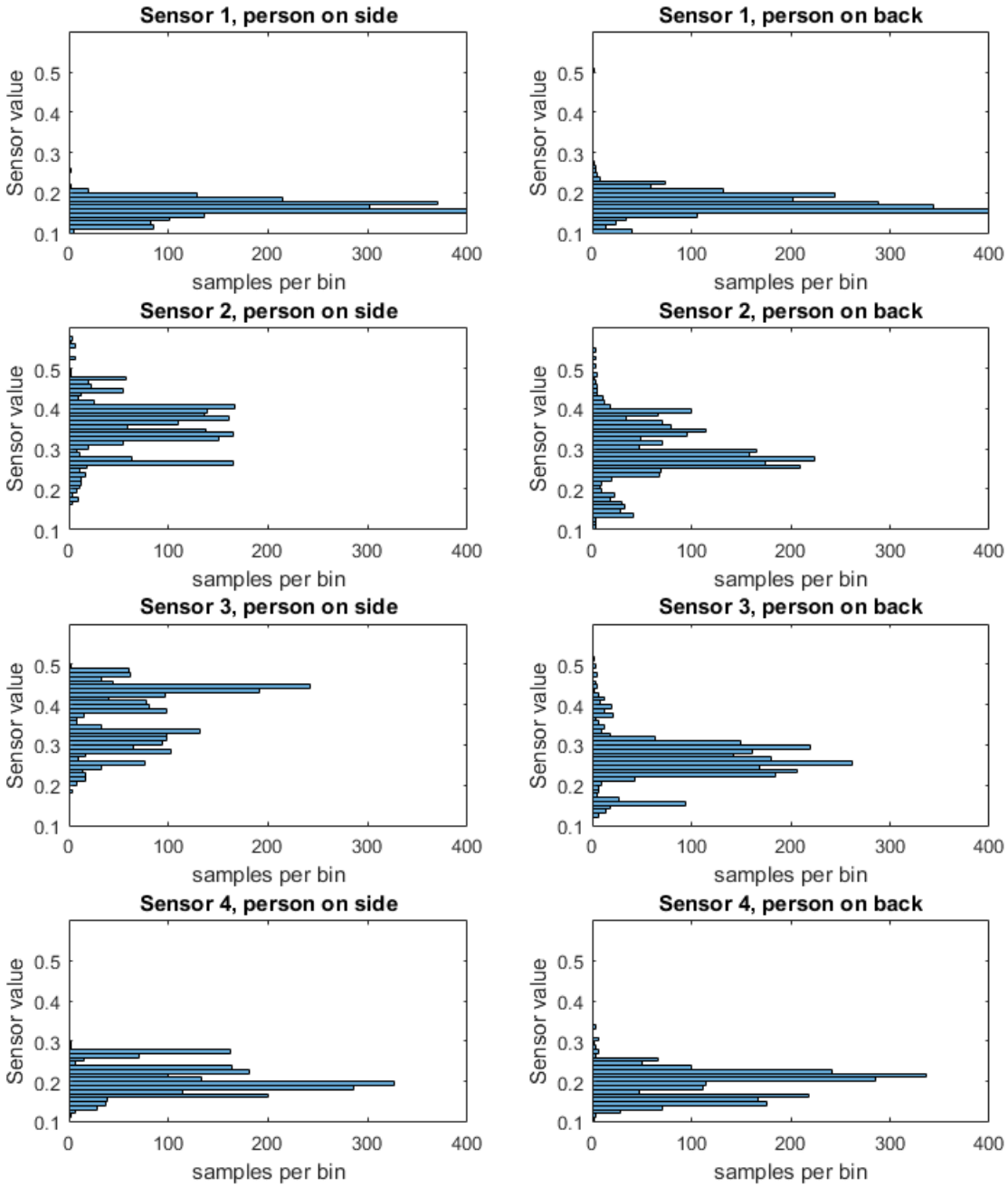



Figure 2.3: histogram of sensor data, The data is split between patient on back and on side.

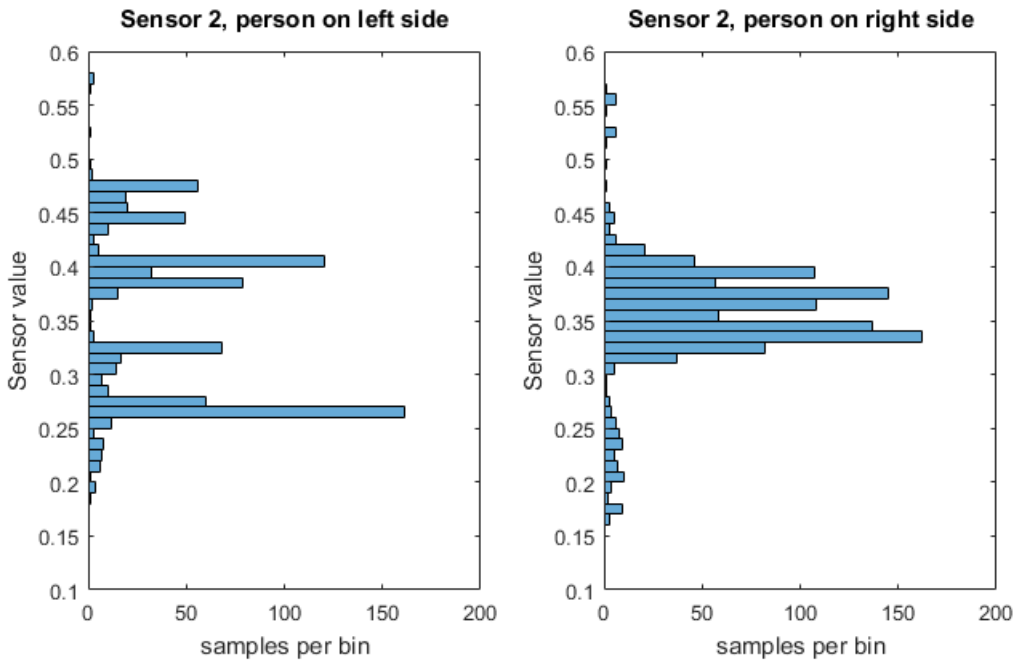


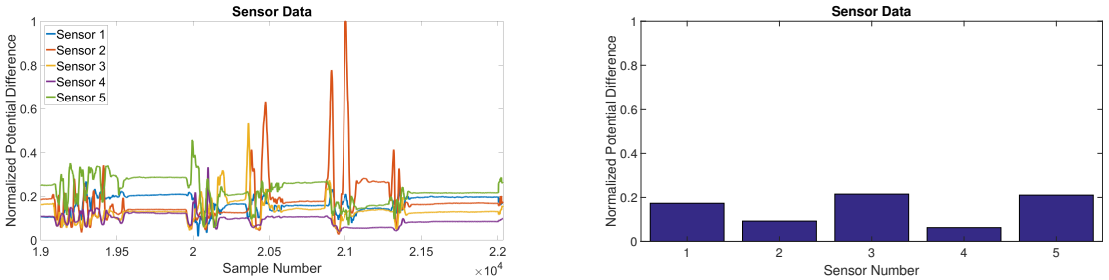

Figure 2.4: histogram of sensor 2. On the left, the patient posture was his left side. On the right, the patient posture was his right side.

the desired information (patient posture) intact. Posture sensing is done using a single time sample or a time average (as explained in subsection 2.3.2). The signal transformations that apply to this are only a small subset of all signal transformations, thus the amount of ways to process this is relatively small. Testing showed (see figure 2.3) that the middle two sensors showed larger values when the test person was on his side. For this reason, the signal transformations that were investigated all focused on extracting this difference from the data. Those transformations were:

- Slope of a curve, that was fitted to a bar plot of the sensor data.
- Fourier transform (not timewise, but sensorwise).
- Double derivative of the curve, fitted to the data.
- Variance between the piezoresistive sensors.

### 2.3.3.1 Raw data

In its initial, raw form, the data can be visualized in two ways, either as a function of time, or as a function of sensor position (a single point in time). The former is plotted as a line plot and the latter as a bar plot in figure 2.5a and 2.5b respectively.




(a) Line plot.

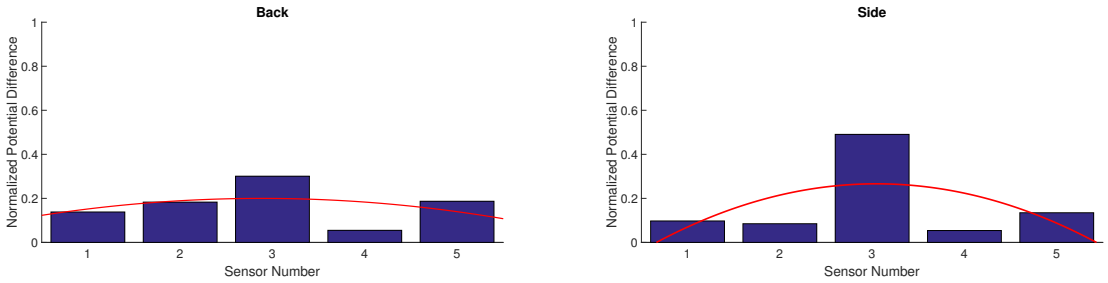
(b) Bar plot.

Figure 2.5: Raw data.

### 2.3.3.2 Polynomial fit

Each piezoresistive sensor behaves independently from the other sensors, however, due to the spatial distribution of the sensors they in effect behave as one collective system. To model this collective behaviour both visually and for later signal processing, a second-degree polynomial fit is applied to the sensor data at a specific point in time. This polynomial fit now contains not only the sensor data, but also information about the spatial variances between the sensors. Two examples of patient posture, the patient on their side and on their back, are visualized with this system in figure 2.6.

With this system alone, the case in which the patient is on their back is visually distinguishable from the same patient on their side in the previously mentioned example. This forms the basis for the signal processing in the following two sections.




(a) The patient is on their back.

(b) The patient is on their side.

Figure 2.6: Spatial sensor data with a second-degree polynomial fit: $f = c_0x^2 + c_1x + c_2$

### 2.3.3.3 Second derivative

There are several methods for evaluating the form of a second-order polynomial; one such method requires taking the (spatial) derivative of the polynomial. The derivative of this polynomial reveals the progression of its slope. Taking the derivative once again returns information on how rapidly the slope changes. Id est, the higher the absolute value of the second derivative, the steeper the polynomial curve is.

In the previous example, the second derivative is 0.0270 when the patient is on their back, while it is 0.0937 when the patient is on their side. Assuming this difference in values is consistent over a collection of spatial samples, these two postures can be distinguished with this algorithm. To test this hypothesis, a preliminary data set was processed with this algorithm; the results are shown in figure 2.7. When compared to a reference signal, the posture as determined by the algorithm is correct 74.7% of the time. The results of the algorithm do not yet fulfill the design requirement of 90% accuracy, thus further investigation in signal processing is required.

### 2.3.3.4 Fourier analysis

A different approach to distinguishing the polynomial curves is by investigating them in the frequency domain. By applying a Fourier transform, the fundamental properties of the curve change. The magnitude of the curve in the frequency domain is now, effectively, determined by the maximum width of the curve in the time domain [17]. If the curve starts and ends at the x-axis, and is trailed and followed by zeros (zero padding), an additional effect occurs, the lateral translation of the curve in the time domain now has no affect on the curve in the frequency domain. This is illustrated in figure 2.8 using three different curvatures as the signals.

Using the preliminary data set, the polynomial curve of every sample was transformed to the frequency domain, after which the maximum magnitude was retrieved. A histogram of the results can be found in figure 2.9. The results are identical to that of the second derivative method. This is due to the fundamental nature being investigated by both algorithms.

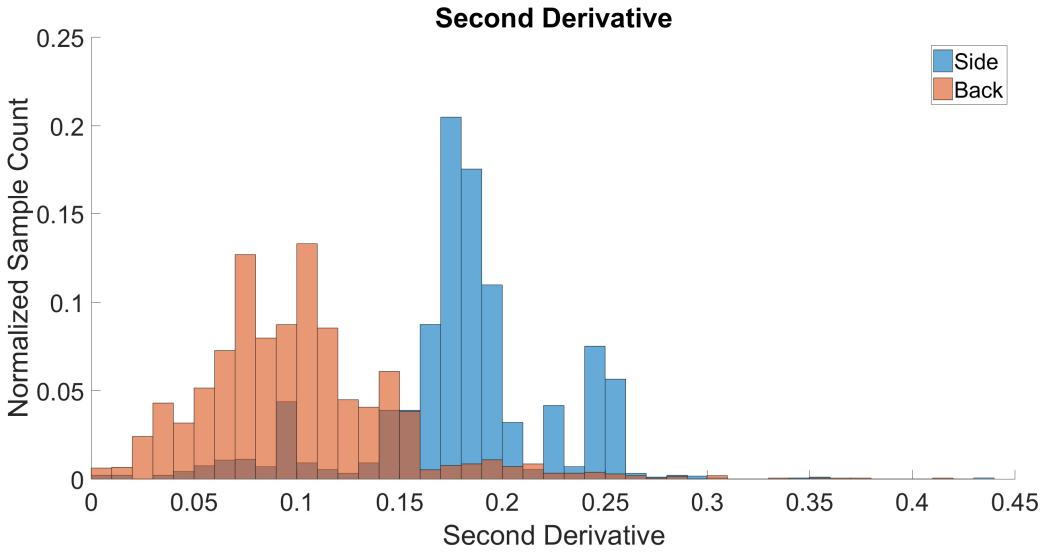
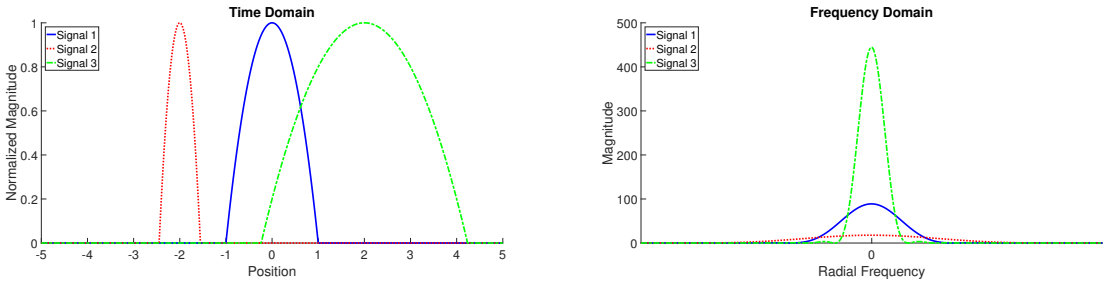



Figure 2.7: Distribution of the second derivative for two different postures, on the back, and on the side. Note: only four piezoresistive sensors were used for the preliminary data set instead of five.




(a) Three signals in the time domain.

(b) The same signals in the frequency domain.

Figure 2.8: Fourier analysis of three parabolic signals.

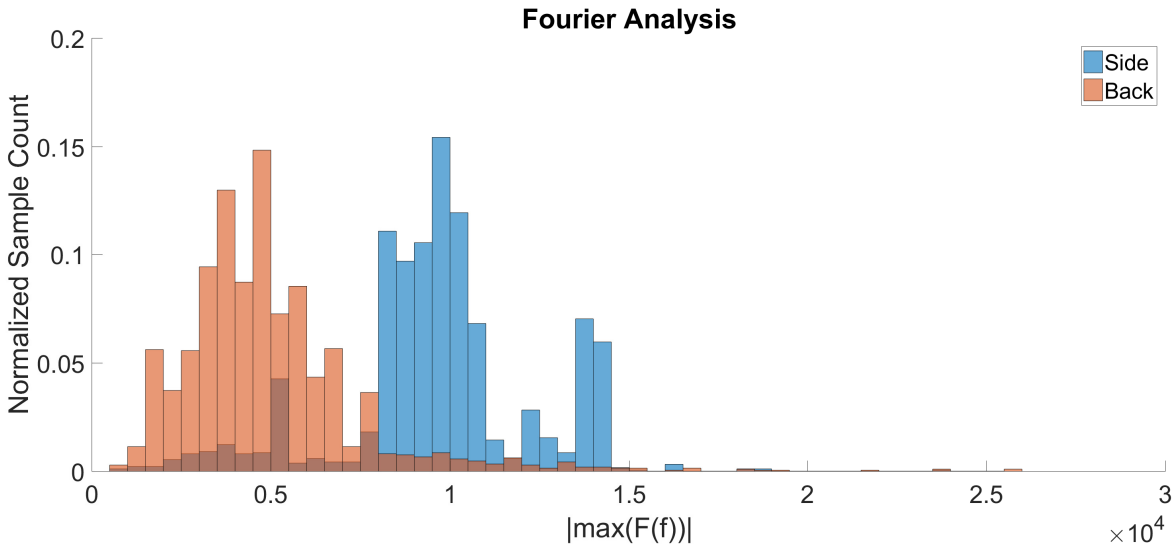



Figure 2.9: Distribution of the Fourier analysis for two different postures, on the back, and on the side. Note: only four piezoresistive sensors were used for the preliminary data set instead of five.

#### 2.3.3.5 Variance

The variance between the 5 piezoresistive sensors can be calculated according to (2.1), the general formula for the variance:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{2.1}$$

where for $x_i$ the value of the $i_th$ piezoresistive sensor is taken and for $\mu$ the average of the five piezoresistive sensors. To clarify, this variance is sensorwise, not timewise. As shown in figure 2.3, the middle two piezoresistive sensors showed a larger value when the person was on his side. When the person was on his back, all sensors were quite even. Thus, it was expected that this variance would be larger when the person was on his/her side in comparison to on his/her back.

To test this, the variance of datasets 1.1 through 1.3 was calculated, sorted by posture. Figure 2.10 shows the histogram of this variance. It can be seen that the variance was very distinct between the two postures. Based on this, it was expected that the variance would be useful in determining posture.

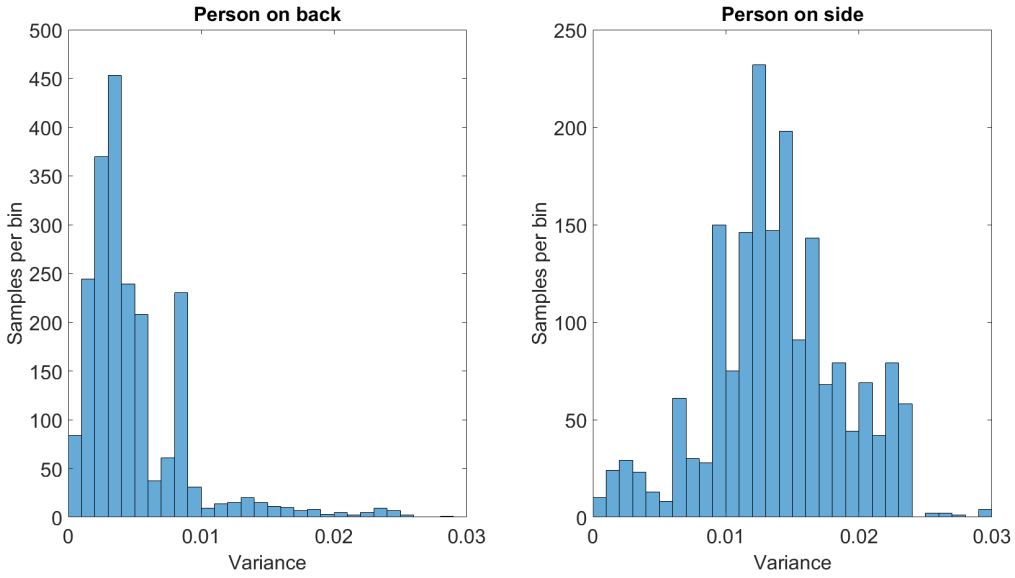



Figure 2.10: On the left, the variance of a person on his back, on the right, the variance of a person on his side.

### 2.3.4 Decision making

Whether or not the data has been transformed first, the algorithm must make a decision what posture this corresponds to. Two methods to do this where investigated in this subsection, both utilizing collected datasets to optimize the decision process. In the first, thresholds are set based on collected datasets. The second is a neural network.

#### 2.3.4.1 Threshold based on distance to mean

The following input is needed for this algorithm:

- A data sample X, of which the posture is to be determined
- A dataset B, comprised of data where people are on their back.
- A dataset S, comprised of data where people are on their side.

This algorithm is based on mean values and works as follows: In preparation, for each sensor, its mean in dataset B and S is calculated. Then, to determine the posture corresponding to sample X:

- For each sensor:
  - Calculate the distance between the sensor value in X and the mean of this sensor in B.

  - Calculate the distance between the sensor value in X and the mean of this sensor in S.

- Sum the distances to B.
- Sum the distances to S.
- If the distance to S is smallest, choose 'person on side'. Otherwise, choose 'person on back'.

For some signal transforms, there is only one variable instead of multiple sensors. For this case, an altered algorithm was made. The mean of the transformed signal (henceforth called *M*) is taken of dataset (B + S) . This M of (B+S) is taken as a decision threshold. The *M* of X is also taken. If this is larger than the threshold, the posture chosen is that of which the *M* is larger ( e.g. if the *M* of S is larger than the *M* of B, a *M* of X larger than the threshold should result in side being chosen)

#### 2.3.4.2 Neural networks

In all the algorithms discussed this far, loss of information occurred at some stage, in one of the following ways:

- In the algorithms based on a signal transformation of the sample that is to be checked, these signal transformations lead to a loss of information. (Even the Fourier transform, since this was done on a curve that was fit to the data).
- In the algorithm based on the distance to the mean, the sample is classified using a dataset. However, only the mean per sensor is utilized: a large loss of information occurs in the utilization of the dataset. For example, the relationship between the sensors is completely lost in this algorithm.

A neural network is better able to utilize a collected dataset in optimizing the decision method. A neural network is trained via an iterative process, in which the transfer of the system is optimized to predict the posture correctly. Often, the end result is merely a local maximum in performance, so repeating training may be necessary to obtain a good performance.

The neural networks were created using the Matlab neural network toolbox. This toolbox also handled subdivision of the data set into training, validation and testing sets. Initially, 70% of the data set was used for training, 15% was used for validation and 15% for testing. Also, the setting was to have each sample from the data set assigned randomly to one of the three sets. In training and testing the network on data sets 2.2 through 2.5, it was found out this subdivision caused a problem, namely that overfitting and overtraining had occurred. The results of these tests can be found in appendix D for reference purposes. Overfitting and overtraining are two ways in which the neural network adapts overly accurate to the training set, to the detriment of its performance for novel data [18]. Overfitting can occur when the amount of neurons in the hidden layer is large relative to the complexity of the problem. This description is not very measurable, but the fact that accuracies of 100% are reached by the neural networks here does appear to support the hypothesis that overfitting has occured. Overtraining is letting the neural network convergence to its optimal error rate for the training set, which once again is too specific to the data set. To prevent overtraining, validation is done during training: a separate data set is also checked against the neural network. This is used to stop training when the error rate for the validation set starts to decrease.

However, in this case, the samplewise random division of the dataset into the training, validation and testing sets did not result in distinct datasets, for the following reason. The datasets used for this training (2.2 through 2.5) had 10 time samples per measurement (which represent 1-5 seconds of someone lying still in a certain position). Thus, these samples are all of the same situation, with hardly any variation between them. The random division caused these 10 samples to be spread across the training, validation and testing, which meant that all of them received information about the exact same situation. Effectively, the same dataset was used in training, validation and testing. This not only led to the network being overtrained, it also made the testing seem great, until it was manually applied to dataset 2.1.

The neural network creation was adjusted so that the training, validation and testing sets each received a sequential block of data, so the datasets would actually be different. A total of 40 neurons was used for the hidden layer, lower amounts showed larger error rates, which seemed indicative of the dataset being too complex for the amount of neurons. Also, the subdivision of data between training, validation and testing was changed to 50%, 30%, and 20% respectively. The reason for increasing the percentage dedicated to testing was to increase the number of situations it was tested against, increasing confidence in the result. The thought behind increasing the validation size was that this would also increase the generalization of the network.

This only left 50% of an already small dataset for training, which would likely have a negative impact on the performance.

#### 2.3.4.3 Posture change detection

The aforementioned methods all attempt to determine the posture of the patient. An alternative to this would be to detect a change in posture, rather than the actual posture. For this, the sensor data is first processed by one of the previous methods, the derivative is then taken, followed by a peak-detection algorithm, and finally compared to the reference data. These steps are illustrated in appendix C. The result is plotted in figure 2.11 together with the reference signal.

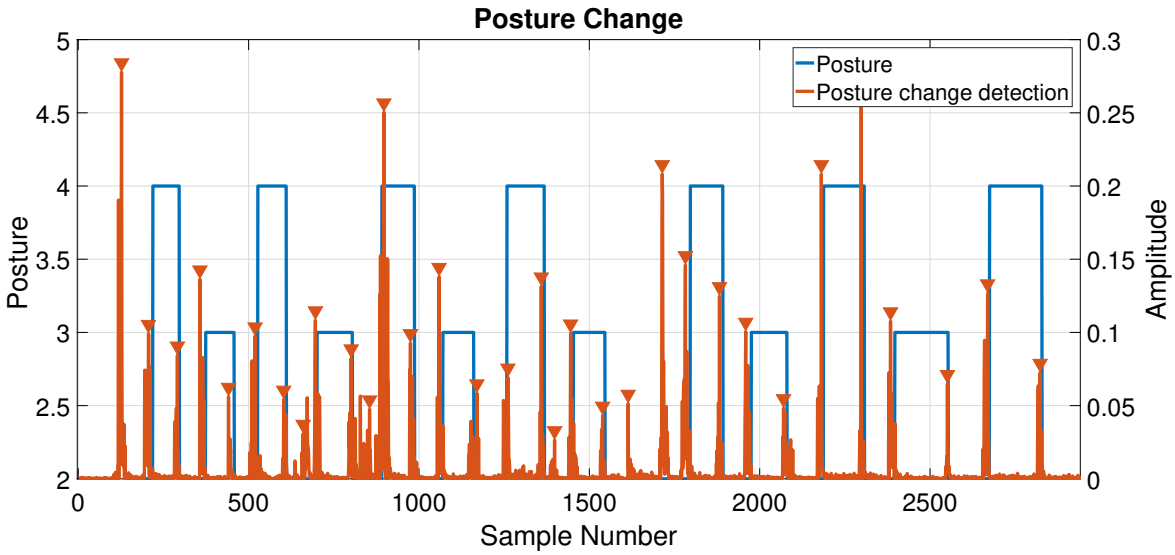



Figure 2.11: Every marked peak indicates a change in posture is detected. The reference signal is also plotted.

With 26 correct, 6 false positives, and no false negatives, the posture change algorithm seems to be effective. There is however one drawback, the algorithm often gives false positives in the case the patients moves slightly on the bed without changing posture. With a false positive, the system will not give an warning if the patient has not moved in a set amount of time, thus not preventing the original problem.

## 2.4 Heart rate and respiration rate sensing

Determining heart rate and respiration rate was an optional part of this project. Priority was given to creating a robust algorithm for posture detection and these stretch goals were not investigated beyond the literature study.

# 3 Implementation

## 3.1 Overview

With every prototype iteration, the designed algorithm has to be implemented, as just a mathematical model is not enough. Two implementation methods have been developed; one for usage with a live data feed, and the other for usage with a (or several) collected data set(s). In both methods, the data needs to be collected, preprocessed, run through the algorithm, compared to reference signals, and finally, the results need to be displayed with a graphical user interface (GUI).

## 3.2 Data acquisition

Before the algorithm is able to do any calculations, the data needs to be retrieved from a microcontroller and stored on the host computer (the computer that runs the algorithm). This is done with the python script in appendix E.1. With this, new sensor data is acquired every 0.0149 seconds, which is $\frac{1}{0.0149} = 67.11$ Hz. The newly acquired data is stored in a buffer on the host computer to ensure that the algorithm receives all the data points; this is necessary in the case that the algorithm takes longer to process the data than that it can retrieve the data. Next, the data needs to prepared for usage with the algorithm.

The algorithm acquires the sensor data together with the reference signals in array of 21 doubles. This is split into separate variables so that each situation can be categorized and investigates separately when testing the algorithm. The table in appendix B.1 gives an overview of the incoming data, together with their position within the data array.

## 3.3 Algorithm software

A host computer receives the data, after which it has to process this. The software used for this processing is a key aspect, as such, three programming languages were considered: Python, Matlab, and C/C++. An overview of the advantages and disadvantages of the programming languages is given in table 3.1.

| Programming language | Advantages | Disadvantages |
|---|---|---|
| Python | Steep learning curve, free, no unnecessary packages | Multiple GUI-programming environments |
| Matlab | Familiar environment, fast prototyping | Expensive for companies, slow in execution |
| C/C++ | Lightweight, good for embedded programming | Complex language (slow for prototyping) |

Table 3.1: Several advantages and disadvantages of three programming languages.

Due to the rapid prototyping nature of the project, Matlab was chosen as the main programming language for the algorithm. Nevertheless, for future iterations of the prototype, python and C/C++ still remain options. C/C++ will become especially useful in the case the algorithm has to run on an embedded system.

Regardless of the mathematical model used by the algorithm, the structure of the Matlab code is the same. An example of this structure can be found in appendix E.2.

## 3.4 GUI

In the final product, the algorithm will send the results to a GUI that will notify the nurse/caretaker if the posture of the patient needs to be changed. In the current prototyping phase, however, the GUI is designed to also display debug information to the observer/user. Figure 3.1 displays a possible implementation of the GUI. Here several key aspects are present. The nurse/caretaker has to be able to enter the patient's information (length, weight, maximum time in a certain posture, etc), view the history of the patient's posture, and most importantly, the GUI has to notify (possibly with an alarm) the nurse/caretaker if the patient has not changed posture in a set amount of time. Furthermore, the GUI has to have the option to display technical details, in the case debugging is required (for repairing and testing the product).

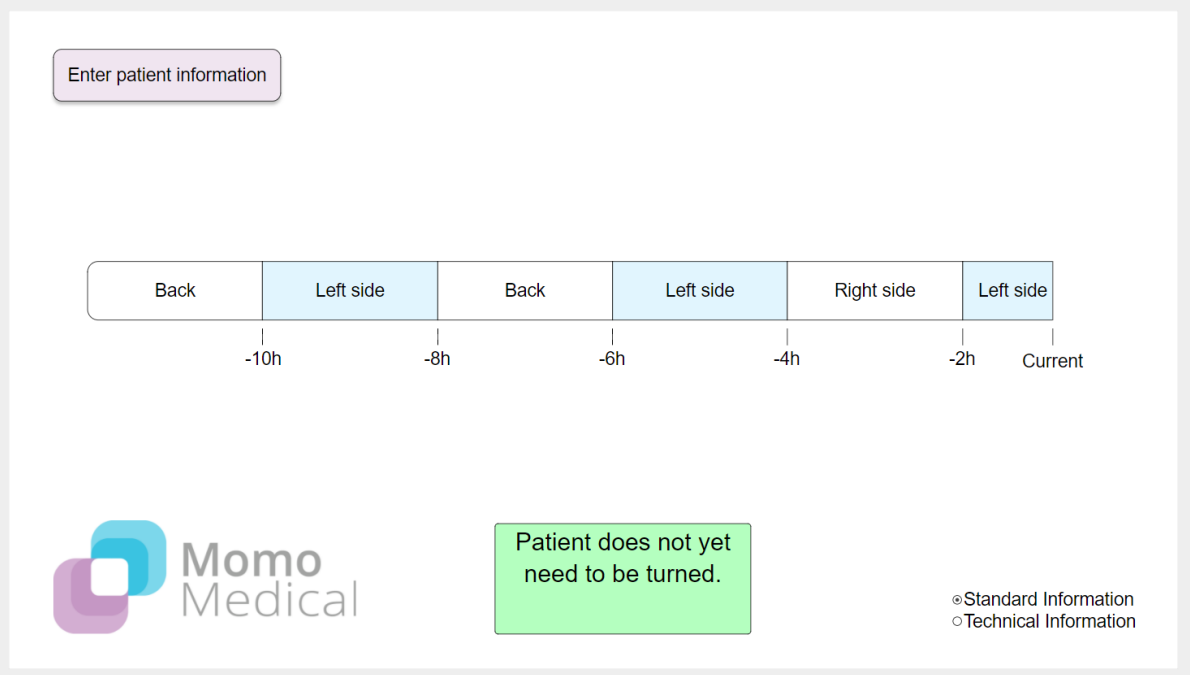



Figure 3.1: A possible implementation of the GUI for the final product.

# 4

# Evaluation

## 4.1 Overview

This chapter describes how the algorithms were evaluated. First, the process by which data was collected is described. Then, the testing of the patient detection and posture detection is described, since both of these required testing to verify if the design requirements were met. For the posture detection, multiple algorithms were tested.

The algorithms were tested non-real-time. No real advantage was found in testing the algorithms in real-time, while it did have a major disadvantage: it could only provide results for those algorithms that were implemented at the time of testing. Having to redo testing when new algorithms were made available or changes were made to existing ones would have been a major time investment. By choosing non-real-time testing, it allowed for the testing of many (variants of) algorithms on the same dataset and thus being able to compare them.

## 4.2 Data collection

To do non-real-time testing, datasets had to be created. The first round of data collection was done using Prototype 1, which was made by the sensor group. Prototype 1 had 4 piezoresistive sensors and no piezoelectric sensors. This prototype was placed under the mattress, which was located on top of a table. Two different people took part in this test. Datasets 1.1 is of the first test person, 1.2 and 1.3 are of the second test person. The test persons were located in the middle of the bed. Here, they turned to the various postures a number of times. Further details about these datasets can be found in appendix A. The sensor data was collected in Matlab via a Raspberry Pi. In the Matlab GUI, the actual/reference posture was set via a series of buttons. For this test, the algorithm based on the curve fit & slope, explained in 2.3.3, was ran in real-time. Both the reference and estimated posture were appended to the data.

The second round of data collection was done using Prototype 2, which was made by the sensor group. Prototype 2 had 5 piezoresistive sensors, as well as 2 piezoelectric sensors. This prototype was again placed under the mattress, which was now located in the hospital bed it was originally part of. Five different people took part in this test. They were measured in various combinations of the following factors:

- Varying the postures between left side, right side, back, stomach, rotated left 30°, rotated right 30°. The 30° postures are commonly used in pressure ulcer prevention, by using a special cushion designed for this.
- Varying the lateral position between the left, middle and right of the bed.
- Varying the bed between horizontal, raised to an angle of 30°, and raised to an angle of 45°.

The sensor data was collected in Matlab via a micro-controller. For each posture, 10 time samples of the piezoresistive and piezoelectric sensors were collected. The exact sampling rate was dependant on the communication speed between Matlab and a Raspberry Pi, but as an indication, this was 2 to 7 Hz, resulting in a measurement time of 1 to 5 seconds. This data collection was done by the test group, by also appending the pressure sensor data in their tests. This provided sensor data for many different situations, but in testing

the algorithms only a single, simple situation was used: the person is in the middle of the bed, the bed is in horizontal mode and no 30° posture is used.

## 4.3 Testing and results

### 4.3.1 Patient detection

A small, yet important part of the algorithm is to first detect if a patient is present. To test this, three data sets were given to the algorithm. The results are listed in table 4.1. The first data set gives a slightly worse result than data set 1.2 and 1.3. This was caused by a small mistake from the observer; half-way through testing, the patient left the bed for several seconds, but this was not indicated in the debug GUI. This leads to false positives (several every second, depending on the sample frequency). The results from the second and third data set are nearly at the required 99% accuracy. Here, most of the false positives and negatives are attributed to the slow response time of the observer (relative to the host computer) in combination with the small data set.

| Data set | Correct | False positives | False negatives | Percentage correct |
|---|---|---|---|---|
| 1.1 | 1631 | 83 | 0 | 95.16 |
| 1.2 | 1583 | 3 | 28 | 98.08 |
| 1.3 | 1315 | 17 | 0 | 98.72 |

Table 4.1: Test results of the patient detection algorithm.

### 4.3.2 Posture detection

Table 4.2 shows the results of determining posture (only back and side) using various algorithms on data sets 1.1 through 1.3. For the merged data sets, only the neural network meets the 90% accuracy requirement. When the neural network is trained with more data, performance improves. This neural network chose between the states 'no person', 'person on back' and 'person on side'. Of the 7% error, 62.9% of all errors were false negatives of the 'no person' state. A possible explanation is that the bed was only without person for a very small amount of the testing time, which caused the neural network to not be properly trained for it. Another issue is that changing the reference state is instantaneous, while in practice, entering and leaving the bed took a couple of seconds, after which the mattress also took a few seconds to regain its shape. This may have reduced the accuracy of the reference.

| Percentage of accurate decisions | | | | | |
|---|---|---|---|---|---|
| | Method | | | | |
| Dataset | Neural Network | Fourier | Fourier w/ Zero Padding | Second Derivative | Variance |
| 1.1 | 80.7% | 65.2% | 47.2% | 65.3% | 89.5% |
| 1.2 | 90.7% | 77.9% | 64.1% | 77.7% | 90.0% |
| 1.3 | 95.9% | 93.2% | 86.6% | 93.3% | 92.9% |
| Average of above | 88.5% | 77.6% | 64.3% | 77.6% | 90.6% |
| Merged 1.1, 1.2, 1.3 | 93.0% | 77.4% | 61.0% | 75.4% | 88.7% |

Table 4.2: Percentage of accurate decisions for multiple algorithms, by data set.

The three initial data sets (1.1, 1.2, and 1.3) returned various results for the different algorithms, ranging from as low as 61.0% to as high as 93.0% (for the merged data sets). Especially the variance test and the neural network display promising results, with accuracy close, and even above the required accuracy.

To extend these results past the preliminary data sets, five extensive data sets were created, with relevant reference signals (see appendix B). Each data set consists of a different test patient walking through the same procedure (a predetermine schematic of the various postures to be assumed by the test patient), as described in section 4.1. These data sets have been subjected to the various algorithms, and their accuracy is displayed in table 4.3. Note that only two postures were distinguished, the patient on their back and on their side.

With the exception of the variance method on data set 2.2, none of the methods come close to the required accuracy of 90%. This may be due to inconsistencies in the mechanical working of the domes on the sensors,

| Percentage of accurate decisions | | | | |
|---|---|---|---|---|
| | Method | | | |
| Dataset | Fourier Test | Fourier w/ Zero Padding | Second Derivative | Variance |
| 2.1 | 57.1% | 64.3% | 57.1% | 51.8% |
| 2.2 | 41.7% | 40.0% | 43.3% | 88.8% |
| 2.3 | 58.3% | 55.8% | 58.3% | 66.4% |
| 2.4 | 66.7% | 50.0% | 66.7% | 62.2% |
| 2.5 | 50.0% | 50.0% | 50.0% | 65.5% |
| Average of above | 54.8% | 52.4% | 55.1% | 66.5% |
| Merged data sets | 41.8% | 55.2% | 42.0% | 56.3% |

Table 4.3: Percentage of accurate decisions for multiple algorithms, by data set.

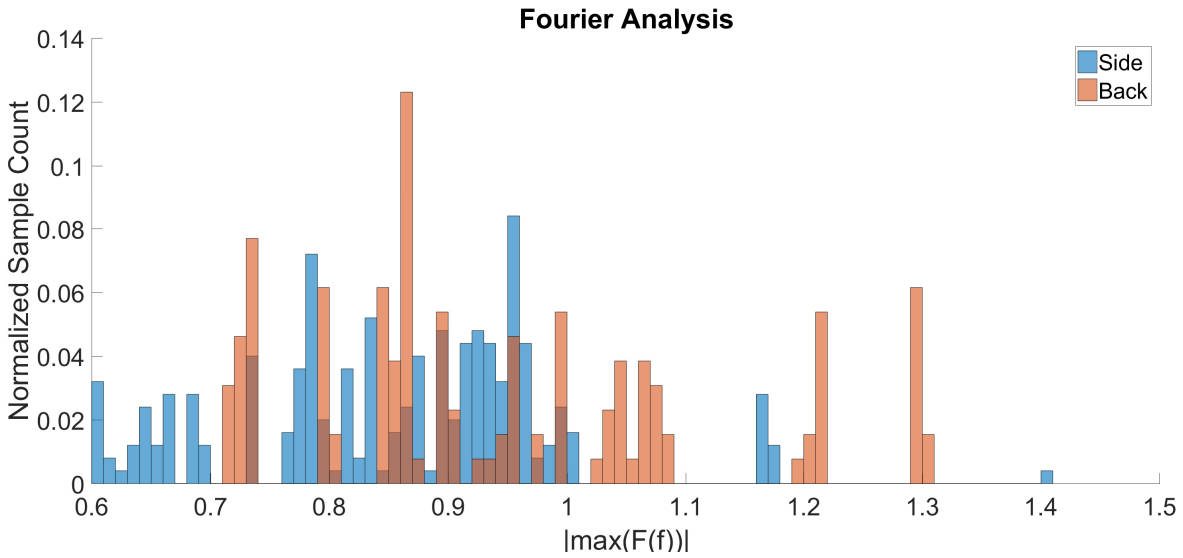



Figure 4.1: A histogram of the Fourier analysis of datasets 2.1 through 2.5.

in combination with the flexing of the base plate when exerting pressure in the middle. This flexing was not a factor in the preliminary tests, due to the flat surface on which the mattress lay (in tests 2.1-2.5 the hospital bed was used, in tests 1.1-1.3 only the hospital mattress was used, on top of a flat table).

The low accuracy can visually be explained by the histogram in figure 4.1. Here, it is clear that both subsets, back and side (based on the reference signal), overlap. This is far from optimal for the algorithm, as now it can barely distinguish the two categories.

The neural network on the other hand, displays more promising results, with 93.0% accuracy with the preliminary data sets, and even 95.9% with the new data sets. Table 4.4 contains these results. Here, the neural network was tested with two cases, one in which it received the raw signal data, and the other in which it receive the precalculated variance.

| Percentage of accurate decisions | | | | |
|---|---|---|---|---|
| | Datasets | | | |
| | 1.1 through 1.3 | | 2.1 through 2.5* | |
| Decision method | Raw data | Variance | Raw data | Variance |
| Mean as threshold | 83.3% | 87.5% | 62.0% | 53.9% |
| Neural network | 93.0% | 94.1% | 95.9% | 69.4% |

Table 4.4: Percentage of accurate decisions for combinations of signal processing and decision methods.
* For comparison to the other datasets, only the test situations where the person was in the middle of the bed, with the bed in horizontal position were included. The 30° rotation was also left out.

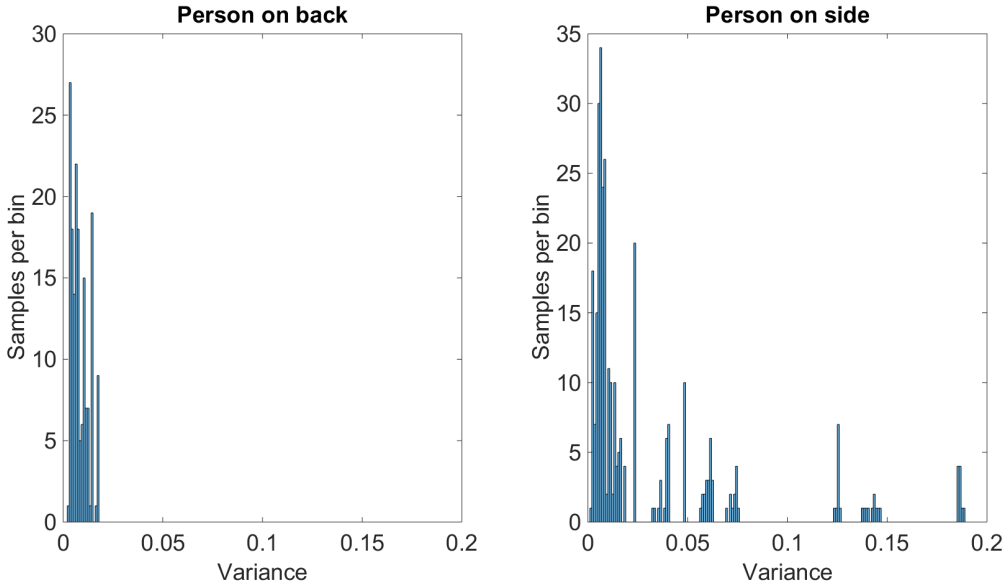



Figure 4.2: A histogram of the variance of datasets 2.1 through 2.5. On the left, the person is on his back, on the right, the person is on his side

For datasets 1.1 through 1.3, the variance proved a useful indicator, while for datasets 2.1 through 2.5, its performance hardly beats choosing randomly. Figure 4.2 shows the variance for back and side of datasets 2.1 through 2.5. It can be seen that the back and side cases are not nearly as distinct as they were for dataset 1.1 through 1.3, so the poor performance can hardly be called surprising.

Once again, only the neural network meets the design requirement of 90% accuracy.

Now that the neural network achieved the goal of distinguishing back and side, another test was done using datasets 2.1 through 2.5, in which the neural network was trained to also distinguish left side and right side. This neural network achieved 95.9% accuracy for the training set, 93.2% accuracy for the validation set and 90.8% for the test set.

# 5 Conclusion

## 5.1 Assessment

### 5.1.1 Human input constraints

**The algorithm should operate without requiring human input. The only exception is the entering of patient information, such as weight and height.** In the prototype, it is still necessary to manually start and stop operation. In the final product, this will not be the case. Thus, this design requirement is met. The entering of height and weight is not required at the moment.

### 5.1.2 Patient detection

**Design requirement: The algorithm must detect whether there is a person on the bed or not. This decision must be correct 99% of the time.**

This goal was not achieved, since only 98% accuracy was obtained.

### 5.1.3 Posture detection

**Design requirement: The algorithm must distinguish the following postures. This decisions must be correct 90% of the time.**

- **The person is on his/her back.**
- **The person is on his/her left side.**
- **The person is on his/her right side.**

The neural network distinguished these with 90.8% accuracy, so this design requirement is met.

## 5.2 Recommendations

For the remainder of this project, the following is planned:

- To improve the patient detection. The threshold for the patient detection was not optimized yet, using a data-based threshold might help, or possibly integrating it into the neural network.
- To integrate the neural network into the real-time GUI.
- To collect additional data to train the neural network.
- To obtain a better understanding of neural networks.

One of the following may still be investigated in the remainder of this project:

- Including patient height and weight as an input for the neural network may improve its performance, this might be worth investigating.

- Investigate the use of a recurrent neural network. This type of network has internal feedback and can thus be used to process a dynamic time series. With this, the data collected during the movement of the patient can also be used in identifying the posture.

Improvements that need to be made if this project is to be incorporated into the final product: In the final product Momo Medical intents to produce, only a microcontroller will be available per bed, with possibly a single server per hospital. This will put constraints on computational complexity. Also, communication bandwidth will be limited. It is proposed that the determining of the patient posture is only done periodically, perhaps once every minute, based on an average of the piezoresistive sensors during this time. This means the neural network can either run on the server with only a small amount of data to it. By reducing the frequency of recomputation, it may also be possible to have the neural network running on the microcontroller. A possible problem with this is that movement during the time window over which the piezoresistive sensors are averaged would result in inaccurate results. A possible way to prevent this is to detect this movement and reset the averaged sensor value.

Furthermore, a significant amount of data will have to be gathered to prepare the neural network for the diverse situations that occur in reality. Using the reference signal setup made by the test group, this can be automated.

# A

# Datasets

| Data set | Name | Sex | Height (m) | Weight (kg) |
|---|---|---|---|---|
| 1.1 | Andy | Male | 1.76 | 84 |
| 1.2 | Douwe | Male | 1.91 | 70 |
| 1.3 | Douwe | Male | 1.91 | 70 |
| 2.1 | Thomas | Male | 1.92 | 75 |
| 2.2 | Andy | Male | 1.76 | 84 |
| 2.3 | Danny | Male | 1.85 | 86 |
| 2.4 | Bas | Male | 1.81 | 70 |
| 2.5 | Douwe | Male | 1.91 | 70 |

Table A.1: Relevant information about the person on the bed in the datasets.

# B

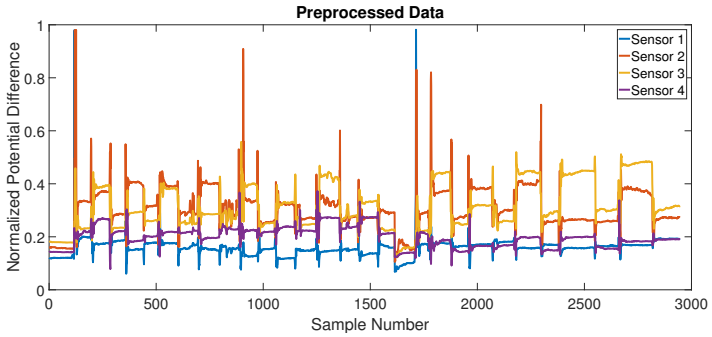
# Reference signals

| Data column(s) | Signal name | Range | Signal description |
|---|---|---|---|
| 1 | Sample | 1-infinity | The sample number |
| 2-6 | Piezoresistive | 0-3.3 V | The potential difference of the five piezoresistive sensors |
| 7-9 | Piezoelectric | 0-3.6 V | The potential difference of the three piezoelectric sensors* |
| 10 | Heart rate | 0 or 1 | Is 1 if a heart pulse is detected, otherwise 0 |
| 11 | x-coordinate | 0-1 | The x-coordinate position of the accelerometer |
| 12 | y-coordinate | 0-1 | The y-coordinate position of the accelerometer |
| 13 | z-coordinate | 0-1 | The z-coordinate position of the accelerometer |
| 14 | Pitch | 0-90 degrees | The pitch of the gyroscope |
| 15 | Roll | -180-180 degrees | The roll of the gyroscope |
| 16 | Person | 2-7 | Test patient number |
| 17 | Posture | 2-7 | The posture of the patient according to an observer |
| 18 | Position | 2-4 | The side of the bed the patient is on (left, center, right) |
| 19 | Angle | 0, 30, 45 degrees | The pitch according to an observer |
| 20 | Multiple People | 0 or 1 | Whether there is one or several people on the bed |
| 21 | Miscellaneous | 0 or 1 | Is 1 if there is a special condition, otherwise 0 |

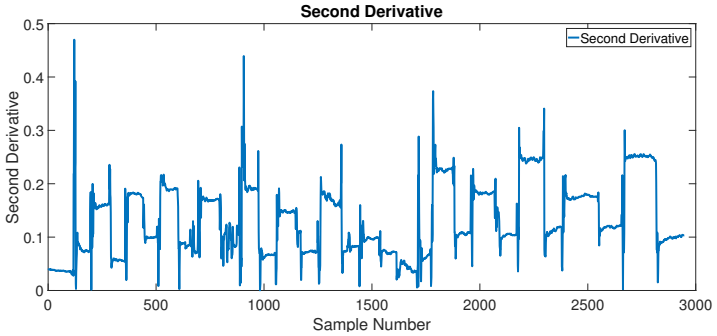
Table B.1: The signals in the data line, including reference signals. *Only two piezoelectric sensors were used (columns 7 and 8).

# C

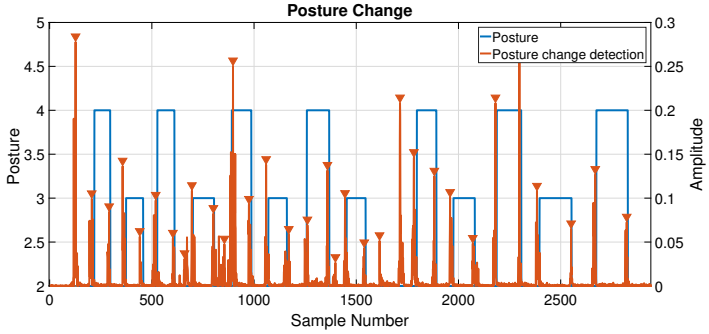
# Posture change




(a) The preprocessed raw data to be evaluated.




(b) The second derivative of every sample, plotted against the sample number.




(c) Every marked peak indicates a change in posture is detected. The reference signal is also plotted.

Figure C.1: Preliminary test of posture change detection algorithm.

# D

# Old results

Tables D.1 and D.2 show results that turned out to be erroneous. The training, validation and testing set were not properly separated, causing overtraining to occur.

## D.1

Some additional testing was done on neural networks. It was tested if the neural network could also handle more complex situations. It was tested if distinguishing left side and right side was also possible. Finally, it was tested what the impact of the amount of hidden neurons in the network was (in previous tests, 100 hidden neurons were used). Table D.1 shows the results of these tests. Dataset 2.2 through 2.5 were used for training, validation and testing by dividing the samples randomly in 70/15/15% proportion, as described before. The error percentage is based on the test data. The initial thought was that it can be seen that the neural network can handle all complexities introduced. For the most complex case, more than 10 neurons are required to obtain the required accuracy.

| Error percentage for data from the same dataset | | | | |
|---|---|---|---|---|
| | | Number of hidden neurons | | |
| Postures determined | Situations included | 10 | 100 | 1000 |
| Back or side | Simple situation only | 0% | 0% | 0% |
| | All situations measured | 0% | 0% | 0% |
| Back, left side or right side | Simple situation only | 0% | 0% | 0% |
| | All situations measured | 19.5% | 2.1% | 1.0% |

Table D.1: The error percentage for various neural networks. The dataset was comprised of datasets 2.2 through 2.5. A sample of the dataset was saved for testing. The simple situation is the same as used used in table 4.2: person in middle, bed is horizontal, no 30° rotation.

| Error percentage for data from a different dataset | | | | |
|---|---|---|---|---|
| | | Number of hidden neurons | | |
| Postures determined | Situations included | 10 | 100 | 1000 |
| Back or side | Simple situation only | 24.5% | 41.8% | 27.3% |
| | All situations measured | 46.9% | 44.9% | 46.9% |
| Back, left side or right side | Simple situation only | 19.1% | 23.6% | 33.6% |
| | All situations measured | 46.4% | 39.8% | 36.4% |

Table D.2: The error percentage for various neural networks. Dataset 2.1 was used for testing. The simple situation is the same as used used in table 4.2: person in middle, bed is horizontal, no 30° rotation.

# E

# Code

## E.1 Data acquisition

```
mbed LPC1768 microcontroller.]
import csv
import sys
import serial
import time
filename = 'Resistive_Output.csv'
port = 'COM3'
baud = 921600
lpc1768 = serial.Serial(port, baud, timeout=0.0149)
i = 0

data = []
data.append([])
data.append([])


while True:
    serialstring = lpc1768.readline() # read all characters in buffer
    serialstring = serialstring.decode('utf-8')
    print (serialstring)
    splitstring = serialstring.split(',')

    for _ in range(10):
        try:
            data[i].append(splitstring.pop(0).strip())
        except Exception:
            #data[i] = [0,0,0,0,0,0]
            continue

    try:
        with open(filename, 'a', encoding='utf-8', newline='') as csvfile:
            output = csv.writer(csvfile, lineterminator='\n')
            output.writerows(data)
            i=0
            data = []
            data.append([])
            data.append([])
    except Exception:
        i += 1
        continue
```

Listing E.1: Python code for data extraction from the mbed LPC1768 microcontroller. The data sheet of the microcontroller is available at [19].

## E.2 Matlab example

```
%example code

%clear old data
clear all;
clc;

%get data
data=[];
filePath = 'C:\Users\ExamplePath\ExampleDataFile.txt';
data = load(filePath);
delete(filePath); %to ensure the data file does not become to large

%preprocess data (the reference signals are not included in this example)
sample = data(:,1); %select the first column
senRes = data(:,2:6)./(8000); %select the second to 6th column for the piezoresistive data
senEle = data(:,7:9)./(8000);

%here the algorithm determines the posture (via different methods)

%some off the used functions in the algorithm
polyfit(); %for determining the 2nd-order polynomial fit
diff(); %for calculating the second derivative of the polynomial
fft(); %for calculating the fourier transform of the polynomial
var(); %gets the variance of the sensor data

%the determined posture is compared to the reference posture

%the result is passed to the GUI
```

Listing E.2: The structure of the Matlab code for the algorithm.

# Bibliography


[1] H. Incorporated., *Pressure ulcer stages*. [Online]. Available: https://myhealth.alberta.ca/Health/pages/conditions.aspx?hwid=zm2442.

[2] J. G. Hospital, *Pressure ulcer prevention*. [Online]. Available: http://www.jgh.ca/en/qiPressureUlcerPrevention?mid=ctl00_LeftMenu_ctl00_TheMenu-menuItem008.

[3] J. L. Severens *et al.*, "The cost of illness of pressure ulcers in the netherlands", *Advances in Skin Wound Care*, vol. 15, pp. 72–77, 2.

[4] J. Posnett and P. J. Franks, "The burden of chronic wounds in the uk", *Nursing Times*, vol. 104, pp. 44–45, 3 2008.

[5] Z. Moore and P. Price, "Nurses' attitudes, behaviours and perceived barriers towards pressure ulcer prevention", *Journal of Clinical Nursing*, vol. 13, no. 8, pp. 942–951, Nov. 2004, ISSN: 1365-2702. DOI: 10.1111/j.1365-2702.2004.00972.x. [Online]. Available: https://dx.doi.org/10.1111/j.1365-2702.2004.00972.x.

[6] (2017). Bedsores (pressure ulcers) symptoms and causes, [Online]. Available: http://www.mayoclinic.org/diseases-conditions/bed-sores/symptoms-causes/dxc-20315617.

[7] T. Deflor *et al.* (2004). Richtlijn decubituspreventie, [Online]. Available: http://www.decubitus.be/richtlijnen/nl/houdingen.htm.

[8] T. Defloor *et al.*, "The effect of various combinations of turning and pressure reducing devices on the incidence of pressure ulcers", *International Journal of Nursing Studies*, vol. 42, no. 1, pp. 37–46, 2005, ISSN: 0020-7489. DOI: http://dx.doi.org/10.1016/j.ijnurstu.2004.05.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020748904000938.

[9] R. Halfens *et al.*, *Rapportage resultaten landelijke prevalentiemeting zorgproblemen*. Universitaire Pers Maastricht, 2015, pp. 27–44, ISBN: 978-94-90411-08-4.

[10] M. Holtzman, R. Goubran, and F. Knoefel, "Motion monitoring in palliative care using unobtrusive bed sensors", in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2014, pp. 5760–5763. DOI: 10.1109/EMBC.2014.6944936.

[11] (). Boditrak, [Online]. Available: http://www.boditrak.com/.

[12] M. B. Pouyan *et al.*, "A pressure map dataset for posture and subject analytics", in *2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, Feb. 2017, pp. 65–68. DOI: 10.1109/BHI.2017.7897206.

[13] R. Behrendt *et al.*, "Continuous bedside pressure mapping and rates of hospital-associated pressure ulcers in a medical intensive care unit", *American Journal of Critical Care*, vol. 23, no. 2, pp. 127–133, 2014. DOI: 10.4037/ajcc2014192. eprint: http://ajcc.aacnjournals.org/content/23/2/127.full.pdf+html. [Online]. Available: http://ajcc.aacnjournals.org/content/23/2/127.abstract.

[14] T. Y. *et al*, "Multiple-input/multiple-output characteristics of piezo devices and an application for triage", *IEEE Sensors Journal*, vol. 17, no. 5, pp. 1434–1442, Mar. 2017, ISSN: 1530-437X. DOI: 10.1109/JSEN.2016.2642992.

[15] M. Brink *et al.*, "Contact-free measurement of heart rate, respiration rate, and body movements during sleep", *Behavior Research Methods*, vol. 38, no. 3, pp. 511–521, 2006, ISSN: 1554-3528. DOI: 10.3758/BF03192806. [Online]. Available: http://dx.doi.org/10.3758/BF03192806.

[16] *Flexiforce sensors user manual*, Rev. 1, Tekscan, Inc, Dec. 2010.

[17] B. Girod *et al.*, *Signals and systems*. John Wiley & Sons, 2001, pp. 204–205, ISBN: 978-0-471-98800-7.

[18] Tetko *et al.*, "Neural network studies. 1. comparison of overfitting and overtraining", *Journal of chemical information and computer sciences*, vol. 35, no. 5, pp. 826–833, 1995.

[19] *Nxp lpc176x data sheet*, NXP, May 2017. [Online]. Available: http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf.